

LIVRODEATAS



SI 22

# SIMPÓSIO DE ENGENHARIA INFORMÁTICA

---

**Departamento de Engenharia Informática**  
Instituto Superior de Engenharia do Porto  
*7 de dezembro de 2022*

## Livro de Atas do SEI'22

---

**TÍTULO:**

Simpósio de Engenharia Informática 2022: Livro de Atas

**COORDENAÇÃO:**

Dulce Mota, Emanuel Silva, Joaquim Filipe Santos, José Marinho, Marílio Cardoso, Nuno Morgado, Piedade Carvalho, Sérgio Moreira

**EDITORA:**

Instituto Superior de Engenharia do Porto (ISEP) – P.Porto

**DESIGN / LAYOUT:**

Gabinete de Design e Multimédia do ISEP

**LOCAL / DATA:**

ISEP / 7 de dezembro de 2022

**ISBN:**

978-989-35087-0-1



Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-NãoComercial-SemDerivações 4.0 Internacional.

Instituto Superior de Engenharia do Porto (ISEP)

Rua Dr. António Bernardino de Almeida, 431

4249-015 Porto

t. +351 22 83 40 500

[www.isep.ipp.pt](http://www.isep.ipp.pt)

<http://sei.dei.isep.ipp.pt/sei22>

©2022 pelo ISEP – P.Porto

## Agradecimentos

A participação de todos no Simpósio é motivo de satisfação para a Direção do DEI. Começamos por agradecer aos Órgãos de Gestão do ISEP, em específico, à Presidente do Conselho Técnico-Científico, Ana Almeida Batista Figueiredo, pelo apoio e presença na abertura do evento. Um agradecimento também muito especial à Direção do DEI, e em particular ao seu Diretor, Joaquim Filipe dos Santos, pelo suporte a toda a infraestrutura do evento. Agradecemos a todos os oradores que contribuíram com a sua reconhecida experiência e inegável valor científico na partilha de conhecimento pela comunidade.

Uma palavra de agradecimento aos autores, aos membros da Comissão Técnico-Científica, aos participantes, e a todos que colaboraram e ajudaram com todo o seu entusiasmo e empenho na realização deste evento.

Finalmente, agradecemos à Celfocus, pelo apoio concedido que contribuiu decisivamente para a realização desta edição do SEI.

# Conteúdo

<b>Agradecimentos</b> . . . . .	<b>3</b>
<b>O Simpósio - Visão</b> . . . . .	<b>7</b>
<b>Comissões</b> . . . . .	<b>9</b>
<b>Suporte Técnico e Secretariado</b> . . . . .	<b>12</b>
<b>Oradores Convidados</b> . . . . .	<b>13</b>
<b>Artigos</b> . . . . .	<b>19</b>
Aplicação para crianças diagnosticadas com espectro de autismo Joana Portugal, João Santos, Marco Malta, Jorge Lima, Te- resa Terroso . . . . .	19
Aplicação <i>Web</i> para Consulta de Dados de Sistemas SIRTEC Marta Almeida, Luís Fernandes, Vítor Cardoso, Pedro Reis, Sandra Aires, António Sousa e Fernando Carvalho . . . . .	28
Aplicações <i>Web</i> para Monitorização e Controlo de Autómatos Bruno Maia, Teresa Canedo, Vítor Cardoso, Pedro Reis, San- dra Aires, António Sousa e Fernando Carvalho . . . . .	36
Base de Dados e Integração com soluções de <i>Data Hologram</i> Rui Soares e F. Jorge Duarte . . . . .	44
Channel TV: Power Platform CRM System Nuno Soares e Emanuel Silva . . . . .	52
Computação em nuvem: Infraestrutura para o desenvolvimento co- laborativo de Aplicações António Fernandes e F. Jorge Duarte . . . . .	60
Computational Simulation Approaches to Low Pass Filtering Fernando Carvalho, Vítor Cardoso, Pedro Reis, Sandra Aires, António Sousa e Pedro Guedes . . . . .	68

Computerized Physician Order Entry orientado à monitorização de sinais vitais Diogo Soares, Ricardo Almeida, Nuno Escudeiro, André Mota, Rita Fontes, Laura Saldanha, Catarina Caçote . . . . .	76
Coronabot: Providing COVID-19 Information with a Chatbot in European Portuguese Ricardo Ferreira and Luís Coelho . . . . .	84
Designing The Talent Marketplace Of The Future E. Luciano Zickler Paz, Susana Nicola, and Nuno Bettencourt	92
Eye Controlled Virtual Keyboard for Motor Impaired Users Aya Marshaha, Daniela Oliveira, Teresa Ferraz, Tomas Sousa, Luis Coelho and Sara Reis . . . . .	101
FARO InTouch Connect: Desenvolvimento de um Gestor de Metadados Afonso Correia e Alexandre Gouveia . . . . .	108
Ferramenta de ensino/aprendizagem do método de Monge João Romano Barbosa and João Paulo Pereira . . . . .	116
Geração Automática de Código-Fonte de um Servidor <i>GraphQL</i> através das Especificações de uma API REST André Botelho and Fátima Rodrigues . . . . .	124
Gestão de Propostas Comerciais Marly Juliana Pimpita . . . . .	132
HeltiSolar - Sistema CRM e Power Platform Simão Gomes, Ricardo Almeida and José Pinto . . . . .	140
Interface para a análise de um sinal ECG André Monteiro, Duarte Guimarães, Patrícia Carvalho, Joaquim Alves, Gina Vilão . . . . .	148
Migração de conteúdos em memória para Redis Ricardo Coelho e Emanuel Silva . . . . .	156
Sistema de apoio à gestão da Porto Design Factory Joel Costa, Constantino Martins e Jorge Duarte . . . . .	164
SMARTHEALTH - Sistema de Recomendação para Cuidados de Saúde Personalizados ao Longo da Vida João Santos, Constantino Martins, and Luiz Faria . . . . .	172
Software as a Service: Modelo de Distribuição e Comercialização de Software Aplicado a Ingest Alexandra Branco e Nuno Bettencourt . . . . .	180
Test Results Framework Miguel Cabral, Pedro Silva, Vítor Cardoso, Pedro Reis, Sandra Aires, Fernando Carvalho and António Sousa . . . . .	188

Índice de Autores . . . . .196

## O Simpósio - Visão

Reconhecendo a importância do artigo científico na divulgação do conhecimento e práticas da engenharia, pretende-se proporcionar aos participantes, mais especificamente, aos estudantes dos diversos níveis do Ensino Superior – Licenciatura, Mestrado e Doutorado – em Engenharia Informática ou áreas afins, a aquisição de competências na escrita científica, partilhando trabalhos técnico-científicos de relevância desenvolvidos no âmbito do seu percurso académico e/ou profissional. Durante as sessões, os estudantes têm oportunidade de apresentar e expor os seus trabalhos académicos, os resultados alcançados e o impacto produzido.

Pretende-se também continuar a fomentar o encontro entre estudantes e profissionais da área da Engenharia Informática, para partilharem, debaterem e refletirem em torno da investigação, desenvolvimento e práticas no domínio da Engenharia Informática.

A 4<sup>a</sup> edição do Simpósio de Engenharia Informática (SEI'22) decorreu no dia 7 de dezembro de 2022 no formato híbrido.

Os temas referência do Simpósio foram os seguintes:

- Análise e Especificação de Sistemas de Informação
- Arquitetura e Desenvolvimento de Software
- Comércio Eletrónico
- Computação Gráfica, Visão e Aplicações
- Ferramentas Computacionais de Trabalho Colaborativo
- Ferramentas Eletrónicas no Ensino e Aprendizagem
- Inteligência Artificial
- Interação Pessoa-Computador
- Internet das Coisas

- Jogos Sérios
- Modelação de Simulação
- Qualidade de Software
- Recuperação de Informação
- Redes de Computadores
- Robótica e Automação Industrial Inteligente
- Segurança da Informação
- Sistemas Críticos
- Sistemas de Apoio à Decisão
- Técnicas e Paradigmas de Programação

# Comissões

## Comissão Organizadora

---

Dulce Mota	Instituto Superior de Engenharia do Porto
------------	---

---

Emanuel Silva	Instituto Superior de Engenharia do Porto
---------------	---

---

Joaquim Santos	Instituto Superior de Engenharia do Porto
----------------	---

---

José Marinho	Instituto Superior de Engenharia do Porto
--------------	---

---

Marílio Cardoso	Instituto Superior de Engenharia do Porto
-----------------	---

---

Nuno Morgado	Instituto Superior de Engenharia do Porto
--------------	---

---

Piedade Carvalho	Instituto Superior de Engenharia do Porto
------------------	---

---

Sérgio Moreira	Instituto Superior de Engenharia do Porto
----------------	---

---

## Comissão Técnico-Científica

A. Augusto Sousa	Faculdade de Engenharia da Universidade do Porto
Alexandre Gouveia	Instituto Superior de Engenharia do Porto
Álvaro Rocha	Instituto Superior de Economia e Gestão
Ana Madureira	Instituto Superior de Engenharia do Porto
Ana Paula Camarinha	Instituto Superior de Contabilidade e Administração do Porto
Ana Raquel Faria	Instituto Superior de Engenharia do Porto
Anabela Serrano	Instituto Superior de Contabilidade e Administração do Porto
Bertil Marques	Instituto Superior de Engenharia do Porto
Davide Carneiro	Escola Superior de Tecnologia e Gestão
Dulce Mota	Instituto Superior de Engenharia do Porto
Emanuel Silva	Instituto Superior de Engenharia do Porto
Fátima Rodrigues	Instituto Superior de Engenharia do Porto
Fernando Duarte	Instituto Superior de Engenharia do Porto
Isabel Sampaio	Instituto Superior de Engenharia do Porto
Joaquim Filipe Santos	Instituto Superior de Engenharia do Porto
José Marinho	Instituto Superior de Engenharia do Porto
Luís Alves	Instituto Politécnico de Bragança
Luiz Faria	Instituto Superior de Engenharia do Porto
Manuela Raposo Rivas	Universidade de Vigo
Marílio Cardoso	Instituto Superior de Engenharia do Porto
Mário Cruz	Escola Superior de Educação
Mário Pinto	Escola Superior de Media Artes e Design
Nuno Bettencourt	Instituto Superior de Engenharia do Porto
Nuno Morgado	Instituto Superior de Engenharia do Porto
Paula Tavares	Instituto Superior de Engenharia do Porto

## Livro de Atas do SEI'22

---

Paulo Maio	Instituto Superior de Engenharia do Porto
Paulo Matos	Instituto Superior de Engenharia do Porto
Paulo Proença	Instituto Superior de Engenharia do Porto
Piedade Carvalho	Instituto Superior de Engenharia do Porto
Ricardo Queirós	Escola Superior de Media Artes e Design
Rosa Reis	Instituto Superior de Engenharia do Porto
Rui Marques	Instituto Superior de Engenharia do Porto
Rui Soares	Escola Superior de Tecnologia e Gestão
Sérgio Moreira	Instituto Superior de Engenharia do Porto
Telmo Matos	Instituto Superior de Engenharia do Porto
Teresa Terroso	Escola Superior de Media Artes e Design

## Suporte Técnico e Secretariado

---

Carla Cunha	Instituto Superior de Engenharia do Porto
-------------	---

---

Lurdes Santos	Instituto Superior de Engenharia do Porto
---------------	---

---

Nuno Fonseca	Instituto Superior de Engenharia do Porto
--------------	---

---

Pedro Rocha	Instituto Superior de Engenharia do Porto
-------------	---

---

## Oradores Convidados

---

Ângelo Martins	Instituto Superior de Engenharia do Porto
Catarina Loureiro	TalkDesk
David Mota	DevScope
Luis Ribeiro	Instituto do Emprego e Formação Profissional, IP
Patrícia Alves Sousa	Celfocus
Ricardo Queirós	Escola Superior de Media Artes e Design

---

## Sessão Plenária

### GAMIFICAÇÃO NO ENSINO DA PROGRAMAÇÃO DE COMPUTADORES



**Ricardo Queirós**  
ESMAD

#### Biografia

Ricardo Queirós é doutorado em Ciência de Computadores e Professor Adjunto na área de Informática na Escola Superior de Media Artes e Design do Instituto Politécnico do Porto onde leciona disciplinas sobre programação Web e móvel. É também investigador na área da gamificação, interoperabilidade em sistemas de e-learning e aprendizagem de linguagens de programação, no Centro de Investigação em Sistemas Computacionais Avançados (CRACS), do INESC TEC Porto. É também Coordenador do Gabinete de Ensino à Distância do Centro de Inovação Pedagógica (CIP) do Politécnico do Porto. É o responsável pela International Computer Programming Education Conference (ICPEC), a primeira conferência internacional dedicada exclusivamente ao ensino da programação de computadores. Atualmente trabalha na conceção e implementação de ambientes de aprendizagem gamificados para fomentar o processo de ensino-aprendizagem de cursos de programação de computadores. É autor de mais de 100 livros e publicações científicas focados nos tópicos de ensino e investigação previamente referidos.

## Mesa Redonda

### OS DESAFIOS PROFISSIONAIS DO INFORMÁTICO NO SÉCULO XXI

Moderador



**Ângelo Martins**  
ISEP

### Biografia

Ângelo Martins é doutorado em Engenharia Eletrotécnica e Computadores pela Universidade do Porto, exerce as funções de docência no Departamento de Engenharia Informática do ISEP desde 1993, atualmente com o cargo de Professor Coordenador. Foi Diretor de Curso da LEI-ISEP entre 2008 e 2018 e é atualmente coordenador da pós-graduação/programa de requalificação SWitCH, uma parceria entre o ISEP e o Porto Tech Hub. É cocoordenador da zona europeia da iniciativa CDIO. É investigador do centro Humanize do INESC TEC. A sua principal área de investigação são os sistemas de informação, nomeadamente os sistemas ubíquos com aplicação à área do Ambient Assisted Living (AAL).

## Intervenientes



**Catarina Loureiro**  
Talkdesk

### Biografia

Catarina Loureiro juntou-se à Talkdesk, para ajudar a construir equipas de R&D na área de Inteligência Artificial como Senior Talent Acquisition. Atualmente trabalha como Senior People Business Partner e é responsável por três grandes áreas de Engenharia e Produto, atuando como uma parceira estratégica e colaborando na estratégia de Pessoas para a Talkdesk. Com background em Psicologia pela Faculdade de Psicologia e Ciências da Educação da Universidade de Coimbra e Mestrado em Psicologia e Desenvolvimento de Recursos Humanos pela Universidade Católica Portuguesa, passou por Lisboa, Coimbra e Porto, cidade na qual começou a sua carreira profissional na área dos Recursos Humanos.



**David Mota**  
DevScope

### Biografia

David Mota é licenciado em Engenharia Informática pela Universidade da Beira Interior, detém ainda um Mestrado na área de Gestão e Sistemas de Informação para as Organizações. Nos últimos anos teve a oportunidade de desempenhar funções em diferentes empresas, tanto na área da consultoria

como na área de suporte. Na Microsoft foi responsável pela introdução de várias tecnologias cloud nas Instituições de Ensino Superior. Atualmente desempenha a função de Academy Manager na Devscope, um projeto que acolhe e acompanha alunos finalistas na realização dos seus diferentes projetos, entre eles, Projetos Finais de curso e Teses de Mestrado, bem como projetos de investigação.



**Luis Ribeiro**  
IEFP

## **Biografia**

Luís Ribeiro tem a Licenciatura em Professores do Ensino Básico/Educação Física, realizou a componente curricular do Mestrado de Ensino de Educação Física nos Ensinos Básico e Secundário e a Pós-Graduação em Ciências da Educação – Especialização em Administração Educacional, ambas na Universidade de Trás-os-Montes e Alto Douro (UTAD). Realizou o Curso de Formação em Gestão Pública (FORGEP) no ISCTE. É Técnico Superior de Segurança no Trabalho e formador com Certificado de Competências Pedagógicas de Especialização em Gestão/Coordenação de Formação. Ingressa na Administração Pública nos Serviços de Ação Social da UTAD, foi Técnico Superior do Centro de Emprego e Formação Profissional de Vila Real do IEFP, I.P. e Técnico Superior do Núcleo de Formação Profissional dos Serviços de Coordenação da Delegação Regional do Norte do IEFP, I.P. sendo atualmente Coordenador do Núcleo de Formação Profissional dos Serviços de Coordenação da Delegação Regional do Norte do IEFP, I.P. É Presidente do Conselho de Administração do CESAE DIGITAL. Tem participado em vários grupos de trabalho sobre Formação Profissional, tem sido membro da comissão organizadora, em congressos e seminários, tem sido orador, em congressos, conferências, seminários e workshops e formador interno do IEFP, I.P.



## Patrícia Alves Sousa

Celfocus

### Biografia

Patrícia Alves Sousa é licenciada em Gestão pela Católica Lisbon School of Business & Economics e em Gestão de Recursos Humanos pela Universidade Lusíada, realizou posteriormente o curso de Executive Coaching na Escola Europeia de Coaching e o Executive Master em Gestão Estratégica de Pessoas no INDEG-ISCTE. Com mais de 15 anos de experiência profissional, trabalhou em várias multinacionais nas áreas Comercial, Marketing, Comunicação e Retalho. Tem sólida experiência na atração de talento de forma estratégica, no alinhamento do negócio com a área de Talent Acquisition e na implementação de uma cultura e mindset Agile nas equipas. Presentemente é Head of Talent Acquisition na Celfocus. Desenvolve a sua atividade enquanto Coach de Carreira desde 2011.

## Artigos



## Aplicação para crianças diagnosticadas com espectro de autismo

Joana Portugal<sup>1</sup>, João Santos<sup>1</sup>, Marco Malta<sup>1</sup>, Jorge Lima<sup>1</sup>[0000-0001-6105-1416], Teresa Terroso<sup>1</sup>[0000-0003-0224-8301]

<sup>1</sup> ESMAD – Escola Superior de Media Artes e Design, P.Porto {40200387,40200305,40200294,jorgelima,teresaterroso}@esmad.ipp.pt

**Resumo.** Este artigo expõe uma aplicação web desenvolvida com o intuito de criar uma ferramenta multiutilizador para crianças com espectro de autismo. Atualmente existem várias plataformas apenas dedicadas para as crianças, sendo que os pais e os professores não conseguem obter informações sobre o progresso. Para além deste facto, pode-se afirmar que as aplicações existentes apenas encontram-se direcionadas para os dispositivos móveis.

O processo de desenvolvimento do projeto foi focado tanto da perspetiva funcional como no design da interface (*UI-User Interface*) e experiência do utilizador (*UX- User Experience*). O produto desenvolvido inclui uma interface web e uma *API REST* que interagem mutuamente. No final, foi realizado um processo de testes e performance em todas as vertentes.

**Palavras-Chave:** Desenvolvimento Web, Autismo, Crianças, Controlo Parental, Emoções.

### 1 Introdução

A Perturbação do Espectro do Autismo [1] (PEA) é uma perturbação do neuro desenvolvimento que se caracteriza por dificuldades na comunicação e interação social, associadas a comportamentos repetitivos e/ou interesses marcados por objetos ou temas específicos. Segundo a OMS, uma em cada 160 crianças tem uma PEA.

O conceito de jogos sérios remete ao ano de 1970. Este termo foi utilizado pela primeira vez no livro “Serious Games”, de Clark Abt [2]. Embora no livro se aborde principalmente jogos de cartas e de tabuleiro, a definição fornecida para jogos sérios permanece atual e continua a ser aceite: “... estes jogos têm um propósito educacional específico e bem pensado e não se destinam a ser usados principalmente para o divertimento”. Nos últimos anos, o conceito de jogos sérios tem ganho uma nova importância, muito impulsionados pela proliferação dos jogos de vídeo e por poderem ser aplicados em diversas áreas, como por exemplo a educação que procura equilibrar o tema com a jogabilidade e a habilidade do jogador reter e aplicar parte do tema no mundo real.

Segundo Elisa Schlunzen, autora de vários artigos sobre a influência das novas tecnologias na educação especial, as tecnologias podem constituir um recurso fundamental

para possibilitar a comunicação das pessoas com necessidades educativas especiais (NEE) [3], permitindo uma manipulação do meio e um melhor desenvolvimento cognitivo.

Neste contexto, este artigo tem como objetivo apresentar o desenvolvimento de uma aplicação web que, através de um conjunto de atividades, permita às crianças com PEA exercitar a manifestação e reconhecimento de emoções. Após a identificação dos requisitos da aplicação, são detalhados os principais aspectos de arquitetura, base de dados de suporte, API (Application Programming Interface) desenvolvida, User Interface (UI) e User Experience (UX). A pesquisa realizada previamente ao desenvolvimento da aplicação permitiu identificar vantagens, desvantagens e aspectos em aberto nas aplicações nesta área de intervenção. Embora não tenha sido possível encontrar uma aplicação web cumprindo os requisitos desejados, existem aplicações móveis de interesse debatidas na Tabela 1.

Aplicação	Prós	Contras
Autimo [4]	Identifica fotos de pessoas; permite a personalização de imagens e textos; regista progresso.	Não possui tecnologia de apoio; não possui idioma português.
Emotismo [5]	Possui idioma português; utiliza ilustrações e fotos em contexto real; recorre a Inteligência Artificial.	Só reconhece emoções primárias (alegria, tristeza, raiva, surpresa, medo).
PicoTea [6]	Permite a comunicação entre tutor e a criança através de mensagens; possui diferentes géneros de atividades para consolidar a habilidade social.	Ausência de elementos de gamificação; interface pouco desenvolvida.
CopyMe [7]	Inclui o uso da câmara para o reconhecimento facial.	Identifica apenas seis tipos de expressões faciais (alegria, tristeza, raiva, surpresa, medo e desgosto).

**Tabela 1.** Prós e Contras de algumas soluções encontradas.

Este artigo está organizado da seguinte forma:

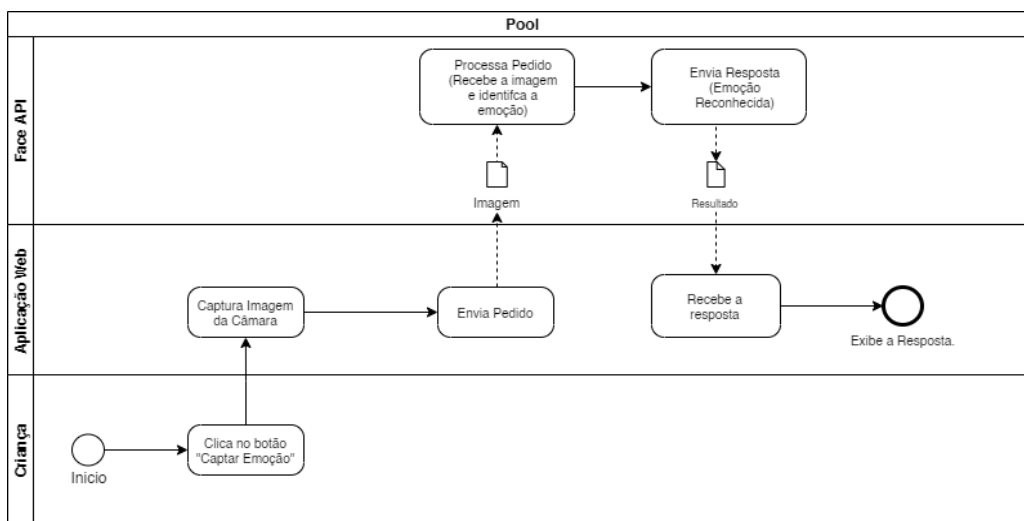
**Secção 2:** Este capítulo é dedicado à descrição do processo pré-código do projeto desenvolvido. Para além disso, menciona-se a necessidade da realização de requisitos, esquemas e diagramas, bem como, a concretização dos protótipos (baixa e alta-fidelidade).

**Secção 3:** A secção de desenvolvimento destaca os diferentes tipos de utilizadores e as funcionalidades implementadas para todos. São destacados os dois tipos de jogos da aplicação e ainda as estatísticas provenientes do histórico.

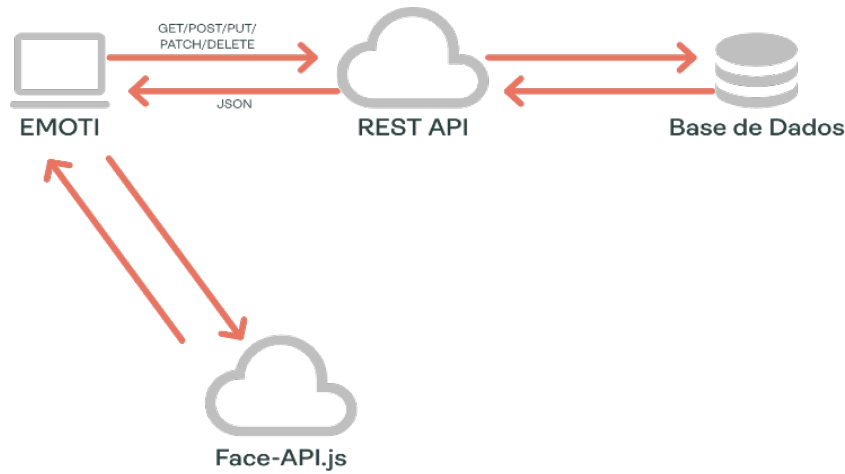
**Secção 4:** Por fim, apresenta-se uma reflexão sobre o projeto realizado, tentando fornecer resposta às questões iniciais, assim como aos objetivos. É dada uma ideia de continuação do projeto expondo perspetivas futuras.

## 2 Planeamento e Estudo

Para o desenvolvimento de uma aplicação é fundamental planear diversos aspetos. De forma a prever o fluxo de Engenharia de Software, foi necessário identificar os requisitos necessários (funcionais e não funcionais), presentes na **Tabela 2**, bem como elaborar esquemas e diagramas, fundamentais para o bom desenvolvimento da aplicação web. Como exemplo, a **Figura 1** apresenta um diagrama BPMN (*Business Process Modeling Notation*) que diz respeito à ligação com a API de reconhecimento facial implementada em um dos jogos. Por sua vez, a **Figura 2** apresenta a arquitetura da aplicação, salientando a arquitetura cliente-servidor entre o *frontend* e o *backend* e a comunicação do *frontend* com uma API externa.



**Figura 1.** Diagrama BPMN referente à conexão com uma API Externa (Face-Api.js)



**Figura 2.** Esquema de Arquitetura da Aplicação

Referência	Tipo de Requisito	Descrição
REQ-001	Funcional	Registrar Utilizadores
REQ-002	Funcional	Fornecer informações sobre o progresso da criança
REQ-003	Funcional	Criar Atividades Personalizadas (professor/tutor)
REQ-004	Funcional	Sugerir Atividades (professor/tutor)
REQ-005	Funcional	Gerir todas as entidades através de um <i>backoffice</i> (Utilizadores /Atividades /Badges)
REQ-006	Funcional	Identificar os alunos com mais dificuldades em reconhecer emoções (professor)
REQ-007	Funcional	Atribuir e armazenar a pontuação no final do jogo à criança
REQ-008	Não Funcional (Usabilidade)	Implementar um sistema de base de dados que permita armazenar os dados necessários ao bom funcionamento do sistema
REQ-009	Não Funcional (Segurança)	Encriptar a palavra-passe

**Tabela 2.** Lista simplificada dos requisitos

Inicialmente foram realizados estudos referentes à interface, à ergonomia e ao design de aplicação e interação antes de iniciar a fase de programação. Esta etapa iniciou-se com a aquisição de inspirações visuais e de conteúdo, tendo a mesma justificação que a pesquisa do estado da arte. Estando concluída esta fase, avançou-se para a construção do protótipo de baixa e alta-fidelidade. Para finalizar esta etapa referente ao design, realizaram-se testes de UI/UX, que eram gravados e continham um guião de tarefas e um formulário pós-sessão (System Usability Scale - SUS), que permitiram realizar uma análise quantitativa/qualitativa em relação aos resultados obtidos.

### 3 Desenvolvimento

De acordo com o planeamento realizado, tal como o protótipo desenvolvido na fase inicial do projeto, a aplicação web consiste em providenciar dois tipos de atividades a todos os utilizadores; essa funcionalidade é mais importante para a criança, visto que afeta o seu progresso no reconhecimento de emoções. Outro tipo de utilizador existente na aplicação é o tutor, que poderá associar crianças para verificar o seu progresso, sugerir e criar atividades, tendo ainda a possibilidade de gerir pedidos de integração das crianças em turmas. Outro utilizador existente na aplicação é o professor, que tem permissões específicas em relação aos outros utilizadores como por exemplo: gerir as suas turmas, adicionar/remover crianças e ainda criar atividades personalizadas. Para que a aplicação web possa funcionar de forma segura e eficaz é necessário que exista um utilizador para gerir todo o website, que controla as atividades, os utilizadores, os badges e as emoções.

Existem dois géneros de atividades: questionários (quizzes) e reconhecimento visual. Nos quizzes são apresentadas quatro opções de resposta por pergunta, onde é pedido que seja identificada a emoção presente numa imagem. Já nas atividades de reconhecimento visual, representada na Figura 3, foi necessário recorrer a uma API externa, face-api.js [8], que conseguisse identificar a emoção expressa pelo utilizador através da funcionalidade da câmara. No final, é apresentado um resumo da atividade realizada, onde indica quais as respostas erradas e as respetivas soluções para o utilizador.



**Figura 3.** Atividade de Reconhecimento Visual

Após a criança, que necessita de estar vinculada a pelo menos um tutor, efetuar uma atividade, é possível visualizar as estatísticas do seu progresso nas atividades, para concluir a taxa de erro em uma determinada emoção ou categoria de imagem. Essa funcionalidade também é visível para um professor, assim que a criança estiver presente em uma turma.

As estatísticas também estão visíveis para o tutor e o professor em relação ao progresso das atividades personalizadas e às questões erradas das mesmas. Os gráficos disponíveis podem ser descarregados por esses utilizadores, com a opção de editá-los

uma vez que são transferidos em formato svg. Na Figura 4, é possível observar um exemplo em relação às estatísticas providenciadas ao tutor.



**Figura 4.** Estatísticas referentes às emoções que a criança já identifica

Após a finalização da prototipagem, foi desenvolvido o frontend da aplicação web. De forma a simplificar o processo de programação recorreu-se a diversas tecnologias sendo elas: Vuejs [9], VueRouter e BootstrapVue [10].

Para o servidor, foi utilizado o Node.JS [11] em conjunto com Express [12] para a construção de uma API com a arquitetura REST. Optou-se por utilizar uma base de dados não relacional para gestão de dados, visto que era necessária uma grande manipulação de dados, e foi utilizado o MongoDB [13]. Assim, foram implementados os pedidos HTTP necessários para o frontend da aplicação, com ajuda da aplicação Postman [14] para testar as respostas que seriam enviadas.

O deploy (alojamento) da API foi realizado na plataforma Heroku [15] e para o website, foi escolhida a plataforma Netlify [16].

## 4 Testes e Validação

De forma a perceber se aplicação possuía uma boa performance e cumpria os requisitos delineados na etapa de planeamento realizaram-se diversos testes (unitários, integração e usabilidade).

Em relação ao protótipo funcional, os testes de UI/UX foram gravados e continham um guião de tarefas e um formulário pós-sessão (System Usability Scale - SUS). Este tipo de teste foi aplicado aos dois géneros de utilizadores, os tutores e os professores. Os resultados obtidos no formulário pós-sessão foram bastante positivos, contudo, ao verificar as gravações foi possível verificar que os inquiridos tinham alguma dificuldade ao simular a navegação no protótipo. Com esta análise, refletiu-se e concluiu-se que era necessário redesenhar o protótipo tendo mais em atenção à ergonomia.

Em relação à aplicação funcional, para medir a performance utilizou-se a ferramenta Google LightHouse [17] e verificou-se que a mesma podia ser melhorada. Nesse

sentido, reduziu-se o tamanho das imagens alojadas, predefiniu-se uma fonte fallback e procedeu-se à minimização dos ficheiros JavaScript e CSS. Com a implementação destas medidas a performance ganhou uma melhoria considerável.

Ainda na aplicação funcional procedeu-se à realização de testes de integração, com API e a base de dados, e unitários recorrendo à ferramenta Jest [18], que permitiu no caso API e da base de dados avaliar se os pedidos HTTP estavam a funcionar corretamente e se realizavam a função pretendida. No caso dos testes unitários a ferramenta Jest permitiu verificar se as funções executavam e devolviam a resposta correta. De forma geral, os testes de integração e unitários, foram bem-sucedidos, tendo em atenção todos os cenários possíveis de execução das rotas.

Por fim, testou-se a usabilidade da aplicação recorrendo à ferramenta Selenium [19], que permitiu simular a interação do utilizador com a aplicação. Os resultados obtidos foram bons, uma vez que todos os processos foram realizados de forma eficiente.

Com a finalização da aplicação funcional, permitiu realizar uma nova avaliação sobre os concorrentes e a sua aceitação no mercado. Nesse sentido é possível concluir que a aplicação web destaca-se em relação às outras propostas ao nível do design, da interação e em relação aos elementos de gamificação disponíveis. No que diz respeito à aceitação, ainda é bastante cedo para conseguir obter uma noção concreta, uma vez, que os meios de divulgação da aplicação foram as redes sociais e também pelo facto da aplicação conter um número de reduzido de utilizadores.

## 5 Conclusão

Com a concretização desta aplicação permitiu obter uma visão mais aprofundada sobre a importância da informática nas doenças psicológicas. Esta plataforma irá permitir que a criança consiga desenvolver as suas habilidades sociais para além do ambiente escolar. Esta ferramenta consegue ainda, auxiliar os tutores e professores quanto ao progresso das crianças. Seguindo o processo de construção de uma aplicação foi possível obter uma plataforma funcional e com uma boa performance. Foi ainda possível testar todas as funcionalidades implementadas na aplicação (como por exemplo: criação de pedido de atribuição a uma turma). Com todo este processo foi possível obter uma ferramenta de ajuda muito distinta das atuais.

Sendo um produto que foi desenvolvido recentemente, nem todos os planos idealizados foram concretizados. Desse modo, futuramente a aplicação poderá incluir as seguintes funcionalidades:

- a criação de novos tipos de jogos, como por exemplo, o jogo tradicional da força.
- a criação de um diário para criança, dando ainda a possibilidade de o tutor e o professor consultá-lo.
- a criação de desafios para as crianças, este elemento de gamificação permitiria ainda atribuir vales de desconto para lojas de brinquedos.
- a possibilidade de a criança em caso de urgência possa realizar na plataforma uma videochamada com o tutor ou um professor.

- alargar o tipo de estatísticas, como por exemplo, incluir tanto a emoção como a categoria para calcular a quantidade de vezes que, por exemplo, a emoção “Feliz” é reconhecida na categoria “Realidade”.

## Referências

1. Perturbação do Espectro do Autismo | Autismo. Disponível em <https://www.cuf.pt/saude-a-z/perturbacao-do-espectro-do-autismo>, consultado a: setembro de 2022.
2. Abt Clark, “Serious Game”, 1970.
3. Schlunzen M., “A tecnologia para inclusão de pessoas com necessidades especiais”, DP&A, Rio de Janeiro, 2005.
4. Autimo | Autómato-Auticiel. Disponível em: <https://auticiel.com/application/autimo/>, consultado a: setembro de 2022.
5. Emotismo | Emotismo: um aplicativo para auxiliar crianças no espectro autista a reconhecer e reproduzir emoções. Disponível em: <https://doi.org/10.5753/cbie.sbie.2020.692>, consultado a: setembro de 2022.
6. PicoTea | “Utilizando la tecnología para la inclusión de personas con TEA”. Disponível em: <https://infopspecs.wixsite.com/app-pictotea>, consultado a: setembro de 2022.
7. CopyMe | CopyMe by Game Studio. Disponível em: <https://appadvice.com/app/copyme/825405687>, consultado a: setembro de 2022.
8. Face-api.js | Documentation. Disponível em: <https://justadudewhohacks.github.io/face-api.js/docs/index.html>, consultado a: setembro de 2022.
9. Vue.js | Progressive JavaScript Framework. Disponível em: <https://vuejs.org/>, consultado a: novembro de 2021
10. Bootstrap Vue | Progressive framework for building interfaces. Disponível em: <https://bootstrap-vue.org/>, consultado a: novembro de 2021.
11. Node.js | JavaScript runtime built on Chrome's. Disponível em: <https://nodejs.org/en/>, consultado a: abril de 2022.
12. Express | Node.js web application framework. Disponível em: <https://expressjs.com/>, consultado a: maio de 2022.
13. MongoDB | The Developer Data Platform. Disponível em: <https://www.mongodb.com/>, consultado a: maio de 2022.
14. Postman | API platform for building and using APIs. Disponível em: <https://www.postman.com/>, consultado a: junho de 2022.
15. Heroku | Run Node apps on the Heroku platform. Disponível em: <https://www.heroku.com/>, consultado a: junho de 2022.
16. Netlify | Develop & deploy the best web experiences in record time. Disponível em: <https://www.netlify.com/>, consultado a junho de 2022.
17. Google Lighthouse | Overview - Chrome Developers. Disponível em: <https://developer.chrome.com/docs/lighthouse/overview/>, consultado a: junho de 2022.
18. Jest | Delightful JavaScript Testing. Disponível em: <https://jestjs.io/>, consultado a: junho 2022.
19. Selenium | Selenium automates browsers. That's it! Disponível em: <https://www.selenium.dev/>, consultado a: junho de 2022.
20. Barbosa Hugo, “Análise do recurso a novas tecnologias no ensino de autistas”. Disponível em: [https://recipp.ipp.pt/bitstream/10400.22/3079/1/DM\\_HugoBarbosa\\_2009\\_MEI.pdf](https://recipp.ipp.pt/bitstream/10400.22/3079/1/DM_HugoBarbosa_2009_MEI.pdf), consultado a: setembro de 2022.

## Aplicação *Web* para Consulta de Dados de Sistemas SIRTEC

Marta Almeida<sup>1</sup>, Luís Fernandes<sup>3</sup>, Vítor Cardoso<sup>1</sup>, Pedro Reis<sup>1</sup>, Sandra Aires<sup>1,2</sup>,  
António Sousa<sup>1,2</sup> e Fernando Carvalho<sup>1</sup>

<sup>1</sup> ISEP – IPP, Porto, fjc@isep.ipp.pt

<sup>2</sup> LEMA, ISEP-IPP, Porto

<sup>3</sup> Tecnocon – Tecnologia e Sistemas de Controlo, S.A., Vale de Cambra

**Resumo.** Os sistemas SIRTEC permitem, para efeitos de controlo de qualidade, a medição dos parâmetros do leite a partir de um processo automático de recolha de amostras em vários postos de fornecimento. Sobre essas amostras, entre outros parâmetros, realiza-se a leitura da temperatura, do volume e das coordenadas GPS (*Global Positioning System*) do local de armazenamento do leite; procedendo-se ao registo numa base de dados SQL *server*.

Este projeto consiste no desenvolvimento de uma aplicação *web*, baseada na interação entre cliente e servidor, para leitura e apresentação dos dados relativos aos sistemas SIRTEC, que se encontram registados na referida base de dados.

O *front-end* é composto por duas páginas: uma de *login* e uma outra que permite a pesquisa, a visualização e a obtenção de relatórios sobre os referidos registos. A informação relativa a esses parâmetros encontra-se organizada por grupos e o acesso do utilizador aos diferentes grupos e sobre cada um deles, depende do tipo de permissão que lhe foi atribuído nas predefinições registadas na base de dados. Assim, o projeto compreende: o estudo da rede de dados armazenados na base de dados da empresa; o planeamento da estrutura da aplicação *web*; o desenvolvimento do *back-end* para gestão de dados e comunicação, realizados nas linguagens PHP e de base de dados SQL *server*, e o desenvolvimento dos serviços de *front-end*, realizados nas linguagens HTML e *JavaScript* (JS), recorrendo ainda à formatação de estilos CSS.

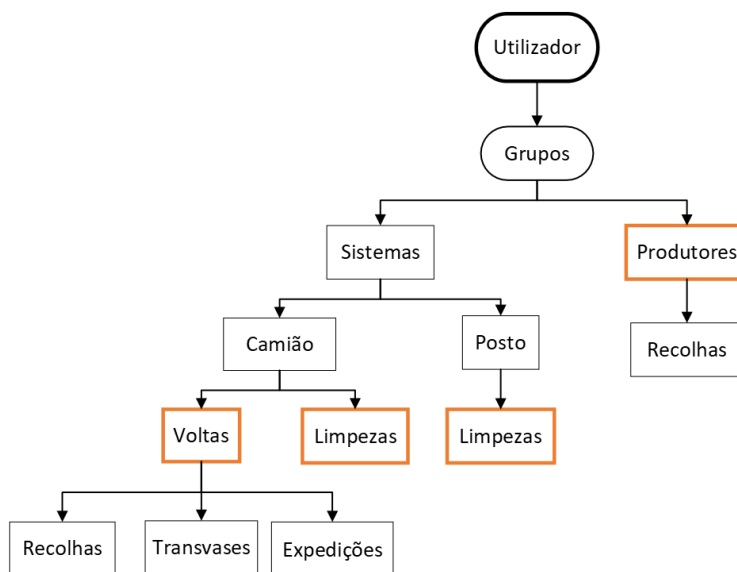
**Palavras-chave:** Backend, Frontend, GPS, JS, PHP, SIRTEC, SQL, Tecnocon

### 1 Introdução

A Tecnocon é uma empresa de automação industrial que inclui, na sua atividade o desenvolvimento de sistemas de medição dos parâmetros do leite a partir da recolha automática de amostras, os sistemas SIRTEC. Os sistemas SIRTEC contemplam, entre outras tarefas: a recolha e armazenamento de amostras de leite usando microbomba simples; a visualização direta da temperatura do compartimento no *display* do controlador; a identificação dos frascos de amostras por código de barras; a identificação dos tanques dos produtores com informação disponível na transferência de dados para a

fábrica; o GPS para localização geodésica global e identificação do produtor; a transferência de dados via GSM (*Global System Mobile*) do sistema para o servidor FTP; etc. Entre outros parâmetros relativos às amostras de leite recolhidas, os sistemas SIRTEC disponibilizam temperaturas, volumes, identificação e localização do produtor, dos tanques e das cisternas. Esta informação encontra-se registada na base de dados SQL *server* da empresa, composta por 21 tabelas, das quais 12 contêm informações que aqui relevam e que se pretendem apresentar numa aplicação *web*.

Este projeto tem como objetivo a implementação de uma aplicação *web* responsiva, que permita consultar os registos do sistema SIRTEC. Para o efeito, em trabalho conjunto com a Tecnocon, especificou-se que a informação fosse organizada de acordo com o diagrama da Fig. 1, que apresenta os vários níveis de pesquisa e informações disponíveis, bem como a relação que se estabelece entre eles. Assim, o utilizador acede a um ou a vários “Grupos” operacionais e de fornecimento que dispõem do mesmo tipo de informação, a partir das opções “Sistemas” ou “Produtores”. A partir da opção “Produtores”, obtêm-se informações referentes às recolhas associadas a produtores. Já no caso de seleccionar a opção “Sistemas”, terá acesso às “Limpezas” ou às “Voltas” e “Limpezas” desse sistema, respetivamente, dependendo se a natureza deste é do tipo “Posto” ou “Camião”. Relativamente a uma “Volta” de um camião, o utilizador poderá ainda consultar informações sobre “Recolhas”, “Transvases” ou “Expedições” do leite.



**Fig. 1.** Diagrama dos níveis de pesquisa da página principal da aplicação *web*.

Este tipo de aplicação *web* que envolve a comunicação entre cliente (*web browser*) e servidor, está devidamente consagrada no estado da arte. No entanto, não é do nosso conhecimento, pelo menos até à presente data, a existência de uma aplicação dedicada, em particular, ao acesso remoto para apresentação dos dados SIRTEC da Tecnocon.

Nas secções seguintes apresentam-se: as ferramentas e tecnologias utilizadas no desenvolvimento da aplicação *web*; a implementação do *back-end* e do *front-end* da aplicação *web*; alguns exemplos de resultados; conclusões e perspetivas de trabalho futuro.

## 2 Ferramentas e tecnologias

A aplicação *web* tem como suporte um servidor de HTTP Apache [1], auxiliado por uma base de dados SQL *server*, que é atualmente usada pela Tecnocon para registrar toda a informação relativa aos sistemas SIRTEC.

O *front-end* da aplicação *web* contempla o desenvolvimento de páginas HTML usando alguns elementos da *framework* Bootstrap [2]. Esses elementos são formatados usando classes CSS, algumas delas da *framework* e outras criadas no âmbito deste projeto. A manipulação do comportamento das páginas HTML faz-se em JS, a partir de eventos e de um conjunto de funções que permitem a criação de componentes dinâmicos, tais como: tabelas e mapas.

O *back-end* da aplicação *web* foi implementado em PHP. Para o efeito, são realizadas inúmeras ligações e *queries* de pedidos de pesquisa à base de dados SQL *server* da parte do servidor de HTTP Apache, usando as potencialidades da biblioteca PHP dedicada.

A comunicação entre o *front-end* e o *back-end* da aplicação *web* realiza-se a partir da Fetch API [3], que permite que os pedidos HTTP sejam realizados diretamente ao servidor com base em objetos de *Request* e *Response*, usando o método *fetch()*.

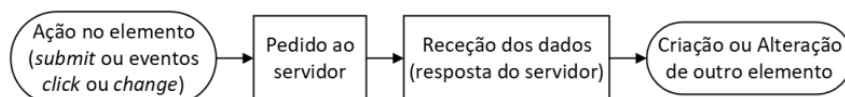
Para cada *query*, a informação residente na base de dados SQL *server* é codificada em formato JSON, dada a sua notável compatibilidade com o JS [4].

## 3 Implementação

### 3.1 *Front-end* da aplicação *web* desenvolvida

O *front-end* da aplicação é composto por duas páginas HTML: uma inicial de apresentação, a partir da qual é possível realizar o *login* e uma outra, a principal, a partir da qual, após a validação das credenciais de acesso, o utilizador pode realizar pesquisas e visualizar os correspondentes resultados aos pedidos que venha a requerer. Assim que o utilizador efetua *login* com sucesso, o servidor HTTP Apache acede à base de dados. Seguidamente, lê o nome dos grupos a que o utilizador tem acesso e envia-os ao cliente para preenchimento dinâmico dos menus de pesquisa. O utilizador é então redirecionado, a partir da função *header()*, para a página principal da aplicação, podendo consultar os dados a que tem permissão – todos os grupos dispõem do mesmo tipo de informação, conforme o diagrama da Fig.1.

As interações que ocorrem ao nível do *front-end*, baseiam-se no diagrama da Fig. 2.



**Fig. 2.** Diagrama geral do funcionamento das páginas da aplicação *web*.

Os pedidos ao servidor realizam-se a partir da submissão de um formulário ou através da ocorrência de eventos DOM do JS, tais como o *onchange* ou o *onclick*. Já a

comunicação entre o servidor e o cliente realiza-se a partir da Fetch API, considerando o protótipo apresentado em Cod. 1 (linha 1), em que o parâmetro: “*funcao*” corresponde ao identificador do elemento selecionado pelo utilizador, no caso, um grupo específico a pesquisar; “*file*” identifica o recurso do servidor que executa o pedido.

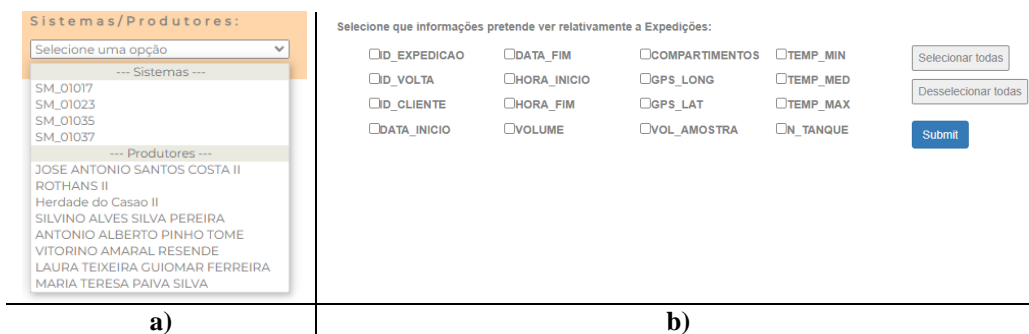
```

1 get_data('grupos','principal.php?q1='+ value);
2
3 async function get_data(funcao, file) {
4   let x = await fetch(file,{method:'POST'});
5   let y = await x.text();
6   const obj = JSON.parse(y);
7   [...]
8 }

```

**Cod. 1.** Pedido do cliente e resposta do servidor a partir do uso da *Fetch API*.

Os elementos que se apresentam na página principal da aplicação podem ser de 4 tipos: menu *drop-down*, menu *checkboxes*, mapa ou tabela. Na Fig. 3 a) apresenta-se um exemplo relativo aos elementos menu *drop-down* e na Fig. 3 b) o menu *checkboxes*.



**Fig. 3.** Exemplos de menu *drop-down* em a) e menu de *checkboxes* em b).

Foi implementada uma função em JS dedicada à criação de tabelas dinâmicas, Fig. 6 e Fig. 7, de acordo com o tipo de pesquisa do utilizador e com base na informação disponível na base de dados. No momento em que os elementos da tabela são criados, segue-se o preenchimento do seu cabeçalho e corpo.

Finalmente, o elemento mapa foi inserido na página com o intuito de representar as coordenadas geodésicas, registadas na base de dados, referentes a locais onde um camião realizou alguma atividade numa volta específica. Para o efeito, foram testadas duas soluções. Na primeira, a *Google Maps* oferece a possibilidade de se “Incorporar um Mapa” a partir do qual qualquer utilizador pode pesquisar um ponto de destino ou uma rota com vários marcadores de localização e obter uma *tag HTML <iframe>*, que é gerada automaticamente [5]. Esta opção tem a vantagem de ser gratuita, porém não permite definir a ordem de passagem nas localizações registadas, o início da volta e localizações exatas e, usar um número ilimitado de marcadores de localização. Na segunda, a *Google Maps Javascript API*, permite personalizar mapas com conteúdo próprio, usando vários serviços e bibliotecas [6]. Neste sentido, para ultrapassar essas limitações, a Tecnocon adquiriu uma chave da API, que permitiu formatar os marcadores

de localização por ordem cronológica da volta do camião e apresentam informações associadas à atividade ocorrida nesse local, a partir dos registos da base de dados.

A aplicação permite ainda gerar “Relatórios de Limpeza”, exportáveis para ficheiros CSV ou PDF, Fig. 8, implementados pela função *objectExporter()* [7]. Na Fig. 4, apresenta-se o diagrama de blocos do esquema lógico da pesquisa por “Limpezas”, que detalha o processamento associado ao respetivo bloco inserido no diagrama da Fig. 1.

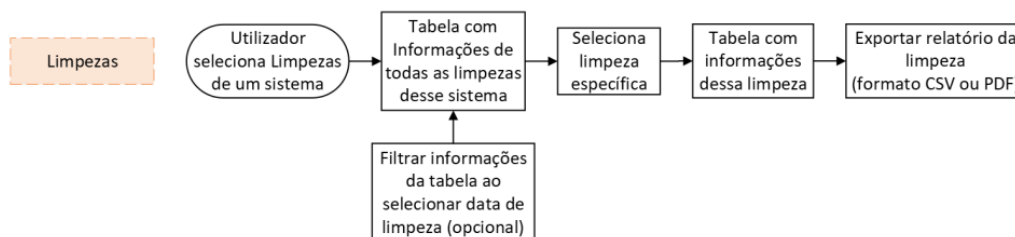


Fig. 4. Diagrama de blocos do esquema lógico da pesquisa por “Limpezas” - exemplo.

### 3.2 Back-end da aplicação web desenvolvida

A Fig. 5 apresenta o diagrama de blocos representativo do modo de funcionamento da aplicação vista do lado do servidor. Implementado em linguagem PHP, inicia-se com um pedido do cliente, faz a ligação à base de dados, interpreta, faz o pedido à base de dados, processa a resposta e submete os dados codificados em formato JSON.

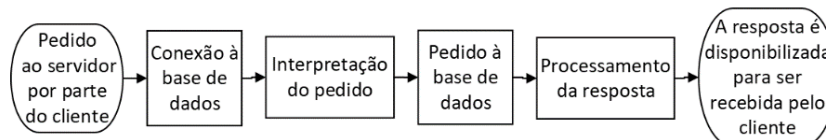


Fig. 5. Diagrama de blocos representativo do *back-end* da aplicação.

A ligação à base de dados concretiza-se a partir da função *sqlsrv\_connect()*, Cod. 2, de acordo com o método *SQL Server Authentication*, recebendo como parâmetros: o nome do servidor, o nome da base de dados e as respetivas credenciais de *login*. A informação pretendida é requerida a partir de *queries*, que a retorna em vetores.

```

1 <?php
2 $serverName=$dbServer;
3 $connInfo=array("Database"=>$dbName, "UID"=>$dbUser,
4 "PWD"=>$dbPwd);
5 $conn = sqlsrv_connect($serverName, $connInfo);
6 $result="SELECT * FROM Utilizadores WHERE USERNAME='$Name'";
7 $query = sqlsrv_query($conn, $result);
8 $info = sqlsrv_fetch_array($query);
9
10 sqlsrv_close($conn);
11 ?>

```

Cod. 2. Código PHP desenvolvido para estabelecer a ligação e requerer informações à base de dados *SQL Server* a partir de uma *query*.

As *queries* realizadas pelo servidor à base de dados dependem do pedido do cliente. Esse pedido baseia-se num identificador associado aos campos selecionados e testados do *array* associativo do PHP com base no seguinte exemplo: *if(\$\_REQUEST["q1"])*. A cada identificador corresponde uma ação em concreto, como por exemplo, selecionar as limpezas de um sistema ou selecionar os nomes das colunas da tabela de recolhas.

O processamento da resposta consiste na execução de um conjunto de funções para que os dados associados a três tipos de pesquisa (“Produtores”, “Limpezas” e “Voltas”) sejam lidos da base de dados e codificados em formato JSON através da função *json\_encode()*, tornando simples a sua leitura do cliente a partir do JS. A título de exemplo, em Cod. 3, apresenta-se o código de uma das funções implementadas, que permite retornar um vetor com todos os nomes que identificam as colunas de uma tabela, cujo nome é passado como um dos argumentos da função. O retorno da função *nomes\_colunas\_tabelas()* é utilizado para preenchimento dos menus de *checkboxes* do *front-end* da aplicação. Neste âmbito, foram ainda criadas funções para: ler o conteúdo das tabelas, cujo protótipo é *function procura\_tabela(\$conn, \$table, \$columns, \$variable, \$value)*; verificar a existência de variáveis temporais (hora e data) e formatá-las de acordo com o estilo de apresentação desejado, cujo protótipo é *function teste\_time(\$scabecalho, \$conteudo)*; procurar coordenadas GPS válidas nas tabelas, se o utilizador selecionou a opção “Mapa”, cujo o protótipo é *function teste\_coluna\_mapa(\$scabecalho, \$conteudo)*.

```
1 function nomes_colunas_tabelas($conn, $tName){
2     $result_columns = "SELECT COLUMN_NAME AS CN FROM
3     INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = '$tName'";
4     $query_columns = sqlsrv_query($conn, $result_columns);
5     while($columns=sqlsrv_fetch_array($query_columns,
6     SQLSRV_FETCH_ASSOC)){
7         $out_columns[] = $columns['CN'];
8     }
9     return $out_columns;
10 }
```

**Cod. 3.** Código PHP desenvolvido para obtenção de nomes localizados nas colunas da base de dados *SQL Server*.

Assim que o utilizador efetua o processo de *logout*, eliminam-se todos os dados das sessões de PHP que mantinham a validação do utilizador e procede-se ao seu redirecionamento para a página de *login*, usando o código de Cod. 4.

```
1 <?php
2 session_start();
3 session_unset();
4 session_destroy();
5 header("Location: inicial.html");
6 ?>
```

**Cod. 4.** Código PHP desenvolvido para realizar *logout* do utilizador e reencaminhá-lo para a página de *login*.

## 4 Resultados

Na Fig. 6, apresenta-se um exemplo de resultados da pesquisa por produtores, disponibilizando: um menu *drop-down* de seleção de grupo e outro de seleção de sistema ou produtor; uma tabela com informações gerais sobre produtor selecionado; um menu de *checkboxes* onde o utilizador seleciona as informações que pretende pesquisar sobre as recolhas realizadas nesse produtor e; uma segunda tabela com essas informações.

Grupos: [GRUPO QUATRO] Sistemas/Produtores: [Tecnocon Testes]

Produtor selecionado:

NOME	N_PRODUTOR	RUA	REGIAO	INFO2	GPS_LONG	GPS_LAT	MAPA
Tecnocon Testes	500	Rua Maria Silva Brandao,90	3850.013 AVANCA	PT 18963695	-8.414588	40.859017	Google Maps

Seleção que informações pretende ver relativamente às recolhas realizadas:

ID\_RECOLHA     VOLUME     COD\_FRASCO\_3     VOL\_REC\_TO\_3  
 ID\_OPERACAO     VOL\_ESTIMADO\_USADO     TIPO\_AMOSTRA     VOL\_REC\_TO\_4  
 DATA\_INICIO     VOL\_PAINELA\_ADD     COMPARTIMENTOS     VOL\_REC\_TO\_8  
 DATA\_FIM     TEMP\_AMOSTRAIS     GPS\_LONG     VOL\_REC\_TO\_6  
 HORA\_INICIO     TEMP\_MIN     GPS\_LAT     VOL\_REC\_TO\_7  
 HORA\_FIM     TEMP\_MED     TANQUE     VOL\_REC\_TO\_8  
 N\_PRODUTOR     TEMP\_MAX     VOL\_REC\_TO\_1     FILAS\_CALIBRACAO\_MEDICAO  
 SEQUENCIA     COD\_FRASCO\_1     VOL\_REC\_TO\_2     UNIQUE

Recolhas efetuadas:

ID_RECOLHA	ID_OPERACAO	DATA_INICIO	DATA_FIM	HORA_INICIO	HORA_FIM	TEMP_MIN	TEMP_MED	TEMP_MAX	GPS_LONG	GPS_LAT	MAPA
39	3	10/12/2020	10/12/2020	07:57:21	07:58:42	29.3	30.4	30.7	-30.2723	-17.4567	Google Maps
42	3	10/12/2020	10/12/2020	08:03:35	08:04:56	26.9	28.8	29.9	Sem informação	Sem informação	...
45	3	10/12/2020	10/12/2020	09:12:49	09:14:31	28.7	29.3	29.6	Sem informação	Sem informação	...

Fig. 6. Página com exemplo de resultados da pesquisa por produtores.

Grupos: [GRUPO QUATRO] Sistemas/Produtores: [SA 01040] Camias: [votas]

Volts efetuadas:

ID_VOLTA	SISTEMA	DATA_INICIO	DATA_FIM	HORA_INICIO	HORA_FIM	CIRCUITO
21	SM_01040	27/11/2021	27/11/2021	16:13:39	16:25:00	308
22	SM_01040	30/11/2021	30/11/2021	15:38:28	15:55:30	308
23	SM_01040	30/11/2021	30/11/2021	10:27:37	11:03:16	308
24	SM_01040	17/01/2022	17/01/2022	08:31:03	11:12:03	1488

Seleção um dia para filtrar as limpezas efetuadas (opcional):

Histórico de localizações da volta 24:

Fig. 7. Página de resultados da pesquisa por volta.

Na Fig. 7, apresenta-se: o menu *drop-down* de seleção de grupo, um segundo menu *drop-down* de seleção de sistema ou produtor e um terceiro menu *drop-down* de seleção de voltas ou limpezas; uma tabela com informações gerais de todas as voltas realizadas pelo sistema selecionado; um calendário onde o utilizador pode filtrar os dados dessa tabela selecionando um dia específico; o mapa do histórico de localizações da volta.

Finalmente, na Fig. 8, apresenta-se o exemplo de um extrato do relatório de limpeza resultante da exportação dos dados para ficheiro PDF.

Passo	Data e Hora de Início	Data e Hora de Fim	Duração (s)	Temperatura Mínima (°C)	Temperatura Média (°C)	Temperatura Máxima (°C)	Volume Circulado (L)
Enxaguamento Inicial	17/11/2021 12:24:26	17/11/2021 12:27:55	209	21.1	25.6	35.0	200.9
Limpeza com produto (soda e/ou ácido)	17/11/2021 12:27:55	17/11/2021 12:43:41	946	29.6	57.0	70.6	1442.8
Enxaguamento Final	17/11/2021 12:43:41	17/11/2021 13:13:55	1814	24.4	28.2	53.2	2297.7

Fig. 8. Exemplo de um Extrato do Relatório de Limpeza.

## 5 Conclusões e perspetivas de trabalho futuro

A aplicação *web* desenvolvida para a empresa Tecnocon, com o objetivo de monitorizar os parâmetros do leite a partir do sistema SIRTEC foi concluída com sucesso. Foi submetida a diversos testes experimentais no sentido de verificar a sua funcionalidade e não foi detetada, até à presente data, qualquer anomalia que exija correções. Os responsáveis da empresa, que testaram a aplicação, deram um parecer positivo às suas funcionalidades e consideram que traz mais valias às suas rotinas diárias operacionais.

Como perspetivas de trabalho futuro, considera-se útil adicionar as seguintes funcionalidades: a possibilidade de representar graficamente o histórico de temperaturas médias registadas por volta e gerar alertas baseados em limiares de temperatura predefinidos; disponibilizar uma página de perfil do utilizador para edição dos seus dados.

## Referências

1. WampServer, <https://www.wampserver.com/en/>, último acesso a 01/03/2022.
2. Bootstrap, <https://getbootstrap.com>, último acesso a 01/06/2022.
3. Fetch API, [https://www.w3schools.com/jsref/api\\_fetch.asp](https://www.w3schools.com/jsref/api_fetch.asp), último acesso a 01/06/2022.
4. JSON – Introduction, [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp), último acesso a 01/06/2022.
5. Iframes, <https://medium.com/@supun1001/how-to-generate-google-embed-links-programmatically-for-iframe-for-routes-only-d6dc225e59e8>, último acesso a 01/06/2022.
6. Maps JavaScript API, <https://developers.google.com/maps/documentation/javascript>, último acesso a 01/06/2022.
7. :dizzy: Javascript Object to CSV, XLS, pdf, doc and DOM to html generator :dizzy:., <https://gharibi.github.io/JsObjExporter/>, último acesso a 01/06/2022.

## Aplicação *Web* para Monitorização e Controlo de Autómatos

Bruno Maia<sup>1</sup>, Teresa Canedo<sup>3</sup>, Vítor Cardoso<sup>1</sup>, Pedro Reis<sup>1</sup>, Sandra Aires<sup>1,2</sup>, António Sousa<sup>1,2</sup> e Fernando Carvalho<sup>1</sup>

<sup>1</sup> ISEP – IPP, Porto, fjc@isep.ipp.pt

<sup>2</sup> LEMA, ISEP-IPP, Porto

<sup>3</sup> ARUM – Engenharia de Sistemas de Automação, Porto

**Resumo.** Recentemente, o modelo de indústria 4.0 tem assumido um papel de extrema importância no desenvolvimento e competitividade das empresas nacionais, em particular, nas que utilizam sistemas de automação nos seus processos de fabrico. Este novo modelo possibilita, de forma melhorada, a monitorização e o controlo automatizado desses processos com base na interação entre a máquina e o sistema digital. Neste trabalho, essa interação realiza-se entre o autómato Siemens S7-1500 para o controlo de máquinas de fabrico e uma aplicação *web*.

A aplicação *web* apresentada é dedicada à monitorização e ao controlo de autómatos e foi desenvolvida na ARUM. Disponibiliza ao utilizador: um painel de controlo com um conjunto de seletores, que permite selecionar parâmetros do autómato e monitorizá-los a partir de uma *dashboard*; botões de envio de comandos de controlo para o autómato. É disponibilizada a partir de duas implementações distintas: um *web server* que se encontra incorporado no módulo *web* do autómato - desenvolvido, recentemente, pela Siemens e disponível nos autómatos de última geração; um *web server* HTTP Apache externo ao autómato, auxiliado por uma base de dados SQL *server*. Esta segunda implementação, alternativa, justifica-se pela necessidade de se gerar o registo alargado do histórico relativo ao controlo de velocidade de uma máquina a partir do autómato ao longo do tempo, sem prejuízo de utilizar totalmente a memória de dados do autómato.

**Palavras-chave:** ARUM, Back-end, Dashboard, Front-end, JavaScript, TIA Portal.

### 1 Introdução

No âmbito dos projetos de estágio realizados pelo curso da LEEC do ISEP, verifica-se que muitas empresas portuguesas têm a necessidade de monitorizar e controlar remotamente os seus processos de fabrico, a partir de um computador, de um *tablet* ou de um *smartphone*. A maioria desses processos são controlados por autómatos. Atualmente, entre outros, o autómato S7-1500 da Siemens, permite o acesso remoto aos seus parâmetros. Normalmente, esse acesso realiza-se: localmente, a partir de consolas de HMI (*Interface Human Machine*) ou de um computador; remotamente, a partir de pla-

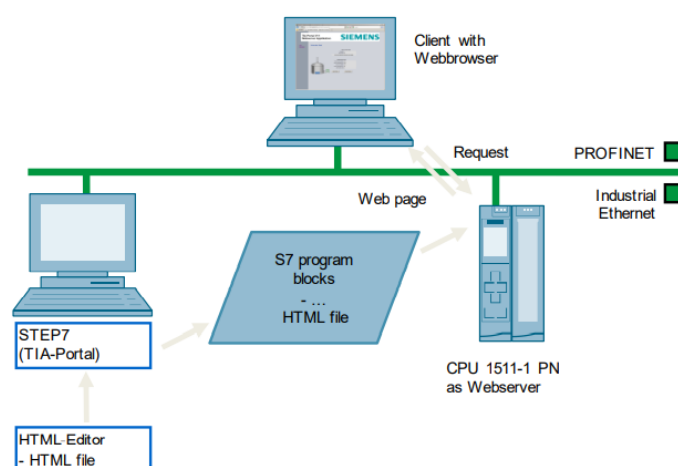
taformas de desenvolvimento, como o TIA Portal (*Totally Integrated Automation Portal*) da Siemens [1] ou, mais recentemente, a partir de um *web server* incorporado num módulo *web* do autômato [2]. Apesar da tecnologia tornar cada vez mais eficiente a interação entre a máquina e o sistema digital, o estado da arte não apresenta um número muito relevante de contribuições científicas na área. Para além de alguns exemplos de pequenos projetos apresentados pela Siemens [3], tanto quanto conhecemos, existe um trabalho baseado na implementação de laboratórios remotos com o objetivo de aumentar o número de experiências realizadas por alunos [4].

Este trabalho surgiu no âmbito do projeto de estágio e da pretensão da empresa ARUM - dedicada ao desenvolvimento de sistemas de automação, agilizar o trabalho dos seus colaboradores e das empresas clientes de sistemas de automação. A ARUM é parceira certificada da *Siemens Industry Automation and Drive Technologies* e tem como principais atividades: desenvolvimento de *software*, comissionamento, assistência técnica, diagramas elétricos em EPLAN e consultoria.

Assim, o projeto tem como objetivo a implementação de uma aplicação *web* que permite o acesso a 18 parâmetros do autômato S7 -1500, para efeitos de monitorização e controlo. O *front-end* da aplicação baseia-se, em parte, na implementação de uma página *web*, composta por uma *dashboard* para monitorização de 15 parâmetros, selecionáveis a partir de um painel de seletores e 3 botões para envio de instruções de comando ao autômato.

A aplicação é disponibilizada ao utilizador a partir de duas implementações distintas, na medida em que utiliza diferentes *web servers* e diferentes modos de comunicação.

Na primeira implementação é utilizado um *web server* criado pela Siemens que é suportado pelo módulo *web* do autômato S7-1500. A aplicação *web* é acedida pelo utilizador a partir de um *web browser* por ligação TCP/IP, Fig.1. Assim, o *web server* disponibiliza um conjunto *standard* de páginas *web*, tais como: início, identificação, registos de diagnóstico, módulo de estados, mensagens, etc [5]. Não obstante, permite que o utilizador crie um *front-end* de acordo com a necessidade das aplicações que deseje desenvolver, que fica disponível a partir da opção “*Customer Pages*”.



**Fig. 1.** Arquitetura geral do sistema desenvolvida pela Siemens e que suporta a primeira implementação realizada, retirado de [5].

De referir ainda que, a partir da plataforma TIA Portal é possível aceder-se aos parâmetros do autómato em tempo real pela rede de comunicação *Profinet/Industrial Ethernet*, Fig.1. A plataforma permite: gerir todo o projeto em termos de programação do autómato e seu diagnóstico; entre outras configurações, atribuir-lhe um número IP e as respetivas credenciais de acesso; definir uma *firewall*; invocar o comando WWW da ferramenta SIMATIC S7 no sentido de carregar a aplicação *web* para o autómato.

Na segunda implementação é utilizado o *web server* de HTTP Apache externo ao autómato, auxiliado por uma base de dados SQL *server* [6]. O autómato comunica com a base de dados a partir do protocolo de comunicação TDS (*Tabular Data Stream*) [6]. Seguidamente, a parametrização é disponibilizada na página da aplicação usando as potencialidades das funções do PHP, dedicadas à comunicação com bases de dados SQL *server*.

Pretende-se dotar a aplicação *web* com funcionalidades de representação gráfica do controlo de velocidade realizado pela máquina ao longo do tempo. Esta pretensão justifica a realização desta segunda implementação, na medida em que é possível registar todo o histórico relativo aos parâmetros do autómato, sem a necessidade de ocupar, de forma totalitária, a sua memória de dados e hipotéticas expansões de memória.

Na primeira implementação, é possível aceder ao autómato a partir de uma página de *login* disponibilizada pelo seu *web server*. A partir do TIA Portal, o administrador pode definir níveis de permissão de cada utilizador. A segunda implementação também disponibiliza uma página de *login* e uma outra de criação do perfil do utilizador, alojando esses dados na base de dados SQL *server*. Essas páginas exibem um *layout* baseado num *form standard*, que não é formalmente apresentado na secção de resultados.

Este trabalho apresenta nas secções seguintes: os métodos e as ferramentas usadas nas duas implementações realizadas; dada a dimensão do trabalho desenvolvido, apresentam-se algumas descrições sucintas sobre os principais aspetos relativos às implementações dos *back-end* e do *front-end*, apoiadas com diagramas de blocos ou fluxogramas; alguns resultados obtidos; as conclusões e perspetivas de trabalho futuro.

## 2 Equipamentos, ferramentas e tecnologias

Em termos de equipamento foi utilizado o autómato Siemens S7-1500, configurável a partir da plataforma TIA Portal.

O *front-end* da aplicação *web* foi desenvolvido com o suporte na *framework Bootstrap*, fazendo uso das linguagens HTML, *Javascript*, AJAX, AnyChart e CSS da formatação de estilos da própria *framework*.

Os *back-end* da aplicação foram, em parte, implementados a partir da plataforma TIA Portal, usando programação SCL (*Structured Control Language*) [7]. Na primeira implementação, o restante *back-end* que permite a comunicação entre o *web server* e o *front-end* baseia-se no uso da biblioteca *S7\_framework.js* [4], implementada em *Javascript* pela Siemens, com dependência da biblioteca *jquery-2.1.3.min.js*. Já o restante *back-end* da segunda implementação, usa SCL para interação do autómato com a base de dados SQL *Server* a partir do protocolo de comunicação TDS e as bibliotecas do PHP para interação da mesma base de dados com o *front-end*, a partir do Apache.

### 3 Implementação

#### 3.1 Arquitetura da primeira implementação

Na arquitetura geral da primeira implementação, o autómato interage ciclicamente com o *web server* a partir de uma função *main* designada de OB1, que invoca a função de sistema WWW sempre que um dos parâmetros é alterado. Os parâmetros do autómato a monitorizar a partir da página HTML, são obtidos com base na *S7\_framework.js* de  $B \rightarrow C$ , e os parâmetros de controlo são codificados e enviados em formato JSON de  $B \leftarrow C$ . A *framework* acede aos parâmetros do autómato, unificados numa *string*, com base na função de leitura *S7Framework.readData* e na função de escrita *S7Framework.writeData*.

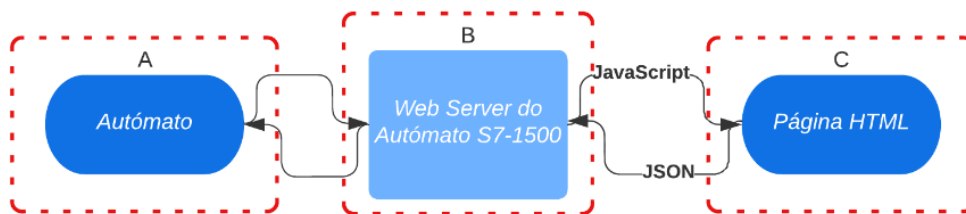


Fig. 2. Diagrama de blocos (blocos A, B e C) relativo à primeira implementação desenvolvida.

O *front-end*, representado pelo bloco C do diagrama de blocos da Fig.2, cujo o *layout* da página HTML se apresenta na Fig. 6, foi obtido a partir da *framework Bootstrap*. Foram ainda adicionadas regras de responsividade para que os elementos se adaptem a diferentes dimensões do *web browser*. Embora a *framework* disponibilize um ficheiro CSS com elementos de formação de estilos, foram criados outros elementos de formação. Um deles, para formatar cores de LED localizados na *dashboard*, que sinalizam o “Estado do Grupo” relativo a máquinas controladas pelo autómato, Fig. 6: cor verde, certas funções estão ativas; cor vermelha, certas funções estão desativadas ou desligadas; cor cinzenta, não há comunicação com determinada função. A *dashboard* disponibiliza ainda botões de controlo do modo operacional da máquina controlada pelo autómato, associadas às *labels* “Manual”, “Semi-Automático” e “Automático”. Só um dos modos operacionais pode estar ativo e a cor do botão diferente da cinzenta, indica o modo operacional atual da máquina, podendo ser alterado pelo utilizador com um *click* no botão desejado, Fig. 6.

#### 3.2 Arquitetura da segunda implementação

Na arquitetura geral da segunda implementação, apresentada na Fig. 3, o autómato comunica com a base de dados externa SQL *server* para leitura e escrita da parametrização do autómato alojada na base de dados do TIA Portal (os mesmos registos da memória de dados do autómato), usando o protocolo de comunicação TDS, de  $A \leftrightarrow B$ .

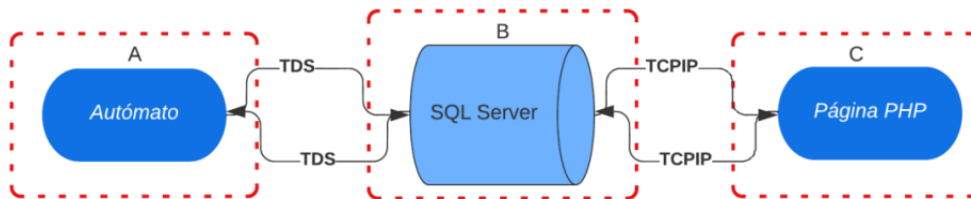


Fig. 3. Diagrama de blocos (blocos A, B e C) da segunda implementação desenvolvida.

Por seu lado, o *web server* HTTP Apache, usando funções do PHP dedicadas, invocadas a partir da página principal da aplicação *web*, comunica com a base de dados SQL *server* para leitura ou escrita da parametrização do autômato, dependendo, respetivamente, se se tratam de parâmetros de monitorização ou comando, Fig. 6, de B ↔ C .



Fig. 4. Gráfico da velocidade de controlo realizado ao longo do tempo pelo autômato, disponibilizado pela *dashboard* e implementado usando a biblioteca AnyChart, escrita em Javascript.

O *front-end* baseia-se numa página PHP com o mesmo *layout* da página HTML da implementação anterior. Esta implementação, utiliza o registo dos parâmetros disponíveis na base de dados SQL *server* para, adicionalmente, representar graficamente o controlo de velocidade ao longo do tempo, realizado pelo autômato sobre uma máquina, num período de tempo predefinido, Fig. 4. Disponibiliza ainda duas páginas PHP: uma dedicada à criação de contas e ao acesso à aplicação a partir do *login*; a outra dedicada à edição do perfil do utilizador, que permite consultar o tipo de permissão e alterar a *password* de acesso. Para o efeito criaram-se três diferentes grupos de permissões: o “Operador”, habilitado apenas a monitorizar os parâmetros do autômato; o “Trabalhador”, habilitado a monitorizar e controlar os parâmetros do autômato; o “Administrador” que, para além de monitorização e controlo, pode criar contas e alterar permissões.

Na implementação do *back-end* relativo ao bloco A do diagrama da Fig. 3, foi necessário enviar os parâmetros do autômato para a base de dados SQL *server*, particionados em duas *strings* (uma contendo os parâmetros de monitorização e outra os de controlo). Isto porque, o protocolo de comunicação TDS, em modo normal, limita o envio a um máximo de 254 caracteres por cada ligação realizada à base de dados [6]. Assim, a interação entre ambos faz-se de forma cíclica e é controlada por 3 temporizadores do autômato: dois para pedidos de envio e um outro para receção dos parâmetros.

A linguagem SCL do TIA Portal dispõe de comandos que permitem a comunicação entre o autômato e a base de dados SQL *server* a partir do protocolo TDS. Por exemplo,

a string “#tempstring” contém um conjunto de parâmetros concatenados e separados por um ponto e vírgula, a enviar de  $A \rightarrow B$ , Fig. 3, submete-se usando os comandos:

```
"SqlCommands".sqlCommands[0] := #tempstring;
"execute" := TRUE;
```

Já a receção de parâmetros de comando de  $A \leftarrow B$ , Fig. 3, realiza-se da seguinte forma:

```
"SqlCommands".sqlCommands[0] := 'Select top(1) manl,semi,autm from dados order by id desc';
"execute" := TRUE;
```

O diagrama de blocos relativo ao bloco C do diagrama da Fig. 3, encontra-se representado com detalhe na Fig. 5. Conforme se pode observar, o *back-end*, interpretado pelo *web server Apache*, foi estruturado a partir de um conjunto de *scripts PHP*, que permite ser facilmente modificado em caso de necessidade: *connect.php*, *receivedata.php*, *senddata.php* e *data.php*. Como exemplo, o código do *script connect.php* é:

```
<?php
$servername= "TIAVI7VM\SQLEXPRESS";
$connectionInfo = array( "Database"=>"MesaGaloProject", "UID"=>"arum",
"PWD"=>"muraaa");
$conn = sqlsrv_connect( $servername, $connectionInfo);
if( $conn === false ) { die( print_r( sqlsrv_errors(), true)); }
?>
```

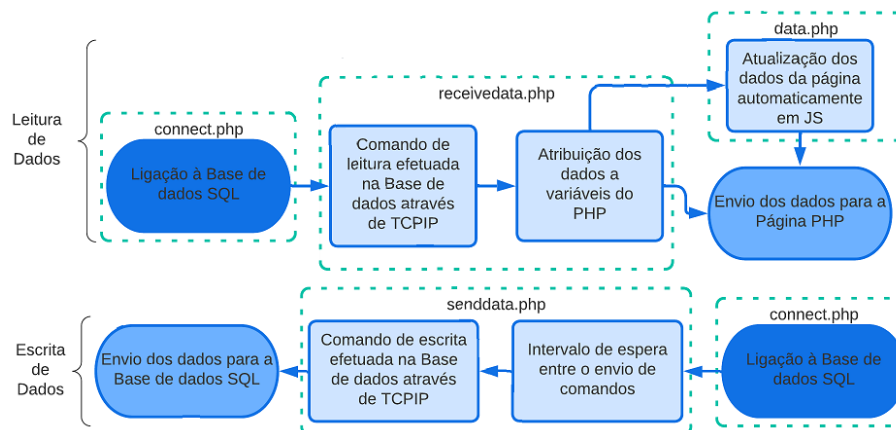


Fig. 5. Diagrama de blocos C, relativo ao diagrama de blocos da segunda implementação.

A página principal da aplicação *web* encontra-se em constante atualização a partir da função *setInterval()* do *javascript*, em períodos de 200 *ms*.

## 4 Resultados

Na Fig. 6, apresenta-se parte do *front-end* relativo à página principal da aplicação *web* desenvolvida, neste caso, correspondente à segunda implementação. Note-se que relativamente à primeira implementação, esta permite a representação gráfica da velocidade de controlo da máquina ao longo do tempo. Disponibiliza um painel de controlo baseado em *checkboxes* e 3 botões, e uma *dashboard* para apresentação de 15 parâmetros do autómato. Os 3 botões controlam os 3 parâmetros relativos ao modo de funcionamento da máquina (que pode ser um grupo de máquinas), a partir das opções “Manual”, “Semi Automático” e “Automático”.

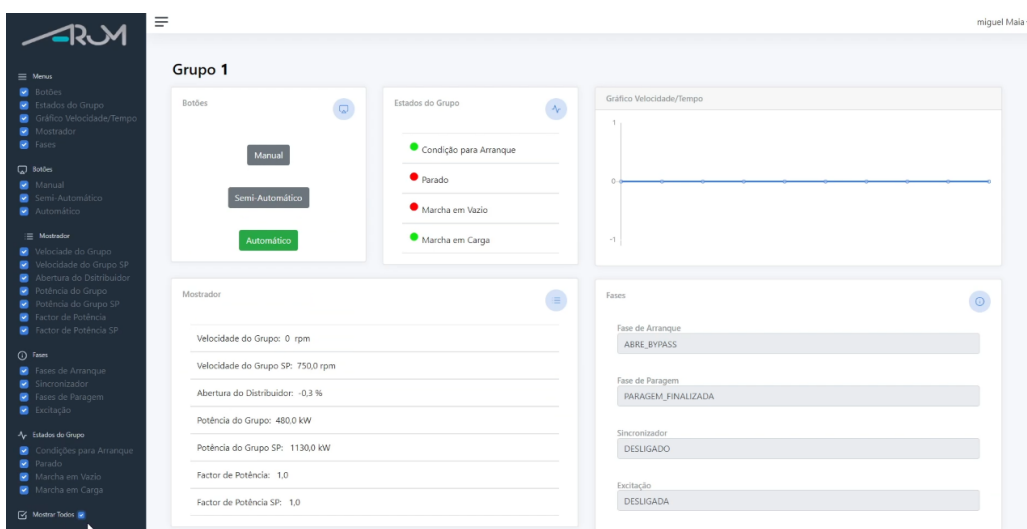


Fig. 6. Página principal da segunda implementação desenvolvida.

Apresenta ainda 4 parâmetros relativos ao estado das máquinas: “Condição para Arranque”, “Parado”, “Marcha em Vazio” e “Marcha em Carga”.

No mostrador, entre outros, apresenta os valores dos parâmetros relativos à velocidade e potência que autómato impõe à máquina.

Finalmente, apresenta ainda a fase de arranque e de paragem e a indicação de uso de sincronizador e excitação da máquina.

## 5 Conclusões e perspetivas de trabalho futuro

As especificações do projeto relativas às implementações da aplicação *web*, delineadas pela ARUM, foram concretizadas com o sucesso e mereceram parecer positivo da parte dos responsáveis da ARUM e dos seus utilizadores. Após a realização de vários testes às funcionalidades das implementações, não foram encontradas ambiguidades.

A segunda implementação foi concretizada para permitir o armazenamento externo do histórico dos parâmetros do autómato numa base de dados SQL *server*.

Como perspectivas de trabalho futuro, considera-se necessário o uso de *store procedures* na segunda implementação, conforme recomenda a Siemens em [5], para aumento do volume de dados a transmitir pelo protocolo de comunicação TDS, evitando o particionamento da *string* de dados. Nesse sentido, considera-se potencialmente útil permitir a alteração dos restantes parâmetros do autómato a partir da aplicação *web*.

Finalmente, considera-se igualmente útil inserir duas caixas de *input* de dados à página principal da aplicação *web*, para que o utilizador possa definir a data de início e de fim do período de tempo a partir do qual pretende representar graficamente a velocidade de controlo que o autómato aplica à máquina ao longo do tempo.

## Referências

1. TIA Portal, <https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html>, último acesso a 01/04/2022.
2. S7-1500 Web Server, [https://cache.industry.siemens.com/dl/files/496/68011496/att\\_959527/v2/68011496\\_Examples\\_for\\_S7WebServer\\_DOC\\_v21\\_en.pdf](https://cache.industry.siemens.com/dl/files/496/68011496/att_959527/v2/68011496_Examples_for_S7WebServer_DOC_v21_en.pdf), último acesso a 01/04/2022.
3. Creating and using user-defined web pages on S7-1200 / S7-1500, <https://support.industry.siemens.com/cs/document/68011496/creating-and-using-user-defined-web-pages-on-s7-1200-s7-1500?dti=0&lc=en-US>, último acesso a 01/04/2022.
4. Matias, R.: Monitorização via internet de processos industriais controlados com autómatos programáveis, ISEL, Lisboa (2014).
5. Create pages on S7-1500, [https://cache.industry.siemens.com/dl/files/496/68011496/att\\_1018669/v1/68011496\\_S7-1200\\_1500\\_Webserver\\_DOC\\_v4\\_en.pdf](https://cache.industry.siemens.com/dl/files/496/68011496/att_1018669/v1/68011496_S7-1200_1500_Webserver_DOC_v4_en.pdf), último acesso a 01/04/2022.
6. Connecting S7-1500 to an SQL database, [https://cache.industry.siemens.com/dl/files/336/109779336/att\\_1104405/v1/109779336\\_SQL\\_DOC\\_en\\_V30.pdf](https://cache.industry.siemens.com/dl/files/336/109779336/att_1104405/v1/109779336_SQL_DOC_en_V30.pdf), último acesso a 01/04/2022.
7. SCL, [https://cache.industry.siemens.com/dl/files/188/1137188/att\\_27471/v1/SCLV4\\_e.pdf](https://cache.industry.siemens.com/dl/files/188/1137188/att_27471/v1/SCLV4_e.pdf), último acesso a 01/04/2022.

# Base de Dados e Integração com soluções de *Data Hologram*

Rui Soares e F. Jorge Duarte

Instituto Superior de Engenharia do Porto, Porto, Portugal

{1191045, fjd@isep.ipp.pt}

**Resumo.** As soluções de *Data Hologram* permitem criar um *Data Decoupling Layer* que promove uma nova visão de como os dados são disponibilizados para as aplicações. Estas são baseadas na confiança, confiança no processo *Extract, Transform and Load* (ETL) que propaga eventos dos *System of Records* e confiança na qualidade dos dados armazenados. A confiança pode ser obtida por uma forte Camada de Observabilidade, que regista, rastreia e monitoriza todos os processos ocorridos dentro do *Data Hologram* ou criando processos de monitorização que avaliam a qualidade dos dados da solução.

Com o objetivo de fornecer a confiança necessária às *Data Holograms*, foi criada uma solução configurável que é capaz de extrair os dados do *Data Hologram* e compará-los com os dados do *System of Records*. Esta solução é capaz de operar sobre amostras de dados que podem ser configuradas por meio de regras. Por fim, a saída dessas execuções é detalhada em *dashboards* predefinidas. Para tal, foram desenvolvidas três APIs com base em REST API, que permitem a extração de informação de várias bases de dados implementadas. Após o tratamento desses dados, estes são enviados para o Elasticsearch para serem visualizados em *dashboards* de simples compreensão, com o auxílio do Kibana.

Com a elaboração desta solução, tornou-se possível garantir maior confiança e segurança na qualidade dos dados armazenados, assim como garantir a rasterização e monitorização dos processos dentro do *Data Hologram*.

**Palavras-chave:** *Data Hologram*, *Customer Data Platform*, Soluções de Assurance

## 1 Introdução

### 1.1 Enquadramento

A Celfocus é uma empresa de tecnologia portuguesa, de extensão internacional e que tem como principal função, a busca de novas tecnologias, tendências e domínios. Uma das abordagens usadas pela Celfocus é o uso de soluções *Data Hologram* (DH), baseadas em soluções de *Data Hub* (coleção simples de dados organizados de várias fontes), que permitem uma partilha de dados mais eficaz, vinculando os *System of Records* (SoR) aos consumidores de dados, enquanto determina que dados necessitam de ser mediados. Um DH é basicamente uma solução para garantir que dados significativos fiquem mais próximos dos canais para melhorar o seu desempenho, garantindo que

algumas operações sejam realizadas mesmo quando houver alguma indisponibilidade do sistema.

A interação com o DH é feita através de uma camada de exposição de serviço, utilizando serviços que permitem às organizações controlar o que está exposto. Essa abordagem ajuda as organizações a terem uma visão mais agregada e rápida das informações, principalmente quando possuem grandes dados históricos e diversas fontes de registros. Permite que os dados sejam modelados e correlacionados, reduzindo a ambiguidade do “o que é um cliente”. Além disso, aumenta os recursos enquanto reduz os impactos no SoR e reduz o impacto do tempo de inatividade na experiência do cliente, com acesso quase em tempo real a dados críticos.

### 1.2 Contextualização

A adoção das soluções de DH é baseada na confiança, confiança no processo ETL que propaga eventos dos SoR e confiança na qualidade dos dados armazenados.

Anteriormente, apesar de existir confiança no processo descrito, a verificação dos dados era feita de forma primitiva, isto é, alguém teria de comparar os dados encontrados no DH com os encontrados nos SoRs. Este era um processo lento e que poderia dar origem a erros, caso os dados não fossem verificados com toda a atenção necessária.

A confiança pode ser obtida por uma forte Camada de Observabilidade, que registra, rastreia e monitoriza todos os processos ocorridos dentro do DH ou com a criação de processos de monitorização que acedem à qualidade dos dados da solução.

A solução de DH é composta por várias camadas, começando pela camada de transporte. Esta é a responsável pelas conexões aos sistemas fonte, responsável por transformar e modelizar os dados para dentro da camada de armazenamento e por lidar com os dados dentro do DH. Os dados podem ter uma representação única, podem ser distribuídos por mais do que um repositório e não precisam de ser modelados seguindo o formato dos sistemas de origem, podendo ser duplicados em várias estruturas para melhorar o desempenho da consulta. Os dados podem e devem utilizar padrões de mercado, como o TMForum SID para o segmento de Telecomunicações ou a BIAN para o segmento de Banca e Serviços Financeiros. Esta camada pode ser uma mistura de diferentes tecnologias [1]. A título de exemplo, parte dos dados pode ser armazenada em Document Based DB e outra em Graph DB. Podemos encontrar também a camada de exposição, responsável por expor os dados aos consumidores finais, sendo a exposição realizada somente via API's. A camada de observabilidade, permite a visão completa e integrada de todos os processos da solução, fornecendo uma forma de garantir a confiabilidade dos dados dentro do DH.

### 1.3 Objetivos

Com a finalidade de conseguir desenvolver a solução proposta, seria necessário satisfazer os seguintes objetivos:

- Criar um processo, microsserviço (MS), que iria lidar com o processo de reconciliação. Este processo consiste no processo de comparação de sistemas.

- O processo de reconciliação reagiria às regras, podendo essas regras ser gerenciadas por meio de uma API, um novo microsserviço.
- As saídas do processo de *Assurance*, deveriam ser armazenadas e um dashboard deveria expor os dados as mesmas. O processo de *Assurance* é o processo de comparação que garante que os sistemas permaneçam sincronizados. Ao encontrar irregularidades nos dados, serão automaticamente corrigidos.

## 2 Estado de arte

### 2.1 Trabalhos Relacionados

Para referência, foram escolhidos dois trabalhos que usam duas das tecnologias mais importantes neste projeto, o Java Spring Boot e o uso do Elasticsearch.

O primeiro, *Assurance case review using System Theory*, tem como foco o caso de segurança, o caso de garantia e o caso de confiabilidade. Desta forma, foram propostos métodos para o representar usando a Notação Estruturante de Metas (GSN). Regras de revisão para casos de garantia são propostas com base no GSN [7].

O segundo, *Interactive Database for the Storage and Maintenance of the Biological Data*, tem como foco principal desenvolver uma nova base de dados de estabilidade de proteínas que se concentrará no armazenamento e manutenção dos dados experimentais. O resultado é uma nova base de dados FireProtDB, fornecendo dados experimentais de todas as fontes de dados fundamentais ao trabalho a desenvolver. Ao mesmo tempo, apresenta os dados por meio de uma interface gráfica [8], tal como no trabalho desenvolvido na Celfocus em que todos os resultados adquiridos do processo principal que irá ser implementado, irão ser expostos em *dashboards*.

### 2.2 Tecnologias Usadas na Celfocus

A Celfocus disponibiliza soluções de *Assurance*. Desta forma, existem determinadas restrições relativas a tecnologias e software que podem ser utilizados no desenvolvimento das soluções, de forma a garantir que as diretrizes impostas pela empresa são cumpridas. A construção dos microsserviços devia seguir a Clean Architecture [2] desenvolvida em Java Spring Boot [3] e as bases de dados criadas teriam de ser implementadas em PostgreSQL [4] e MongoDB [5]. Para além disso, a parte que envolve a criação das *dashboards* e do processamento dos resultados deveria ser implementada com a ferramenta Elasticsearch [6] em conjunto com o Kibana.

## 3 Requisitos e Desenho

### 3.1 Requisitos Funcionais

Os requisitos funcionais a implementar foram divididos em casos de uso:

**Tabela 1.** Descrição dos casos de uso.

Caso de Uso	Descrição
<b>UC1 – Implementar MS de Gestão de Regras e Entidades</b>	Criar um MS (microserviço) com a funcionalidade de poder criar e gerir as regras e entidades que são usadas para realizar os processos de <i>Assurance</i> .
<b>UC2 – Implementar MS Assurance</b>	Criar um MS que faça todo o processo de <i>Assurance</i> , Este processo passa por adquirir as regras ao MS de gestão de regras e entidades, aplicar essas regras na base de dados, adquirir e comparar os dados obtidos no MS <i>Legacy</i> e finalmente, enviar os dados para o Elasticsearch.
<b>UC3 – Implementar MS Legacy</b>	Criar um MS que tem como objetivo simular a aquisição de dados dos SoRs.
<b>UC4 – Implementar Registo de Execução de Assurance</b>	Implementar a comunicação entre o MS <i>Assurance</i> e o Elasticsearch, assim como a configuração deste.
<b>UC5 – Implementar Dashboard de Execução de Assurance</b>	Criar um <i>dashboard</i> com a ferramenta Kibana para facilitar a compreensão dos dados guardados em Elasticsearch.
<b>UC6 – Implementar Caso de Uso de Observabilidade</b>	Implementar uma coleta de métricas de <i>logs</i> e criar um <i>dashboard</i> para mostrar a informação de <i>log</i> .

### 3.2 Requisitos Não Funcionais

Os requisitos não funcionais para a solução desenvolvida eram os seguintes: processamento de dados de forma dinâmica; MS de *Assurance* não necessita de mudanças a nível de código caso seja adicionado um novo documento na sua base de dados; as *dashboards* criadas deviam ser em inglês; os dados descritos nas *dashboards* corresponderiam aos guardados em resultados do processo de *Assurance*; deveria ser um processo eficaz que durasse menos de 60 segundos; deveria suportar múltiplas entidades e chamada de múltiplos MS pelo sistema fonte; as *dashboards* deveriam ser criadas com a ferramenta Kibana.

### 3.3 Desenho

O sistema segue o diagrama de fluxo de processos apresentado na figura 1. Neste diagrama é possível observar o funcionamento geral do processo de *Assurance*. Começa por receber um pedido com o nome de uma entidade existente no MS de Regras e Entidades. Caso o nome enviado exista no MS, necessita de verificar se existem regras no mesmo. Caso haja regras, estas são aplicadas no MongoDB para obter os dados da mesma. Para cada entidade resultante da aplicação das regras é executada a ligação ao MS *Legacy* que retorna nova informação. Depois de realizada a comparação entre os dados, é enviada informação para o Elasticsearch.

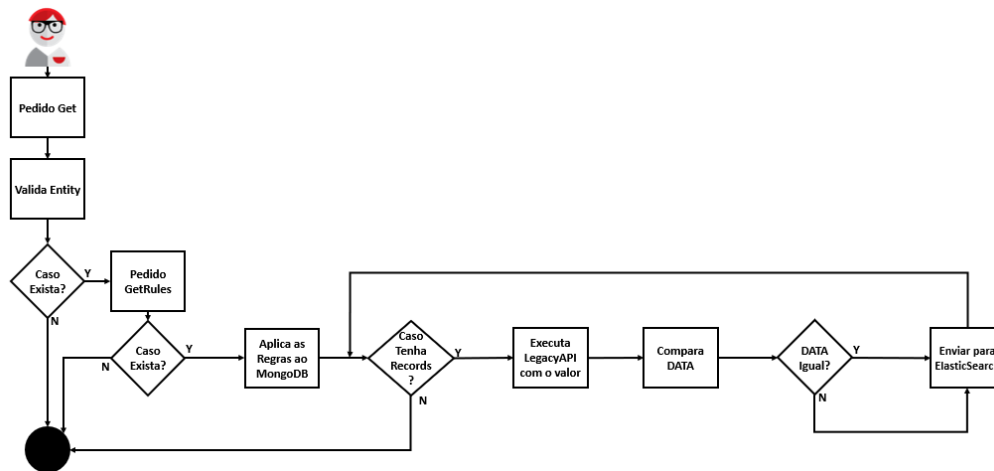


Fig. 1. Diagrama de Fluxo de Processos

## 4 Implementação da solução

O desenvolvimento da solução proposta neste projeto dividiu-se em dois momentos. O primeiro consistiu no desenvolvimento de três microsserviços que fornecem dados para que todo o processo de *Assurance* possa ser realizado. No segundo momento, os dados adquiridos do processo são enviados e guardados, para uma futura análise dos mesmos, criando *dashboards* com os dados essenciais para análise do processo.

Seguiu-se uma metodologia SCRUM [9], tendo sido realizadas diariamente pequenas reuniões para que se demonstrasse os avanços no desenvolvimento do projeto, fossem identificadas possíveis correções, resolvidas dúvidas que surgissem e fossem expostas possíveis alterações ou sugestões. Desta forma, foram propostos marcos, para que todo o projeto fosse implementado no prazo definido.

### 4.1 UC1 – Implementar MS de Gestão de Regras e Entidades

Este caso de uso foi dividido em duas partes distintas, sendo que a segunda foi subdividida em partes mais simples, com o objetivo de facilitar a compreensão de todas as funcionalidades que deviam ser implementadas. A primeira parte deste caso de uso consiste na criação de uma base de dados relacional e a implementação de um MS que crie e gere entidades e regras. A segunda subdivide-se em: POST de uma entidade, POST de uma regra com as condições associadas à mesma, PUT de uma entidade através do nome da mesma, PUT de uma regra através do nome da mesma; GET de uma entidade através do nome ou de todas em conjunto; GET de regras através do nome da entidade associada; e GET de entidades de *Assurance* através do nome da entidade associada ou pelo nome em conjunto com a data de execução da entidade de *Assurance*.

Foi escolhida a base de dados PostgreSQL porque para além de ser uma das imposições da empresa, tem uma instalação e configuração bastante simples, não tendo custos adicionais.

Na implementação deste caso de uso, foi usada uma ferramenta do *swagger* que permite, após a configuração de um ficheiro “.swager.json”, criar um esqueleto para o

projeto que iria ser criado. Com o uso desta ferramenta, as classes dos *Crontrrollers*, assim como as bases dos DTOs (*Data Transfer Object*) e dos *Services* foram criadas automaticamente, facilitando a implementação da parte mais técnica do MS.

Para além desta ferramenta, foi usada também a dependência *org.mapstruct* para gerar automaticamente *mappers* que permitiam passar dados de classes mais interiores para as exteriores e vice-versa. Por exemplo, de dados em *Core* para DAO (*Decentralized Autonomous Organization*). Com a finalização da implementação de todas as funcionalidades descritas nas fases mais simples do MS, foram criadas entidades e regras que seriam posteriormente usadas para a realização do processo de *Assurance*.

## 4.2 UC2 – Implementar MS *Assurance*

Para este caso de uso foi escolhida a base de dados MongoDB porque para além de ser uma das imposições da Celfocus, esta permite uma ligação simples com o MS. Após a instalação e configuração da mesma, foram introduzidos dados relevantes. Neste caso, foram introduzidos dados de três *customers* diferentes, com o uso do MongoDB Shell, onde é possível a introdução de dados através de um ficheiro json. Para a implementação deste MS, foi necessário criar ligações com todos os outros MS, assim como com o Elasticsearch.

Foi criado um projeto com o uso da ferramenta Spring Initializr fornecida com a aplicação IntelliJ. Concluída a criação do projeto, procedeu-se à criação de um *Controller*. Este recebe um pedido com o nome da entidade pela qual se pretende adquirir as regras ao MS de Gestão de Regras e Entidades. Com a chegada deste pedido, o nome obtido é enviado para o *Service* que desencadeia o processo de *Assurance*.

Este processo inicializa-se com um pedido GET ao MS de Gestão de Regras e Entidades, com o nome recebido anteriormente e recebe toda a informação ligada à entidade.

Para cada regra associada à entidade, é realizada uma recolha de dados na base de dados. Esta é realizada de forma dinâmica, com o objetivo de conseguir adquirir qualquer informação da base de dados, independentemente da *collection* em que estes se encontram, dos operadores necessários para a pesquisa e das conjunções entre os mesmos. Este processo, apesar de ser eficaz, tem algumas incompatibilidades com regras mais complexas.

Após adquiridos os dados provenientes da aplicação das regras na base de dados, é feita para cada entidade obtida, a comparação da mesma com dados obtidos através da ligação com o MS de *Legacy*. Esta comparação, tal como todo o processo, tem como objetivo principal ser feito de forma dinâmica. Assim, os dados encontram-se sempre nos formatos Object ou JSONObject, de forma a garantir o dinamismo pretendido com o processo.

Após a comparação dos dados obtidos, é atualizada a data em que o processo de *Assurance* foi efetuado para a entidade na base de dados, MongoDB. Esta atualização também é feita de forma dinâmica, finalizando-se com a criação da *AssuranceEntity*, no MS de Gestão de Regras e Entidades, de forma a guardar a data em que a entidade realizou o processo de *Assurance*.

### 4.3 UC3 – Implementar MS *Legacy*

Este caso de uso, tal como os anteriores, foi inicializado pela criação de um projeto, através da ferramenta Spring Initializr, fornecida com a aplicação IntelliJ. Sendo que a funcionalidade deste MS é apenas inicializada através de um pedido GET, feito pelo MS de Assurance, foi criado um Controller que recebe o pedido. Neste irá estar incluído o `fieldId`, nome pelo qual o campo que se pretende procurar é conhecido no ficheiro criado, e o `fieldValue` que tem de ser pesquisado nos dados do ficheiro.

Após recebido o pedido, o *Controller* envia os dados recebidos para o *Service*. Neste é feito o processamento dos dados, isto é, neste os dados do ficheiro são analisados e quando forem encontrados os dados pretendidos, são retornados para o *Controller* que consequentemente envia os mesmos para o MS de *Assurance*.

O caso de uso conclui-se com a criação de um ficheiro no formato `.csv` que simule os dados encontrados nos ficheiros SoRs.

### 4.4 UC4 – Implementar Registo de Execução de *Assurance*

Este caso de uso tem como principal objetivo, enviar os dados adquiridos pela realização do processo de Assurance para o Elasticsearch, para que possam ser posteriormente analisados com a criação de dashboards com a ferramenta Kibana.

Os dados escolhidos para o envio, foram dados relevantes para garantir a confiança do processo, analisar o tempo que demoraria a executar e a guardar a quantidade de dados analisados no processo, assim como os resultados das comparações entre os dados guardados no MongoDB e os adquiridos na chamada do MS de *Legacy*, isto é, com os dados dos SoRs. Foram criadas dois *templates* no Kibana, um para as regras e outro para as entidades.

### 4.5 UC5 – Implementar Dashboard de Execução de *Assurance*

A implementação deste caso de uso foi realizada na ferramenta Kibana, com o objetivo de criar uma dashboard com dados relevantes do processo de Assurance.

Desta forma, foram criadas 7 vistas para possibilitar uma análise completa de todos os dados fundamentais. As duas primeiras vistas expõem a quantidade de registos de entidades e de regras. A vista seguinte representa o número de entidades que cada regra obteve da aplicação das regras, em função da quantidade de registos na forma de um gráfico. De seguida, foi implementada uma vista que apresenta a duração do processamento de cada regra, também em função da quantidade de registos novamente em forma de gráfico. A quinta vista apresenta a percentagem de comparações em que os dados obtidos não foram semelhantes, representadas com um *False* e as comparações semelhantes representadas com um *True*.

### 4.6 UC6 – Implementar Caso de Uso de Observabilidade

A finalidade deste caso de uso é obter logs do processo de Assurance. Estes logs são necessários para caso ocorra algum erro, este seja facilmente identificado. É identificada também, a funcionalidade que causou o erro. Também é possível identificar

anomalias que possam ocorrer não só no processo de Assurance, mas também na conexão com a base de dados do MongoDB ou com a ligação aos outros MS.

Foram estipulados os dados necessários para uma boa compreensão dos *logs*, de forma a obter a informação necessária para o bom funcionamento do programa. Foi criado um *template* no Kibana com os dados relevantes à compreensão dos *logs*, de forma a garantir a estrutura dos dados pretendida.

Após a implementação dos *logs* e do envio dos mesmos para o Elasticsearch, estes são visualizados através de *dashboards* criadas no Kibana. Foram criadas 4 vistas, de forma a facilitar a compreensão dos *logs*. A primeira, informa a quantidade de *logs* existentes no sistema. A segunda, é representada por um gráfico que demonstra a quantidade de registos de *log* efetuados por dia. As duas últimas, são representadas por duas tabelas, uma tabela denominada Master com os Ids dos *logs*. A tabela Detail demonstra os dados escolhidos anteriormente, facilitando e tornando mais eficaz a procura de informação.

## 5 Conclusões

Todos os requisitos funcionais e não funcionais foram satisfeitos a 100%, visto que todas as funcionalidades pretendidas nos casos de uso foram desenvolvidas, testadas e avaliadas e todos os requisitos impostos para a realização da solução foram completados. Com este projeto construiu-se uma peça bastante importante para fornecer credibilidade às soluções *Data Hologram* e desta forma, contribuir para a garantia dos dados e processos da solução. Construiu-se também a base para uma solução genérica e agnóstica da solução que permite a aplicação desta solução em múltiplas instâncias do *Data Hologram*.

## Referências

1. “Information Framework (SID),” *TM Forum*. <https://www.tmforum.org/oda/information-systems/information-framework-sid/> (accessed Oct. 23, 2022).
2. “Clean Coder Blog,” <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> (accessed Sep. 29, 2022).
3. “Spring Boot,” <https://spring.io/projects/spring-boot> (accessed Sep. 29, 2022).
4. “PostgreSQL: About,” <https://www.postgresql.org/about/> (accessed Sep. 29, 2022).
5. “MongoDB: The Developer Data Platform,” *MongoDB*. <https://www.mongodb.com> (accessed Sep. 29, 2022).
6. “Free and Open Search: The Creators of Elasticsearch, ELK & Kibana | Elastic.” <https://www.elastic.co/> (accessed Sep. 29, 2022).
7. S. Yamamoto and S. Morisaki, “Assurance case review using System Theory,” *Int J Jpn Assoc Mgt Sys*, vol. 11, no. 1, pp. 35–40, Dec. 2019, doi: 10.14790/ijams.11.35.
8. “final-thesis.pdf.” Accessed: Oct. 29, 2022. [Online]. Available: <https://dspace.vutbr.cz/bitstream/handle/11012/200160/final-thesis.pdf?sequence=-1>
9. “What Is Scrum Methodology? & Scrum Project Management,” Sep. 30, 2019. <https://www.digite.com/agile/scrum-methodology/> (accessed Sep. 29, 2022).

## Channel TV: Power Platform CRM System

Nuno Soares<sup>1</sup> e Emanuel Silva<sup>1</sup>

<sup>1</sup> Instituto Superior de Engenharia do Porto, Porto, Portugal  
{1190916, ecs}@isep.ipp.pt

**Resumo.** Perante a necessidade de se adaptar à nova realidade da era digital, o mercado das empresas que operam na área televisiva é cada vez mais competitivo e exigente. O nascimento de plataformas de Streaming é um desafio que as empresas neste setor têm de enfrentar. Neste sentido, é essencial existir um investimento na área digital para manter a competitividade no mercado.

Desta forma, a empresa Channel TV, maior rede de televisão pública comercial francesa, tem vindo a investir cada vez mais em conteúdo digital. Tendo em conta o aumento de visualizações que este investimento trouxe, a empresa pretende melhorar o seu sistema de BackOffice, de forma a melhorar a gestão de dados entre os diversos departamentos da empresa. Para além disso, pretende ainda melhorar a prestação de serviços de apoio ao cliente, com o objetivo de aumentar a sua satisfação.

Ao longo do presente artigo será descrito o processo iterativo que foi realizado para o desenvolvimento de uma solução que visa resolver o problema apresentado pela empresa. Este processo envolve a recolha e análise de requisitos, desenho e implementação de uma solução, bem como a sua testagem, análise dos resultados e conclusões obtidas.

**Palavras-chave:** Desenvolvimento Low-Code, Plataforma CRM, Apoio ao Cliente, BackOffice, Microsoft Power Platform, Dynamics 365.

### 1 Introdução

A evolução da tecnologia e dos meios de comunicação, vem a moldar e influenciar o mundo em que vivemos. Com a vinda da 4ª revolução industrial, esta dependência tornou-se mesmo uma realidade, influenciando as leis e convenções que regem não só a indústria, como também a sociedade humana em que vivemos. Tal como muitas outras áreas, o mundo empresarial sofreu grandes alterações e evoluções. Para as empresas se conseguirem destacar nesta era digital necessitam de evoluir e se adaptar.

O presente artigo foi elaborado sobre uma proposta requerida pela empresa Channel TV (o nome real omitido por motivos de confidencialidade), a maior rede de televisão pública comercial francesa, que no ano passado conseguiu obter mais de 21% do número total de espectadores franceses.

Perante esta necessidade de se adaptar à nova realidade da era digital, a empresa tem vindo a investir cada vez mais em conteúdo digital. Neste momento, conta com

mais de 14500 horas de conteúdo disponível para Streaming e, cerca de 24000 utilizadores registados na sua plataforma digital.

Devido ao aumento do número de visualizações da Channel TV, surgiu a necessidade de melhorar a gestão da relação com os clientes e facilitar o trabalho dos colaboradores da empresa. Tendo isto em conta, a Hitachi Solutions propôs o desenvolvimento de um sistema fiável, intuitivo e fácil de usar pelos seus funcionários.

### 1.1 Objetivos

O objetivo geral deste trabalho é elaborar um sistema único e conciso, capaz de facilitar a gestão dos diversos departamentos da Channel TV, melhorando assim a experiência do utilizador, auxiliar na relação entre os clientes e os agentes publicitários, bem como prestar suporte ao cliente sempre que necessário. Tendo isto em conta, foram definidos os seguintes objetivos específicos:

- Recolher informação sobre potenciais consumidores e agentes publicitários;
- Gerir a informação sobre os diversos contactos dos clientes;
- Criar e gerir os contratos com os clientes e as suas respetivas informações;
- Facilitar a renovação e auditoria dos contratos;
- Gerir os diversos produtos fornecidos pela empresa;
- Facilitar a prestação de apoio ao cliente quando necessário;
- Melhorar o serviço pós-venda da empresa;
- Facilitar o registo do processo de venda;
- Automatizar processos de negócio, de modo a otimizar a prestação dos serviços.

## 2 Estado da arte

Tendo em conta a crescente necessidade de criação de software personalizado que surgiu durante a pandemia de COVID-19, as empresas recorreram à utilização de aplicações Low-Code [2] para solucionar os seus problemas.

Devido à natureza da abordagem de desenvolvimento Low-Code, é possível acelerar a criação de aplicações funcionais, pois como não existe a necessidade de programar manualmente toda a aplicação, é possível reduzir drasticamente o tempo de desenvolvimento. É ainda possível integrar as soluções desenvolvidas com sistemas externos, para tal, são utilizados e configurados conetores pré-existentes que estabelecem a ligação com estes sistemas externos. Como as funcionalidades mais básicas do sistema já se encontram desenvolvidas, o programador pode dedicar-se ao desenvolvimento de funcionalidades mais específicas e prioritárias para o negócio [1], [2].

### 2.1 Abordagem *Low-Code* vs. Tradicional

Existe uma ideia pré-concebida de que o desenvolvimento tradicional e o desenvolvimento Low-Code são incompatíveis, contudo, a realidade é mais complexa. A

abordagem utilizada em cada um dos desenvolvimentos acaba por seguir os mesmos princípios lógicos e os objetivos de desenvolvimento são os mesmos [1], [3].

Cada tipologia de desenvolvimento apresenta as suas vantagens e desvantagens. Neste sentido, dependendo dos requisitos do projeto, deve-se utilizar a abordagem mais adequada para o problema, tendo em consideração os seus pontos fortes e fracos.

#### *Low-Code*

O desenvolvimento Low-Code apresenta algumas características intrínsecas, nomeadamente, a não necessidade de programar todas as funcionalidades da solução, a utilização de módulos e/ou templates pré-existentes e o uso de uma interface drag-and-drop. As vantagens que mais se destacam são as seguintes [1] – [3]:

- Velocidade – Devido à falta da necessidade de programar todas as funcionalidades da solução, a velocidade de desenvolvimento e, o tempo necessário para o desenvolvimento de uma solução, é reduzido drasticamente;
- Reutilização – Devido à existência de módulos pré-existentes, o desenvolvedor pode proceder à reutilização das funcionalidades em diferentes projetos;
- Custo de desenvolvimento de software – O custo de desenvolvimento deixa de ser proveniente de despesas com os desenvolvedores de software, que têm um preço por hora associado, para o custo da plataforma de Low-Code.

Algumas das desvantagens relevantes são [1] – [4]:

- Qualidade – A qualidade destas aplicações tende a ser inferior, face a aplicações cujo código e arquitetura são definidos especificamente para o caso de uso em questão, pois o código gerado pela plataforma tem de ser mais genérico;
- Personalização – Como o fluxo e gestão do funcionamento do programa é gerido pela plataforma em que a solução foi desenvolvida, estas soluções encontram-se limitadas às opções de personalização disponibilizadas pela plataforma;
- Opções de Deployment – As opções de Deployment encontram-se limitadas a serviços de cloud especificados pela plataforma de desenvolvimento;
- Dependência de terceiros – Na maioria dos casos, caso a plataforma deixe de ser mantida ou deixe de existir, existe a impossibilidade de utilizar a solução desenvolvida numa plataforma Low-Code noutra ambiente.

#### *Tradicional*

Em relação ao desenvolvimento tradicional, este destaca-se positivamente pelo controlo e flexibilidade que proporciona para a aplicação, visto que toda a aplicação pode ser configurada e manipulada pelo desenvolvedor, bem como pela possibilidade de realizar o *Deployment* da aplicação para qualquer *Cloud* desejada [1] – [4].

Apesar disto, esta abordagem consome muito mais tempo que a utilização de código gerado automaticamente por ferramentas *Low-Code*. Isto faz com que o tempo de desenvolvimento e as competências técnicas necessárias para o desenvolvimento tradicional sejam maiores perante o desenvolvimento *Low-Code*.

Contudo, podemos afirmar que, a melhor abordagem para o desenvolvimento de software é a utilização de ambas as abordagens. Isto permite obter as vantagens das duas abordagens, agilizando o processo de desenvolvimento de software, enquanto se

mantém as vantagens de uma abordagem tradicional, obtendo-se assim uma solução de elevada qualidade e eficiência que soluciona os problemas do cliente.

## 2.2 Sistemas de gestão de informação

Os sistemas de gestão de informação têm manifestado cada vez mais um papel determinante para tornar um negócio bem-sucedido. Na verdade, estes sistemas são desenvolvidos focando-se em três pontos fundamentais: a redução de custos, aumentar a qualidade do serviço e aumentar as receitas geradas pela empresa [5].

### Sistemas CRM

Os sistemas informação de Customer Relationship Management (CRM), são constituídos por componentes que permitem melhorar a interação da empresa com os clientes, sendo os seguintes os mais relevantes [6]:

- Recolher informação sobre os contactos de todos os clientes;
- Classificar e organizar os clientes, tendo em conta as suas relevâncias;
- Permitir a troca de informações entre os colaboradores;
- Recolher e armazenar informações sobre os competidores.

Desta forma, através da análise dos dados recolhidos e armazenados no sistema, este deve ser capaz de melhorar a estratégia da empresa face aos clientes, e ao mesmo tempo tornar mais eficiente o trabalho dos colaboradores ao interagir com os clientes.

Algumas das vantagens mais evidentes da utilização deste tipo de sistemas, são:

- Libertar o colaborador da empresa de tarefas mais complexas e exaustivas, através da automatização das mesmas;
- Possibilitar aumentar o número de negócios efetuado;
- Manter os dados dos clientes todos armazenados num único sistema que seja acessível a todos os colaboradores responsáveis.

### Sistemas ERP

As soluções Enterprise Resource Planing (ERP), cobrem vários aspetos da gestão da produção, e ainda várias funções de negócio, como vendas, finanças, entre outros. Os sistemas podem ser descritos através dos seguintes pontos [7]:

- Ferramentas de gestão empresarial que balanceiam a oferta e a procura;
- Capacidade de conectar os consumidores e fornecedores numa rede integrada;
- Estabelecimento de processos de apoio a decisão;
- Fornecimento de serviços de aumento de produtividade.

Desta forma, as principais razões pelas quais as empresas adotam a implementação de sistemas ERP são [7]:

- Melhorar e atualizar o sistema para ter novas funcionalidades competitivas;
- Ajustar o crescimento da empresa;

- Redução de custos, através da automação de diversos processos de negócio.

### Sistemas CRM e ERP.

Nos dias de hoje, um dos maiores focos do mercado é a satisfação do consumidor, e as empresas tentam a todo o custo melhorar esta relação, enquanto tentam ao mesmo tempo conciliar uma manutenção e gestão dos processos de negócio da empresa.

Tendo isto em conta, é cada vez mais evidente que é necessária a integração de sistemas de CRM e de ERP numa empresa, pois estes têm o mesmo objetivo que é aumentar o lucro gerado da empresa. Por um lado, um sistema ERP tem como objetivo primário a redução de custos e a otimização de processos de negócio, por outro, os sistemas CRM, focam-se no aumento do número de vendas e na melhoria da relação com os clientes. Podemos assim concluir que, a integração destes sistemas é essencial para a competitividade de uma empresa no mercado, pois permite à mesma reduzir os seus custos de operação e produção, enquanto aumentam o número de vendas e clientes, gerando assim um maior lucro para a mesma.

## 3 Análise e desenho da solução

De seguida é apresentado o resultado da análise do problema, nomeadamente, o levantamento dos requisitos e da arquitetura do sistema.

### 3.1 Requisitos funcionais

Os requisitos funcionais correspondem ao conjunto de funcionalidades que uma solução deve oferecer, ou seja, qualquer funcionalidade que o sistema deve ser capaz de realizar [8]. A Tabela 1, descreve os requisitos funcionais do projeto em mãos.

**Tabela 1.** Requisitos Funcionais

UC01 – Criar novos registos no sistema
UC02 – Atualizar/Editar registos existentes no sistema
UC03 – Aceder a uma Dashboard com informações relevantes
UC04 – Visualizar o tempo restante para cumprir as KPIs
UC05 – Distribuir os Cases pelas diversas Queues
UC06 – Registrar as lições aprendidas durante a resolução de Cases
UC07 – Gerir os Bookable Resources da organização
UC08 – Alocar os Bookable Resources para a resolução de Work Orders
UC09 – Aceder a uma Schedule Board com as alocações dos Bookable Resources

### 3.2 Requisitos não funcionais

Os requisitos não funcionais focam-se nas restrições impostas sobre o sistema. Têm como foco principal definir como o sistema se deve comportar para responder aos

requisitos funcionais, de modo a proporcionar uma boa experiência ao seu utilizador [9]. O modelo FURPS+ foi utilizado no levantamento destes requisitos, dos quais foram identificados os seguintes:

**Funcionalidade**

- Deverá ser mantido um histórico de alterações e acesso aos dados do sistema;
- Deverá ser possível observar o progresso do processo de vendas e de resolução de incidentes, bem como devem ser agilizados os processos;
- Os dados devem apenas ser acedidos pelas entidades devidamente autorizadas;

**Usabilidade**

- O sistema desenvolvido deverá apresentar uma boa documentação que permita a futuros desenvolvedores entender a organização, desenho e implementação;

**Confiabilidade**

- O sistema deve-se encontrar devidamente testado com testes funcionais, de forma a garantir a sua qualidade, robustez e confiabilidade;

**Suportabilidade**

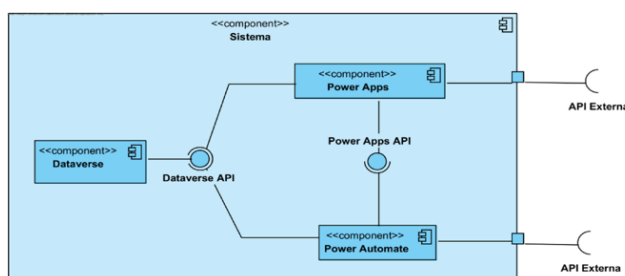
- O sistema deve ser suportados por diferentes formatos e dispositivos;

**Restrições de Implementação**

- O sistema deve ser desenvolvido utilizando a plataforma de desenvolvimento Low-Code da Microsoft, Microsoft Power Platform, e deve fazer uso do seu portefólio de plataformas CRM, Dynamics 365.

**3.3 Arquitetura do Sistema**

A plataforma de desenvolvimento Low-Code a utilizar durante o desenvolvimento da solução, Microsoft Power Platform, é composta por diversos componentes. Contudo, apenas serão utilizados os requeridos para a elaboração da mesma. Na Fig. 1, podemos observar um artefacto, diagrama de componentes, onde se encontram representados os diversos componentes do sistema e as interações entre si. O componente Dataverse, é responsável por assegurar a segurança e a persistência dos dados no sistema, por outro lado, o Power Apps é utilizado para criar interfaces gráficas e a estruturar a forma como estas interagem entre si, e o Power Automate, tem como função automatizar processos do sistema.



**Fig. 1.** Diagrama de Componentes

## 4 Implementação

O desenvolvimento da solução foi efetuado utilizando as tecnologias Microsoft, mais concretamente, a sua plataforma de desenvolvimento Low-Code, Microsoft Power Platform, e o repositório de plataformas Low-Code, Dynamics 365. Para garantir a segurança e integridade dos dados, desde o momento inicial do desenvolvimento foi tido em consideração as permissões de acesso ao sistema e aos seus dados.

De modo a implementar os requisitos identificados, é necessário proceder à modelação dos dados através da definição das entidades necessárias no Dataverse. Algumas das entidades já se encontram definidas no Dynamics 365, pelo que apenas é necessário incluí-las na solução, e proceder à alteração das vistas pré-existentes, tendo em conta as necessidades e requisitos do sistema. As restantes têm de ser devidamente criadas e personalizadas para permitir o cumprimento dos requisitos do sistema.

De modo a otimizar a prestação dos serviços, foram implementados vários mecanismos de automação, desde BPFs<sup>1</sup>, Flows<sup>2</sup> e Workflows<sup>3</sup>. Através destes, o colaborador da empresa não necessita de realizar alguns processos de forma ineficiente e demorada, tal como a renovação automática de contratos, onde os dados são copiados para o novo com as devidas alterações. Para além disso, são realizados automaticamente sem a necessidade de intervenção humana, permitindo que o colaborador possa reutilizar o seu tempo para a conclusão de tarefas, eventualmente, mais essenciais.

Com o intuito de melhorar o serviço pós-venda da empresa, foram implementados vários mecanismos e estabelecidas métricas que os colaboradores da organização têm de cumprir aquando da prestação de apoio ao cliente. O cumprimento dos tempos estabelecidos permite melhorar a satisfação do cliente.

### 4.1 Testes

A realização de uma boa cobertura de testes é essencial para prevenir bugs, reduzir custos de desenvolvimento, pois permite a descoberta antecipada de problemas no sistema e permite ainda a melhoria da performance do sistema [10].

Para a realização de testes à plataforma de CRM, a empresa faz uso de um software proprietário de acesso exclusivo a colaboradores da organização. Este software recorre à utilização de um ficheiro Excel onde se discrimina os passos a seguir e os resultados esperados. Contudo, este software não se encontra disponível aos estagiários, impossibilitando a elaboração de testes, através deste software à plataforma CRM desenvolvida. Foram realizados testes funcionais manuais para verificar que todas as

---

<sup>1</sup> Business Process Flows – Processo de negócio que permite assegurar que os utilizadores introduzam os dados no sistema de maneira consistente (<https://learn.microsoft.com/en-us/power-automate/business-process-flows-overview>).

<sup>2</sup> Conjunto de passos pré-definidos que vão ser executados por um evento, definido pelo utilizador através do componente Power Automate (<https://learn.microsoft.com/en-us/power-automate/flow-types>).

<sup>3</sup> Conjunto de etapas pré-definidas para atingir um certo objetivo ou cumprir uma finalidade. Estas são definidas através do componente Power Apps (<https://powerusers.microsoft.com/t5/Power-Apps-Community-Blog/PowerApps-Workflow-Processes-Part-1-Introduction/ba-p/98289>).

funcionalidades eram cumpridas com sucesso. Para os restantes componentes, foram utilizadas ferramentas próprias de realização de testes funcionais e end-to-end das aplicações do tipo Canvas App e os Flows do Power Automate.

## 5 Conclusões e trabalho futuro

Todos os objetivos definidos inicialmente (na secção 1.1), bem como todos os requisitos, tanto funcionais como não funcionais, foram cumpridos, atingidos e implementados com sucesso. Por tudo isto, o autor considera que a solução apresentada responde integralmente às necessidades apresentadas pela empresa.

Contudo, detetou-se que seria pertinente implementar as seguintes melhorias:

- Redesenho do layout das aplicações móveis;
- Automatizar novos processos;
- Implementar uma aplicação móvel para os colaboradores de apoio ao cliente;
- Nova aplicação móvel para os recursos da empresa acederem as suas alocações;
- Melhoria e revisão das permissões de segurança.

## Referências

1. C. Tozzi, “A practical take on low-code vs. traditional development”, 2022, <https://www.techtarget.com/searchsoftwarequality/tip/A-practical-take-on-low-code-vs-traditional-development>, último acesso 16/07/2022.
2. OutSystems, “What is Low Code”, 2022, <https://www.outsystems.com/guide/low-code/>, último acesso 16/07/2022.
3. “Low Code vs. Traditional Development”, <https://steelkiwi.com/blog/traditional-coding-vs-no-codelow-code-development/>, último acesso 22/07/2022.
4. “Low Code Platform: The Future of Software Development”, <https://www.cuelogic.com/blog/low-code-platform>, último acesso 23/07/2022.
5. K. C. Laudon and J. P., “*Management information systems: managing the digital firm*”, 15th ed. (2018).
6. Salesforce, “CRM Software: Customer Relationship Management”, <https://www.salesforce.com/ap/crm/>, último acesso 02/08/2022.
7. FCNN, “The Role of ERP resources planning systems”, <https://eds.p.ebscohost.com/eds/pdfviewer/pdfviewer?vid=1&sid=83b3a0b3-b938-419d-b1b1-7b37b6970892%40redis>, último acesso 03/08/2022.
8. Guru99, “What is a Functional Requirement in Software Engineering?”, <https://www.guru99.com/functional-requirement-specification-example.html>, último acesso 26/08/2022.
9. “What are Non Funcional Requirements?”, <https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples>, último acesso 29/08/2022.
10. IBM, “What is Software Testing and How Does it Work?”, <https://www.ibm.com/topics/software-testing>, último acesso 07/09/2022.

# Computação em nuvem: Infraestrutura para o desenvolvimento colaborativo de Aplicações

António Fernandes e F. Jorge Duarte

Instituto Superior de Engenharia do Porto, Porto, PT  
{1190402,fjd}@isep.ipp.pt

**Resumo.** Este trabalho consistiu numa avaliação da migração do centro de dados da ARMIS a partir da empresa Ozono (empresa do grupo ARMIS) no Porto para a nuvem, tendo sempre em conta o Well-Architected Framework (WAF) e os seus 5 pilares. Teve como objetivo, analisar todas as possibilidades (solução local, na nuvem ou modelo híbrido) e escolher a mais adequada. No início utilizou-se o Azure Migrate para efetuar um levantamento detalhado de Hardware (CPU, RAM e disco), o inventário de Software (aplicações e serviços), os servidores SQL e as dependências entre servidores. De forma a verificar o consumo energético mensal, foram identificadas as características de cada dispositivo. Para concluir o estudo sobre os custos, com o administrador de sistemas, foram identificados os custos de software. Na fase seguinte, ocorreram reuniões com todas as equipas da empresa, onde foi possível identificar o uso do centro de dados por cada uma, as suas necessidades e preocupações, sendo que ainda ocorreram duas reuniões com o administrador de sistemas, de forma a conhecer melhor o centro de dados e todos os servidores e serviços que este suporta. Foram ainda analisados orçamentos de armazenamento e confrontaram-se os mesmos com os custos em Azure. Tendo em conta todo o levantamento efetuado, decidiu-se que o cenário ideal seria uma arquitetura híbrida. Foram identificados os serviços necessários (Azure), os respetivos custos e por fim redesenhou-se a arquitetura.

**Palavras-chave:** Computação em nuvem, Well-Architected Framework, Engenharia de nuvem, Centro de dados

## 1 Introdução

Muitas das empresas têm os seus próprios servidores, nos seus centros de dados, mas, desde há alguns anos, a nuvem veio revolucionar o mundo dos centros de dados, os trabalhos em ambiente colaborativo, a disponibilidade de serviços e a recuperação em situações de desastre. A computação em nuvem é uma tecnologia que permite hospedar vários recursos (como servidores, bases de dados), programas e informações através da internet [1]. Esta permite reduzir custos de infraestrutura, a poupança de espaço, a centralização de informação, o dimensionamento automático (capacidade de escalar mediante as necessidades) e ainda o trabalho remoto.

O objetivo deste projeto foi avaliar um centro de dados através de um levantamento de todos os servidores (hardware, sistema operativo e software), seguido de uma análise

de custos e de reuniões com as equipas da ARMIS, de forma a averiguar as necessidades de cada uma, para no final, se desenhar a nova arquitetura de rede.

Um dos marcos importantes deste trabalho foi criar mecanismos de tolerância a falhas através do Azure (fornecedor de nuvem escolhido pela empresa) e avaliar o que se deveria migrar para a nuvem. Cada vez mais, é visível que a informação é um dos bens mais valiosos na sociedade. Como tal, queremos sempre que esteja disponível a toda a hora e em qualquer lugar.

De forma a criar uma arquitetura de acordo com as boas práticas da engenharia da nuvem, recorreu-se ao uso do Well-Architected Framework do Microsoft Azure.

## 2 Boas práticas e ferramentas essenciais ao trabalho

### Well-Architected Framework

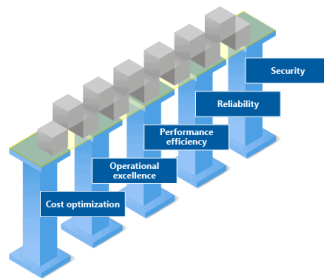


Figura 1 - Pilares do Well-Architeted Framework do Azure [2]

O **Well-Architected Framework** [3], ilustrado na figura 1, é um guia sobre os pilares, padrões e boas práticas de design em nuvem. Os pilares que define são:

- **Pilar da otimização de custos** [4]: defende que os custos devem ser otimizados e deve existir uma monitorização constante dos custos e dos recursos utilizados, de forma a perceber se são os ideais, excessivos ou se estão abaixo das necessidades (através do Azure Advisor, por exemplo).
- **Pilar da confiabilidade** [4]: garante que as cargas de trabalho são resilientes, resistentes a falhas e que permitam a maior disponibilidade possível.
- **Pilar da segurança** [5]: as aplicações e os dados devem estar seguros.
- **Pilar da excelência operacional** [3]: abrange todos os processos e operações que mantêm a infraestrutura e as aplicações a funcionar. É importante que as implementações sejam fiáveis e previsíveis, sendo a forma mais simples de reduzir o erro humano, a automatização de processos (DevOps, por exemplo).
- **Pilar da eficiência de desempenho** [6]: é a capacidade de a carga de trabalho ser escalável mediante as necessidades/exigências (através do dimensionamento automático, que consiste no aumento ou redução dos recursos, de forma automática, mediante as necessidades).

### Azure Site Recovery

O Azure Site Recovery [7] é uma ferramenta do Azure que permite automatizar os procedimentos de recuperação em situações de desastre. Através da instalação de uma aplicação nos servidores, os mesmos ficam de imediato associados à conta Azure e a partir daí, é possível orquestrar todo o cenário de recuperação de desastres [8].

#### **Azure Migrate**

O Azure Migrate [4] é um software que facilita o processo de migração para a nuvem, desde a avaliação (levantamento de servidores, bases de dados e aplicações), passando pelas sugestões e dicas para migrar para a nuvem, terminando com a automatização do processo de migração.

### **3 Análise**

#### **3.1 Ficheiros exportados do Azure Migrate**

Após a instalação do Azure Migrate num servidor, efetuou-se automaticamente o levantamento completo do centro de dados. Da análise dos ficheiros foi possível observar:

- Informação de todos os servidores e máquinas virtuais.
- Informação de todas as bases de dados SQL server.
- Custos associados a cada uma das máquinas virtuais e bases de dados.
- Restrições de migração (alguns impedimentos que poderiam vir a verificar-se no momento da migração).

As sugestões passaram por migrar todas as máquinas virtuais e servidores (foram sugeridos os tamanhos a utilizar) e manter as bases de dados SQL server em máquinas virtuais. Com a Total Cost Ownership Calculator (TCO) do Azure [9], foi possível constatar que se pouparia bastante ao migrar o centro de dados para a nuvem.

#### **3.2 Levantamento com o administrador de sistemas**

Das reuniões com o administrador de sistemas da ARMIS, chegou-se à conclusão de que existiam serviços sob os quais seria crítico avaliar uma migração dos mesmos para a nuvem, bem como o armazenamento que dá suporte às máquinas de produção, que necessitava de ser atualizado urgentemente já que iria perder o suporte até ao final do ano 2022. Os serviços avaliados foram: Microsoft Exchange [10] (serviço de e-mail), Redmine [11] e o site da ARMIS. Outra preocupação demonstrada, foi a forte necessidade de garantir um cenário de recuperação em situações de desastre.

#### **3.3 Levantamento com as Equipas**

Após a realização de reuniões com todas as equipas, de forma a identificar a situação atual de cada equipa relativamente ao uso do centro de dados e das suas necessidades, chegou-se à conclusão de que várias equipas necessitavam de uma atualização das

máquinas do centro de dados (máquinas com mais RAM, armazenamento e CPU) e outras necessitavam de máquinas virtuais criadas em Azure. Dessas máquinas em Azure, destacam-se a da equipa de Engenharia de Software, utilizada para efetuar testes de carga (a razão de ser criada em Azure, é não impactar a rede interna da empresa) e a máquina da equipa Microsoft Enterprise Solutions da ARMIS que servirá para o desenvolvimento com sharepoint e que necessita de capacidade de escalar verticalmente (aumentar os seus recursos de forma automática), de forma a garantir a adaptação às necessidades momentâneas da equipa. Caso a equipa precise de executar mais projetos, a máquina escala verticalmente [12], para assim garantir eficiência de performance com o aumento dos recursos que a mesma possui. Assim que esses recursos deixem de ser utilizados, a máquina reduz os seus recursos, também automaticamente.

### 3.4 Custos do centro de dados

Os custos mensais atuais do centro de dados estão divididos em:

- Licenças de software: 317 €.
- Energia: 350 €.

Os custos de refrigeração foram incluídos nos custos de energia. Em suma, o total de custos mensais (note-se que os custos de manutenção pela equipa IT não são considerados por serem pontuais), rondaria os 667 euros mensais. Após o estudo, verificou-se que a maioria do custo energético está associado a servidores.

### 3.5 Análise de orçamentos de armazenamento

Após a análise dos orçamentos de armazenamentos IBM e da Huawei, a administração decidiu investigar o custo em nuvem do mesmo armazenamento:

**Storage Accounts (Microsoft Azure Estimate) >> West Europe**

Managed Disks, **Standard SSD**, ZRS Redundancy, E60 Disk Type 1 Disks.

Pay as you go **8,192 TB**.

O custo mensalmente seria de 910,82€ enquanto o armazenamento mais equivalente da IBM seria de 16740€. Concluindo, o valor em nuvem ao fim de cerca de dois anos, atingiria o custo de aquisição do equipamento. Embora o Azure oferecesse alta disponibilidade e resistência a falhas, a empresa decidiu manter o armazenamento local.

## 4 Design

Após analisar os documentos gerados pelo Azure Migrate, é possível identificar os tamanhos das máquinas virtuais necessários no Azure, os custos estimados e ainda que as bases de dados devem ser mantidas em máquinas virtuais (para facilitar o controlo das versões e assim controlar melhor as retrocompatibilidades de versões [13]). Chegou-se à conclusão de que o custo final seria dividido em dois tipos:

- Custo fixo mensal (para sempre) - Este custo totaliza 915,39 € e contempla:
  - Azure Site Recovery (4 servidores).

- VPN Gateway.
- App Service.
- Azure Container Instances.
- Máquinas virtuais para testes de carga e para projetos Sharepoint (onde existe maior exigência no que toca a controlo de versões do SQL server e Sharepoint).
- Custo em cenário de recuperação em situação de desastres (para um período de 24h) - Este custo totaliza 262,64€ e contempla 4 máquinas virtuais que apenas serão criadas quando ocorrer a falha no centro de dados local. Foi considerado um período de falha de 24h.

Todos os recursos identificados acima têm a localização West Europe (por ser a mais próxima do centro de dados e para assim reduzir as latências).

### Arquitetura:

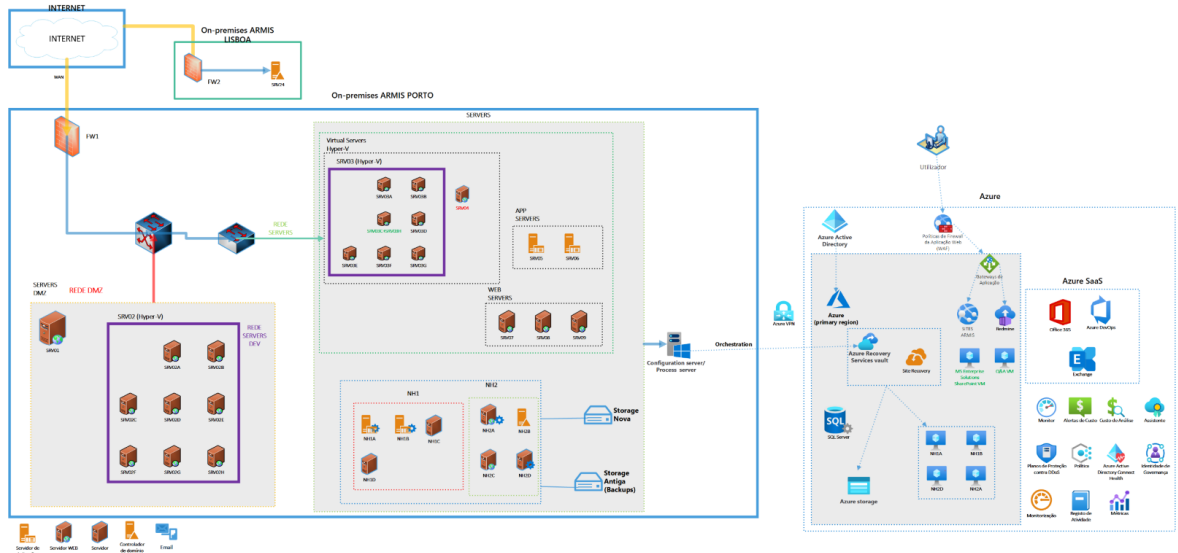


Figura 2 - Arquitetura Proposta

A figura 2 apresenta a nova arquitetura. A aplicação Redmine e o site da ARMIS também foram avaliados e chegou-se à conclusão de que o número de pedidos do Redmine não justificava o uso de Kubernetes. Apenas se usariam Containers. No entanto, a aplicação deveria ser constantemente monitorizada através do Azure Advisor, de forma a avaliar o número de utilizadores, e caso esse número aumentasse drasticamente, podia-se considerar o uso de Kubernetes para tornar a solução mais escalável. O site da ARMIS foi colocado num Azure App Service, pela excelente disponibilidade do serviço PaaS. O armazenamento antigo será utilizado para backups (encontra-se no termo do suporte) e será substituído por um novo equipamento. Quanto aos servidores de bases de dados mais críticos (NH1A, NH2A e NH2D) e o servidor de backups (NH1B), serão mantidos sob uma solução de recuperação de desastres (Azure Site Recovery) que garantirá através de cópias de

armazenamento e da imagem dos servidores locais que, quando a infraestrutura local falhe, os servidores em questão, serão criados no Azure, com a imagem e o armazenamento em cópia de segurança.

Atendendo aos pilares do WAF:

- **Pilar de segurança:** Através do **Azure Active Directory**, dos serviços de gestão de identidades, das políticas de segurança e proteções contra *DDoS*, juntamente com a firewall robusta, será garantida uma maior segurança da infraestrutura. A equipa de Ciber Segurança da Ozono deverá definir os procedimentos de segurança em situações críticas e em situações normais do dia a dia dos colaboradores. É crucial continuar a usar o MS Defender. Sugere-se ainda uma implementação do MFA (Multi-Factor Authentication), de forma a garantir que os servidores têm uma camada bastante firme no que toca a *log in* e acessos aos mesmos.
- **Pilar da confiabilidade:** Todo o design assenta em recuperação de desastres dos servidores (NH1A, NH2A, NH1B e NH2D) e aplicações e serviços mais críticos da ARMIS. Através do Azure Site Recovery, é garantido o funcionamento da infraestrutura do lado do Azure. O mecanismo serve-se de cópias da imagem e do armazenamento da máquina virtual (VM) ou do servidor local (que são efetuados para o Azure, através de mecanismos seguros) para criar a máquina virtual no Azure aquando da falha local. Quando se detetar que as máquinas em questão no local estão em baixo, o Azure Site Recovery inicia a criação de uma VM com aquela imagem e armazenamento e ativa os serviços da mesma.
- **Pilar da eficiência de performance:** A aplicação Redmine pode começar a receber mais pedidos, mas nada que para já, justifique o uso de Kubernetes. Em engenharia da nuvem é necessário ter sempre em conta que não se deve usar ferramentas excessivas para coisas simples. O comportamento da aplicação deve ser monitorizado e em caso de necessidade, devem-se aplicar os Kubernetes para permitir a orquestração e dimensionamento automático da solução. Razão pela qual na arquitetura apenas aparecem containers.
- **Pilar da otimização de custos:** Após a implementação, é necessário monitorizar os custos, através do Azure Advisor e do Azure Cost Management, de forma a garantir que apenas se gasta o necessário e não são excedidos limites.
- **Pilar da excelência operacional:** A utilização do **DevOps** em quase todos os projetos, é a forma mais correta de garantir também a excelência operacional, já que a maioria das tarefas ficam automatizadas. A ideia é levar a que as equipas deixem de desperdiçar tempo com tarefas manuais e se dediquem a 100% apenas aos processos de engenharia e arquitetura da nova solução, ao desenvolvimento do código e testes e operação (monitorização e suporte, em caso de problemas) das soluções como sugere a Figura 3.

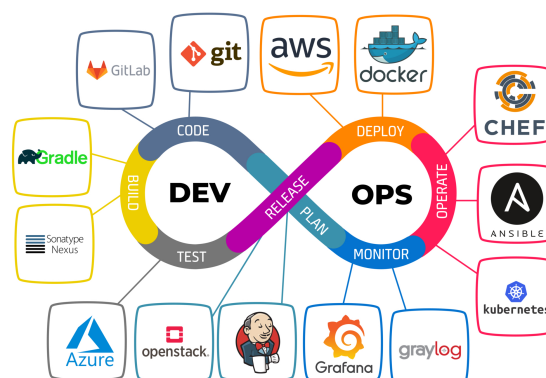


Figura 3- DevOps e algumas das ferramentas a utilizar por fase de projeto [14]

## 5 Conclusões e Trabalho Futuro

Com a análise de todos os requisitos, foi possível concluir que, a arquitetura mais benéfica para a ARMIS seria uma híbrida (não é possível migrar tudo para nuvem, em algumas situações por questões de compatibilidade e suporte). O WAF é utilizado, não só quando se efetua um design, mas também aquando da implementação no Azure. O mesmo disponibiliza *avaliações* para verificar numa escala percentual o quanto são cumpridos os pilares do mesmo no centro de dados (*local* ou nuvem).

O Azure Site Recovery é uma ferramenta essencial para garantir a continuidade do negócio em situações de falha. Assim, consegue-se garantir que, após uma falha, rapidamente se aloca automaticamente uma VM com a *imagem* e armazenamento da original. Com isto, é possível reduzir a demora no processo de recuperação de desastres e tornar o processo mais eficiente e automatizado (pilares da confiabilidade e excelência operacional).

O Azure Migrate é uma ferramenta bastante intuitiva e eficiente que consegue automatizar um levantamento (máquinas virtuais, servidores, SQL Server, inventário de software e ainda análise de dependências de cada servidor), efetuar avaliações que servem para guiar a equipa na migração (recomenda os tamanhos das máquinas, das bases de dados e do armazenamento; avalia qual a melhor forma de migrar as bases de dados, se para máquinas virtuais, como foi o caso, ou mantê-las em PaaS) e ainda automatizar o processo de migração para o Azure. É essencial que as empresas mantenham documentações atualizadas sobre os *centros de dados* e, de facto, o Azure Migrate é a solução ideal para que isso aconteça. Cada vez mais, o mundo é baseado em nuvem. É a tecnologia do presente e será sem dúvida uma das mais notórias no futuro.

## Referências

- [1] «Cloud computing: o que é, para que serve e como usá-la?», *Rock Content - BR*, 6 de agosto de 2018. <https://rockcontent.com/br/blog/cloud-computing/> (acedido 21 de setembro de 2022).

- [2] «<http://d086c02129004450a2b147b55be5f6ce.pages.ubembed.com/90a6add6-32e5-4cb3-ae85-96d8a724b1d7/>».  
<https://d086c02129004450a2b147b55be5f6ce.pages.ubembed.com/90a6add6-32e5-4cb3-ae85-96d8a724b1d7/g.html?closedAt=0> (acedido 30 de novembro de 2022).
- [3] Christianlauer, «The Azure Well-Architected Framework», *CodeX*, 30 de novembro de 2021. <https://medium.com/codex/the-azure-well-architected-framework-7ebc0e73ae57> (acedido 6 de abril de 2022).
- [4] david-stanford, «Microsoft Azure Well-Architected Framework - Azure Architecture Center». <https://docs.microsoft.com/en-us/azure/architecture/framework/> (acedido 5 de setembro de 2022).
- [5] robbyatmicrosoft, «Security - Training». <https://docs.microsoft.com/en-us/training/modules/azure-well-architected-introduction/7-security> (acedido 12 de setembro de 2022).
- [6] robbyatmicrosoft, «Performance efficiency - Training». <https://docs.microsoft.com/en-us/training/modules/azure-well-architected-introduction/5-performance-efficiency> (acedido 12 de setembro de 2022).
- [7] «Azure Site Recovery | Microsoft Azure». <https://azure.microsoft.com/pt-pt/products/site-recovery/> (acedido 21 de setembro de 2022).
- [8] «Recuperação – Wikipédia, a enciclopédia livre». <https://pt.wikipedia.org/wiki/Recupera%C3%A7%C3%A3o> (acedido 8 de outubro de 2022).
- [9] F. Ferreira, «Calculadora de TCO do Azure», *Microsoft Azure Experts*, 13 de novembro de 2019. <https://www.azureexperts.com.br/calculadora-de-tco-do-azure/> (acedido 29 de julho de 2022).
- [10] wwlpublish, «Explore Exchange Online as a Microsoft 365 Enterprise Administrator - Training». <https://learn.microsoft.com/en-us/training/modules/explore-exchange-online/> (acedido 8 de outubro de 2022).
- [11] «Overview - Redmine». <https://www.redmine.org/> (acedido 8 de outubro de 2022).
- [12] wwlpublish, «Descreva benefícios e considerações na nuvem - Learn». <https://docs.microsoft.com/pt-pt/learn/modules/fundamental-azure-concepts/benefits-of-cloud-computing> (acedido 23 de março de 2022).
- [13] MashaMSFT, «Compatibilidade com versões anteriores de replicação - SQL Server». <https://learn.microsoft.com/pt-br/sql/relational-databases/replication/replication-backward-compatibility> (acedido 8 de outubro de 2022).
- [14] I. e P. do R. A. de não ter suas origens na área de RH, aplicou-para desenvolver o R. 20 anos de experiência na contratação de profissionais e na montagem e gerência de equipes técnicas A. como S.-D. da empresa 4linux F. fundador e presidente do LPIBrasil, a maior certificação profissional linux do mundo P. experiência em gerência técnica, comercial e financeira G. em E. E. pela U. de S. P. com especialização em Computação, e fez E. em A. de E. na F. G. V. M. diversas palestras em eventos de informática, «Qual o nível de maturidade DevOps da sua empresa?», *Blog 4Linux*, 27 de junho de 2019. <https://blog.4linux.com.br/qual-o-nivel-de-maturidade-devops-da-sua-empresa/> (acedido 30 de novembro de 2022).

# Computational Simulation Approaches to Low Pass Filtering

Fernando Carvalho<sup>1</sup>, Vítor Cardoso<sup>1</sup>, Pedro Reis<sup>1,2</sup>, Sandra Aires<sup>1,3</sup>,  
António Sousa<sup>1,3</sup> e Pedro Guedes<sup>1,2</sup>

<sup>1</sup> ISEP – IPP, Porto, fjc@isep.ipp.pt

<sup>2</sup> CRAS, INESC TEC, Porto

<sup>3</sup> LEMA, ISEP-IPP, Porto

**Abstract.** Scientific knowledge in mathematics and physics is very useful to engineers. Yet, sometimes, when teaching those subjects to engineering students in the first years of college, passing this message to them is not trivial. The work described in this paper was made in the ambit of a didactic experience carried on with the author's students at Instituto Superior de Engenharia do Porto (ISEP) in the electrical engineering course. It consisted in the modelling and simulation of a RLC circuit working as a 2nd order low pass filter applied to a square wave input signal. Kirchoff's 2nd law is used to build a mathematical model for this circuit consisting on a system of two first order linear ordinary differential equations (ODEs). First the input signal is approximated using Fourier series; then the circuit is tuned to have the same frequency response as the 2nd order Butterworth filter and finally the simulation is made and the output signal is calculated by numerically solving the ODEs system with the SCILAB software, either by calling its "ode" function or by using the Euler's integrative method that the authors implemented from scratch using traditional programming, or finally by using XCOS – a tool included in SCILAB that allows pictographic programming. This work was successful in making the bridge between mathematics and engineering.

**Keywords:** ODE; Euler's method; Fourier series; RLC filter; SCILAB; XCOS.

## 1 Introduction

Many physical phenomena associated, for instance, to mechanical or electrical systems, where the variables involved vary over time and are mutually dependent, can be described, based on [1], by a system of linear differential equations of constant coefficients:

$$\sum_{i=1}^m \sum_{k=0}^n a_{l,i,k} \frac{d^k y_i}{dt^k} = f_l(t), \quad l = 1, \dots, b \quad (1)$$

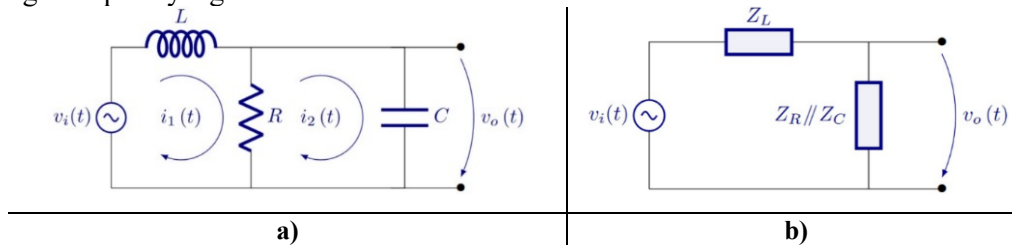
In this system of  $b$  linear differential equations:  $n$  and  $m$  represent, respectively, the order and the number of the ODEs system dependent variables;  $a_{l,i,k}$  are constant coefficients;  $f_l(t)$  with  $l \in \{1, \dots, b\}$  are continuous functions defined on some interval  $I \subset \mathfrak{R}$ ; and,  $y_i(t)$  are the dependent variables.

If the initial conditions are unknown, general expressions involving arbitrary constants can be obtained for each system dependent variable  $y_i(t)$ . Those expressions represent an infinite number of solutions belonging to the same family. Otherwise, if the initial conditions (the values for  $y_i(t_0)$  and  $\frac{d^k y_i}{dt^k}(t_0)$  where  $t_0$  is the initial instant,  $i \in \{1, \dots, m\}$  and  $k \in \{1, \dots, n-1\}$  are known we have an initial value problem. Typically, both cases can be analytically solved using Laplace transform as a linear equations system, but this approach is outside of our scope. For initial value problems, the integrative numerical methods as Euler's method always work.

If  $f_l(t) = 0$  with  $l \in \{1, \dots, b\}$ , the system is homogeneous, otherwise it is nonhomogeneous. The solutions of the nonhomogeneous system  $y_i(t)$  are composed of the homogeneous solutions  $y_{i,h}(t)$  and the particular integrals  $y_{i,p}(t)$ . This is described by:

$$y_i(t) = y_{i,h}(t) + y_{i,p}(t) \quad (2)$$

For each system dependent variable, on one hand  $y_{i,h}(t)$  represents the transient or response in the unforced state, on the other hand  $y_{i,p}(t)$  represents the system response in the steady or forced state [2]. This is the point from which we proceed to link mathematics and electrical engineering, in the Mathematics Laboratories curricular unit of the Electrotechnical Engineering Course at Instituto Superior de Engenharia do Porto, aiming at achieving a meaningful learning for our students. Thus, as an application, this work presents the RLC electrical circuit whose scheme is represented in the Fig. 1. This circuit implements a 2nd order Butterworth low-pass filter [3-5]. By definition, a low-pass filter allows low-frequency signals to pass, blocking or hindering the passage of high-frequency signals.



**Fig. 1.** a) 2nd order RLC filter; b) equivalent voltage divider circuit based on impedances.

The system of differential equations that models the RLC electrical circuit is obtained based on the 2nd Kirchoff's law.

It has two first-order differential equations, resulting from two loops, as follows:

$$\left\{ \begin{array}{l} v_i(t) = L \frac{di_1}{dt} + R(i_1(t) - i_2(t)) \\ 0 = R(i_2(t) - i_1(t)) + v_0(t) \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} v_i(t) = L \frac{di_1}{dt} + v_0(t) \\ 0 = R \left( C \frac{dv_0}{dt} - i_1(t) \right) + v_0(t) \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \frac{di_1}{dt} = \frac{v_i(t) - v_0(t)}{L} \\ \frac{dv_0}{dt} = \frac{i_1(t)}{C} - \frac{v_0(t)}{RC} \end{array} \right. \quad (3)$$

,  $i_1(t_0) = i_0, v_0(t_0) = v_0$

In this work, SCILAB is used as support platform to solve the system and represent the filtered signal  $v_o(t)$ . The input signal corresponds to the square wave represented by the Fourier approximation  $v_i(t)$ . Values for  $R, L$  and  $C$  of the circuit are chosen so that the filter has the same frequency response as the Butterworth filter [4].

To solve the system and obtain the filtered signal  $v_o(t)$ , two different approaches are presented: the use of numerical integrative methods by calling SCILAB “ode” function or by implementing the Euler’s method, and the use of pictographic programming based on the XCOS Block diagram tool [2].

This paper presents, in section 2, information about how the RLC circuit filter tune is made, how the Euler’s integrative method works, as well as about SCILAB and XCOS; in sections 3 and 4, respectively some implementation details and results. Finally, in the last section, some conclusions are presented, about the simulation results and about the merits of this didactic experience involving computational simulations.

## 2 Methodologies and Programming Tool

### 2.1 RLC filter tune using the Butterworth filter features

Given the impedances  $Z_C, Z_L$  e  $Z_R$ , respectively, relative to the capacitor, the coil and the resistance, the transfer function, determined from the voltage divider of Fig. 1b), is:

$$H_{RLC}(j\omega) = \frac{Z_R // Z_C}{Z_L + Z_R // Z_C}. \quad (4)$$

So, considering that  $Z_R = R, Z_L = j\omega L$  e  $Z_C = 1 / j\omega C$ , results:

$$H_{RLC}(j\omega) = -\frac{R}{L\omega(CR\omega - j) - R}. \quad (5)$$

The respective modulus and phase are given by:

$$MH_{RLC}(\omega) = \frac{R}{\sqrt{\omega^4 (RLC)^2 - 2R^2LC\omega^2 + L^2\omega^2 + R^2}} \quad (6)$$

$$FH_{RLC}(\omega) = \arctan\left(\frac{L\omega}{RLC\omega^2 - R}\right). \quad (7)$$

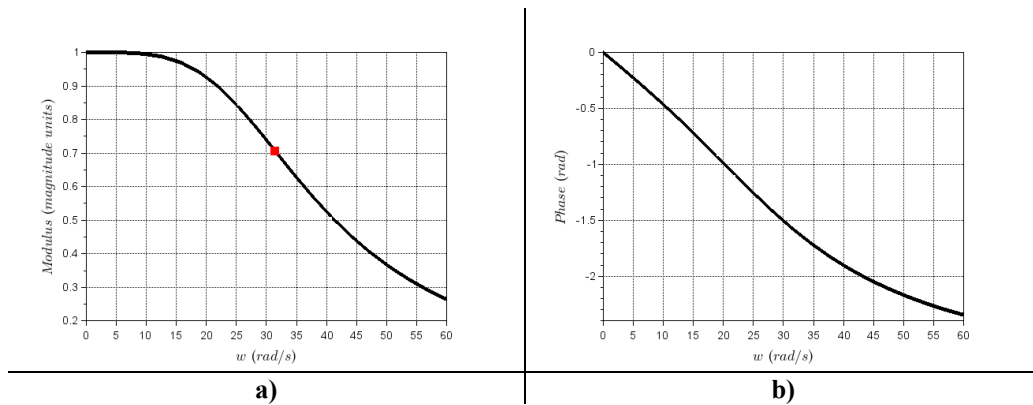
The parameters  $R$ ,  $L$  and  $C$  of the 2nd order low pass RLC filter represented in Fig. 1a) must be chosen so that the filter has the same frequency response as the Butterworth filter. For this purpose:

- 1) The RLC filter resonance angular frequency,  $\omega = 1/\sqrt{LC}$ , must take the value of the cut-off frequency  $\omega_c$ ;
- 2) The RLC filter quality factor  $Q = MH_{RLC}(\omega_c) = R/(L\omega_c)$ , must take the value of the normalized Butterworth filter quality factor, that is equal to  $H_0/\sqrt{2}$  [3-5] - this reference tune point is represented in Fig. 2a).

In short, the calibration is carried out based on the following system of equations:

$$\begin{cases} \omega_c = \frac{1}{\sqrt{LC}} \\ Q = \frac{R}{L\omega_c} \end{cases} \quad (8)$$

In a purely academic exercise, if one considers the sign  $v_i(t)$  in Fig. 4a), assuming  $\omega_c = 10\pi \text{ rad/s}$  (located at the harmonic of order 10),  $Q = 1/\sqrt{2}$ ,  $C = 1F$ , we get  $R = \sqrt{2}/20\pi \Omega$  and  $L = 1/100\pi^2 H$ . With these values for the electric components there is a perfect match between the continuous modulus and phase spectra for the RLC and the Butterworth filters, that have the continuous spectra represented in the Fig. 2.



**Fig. 2.** Images of the RLC filter continuous modulus spectrum  $MH_{rlc}(\omega)$ , with the Butterworth filter calibration characteristic point in a) and respective continuous phase spectrum  $FH_{rlc}(\omega)$  in b), after the filter tune.

## 2.2 Euler's Integrative method

The Euler's integrative method results from the first order Taylor approximation around a known point. It can be used to solve any initial value problem that includes an ODEs system where each dependent variable first order derivative is a function of the system variables [2].

The RLC filter can be simulated by solving such an initial value problem – see the ODEs system in (3). Using the Euler’s method to implement the simulation consists in working with the iterative system of equations in (9) to calculate its solution, where  $h$  is the integration step and  $p$  the number of iterations, determined by  $p = (b - a) / h$  considering  $a$  and  $b$  the lower and upper time integration limits, respectively.

$$\begin{cases} t_j = t_{j-1} + h, i_{1j=0} = i_0, v_{oj=0} = v_o \\ i_{1j} = i_{1j-1} + \frac{v_i(t_{j-1}) - v_o(t_{j-1})}{L} h \\ v_{oj} = v_{oj-1} + \left( \frac{i_1(t_{j-1})}{C} - \frac{v_o(t_{j-1})}{RC} \right) h \end{cases}, j = 1, 2, \dots, p \quad (9)$$

### 2.3 SCILAB and XCOS

SCILAB is an Open source software for numerical computation – in many aspects it is a MATLAB clone. Being also a development platform, it provides a console and an editor for traditional programming and a tool for pictographic programming – XCOS – very similar to MATLAB SIMULINK. With the help of SCILAB and XCOS dynamical systems simulations can be implemented very fast and effortless.

## 3 Implementation details

Code 1a) shows the code of function “vi” for generating the square waveform. A standard period of the signal is defined, starting at  $-T/2$ , and repeating from the function “pd”. The function “f” in Code 1b) represents the Fourier series of the square wave, truncated to the NN first terms.

<pre>function y1=vi(t1) y1=[]; deff('y=pd(t)', 'y=t-     T*floor((t-shift)/T)') for i=1:length(t1) t = pd(t1(i)); if t&lt;-1 y=0 elseif t&gt;=-1 &amp; t&lt;0 y=0; elseif t&gt;=0 &amp; t&lt;1 y=1; else y=0 end y1=[y1,y]; end endfunction</pre>	<pre>function SFN=f(t) if t&lt;0 SFN=0; elseif t&gt;=0 harmonicos1 = feval(t,0:NN, harmonico1); SFN = sum(harmonicos1,'c'); SFN = SFN'; end endfunction</pre>
a)	b)

**Code 1.** Scilab code extracts concerning the implementation: of the original signal in a), Fourier series approximation in b).

The Fourier series general term is obtained from the Code 2 by determining the coefficients  $a_0$ ,  $a_n$  e  $b_n$ .

```
//Definition of the Fourier Series Coefficients
a0=2/T*(integrate('0','t',-1,0)+integrate('1','t',0,1));
an=[];
for n=1:NN
    an=[an; clean(2/T*(integrate('0*cos(n*w0*t)','t',-1,0)
+integrate('1*cos(n*w0*t)','t',0,1)))]; end
bn=[];
for n=1:NN
    bn=[bn; clean(2/T*(integrate('0*sin(n*w0*t)','t',-1,0)
+integrate('1*sin(n*w0*t)','t',0,1)))]; end
deff('y=harmonic1(t,n)','if(n==0) y=a0/2; else
    y=an(n)*cos(n*w0*t)+bn(n)*sin(n*w0*t); end');
```

**Code 2.** Scilab code extracts concerning to the determination of the Fourier series general term.

### 3.1 Problem resolution using numerical methods

Code 3 shows the code for obtaining the filtered signal  $v_o(t)$  using the “ode” function of SCILAB, where the system is represented in the matrix form by the function “ODE\_system”.

```
//Solution of the system of equations based on Scilab's ODE function
//Variables initialisation:
t0=0; tf=2.4; step=0.001; t1=t0:step:tf;
//Initial conditions:
t0=0; ili=0; voi=0.75; Yinicial=[ili;voi];
//Definition of the system of differential equations
deff('dY=ODE_system(t,Y)','dY(1)=(f(t)-Y(2))/L;dY(2)=Y(1)/C-Y(2)/(R*C)');
Y1=ode(Yinicial,t0,t1,ODE_system);
il=Y1(1,:) //il(t);
vo=Y1(2,:) //vo(t);
```

**Code 3.** Filter the signal using a built-in SCILAB's “ode” function.

In Code 4, the code that allows obtaining the filtered signal  $v_o(t)$  from the Euler’s integrative method, is presented.

```
//*****Euler's method*****
//Variables initialisation:
t0=0; tf=2.4; step=0.001;
t1=t0:step:tf;
il_=zeros(1,length(t1)); vc_=zeros(1,length(t1));
y0=feval(t1,vi); SF=feval(t1,f);
//Initial conditions:
il_(1)=0; vc_(1)=0.7;
for i=2:length(t1)
    il_(i) = il_(i-1) + ((SF(i-1)-vc_(i-1))/L)*step;
    vc_(i) = vc_(i-1) + (il_(i-1)/C - vc_(i-1)/(R*C))*step;
end
```

**Code 4.** Obtaining the filtered signal  $v_o(t)$  by implementing the Euler’s integrative method.

### 3.2 Problem resolution using the pictogram of the XCOS didactic tool

In Fig. 3, the ODEs system implementation pictogram to obtain the filtered signal  $v_o(t)$  is presented.

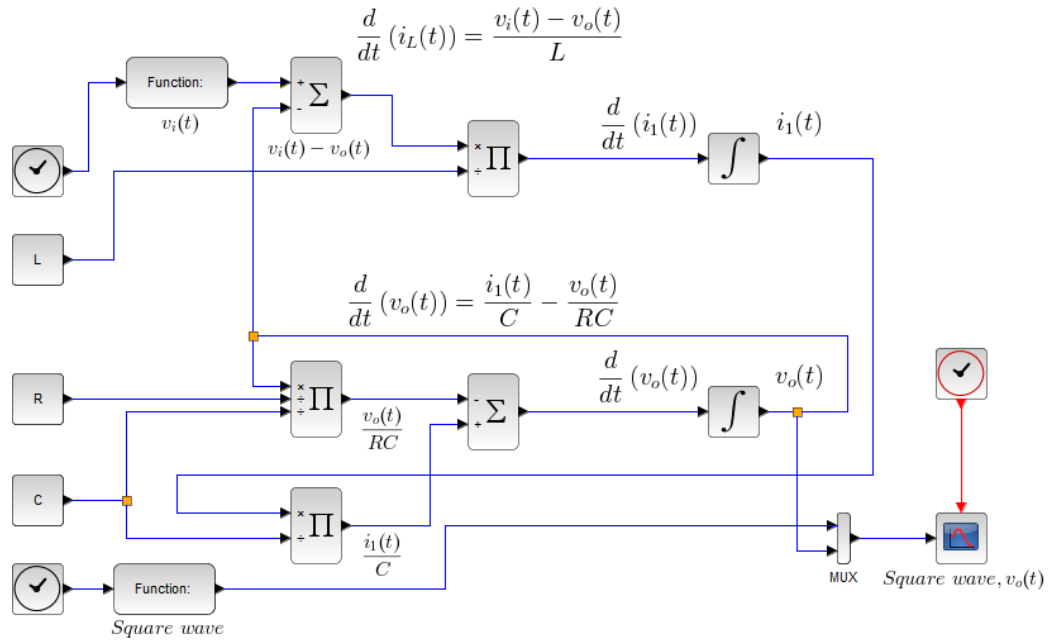
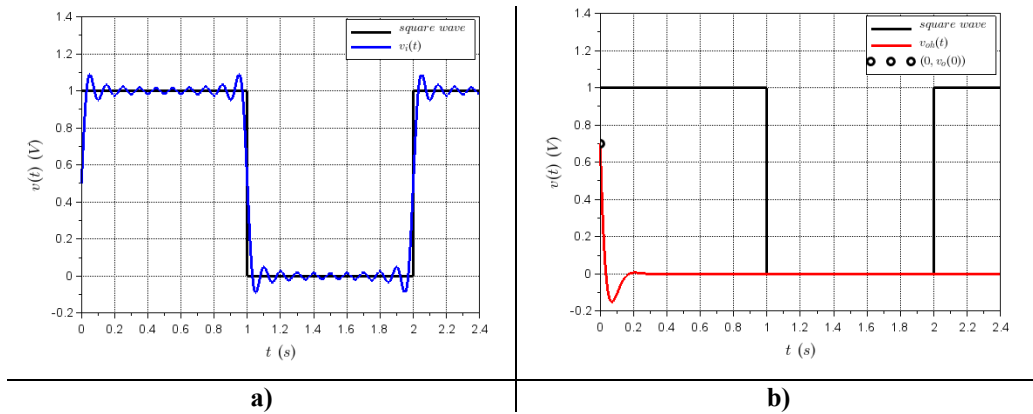
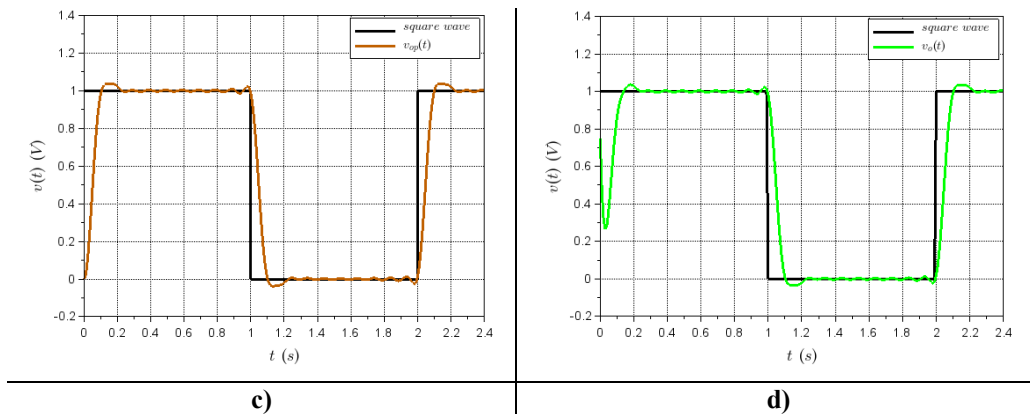


Fig. 3. ODEs system implementation pictogram of 2nd order RLC filter.

## 4 Results

Fig. 4a) presents the 20<sup>th</sup> order Fourier approximation of the square wave signal,  $v_i(t)$  and Fig. 4d), the corresponding filtered signal  $v_o(t)$ . Then  $v_o(t)$  is equal to the sum of its transient component, presented in Fig. 4b) which also shows the initial condition, with its steady state component, presented in Fig. 4c).





**Fig. 4.** – Image with: the original signal  $v_i(t)$  in a), the transient component  $v_{oh}(t)$  in b), the steady state component  $v_{op}(t)$  in c) and the filtered signal obtained from any of the previous approaches (SCILAB “ode” function, Euler’s Method or Pictogram)  $v_o(t)$  in d).

## 5 Conclusions

This work presents a computational simulation approach to low-pass filtering. It was made in the ambit of a didactic experience carried on with the author’s students at ISEP, in the Electrical Engineering course. This didactic experience had the merit of showing the usefulness of mathematics and physics in engineering – students had the opportunity to study and apply theoretical models of an RLC electric circuit, as well as mathematics subjects such as Fourier series, ODEs and numerical integrative methods. It also had the merit of using computers and helping students to develop their programming skills while using SCILAB – a powerful Open source software for numerical computation. Simulation results showed them that a RLC circuit can be calibrated to exhibit exactly the same frequency response as the theoretical Butterworth filter and that using such filters introduce nonlinear phase shifts in the input signal harmonics which cause delay and distortion to be present in the filtered signal. This output signal is the sum a transient component and a steady-state component.

## References

1. Blume, F.: Applied Calculus for Scientists and Engineers: A Journey in Dialogues. 1st edn. Jones and Bartlett Publishers (2005).
2. Castro, A., Viamonte, A. and Sousa, A.: Cálculo II, ENGEBOOK. 2th edn. Publindústria (2016).
3. Parkers, T. W. and Burrus, C. S.: Digital Filter Design. John Wiley & Sons (1987).
4. Butterworth, S.: On the Theory of Filter Amplifiers in Wireless Engineer, vol. 7. (1930). pp. 536-541.
5. Cardoso, A.: Síntese, análise e realização de filtros passivos, Master’s thesis (1995). Available at <https://repositorio-aberto.up.pt>.

## Computerized Physician Order Entry orientado à monitorização de sinais vitais

Diogo Soares<sup>1,2</sup>, Ricardo Almeida<sup>2</sup>, Nuno Escudeiro<sup>2</sup>, André Mota<sup>1</sup>, Rita Fontes<sup>1</sup>,  
Laura Saldanha<sup>1</sup>, Catarina Caçote<sup>1</sup>

<sup>1</sup> Glintt – Healthcare Solutions, S.a., R. Eng. Ferreira Dias 728, Porto, Portugal

<sup>2</sup> Instituto Superior de Engenharia do Porto, R. Dr. António Bernardino de Almeida 431, 4249-015 Porto, Portugal

**Resumo.** Este artigo explicita o processo de conceptualização e desenvolvimento de uma versão Minimum Viable Product (MVP) de um Computerized Physician Order Entry (CPOE), a ser incluído na aplicação hospitalar Viewer, da Glintt-HS. Este Sistema representa uma solução que agrega diferentes fontes de informação, centralizando pedidos médicos de diferentes áreas. Dado que não seria exequível abranger várias áreas hospitalares num curto / médio prazo, este CPOE foca-se na área de monitorização de sinais vitais, impulsionada pelo projeto Wireless biomonitoring stickers and smart bed architecture: towards untethered patients. Sinteticamente, o projeto envolve a utilização de dispositivos aderentes à pele e sem fios para monitorizar sinais vitais. Assim, este artigo responde à hipótese de conceptualização de um CPOE que permita a gestão transversal para a área de monitorização de sinais vitais. Esta hipótese oferece uma abordagem diferenciadora ao adicionar a área de monitorização de sinais vitais a um CPOE, que normalmente inclui apenas as áreas de imagiologia, laboratório e farmácia. Para resolver este problema foi selecionada a metodologia Design Science Research Methodology (DSRM), de forma a obter uma solução capaz de associar dispositivos a pacientes, criar pedidos de monitorização com recorrência temporal e consultar resultados de monitorizações. Assim, é feito o levantamento do estado da arte para ampliar o conhecimento da área de negócio e conhecer o que o mercado oferece. São ainda especificados os requisitos e o processo de concepção da solução é documentado. Por fim, o desenvolvimento é evidenciado e avaliado. No final, a solução correspondeu às expectativas, cumprindo a hipótese, objetivos e requisitos definidos.

**Palavras-Chave:** Computerized Physician Order Entry, Sinais Vitais, Monitorização

### 1 Introdução

Os vários pedidos, de diferentes áreas médicas, para o tratamento de doentes estão muitas vezes dispersos pelas aplicações com que os profissionais de saúde trabalham. O tempo do utilizador não é otimizado devido a esta experiência múltipla a que é sujeito.

De forma a agilizar este trabalho, é importante mitigar o efeito desta dispersão. Assim, nasce a oportunidade de criar um sistema que permita aos profissionais de saúde ter uma experiência agregadora, ágil e unitária no plano de tratamento do doente.

Um CPOE permite a agregação de pedidos de prescrição de exames, laboratório e imagiologia numa única experiência, com uma visão centralizada. No entanto, identificou-se a necessidade de incluir mais áreas funcionais nesta experiência de tratamento do doente. Dado que o comum CPOE prevê apenas as áreas supracitadas, é necessária a conceptualização e implementação de um CPOE com responsabilidades em mais áreas funcionais, como Cardiologia, Dermatologia, Ortopedia ou qualquer outra área. Este artigo documenta este processo, através de um Minimum Viable Product (MVP) no contexto do projeto Wireless biOMonitoring stickers and smart bed architecture: toWards untethered patients (WoW) – que está a desenvolver uma iniciativa vanguardista nas áreas de saúde digital e hospitalização domiciliária, usando dispositivos sem fios e aderentes à pele, com funções de medição de variados sinais vitais. Desta forma, a solução deverá facilitar ao profissional de saúde o processo de leitura e de requisição de pedidos no âmbito das necessidades da área de Sinais Vitais.

Para dar a melhor resposta possível ao problema, decidiu-se adotar a metodologia Design Science Research Methodology (DSRM) [1]. A DSRM cria e avalia artefactos orientados para resolver problemas organizacionais. Esta metodologia deve ser baseada em teorias e conhecimento existentes, produzindo uma solução útil, de qualidade e avaliada com rigor, para um problema e área específica [1].

Identifica-se como objetivo principal a conceptualização e desenvolvimento da solução CPOE (MVP) para a área de Monitorização Sinais Vitais. Os contributos esperados incluem o acréscimo de valor à aplicação hospitalar Viewer, que irá trabalhar com a nova solução, assim como valor acrescentado que se gera para a comunidade científica através da conceptualização de um sistema CPOE diferenciado, que acrescenta responsabilidades de áreas extraordinárias a este sistema.

## 2 Estado da Arte

### 2.1 Computerized Physician Order Entry

O Computerized Physician Order Entry (CPOE) é um sistema computacional que permite a um profissional de saúde autorizado fazer pedidos médicos [2].

Os erros médicos nos hospitais são comuns e podem ser uma ameaça para a segurança e saúde dos pacientes. Na fase de prescrição é particularmente comum acontecerem, frequentemente devido à documentação incorreta das ordens de medicação pretendidas. Assim, os sistemas CPOE são propostos como um elemento importante para a melhoria da segurança [3].

Explicando o seu funcionamento, os pedidos são transferidos através de uma rede computacional para os profissionais de saúde, incluindo enfermeiros, terapeutas e médicos, e também para imagiologia, farmácia e laboratório. Este é um sistema que converte a forma de prescrição manual ou com recurso a papel, para uma operação eletrónica, ajudando a reduzir custos [4].

Uma solução CPOE pode ainda funcionar integrada numa infraestrutura tecnológica mais alargada, estendendo habitualmente a funcionalidade Eletronic Health Record (EHR) [5].

A Fig. 1 ilustra a integração e workflow de um sistema CPOE com um EHR, fazendo uso dos dados dos pacientes. Esta ergonomia possibilita e agiliza o processo de prescrição para diferentes áreas clínicas por parte do CPOE, nomeadamente as áreas de Farmácia, de Laboratório e de Imagiologia.

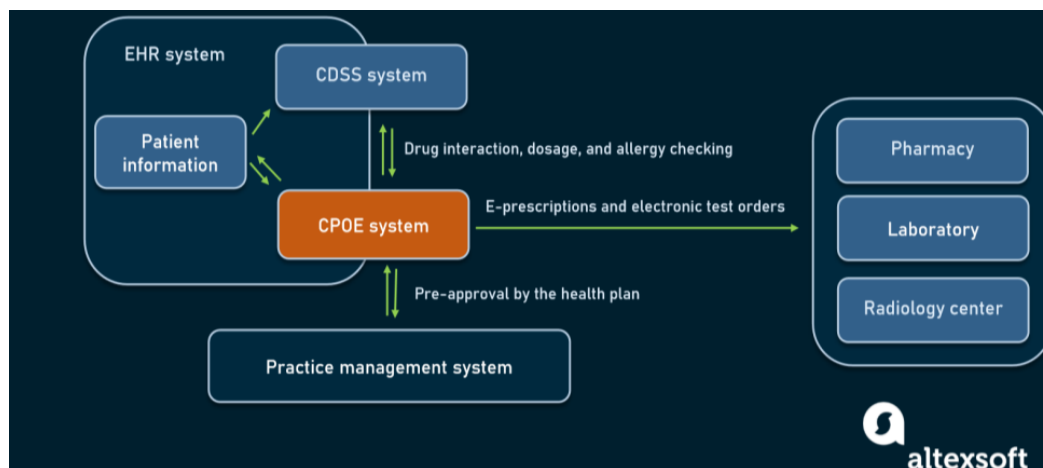


Fig. 1. Sistema CPOE integrado numa infraestrutura de software hospitalar [5]

Os sistemas CPOE têm vindo a ser adotados à escala global [6], promovendo a segurança dos pacientes e a melhoria da qualidade e modernização da prática da saúde. No entanto, estes sistemas afetam de uma forma complexa a prestação de cuidados de saúde, acarretando benefícios, mas também riscos e desvantagens [7].

Relativamente a benefícios, o CPOE pode ser um mecanismo efetivo para melhorar a segurança dos pacientes através da redução de erros [2] que contribuem para a prevenção global de eventos adversos relacionados com medicação [8].

Este sistema está também associado a uma maior rapidez de envio de ordens para terceiros, como por exemplo a farmácia [9] e à automatização do processo de pedidos médicos, que resulta em pedidos mais legíveis, completos e normalizados [10].

Abordando as desvantagens, verifica-se a dependência na tecnologia [11], situações relativas a aumento de carga de trabalho para os funcionários médicos que usam o sistema [11, 12] e o aparecimento de novos erros de prescrição de medicação, relacionados com erros do software e falta de conhecimento do sistema por parte dos profissionais de saúde [12], que se justificam por eventuais deficiências da interface, ecrãs que induzem o utilizador em erro, workflows incorretos e má utilização do sistema [3].

## 2.2 Âmbito CPOE

**Sinais Vitais.** Os sinais vitais [13] são uma medida objetiva das funções fisiológicas essenciais de um organismo vivo. Tradicionalmente, os sinais vitais identificados são a temperatura, frequência cardíaca, pressão arterial e frequência respiratória. Definem-se

como “vitais” pelo facto da sua medição e avaliação representarem o primeiro passo crítico para avaliações clínicas. São por isso alvo de avaliação nos processos de triagem de tratamentos e urgências hospitalares, refletindo e definindo a prioridade dos pacientes no acesso aos cuidados médicos.

A avaliação dos sinais vitais de um paciente pode ainda prever futuros resultados clínicos, assim como eventuais regressos às urgências e a frequência de readmissão nos hospitais [13].

O aumento de perturbações crónicas, como a hipertensão, condições respiratórias e diabetes, assim como o surgimento da pandemia da covid-19, têm contribuído para um aumento e valorização do mercado de dispositivos de monitorização de sinais vitais. Este crescimento deve-se também à incorporação do uso destes dispositivos de biomonitorização fora de contexto hospitalar, como em atletas, para a melhoria do seu desempenho desportivo [14, 15], em diabéticos para a monitorização de glucose e administração de insulina [16] e em utilizadores de próteses para o seu controlo [17].

**Projeto Wireless biomonitoring stickers and smart bed architecture: towards untethered patients.** No âmbito funcional de Monitorização de Sinais Vitais foi escolhido o projeto de inovação Wireless biOmonitoring stickers and smart bed architecture: toWards untethered patients (WoW), para a fase inicial de desenvolvimento do CPOE.

A solução proposta pelo projeto WoW permite que sejam dadas altas hospitalares antecipadas a certos pacientes, sem que seja comprometida a sua saúde, através da monitorização pós hospitalar e sem fios de sinais vitais. Para isto, o WoW procura construir um sistema baseado em dispositivos de biomonitorização sem fios que se fixam à pele, recolhendo diferentes sinais vitais. Estes dispositivos dividem-se entre sensores e smartboxes. Os sensores realizam as monitorizações e são adesivos com aderência à epiderme, usam energia wireless e são capacitados para trabalhar com transferência de dados. Cada sensor pode associar-se a uma e só uma smartbox. As smartboxes são dispositivos que podem ser ligados às camas dos pacientes, em casa ou no hospital, recebendo dados de monitorização dos sensores adesivos.

Desta forma, é necessária a implementação de um software, permitindo as monitorizações, através da recolha, processamento e transmissão centralizada de dados.

### 2.3 Trabalhos Relacionados

De forma a analisar o estado da arte foram estudados alguns trabalhos relacionados com CPOE, como o software PatientKeeper CPOE [18], o Practice Fusion [19] e o Power Orders [20]. Entre estas soluções, o Patientkeeper CPOE é a única que se apresenta como uma solução independente, enquanto as outras são sistemas com responsabilidades não exclusivas de um sistema CPOE ou dependentes de sistemas de terceiros. Ou seja, apresentam características indesejadas neste contexto. O Patientkeeper CPOE apresenta assim compatibilidade com sistemas externos, no entanto, tal como todas as outras, limita-se a dar resposta às áreas de exames, laboratório e imagiologia.

## 2.4 Contextualização Tecnológica – Padrões de Comunicação de Saúde

A interoperabilidade [21] é a capacidade de diferentes aplicações, dispositivos e fontes de informação, acederem, trocarem, integrarem e fazerem uso dos mesmos dados de uma forma cooperativa e coordenada. A Health Information Exchange (HIE) [21] permite a troca de informação clínica entre sistemas de informação de saúde, mantendo a integridade total da informação. O objetivo é o acesso e recolha de dados clínicos, de forma que o doente seja afetado positivamente com cuidados eficientes e eficazes.

Para a correta troca de informações foram estudados padrões de saúde, como o Fast Healthcare Interoperability Resources (FHIR)[22] e o Consolidated Clinical Document Architecture (C-CDA)[23]. Entre as diferenças encontradas, destaca-se o facto do FHIR permitir pedidos com filtro de informação, ao contrário do C-CDA que não permite este filtro. Por outro lado, o FHIR apresenta maior facilidade na implementação de soluções e é compatível com várias linguagens de *markup*, ao contrário do C-CDA, apenas compatível com XML. Assim, foi selecionado o FHIR.

## 3 Análise

### 3.1 Domínio e Casos de Uso

Além das entidades já identificadas (**Smartbox** e **Sensor**), destaca-se no domínio do problema o **Paciente** e **Profissional de saúde** – indivíduo monitorizado e indivíduo que pede a monitorização, respetivamente. A entidade **Sinal Vital** representa o alvo da monitorização e, o resultado da mesma, corresponde à entidade **Observação**. Por último, a entidade **Intervenção** equivale a uma monitorização, à qual pode ser definida uma periodicidade temporal, correspondente à entidade **Recorrência**.

De acordo com as funcionalidades necessárias, a solução deve permitir ao utilizador associar dispositivos de monitorização a pacientes, consultar aqueles associados a cada paciente (smartboxes e sensores), adicionar pedidos de monitorização a sinais vitais e a consultar os seus resultados. Assim, seguem-se os casos de uso:

- **UC1:** Associar Dispositivos – Para o profissional de saúde associar dispositivos de monitorização a um determinado paciente.
- **UC2:** Consultar Dispositivos - Para o profissional de saúde ver os dispositivos de monitorização associados a um determinado paciente.
- **UC3:** Adicionar Intervenções – Para o profissional de saúde criar pedidos para monitorizações a dispositivos, definindo recorrência temporal.
- **UC4:** Consultar Observações - Para o profissional de saúde ver os resultados dos sinais vitais relativos às monitorizações ao paciente.

### 3.2 Arquitetura

Para a arquitetura da solução foram estudadas várias alternativas, tendo sido selecionada uma arquitetura com base em microaplicações.

Nesta arquitetura são usadas peças do tipo *widget*, *middleware*, *composite* e *microservice*. Cada peça *widget* corresponde a um *microfrontend* sem conhecimento direto de negócio - um por cada um dos casos de uso. A peça do tipo *middleware* é responsável por tratar os recursos FHIR, filtrando a sua informação e enviando-a de forma adequada para as peças *widget*. A peça do tipo *composite* é responsável por construir pedidos por recursos FHIR que são enviados às peças do tipo *microservice*, que, por sua vez, realizam os pedidos dos recursos FHIR à base de dados.

Em relação à implantação, existem dois servidores distintos, o Servidor Glintt Performance/ Redundância e o Servidor Glintt Base, ambos máquinas virtuais. Estes alojam os vários ambientes de execução virtuais, os Docker Containers. A tecnologia *Docker Swarm* faz a implantação das várias peças e escalona os containers entre os dois servidores de acordo com a disponibilidade destes, gerindo a carga de ambos e garantindo um maior desempenho e redundância do sistema.

## 4 Implementação

Na implementação destaca-se a utilização de FHIR na troca de informação clínica, onde se recorreu essencialmente aos seguintes recursos FHIR: *Device*, *DeviceRequest*, *ServiceRequest* e *Observation*.

O FHIR *Device* é utilizado neste cenário para representar um dispositivo de monitorização, (*smartbox* ou respetivos sensores), enquanto que a associação entre o dispositivo e o paciente é representada pelo recurso FHIR *DeviceRequest*, que contém informações como a data de associação e referências para outros recursos FHIR.

O recurso FHIR *ServiceRequest* é adequado para representar a criação de um pedido de calendarização de monitorizações com recorrência, tendo um campo que referencia o dispositivo (FHIR *Device*), que deve realizar a intervenção, sendo criada uma instância do mesmo para cada um destes tipos de pedido.

Por fim, o recurso FHIR *Observation* permite agregar informação sobre resultados clínicos. Por cada sinal vital e por cada monitorização, é usada uma instância deste recurso, que apresenta o valor, unidade de medida e a data / hora do resultado.

## 5 Resultados e Avaliação

Após associação de dispositivos a um paciente, calendarização e execução de monitorizações, os resultados dos sinais vitais podem ser consultados (ver Fig.2). Estes resultados foram recolhidos através de monitorizações feitas com intervalos de uma hora, relativos a frequência cardíaca (batidas por minuto), frequência respiratória (respirações por minuto), saturação de oxigénio (percentagem de oxigénio no sangue), temperatura (graus Celsius) e pressão arterial (milímetros de mercúrio). Todos os valores recolhidos apresentam, neste caso, resultados normais e saudáveis.

Frequência Cardíaca(bpm)	63 06h00	66 07h00	65 08h00	73 09h00	- -
Frequência Respiratória(cpm)	13 06h00	13 07h00	12 08h00	15 09h00	- -
Saturação de O2(%)	98 06h00	98 07h00	98 08h00	98 09h00	- -
Temperatura(°C)	36.1 06h00	36.1 07h00	36.3 08h00	36.5 09h00	- -
Pressão Arterial(mmHg)	120/82 06h00	125/81 07h00	123/81 08h00	132/86 09h00	- -

**Fig. 2.** Resultados UC4: Consultar Observações

Para avaliar a solução, foram realizados testes unitários, de aceitação, de integração e de carga e usado o modelo de avaliação Quantitative Evaluation Framework, que estimou em 94% a qualidade do sistema real da solução implementada.

## 6 Conclusões e Trabalho Futuro

Conclui-se que foi cumprida a hipótese de conceptualizar um CPOE capaz de fazer a gestão transversal para a área de monitorização de sinais vitais. Relativamente a trabalhos futuro, existem diversas áreas que podem ser acrescentadas ao CPOE, de forma a tornar-se um software completo. É uma meta ambiciosa e exigente, no entanto este produto só assim poderá ganhar maior relevância e perder o estatuto de MVP. Paralelamente, esta questão foi também uma limitação, dado que a implementação de apenas uma área nunca seria suficiente para desenvolver um CPOE completo.

## Referências

1. Gregório, J. *et al.* (2021) ‘The role of Design Science Research Methodology in developing pharmacy eHealth services’, in *Research in Social and Administrative Pharmacy*. Elsevier, pp. 2089–2096. doi: 10.1016/J.SAPHARM.2021.05.016.
2. Salvaneschi, P. (2010) ‘Safety testing of computerized provider order entry systems’, *Proceedings - International Conference on Software Engineering*, pp. 74–82. doi: 10.1145/1809085.1809095.
3. Jungreithmayr, V. *et al.* (2020) ‘The impact of a computerized physician order entry system implementation on 20 different criteria of medication documentation-a before-and-after study’. doi: 10.1186/s12911-021-01607-6.
4. Aghazadeh, S., Aliyev, A. Q. and Ebrahimnejad, M. (2011) ‘The role of computerizing physician orders entry (CPOE) and implementing decision support system (CDSS) for decreasing medical errors’, *2011 5th International Conference on Application of Information and Communication Technologies, AICT 2011*, pp. 4–6. doi: 10.1109/ICAICT.2011.6110916.
5. CPOE Systems <https://www.altexsoft.com/blog/cpoe-systems-and-electronic-prescribing-software/>, último acesso a outubro de 2022
6. Baysari, M. T. *et al.* (2017) ‘Longitudinal study of user experiences of a CPOE system in a pediatric hospital’, *International Journal of Medical Informatics*, 109, pp. 5–14. doi: 10.1016/j.ijmedinf.2017.10.018.

7. Khanna, R. and Yen, T. (2014) 'Improving Health Care Quality: Reviews Computerized Physician Order Entry: Promise, Perils, and Experience'. doi: 10.1177/1941874413495701.
8. Griffon, N. *et al.* (2017) 'Physician satisfaction with transition from CPOE to paper-based prescription', *International Journal of Medical Informatics*, 103, pp. 42–48. doi: 10.1016/j.ijmedinf.2017.04.007.
9. Subramanian, S. *et al.* (2007) 'Computerized physician order entry with clinical decision support in long-term care facilities: Costs and benefits to stakeholders', *Journal of the American Geriatrics Society*, 55(9), pp. 1451–1457. doi: 10.1111/J.1532-5415.2007.01304.X.
10. Dhamanti, I. *et al.* (2021) 'Implementation of Computerized Physician Order Entry in Primary Care: A Scoping Review'. doi: 10.2147/JMDH.S344781.
11. Korb-Savoldelli, V. *et al.* (2018) 'Prevalence of computerized physician order entry systems– related medication prescription errors: A systematic review', *International Journal of Medical Informatics*, 111, pp. 112–122. doi: 10.1016/J.IJMEDINF.2017.12.022.
12. Wang, B. and Liu, X. (2019) The Implementation and Effects of Computerized Physician Order The Implementation and Effects of Computerized Physician Order Entry in Healthcare Settings Entry in Healthcare Settings Recommended Citation Recommended Citation THE IMPLEMENTATION AND EFFECT.
13. Sapra, A., Malik, A. and Bhandari, P. (2022) 'Vital Sign Assessment', *StatPearls*. Available at: <https://www.ncbi.nlm.nih.gov/books/NBK553213/> (Accessed: 31 July 2022).
14. Li, Y., Chen, W. and Lu, L. (2021) 'Wearable and Biodegradable Sensors for Human Health Monitoring', *ACS Applied Bio Materials*, 4(1), pp. 122–139. doi: 10.1021/ACSABM.0C00859.
15. Salvo, P. *et al.* (2018) 'A wearable sweat rate sensor to monitor the athletes' performance during training', *Science and Sports*, 33(2), pp. e51–e58. doi: 10.1016/J.SCISPO.2017.03.009.
16. Cappon, G. *et al.* (2019) 'Continuous glucose monitoring sensors for diabetes management: A review of technologies and applications', *Diabetes and Metabolism Journal*, 43(4), pp. 383–397. doi: 10.4093/DMJ.2019.0121.
17. Tavakoli, M., Benussi, C. and Lourenco, J. L. (2017) 'Single channel surface EMG control of advanced prosthetic hands: A simple, low cost and efficient approach', *Expert Systems with Applications*, 79, pp. 322–332. doi: 10.1016/J.ESWA.2017.03.012.
18. PatientKeeperCPOE <https://www.patientkeeper.com/about-us/news/press-release.html?id=22747>, último acesso a fevereiro de 2022
19. Practice Fusion <https://www.practicefusion.com/>, ultimo acesso a fevereiro de 2022
20. PowerOrders <https://www.cerner.com/-/media/cerner-media-united-states/certified-health-it/cerner-2015-edition-certified-health-it-costs--limitations-disclosures-v32521revised.pdf?vs=1&hash=3520C2110046B4BBEC2BB4CBBC5D968F>, ultimo acesso a fevereiro de 2022
21. Interoperabilidade <https://www.himss.org/resources/interoperability-healthcare>, último acesso a fevereiro de 2022
22. FHIR <https://www.hl7.org/fhir/summary.html>, último acesso a fevereiro de 2022
23. C-CDA <https://www.particlehealth.com/blog/what-is-ccda-consolidated-clinical-document-architecture>, último acesso a fevereiro de 2022
24. Escudeiro, P. *et al.* (2008) 'Quantitative Evaluation Framework (QEF)', *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 176 LNICST, pp. 117–124. doi: 10.1007/978-3-319-51055-2\_15.

# Coronabot: Providing COVID-19 Information with a Chatbot in European Portuguese

Ricardo Ferreira<sup>1</sup>[0000-0003-0494-6752] and Luís Coelho<sup>1</sup>[0000-0002-5673-7306]

<sup>1</sup> Porto School of Engineering, Porto, Portugal  
{1170960, lfc}@isep.ipp.pt

**Abstract.** The several lockdowns and limitations imposed by COVID-19 has caused a migration of many healthcare services to cloud and to digital versions, when support technologies were available. Additionally, the surge of a new disease has flooded healthcare information services with a wave of questions, creating huge response challenges and leading to many service crashes. In this context, chatbot technology can be a helpful tool and provide effective response in a large scale, allowing to cope with unusual demands. In this paper we present the CoronaBot, a chatbot to explain and provide facts about COVID-19, based on the users' questions. To ensure the quality of the information, all the given responses were based on reliable reference organizations in the areas of health and statistics. To support the explanation of the chatbot development, an initial theoretical background is provided, covering an brief chronological perspective as well as the core related concepts. The steps taken during the development of the Coronabot will be thoroughly presented, covering architecture and implementations aspects. The demonstration and explanation of the different stages are also provided. Results regarding the utilization of the Coronabot by the testers are also shown, and some possible future developments are discussed.

**Keywords:** Artificial Intelligence, Machine Learning, Supervised Learning, Natural Language Processing

## 1 Introduction

To slow down the spread of pandemic caused by the COVID-19, various recommendations regarding hygiene, social distancing, and vaccination were provided by competent entities in the field, such as the World Health Organization (WHO) and the Portuguese General-Directorate of Health (DGS). However, the high amount of available information makes it difficult to find the answer to a specific question from consumers who are not used to performing such a search. When searching for an answer to a specific question, while avoiding spending more than strictly necessary, a specialized chatbot can be used. This technology has existed in online environments for some time and is commonly used to facilitate Customer Service on e-commerce sites. Nonetheless, chatbots are currently shifting their focus to widely used instant messaging interfaces, such as Facebook Messenger, reaching end-user more easily [1]. A study carried out in

2012 through an online survey with 146 chatbot users from the United States of America shows that 84% of them use Messenger as an interaction platform [2].

Instant messaging platforms are considered one of the main communication technologies and are widely used for information exchange. Allied with these facts, it is estimated that the global chatbot market should reach a value equivalent to 2 billion dollars by 2024, growing with a 29.7% Compound Annual Growth Rate (CAGR) [3]. Simultaneously, Python has become one of the most popular programming languages in recent years, being a leading-edge language in the fields of Data Science (DS) and Machine Learning (ML) [4]. Considering all this information, the authors created a chatbot, developed through the Python language, focused exclusively on answering questions about the Coronavirus, a prominent disease in modern society. The main objective was to develop a chatbot capable of understanding and answering various questions related to COVID-19. Associated with this, it must provide information that is sourced from reference organizations in the areas of health and statistics. launching it in the cloud, after achieving positive results in the scoring tests. Finally, it is essential to enable interaction between Facebook Messenger users and Coronabot, since only then will the chatbot be completely functional to the public.

## 2 Chatbot Historical Chronology and Current Landscape

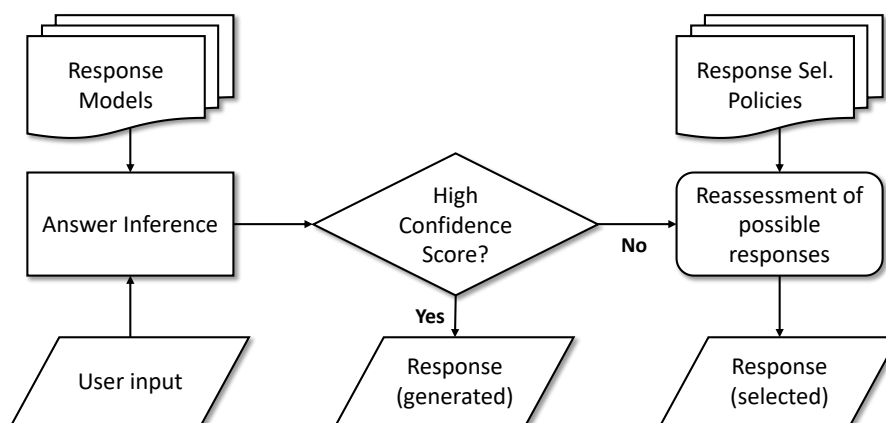
The possibility of interacting with a machine is a dream since the appearance of basic computational systems. With this in mind, a test was created by Turing (1950), exploring the ability of machines to trick people into thinking that they (the machines) are real people [5]. Eliza, the first chatbot to be publicly presented, in 1966 at the MIT, was able to communicate with humans using a pattern matching scheme [6]. In 1972, PARRY, a more realistic chatbot developed at Stanford, was able to simulate the dialogue between a therapist and a schizophrenic patient. When used to train psychologists, many could not distinguish between PARRY's dialogue and that of a real schizophrenic patient [7]. The Artificial Linguistic Internet Computer Entity (ALICE) was created by Richard Wallace in 1995. The chatbot engine and the language knowledge model are separated, so that it is possible to install alternative linguistic models [8]. ALICE engine uses Artificial Intelligence Markup Language (AIML) to answer user questions and is freely available to the public [9].

During this century, significant advances have been made in chatbot-related fields. One of the most outstanding achievements is the emergence of voicebots, made possible by the advancements in speech synthesis and recognition technologies. Voicebots are part of the chatbot family, but their communication is done by capturing and recognizing the user's speech and issuing a voice response [10]. Apple's SIRI, presented in 2010, with the objective of becoming a personal assistant that, using AI, improves as it interacts with its user [11]. Microsoft's Cortana, released in 2014, is a digital assistant that can be used as a personal organizer, a research tool (both locally on the computer and for web searches), to send text messages, emails, and much more [12]. Amazon's Alexa, announced in 2014, is a voice-controlled digital assistant, but according to a 2018 study, it is mainly used to check weather forecasts, play music, and control

devices, the latter being a prominent area within its ecosystem [13]. The Google Assistant, 2016, a similar system, announces a better privacy policy, and the possibility to participate in discussions in various languages [14]. Despite their differences, most of the described AI personal assistants are able to answer questions, provide reminders, take notes, and make translations, which fulfill a wide range of daily task requirements. Concerning the healthcare area and COVID-19 specifically, the HealthBuddy is a chatbot that was born from a joint initiative between the WHO and the United Nations Children's Fund (UNICEF), supported by experts from these institutions. Its main objective is to provide reliable information about COVID-19 and combat misinformation related to this disease, being accessible worldwide in multiple languages [15].

### 3 Development

Intending to select the best answer and respond to the user in the most efficient way, chatbots often follow the logic represented in Fig. 1.



**Fig. 1.** Flowchart of the response process of a generalized chatbot

First, user input/interactions are evaluated, considering the response models, so that a response hypothesis can be generated. While generating the hypothetical answer a confidence score is also provided and evaluated. If it surpasses a given pre-defined threshold, then it is considered as being correct and it is sent to the user, otherwise, using context dependent selection policies, an alternative response is defined with a higher confidence score, which is then provided to the user. These processes are what enable the synthesis and analysis of computational agents capable of acting intelligently or simulating intelligence [16].

The answer inference engine and the response model are built around dialogue possibilities associated with the users' objectives. In the case, working in a limited domain, we can define distinct interaction stages related with the chatbot functionalities: To be able to a) understand greetings (introduction or farewell), b) understand when information is requested and different types of information, c) understand the scope of the

request (geographical span concerning statistics or population group concerning disease information), and d) provide a contextualized list of symptoms or information.

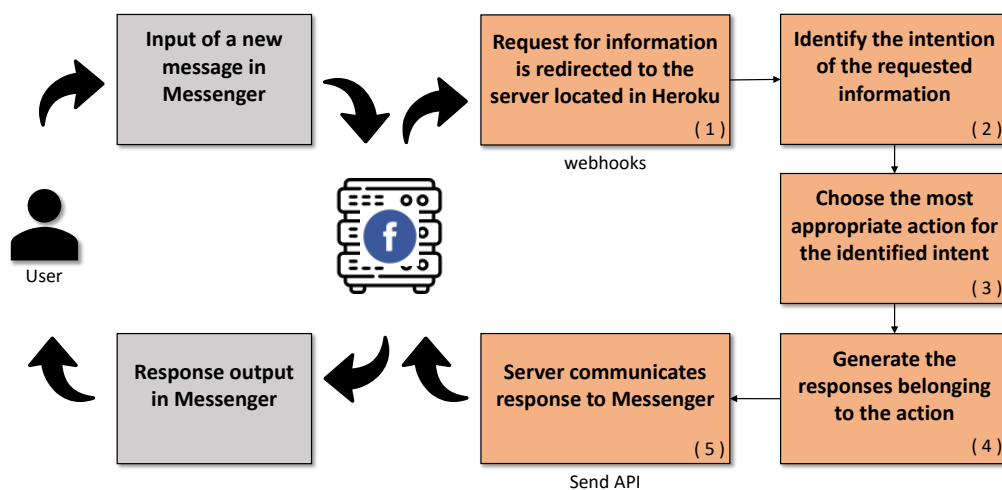
The development of the chatbot was supported by the RASA framework, an open source natural language understanding (NLU) and dialogue management library (developed with the Python programming language) that makes use of artificial intelligence algorithms, namely TensorFlow. The first step requires the definition of an initial set of possible statement or questions coming from the user with an associated intent and entity, with an optional regex expression for generalization. “Intent” and “entity” are chatbot related concepts, with the first referring to the goal of the user when provides inputs and the last being the object of the intent. In the RASA framework these can be used to position the limit the dialogue’s scope and to provide better answers. For each intent, several different sentences should be provided, covering syntax or vocabulary variations. This preparation step will contribute to an improved system performance. The next step is to train the prediction model, which is a fairly automated process. Besides the data from the previous step, it is necessary to specify the base language (to ensure language specific tools are used) and a processing pipeline (usually from a set of predefined possibilities). If adequate parameters are provided, then a successful training is achieved, and the system will be able to extract the correct intent and entity for a given user input.

Although the prediction model was already created, it wasn’t possible to carry out a full conversation yet. For this to happen, it is necessary to train an additional model for dialogue management. The RASA Core, a complementary module of the NLU engine, was used for this step. To define what responses the Coronabot could give, it was necessary to provide it with scripts that simulated real conversations. To accomplish this, the following should be defined: a) The domain or context limits in which Coronabot could operate, by providing the intents that are recognized, the adequate response templates (possible responses), and actions that can performed; b) A dialogue logic, containing example scripts of how conversation can flow within a given intent; c) The templates, which are examples of basic dialogs that allow the system to perform simple conversations. After this, the dialogue engine is ready to be trained and tested.

For testing and operating the CoronaBot it was deployed and hosted at Heroku, a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. The decision to use this platform was based on its support of the project’s technologies, no cost within the project’s scope and the possibility with simple integration with other services. The last motivation was especially important since we wanted to reach a large number of users by integrating the chatbot engine with popular messaging services or platforms. Once an Heroku account has been created, it is possible to launch the application in the cloud. This action is necessary to later connect the chatbot to other platforms. After the application was created and the locally stored files were deployed, access to the application in the cloud can be enabled. In addition, the locations of the dialog and prediction models must be specified, so that all Coronabot features could be used and for Heroku to specify the commands that should be executed by the application when it starts.

Facebook social network is widely disseminated and contains one of the most used messaging services, making it an excellent test bed for the here presented project. To

integrate the Coronabot with Facebook, it was first necessary to associate it with a Facebook account and to create a Facebook hosting page. After this, under the developers console, a new Facebook app, with its own identification number, must be created. To access the API, a token is generated, giving access to callback URLs, opening the possibility to create background interactions with external applications. Using Facebook for Developers, it is necessary to specify that Messenger will be required for operation by including it “as a product”. For send data, the webhook responsible for communicating with Heroku was defined. To do so, it was necessary to use the webhooks sub-field, and define a Callback URL and a *Verify Token*. The Callback URL was the one associated with the application created within Heroku, and the *Verify Token* refers to the sentence used as a verification code. With the possibility to communicate with Coronabot using Facebook, it is important to understand what happens during this communication. The interaction process is intricate and uses a set of files that must be in specific locations. To better describe this process, in **Fig. 2** a simplified flow diagram is presented, covering the interaction pipeline of the Heroku hosted chatbot with Facebook Messenger.



**Fig. 2.** Schematic of the interaction pipeline between the user and the chatbot, using Facebook Messenger as deployment platform.

During the process in box 1, a new user message is sent to the Coronabot server, through the link previously provided to Facebook for Developers. Receiving a new message, the Procfile file is activated, triggering the download of the Portuguese language module from spaCy and the facebook.py file. From here on, facebook.py is the file responsible for steps 2, 3, 4, and 5. To complete steps 2, 3, and 4, facebook.py contains information regarding the location of the files that make up the chatbot, as well as the training models. The files corresponding to the chatbot must be in the folder ‘models/nlu/default/coronabot/’, and the models are in ‘models/dialogue’. After these steps are complete and the chatbot already selected a reply, a new contact is made with Messenger (box 5), using the access credentials. With the contact established successfully, the reply is sent to the user.

## 4 Evaluation, Results and Discussion

The preliminary evaluation was carried out with the participation of 6 testers, 3 females and 3 males, with 25 years being the average age of the group, with the youngest member 16 years old and the oldest 48 years old. Each of the testers was introduced to the features of the Coronabot, and they were not given any indication of how to proceed with the questions/requests, much less told how to write their sentences. This was to ensure that no language (vocabulary or syntax) bias was transmitted and to achieve minimal influence on the part of the developers.

Dealing with language can be challenging and after the first evaluation sessions a recurrent error was detected. Some of the female testers used the interjection "Obrigada!" to thank the bot for the service provided. This word is sometimes used as a feminine version of the work "Obrigado", the most common form. At this stage, the chatbot could not recognize the intent of the sentence, thus attributing a wrong answer. After this bug was detected, a set of feminine words variations was included.

Although the features of the Coronabot were carefully explained, the interpretation of some testers managed to circumvent the type of question/request the bot would be prepared to receive. This led to some intentions being misinterpreted, which in turn caused the sending of undocked responses from the context. For example, when it was said that Coronabot can provide a list of symptoms associated with COVID-19, the expected interactions would be in the order of "I'd like to know more about symptoms associated with coronavirus.", or "What are the symptoms of Covid-19?". However, some testers used expressions like "I have a fever." or "I have a cough.", which are not included in the examples provided during the development of the Coronabot, since it is not intended to diagnose pathology, causing some interaction challenges. On the other hand, still related to symptoms, some testers asked how they should proceed if they had any of the symptoms, both in terms of medication and in professional follow-up. Although this functionality is not integrated into Coronabot, it may be interesting to implement it in possible future developments of this project, since more functionalities would make this chatbot more efficient in dispensing information.

Regarding the response time, most of them reported that the first interaction takes the longest to be answered, with time gaps varying between approximately 30 seconds and 3 minutes. These are not acceptable in a ready to market product and are related to the fact that the application provided by Heroku enters a state of hibernation if there is no interaction for some time. Therefore, these response times are associated with the time it takes for the application to come back online, and to perform the necessary preparations to be functional. The remaining responses arrive faster, and sometimes the response time is so short that it can be considered almost instantaneous, considering the initial situation. Frequent usage of the system also contributes to eliminate this delay.

After each tester had finished testing Coronabot, they were asked to make a qualitative assessment of their experience. The answers were highly positive, and the project fulfilled the objectives it initially set out to achieve. In general, the testers managed to generate several useful pieces of information for the project, namely regarding the variety of sentences, some of which were implemented in the examples contained in a data.json file, to reinforce future dialogs, by re-training, and to improve performance.

According to what could be observed by the results obtained, Coronabot is still not a fully natural conversational agent, one that could be perceived as a human, but it has managed to perform most of the functions with a good degree of precision if the questions are specific. When operating in a wide scope context, despite sometimes failing to identify the intent, the chatbot had a comparatively considerably high success rate of 74%. This value was calculated by counting the success of the user when accessing the chatbot with a specific intention, divided by all the accesses to the chatbot.

This project was intended to be a proof of concept, with the purpose of integrating several distinct web technologies, while using a small dataset. Thus we believe that the integration of the tools is a promising framework for chatbot development and for further improvements.

## 5 Conclusion and Future Work

Before drawing any conclusions, it is important to remember that the main objective of this project was to develop a chatbot capable of answering questions related to COVID-19. It was possible to observe Coronabot in action, which highlighted several of the advantages of the chatbots to assist human users, such as its 24-hour availability and its almost immediate response time. Of course, there are also disadvantages, but with the evolutions in the fields of AI and ML, these become increasingly reduced. Also associated with the observation of Coronabot in action, when interacting with the testers, the importance of having a test group for whatever work is intended to be used by a population was highlighted. Without the results provided by the latter, some failures and lapses would have gone unnoticed, and Coronabot would not have been able to evolve so quickly. It was also noted the importance of social networks, in terms of their role as a weapon against (or in favor of) misinformation since many people use them to inform themselves about the most varied topics.

Associating a chatbot with Facebook does not only bring advantages. Despite gaining greater visibility to the public, some of the page's customization capabilities are lost, and there may be delays in communication between Facebook and Heroku servers, and sometimes a message may even go unanswered, despite Coronabot having been able to respond correctly to it during the training phase. Another limitation would be the absence of an automatic spell checker. This opens the possibility for bugs derived from unknown words to occur, making it difficult for the Coronabot to communicate with the users. Another option would be to provide more than one language for Coronabot since it can only communicate in Portuguese, using spaCy's various language modules, which currently count 15 different languages. Furthermore, it would only be necessary to create a data.json file for each language and make some slight changes to the code that contains the specified language module to be used. Finally, it would be important to try to humanize Coronabot more, as this makes any conversational agent more appealing.

## References

- [1] M. Van Eeuwen, "Mobile conversational commerce: messenger chatbots as the next interface between businesses and consumers," *Univ. Twente*, p. 15, 2017.
- [2] L. Bradeško and D. Mladenčić, "A Survey of Chabot Systems through a Loebner Prize Competition," vol. 2, pp. 1–4, 2012.
- [3] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, "Xatkit: a Multimodal Low-Code Chatbot Development Framework," *IEEE Access*, vol. 8, pp. 15332–15346, 2020, doi: 10.1109/aACCESS.2020.2966919.
- [4] M. Hu and Y. Zhang, "The Python/C API: Evolution, Usage Statistics, and Bug Patterns," *SANER 2020 - Proc. 2020 IEEE 27th Int. Conf. Softw. Anal. Evol. Reengineering*, pp. 532–536, 2020, doi: 10.1109/SANER48275.2020.9054835.
- [5] G. Marcus, F. Rossi, and M. Veloso, "Beyond the turing test," *AI Mag.*, vol. 37, no. 1, pp. 3–4, 2016, doi: 10.1609/aimag.v37i1.2650.
- [6] H. yeung Shum, X. dong He, and D. Li, "From Eliza to XiaoIce: challenges and opportunities with social chatbots," *Front. Inf. Technol. Electron. Eng.*, vol. 19, no. 1, pp. 10–26, 2018, doi: 10.1631/FITEE.1700826.
- [7] K. R. Stephens, "What Has the Loebner Contest Told Us About Conversant Systems?," *Retrieved Oct.*, vol. 28, p. 2005, 2002, [Online]. Available: <http://behavior.org/resources/325.pdf>.
- [8] B. A. Shawar and E. Atwell, "ALICE chatbot: Trials and outputs," *Comput. y Sist.*, vol. 19, no. 4, pp. 625–632, 2015, doi: 10.13053/CyS-19-4-2326.
- [9] B. S. Bani and A. P. Singh, "College enquiry Chatbot using ALICE," *Int. J. New Technol. Res.*, vol. 3, no. 1, pp. 1–3, 2017.
- [10] S. Revathy, R. Niranjani, and J. Roslin Kanushya, "Health care counselling via voicebot using multinomial naive bayes algorithm," *Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020*, no. Icces, pp. 1063–1067, 2020, doi: 10.1109/ICCES48766.2020.09137948.
- [11] E. Naone, "Intelligent software assistant," *Technol. Rev.*, vol. 112, no. 2, pp. 1–2, 2009.
- [12] B. Singh and U. Singh, "A forensic insight into Windows 10 Cortana search," *Comput. Secur.*, vol. 66, pp. 142–154, 2017, doi: 10.1016/j.cose.2017.01.007.
- [13] I. Lopatovska *et al.*, "Talk to me: Exploring user interactions with the Amazon Alexa," *J. Librariansh. Inf. Sci.*, vol. 51, no. 4, pp. 984–997, 2019, doi: 10.1177/0961000618759414.
- [14] A. Ara and S. Jawaligi, "NodeMCU (ESP8266) Control Home Automation using Google Assistant," *Int. Res. J. Eng. Technol.*, vol. 6, no. 7, pp. 3644–3648, 2019.
- [15] H. R. Saeidnia, Z. Mohammadzadeh, and M. Hassanzadeh, "Evaluation of mobile phone healthcare applications during the Covid-19 pandemic," *Public Heal. Informatics Proc. MIE 2021*, pp. 1100–1101, 2021, doi: 10.3233/SHTI210363.
- [16] D. L. Poole and A. K. Mackworth, "What is Artificial Intelligence?," in *Artificial Intelligence: foundations of computational agents*, 2010, pp. 3–6.
- [17] J. P. Silva, "Year One of Heroku," in *Startups in Action: The Critical Year One Choices That Built Etsy, HotelTonight, Fiverr, and More*, Berkeley, CA: Apress, 2020, pp. 85–97.

---

# Designing The Talent Marketplace Of The Future

E. Luciano Zickler Paz, Susana Nicola, and Nuno Bettencourt

Interdisciplinary Studies Research Center, Institute of Engineering of Porto,  
Polytechnic of Porto (ISRC/ISEP/IPP)  
{1210095,sca,nmb}@isep.ipp.pt

**Abstract.** The future of work is deepening the skill gaps of organisations and disengagement at work is rampant. This work's objective is to design an internal gamified SaaS Talent Marketplace to find hidden champions, engage the workforce, and contribute to organisational and personal development. The novelty of the proposed solution lays in the possibility of integration with various services and the use of a hybrid Recommender System based on ontologies and collaborative filtering. Furthermore, this proposal is sustained by a new conceptual framework that combines the Technology-Organisation-Environment model to drive technology adoption of the design and the Design Science Research Methodology for future contextual evaluation.

**Keywords:** Talent marketplace, Skill matching, Technology adoption framework, Recommendation

## 1 Introduction

Due to the adoption of new disruptive technology, demographic and socio-economic changes, the reality of work has changed, bringing about the future of work [1]. A transition from static to project-based work is underway due to specialisation of knowledge, automation and new ways of work (*e.g.*, remote) [2]. This makes it increasingly hard to find the right person for the job, increasing skill gaps in the workforce [3].

Top talent attraction and retention has become the most important internal issue for organisations [4]. When people are satisfied with their jobs, they are also able to realise their full potential, which means that they can contribute to a positive organisational outcome: talent retention, safety, sales, and production improve [5]. However, fewer than 20% of the workforce is engaged in their job in the EU and almost 15% is actively disengaged [6]. This leads to heightened attrition and lessened productivity.

In response to these challenges, organisations are adopting internal Talent Marketplaces (TMs) to “stave of attrition” [7], providing training and internal mobility (internal hiring or project-based deployment) opportunities to their workforce. TMs often work in conjunction with Human Capital Management

(HCM) suites, usually operating as cloud solutions. Core functionalities of HCM services are HR administration, talent management and workforce management, which makes it easy to import business data and processes into the TM when working together.

In this paper, a scalable, human-centred, gamified TM is presented. The TM's design decisions should be based on technology adoption probability, thus a conceptual framework for its design will be discussed.

The TM shall provide economical and organisational value for businesses, as well as growth opportunities and sustainable engaging work for the whole workforce. The goals of the proposed TM are: produce novel and serendipitous matches between collaborators and projects, finding more candidates for a given task; provide continuous learning opportunities based on career goals and inform organisational development of strategic objectives and initiatives attending to employee's interests and insights into human capital development; better engage and retain the workforce; easily deployable to businesses, easily integrated with other services, and present easy to use interfaces for faster employee adoption.

In section 2, the methodology followed on this paper is laid out, that is, establishing a conceptual framework for the design and evaluation of the proposed TM, according to its goals and requirements. Section 3 presents the proposed TM design. Section 4 pertains to the value analysis of the TM attending to the conceptual framework and the perceived merit of this design. In section 5, the value analysis results are discussed and a conclusion presented.

## **2 Methodology and Conceptual Framework**

The Technology-Organisation-Environment (TOE) model [8] has previously been used in SaaS design and evaluation [9]. It is well suited to explain organisational level technology adoption. According to an extensive literature review of the application of the TOE framework to cloud platforms by [9] the following factors are to be used: perceived value, security concerns, configurability and customisation, organisational readiness, top management support, competitive pressure, and vendor qualities. The TOE framework is adopted to guide design and analyse the value proposition of the TM.

The technological context directly influences on the TM design choices; the perceived value, referring to the degree of innovation and usefulness in comparison to existing solutions is also used in the value analysis. Organisational readiness (having the necessary technology and financial means) and top management support (organisational intent) are used to establish the TM's merit. Environmental factors are also used to validate the artefact. Competitive pressure or the industry stress for a prospective customer business has a positive impact on technology adoption. Vendor quality refers to the TM's market positioning and trust.

Organisational and environmental context are evaluated with help of the Business Model Canvas (BMC) approach [10], this includes a market segmentation, analysing needs and trends of a given segment, conducting decision-unit and end-customer (user) investigations (interviews, studies or behavioural research) and comparative analysis with existing products and competitors. The

market segment under scrutiny is the technical consultancies (Capgemini, IBM, Accenture, among others) and IT sector, as they were found to be closely related, open to new technology, have the necessary financial means and have the fastest changing skill requirements among sectors. Findings are backed by some decision-unit interviews and behavioural research studies pertaining to the field of organisational psychology and user-experience design.

The TM is modelled from the needs of organisations to maximise their internal competencies (and close skill gaps), provide continuous re- and upskilling of their collaborators to attain business objectives and critical competencies development, and keeping their workforce engaged. To this end, the Design Science Research Methodology (DSRM) pertaining to the role of digital innovation in research, as proposed in [11] together with Pfeffer's nominal process sequence as research plan canvas [12] are employed to validate and evaluate the TM contextually. The DSRM accompanies the TM through its entire lifecycle (design, development, deployment, reuse etc.) with the goal of iteratively evaluating the artefact. After each iteration, the artefact needs to be practically demonstrated and add value to the knowledge base from which it is evaluated. For instance, the initial iteration of the skill marketplace component is to be demonstrated through a field study at ISEP, recommending appropriate internships to students, and results publicised for addition to the information systems knowledge base.

### 3 Proposal Design

The internal TM is a Software as a Service (SaaS) that helps organisations find and explore their full potential. It is a marketplace where collaborators get tailored project recommendations, allowing them to work on projects that fit their skills and interests; boosting productivity and engagement, while combating idle time and misplaced resources. The Recommender System (RS) of the TM also matches people to continuous learning opportunities according to their career preferences, contributing to the development of critical skills and future leaders in the organisation.

The SaaS TM provides four key capabilities : competency building, skill marketplace, analytics, and future planning. The TM matches collaborators to suitable opportunities through the skill marketplace and competency building component . The competency building provides learning path and career path options for the employees and offers continuous learning through external training providers.

The skill marketplace is driven by skill demand and project offerings, allowing both internal and external workforce (contractors, freelancers, volunteers, services, bots, etc.) to search and get recommendations on suitable projects. To track individual and organisational progress, analytics on work engagement, productivity and skill-capacity are recollected from the TM. Use cases of analytics include warnings (*e.g.*, skill shortages) and allowing predictions for future resource and strategy planning (future planning). This is fundamental for identifying future critical competencies that should be integrated into members' learning path choices.

On the TM organisation members convey their career path wishes or direction they would like to develop in, as well as their interests and motivations. Project and talent data is integrated into the TM. Talent data is aggregated into skill-profiles (one for every member). Using Artificial Intelligence (AI) methods, skill-profiles and project requirements are matched together automatically and efficiently at the skill marketplace (figure 2). Members get project recommendations tailored to their smart profile and can get in touch or apply to those projects from within the platform, establishing direct contact between the project stakeholders without need of HR intervention. Projects can range from multi-year full-time jobs to knowledge or skill leasing, where individuals briefly help on a certain task. The RS also recommends learning paths according to career preferences.

Hidden champions are found in the talent pools thanks to the use of a hybrid RS optimised to make novel and serendipitous recommendations. The measure of novelty makes use of the “long tail” of recommendations, that is less frequent relationships [13], while serendipity relates to finding relevant and unexpected recommendations. Ontologies and collaborative filtering will be used for the first time together on a TM RS. Hybrid recommender systems [14–16] and the use of ontologies and collaborative filtering specifically, have been found to provide meaningful serendipitous results [17] and solve the scalability and scarcity problem encountered in RS [18]. Talent pool data includes hard and soft skills, as well as Personal Preference Indicators (PPI). PPI could include employee career goals, measures of engagement, type of work preferences (remote, on-site, project-based, etc.), time preferences (full-time, half-time, occasional, etc.), among others.

To motivate employees into using the SaaS and having a positive impact on both the organisation and their own personal growth, gamification techniques are employed. A reward system is put in place both for project-work and for progress on their learning path, figure 2. As a reward for working on projects, employees receive platform currency that they can use for training offerings. Additional rewards (e.g., assigning time for personal initiatives or monetary compensation) can be set by corporations if deemed beneficial. Furthermore, individuals are able to track their progress on selected learning and career paths with badges and official accreditations for major milestones.

Continuous learning of employees and skill matching lead to bridging skill gaps and ensuring a culture of lifelong learning (LLL) within an organisation, building up organisations’ capacities. As companies can plan their future according to workforce’s selected learning paths and existing skills, organisations can aim for a sustained, inclusive, and sustainable growth. Ultimately, stakeholders’ preferences and motivations decide on future projects and critical competencies inside an organisation: strategic objectives are driven by stakeholder interest. Workforce and organisational vision become aligned.

In figure 3 the SaaS TM system component diagram with process flow and technologies is presented. The Data management centre extracts customer organisation data from various sources, transforms and stages it, and loads the

harmonised Resource Description Framework (RDF) data into the data-mart. Skill-profiles are created that reflect members preferences, goals and skills; these can be accessed through the Single Page Application (SPA) and are stored in talent pool data-marts. Collaborators and managers can further edit or add skills and experience manually using the SPA.

For end user experience simplicity, the TM's SPA is integrated into frequently used collaborative platforms, the likes of Microsoft Teams, Slack or Discord. This means that end users do not need to adapt to a new User Interface (UI) as they are already accustomed to using these platforms tools and check-in frequently.

Members receive project and training recommendations over the SPA and can accept, decline or otherwise interact with these offers. End user behaviour is fed back to the RS to improve future recommendations. Training can be accessed via external training platforms.

Managers can feed project and talent information into the TM directly over the SPA or using present technology, like Enterprise Resource Planners (ERP) or HCM systems. External services are integrated into the TM as data sources.

The laid-out architecture is highly modular and easy to integrate (*e.g.*, SPA UI highly adaptable, ETL together with data warehousing allows for different data sources). The modularity allows for iterative and independent component lifecycles, making it easy to apply the DSRM onto future system evaluation.

## 4 Value Analysis

The TM unique value proposition is finding hidden talents where other market-places fail.

**Technological Context.** The closest competing SaaS TM are Kalido.me and Tandemploy. Kalido uses semantic similarity in text mining and Tandemploy ontologies and rule based for their RS. The proposed RS is expected to produce more novel and serendipitous recommendations as evidenced by benchmark studies of ontology driven hybrid RS on LLL opportunities [17].

[19] found that gamification has a positive correlation with work productivity for leaders, while [20] found that together with learning readiness it influenced intrinsic motivation and adoption of LLL. The authors pose that gamification together with technological readiness are key to foster adoption of the TM. Use, acceptance and adoption of automated systems, like the RS, has been shown to be positively correlated to intrinsic and extrinsic rewards [21]. Gamification is generally considered an extrinsic incentive, while feedback is intrinsic; the reward system provides both types of reward.

It is strongly believed that employee engagement is a deciding factor for task outcome [5]. According to [22] there are three main factors contributing to personal engagement at work: meaningfulness, safety, and psychological availability. Presenting organisation members with projects that reflect their skills and preferences, and giving them choice, leads to individuals becoming aware of their abilities, preferences, and their role inside the organisation, increasing availability. People identify with their task and find meaning applying their persona to the project. Engagement is thus expected to increase in the workforce using the TM.

**Organisational Readiness and Competition.** The IT sector is a very competitive market with its value deriving from innovation. Human capital is especially valuable as innovation stems from talent; TM adoption is probable.

**Decision-Unit Support.** The gamified TM was first pursued by the authors during the DigiEduHack2020 hackathon, during which various interviews and feedback sessions were held with top managers of large and medium tech enterprises and political leaders regarding the necessity, desired attributes and context of a viable TM. In table 1, the views of ten decision-units are presented.

**Table 1.** Political leadership and top managers view on TM

Expressed views over a TM	Politicians (N=4)	Managers (N=6)
<i>Offers a significant competitive advantage</i>	4	6
<i>Reduces resource misplacement</i>	4	5
<i>Desire to implement in under 4 years</i>	-	4
<i>Own solutions are on the works</i>	-	2

The feedback shows that there is a high confidence in that TMs represent a competitive advantage. 67% of interviewed managers are ready to adopt TMs.

In addition to these results, several studies and reports by lighthouse organisations point out the necessity of implementing internal solutions to the skill problems in the future of work [3, 4, 6, 23]. According to [24] 80% of executives believe that at least half of new roles should be filled internally instead of hiring externally. [7] argue that TM are a competitive advantage for organisations. The finish government has already implemented a TM to match their ministries' workers with opportunities [25].

**Positioning** When it comes to positioning the TM on the market, ease of use and customer's technological readiness are fundamental. Ease of integration with the present technologies and processes of customers is thus fundamental. Existing talent marketplaces are rigid when it comes to data integration, Tandem-ploy and Kalido.me both use partner businesses data storage and ETL, like SAP cloud integration, making it difficult to integrate sources outside of the partner ecosystem, *e.g.*, the customer uses SAP's ERP but BambooHR for HCM, difficulties can arise when the ETL process is exclusively performed by SAP cloud for a given TM. The proposed TM is ready for ETL and data warehousing from different sources.

Another advantage of the TM is the use of well-established collaborative platforms as integrated UI for end users. This increases TM adoption probability by end users, as it increases exposure, technological readiness, and ease of use.

Trust is a main factor in positioning, organisations only adopt technology if they trust its vendor, the same goes for the individual end user, whose willingness to adopt the TM depends on the trust they deposit into it. Creating trust in the vendor has to do with partnerships, customer support and marketing strategies (*e.g.*, branding), among other things, and exceeds the scope of this study. However, end user trust can be analysed. Granting end users access to their complete skill-profiles, individual analytics and relative position to their peers, not only increases end user awareness of their own abilities and progress, but also increases recommendations' explainability and trust in the application.

Privacy is also a major concern for a big part of end users, by explaining how data is stored and processed, fears can be mitigated.

## 5 Discussion and Conclusion

The TOE framework is considered superior for predicting organisational adoption of technology than diffusion of innovation (DOI) models, however, behavioural models, like the technology acceptance model (TAM) or theory of planned behaviour (TPB), are normally used to predict individuals technology adoption [9]. The point could be made that for the proposed framework to have validity in the context of TMs, behavioural aspects should be regarded, as information is biased by decision makers at investigated organisations.

Future research should apply concepts such as beliefs, intention, norms and expectations to better design user experience, especially when applying gamification. Still, TOE provides a complete approach for establishing perceived value of the TM design for organisations. Applying the BMC to sustain findings inside TOE, makes use of some behavioural studies. [26] even found evidence of TOE being consistent with applying behaviour models directly on decision makers.

The proposed TM presents significant advantages over existing talent marketplaces, mainly in expected increased serendipity performance of the RS in use, and integration with other services and data sources. Novel contributions to the field of talent management and talent marketplace design include the use of TOE and DSRM conceptual frameworks to design and evaluate the artefact, using RS technology that has not been previously used for TMs, and using collaborative platforms as end user UI to guarantee ease of use. When studying the impact of the proposed TM on knowledge intensive organisations, especially the technology sector, evidence for increased perceived value, organisational readiness and decision-maker support have been found.

TOE was applied to establish a conceptual framework for SaaS TM adoption to guide TM design, resulting into a value adding, explainable design. Finally, the DSRM was proposed as continuous evaluation framework for the proposed TM so that its application can iteratively be tested in the context of the knowledge base, guaranteeing the TM's quality during its entire lifecycle.

## References

1. E.J. Altman, J. Schwartz, D. Kiron, R. Jones, D. Kearns-Manolatos: Workforce Ecosystems: A New Strategic Approach to the Future of Work. MIT Sloan Management Review and Deloitte. (2021).
2. Stockton, H., Filipova, M., Monahan, K.: The evolution of work—seven new realities. Deloitte Insights. (2018)
3. Lau, Y.: How organisations can become project-based in the future of work. Forbes. (2021).
4. Sage-Gavin, E., Hines, K., Fuller, J.: Is the talent you need hiding in plain sight?. Accenture Strategy & Consulting. (2021)
5. Shahid, A.: The Employee Engagement Framework: High Impact Drivers and Outcomes. Journal of Management Research. 11, 45 (2019).

6. Sinyan, P., Nink, M.: How European Companies Can Fix Their Workplaces. *Workplace*. (2021)
7. Field, E., Hancock, B., Schaninger, B.: Stave off attrition with an internal talent marketplace. [Podcast] McKinsey. (2022).
8. Tornatzky, L., Fleischer, M.: *The processes of technological innovation*. Lexington Books, Lexington MA (1990).
9. Valdebenito, J., Quelopana, A.: Conceptual Model for Software as a Service (SaaS) Enterprise Resource Planning (ERP) Systems Adoption in Small and Medium Sized Enterprises (SMEs) Using the Technology-Organisation-Environment (T-O-E) Framework. *Advances in Intelligent Systems and Computing*. 918, 143–152 (2019).
10. Osterwalder, A., Pigneur, Y.: *Business model generation: a handbook for visionaries, game changers, and challengers*. Wiley, Hoboken, NJ (2010).
11. Hevner, A., vom Brocke, J., Maedche, A.: Roles of Digital Innovation in Design Science Research. *Business and Information Systems Engineering*. 61, 3–8 (2019).
12. Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., Bragge, J.: *The Design Science Research Process: A Model for Producing and Presenting Information Systems Research*. 1st International Conference, DESRIST 2006 Proceedings. (2006)
13. Vargas, S., Castells, P.: Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. *RecSys'11*. Chicago, USA (2011).
14. Maurya, A., Telang, R.: Bayesian Multi-View Models for Member-Job Matching and Personalized Skill Recommendations. *2017 IEEE International Conference on Big Data (BIGDATA)*. IEEE, Boston, MA, USA (2017).
15. Sieg, A., Mobasher, B., Burke, R.: Improving the effectiveness of collaborative recommendation with ontology-based user profiles. *HetRec 2010, 4th ACM Conference on Recommender Systems, RecSys 2010*. 39–46 (2010).
16. Yang, S., Korayem, M., AlJadda, K., Grainger, T., Natarajan, S.: Combining content-based and collaborative filtering for job recommendation system. *Knowl Based Syst*. 136, 37–45 (2017).
17. Urdaneta-ponte, M.C., Méndez-zorrilla, A., Oleagordia-ruiz, I.: Lifelong learning courses recommendation system to improve professional skills using ontology and machine learning. *Applied Sciences (Switzerland)*. 11, (2021).
18. Nilashi, M., Ibrahim, O., Bagherifard, K.: A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Syst Appl*. 92, 507–520 (2018).
19. Gerdenitsch, C., Sellitsch, D., Besser, M., Burger, S., Stegmann, C., Tscheligi, M., Kriglstein, S.: Work gamification: Effects on enjoyment, productivity and the role of leadership. *Electron Commer Res Appl*. 43, (2020).
20. Thongmak, M.: A model for enhancing employees' lifelong learning intention online. *Learn Motiv*. 75, (2021).
21. Barr, H.M., Smitherman, R.C., Mesmer, B., Weger, K., Bossuyt, D. van, Semmens, R., Tenhundfeld, N.L.: Use, Acceptance, and Adoption of Automated Systems with Intrinsic and Extrinsic Motivation Based Incentive Mechanisms. *2022 Systems and Information Engineering Design Symposium, SIEDS 2022*. 298–303 (2022).
22. Kahn, W.A.: Psychological Conditions of Personal Engagement and Disengagement at Work. *Academy of Management Journal* **33(4)**, 692–724 (2017)
23. Deloitte: Are you overlooking your greatest source of talent? *Deloitte Insights*. *Deloitte Review*, 36–42 (2018).
24. Ellingrud, K., Gupta, R., Salguero, J.: Building the vital skills for the future of work in operations. *McKinsey & Company* (2020).

25. Ketamo, H., Passi-Rauste, A., Vesterbacka, P., Vahtivuori-Hänninen, S.: Accelerating the Nation: Applying AI to Scout Individual and Organisational Human Capital. International Conference on Innovation and Entrepreneurship. (2018).
26. Li, J.C.F.: Roles of individual perception in technology adoption at organisation level: Behavioral model versus toe framework. Journal of System and Management Sciences. 10, 97–118 (2020).

# Eye Controlled Virtual Keyboard for Motor Impaired Users

Aya Marshaha<sup>1</sup>, Daniela Oliveira<sup>1</sup>, Teresa Ferraz<sup>1</sup>, Tomas Sousa<sup>1</sup>,  
Luis Coelho<sup>1,2</sup> [0000-0002-5673-7306] and Sara Reis<sup>1,2</sup>

<sup>1</sup> Physics Department, Polytechnic of Porto - School of Engineering, Porto, Portugal

<sup>2</sup> Center for Engineering Innovation and Industrial Technology – ISEP, Porto, Portugal  
{1191788, 1170386, 1161323, 1161566, lfc, ssr}@isep.ipp.pt

**Abstract.** Computers are currently an essential tool in human life and human-interaction devices must be inclusive. Motor disabled people might require specific strategies, making eye-controlled screens a useful asset to increase their quality of life. In this paper we present an effective and simple to use virtual computer keyboard that makes writing using only the eye gaze and blinking possible. The obtained results were very good, since a high accuracy rate was achieved, and promising, since all test users were able to quickly learn to use this tool. Nevertheless, some challenges were identified, mostly caused by the quality and resolution of the cameras. Lighting, structure of the eyes and shape of the face are also conditions that were identified to impact the performance of the keyboard. All of these can be effectively eliminated by performing adjustments in the parameters of the supporting algorithms. Automating this process is envisioned as future work, allowing an easy adaptation of the system to the environmental conditions and to the user's personal anatomical features.

**Keywords:** Virtual keyboard, human-computer interaction, gaze, eye.

## 1 Introduction

Recently, due to the fast improvement in technology there has been a considerable application of human computer interfaces. A lot of systems had been advanced for normal people, who have the normal ability to do volunteer actions. But there was an urgency to develop these kinds of systems principally for the people who are only able to perform simple voluntary actions, as happens with severe spinal cord injuries. Often, one of the only actions that these disabled people can perform voluntarily is the blinking of their eyes, making the use of computers a challenging task.

In the digital world, interaction with multimedia information occupies a large part of everyday activities, be it leisure or work, involving various operations that are performed by devices such as the computer mouse or keyboard [1]. For people with limb disabilities and senior citizens, computers are an essential tool for communication, education and entertainment, however, computer access can be hampered by a person's disability. Though speech is the best interface for assistive technologies [2] it is

not always feasible. Often, disabled users use a mouse-like device or a joystick, that are operated with their heads, to interact with a computer. Typing is achieved with the help of an on-screen keyboard, but many times the placement of this mouse/joysticks is considered unpleasant and uncomfortable by the user [3]. However human-machine interfaces can be made simpler using recent technologies. The design of modern interfaces and the diverse types of interaction controls that compose them have been studied in recent years using the principles of human information processing and universal design [4]. When interacting with multimedia devices, the eyes process a lot of information, quickly and effectively. On the other hand, their activity can be detected, with eye monitoring being successfully used in the analysis and assessment of cognition during multimedia learning [1].

The human eye is constituted by several structures: the pupil, which allows light to enter the eye and has a dark hue because of the absorbing pigment in the retina; the iris, which is pigmented and coloured to the eye, and in the center of the opening is the pupil, which is controlled by the circular muscle of the iris to allow more or less light to enter the eye; the cornea is a transparent outer surface that covers the pupil, the iris, and is a lens that allows a sharp image to be produced at the level of retinal photoreceptors; and the sclera, which is part of the support wall of the eyeball and continues with the cornea, making a connection with the dura mater of the central nervous system [4]. Each of these structures have specific anatomical characteristics allowing to be easily detected, segmented, and processed, using hardware and software computer vision tools.

As an input device, gaze estimation systems provide information about a person's eye locations and movements so that the point of view can be established at any time. Several different invasive or non-invasive methods for eye movement measurement have been studied to aid in improving the estimation of eye data. Eye monitoring technology has evolved, increasing in accuracy, and lowering in costs. However, using eye gaze as input remains a challenge due to limited viewing angle, limited camera resolution, challenging illumination conditions, calibration errors, drift and inherent eye instability, and the eye positions reported by detectors do not always accurately match the point at which user is looking at. When considering eye-based computer input strategies we can identify software only solutions but also hardware-based possibilities. Additionally, it is also important to explore and identify the best eye-based events that can be used to trigger actions on the target device.

When querying scientific repositories with keywords such as “gaze estimation” or “eye tracking” combined with “disability” or “interaction”, we can observe that the interest in this topic has been raising in the last five years, covering innovative devices, approaches, purposes and interaction strategies. For example, Marcel Tresanchez et al. [5] presented an application where an optical mouse sensor is repurposed as an infrared vision system that is able to monitor eye pupil's displacement and detect eye blinking events. It uses a human interface device (HID) standard class of the universal serial bus (USB) increasing its compatibility with most computer platforms without additional software. Nevertheless, being a head-mounted device, the discomfort created during its usage can limit its adoption. Software based approaches, can often provide a better user experience. Hari Singh et al. presented an object acquisition and

selection approach in human computer interaction systems where the objects placed over computer screen are automatically scanned and the user performs voluntary eye blinks for object selection when the focus comes over the object of interest. Scanning means moving the focus over objects placed on the computer screen one by one and the scanning time is the time taken to move focus from one object to the next object. The user is not required to perform any physical movement, the moving part is only the eyelids. Eye blink is used to perform mouse analogous operations [6]. Yongqiang Liang et al. presented a method of interactive state classification and intention prediction based on eye tracking, where five states of interaction are defined. These are monitoring state, tracking state, decision state, burst state and off-loop state, design induced experiments were conducted to collect eye movement data in each state, a single factor analysis of variance shows that the seven eye movements in the five interactive states has a significant difference. This project was directed to provide intelligent user interface state awareness and intention prediction, but not interaction. [7]. Concerning pupil segmentation strategies, besides traditional image processing pipelines, deep-learning based strategies are also emerging. In Akinlar et al. a fully convolutional neural network has been used to perform semantic segmentation and accurately extract pupil contour using ellipse shaped priors [8].

In this paper we present a software-based approach for a virtual eye-controlled keyboard. Using the laptop's webcam, through an easy-to-use and learn interface, the user should be able to perform all the functions, as in a traditional physical keyboard, but using the eyes movements and the eye blinking actions.

## 2 Development

The system operation was divided in three main phases, being the first the segmentation of the face and eyes, the second covering the detection of events, and finally the generation of keyboard like outputs. The main function blocks that compose the proposed virtual keyboard system are represented in Fig. 1.

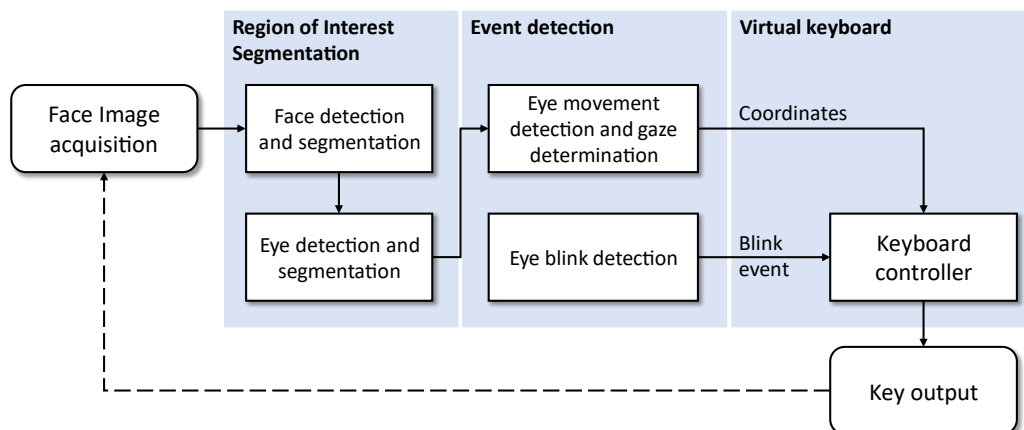


Fig. 1. Operational pipeline for the virtual eye-controlled keyboard.

The process starts with the image acquisition of the user's face. To enable the use of the virtual keyboard by most users we have decided to use simple low-cost hardware, as is the existing webcam on most laptop computers. To ensure a basic performance of the system, a minimum resolution of 640x480, and an acquisition rate of 30 frames per second, are required. Higher specs can boost the quality of the results, either in location precision but also on the smoothness of eye-gaze tracking. On the software side, the OpenCV library allows to control the image acquisition process and to capture sequence of images. For each image frame, the regions of interest, face and eyes, are detected and the key points are then segmented. To detect the face, a model that identifies 68 different points in the face was used. This model is able to simultaneously identify the location of the face and to accurately provide coordinates for several anatomical landmarks, being twelve related with the eyes (six for each eye). All landmarks, or coordinate points, have a specific immutable index, which allows to filter only the eyes related information. With this, it is possible to perform the eye segmentation (with a convex hull on the contour points) and to measure the eyelid position, as shown in Fig. 2. By creating lines that cross the eye horizontally and vertically, it is also possible to estimate the eyes perspective in relation to the camera. When the vertical line crosses the horizontal line near the middle the user will be facing the screen directly, any deviations may point to a movement of the head.

The eye gaze detection started by isolating the eyes from the rest of the face using a mask to extract exactly the eyes and exclude the surroundings. Using the landmark points, we were able to successfully create the eye mask, as in Fig. 2b. With the eye isolated from the rest of the face, the image was converted to grayscale and then a binary image of the eye was created. With this, it was possible to distinguish between the pupil and the sclera. The idea is to divide the eye image in two halves and for each half, calculate the number of white pixels that are visible (visible sclera). Since our image was mirrored, when the sclera is more visible in the right part it means the eye is looking left and vice versa. The ratio between the number of white pixels present in each side of the eye is possible to estimate the displacement.

The same procedure is used to estimate the vertical displacement. This method allows to use only one of the eyes, because typically the eyes both look at the same direction, however, detecting the gaze with both eyes showed to be more precise. In Fig. 2c, we can observe the image binarization with a threshold of 130 (for the specific lighting of the room) where the pupil and the sclera are completely distinguishable. To estimate the relation between eye displacement and screen coordinates an initial calibration step is required. In this stage, the user is asked to sequentially look to five different locations on the screen. Initially a circle at the screen center and then, to quarter-circles located at screen corners. These locations define the eye displacement limits on the screen.



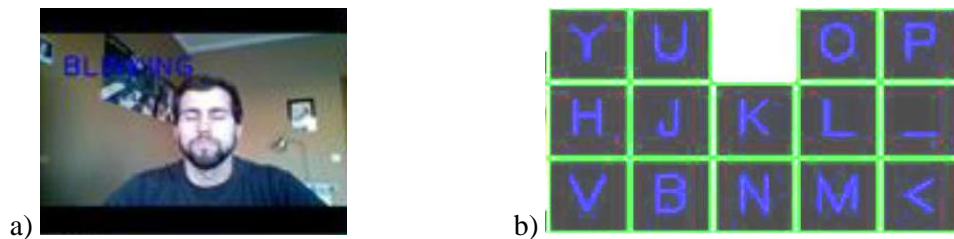
**Fig. 2.** Examples of eye image with height and width estimation (left), eye segmentation mask (center), and gaze estimation (right) cases. (Different scaling levels are used for the sake of easier process explanation.)

The next step was to detect the eye blinking event. There are several ways to detect the blinking, but since the vertical line on the eye decreases in length when the eye closes, so the ratio of the horizontal and vertical line was calculated for each frame and when it goes below a certain value it means the eye is closed and that there was a blinking motion. The landmark coordinates are extremely useful on these calculations since they allow to detect geometrical interactions easier. The click event was achieved when the eyes' closed event was successfully detected for at least one continuous second.

The last step of this work was to design a user-friendly virtual keyboard. To reduce the number of key possibilities, prone to pointing accuracy errors, the keyboard was divided in two different sides, with the left side contains all the letters on the left side of a “qwerty” keyboard and vice-versa for the right side. The keys are rectangles that divide a black window, with calculated dimensions, and each rectangle has a letter drawn inside of it. Each 10 frames one of the keys will become white (active) and when the eyes are closed for more than 20/30 frames (approximately 1 second) the active key is printed to the white board. This eyes-closed interval is needed, since humans need to blink often (to lubricate the eyes) and if with every blink a key was pressed the keyboard would be useless. After the key press is given, the user has the the option of selecting which side of the keyboard he wants to use and, the user must look left, if he wants the left part of the keyboard and the opposite, for the right part.

### 3 Results and discussion

The developed system was tested using the virtual keyboard and a “WordPad” window in which the user, by eye movement and blinking, could write text on. The keyboard allows the user to write every letter of the English “qwerty” keyboard and give spaces between words. As an example, an eye closing event is shown in Fig. 3a, with a highlighted letter (that is being printed) in the keyboard, in Fig. 3b, in this case, the “I” letter. If the users close the eye for longer than one second, the letter will be printed to the WordPad application and the keyboard side selection menu will pop up again. The process is the same until the user finishes his writing. Whenever the user wants to “press” a key, his eyes are closed so, to help him detect if he pressed the letter or no, a confirming sound is reproduced.



**Fig. 3.** Example of a simple eye blinking event of both eyes, representing the selection of a virtual keyboard key (left), and the representation of half virtual keyboard (right).

The system was evaluated by four persons (without disability), all in similar conditions, following the same predetermined text. An accuracy of 98.8% for a text composed by 212 characters. These preliminary results are promising and point to an effective usage of the keyboard. However, there are some limitations and problems that need to be solved to enhance the keyboard performance. The eye detection process is accurate and its robustness, supported by the 68 landmarks face model, helped to achieve the current results. However, the system showed to have some difficulties with landmark identification when the eyes were closed. This is an expected behavior since the eye gaze coordinates, that are pupil based, cannot be tracked for closed eyes. Additionally, the gaze detection accuracy suffers when the user is changed, and the calibration is not performed. This initial user dependent adjustment is mandatory, to calculate the specific ratios that are used to identify where the gaze is. Part of this problem comes from the method used to identify the direction of the eye-gaze or from the varying quantity of the visible sclera. On the other side, hardware limitations also impacted the system's performance. Image acquisition in different light conditions was challenging but the low resolution of the image sourcing webcam showed to be critical for the success of the segmentation algorithm.

Using the proposed virtual keyboard may require a longer writing time because blinking movements must be trained to be repeatedly fast. For this reason, the use of a word ending recommender system or even a grammar capable of suggesting next words could represent a good improvement of the "typing" speed.

## 4 Conclusion

This project consisted in the development of a tool for motor disabled people to be able to write without the use of their limbs, only being needed the eyes to write on a screen. The proposed system showed to be easy to use and with a short learning curve. The obtained results are preliminary, but a very high accuracy was obtained.

Future work consists in reducing the amount of time required to write. Introducing faster typing strategies, as recommender system, or to increase the number of available keys on the screen, with the use of higher resolution cameras. Testing the keyboard with the people it was designed for, since usage tests with real users can provide important feedback that is needed for the improvement of the keyboard. The combination of the keyboard with other tools, such as smartphones, computers, wheelchairs, or others, could increase its value and usefulness.

## References

1. Kumar, C., Menges, R., Staab, S.: Eye-Controlled Interfaces for Multimedia Interaction. *IEEE MultiMedia*. 23, 6–13 (2016). <https://doi.org/10.1109/MMUL.2016.52>.
2. Calado, A., Coelho, L.: Speech as the basic interface for assistive technology. *DSAI 2009- Proceedings of the 2th International Conference on Software Development for Enhancing Accessibility and Fighting Info-Exclusion*. (2009).
3. Morris, T., Blenkhorn, P., Zaidi, F.: Blink detection for real-time eye tracking. *Journal of Network and Computer Applications*. 25, 129–143 (2002). <https://doi.org/10.1006/jnca.2002.0130>.
4. Diego-Mas, J.A., Garzon-Leal, D., Poveda-Bautista, R., Alcaide-Marzal, J.: User-interfaces layout optimization using eye-tracking, mouse movements and genetic algorithms. *Applied Ergonomics*. 78, 197–209 (2019). <https://doi.org/10.1016/j.apergo.2019.03.004>.
5. Tresanchez, M., Palacín, J., Tomàs Pallejà: Optical Mouse Sensor for Eye Blink Detection and Pupil Tracking: Application in a Low-Cost Eye-Controlled Pointing Device. *Journal of Sensors*. 2019, 19 (2019). <https://doi.org/10.1155/2019/3931713>.
6. Singh, H., Singh, J.: Object acquisition and selection using automatic scanning and eye blinks in an HCI system. *J Multimodal User Interfaces*. 13, 405–417 (2019). <https://doi.org/10.1007/s12193-019-00303-0>.
7. Liang, Y., Wang, W., Qu, J., Yang, J.: Application of Eye Tracking in Intelligent User Interface. *J. Phys.: Conf. Ser.* 1169, 012040 (2019). <https://doi.org/10.1088/1742-6596/1169/1/012040>.
8. Akinlar, C., Kucukkartal, H.K., Topal, C.: Accurate CNN-based pupil segmentation with an ellipse fit error regularization term. *Expert Systems with Applications*. 188, 116004 (2022). <https://doi.org/10.1016/j.eswa.2021.116004>.

---

# FARO InTouch Connect: Desenvolvimento de um Gestor de Metadados\*

Afonso Correia e Alexandre Gouveia

Instituto Superior de Engenharia, Politécnico do Porto  
{1190329,aas}@isep.ipp.pt

**Resumo** O projeto descrito neste artigo resultou da necessidade dos colaboradores da Faro Technologies desenvolverem um portal para gestão dos metadados das aplicações por si distribuídas. Este portal permitiria simplificar o processo de criação de aplicações e obtenção de metadados, contribuindo desse modo para aumentar a eficácia das equipas. Os objetivos do projeto consistiram na criação da *landing page*, bem como na criação de mecanismos de autenticação, gestão de fontes de texto e gestão de requisitos. Neste artigo descreve-se a análise do problema e o desenho da sua solução, no qual foram utilizadas as *frameworks* Blazor Server e .NET. Realizou-se uma investigação intrínseca sobre padrões arquiteturais e arquiteturas de software, tendo sido escolhida a adoção de MVVM (Model-View-ViewModel), Clean Architecture e CQRS (Command Query Responsibility Segregation). A solução foi implementada, tendo sido concretizado com sucesso a generalidade dos objetivos estabelecidos. O produto obtido está atualmente a ser utilizado pela Faro, contribuindo para a melhoria dos seus processos.

**Palavras-chave:** Gestor de metadados, Autenticação, Fontes de texto.

## 1 Introdução

Devido a obrigações legais relacionadas com a cidade portuguesa, a Faro Technologies Inc. opera em Portugal segundo o nome Antares, Desenvolvimento de Software, Lda. Por questão de praticidade, sempre que necessário fazer referência à empresa, esta será identificada como Faro.

A Faro Technologies é uma empresa industrial americana, líder na comercialização de hardware portátil de medição 3D. Para além disso, a empresa também desenvolve software, nomeadamente sistemas CAM (Computer Aided Manufacturing).

De forma a facilitar a distribuição das aplicações da Faro para os seus clientes, bem como a atualização para novas versões, começou a ser desenvolvida a Faro InTouch, uma aplicação distribuidora de software. Esta aplicação apresenta um catálogo de aplicações desenvolvidas pela Faro disponíveis para os seus utilizadores.

---

\* Apoiado por Antares, Desenvolvimento de Software, Lda.

O projeto em questão, Faro InTouch Connect, pretende a criação de uma aplicação para servir a Faro InTouch. Cada aplicação distribuída pela Faro InTouch está associada a um conjunto de parâmetros, tal como o seu nome, versões, pré- e pós-requisitos necessários, etc. Estes parâmetros são designados por metadados. A Faro InTouch Connect é um portal que possibilita a gestão destes metadados para as diferentes aplicações. Para além disso, facilita a criação de novas aplicações, pois será possível definir os metadados necessários diretamente a partir do portal Faro InTouch Connect.

### 1.1 Problema

Os metadados das aplicações disponibilizadas pelo Faro InTouch são definidos num ficheiro json, onde é feita a sua manutenção. Esta situação não é a ideal, pois é pouco prática, intuitiva e propensa a erros. Para além disso, força os desenvolvedores a criarem um ficheiro com toda a informação necessária para cada nova aplicação desenvolvida.

Este problema serviu de motivação para a criação de uma solução – a Faro InTouch Connect. Devido à dimensão do projeto em mãos, foram definidos objetivos técnicos a cumprir, relacionados com os módulos a desenvolver, e que se apresentam de seguida:

- Criação de mecanismos de autenticação;
- Criação da landing page da aplicação;
- Criação de mecanismos de gestão de fontes de texto;
- Criação de mecanismos de gestão de requisitos.

Fontes de texto referem-se às fontes tipográficas utilizadas pelas demais aplicações integradas na Faro InTouch. Requisitos entendem-se como executáveis necessários para utilizar uma aplicação. Podem ser pré-requisitos se necessários antes da instalação da aplicação, ou pós-requisitos se necessários após a sua instalação.

A criação do portal Faro InTouch Connect possibilita a gestão dos metadados das aplicações da Faro e facilita a criação de novas aplicações. O desenvolvimento desta nova ferramenta beneficia as equipas da Faro, permitindo uma maior eficiência na sua atividade.

## 2 Estado da Arte

O projeto descrito neste artigo assenta num conjunto de tecnologias e padrões arquiteturais relacionados com a implementação de uma solução web. Esta secção aborda sucintamente os três padrões utilizados, que compõem o cerne arquitetural do sistema desenvolvido.

O desenvolvimento do *frontend* seguiu o padrão arquitetural MVVM [1,2,3]. Este padrão define as camadas Model, View e ViewModel. A camada Model representa os conceitos do modelo de domínio. Esta camada é responsável por modelar os objetos e entidades de negócio para a aplicação. A View é a interface

do utilizador. É composta por toda a estrutura e componentes que aparecem no ecrã. É responsabilidade da View representar a informação do Model e capturar os *inputs* do utilizador. O ViewModel é a camada de lógica da View. Esta camada prepara os dados do Model, de forma a abstrair a View da informação de negócio.

A Clean Architecture [4,5] resulta da tentativa em reunir numa única arquitetura, as características principais das arquiteturas mais populares, produzindo sistemas independentes de *frameworks*, UI (User Interface), bases de dados, agentes externos e facilmente testáveis. Esta foi a arquitetura usada no desenvolvimento do *backend*. A principal regra desta arquitetura define o fluxo de dependências, ou seja, cada camada apenas pode manter dependências com camadas interiores. Da camada mais interior para a mais exterior, esta arquitetura define as camadas: (i) *Entities*, que encapsula as regras de negócio relacionadas com a organização; (ii) *Use Cases*, que encapsula as regras de negócio relacionadas com a aplicação; (iii) *Interface Adapters*, que é composta por um conjunto de adaptadores de dados; e (iv) *Frameworks and Drivers*, que é composta pelas ferramentas e detalhes utilizados na implementação.

CQRS [6,7] é um padrão de desenvolvimento de software que pretende separar as operações de leitura e de escrita na base de dados de uma aplicação. Commands são utilizados para operações de escrita/atualização de dados, enquanto que Queries são usadas para operações de leitura de dados. É possível manter os dados numa só base de dados e isolar as operações usando modelos de dados diferentes para escrita e leitura, ou separar fisicamente os dados em bases de dados diferentes.

### 2.1 Trabalhos relacionados

Identificaram-se duas aplicações que permitem a gestão de metadados – Select Star e Metadata Manager.

A Select Star [8] é uma plataforma de gestão de metadados que obtém informação através de bases de dados, logs e histórico de pesquisas, analisa os dados adquiridos e armazena-os em tabelas. Uma das funcionalidades que expõe é a gestão dos dados armazenados através de um portal.

O Metadata Manager [9] é um portal web para gestão e processamento de metadados. O fluxo de execução da aplicação começa por extrair os metadados das fontes selecionadas, modelar os metadados para os modelos associados e carregar a informação para armazéns de dados. Após o carregamento dos dados, é possível navegar, procurar e editar os metadados guardados.

Ambas as soluções apresentadas seriam viáveis. Como principal vantagem da sua adoção pode referir-se a poupança de recursos monetários e temporais, a curto prazo, uma vez que evitar-se-ia o desenvolvimento do sistema proposto. No entanto, os encargos com o pagamento da subscrição mensal destas aplicações, a exposição dos metadados das aplicações da Faro a estas empresas e a dependência externa para estas plataformas foram fatores decisivos para a prossecução deste projeto.

### 3 Análise do problema e desenho da solução

A partir da compreensão do problema identificam-se as entidades e os objetos do domínio a ser modelado. O domínio é dividido em agregados que se relacionam com os módulos de implementação do sistema. A Figura 1 ilustra a modelação obtida.

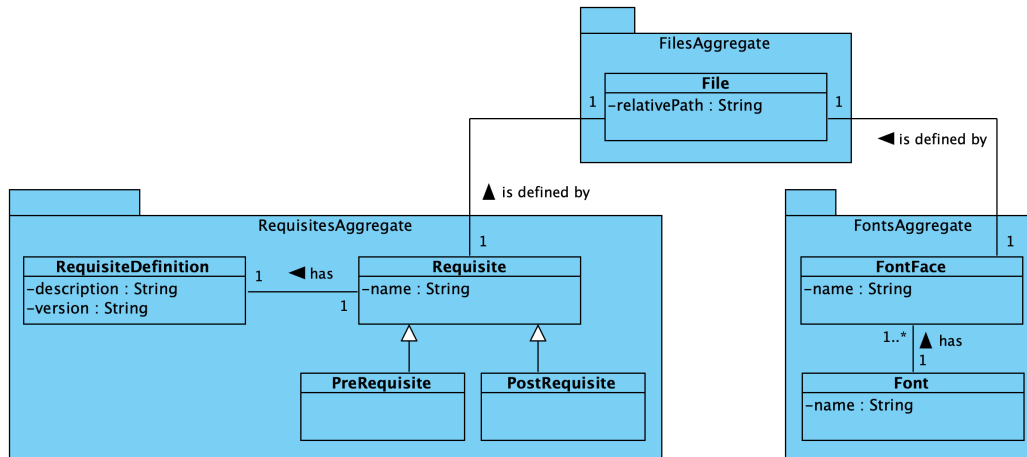


Fig. 1. Modelo de domínio simplificado

*Font* representa uma fonte de texto. Uma *Font* pode ter várias *FontFace*, representantes das faces/estilos que uma fonte de texto pode tomar (e.g. *Roboto* é uma *Font* e *Roboto Italic* uma *FontFace*).

*Requisite* representa um executável necessário para a utilização de uma aplicação (e.g. JRE – Java Runtime Environment – é um requisito para aplicações Java) e é considerado um pré- ou pós-requisito, dependendo do momento de execução relativamente à instalação da aplicação. Um *Requisite* define a sua informação técnica através do *RequisiteDefinition*.

Tanto *FontFace* como *Requisite* são definidos através de *File*, usado para representar um ficheiro.

#### 3.1 Requisitos funcionais

Os requisitos funcionais são apresentados segundo os casos de uso da solução:

- UC1: Authenticate
- UC2: Unauthenticate
- UC3: Display available fonts
- UC4: Display available faces by font
- UC5: Add faces
- UC6: Filter fonts by name
- UC7: Display requisites
- UC8: Display requisite details
- UC9: Add requisite
- UC10: Edit requisite
- UC11: Duplicate requisite

### 3.2 Outros requisitos funcionais e de qualidade

Requisitos de qualidade definem propriedades e restrições do sistema. Para a especificação destes requisitos foi usado o modelo FURPS+. Apresentam-se de seguida outros requisitos funcionais e de qualidade identificados.

**Funcionalidade** O uso da aplicação está limitado aos colaboradores da organização, sendo necessária autenticação para acesso ao seu conteúdo.

**Usabilidade** A interface gráfica deve ser dotada de ajudas visuais, como ícones ou dicas.

**Confiabilidade** Os dados apresentados após recarregamento da página web devem estar sempre em conformidade com a informação presente base de dados.

**Performance** As chamadas externas para obtenção de dados não devem demorar mais de 1 minuto.

**Suportabilidade** A aplicação deve estar preparada para correr em qualquer browser moderno.

**Restrições** “Clean Architecture” deve ser adotada como arquitetura de software para o *backend*.

### 3.3 Design arquitetural

O desenho da solução encontra-se devidamente detalhado e documentado, através de diferentes vistas (lógica, física, de processos e de implementação) e níveis de granularidade. Como exemplo apresenta-se na Figura 2 uma vista lógica, que representa a organização dos componentes da solução.

## 4 Implementação da solução

No desenvolvimento do *frontend* utilizou-se a *framework* Blazor Server. O *backend* foi implementado em .NET.

Foram definidos os seguintes ambientes de desenvolvimento, cada um com um propósito e configuração própria:

- *Local*, ambiente usado para o desenvolvimento local de uma versão da aplicação;
- *Development*, ambiente usado para o desenvolvimento de uma imagem da aplicação implantada;
- *Staging*, ambiente usado para a realização de testes de software e controlo de qualidade;
- *Production*, ambiente que representa o estágio final de uma instância da aplicação.

Cada versão da aplicação é implantada nos ambientes *Development*, *Staging* e *Production*, sendo possível aceder à versão mais recente destes ambientes através do URL (Uniform Resource Locator) correspondente.

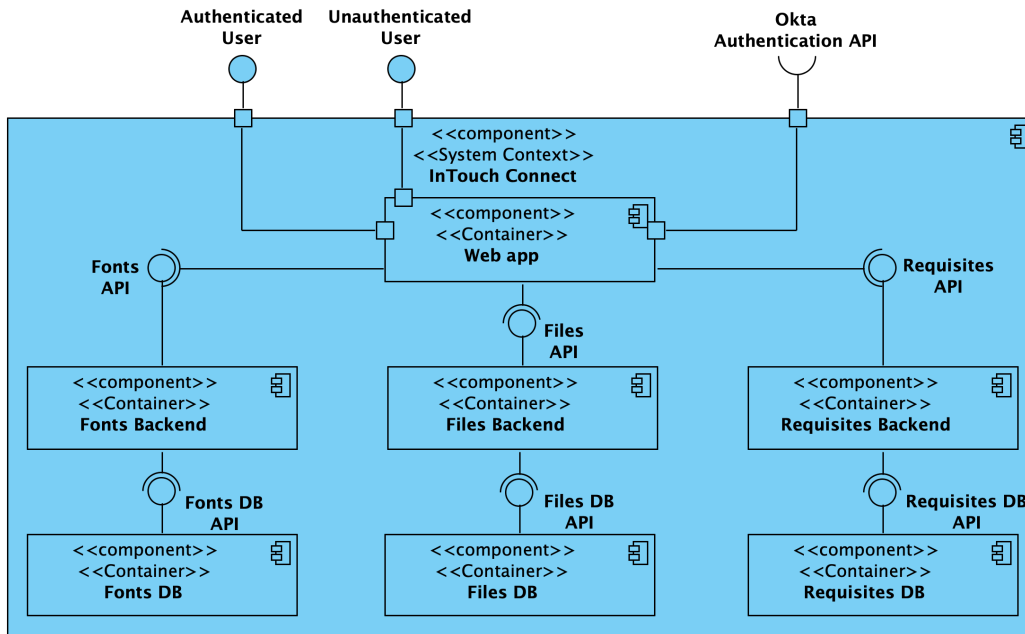


Fig. 2. Vista l3gica de n3vel 2 da solu33o

#### 4.1 Componentes

Cada componente da solu33o seguiu um processo pr3prio de implementa33o e, quando poss3vel, abstra33o, seja por classes base ou interfaces. Esta sec33o descreve os componentes utilizados.

**View** Representa a interface com o utilizador, dividindo-se em layouts, p3ginas e componentes.

**ViewModel** Possui as propriedades que a View requer, bem como toda a l3gica associada.

**StateProvider** Faz a gest3o do estado da sess3o e disponibiliza-a ao resto da aplica33o.

**Service** Realiza opera33es segundo as regras de neg3cio do sistema.

**Model** Representa o modelo de dom3nio.

**Controller** Controlam o fluxo de execu33o.

**Command** Executam opera33es de escrita na base de dados.

**Query** Executam opera33es de leitura da base de dados.

**QueryHandler** Respons3vel por criar a query a ser interpretada pela base de dados.

**Repository** Implementa a camada de intera33o com as funcionalidades da base de dados.

O fluxo normal de uma funcionalidade segue os seguintes passos:

1. Comunica33o entre o *browser* e a View/ViewModel correspondente;
2. Utiliza33o do Model para modelar o (poss3vel) *input* do utilizador;

3. Consulta do estado da sessão através do `stateProvider`;
4. Chamada ao *backend* através do `Service`. As API (Application Programming Interface) foram expostas através de *NuGet Packages*;
5. Receção do pedido no `Controller`;
6. Criação do `Command` ou `Query` correspondente;
7. Consulta (se necessário) das regras internas no `Model` do *backend*;
8. Comunicação com a base de dados através do `Repository` ou `QueryHandler`.

### 4.2 Módulos

O módulo de criação de mecanismos de autenticação envolveu a criação e integração de uma Okta app com a solução, estabelecendo autenticação através do protocolo OIDC (Open ID Connect). Uma Okta app é uma aplicação criada e disponibilizada através do sistema de autenticação Okta, que permite gerir acessos e permissões a um servidor de autenticação/autorização. O protocolo OIDC consiste numa camada de identidade definido por cima do protocolo OAuth 2.0, que permite verificar a identidade de utilizadores através do processo de autenticação realizado num servidor de autorização. Adicionalmente, foram também desenvolvidos os casos de uso UC1 e UC2, relacionados com o login e logout da aplicação.

O módulo de criação da *landing page* não esteve relacionado com nenhum caso do uso, envolvendo apenas a tarefa de criação de uma vista navegável que possibilitasse o começo das restantes funcionalidades.

O módulo de gestão de fontes envolveu o desenvolvimento dos casos de uso UC3, UC4, UC5 e UC6. O *backend* responsável pela gestão de fontes já estava construído, pelo que não fazia parte dos requisitos do projeto. No entanto, no *frontend* estes casos de uso foram completamente implementados.

O módulo de gestão de requisitos envolveu o desenvolvimento dos casos de uso UC7, UC8, UC9, UC10 e UC11, tanto na vertente de *frontend* como de *backend*.

### 4.3 Testes

A avaliação do *frontend* consistiu na realização de testes funcionais, utilizando uma grande variedade de *inputs* para uma correta e completa validação das funcionalidades.

Para a avaliação do *backend* realizaram-se testes unitários, de forma a validar o comportamento de cada unidade de software, bem como de testes de integração, de forma a validar a integração das diferentes unidades de software.

## 5 Conclusão

A Tabela 1 mostra o grau de realização por objetivo. É ainda relevante referir o facto de que, no módulo de gestão de requisitos, o único caso de uso parcialmente desenvolvido foi o UC7 Display Requisites, com um grau de realização de 20%.

**Tabela 1.** Grau de realização por objetivo

Objetivo	Grau de realização
Criação de mecanismos de autenticação	100%
Criação da landing page da aplicação	100%
Criação de mecanismos de gestão de fontes de texto	100%
Criação de mecanismos de gestão de requisitos	10%

É possível atribuir um grau de realização de 100% a todos os atributos de qualidade identificados pelo modelo FURPS+, com exceção do atributo de usabilidade, que possui um grau de realização de apenas 70%.

O trabalho desenvolvido foi sempre sustentado em elevados padrões de qualidade, passando pela pesquisa e análise de tecnologias e padrões arquiteturais, desenho do sistema com diferentes níveis de granularidade e implementação cuidada da solução. O progresso no desenvolvimento foi validado e revisto de forma a cumprir com a exigência imposta pela empresa. Desta forma, foi possível criar uma aplicação que é atualmente utilizada pela Faro, permitindo uma melhoria dos seus processos.

Como trabalho futuro perspectiva-se a conclusão do módulo de gestão de requisitos. Refira-se ainda que a aplicação está preparada para se iniciar o desenvolvimento de novos módulos, a qualquer altura.

## Referências

1. J. Gossman, “Tales from the Smart Client: Advantages and disadvantages of M-V-VM” (2006). <https://docs.microsoft.com/en-gb/archive/blogs/johngossman/advantages-and-disadvantages-of-m-v-vm>
2. “The MVVM Pattern” (2012). <https://msdn.microsoft.com/en-us/library/hh848246.aspx>
3. V. Setia, “MVVM Architecture” (2020). <https://www.c-sharpcorner.com/article/mvvm-architecture>
4. R. C. Martin, “The Clean Architecture” (2012). <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
5. Barath, “The Clean Architecture — Beginner’s Guide” (2022). <https://betterprogramming.pub/the-clean-architecture-beginners-guide-e4b7058c1165>
6. M. Fowler, “CQRS” (2011). <https://martinfowler.com/bliki/CQRS.html>
7. “Padrão CQRS”. <https://docs.microsoft.com/pt-pt/azure/architecture/patterns/cqrs>
8. “Select Star”. <https://blog.selectstar.com>
9. “Metadata Manager”. <https://docs.informatica.com/data-integration/metadata-manager>

# Ferramenta de ensino/aprendizagem do método de Monge

João Romano Barbosa<sup>1</sup> and João Paulo Pereira<sup>2</sup>

<sup>1</sup> Institute of Engineering of Porto – Polytechnic of Porto (ISEP/P.PORTO)  
1190751@isep.ipp.pt

<sup>2</sup> Institute of Engineering of Porto – Polytechnic of Porto (ISEP/P.PORTO)  
Interdisciplinary Studies Research Center (ISRC)  
jjp@isep.ipp.pt

**Resumo** Ao longo das últimas décadas, a presença tecnológica tem vindo a aumentar na área educativa originando novas aplicações e plataformas digitais de suporte ao ensino. Esta mudança tecnológica eleva o aproveitamento dos estudantes e promove um melhor desempenho escolar. Este artigo tem como objetivo explorar uma nova ferramenta de apoio ao ensino do método de Monge abordado na disciplina de Geometria Descritiva (GD). A ferramenta desenvolvida permite ao utilizador interagir com os conteúdos programáticos da disciplina, facilitando o desenvolvimento da capacidade de observação espacial exigida pela mesma. Para a avaliação da proposta de solução, foi implementado um protótipo no sentido de permitir ao utilizador melhorar esta capacidade. Os resultados obtidos foram muito encorajadores, havendo, no entanto, ainda espaço para futuros aperfeiçoamentos.

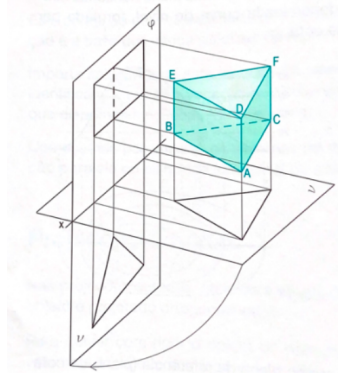
**Palavras-chave:** Desenvolvimento *web*, Computação Gráfica, Geometria Descritiva, Ensino Secundário

## 1 Introdução

Nos últimos anos, a tecnologia e o mundo digital têm demonstrado um crescimento exponencial e, conseqüentemente, têm surgido cada vez mais soluções tecnológicas nas diversas áreas de estudo. O ensino de GD não é uma exceção e a área de educação tem sofrido alterações com a introdução de novas tecnologias na sala de aula, tanto a nível físico (e.g. computadores e projetores), como a nível digital (e.g. escola virtual e *moodle*).

Atualmente, o lecionamento de GD insere-se no ensino secundário e envolve a aprendizagem do método de Monge. O método de Monge recorre a sistemas de representação para representar os elementos geométricos tridimensionais na folha de papel. A Representação Diédrica (RD) é um dos sistemas de representação lecionados na disciplina de GD. Na RD, o objeto em estudo é projetado para dois planos perpendiculares (plano frontal  $[\phi]$  e horizontal  $[v]$ ) através de retas projetantes perpendiculares a esses mesmos planos, como é possível observar

na Figura 1. Para representar o sistema no modelo bidimensional é necessário recorrer ao rebatimento do plano de projeção horizontal ( $v$ ) sobre o plano de projeção frontal ( $\varphi$ ) [1].



**Fig. 1.** Representação Diédrica [1]

No ensino verifica-se uma certa dificuldade por parte dos estudantes em estabelecer a correspondência entre as duas representações [2]. Sendo assim, a solução proposta tem como objetivo desenvolver uma ferramenta de apoio ao estudante no âmbito de facilitar e acelerar o desenvolvimento da capacidade de observação espacial. A solução desenvolvida também pretende desempenhar o papel de ferramenta de apoio e promoção ao ensino da disciplina.

Pretende-se assim, a realização de uma aplicação que permita ao utilizador observar e interagir com os dois sistemas de representação (sistema tridimensional e o sistema bidimensional, através do método de Monge) em simultâneo.

O artigo divide-se em seis secções sendo que, este tem como objetivo introduzir o problema proposto e a motivação para a realização deste projeto. Na segunda secção é apresentada a recolha de informação sobre soluções e tecnologias existentes que se enquadram no contexto do problema. A terceira secção é dedicada à análise do problema e ao desenho da solução. A quarta secção foca-se nos detalhes da implementação da solução e esclarece o funcionamento das ferramentas disponibilizadas pelo sistema. A quinta secção aborda uma sessão de avaliação de usabilidade da ferramenta. Por fim, a última secção encerra o artigo com uma breve conclusão a respeito do trabalho realizado e dos objetivos atingidos.

## 2 Estado da Arte

Esta secção apresenta o estudo preliminar de duas soluções que se enquadram no contexto do problema e as tecnologias que contribuíram para a realização do projeto.

## 2.1 Trabalhos Relacionados

Na análise dos trabalhos relacionados foram considerados três aspetos fundamentais:

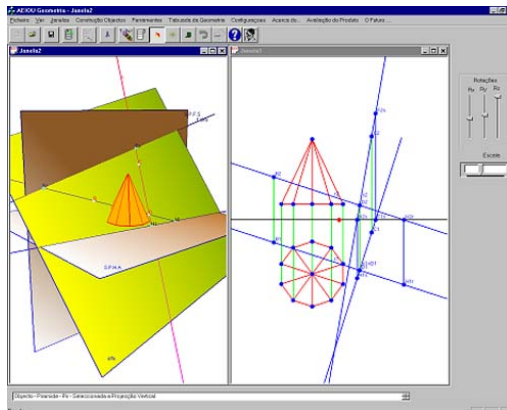
- **Acessibilidade:** Acessível e compatível com a maioria dos dispositivos regularmente utilizados pelos alunos e docentes de GD.
- **Dinamismo:** Dinâmico na visualização do espaço tridimensional de modo que o utilizador possa ter a melhor perceção espacial dos elementos geométricos expostos.
- **Funcionalidade:** Descreve o maior número de funcionalidades com o intuito de demonstrar o conteúdo da área de GD.

Dos diversos programas existentes destacam-se duas soluções: “AEIOU - Geometria Descritiva” e “Plataformas *web* de aprendizagem”.

Premiado no IV concurso da Microsoft Portugal, o programa “AEIOU – Geometria Descritiva” foi desenvolvido para apoiar os alunos na visualização e compreensão da GD no ensino secundário. O *software* foi desenvolvido na linguagem Visual C++ usufruindo da biblioteca OpenGL. O programa disponibiliza duas janelas de visualização, uma para o modelo tridimensional e a outra para o Sistema de Dupla Projeção Ortogonal (SDPO) [3]. Em relação aos aspetos fundamentais referidos acima, a aplicação é bastante completa e demonstra estar a par das regras utilizadas no ensino de GD. A sua interface permite inserir qualquer elemento geométrico e o seu ambiente proporciona a resolução de exercícios e até mesmo a visualização da sua solução (Figura 2). Contudo, esta solução revela uma interface pouco atualizada e sendo um programa executável relativamente antigo pode apresentar incompatibilidades entre os sistemas operativos mais recentes. A nível de funcionalidades, o programa disponibiliza diversas ferramentas para demonstrar o conteúdo da disciplina porém, não suporta os métodos auxiliares de rebatimento.

Na área de GD, existe uma grande prevalência de plataformas *web* que disponibilizam conteúdo de apoio à matéria lecionada no programa da disciplina. Plataformas como “GD Online” e “GeoGebra” disponibilizam conteúdo de apoio ao estudo como por exemplo: resoluções de exercícios por passos, vídeos explicativos e a visualização estática tridimensional e bidimensional [4,5]. Ao contrário da solução anterior estas plataformas são de fácil acesso e não apresentam quaisquer problemas de compatibilidade. No entanto, estas plataformas não possibilitam qualquer tipo de interatividade com os elementos geométricos apresentando assim, uma fraca componente dinâmica. Contudo, as plataformas *web* na sua maioria abordam o programa completo de GD, demonstrando assim, uma forte componente funcional.

Sendo assim, a proposta de solução procura a conjugação destas duas abordagens tirando partido das vantagens assinaladas e reduzindo os problemas identificados nas mesmas.



**Fig. 2.** Representação de uma pirâmide e de um plano no programa “AEIOU - Geometria Descritiva” [6]

## 2.2 Tecnologias Existentes

Visto que o ambiente *web* proporciona uma grande acessibilidade com os dispositivos modernos, a ferramenta desenvolvida recorre a tecnologias compatíveis com este ambiente. Posto isto, a biblioteca Web Graphics Language (WebGL) é a tecnologia escolhida para projetar e expôr os elementos geométricos da GD no ambiente *web*. WebGL é uma Application Programming Interface (API), multi-plataforma, capaz de renderizar gráficos tridimensionais de baixo nível, baseado na tecnologia OpenGL for Embedded Systems (OpenGL ES). Esta tecnologia proporciona o tridimensionalismo às plataformas *web* sem o recurso a *plugins* e está disponível em quase todos os *browsers* com maior representatividade: Safari (Apple), Chrome (Google), Edge (Microsoft), e Firefox (Mozilla). Esta API possibilita a introdução dos conceitos de GD na janela de um *browser* e a exposição dos elementos geométricos estudados de forma dinâmica, evitando a instalação de *software* adicional nos dispositivos dos utilizadores [7].

Contudo, o WebGL é uma API de “baixo nível”, introduzindo um nível de complexidade desnecessário no desenvolvimento da solução. Sendo assim, o uso de uma *framework* é recomendado para facilitar e acelerar o processo de desenvolvimento. Existem enúmeras *frameworks* de WebGL porém, conclui-se que a *framework* mais adequada seria a ThreeJS. Esta decisão teve em conta dois fatores determinantes: a *performance* da *framework* e a facilidade de implementação da mesma [8].

Para suportar a implementação da *framework* ThreeJS e o ambiente *web* são utilizadas as tecnologias Hypertext Markup Language (HTML), Cascading Style Sheet (CSS) e JavaScript (JS)/TypeScript (TS). Estas tecnologias permitem a construção de plataformas *web* em que, o HTML estabelece a estrutura da página, o CSS descreve o estilo da página e o JS/TS define a lógica e a funcionalidade da mesma [9,10,11].

### 3 Análise e Desenho da Solução

A análise do problema e o desenho da solução são etapas no processo de engenharia de *software* em que se especificam: os requisitos do sistema, a modelação do sistema por diferentes níveis de abstração e descrevem-se as componentes da solução.

#### 3.1 Domínio do problema

Para o desenvolvimento da solução, é necessário modelar conceitos previamente abordados e estabelecer relações entre os mesmos. Na Figura 3, podemos observar o modelo de domínio do problema simplificado. O problema divide-se em duas componentes: a componente bidimensional (simulação da folha de papel) e a componente tridimensional (mapeamento e visualização 3D). Os dois objetos responsáveis pela demonstração de cada uma das componentes são: o GeometryMongeRenderer referente à componente bidimensional e o Geometry3DRenderer referente à componente tridimensional. Cada um destes possui uma câmara (PerspectiveCamera e OrthographicCamera) e uma cena apropriada (GeometryMongeScene e Geometry3DScene) que permitem a visualização dos objetos criados pelo utilizador. Estes objetos podem ser criados ou removidos através da Toolbox (Caixa de ferramentas) pertencente ao GeometryMongeRenderer. A Toolbox contém uma lista de ferramentas que permitem ao utilizador criar ou remover elementos geométricos na RD (MongePointProjection e MongeLineProjection) e na Representação Tridimensional (RT) (PointProjection e LineProjection). Com base nas regras de GD, os elementos geométricos podem ser, posteriormente, mapeados em objetos tridimensionais (Point, Line e Plane).

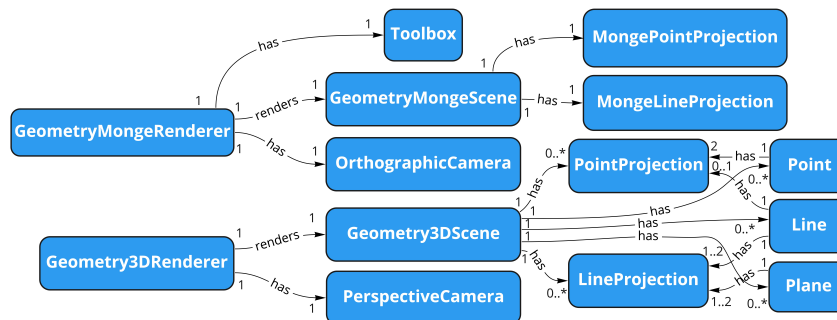


Fig. 3. Modelo de domínio simplificado

A modelação do sistema foi a etapa mais exigente para o funcionamento correto da ferramenta pois, este processo requer a conjunção de dois aspetos: a compreensão dos conhecimentos de GD e o domínio da programação orientada a objetos a ser aplicada na implementação.

## 4 Implementação da Solução

O desenvolvimento da Graphical User Interface (GUI) foi de extrema importância não só para a concretização das funcionalidades propostas, mas também, para a melhor experiência possível do utilizador. Na Figura 4, é possível observar a GUI do sistema com as componentes identificadas por caixas de cores: a RT (amarelo), a RD (azul) e a caixa de ferramentas (vermelho).

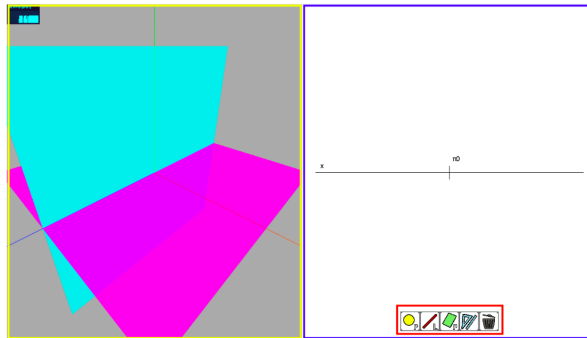


Fig. 4. GUI do sistema

O sistema recorre à *framework* ThreeJS para a criação digital das projeções e dos elementos geométricos nos sistemas de representação. As classes responsáveis por este processo (*Geometry3DRenderer* e *GeometryMongeRenderer*) usufruem de objetos *WebGLRenderer* da *framework* mencionada. No entanto, este objeto por si só não apresenta utilidade para a aplicação e, por isso, é necessário associar ao *renderer* um objeto do tipo *Scene* e um objeto do tipo *Camera*. O objeto *Scene* é responsável por guardar e manipular os objetos gráficos gerados e o objeto *Camera* dita ao *renderer* a forma como os objetos são observados. Como descrito no desenho da solução, cada *renderer* tem associado a respetiva *scene* mas também, tem associado a *camera* correta (Figura 5).

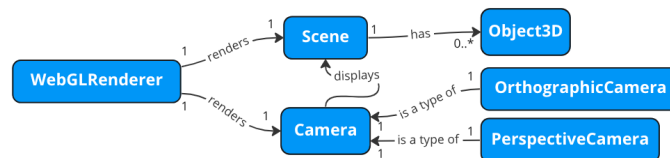
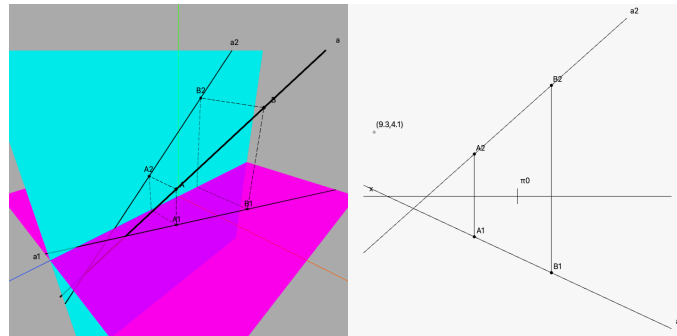


Fig. 5. Modelo de domínio simplificado da *framework* ThreeJS

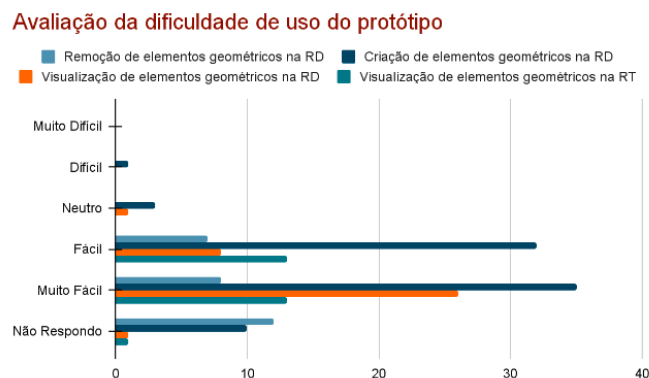
É necessário referir este aspeto, pois o objetivo da aplicação é mostrar os dois sistemas de representação em simultâneo (Figura 6).



**Fig. 6.** Demonstração de uma reta a passar por dois pontos criados com a ferramenta

## 5 Avaliação da Solução

O protótipo desenvolvido foi objeto de estudo numa sessão de avaliação da usabilidade do sistema. A avaliação da usabilidade da solução foi realizada no dia 13 de junho de 2022 por um grupo de nove alunos e um docente de GD do décimo ano da escola secundária João Gonçalves Zarco, em Matosinhos. Nesta sessão, reuniu-se o *feedback* dos participantes envolvidos através de um questionário. No final do questionário, foi facultado um espaço para os participantes poderem fazer críticas e sugestões ao sistema, no formato de resposta aberta. Realça-se a crítica à sensibilidade do cursor para a seleção das coordenadas dos pontos na RD e à GUI pouco apelativa. É possível observar na Figura 7 os resultados obtidos do questionário referentes à componente de avaliação da dificuldade de uso do protótipo. Quanto à funcionalidade do sistema, 90% dos participantes caracterizaram como satisfatória e todos os participantes consideraram a simulação do espaço tridimensional útil para a aprendizagem do método de Monge demonstrando assim, uma forte componente dinâmica.



**Fig. 7.** Resultados obtidos na componente de avaliação da dificuldade de uso do protótipo

## 6 Conclusões

A ferramenta desenvolvida neste projeto permitiu adquirir e consolidar não só conhecimentos tecnológicos, como também, conhecimentos pedagógicos. O projeto revelou a importância da presença da engenharia informática na transformação de outras áreas de estudo, neste caso, no apoio ao ensino de GD.

Os resultados obtidos da sessão de avaliação revelaram grande potencial da solução melhorar a aprendizagem dos conteúdos de GD. Os resultados positivos relativos ao dinamismo que a solução proporciona realçam a importância da tecnologia e a forma como esta foi implementada. No entanto, a nível de acessibilidade, conclui-se que existe um grande interesse em melhorar a GUI do sistema e tornar a utilização da ferramenta mais fácil. A ferramenta também se pode tornar mais complexa e completa, originando funcionalidades como: a persistência de dados, uma versão *offline* da aplicação, edição *online* entre utilizadores e a compatibilidade com outras plataformas, nomeadamente, mobile.

Desta forma, fica o desejo da continuação do desenvolvimento da ferramenta e a aplicação da mesma nas salas de aula, cumprindo assim o seu objetivo.

## Referências

1. Müller, M. J. (2022). MANUAL DE GEOMETRIA DESCRITIVA A - 10.o ANO (1o Edição). Porto Editora.
2. Justiniano, R. J. (2014). Dificuldades de aprendizagem na disciplina de Geometria Descritiva A : um caso de estudo na Região Autónoma dos Açores. Universidade Lusíada de Lisboa.
3. Alves, D. (2012). Prática de Ensino Supervisionada Geometria Descritiva A (10o ano) Oficina de Artes (12o ano) PROGRAMAS DE GEOMETRIA DINÂMICA Novas Metodologias de ensino/aprendizagem. Universidade da Beira Interior
4. GD Online. (2022). GD Online. <https://www.gdonline.pt/>. [Acedido em Abril de 2022]
5. GeoGebra. (2022). GeoGebra — Apps Clássicas de Matemática - usados por mais de 100 milhões de alunos e professores em todo o mundo. <https://www.geogebra.org/?lang=pt-PT>. [Acedido em Abril de 2022]
6. AEIOU - Geometria Descritiva. (2022). AEIOU - Geometria Descritiva — Ensino da Geometria Descritiva Assistido por Computador. <http://www.estgv.ipv.pt/PaginasPessoais/fmorgado/aeiougd/imagens.htm>. [Acedido em Abril de 2022]
7. Khronos. (2022). WebGL Overview - The Khronos Group Inc. <https://www.khronos.org/webgl/>. [Acedido em Abril de 2022]
8. Johansson, J. (2021). Performance and Ease of Use in 3D on the Web - Comparing Babylon.js with Three.js. Faculty of Computing Blekinge Institute of Technology.
9. Flanagan, D. (2011). JavaScript: The Definitive Guide (M. Loukides & T. Elsey, Eds.; 6th ed.). O'Reilly.
10. Microsoft. (2022). TypeScript: Documentation - TypeScript for the New Programmer. <https://www.typescriptlang.org/docs/handbook/typescript-fromscratch.html>. [Acedido em Abril de 2022]
11. W3. (2022). HTML & CSS - W3C. <https://www.w3.org/standards/webdesign/htmlcss>. [Acedido em Abril de 2022]

# Geração Automática de Código-Fonte de um Servidor *GraphQL* através das Especificações de uma API REST

André Botelho<sup>1</sup>[0000-0003-2326-0780] and Fátima Rodrigues<sup>1,2</sup>[0000-0003-4950-7593]

<sup>1</sup> Instituto Superior de Engenharia do Porto, Portugal

<sup>2</sup> Interdisciplinary Studies Research Center  
1192232@isep.ipp.pt, mfc@isep.ipp.pt

**Resumo.** À medida que os projetos de engenharia de software aumentam em escala e complexidade as vantagens de uma abordagem de desenvolvimento orientado pelo modelo começam a ultrapassar os custos do desenvolvimento de geradores de código e das ferramentas que os acompanham.

O artigo presente descreve a arquitetura, implementação e implantação de um gerador de código em contexto empresarial descrevendo o modo como foi utilizado para, a partir das especificações da API REST de uma aplicação, gerar código fonte para um servidor GraphQL que fornece uma interface alternativa para interagir com a API REST subjacente realizando operações de consulta de dados.

A abordagem selecionada assenta num método de desenvolvimento orientado ao modelo, onde as especificações da API REST servem como fonte única de verdade para a geração automática da camada GraphQL em tempo de compilação, mas sem prejuízo da legibilidade do código gerado. É realizada, também, uma análise da viabilidade do método de desenvolvimento onde se conclui que, em determinadas situações, a utilização de um método de desenvolvimento orientado ao modelo é vantajoso quando comparada com métodos tradicionais de desenvolvimento.

O projeto desenvolvido mostra os pontos fortes do desenvolvimento orientado ao modelo e os seus benefícios em determinados contextos.

**Palavras-chave:** Caso de Estudo, Desenvolvimento Orientado ao Modelo, Geração Automática de Código, Integração de Sistemas.

## 1 Introdução

O *Pulsar* é um gestor de operações inteligente que suporta a realização de diversas operações de gestão dentro de uma empresa, desde gestão de recursos humanos à gestão financeira, de projetos, de orçamentos, entre outros. A API do *Pulsar* está documentada no formato *OpenAPI* através da ferramenta *Swagger*. Esta API possui uma dimensão elevada de entidades: tomando em consideração apenas os *end-points* com o verbo *HTTP GET* existem mais de 830 *end-points* e mais de 1750 *schemas* constituídos por um total de mais de 6130 propriedades; existindo também a agravante desta API se encontrar em desenvolvimento ativo, sofrendo constantes alterações. Analisando as

características do problema, nomeadamente a grande dimensão das especificações, a sua frequente alteração e, talvez o mais importante, o facto destas se encontrarem documentadas num formato que facilita o seu processamento por meios digitais, foi concluído que a utilização de um gerador de código seria uma abordagem vantajosa do ponto de vista do esforço necessário para desenvolver a aplicação [1].

A arquitetura *REpresentational State Transfer (REST)* tem sido a escolha proeminente no que toca a *APIs WEB* [2], fornecendo interfaces simples para interagir com sistemas externos, no entanto, *Graph Query Language (GraphQL)* é uma alternativa moderna a esta arquitetura [3], capaz de fornecer as mesmas funcionalidades de uma *API REST* mas dando também facilidades acrescidas ao desenvolvedor como *schemas* consistentes, documentação automatizada e *APIs* descobríveis [4].

A equipa de desenvolvimento do *Pulsar* mantém uma *API REST* para interagir com o *Pulsar*, no entanto, pretende realizar uma mudança gradual para uma *API GraphQL*. Esta mudança gradual implicaria, por um período, fornecer duas *APIs* em formatos distintos para interagir com o mesmo sistema. A solução encontrada foi a de criar uma camada *GraphQL* que não interagisse diretamente com o sistema, mas que o fizesse por meio da *API REST* existente. Através de uma metodologia de desenvolvimento orientada ao modelo, esta camada poderia ser gerada automaticamente com recurso às especificações da *API REST* existentes.

## 2 Geradores de Código

Podemos definir um gerador de código como um sistema cujo objetivo é a geração automática de código de alto-nível [5, pp. 11–12] por *software*, baseado numa ou mais especificações [6, p. 226], de modo a automatizar trabalho repetitivo, mas também quando é necessário obter uma representação textual de um modelo [7].

A utilização de modelos como artefactos primários de desenvolvimento e não só para documentação de um sistema é uma característica diferenciadora da metodologia de *Model Driven Design (MDD)* [5, p. 1]. Segundo [1, p. 334] existe a necessidade de reduzir custos, melhorar performance, aumentar a produtividade e manter a competitividade, esta necessidade leva ao maior controlo dos centros de custo do processo de desenvolvimento; os *MDD* auxiliam no controlo de custos pois deslocam o esforço da implementação do sistema do desenvolvedor humano para o gerador automático. Segundo [5, p. 3] a distância entre o modelo e a implementação de um sistema pode ser reduzida por um gerador de código. É ainda sugerido por [8, p. 377] que o objetivo do *MDD* é aumentar o nível de abstração utilizado pelos desenvolvedores.

Na literatura existem diversas tipologias distintas para classificar geradores de código, tanto quanto à sua arquitetura, como ao seu *output* ou funcionamento. Jörges [5, pp. 27–33] realiza um sumário de várias tipologias de acordo com: o método de descrição das transformações do modelo, o guia das transformações do modelo, a forma do destino, o padrão de geração e ainda de acordo com a porção de código gerada pelo gerador; alternativamente Dimitros, Richard, e Fiona [6, p. 226] categorizam geradores de código de acordo com a sua orientação a código ou a modelo.

A mesma diversidade observada na classificação categórica dos geradores de código também se verifica nas suas arquiteturas. Algumas das arquiteturas foram descritas por Arnoldus, Brand, e Serebrenik [7, pp. 138–141]:

- *Gerador de Etapa Única*: Todo o processo e cálculos são realizados num único modelo;
- *Gerador de Duas Etapas*: Existe uma divisão entre a etapa de transformação do modelo, e a etapa de emissão código;
- *Arquitetura Modelo-Visualização-Controlador (MVC)*: Decompõe o processo de geração de código em três etapas, cada uma com a sua responsabilidade. O modelo executa os cálculos, o controlador envia mensagens para o modelo e disponibiliza uma interface para que a visualização comunique com ele, e a visualização controla o formato do *output* final.

A arquitetura *MVC* é bastante comum para a geração de páginas *WEB* no lado do servidor, como nas *frameworks ASP.NET* e *JSP*. Esta arquitetura é normalmente emparelhada com o uso de templates que consistem em porções estáticas de texto e porções dinâmicas que são avaliadas por um motor de template [5, p. 29].

Existem vários benefícios no que conta à utilização de geradores de código, a organização *Dessault Electronique* [1] realizou uma análise custo-benefício da utilização de geradores de código num dos seus contratos, concluindo que no contexto em que o custo agregado do desenvolvimento e operação para o gerador de código é menor do que o custo total do programa de *software* a desenvolver, e no contexto em que o gerador de código não é dedicado a um único projeto e pode ser reutilizado, o desenvolvimento de um gerador de código é financeiramente lucrativo [1, pp. 234–235] apresentando vários benefícios técnicos, como a conformidade entre as especificações e a informação retida em *software*; e a geração do código fonte sem erro humano [1, p. 339].

Mesmo sendo lucrativa, a implementação de um gerador de código também está associada a certos custos como os de produzir e validar as ferramentas, os de modificar partes do *software* existentes, e os recorrentes para manter as ferramentas [1, p. 340], e criar e manter artefactos para a geração de código [8, p. 391].

Por ser uma arquitetura relativamente moderna, existem várias equipas de desenvolvimento que procuram migrar as suas *APIs REST* para *GraphQL* pelo que procuram soluções que realizem esta migração de modo automático, no entanto, muitas destas tecnologias não se encontram em desenvolvimento [9] ou possuem falhas no que toca à sua capacidade de processar *schemas* recursivos [10].

A biblioteca *openapi-to-graphql* [11] implementa diversas funcionalidades avançadas no que toca à geração de *schemas GraphQL* através de especificações *OpenAPI*, no entanto, esta biblioteca não gera o servidor *GraphQL* pelo que não consegue lidar com entidades com parâmetros dinâmicos, é incapaz de gerar os pedidos *HTTP* para a *API REST*, e também não é capaz de criar *schemas* que possam ser lidos por humanos.

Devido à falta de soluções adequadas foi necessário desenvolver um novo gerador que será seguidamente descrito.

### 3 Implementação

#### 3.1 Introdução

A API REST do *Pulsar* é desenvolvida na linguagem de programação *Java*, esta linguagem permite ao desenvolvedor adicionar meta-dados aos diferentes símbolos de um programa através de “anotações.” A equipa de desenvolvimento do *Pulsar* utiliza estas anotações, em conjunto com uma ferramenta chamada *Swagger* para gerar ficheiros *JSON* no formato *OpenAPI* que descrevem a API.

O projeto desenvolvido utiliza maioritariamente esta documentação como especificação do modelo de domínio para a geração da camada *GraphQL*. Adicionalmente foi desenvolvido um segundo processador de anotações que extrai do código do *Pulsar* informação sobre os direitos de acesso aos diferentes *end-points* da API para que possam ser verificados diretamente na camada *GraphQL*.

#### 3.2 Arquitetura

A solução adotada assume a API do *Pulsar* como a "fonte única de verdade" evitando falhas humanas e dessincronizações ao criar a ponte de comunicação entre os *end-points REST* e a camada *GraphQL*. É também possível desencadear a geração automática da camada *GraphQL* a partir do sistema de implantação já utilizado no projeto *Pulsar*, isto torna a sincronização entre a API REST e a camada *GraphQL* um processo transparente que não envolve esforço dos desenvolvedores para além dos esforços necessários em manter os artefactos necessários para a geração de código.

O gerador criado foi desenhado de modo a poder ser reutilizado em outros projetos. É um gerador de duas etapas que implementa a arquitetura *MVC*, na primeira etapa o gerador analisa as especificações da API e transforma-as numa representação intermédia, esta representação é o "contexto" utilizado na segunda etapa, para a geração de código através de templates. As interações entre os principais componentes e os artefactos do projeto está ilustrado na Fig. 1.

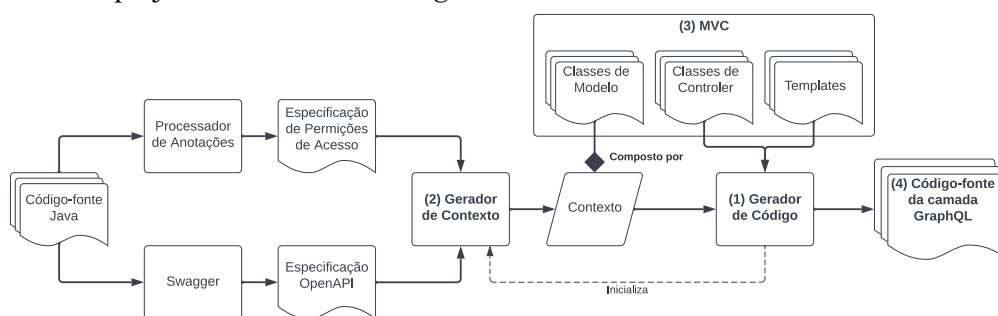


Fig. 1. Relação entre os componentes do gerador de código e os artefactos produzidos.

É assim possível afirmar que o projeto desenvolvido engloba quatro componentes principais: (1) o componente do gerador de código, responsável por despoletar o gerador de contexto e iniciar o processo de renderização; (2) o gerador de contexto, responsável

por instanciar classes de modelo de acordo com as especificações *OpenAPI* do *Pulsar*; (3) o componente *MVC* que engloba o modelo de domínio do gerador (*model*), os templates utilizados na geração de código (*view*) e os controladores (*controller*) para enviarem mensagens entre as *views* e as classes de modelo; e por último, (4) o componente *GraphQL* que é a camada gerada pelo gerador de código, responsável por transformar os pedidos *GraphQL* em pedidos *REST* e realizar a gestão de acessos. As interações entre estes componentes podem ser visualizadas no gráfico da Fig. 2.

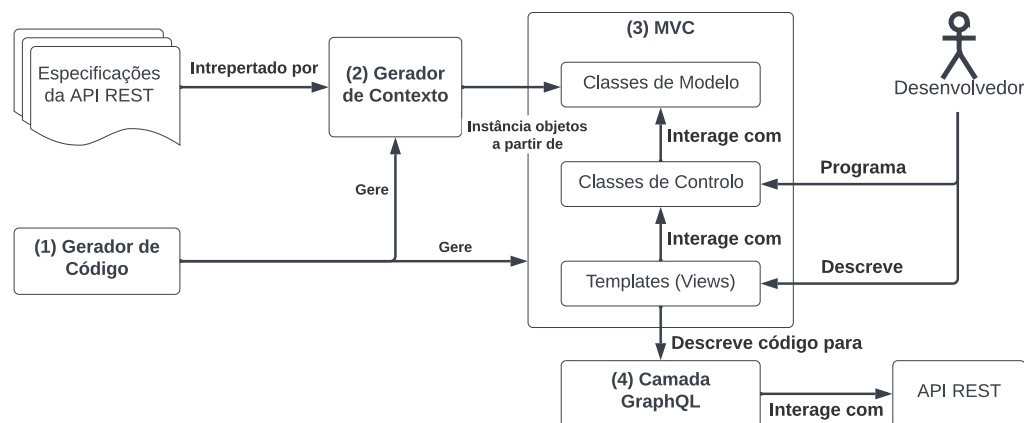


Fig. 2. Interação entre os diversos componentes do sistema.

O desenvolvedor, para alterar o código fonte da camada *GraphQL*, na maioria dos casos, apenas necessita de alterar os templates e os controladores.

O modelo de domínio do componente *GraphQL* contempla o mesmo domínio da *API REST*, ou seja, um modelo de domínio semelhante ao da aplicação subjacente, visto que a sua finalidade é interagir com a mesma numa razão de um-para-um. Este modelo é implementado automaticamente pelo gerador de código através das especificações da *API REST* logo a sua análise não é pertinente para o desenvolvedor, reduzindo os custos associados com esta tarefa. O componente *GraphQL* é completamente independente dos restantes componentes e, após ser gerado, não necessita deles para operar.

O modelo de domínio do componente *MVC* contempla a estrutura da *API REST*, representando os diferentes pedidos *HTTP* que podem ser realizados à *API* e a estrutura dos dados das suas respostas. É necessário acentuar o facto de que este modelo é um conjunto de classes que modelam os diferentes elementos da estrutura de uma *API REST*. Quando o gerador de contexto realiza o processamento das especificações da *API REST* e cria objetos a partir das classes de modelo, o conjunto de objetos criados é denominado de contexto do gerador de código. Quando um template é renderizado, este é renderizado com o auxílio de um controlador que por sua vez interage com o contexto atual. O mesmo contexto pode ser renderizado diversas vezes utilizando o mesmo controlador, mas criando ficheiros de código-fonte distintos ao ser renderizado em contextos diferentes. Não é correto dizer que o modelo de domínio do gerador é alterado para cada renderização, mas é possível afirmar que o contexto de cada renderização difere.

O gerador de contexto está intrinsecamente conectado ao modelo de domínio das especificações da *API REST*, visto que a sua responsabilidade é interpretar o ficheiro das especificações e gerar o contexto requerido pelas classes do componente *MVC*.

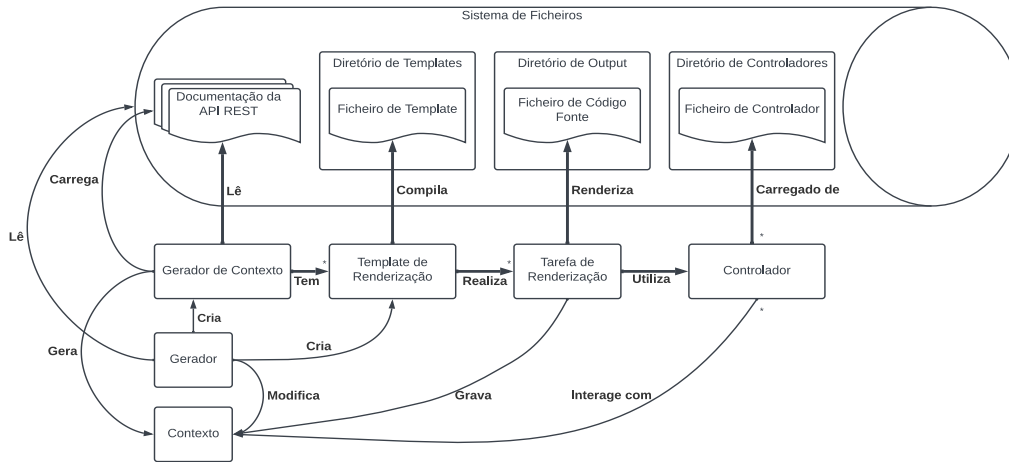


Fig. 3. Modelo de domínio do gerador de código.

O modelo de domínio do componente gerador de código ilustrado na Fig. 3 apenas contempla os diferentes elementos necessários para implementar a arquitetura *MVC* e despoletar a geração de um contexto. O componente de geração de código é agnóstico ao contexto e ao código aplicacional que está a gerar, funcionando como uma ferramenta generalizada que poderá ser utilizada em projetos futuros.

A subdivisão do projeto em quatro domínios diferentes é uma abordagem que auxilia a perceção do problema, permitindo ainda a generalização do gerador de código face ao modelo de domínio da aplicação a ser gerada.

### 3.3 Implantação

De modo a simplificar o processo de geração de código, esta tarefa pode ser integrada no processo de implantação atual do *Pulsar*.

O código-fonte do *Pulsar* está atualmente alojado no serviço *BitBucket* que está interligado com o sistema *Jenkins*, permitindo a criação de *pipelines* – pequenos programas que permitem a execução de código no ambiente remoto onde se encontram os repositórios de controlo de versões. Estes *pipelines* têm normalmente como objetivo automatizar os processos de implantação de código, tendo a possibilidade de executar tarefas como *build* (compilação do código), *deployment* (*upload* de artefactos para ambiente externo) e, no caso do projeto atual, são capazes de automatizar o processo de geração de código, este processo está representado na Fig. 4.

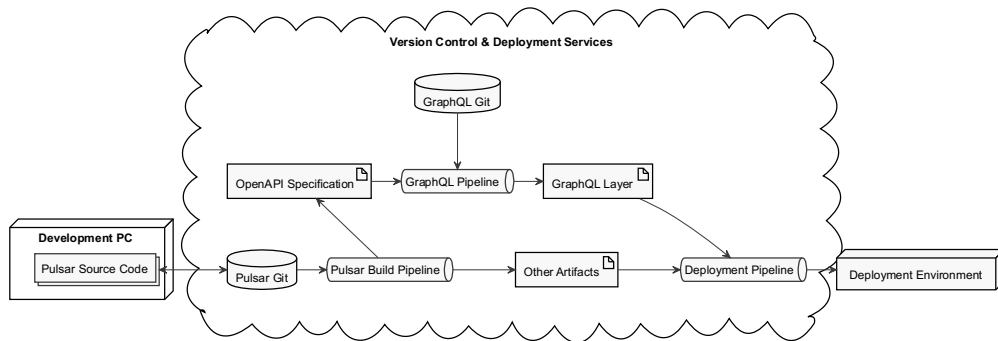


Fig. 4. Processo de implantação.

Ao iniciar o processo de implantação o código do *Pulsar* é enviado para a *pipeline* de *build* onde (entre outros artefactos necessários para executar o programa) é gerada a especificação da *API REST*. Esta especificação é então utilizada como um *input* para a *pipeline GraphQL* que gera automaticamente a camada *GraphQL* com o gerador de código implementado no projeto. Em seguida, o sistema gerado automaticamente e os artefactos gerados pela *pipeline* de *build* do *Pulsar* são implantados no ambiente de implantação seleccionado pelo desenvolvedor.

#### 4 Resultados Obtidos

Uma medida que pode ser utilizada para avaliar a solução proposta é o volume de código escrito e gerado. Para o código escrito manualmente é considerado todo o código que foi escrito por um desenvolvedor, incluindo código que interage com o gerador, o código-fonte do gerador, templates, controladores, entre outros. Para o código gerado automaticamente é considerado todo o código que é gerado como *output* do gerador, incluindo *schemas GraphQL* e testes automaticamente gerados.

	Nº de Linhas	Nº de Palavras	Nº de Caracteres
Escritos Manualmente	2.630	10.586	103.857
Automaticamente Gerados	198.709	414.987	7.560.556
Total	201.339	425.573	7.664.413

Tabela 1. Volume de linhas escritas manualmente e automaticamente geradas.

É de notar que os resultados apresentados não englobam a totalidade do volume de código escrito pois não tomam em consideração nem as revisões escritas manualmente, nem gerações de código realizadas com diferentes versões da *API REST*. No entanto é possível observar que cerca de 98.64% do código final da aplicação foi gerado automaticamente, mostrando uma redução significativa do custo de desenvolvimento.

## 5 Limitações e Trabalho Futuro

A correspondência entre a *API* do *Pulsar* e a camada *GraphQL*, apesar de estar assegurada pelo gerador de código, depende da correção da documentação do *Pulsar*, no entanto, existem diversos *end-points* em que o formato do *output* é alterado do *schema* documentado para um *schema* alternativo ou até para um formato de dados binário dependendo do pedido *HTTP*.

O projeto é também limitado pela completude da documentação da *API REST*. Certas informações como restrições de acesso a campos de entidades não estão documentadas de uma forma sistematizada, pelo que a camada *GraphQL* não pode ser gerada tendo esse parâmetro em conta.

O *GraphQL* impossibilita o retorno de objetos indefinidamente aninhados, no entanto, isto é possível na *API REST*, esta é uma limitação incontornável que surge de diferenças intrínsecas entre as duas arquiteturas.

Um melhoramento possível seria a reestruturação do gerador de acordo com as tipologias propostas por [7, pp. 2–3] para gerar código garantidamente seguro, ou seja, a arquitetura do gerador tornaria impossível a geração de um *output* sintaticamente incorreto.

## Referências

1. H. le Doeuff, “Automatic generation of ada source code for the Rafale Mission computer,” in *Ada in Europe*, 1994, pp. 333–348.
2. R. T. Fielding and R. N. Taylor, “Architectural Styles and the Design of Network-Based Software Architectures,” 2000.
3. L. Byron, “GraphQL First Stable Release,” Jun. 11, 2018. <https://github.com/graphql/graphql-spec/releases/tag/June2018> (accessed Jun. 21, 2022).
4. “Introspection | GraphQL,” *graphql.org*. <https://graphql.org/learn/introspection/> (accessed Sep. 10, 2022).
5. S. Jörges, “The State of the Art in Code Generation,” in *Construction and Evolution of Code Generators: A Model-Driven and Service-Oriented Approach*, S. Jörges, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 11–38. doi: 10.1007/978-3-642-36127-2\_2.
6. R. F. and P. F. A. C. Kolovos Dimitrios S. and Paige, “An Agile and Extensible Code Generation Framework,” in *Extreme Programming and Agile Processes in Software Engineering*, 2005, pp. 226–229.
7. J. Arnoldus, M. van den Brand, A. Serebrenik, and J. J. Brunekreef, *Code Generation with Templates*. Paris: Atlantis Press, 2012. doi: 10.2991/978-94-91216-56-5\_1.
8. M. J. Rutherford and A. L. Wolf, “A Case for Test-Code Generation in Model-Driven Systems,” in *Generative Programming and Component Engineering*, 2003, pp. 377–396.
9. B. Dail, “JSON to GraphQL,” GitHub, Oct. 09, 2021. <https://github.com/Aweary/json-to-graphql> (accessed Sep. 20, 2022).
10. R. Krivtsov, “Swagger-to-GraphQL,” GitHub, Sep. 18, 2022. <https://github.com/yarax/swagger-to-graphql> (accessed Sep. 20, 2022).
11. International Business Machines, “OpenAPI-to-GraphQL,” GitHub, Aug. 18, 2022. <https://github.com/IBM/openapi-to-graphql>

## Gestão de Propostas Comerciais

Marly Juliana Pimpita

Instituto Superior de Engenharia do Porto, Portugal

**Resumo.** A empresa OPT (Otimização e Planeamento de Transporte), devido a complexidade do seu processo comercial, precisa garantir que os custos dos investimentos nos projetos sejam cobertos pela comercialização dos mesmos, por meio de apresentação de propostas comerciais. Estas propostas podem ser aceites ou não pelos seus clientes, isto é, adjudicadas ou não adjudicadas. Por isso, este trabalho foi desenvolvido com o objetivo de auxiliar na gestão de propostas comerciais a fim de otimizar e padronizar o processo de submissão das propostas de modo que as propostas adjudicadas mantenham o seu padrão e as não adjudicadas sejam melhoradas, de forma a minimizar os custos e maximizar lucros. Deste modo, conhecer os motivos das adjudicações e não adjudicações das propostas auxiliará na otimização da gestão das propostas comerciais. Este artigo apresenta gráficos, resultantes da manipulação dos dados extraídos do servidor da OPT referentes ao processo comercial, que servem como base para análise e melhoria do processo comercial.

**Palavras-chave:** otimização, proposta comercial, processo comercial, análise, padrão.

### 1 Introdução

Inicialmente, podemos pensar que para uma empresa de serviços ter um projeto bem-sucedido no mercado, basta que este projeto seja de qualidade. Mas, a qualidade de um projeto não é o único fator determinante do seu sucesso comercial. Outro fator não menos importante é a maneira como fazemos a proposta, garantindo custos efetivos sem pôr em risco a qualidade do serviço.

A OPT é uma empresa de serviços, de desenvolvimento de Sistemas para Planeamento, Otimização, Gestão e Informação ao Público sobre transportes, isto há trinta anos. Ela possui um ERP (*Enterprise Resource Planning*) com o registo de toda a sua atividade. A sua atividade comercial consiste na criação de processos e de propostas relativas a estes processos direcionados a uma entidade. Estas propostas de sistemas muitas vezes não são aceites pelos clientes. Pelo que se pretende analisar as razões destas situações, para que seja essencial garantir que os custos das propostas comerciais sejam cobertos pelos projetos, que as oportunidades de negócio não sejam perdidas e simultaneamente que exista coerência na sua definição.

## 2 Estado da Arte

Dois dos trabalhos relacionados a este projeto encontrados foram: uma monografia de um Sistema Para Controle De Proposta Comercial aplicado à Empresa *Unique Solution* (Dapont, 2013) e um artigo de análise de melhoria do processo comercial nas empresas de projeto (Oliveira e Melhado, 2003).

A seguir passo a descrever cada um deles e as suas semelhanças com este projeto.

### 2.1 Sistema de Controle De Proposta

O Sistema de Controle De Proposta Comercial é um sistema de processamento de transações na área de venda da empresa (Figura 1). O objetivo é otimizar o tempo e a gestão das negociações comerciais com aos clientes, gerando assim a diminuição dos riscos de perdas de negociações comerciais. Este também é o objetivo deste projeto, que a análise dos dados resulte na otimização da gestão das propostas comerciais.

Este sistema foca em melhorar a gestão de controle de propostas comerciais da empresa *Unique Solution*, permitindo ao administrador acompanhar as informações diárias de todas as negociações, disponibilizando informações sobre clientes para auxiliar os vendedores na tomada de decisões ou informações para identificar possíveis motivos para o não fechamento de uma negociação com o cliente ou ainda diversas informações para melhoria de vários fatores do setor comercial. Neste projeto, um dos principais fatores de análise é a não adjudicação das propostas, então, pretende-se analisar os dados e encontrar os possíveis motivos da não adjudicação das propostas comerciais e trabalhar para melhorar e reduzir as perdas de oportunidades de negócios.

Código	Atividade De	Responsável	Data Contato	Data Retorno	Cliente	Status	Feedback
1	Cliente	Cleiton	06/12/2013	16/12/2013	AC Tech	Iniciada	Verificar possível consultoria
5	Consultoria	Cleiton	02/12/2013	19/12/2013	JS Express	Em Aberto	Retornar contato as 14h00
7	Sistema	Cleiton	16/12/2013	17/12/2013	Maxi Paper	Concluída	Verificar se analisou a proposta
6	Sistema	Consultor	23/12/2013	30/12/2013	AC Tech	Iniciada	Verificar se analisou a proposta

Fig. 1. Tela Inicial do Sistema

(Sistema para controle de Proposta Comercial aplicado à Empresa Unique Solution, 2013)

## 2.2 Gestão do Processo Comercial em Empresas de Projeto

O artigo de Gestão do Processo Comercial em Empresas de Projeto faz uma análise e propõe melhorias no processo comercial das empresas de Projeto. Nele são citadas as principais dificuldades no setor de projeto, debatidos aspetos importantes no desempenho das empresas e propostos sugestões para melhoria da sua gestão.

Um dos pontos abordados é a qualidade do contato com os clientes. A importância de ter funcionários apropriadamente treinados ou orientados para melhor execução desta tarefa. Estes funcionários precisam ter conhecimento da estrutura e cultura da organização, de modo a não comprometerem o relacionamento comercial com o cliente e serem capazes de fechar negócios. Na OPT, os funcionários recebem formações e orientações frequentes sobre contato com clientes, para uma melhor comunicação e relação comercial entre eles.

Um outro ponto abordado é a elaboração das propostas comerciais. A necessidade de uma análise crítica dos requisitos do cliente antes da elaboração da proposta. Isto é fundamental para que a empresa não aceite um projeto fora do seu domínio, que podem causar aumento no custo dos recursos, pela necessidade de um estudo profundo acerca do(s) aspeto(s) fora do domínio da empresa não identificado(s) aquando da formulação da proposta, ou, isto pode resultar na produção de um projeto que não tem os requisitos do cliente implementados conforme desejado por ele, em outras palavras, um péssimo projeto, o que levaria a uma perda categórica do cliente e/ou uma ação judicial na busca por indemnização do prejuízo(s) causado(s). Estas situações seriam bastante desagradáveis e custosas para uma empresa. De forma a evitar este problema e qualificar da melhor forma o cliente, a OPT faz uma análise de risco, no processo de qualificação do processo, onde são analisados diversos fatores do cliente e do projeto, de modo a decidir se vale a pena investir naquele projeto ou não.

## 3 Contexto

A OPT, por ser uma empresa de desenvolvimento de Sistemas Informáticos que visa a gestão e otimização do sistema de transporte, foca na criação de soluções inovadoras que resolvam problemas genéricos deste sector e que possam ser utilizadas por várias empresas. Isto porque é a OPT quem faz o investimento aos sistemas e espera o retorno nas vendas dos mesmos. Alguns produtos deste tipo que a empresa tem disponível no mercado são:

- **Gist (Gestão Integrada dos Sistemas de Transportes):** “O sistema GIST é um pacote de software com diversos módulos que permitem [o acesso] a informação base relativa à rede de transportes, às linhas e às viagens a realizar. Permite ainda gerar, de uma forma otimizada, os horários das viaturas e dos motoristas, incluindo o seu escalamento diário” (<https://www.opt.pt/gist-3/>).
- **Infopub:** “O sistema InfoPub Web permite a geração automática de mapas e horários à paragem, diminuindo o tempo de produção destes esquemas e assegurando a rápida produção de informação consistente e atualizada.” (<https://www.opt.pt/infopubweb/>).

- **DRT ou Mobus:** é um serviço de transporte que visa melhorar a eficácia da mobilidade em zonas de baixa densidade.

Deste modo, são criados os processos, relacionando-os aos produtos, para um determinado cliente, e a seguir são criadas as propostas e associadas estes processos.

O processo é uma ação contínua que expressa continuidade na realização de determinada atividade. Neste caso, o nosso processo comercial expressa a comercialização de um produto/serviço da empresa OPT.

Um processo comercial tem o valor do pipeline não definido (vazio) ou em um destes estágios:

- **Oportunidade:** o primeiro contato, quando se percebe que há uma oportunidade de negócio;
- **Qualificação:** consiste em qualificar a oportunidade através de uma análise de risco onde são analisados fatores como o âmbito, número de intervenientes, planeamento e entre outros, e decide-se se vale a pena dar continuidade;
- **Proposta:** elaboração e apresentação da proposta;
- **Análise pelo cliente:** estágio em que o cliente analisa a proposta;
- **Negociação:** negociação da proposta pelas partes;
- **Adjudicação:** quando a proposta é aceite;
- **Perda:** quando o cliente aceita a proposta da concorrência;
- **Desistência:** quando o cliente por alguma razão (valor da proposta alto, mudança nas condições económicas, ou outra razão) decide simplesmente desistir da proposta;
- **StandBy:** quando há mudanças nas condições e por isso a proposta precisa ser reavaliada;
- **Transferência:** quando as condições mudam ao ponto de ter de transferir esta proposta para uma nova com as atualizações pedidas/necessárias.

Assim, as propostas não são adjudicadas por motivos de perda, desistência, standby ou transferência. A Figura 2 ilustra o fluxo do processo nos estágios do pipeline:

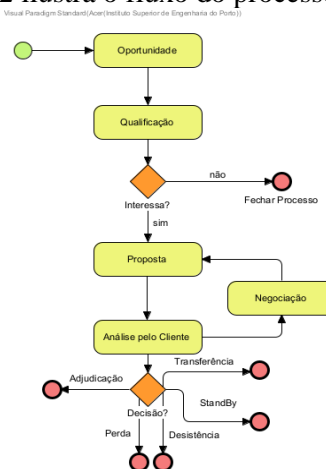


Fig. 2. Fluxo dos estados do processo

## 4 Implementação da Solução

### 4.1 Descrição da solução

A solução consiste na criação de um repositório de dados, dados estes extraídos do ERP da organização, sobre o qual deverão ser desenvolvidas funcionalidades de análise de acordo com a tipologia de produtos e clientes envolvidos no processo, tendo em conta os parâmetros registados.

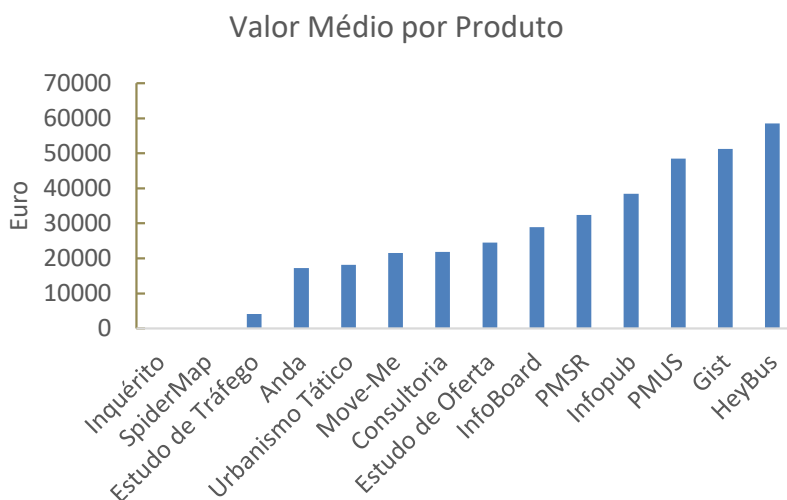
Após a alimentação do repositório com dados vindo do ERP, são realizadas alterações sobre os dados para permitir a correta manipulação destes. Estas alterações incluem:

- A criação da entidade produto (não existente no servidor) e a opção de adicionar ou editar estes produtos.
- A adição do campo “veículos” nas entidades, com o número de veículos da entidade e a inserção destes números.
- A inserção de valores de propostas não disponibilizadas diretamente pelo servidor.
- A associação dos produtos aos processos de forma automática e manual.

Após estas alterações, já é possível realizar as manipulações dos dados que originam dois gráficos: o gráfico do valor médio dos produtos e o gráfico do rácio entre o valor da proposta e o número de veículos da entidade. Gráfico estes, usados como objeto de análise.

**Associação dos produtos aos processos de forma automática.** A associação dos produtos aos processos de forma automática foi feita por meio de um algoritmo. Como já referido anteriormente, o servidor da organização não contém a entidade “produto”. A forma com que alguns produtos eram associados aos processos era colocando o nome do produto no nome ou na referência do processo. Por isso, o algoritmo consiste em procurar pelos nomes dos produtos nos nomes ou referências dos processos, caso encontre associa o produto ao processo. Tendo em conta que alguns nomes de produtos podem ser sufixos ou prefixos de outras palavras, verifica-se se o termo não pertence a uma outra palavra antes de fazer a associação, se pertencer a associação não é feita. Depois desta associação dinâmica, são feitas associações de forma manual para a boa parte dos processos que não têm os produtos associados por este algoritmo.

**Valor médio dos produtos.** O gráfico a seguir nos mostra o valor médio das propostas adjudicadas, por produto. Valores em euro.



**Fig. 3.** Gráfico das médias dos valores adjudicados por produto

Com esta figura (Figura 3), podemos saber quais são os produtos com a média do valor das propostas adjudicadas mais alta e mais baixa.

**Rácio entre o valor da proposta e o número de veículos da entidade.** Este gráfico (Figura 4) apresenta o rácio entre os valores das propostas dos processos e o número de veículos da entidade, por produto. Isto porque os projetos podem ser de diferentes tamanhos (número de veículos), e para ser equivalente a comparação entre eles foi necessário fazer o rácio entre o valor da proposta e número de veículos. Os valores apresentados não são apenas de processos com propostas adjudicadas, o que se pretende também é analisar as propostas nos outros estados e perceber o porquê de não terem sido adjudicadas. Os valores no sistema podem ser agregados por dias, semanas, meses e anos pelas funções de máximo, mínimo, soma, média e contagem. Na figura aqui apresentada os dados estão agrupados em meses pela função média. Então temos os valores médios das propostas referentes aos produtos, por cada mês. Os produtos/serviços da empresa apresentados neste gráfico são: consultoria, gist e infopub.

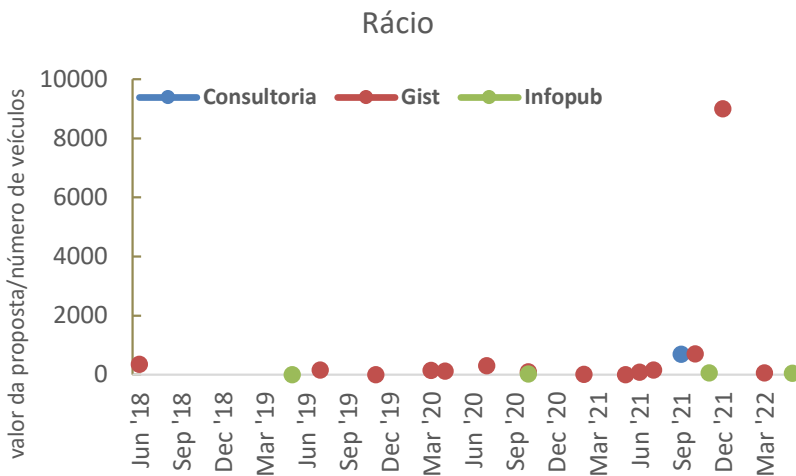


Fig. 4. Rácio entre os valores das propostas e o número de veículos da entidade por produto

É possível visualizar o valor do rácio e da proposta, o estado do pipeline, o nome e o número de viaturas da entidade, por meio de um *tooltip*. Como mostra a figura 5.

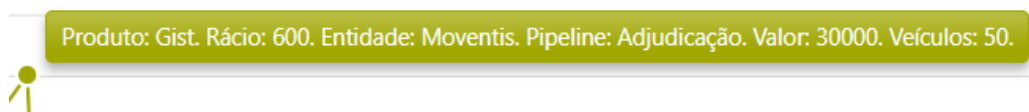


Fig. 5. Um *tooltip* de um dos pontos do gráfico da Figura 4

Os dados apresentados nos gráficos (Figuras 4 e 5), demonstram que as propostas com valor definido referentes as entidades com números de veículos definidos, resultou de serem apenas as propostas adjudicadas e relativas a três produtos. Mas ainda assim, é possível verificar que 81,8% destas propostas são de entidades com número de viaturas acima de 100 e que o produto “Gist” tem mais propostas do que outros produtos e com valores muito mais altos.

## 5 Conclusão

O sistema em causa intenta o melhoramento na elaboração e gestão das propostas comerciais. Fazendo com que haja manipulação e apresentação dos dados ao cliente a fim de possibilitar uma análise que resultará na identificação de padrões. Esta funcionalidade facilita e agiliza o processo de análise, potencializando as chances de deteção dos motivos por trás das perdas de negócios.

Com os gráficos apresentados é possível analisar o estado das propostas relacionado com os produtos e seus valores, e com as entidades e suas dimensões, e por meio

disto identificar os padrões. Com o gráfico do rácio podemos afirmar que mais dados representados tornariam a nossa conclusão mais confiável. Porém, com os dados apresentados podemos concluir que as propostas adjudicadas são na sua maioria relativas à entidades de tamanho médio e grande e que o produto “Gist” é o produto com melhor resultado tanto em quantidade de propostas quanto no valor delas. É desta forma que este projeto cumpre com o seu objetivo de auxiliar na gestão das propostas comerciais, porque o propósito principal do cliente com este projeto era perceber a relação entre as adjudicações, os produtos, o valor das propostas e as entidades. Relação esta demonstrada nos gráficos.

## Referências

1. OPT Página Inicial, <https://www.opt.pt/>, último acesso 25/11/2022.
2. O que é o ERP?, <https://dynamics.microsoft.com/pt-pt/erp/what-is-erp/>, último acesso 27/11/2022.
3. Dicionário Online de Português, <https://www.dicio.com.br/processo/>, último acesso 29/10/2022.
4. Dapont (2013). Sistema para Controle de Proposta Comercial aplicado à empresa Unique Solution. Monografia. Universidade Regional de Blumenau. Blumenau. Brasil.
5. Oliveira e Melhado (2003). “Gestão do Processo Comercial em Empresas de Projeto”. Workshop Brasileiro de Gestão do Processo de Projeto na Construção de Edifícios. Belo Horizonte. Brasil. Setembro.
6. Construindo um processo comercial: o que é, por que é importante definir e quais são as 7 etapas principais, <https://www.agendor.com.br/blog/processo-comercial-aula-1/>, último acesso 27/11/2022.
7. Como escrever uma proposta comercial (exemplos e modelos), <https://pt.venngage.com/blog/proposta-comercial/>, último acesso 27/11/2022.

---

***HeltiSolar - Sistema CRM e Power Platform*** \*Simão Gomes<sup>1</sup>, Ricardo Almeida<sup>2</sup>, and José Pinto<sup>3</sup><sup>1</sup> ISEP, Porto, Portugal

1191061@isep.ipp.pt

<sup>2</sup> GILT-ISEP, Porto, Portugal

ral@isep.ipp.pt

<sup>3</sup> Hitachi Solutions Portugal, Porto, Portugal

jpinto@hitachisolutions.com

**Resumo** No contexto da sociedade atual, o mercado de produção de energia, e em particular o mercado de energias renováveis, é cada vez mais importante. Este é um mercado caracterizado por rápidas mudanças em termos tecnológicos e de preferências de consumidores, pelo que as empresas têm de estar em constante adaptação.

É neste sentido que a empresa *HeltiSolar*, uma empresa do setor energético, definiu como objetivo o desenvolvimento de uma solução que permitisse a gestão dos três segmentos de mercado da empresa e que melhorasse a eficiência na produção e a relação da empresa com os seus clientes e fornecedores. Assim, no presente documento, encontra-se detalhado o processo de desenvolvimento de um protótipo de uma plataforma de *Customer Relationship Management* (CRM) que permitisse resolver o problema identificado, ou seja, que permitisse, por exemplo, gerir as linhas de produção de painéis solares, gerir os clientes e gerir o inventário da *HeltiSolar*. Este processo envolve a identificação e análise de requisitos, o desenho da solução, a sua implementação e testes e a análise dos resultados.

**Palavras-chave:** *Microsoft Power Platform*, Desenvolvimento *Low-Code*, *Dynamics 365*, Energia Solar

## 1 Introdução

No contexto atual da sociedade, a produção de energias renováveis é cada vez mais importante, sendo que, por exemplo, em 2021 as energias renováveis representavam 81% da nova capacidade instalada a nível global e nesse mesmo ano a capacidade global de produção de eletricidade a partir de fonte renovável aumentou 9,1% comparativamente com 2020 [1]. Contudo, e tal como muitos outros mercados, o mercado das energias renováveis caracteriza-se por rápidas mudanças quer em termos tecnológicos, quer em termos de preferências de consumidores [2].

---

\* Apoiado pela empresa *Hitachi Solutions Portugal*, empresa a qual se agradece a oportunidade e a colaboração que permitiram a elaboração do projeto e deste artigo científico.

Deste modo, o modelo de negócio de muitas empresas deste setor tem vindo a ser alterado. De facto, há algumas décadas o mercado era direcionado para as vendas e o cliente tinha de se adaptar à oferta existente. No entanto, nos últimos anos este modelo tem vindo a ser alterado e devido à abertura do mercado, tem existido uma tendência das empresas em prestar um serviço mais personalizado aos seus clientes [2], o que resulta num aumento no investimento das organizações no *Customer Relationship Management* (CRM).

A *HeltiSolar* é uma empresa que atua no setor da energia solar, e esta possui um sistema desatualizado tecnologicamente e funcionalmente, o que faz com que a eficiência em termos de produção, e a relação com clientes e fornecedores não se encontre ao nível pretendido. Deste modo, o objetivo geral deste projeto é o desenvolvimento de uma solução que permita à *HeltiSolar* melhorar a eficiência na produção e a relação com fornecedores e antigos e novos clientes, e, portanto, após a finalização deste projeto a solução deverá incluir:

- Gestão da distribuição dos produtos aos clientes.
- Gestão das linhas de produção e dos processos de manutenção de painéis solares.
- Gestão dos clientes da empresa e potenciais novos clientes.
- Gestão do inventário da empresa.
- Automatização de processos.

Desta forma, e para desenvolver o protótipo mencionado, foram identificados os requisitos não funcionais e os requisitos funcionais.

Assim, relativamente, a requisitos funcionais, estes são as funcionalidades que integram o sistema/solução e que o software deve fornecer [3], sendo os requisitos funcionais identificados os seguintes:

- UC1: Registrar painel solar
- UC2: Calcular energia produzida por painel solar
- UC3: Fornecer *feedback* a partir de formulário
- UC4: Fornecer *feedback* a partir de *chatbot*
- UC5: Gerir processo de manutenção
- UC6: Criar processo de manutenção
- UC7: Associar contentor a um cliente
- UC8: Gerir processo de vendas
- UC9: Efetuar previsão de energia
- UC10: Consultar Dashboard

Já relativamente a requisitos não funcionais, estes são restrições impostas a um sistema, que têm como objetivo tornar o sistema mais eficiente e melhorar a experiência dos utilizadores [4]. Para identificar estes requisitos foi utilizado o modelo FURPS+.

#### **Funcionalidade**

- As funcionalidades devem estar protegidas por meios de autenticação e autorização.

### Usabilidade

- O Sistema deve estar documentado de forma a permitir a futuros desenvolvedores entender a sua organização, *design* e implementação.

### Confiabilidade

- As funcionalidades devem estar testadas por meio de testes de integração, *end-to-end* ou unitários.

### Suportabilidade

- A solução deve ser compatível com vários dispositivos e formatos (mobile, tablet e computador).

### Outros (+)

- Restrições de implementação: As tecnologias *Microsoft Power Platform* e *Dynamics 365* devem ser utilizadas para o desenvolvimento da solução.

Para desenvolver a solução mencionada, foi , portanto, usado o *Microsoft Power Platform* que é uma plataforma *low-code* composta por vários componentes. Contudo, no contexto do desenvolvimento desta solução foram apenas utilizados o *Power Apps*, o *Power Virtual Agents*, o *Dataverse*, *Power Automate* e o *AI Builder*.

## 2 Estado da Arte

No contexto deste trabalho foram aplicados os conceitos de CRM e *Low Code*, pelo que é desde logo importante compreender estes conceitos.

### 2.1 *Customer Relationship Management*

O CRM é uma ferramenta que permite organizar contactos e gerir relações com atuais e potenciais clientes [5], que abrange estratégias, técnicas, ferramentas e tecnologias utilizadas por organizações para desenvolver, reter e adquirir clientes. Neste sentido, os sistemas CRM compilam a informação sobre o cliente obtida em diferentes canais/pontos de contacto entre a empresa e o cliente, tais como, o *website* da empresa, *emails*, redes sociais e departamento de Marketing num único local.

Relativamente ao CRM, realça-se ainda, o facto de que este pode ser dividido em quatro categorias [5] :

- CRM Operacional.
- CRM Colaborativo.
- CRM Analítico.
- Marketing CRM.

Por fim, estes sistemas podem ser ainda de dois tipos: *On-Premise* ou *On-Demand*. Os sistemas CRM *On-Premise*, são sistemas cujas infraestruturas, hardware, sistema operativo e software complementar estão localizados nas instalações do cliente. Já em sistemas CRM *On-Demand*, as infraestruturas, hardware, sistema operativo e software complementar estão localizadas na nuvem. A escolha entre estes dois tipos deve ser feita pela empresa, tendo em conta os seus requisitos [6].

### 2.2 Low Code

O *low-code* é um termo introduzido por Clay Richardson e John Rymer, em 2014, tratando-se o mesmo de uma abordagem de desenvolvimento de software com o mínimo desenvolvimento manual de código e que se baseia em código visual e de interfaces *drag and drop* [7]. Deste modo, pode-se considerar que uma plataforma de desenvolvimento *low-code* tem 3 componentes principais [8]:

- Um Ambiente de Desenvolvimento Integrado (IDE do inglês *Integrated Development Environment*) visual que é o *core* da plataforma de desenvolvimento e que consiste na interface visual de *drag and drop* usada.
- Ligadores (do inglês *Connectors*) a diversos sistemas externos que ligam a plataforma a diversos serviços de *backend*, bases de dados e APIs.
- Gestor de vida útil da aplicação.

Para além destes componentes, uma plataforma de *low-code* fornece diversas funcionalidades *Out-Of-The-Box* (OOTB) que eliminam a necessidade de desenvolvimento de módulos da aplicação de raiz, permitindo uma implementação mais rápida [8]. Neste sentido, plataformas *low-code* permitem ainda criar produtos adaptáveis e incluir pessoas com poucos conhecimento de programação em equipas de desenvolvimento. Desta forma, segundo a Gartner, 65% de todas as funções de desenvolvimento de aplicações irão incluir pelo menos uma plataforma *low-code* até 2024 e 66% das grandes empresas utilizarão pelo menos 4 plataformas *low-code* [9].

### 2.3 Comparação plataformas CRM

No contexto deste artigo são comparadas três plataformas CRM: o *Dynamics 365*, o *SalesForce Customer 360* e o *Oracle Customer Experience*. Assim na Tabela 1 são descritas forças e fraquezas de cada uma das plataformas obtidas a partir de comparações retiradas de um artigo de comparação do *Dynamics 365* e do *Oracle CX* [10] e de outro que compara o *Dynamics 365* e o *Customer 360* [11].

**Tabela 1.** Forças e Fraquezas das Plataformas CRM

	<i>Dynamics 365</i>	<i>SalesForce Customer 360</i>	<i>Oracle Customer Experience</i>
<b>Forças</b>	Flexibilidade do Preço. Integração com as restantes tecnologias da Microsoft. Solução estável e escalável.	Solução baseada na nuvem. Design simples. Gestão de atividades intuitiva.	Integra uma grande variedade de soluções de terceiros. Acesso a funcionalidades da Oracle.
<b>Fraquezas</b>	Navegação na plataforma. Preço por utilizador. Processo de personalização desafiante.	Falta de integração com o módulo de Field Service. Necessários ligadores para integrar softwares como a solução ERP.	Interface não tão apelativa. Processo complexo para importar solução.

Por fim, o relatório "*Gartner Magic Quadrant for CRM Customer Engagement Center*" 2021 [12] indica que as três plataformas são consideradas líderes no mercado e bastante desenvolvidas, destacando, no entanto, o *Customer 360* da *SalesForce* como a plataforma mais completa no mercado. Contudo, a tecnologia escolhida pela *Hitachi Solutions Portugal* para o desenvolvimento deste projeto foi o *Dynamics 365*, uma vez que a empresa é parceira da *Microsoft*.

#### 2.4 Comparação plataformas *Low-Code*

No contexto deste artigo são comparadas três *Low-code*: o *Power Platform*, o *Mendix* e a *OutSystems*. Deste modo, o relatório "*The Forrester Wave™: Low-Code Development Platforms For Professional Developers, Q2 2021*" [13], as plataformas *Mendix* e *OutSystems* são as plataformas que apresentam melhores capacidades atuais, obtendo melhores pontuações que o *Power Platform* para as ferramentas de dados e integração e ferramentas para testes, implantação e gestão. Para além disso [13] e [14] destacam a *OutSystems*, uma vez que esta possibilita desenvolvimento colaborativo, algo que o *Power Platform* não permite, e é mais acessível que o *Mendix* e o *Power Platform* em termos de escalabilidade de preços. Adicionalmente, na Tabela 2 são descritas forças e fraquezas de cada uma das plataformas obtidas a partir do relatório "*Magic Quadrant for Enterprises Low-Code Application Platforms*" de 2021 [14].

**Tabela 2.** Forças e Fraquezas das Plataformas *Low-Code*

	<i>Microsoft Power Platform</i>	<i>Mendix</i>	<i>OutSystems</i>
<b>Forças</b>	Inovação na abordagem na IA. Capacidade de dar resposta ao mercado. Modelo de negócio.	Inovação. Qualidade do produto e da User Experience. Viabilidade da Plataforma.	Qualidade do produto. Capacidades low-code avançadas. Inovação, como uso de IA em testes.
<b>Fraquezas</b>	Preço. Limitações na implementação de processos de negócio complexos.	Preços premium. Estratégia de marketing. Estratégia geográfica.	Preço. Inexistência de foco numa indústria. Falta de capacidade em modelar decisões complexas.

Para concluir, resta referir que todas estas plataformas são consideradas líderes no mercado, pelo que qualquer uma destas plataformas é considerada uma boa opção para desenvolvimento de aplicações seguindo uma abordagem de *Low-Code*. Uma vez que a *Hitachi Solutions* é parceira da *Microsoft*, foi necessário usar o *Power Platform* para o desenvolvimento da solução.

### 3 Implementação

O desenvolvimento da solução foi feito com o *Dynamics 365* e o *Power Platform*, que são, respetivamente, a plataforma CRM e de desenvolvimento *low-code* da *Microsoft*. Como tal, para implementar os requisitos identificados, foram modelados os dados necessários, ou seja, definiram-se as tabelas necessárias para a solução no *Dataaverse*. Sendo este um dos componentes do *Power Platform*, e tratando-se de desenvolvimento *low-code*, algumas entidades já existiam no *Dynamics 365*, pelo que foram unicamente modificadas de forma a cumprir os requisitos do projeto. No entanto, outras não se encontram definidas OOTB pelo que tiveram de ser criadas e adicionadas ao sistema.

Adicionalmente, e de forma a tornar mais eficiente o sistema, foram implementados diversos mecanismos de automação, como *Flows*<sup>4</sup> e *Workflows*<sup>5</sup>, e foram adicionados *plugins*<sup>6</sup> para estender as funcionalidades do sistema. Com isto, os processos do sistema tornaram-se mais eficientes e a vida dos colaboradores foi simplificada. Assim, por exemplo o colaborador não necessita de preencher

<sup>4</sup> Método para a automatização de processos repetitivos

<sup>5</sup> Série de métodos/funções executadas em sequência e de forma a automatizar processos de negócio

<sup>6</sup> Biblioteca de classes desenvolvida em linguagens suportadas pela plataforma .Net como o C# que incluem lógica de negócio personalizada que pode ser integrada no *Dynamics 365*

dados como a área do painel solar ou energia produzida pelo mesmo, sendo essa informação calculada automaticamente pelo sistema.

Neste seguimento, foi ainda desenvolvido, um BPF<sup>7</sup> que permitiu uniformizar o processo de manutenção efetuado na empresa, e com isto reduzir erros humanos na realização desses processos, e foi ainda desenvolvida uma *Canvas App*<sup>8</sup> para *tablets* para a gestão do processo de manutenção de um painel em contexto móvel, o que permitiu aos colaboradores efetuar o registo dos dados dos processos de manutenção a partir do seu posto de trabalho, e com isto o registo de informação de processos de manutenção de painéis solares foi otimizado.

Não menos importante, foi desenvolvido um *website* para permitir aos clientes fornecer *feedback* à empresa, e neste *website* foi adicionado um agente virtual que simplifica e melhora o processo de apoio ao cliente. Por fim, foram tidas em conta as permissões de acesso de cada colaborador no sistema, de forma a garantir que o sistema é seguro e que as permissões são as mínimas possíveis.

Na Figura 1 encontra-se uma imagem do sistema e em particular do BPF e do formulário desenvolvidos para efetuar o processo de manutenção de um painel solar.

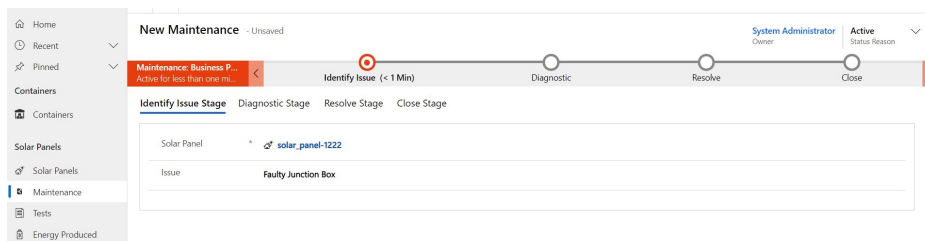


Fig. 1. Exemplo de BPF e Formulário no sistema

## 4 Avaliação do Protótipo e Conclusões

Todos os objetivos definidos inicialmente e identificados na secção 1.1 foram cumpridos e atingidos. Adicionalmente, os requisitos não funcionais apresentados e os requisitos funcionais apresentados na secção 1 foram também cumpridos e implementados com sucesso, como se conseguiu verificar a partir dos testes de integração, *end-to-end*, unitários e de aceitação que foram feitos para os componentes e aplicações desenvolvidas no projeto.

Deste modo, é possível considerar que a solução responde às necessidades apresentadas pela empresa, pelo que a avaliação da solução é positiva.

No entanto, existem algumas melhorias que poderiam ser adicionadas e que enriqueceriam a solução. Deste modo, poderiam ser implementadas as seguintes melhorias:

<sup>7</sup> Guias visuais para auxiliar utilizadores na execução de processos de negócio

<sup>8</sup> Tipo de aplicação personalizada em que o desenvolvedor tem flexibilidade e liberdade no *design*, sendo normalmente desenvolvidas para tablets ou telemóveis

- Melhorias na interface do utilizador.
- Adicionar funcionalidades no agente virtual como rastreamento de encomendas e análise sentimental.
- Adicionar processos automáticos para a gestão de *stock* como uma notificação via *email* ou via *Teams* quando um produto está em falta.

## Referências

1. P. A. Jorge, “Renováveis continuam a crescer – o jornal económico.” <https://jornaleconomico.pt/noticias/renovaveis-continuam-a-crescer-887933>, maio 2022. Último acesso 23/05/2022.
2. P. N. C. Vilas, “Crm na estratégia nacional para a energia o caso edp-energias de portugal,” Master’s thesis, Universidade Autónoma de Lisboa, 2015.
3. M. Martin, “What is a functional requirement in software engineering? specification, types, examples.” <https://www.guru99.com/functional-requirement-specification-example.html>, 6 2022. Último acesso 30/05/2022.
4. A. Papworth, “Non functional requirements — quick guide for the business analyst in 2022.” <https://businessanalystmentor.com/non-functional-requirements/>, fevereiro 2022. Último acesso 19/07/2022.
5. M. Guay, “What is a crm? beginner’s guide to crm software — zapier.” <https://zapier.com/blog/what-is-a-crm/>, 4 2022. Último acesso 29/05/2022.
6. Efficacy, “Differences and advantages of a on-premise crm and on-demand crm.” <https://www.efficacy.com/on-premise-crm-on-demand-crm/>, 2019. Último acesso 29/05/2022.
7. D. Oliveira, “Quais são as vantagens das plataformas low-code?.” <https://pt.primaverabss.com/pt/blog/low-code-programacao/>, fevereiro 2022. Último acesso 29/05/2022.
8. Kissflow, “Low-code — a complete guide to low code development [wiki].” <https://kissflow.com/low-code/low-code-overview/>, junho 2022. Último acesso 29/05/2022.
9. I. Koksál, “The rise of low-code app development.” <https://www.forbes.com/sites/ilkerkoksál/2020/04/29/the-rise-of-low-code-app-development/>, abril 2020. Último acesso 29/05/2022.
10. N. Wright, “Head to head: Microsoft dynamics vs oracle — nigel frank.” <https://www.nigelfrank.com/insights/head-to-head-microsoft-dynamics-vs-oracle>, fevereiro 2018. Último acesso 29/05/2022.
11. N. Wright, “Head to head: Microsoft dynamics vs salesforce — nigel frank,” fevereiro 2018. Último acesso 29/05/2022.
12. N. Bajwa, “Microsoft named a leader in 2021 gartner® magic quadrant™ for crm customer engagement center - microsoft dynamics 365 blog.” <https://cloudblogs.microsoft.com/dynamics365/bdm/2021/10/27/microsoft-named-a-leader-in-2021-gartner-magic-quadrant-for-crm-customer-engagement-center/>, october 2021. Último acesso 30/05/2022.
13. J. Bratincevic, R. Koplowitz, S. Powers, S. Sjoblom, and K. Hartig, “The forrester wave™: Low-code development platforms for professional developers, q2 2021.” <https://reprints2.forrester.com/#/assets/2/225/RES161668/report>, maio 2021. Último acesso 30/05/2022.
14. J. Wong, K. Iijima, A. Leow, A. Jain, and P. Vincent, “Magic quadrant for enterprise low-code application platforms,” tech. rep., Gartner, september 2021.

## Interface para a análise de um sinal ECG

André Monteiro, Duarte. Guimarães, Patrícia Carvalho, Joaquim Alves, Gina Vilão

Instituto Superior de Engenharia do Porto,  
R. Dr. António Bernardino de Almeida, Porto, Portugal  
{1170642; 1180770; 1210161; jaa; gmr}  
@isep.ipp.pt

**Resumo.** A modelização de sistemas fisiológicos tem vindo a apresentar-se cada vez mais como uma mais-valia nos dias de hoje, principalmente na área da saúde. Este projeto teve como objetivo a criação de uma aplicação que lesse o sinal obtido experimentalmente, realiza-se o seu tratamento removendo os vários ruídos que o afetam, e que auxiliasse na avaliação e reprodução de um sinal fisiológico após os vários filtros aplicados. O sinal utilizado neste projeto foi um eletrocardiograma obtido da base de dados da Universidade de Tecnologia de Brno. A interface interativa desenvolvida permite a filtragem do sinal de uma forma eficiente, através das diversas funções que se introduziram, tais como remoção de ruído, remoção da tendência não linear, transição para o domínio das frequências e vice-versa. Para a análise do sinal, a aplicação permite a seleção de intervalos dentro do sinal em estudo, permite a aplicação de vários filtros na remoção dos ruídos. Do tratamento do sinal em estudo a aplicação permite extrair dados relevantes ao estudo e análise do sinal ECG, como a deteção dos picos R e dos pontos Q e S, fornecendo um relatório final das várias etapas do tratamento do sinal inicial.

**Palavras-chave:** ECG, simulação, modelização, matlab.

### 1 Introdução

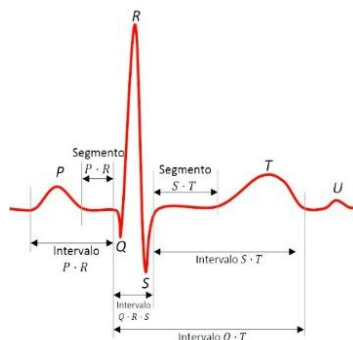
A modelização de sistemas fisiológicos tem sido uma ferramenta cada vez mais usada em ambiente hospitalar, facilitando assim a escolha dos métodos e terapias a serem aplicadas a cada doente conforme as variáveis individuais de cada ser. Neste projeto pretendeu-se desenvolver uma aplicação que analise e reproduza de forma interativa o registo gráfico da atividade elétrica do coração, eletrocardiograma (ECG)[1].

Para descrever a modelização de um sistema fisiológico, pode-se partir da sua terminologia. Um sistema pode ser considerado qualquer conjunto de processos e/ou objetos que se encontrem interconectados. Um modelo é uma representação que aproxima o comportamento de um sistema real, sendo que essa representação é específica até um certo nível, apresentando assim limitações significativas [2].

Na modelização de um sistema o *input* ou o *output*, ou seja, a informação que o sistema recebe ou envia, é representado por uma quantidade, propriedade ou condição

física que está a ser medida (o sinal biológico). No entanto, os sinais biológicos são tipicamente limitados em amplitude e distorcidos por ruídos afetos as condições experimentais quando adquiridos. [2]

O eletrocardiograma é um registo dos fenómenos elétricos que se originam durante a atividade cardíaca, figura 1. Cada contração do músculo cardíaco ou das válvulas cardíacas emite um impulso elétrico na qual é captado por um aparelho denominado eletrocardiógrafo. Deste modo o sinal de um ECG pode ser classificado como um sinal bioelétrico [3].



**Fig.1.** Representação típica da forma de onda e intervalos típicos do ECG[4].

A primeira onda do ECG normal, representa a ativação elétrica (despolarização) das aurículas, onda arredondada, simétrica, de pequena amplitude (0,2-0,3mv) e duração menor que 0,12s, onda P. O segmento PR conecta a onda P ao complexo QRS.

O complexo QRS representa o tempo necessário para que o impulso passe do nódulo sino-auricular para o nódulo auriculoventricular, com uma duração de 0,12s a 0,20s, representa a despolarização dos ventrículos. Posteriormente à despolarização dos ventrículos, isto é, a contração ejeta o sangue dos ventrículos e bombeia-o através das artérias, criando um pulso, com a duração entre 0,06 e 0,10s. A onda R, é a onda do complexo QRS com maior intensidade, representa a despolarização das paredes ventriculares esquerda e direita. O segmento ST representa o fim da despolarização e o início da repolarização ventricular, o ponto de junção entre o final do complexo QRS e o início do segmento ST é denominado de ponto J. A onda T, corresponde esta repolarização ventricular. O intervalo de tempo medido entre o início do QRS ao final da onda T é a soma da despolarização e repolarização ventriculares e representa a sístole ventricular e varia inversamente em relação à frequência cardíaca: quanto > frequência cardíaca < o intervalo QT. Ocasionalmente pode ser identificada a onda U, que representa a fase final da repolarização ventricular.

## 2 Metodologia

Neste projeto recorreu-se à plataforma *PhysioNet*, aonde foi adquirido um sinal original numa base de dados da Universidade de Tecnologia de Brno[5]. Esta base de dados foi criada para o desenvolvimento, avaliação e comparação objetiva de algoritmos de deteção de ondas P. Para além disso, existem cerca de 50 registos de sinais de

ECG de 2 derivações de 2 minutos, onde todos os dados são fornecidos no formato *WaveForm Database (WFDB)*.

Na aquisição de sinal existe sempre associado o ruído da rede elétrica que está relacionada com o local de aquisição do sinal. Este ruído é da ordem dos 50 Hz na grande parte da Europa, Ásia e Oceânia, enquanto na América, é aproximadamente de 60 Hz. Além deste existe ainda o ruído branco dos equipamentos que se mistura com o ruído da rede elétrica.

Por conseguinte, neste projeto aplicaram-se alguns filtros com o objetivo de eliminar as frequências referidas anteriormente.

Primeiro, o sinal original ECG antes de ser filtrado é representado graficamente quer no domínio do tempo, quer no domínio das frequências (nesta última é realizado a conversão para este mesmo domínio e é representada a resposta espectral).

De seguida, é eliminada a tendência não linear através do filtro passa-alto *Butterworth*, tendo sido representada também graficamente no domínio do tempo e da frequência, revelando a resposta espectral.

Posteriormente, aplicaram-se filtros passa-baixo, FIR, para a remoção do ruído, apresentando-se um gráfico no domínio do tempo do sinal e a sua resposta espectral (domínio das frequências).

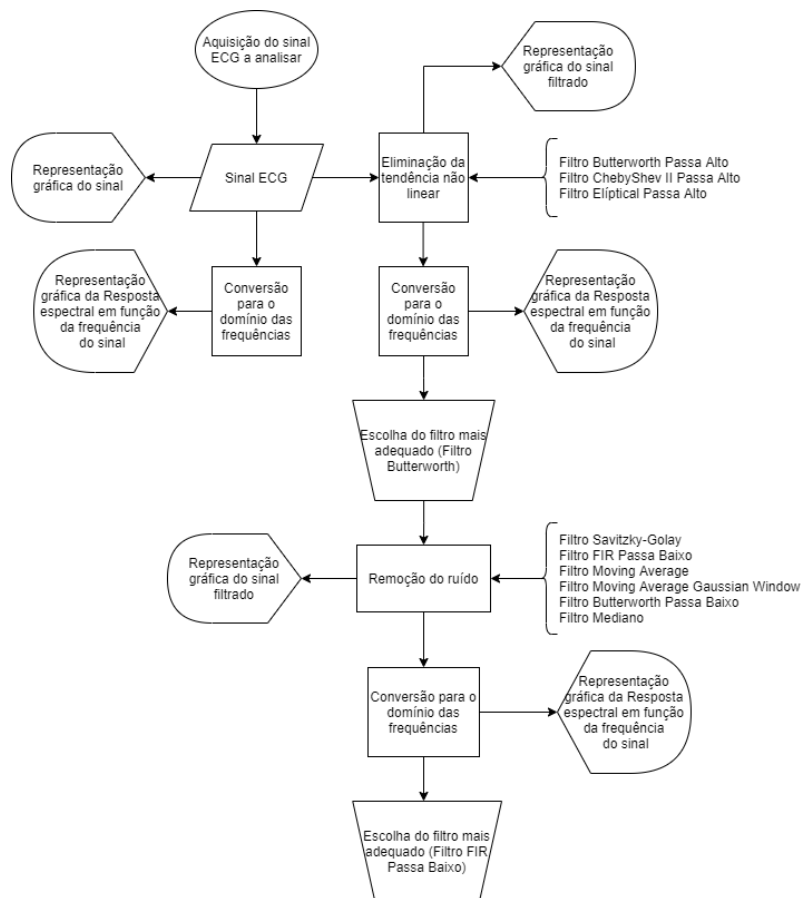


Fig.2. Diagrama de blocos

A escolha dos filtros, tal como é demonstrada no diagrama de blocos da figura 2, foi avaliada através do teste de vários tipos de filtros, tanto passa-alto como passa-baixo, escolhendo no final do teste o tipo de filtro mais adequado ao sinal em estudo.

Para uma melhor compreensão e visualização dos resultados o programa é constituído por uma interface interativa desenvolvida com o recurso do GUIDE (*Graphical User Interface Development Environment*) do Matlab. Na figura 2, encontra-se representado as funções de cada componente da aplicação.

### 3 Resultados e discussão

#### 3.1 Estrutura da aplicação

A aplicação é constituída por vários menus principais e secundários. O menu principal está representado na figura 3.

O programa inicia com apenas uma opção possível, onde é necessário escolher o sinal ECG de um diretório para posterior análise. O sinal necessita de se encontrar no formato wdfb (*waveform database*), pelo que os ficheiros \*.dat e \*.hea são fundamentais para a correta leitura do sinal. Após a seleção do ficheiro, são retornadas todas as amostras da amplitude, tempo do sinal, como também a frequência de amostragem.



Fig. 3. Estado inicial do Menu principal.

A seleção do sinal abre o acesso à opção seguinte, onde direciona para outro GUI (*Graphical User Interface*) como mostra a figura 4:

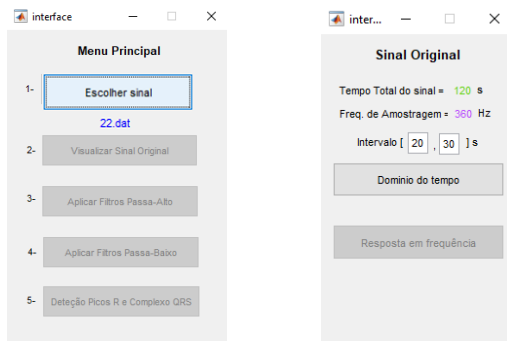
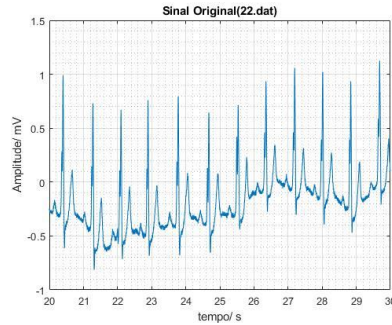


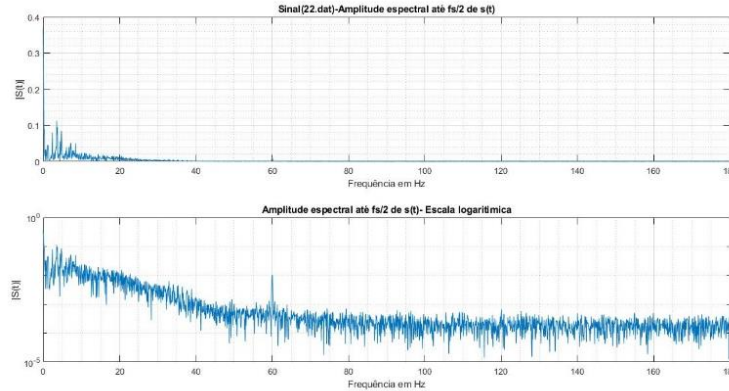
Fig. 4. Acesso à opção Sinal Original e abertura de GUI de visualização do sinal original.

Nesta nova GUI é possível visualizar o tempo total do sinal, frequência de amostragem e selecionar o intervalo mais adequado para a sua análise. É possível selecionar qualquer intervalo válido para dentro dos limites de tempo do sinal. A seguinte figura representa o sinal ECG no domínio do tempo, figura 5.



**Fig. 5.** Representação do sinal ECG no domínio dos tempos no intervalo [20, 30] s.

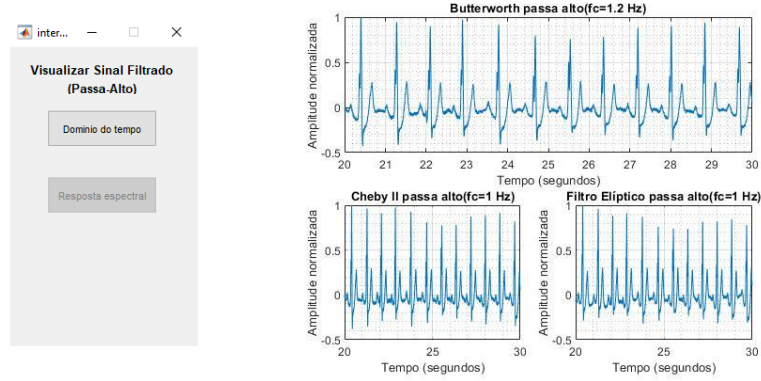
Após se obter o gráfico que relaciona a amplitude com o tempo, foi utilizada a função da transformada rápida de Fourier utilizando o algoritmo de FFT (*Fast Fourier Transform*). Sendo que  $n$  corresponde ao número de amostras mais adequado (inteiro mais pequeno para a potência de 2, com o comprimento intervalo selecionado de dados do sinal), tendo melhor desempenho a nível de cálculo. Desde modo, é possível visualizar a amplitude espectral até metade da frequência de amostragem na escala linear e na escala logarítmica como demonstra a seguinte figura.



**Fig. 6.** Amplitude espectral até  $f_s/2$  do sinal ECG na escala linear e na escala logarítmica.

Deste modo, verifica-se que é necessário filtrar a componente de 0,5 Hz e de 60 Hz do sinal. A seleção da resposta espectral abre o acesso à opção aplicar filtros passa-alto, onde direciona para outro GUI. Para uma melhor análise do sinal, de acordo com o que se consegue detetar visualmente, foram aplicados diversos filtros digitais para a remoção da tendência não linear do sinal e o ruído provocado por interferências causadas por artefactos. Para a remoção dos desvios da linha base, é necessário que se removam frequências baixas do sinal. Assim sendo, foi aplicado os filtros passa-alto *Butterworth* com uma frequência de corte,  $f_c$ , de 1,2 Hz, filtro *Chebby II* com  $f_c = 1$  Hz e filtro Elíptico com  $f_c = 1$  Hz sobre o sinal original.

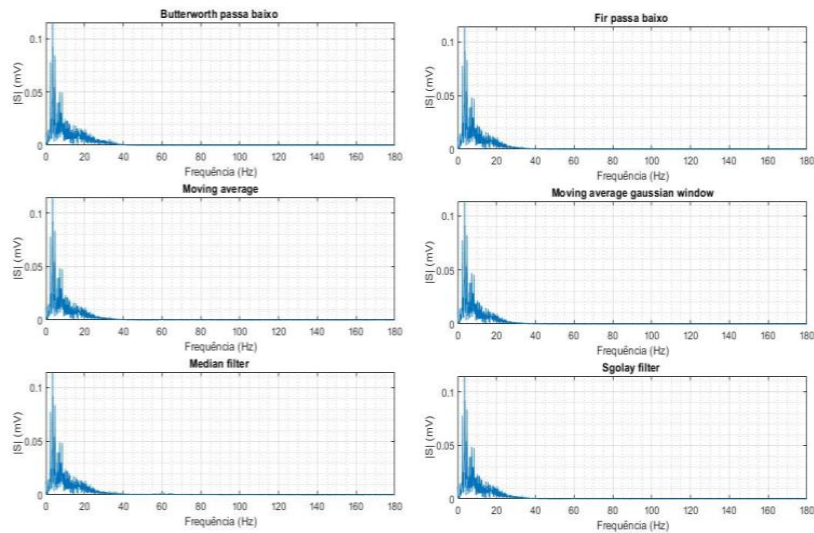
Para uma melhor comparação foi normalizado o sinal de modo que a amplitude do mesmo ficasse entre  $-1$  e  $1$ . O resultado é apresentado na figura seguinte:



**Fig. 7.** GUI de visualização do sinal filtrado (passa-alto). Representação gráfica do sinal ECG após aplicação dos filtros passa-alto.

A aplicação permite continuar a aplicar os restantes filtros pretendidos e apresenta graficamente o resultado de cada um. No sinal ECG de teste, foram aplicados os filtros passa-baixo *Butterworth* com  $f_c = 38\text{Hz}$ , FIR com  $f_c = 38\text{Hz}$ , *Moving Average*, *Moving Average Gaussian Window*, *Median* e *Sgolay* sobre o sinal filtrado por *Butterworth* passa-baixo com  $f_c = 0,37\text{ Hz}$ , para visualizar qual dos filtros se adequava melhor ao sinal em estudo.

Para a remoção do ruído provocado por artefactos, é necessário que se removam frequências altas do sinal. Assim, como se pode observar na figura 8, da resposta em frequência do sinal filtrado, os filtros aplicados removeram com sucesso as frequências pretendidas.



**Fig. 8.** Resposta de frequência do sinal ECG após aplicação de filtros passa-baixo.

### 3.2 Análise do sinal

Após a filtragem do sinal, pode-se analisar o sinal obtido e detetar as coordenadas dos picos R e, por sua vez, a dos pontos Q e S. As coordenadas desses pontos são determinadas através de restrições de amplitude e deste modo considerou-se que os picos R devem ter uma amplitude superior a 0,6, os pontos Q uma amplitude compreendida entre -0,2 e -0,05 e os pontos S uma amplitude inferior -0,2.

A figura 9 representa os picos das ondas R e os pontos Q e S detetados.

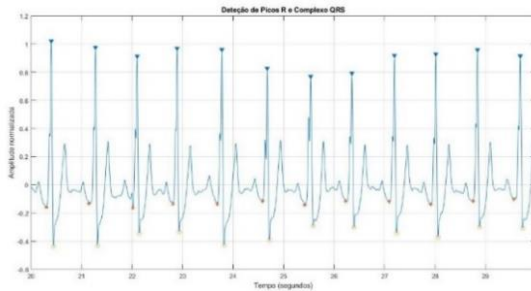


Fig. 9. Representação gráfica dos picos das ondas R e dos pontos Q e S no sinal ECG.

Com a aquisição desses conjuntos de pontos, é possível deste modo traçar as variações de amplitude, assim como realçar todos os complexos QRS presentes na amostra do sinal em estudo, como mostra a figura 10.

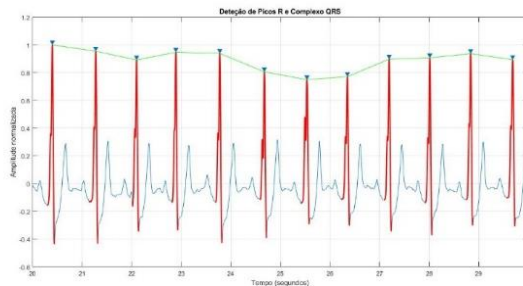
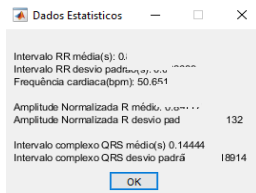


Fig. 10. Representação gráfica dos picos das ondas R, variações de amplitude e do complexo QRS no sinal ECG no intervalo de [20, 30] s.

Por fim, o algoritmo abre uma janela que retorna informação relevante, como o intervalo RR médio e o seu desvio, isto é, a média do intervalo de tempo entre dois picos R consecutivos, frequência cardíaca, amplitude normalizada média e o seu desvio, assim como o intervalo do complexo QRS médio e o seu desvio, figura 11.



Designação	Valor
Intervalo RR médio(s)	0.844
Intervalo RR desvio padrão(s)	0.032
Frequência cardíaca(bpm)	50.651
Amplitude Normalizada R médio	0.891
Amplitude Normalizada R desvio padrão	0.077
Intervalo complexo QRS médio(s)	0.144
Intervalo complexo QRS desvio padrão(s)	0.0189

Fig. 11. Mensagem informativa de retorno do algoritmo do sinal ECG no intervalo de [20, 30]s.

## 4 Conclusão

A aplicação foi desenvolvida com sucesso, conseguindo implementar a leitura de um sinal ECG experimental, o tratamento do sinal removendo os ruídos associados a sua aquisição experimental usando vários tipos de filtros e obter um sinal ECG final no qual se possa analisar as várias ondas existentes e seus tempos de duração, reportando um relatório final.

A aplicação permite rapidamente visualizar o tempo total do sinal que se quer avaliar, bem como a frequência de amostragem e selecionar o intervalo pretendido para a sua análise. Dispõe de um conjunto de filtros que podem ser aplicados, e ainda possibilita a comparação de sinais entre si tanto no domínio do tempo, como no domínio das frequências. Ainda, de forma a facilitar a análise de um sinal ECG, a aplicação permite detectar os picos R e, por sua vez, os pontos Q e S, assim como os intervalos médios de duração entre ondas, podendo o utilizador analisar de uma forma rápida o sinal ECG e concluir sobre o estado de saúde do paciente e qual a patologia presente.

No caso particular, do sinal ECG usado para teste, verificou-se que o filtro passa-alto, considerando na resposta no domínio dos tempos e em frequência do sinal filtrado, o filtro que apresenta o melhor resultado para o objetivo pretendido foi o filtro *Butterworth*. E, de acordo com a resposta em frequência, o filtro passa-baixo que se mostrou mais adequado foi o FIR, pelo facto de, para além de eliminar as frequências acima de 38 Hz, apresenta capacidade de atenuar as amplitudes das frequências importantes.

No entanto, a aplicação está desenvolvida para serem aplicados os vários filtros usualmente usados no tratamento do sinal ECG e comparar graficamente quais são os que de adequam ao sinal em estudo.

## Referências

- [1] A. L. Goldberger, Z. D. Goldberger, and A. Shvilkin, "Essential Concepts," in *Goldberger's Clinical Electrocardiography*, Elsevier, 2018, pp. 2–5. doi: doi.org/10.1016/j.procs.2017.09.138.
- [2] K. S. Nikita and K. P. Michmizos, "Physiological systems modeling, simulation, and control," in *Handbook of Research on Biomedical Engineering Education and Advanced Bioengineering Learning: Interdisciplinary Concepts*, 2012. doi: 10.4018/978-1-4666-0122-2.ch017.
- [3] Â. P. Ramos and B. S. Sousa, "Eletrocardiograma : princípios , conceitos e aplicações," *Cent. Estud. Fisiol. do Exerc.*, no. Figura 1, 2007.
- [4] C. D. M. Regis, L. G. Caldeira, and E. C. Gurjão, "Avaliação da Amostragem Compressiva em Sinais de ECG e Imagens Digitais," *Rev. Principia - Divulg. Científica e Tecnológica do IFPB*, vol. 1, no. 29, p. 95, 2016, doi: 10.18265/1517-03062015v1n29p95-104.
- [5] M. Maršánová, L. Nemcova, A. Smisek, R. Smital, L., & Vitek, "Brno University of Technology ECG Signal Database with Annotations of P Wave (BUT PDB)," *PhysioNet*, doi: 10.13026/hwvj-5b53.

# Migração de conteúdos em memória para Redis

Ricardo Coelho<sup>1</sup> e Emanuel Silva<sup>1</sup>

<sup>1</sup> ISEP – Instituto Superior de Engenharia do Porto  
{1191446, ecs}@isep.ipp.pt

**Resumo.** Atualmente, instituições financeiras como os bancos, necessitam de ter um sistema robusto e com boa performance devido a ser um setor importante e delicado. As tecnologias que residem dentro dos sistemas dos bancos vão ficando desadequadas e, portanto, é necessário um olhar atento às novidades tecnológicas com o objetivo de aproveitar os benefícios que as mesmas proporcionam. Este artigo apresenta uma solução para a renovação da arquitetura do sistema do banco Millennium BCP (MBCP) com o intuito de melhorar a gestão da memória ocupada durante o seu funcionamento (minimizar a memória ocupada) e garantir a não afinidade de sessões por parte das instâncias do sistema. Foram utilizadas tecnologias como Java 8, Redis, Jedis, Gson, JMX e Tomcat. Ao longo deste artigo são apresentadas todas as etapas realizadas na implementação da solução e os resultados obtidos. De acordo com a análise dos resultados, conclui-se que a solução desenvolvida cumpre os objetivos propostos e apresenta vantagens significativas para o sistema tornando-o mais eficaz e mais moderno.

**Palavras-chave:** Arquitetura do sistema, Cache de informação, Sessões de clientes, Memória do sistema.

## 1 Introdução

O setor bancário é uma área bastante delicada e importante para todas as pessoas. As tecnologias que nele existem tendem a ficar desatualizadas e, como tal, é preciso mantê-las atualizadas para que se consiga aproveitar os benefícios que elas trazem. O projeto apresentado neste artigo surgiu da possibilidade de implementar uma base de dados NOSQL (Redis [1]) no sistema do banco e, assim, conseguir libertar memória e consequentemente renovar a tecnologia da arquitetura transacional do Millennium BCP (MBCP). O contexto deste projeto foca-se numa parte mais arquitetural do sistema, o que envolve muita pesquisa e leva a pensar, como seria a melhor maneira de implementação das novas tecnologias dentro da aplicação do banco. Isto faz com que seja um trabalho desafiante e cauteloso tendo de pensar em todas as possibilidades para que não haja falhas nem aconteçam situações indesejadas ou mesmo consideradas graves.

### 1.1 Descrição do problema

O problema a resolver baseia-se na implementação de uma base de dados Redis com a expectativa de melhorar o desempenho dos sistemas do banco. Pretende-se assegurar a

não afinidade de sessões, isto é, fazer com que as sessões dos clientes sejam armazenadas fora do sistema, garantindo que os clientes não tenham de manter uma afinidade com as máquinas guardando as sessões na base de dados externa. Para além disso, pretende-se minimizar a ocupação de memória do sistema. Em acréscimo, visa-se a otimização da memória e dos acessos ao sistema junto com uma renovação tecnológica da arquitetura da plataforma transacional do MBCP (componente principal), abrindo caminho para uma possível reestruturação em microserviços futuramente.

## 1.2 Objetivos

Na sequência do problema em mãos, definiram-se que os principais objetivos que a base de dados Redis deveria cumprir, seriam os seguintes:

- Garantir a não afinidade de sessões – guardar as sessões dos clientes no Redis;
- Armazenar os dados das bases de dados do sistema no Redis;
- Armazenar os dados dos web services do sistema no Redis;
- Armazenar os dados dos ficheiros do sistema no Redis.

## 2 Estado da Arte

Com o objetivo de tentar entender as melhores maneiras de gerir as sessões dos clientes, é importante explicar o que é a afinidade de sessões visto que o pretendido é excluir essa relação com as instâncias do sistema. A afinidade de sessão consiste em melhorar o desempenho da aplicação guardando em cache de memória *cookies* com a informação da sessão de um cliente, mais concretamente, as credenciais de login. Assim, sempre que um cliente pretenda solicitar algo e a sessão dele não tenha expirado, todos os seus pedidos irão conter a sua sessão devido à mesma estar armazenada em cache [2].

Com a evolução da tecnologia e o progresso dos sistemas da atualidade, o volume de dados que um sistema acumula ao longo do tempo é, potencialmente, bastante elevado. Neste sentido, é importante relembrar a forma como as informações são armazenadas. A Fig. 1 mostra a hierarquia da memória do sistema, separando o armazenamento do computador em 3 patamares que são baseados no tempo de resposta, complexidade e capacidade [3].

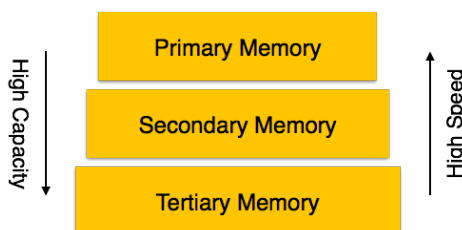


Fig. 1. – Hierarquia de memória

Com base na Fig. 1, é possível verificar que existem três locais de armazenamento de dados. A memória primária (Primary Memory) é a que guarda menos informação, mas por sua vez é a mais rápida delas todas, devido ao facto de estar diretamente ligada ao processador (CPU). O armazenamento secundário (Secondary Memory) foca-se em armazenar dados para uso futuro ou backups. A memória terciária (Tertiary Memory) é onde se guardam enormes volumes de dados [4]. O uso da base de dados NoSQL Redis traz algumas vantagens frente a uma base de dados relacional. Só o facto de ser estruturada por key-value já é uma boa vantagem, porque cada key contém a sua informação, quer seja, strings, números ou objetos complexos. É como se fosse uma base de dados relacional, mas com apenas duas colunas. Contudo, o principal motivo do uso deste servidor externo é porque guarda toda a sua informação na memória primária e as bases de dados relacionais na secundária [5]. Para além disso, a tecnologia Redis era conhecida pelo banco que também já possuía o licenciamento da base de dados para a sua utilização.

### 3 Análise e desenho da solução

A análise de requisitos explora a recolha de informação fundamental, de forma a ser possível o desenvolvimento de uma solução que responda ao problema que se destina a resolver. Numa fase inicial, são então levantados os requisitos funcionais e não funcionais da solução. É com base nestes requisitos que são tomadas decisões no que toca a performance e forma de implementação da solução. Para além disso, é apresentada uma perspectiva mais técnica do problema, onde é realizada uma descrição mais detalhada e completa da solução que inclui não só os métodos, técnicas e tecnologias utilizadas, mas também o processo de compreensão do problema.

#### 3.1 Requisitos funcionais e não funcionais

Para assegurar um bom entendimento e compreensão por todas as partes interessadas, foram realizadas reuniões semanais com o cliente. Os requisitos foram sofrendo mudanças ao longo do tempo e resultaram nos seguintes casos de uso:

- UC1 - Armazenar e/ou fornecer as sessões dos clientes no Redis;
- UC2 - Armazenar e/ou fornecer os dados das bases de dados do sistema no Redis;
- UC3 - Armazenar e/ou fornecer os dados dos web services do sistema no Redis;
- UC4 - Armazenar e/ou fornecer os dados dos ficheiros do sistema no Redis.

Neste projeto pretende-se que o sistema seja responsável pela automatização de todos os requisitos e sempre que alguma informação seja necessária, a sua recolha e a sua gravação seja eficaz.

#### 3.2 Requisitos não funcionais

Os requisitos não funcionais, descrevem os limites e características de qualidade que a solução deve possuir. Os requisitos não funcionais apresentados seguem a especificação FURPS+, “um modelo para classificação de atributos de qualidade de software”

[6]. Cada letra representa uma categoria sendo estas Funcionalidade, Usabilidade, Reliabilidade, Performance, Suportabilidade e o “+” representa outros aspetos como Restrições de Design, Requisitos de Implementação, Requisitos de Interface e Requisitos Físicos. Os requisitos considerados mais relevantes são:

- Requisitos de Design (+):
  - Adotar as boas práticas de desenho de software;
- Requisitos de implementação (+):
  - Utilizar a base de dados Redis;
  - Implementar a solução em Java;

### 3.3 Casos de uso

- UC1 – Armazenar e/ou fornecer as sessões dos clientes no Redis

Tem o objetivo de representar o fluxo que a solução efetua. Para a sua representação foi elaborada uma figura (Fig. 2) simplista com a sequência de passos que o sistema realiza para obter a solução pretendida. Este fluxo conta com um cliente a pretender aceder aos serviços do banco (PTC - componente em que a solução se situa) e, portanto, o seu JSESSIONID é utilizado como key para a identificação do cliente. A sua respetiva sessão será guardada como value em Redis para que posteriormente o utilizador possa aceder aos serviços desejados.

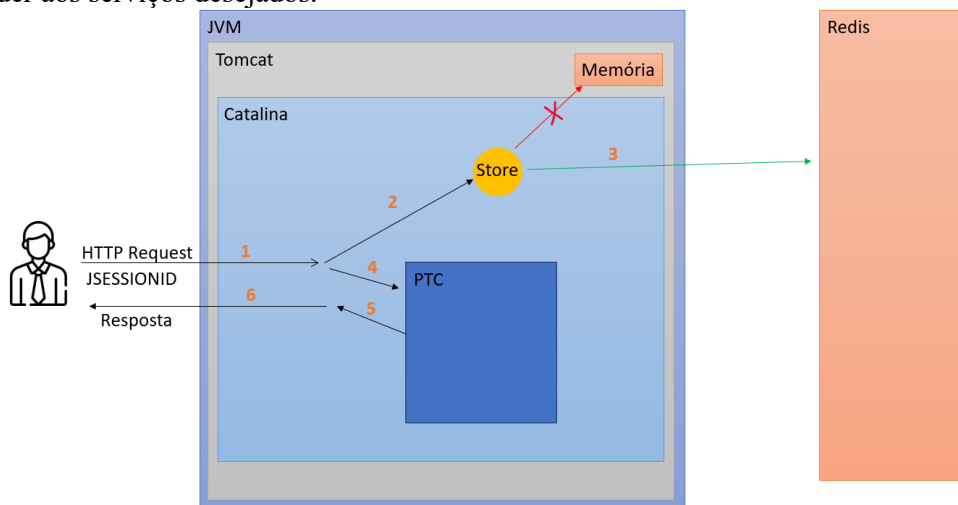


Fig. 2. – Fluxo de execução UC1

- UC2, UC3, UC4 – Armazenar e/ou fornecer informação (base de dados, web services e ficheiros) em Redis

Para estes casos de uso o que se pretende é guardar e fornecer informação útil que os clientes necessitem em Redis. Para que isto aconteça, o sistema executa várias etapas. Começa por verificar se o tempo para atualizar os dados expirou. Em caso afirmativo, é efetuada a atualização da toda a informação e continua para o passo seguinte, caso contrário, esta passo é ignorado e salta à frente. Após a verificação, o sistema procede para

a próxima etapa que será gerar uma *key*, para ser utilizada na pesquisa em Redis e nos campos necessários para a requisição da informação. Por fim, procura no servidor externo (Redis) a informação desejada com a *key* e os campos gerados.

## 4 Implementação

As tecnologias utilizadas para esta implementação, em conformidade com os requisitos não funcionais impostos, focam-se no desenvolvimento da solução em Java, mais precisamente na versão 8. No que toca à base de dados, foi utilizado o Redis, outro dos requisitos não funcionais obrigatórios. Por fim, foram utilizadas bibliotecas Java para a integração de outras tecnologias como Gson para a serialização e deserialização de objetos e o Jedis um cliente Java para consumir a API de Redis.

As tecnologias referidas, são amplamente conhecidas e utilizadas no desenvolvimento de software da atualidade.

### 4.1 Especificações, vantagens e funcionalidades importantes

Um dos pontos importantes desta solução é o facto de o servidor Redis ter sido instalado numa máquina da responsabilidade do banco com o sistema operativo Linux.

Uma das vantagens de usar Jedis é, portanto, a compatibilidade com a tecnologia JMX, onde a sua implementação é inicializada juntamente com o Jedis e com isto, os recursos do sistema são controlados e geridos com o intuito de perceber se existe alguma complicação ou não. A Fig. 3 ilustra informação proporcionada pelo JMX.



Fig. 3. - Representação da tecnologia JMX com estatísticas do Sistema

No decorrer da implementação desta solução foram adotados padrões de arquitetura de software no seguimento da adoção de boas práticas, como por exemplo os princípios básicos de programação orientada a objetos (POO), onde as suas características se refletem principalmente no Encapsulamento, Herança e Polimorfismo. Com estas práticas, o objetivo de não haver duplicação de dados e caso algum dia seja necessário trocar de tecnologias, as alterações que serão feitas ao código serão, potencialmente, menores e mais fáceis.

## 4.2 Armazenar e/ou fornecer as sessões dos clientes no Redis (UC1)

A solução conta com a integração direta do Tomcat com o Jedis. Desta forma, o próprio servidor da aplicação consegue gerir as sessões em Redis de forma transparente para a aplicação e, conseqüentemente, para o programador.

Para realizar tal solução, foi desenvolvida uma `store1` personalizada para o objetivo pretendido dentro da componente `Catalina2` do Tomcat. Com essa `store` o servidor deixa de guardar os dados em memória, definido por padrão, e passa a guardar a informação pretendida na base de dados Redis. Esta configuração pode ser efetuada através da alteração do ficheiro `context.xml`, tal como ilustrado na Fig. 4.

Com a configuração adequada do servidor, e após a chegada de um pedido a esse servidor, a `store` personalizada desenvolvida recolhe a `key` passada no pedido, mais especificamente, no `http header jsessionid` e verifica se a mesma existe no Redis ou não. Caso não exista, essa `key` é armazenada no Redis, serializada em `Json`, podendo assim realizar o pedido solicitado e posteriormente qualquer outra instância ter acesso à sessão do cliente.

```
<!-- Uncomment this to disable session persistence across Tomcat restarts -->  
<Manager className="org.apache.catalina.session.PersistentManager">  
  <Store className="net.bcpcorp.ptc.catalina_store.JedisStore" redisAddress="redis://127.0.0.1:6379" />  
</Manager>
```

Fig. 4. - Configuração para o Tomcat utilizar a `store` personalizada

Para além disto, é necessário adicionar a `store` e todas as dependências, incluindo as dependências transitivas das mesmas, ao `class path` do Tomcat, para que o mesmo consiga usufruir da componente desenvolvida.

Relativamente à serialização utilizada, a mesma pode assumir qualquer tipo de formato, sendo que a usada foi `Json`.

É importante referir também que, por padrão, a serialização é feita quando o Tomcat é desligado, o que não ia ao encontro com o que era pretendido. Desta forma, foi utilizada uma `vault4`, configurada no `server.xml` (ficheiro de configuração do Tomcat), que faz com que a serialização seja realizada a cada pedido solicitado. Com isto, consegue-se obter uma solução genérica para funcionar em qualquer projeto onde se pretenda guardar sessões externamente ao sistema.

## 4.3 Armazenar e/ou fornecer os dados das bases de dados/web services/files do sistema no Redis (UC2, UC3 e UC4)

Esta solução conta com o uso de classes abstratas que foram desenvolvidas para tratar dos 3 tipos de dados que existem, sendo estes de base de dados, `web services` e ficheiros. O objetivo é armazená-los dentro do Redis fazendo com que qualquer instância do sistema tenha acesso a toda a informação, levando assim, a um alívio de memória no sistema.

Relativamente à forma como os dados são organizados dentro do Redis, cada tipo de dados tem o seu formato o que os torna fáceis de identificar. Como tal, a `key` pertencente a uma informação de base de dados é diferente de um ficheiro, `web service` ou até

mesmo de uma sessão de um cliente. Cada value de uma key contém a informação dentro de um HashMap onde a mesma pode ser obtida através de índices únicos (entity) e não únicos (collection). Estes índices ajudam a obter os dados pretendidos sendo estes objetos ou listas de objetos Java, respetivamente. Desta forma, toda esta informação permanece partilhada entre instâncias e torna o objetivo possível.

Outro aspeto importante sobre as caches, é o facto de as classes abstratas possuírem uma funcionalidade de refrescamento, que no fundo, faz com que os dados dentro do Redis sejam atualizados. O desenvolvimento desta funcionalidade usa comandos que o Jedis implementa, ou então, exporta um ficheiro para o servidor onde o Redis se situa e a partir daí o servidor trata da atualização da informação. O sistema conta com 3 tipos de refrescamento de dados:

1. o refrescamento passivo que é efetuado quando o sistema recebe um pedido e caso “x” tempo tenha passado;
2. o refrescamento ativo que pode ser invocado por qualquer classe;
3. o refrescamento externo que é realizado através de exportação de um ficheiro em formato .RESP<sup>1</sup>, que é copiado/uploaded para a máquina do servidor Redis. Após isso, os dados são carregados por Bulk Loading<sup>2</sup> via scripts executados pelo cron<sup>3</sup> para a base de dados.

#### 4.4 Testes

Para testar as sessões foi utilizada a ferramenta Postman, podendo assim, simular pedidos REST para o servidor Tomcat e conseqüentemente guardar a sessão do pedido em Redis e fornecê-la às instâncias do sistema. O resultado destes testes foi positivo e, com as sessões de clientes a serem armazenadas, proporcionou-se um melhor funcionamento de Load Balancing ao sistema.

Em relação, às caches, as funcionalidades implementadas foram testadas de maneira a que a informação retornada ao cliente e armazenada em Redis fosse ao encontro do objetivo pretendido.

Após a conclusão dos testes foi verificado na base de dados Redis se todas as keys pretendidas estariam criadas, sendo que o resultado foi ao encontro do que era esperado. Foi também testado o refrescamento de dados com um ficheiro de grande dimensão.

O resultado foi que o refrescamento passivo e ativo não conseguiu atualizar os dados devido a um erro de timeout de 5 minutos. Por outro lado, o refrescamento externo ao contrário dos outros dois, atualizou os dados em menos de 2 segundos. Para além do refrescamento externo conseguir melhores resultados face aos outros, evita-se preocupações de sincronismos, como por exemplo problemas de concorrência em que uma instância possa guardar informação ao mesmo tempo e acabe por destruir dados.

---

<sup>1</sup> RESP é uma extensão de arquivo que está associada a arquivos Microsoft Windows Report Log.

<sup>2</sup> “Bulk loading is the process of loading Redis with a large amount of pre-existing data.” [9]

<sup>3</sup> “O cron pode ser interpretado como um serviço do Linux que é carregado durante o processo de boot do sistema.” [10]

## 5 Conclusões

No que toca aos requisitos identificados, estes foram todos alcançados, sendo esta afirmação comprovada pelos testes funcionais apresentados na secção 4.4. Para além disto, a solução foi apresentada à entidade do banco responsável pelo projeto para avaliação/validação e o mesmo aprovou tudo o que foi realizado. É importante referir que a solução desenvolvida ainda vai passar por uma fase adicional de testes de aceitação após a entrega da mesma, para que quando chegue a produção não haja qualquer tipo de bug durante o seu funcionamento. Os Resultados obtidos foram ao encontro do que foi planeado. Com esta solução, o sistema do banco possuiu uma versão mais inovada da sua cache com todas as vantagens que o Redis pode trazer.

Em relação aos resultados obtidos por parte dos métodos de refrescamento, ativo e passivo, concluiu-se que, para atualizar uma grande quantidade de dados, poderão não ser completamente confiáveis e, portanto, recomenda-se a utilização do processo de refrescamento externo. Relativamente a pequenas e médias quantidades de dados, qualquer método utilizado é seguro e confiável para atualizar dados do sistema em Redis. Para trabalho futuro, esta solução conta com a instalação de Redis em produção, migrar progressivamente classes existentes para a nova framework e deixa em aberto a possibilidade de uma reestruturação do sistema para microserviços.

## Referências

1. "Redis", <https://redis.io/>, acedido em 2022/09/29.
2. "IBM DataPower Gateway", <https://www.ibm.com/docs/pt-br/datapower-gateway/2018.4?topic=groups-session-affinity>, acedido em 2022/09/13.
3. "Hierarquia de memória", [https://stringfixer.com/pt/Memory\\_hierarchyprim%C3%A1ria](https://stringfixer.com/pt/Memory_hierarchyprim%C3%A1ria), acedido em 2022/09/29.
4. "DBMS - Storage System", [https://www.tutorialspoint.com/dbms/dbms\\_storage\\_system.htm#](https://www.tutorialspoint.com/dbms/dbms_storage_system.htm#), acedido em 2022/09/29.
5. "Redis vs RDBMS", <https://www.javatpoint.com/redis-vs-rdbms>, acedido em 2022/09/29.
6. "FURPS", <https://pt.wikipedia.org/wiki/FURPS>, acedido em 2021/10/10.
7. "Interface Store", Apache Tomcat, <https://tomcat.apache.org/tomcat-8.0-doc/api/org/apache/catalina/Store.html>, acedido em 2022/09/29.
8. "Meet Tomcat Catalina", <https://www.mulesoft.com/tcat/tomcat-catalina>, acedido em 2022/09/29.
9. "web-servers / tomcat-vault", [https://github.com/web-servers/tomcat-vault/blob/main/Using\\_Crypt.md](https://github.com/web-servers/tomcat-vault/blob/main/Using_Crypt.md), acedido em 2022/09/29.
10. "Bulk Loading", <https://redis.io/docs/reference/patterns/bulk-loading/>, acedido em 2022/09/29.
11. "Linuxcron", <https://www.infowester.com/linuxcron.php>, acedido em 2022/09/29.

## Sistema de apoio à gestão da Porto Design Factory

Joel Costa, Constantino Martins e Jorge Duarte

Instituto Superior de Engenharia do Porto, Porto, PT  
{1120844,acm,fjd}@isep.ipp.pt

**Resumo.** Este trabalho apresenta toda a análise que foi feita para resolver um problema apresentado pela entidade Porto Design Factory (PDF) que é uma plataforma educacional experimental de cocriação para a educação interdisciplinar, a inovação e a aceleração de novas ideias, produtos e negócios.

A PDF é o catalisador de uma cultura de ensino experimental assente na *passion based learning*, visando a promoção de um conjunto de projetos multidisciplinares que coordena e nos quais se envolve. A PDF necessitava de uma aplicação de apoio à gestão que se enquadrasse nas características da própria PDF. Pretendia que essa aplicação a ajudasse a ultrapassar as barreiras sentidas no seu atual sistema de gestão interno face ao crescimento acelerado de projetos e equipas. Ambicionava que a aplicação lhe permitisse crescer mais para alocar mais projetos e mais equipas, em desenvolvimento de novos produtos inovadores que podem modelar o futuro próximo. Por outro lado, a ferramenta permitiria também que a sua equipa conseguisse estar mais focada em assuntos realmente importantes.

Este trabalho compreendeu o levantamento de trabalhos existentes no mercado na área do pretendido desenvolver, um estudo que englobou as frameworks e linguagens de programação existentes, capazes de dar resposta ao desenvolvimento da ferramenta. Compreendeu também a análise, design e implementação da aplicação de forma a satisfazer os requisitos pretendidos. O feedback dos utilizadores foi positivo, o que comprova a utilidade da aplicação.

**Palavras-chave:** Plataforma, Domínios, Casos de uso, Linguagens, *Framework*, Aplicações.

### 1 Introdução

A PDF deparou-se com a dificuldade em saber o estado de utilização de recursos humanos e materiais. O não conhecimento do estado dos recursos dificultava a otimização dos serviços e o desenvolvimento dos projetos multidisciplinares em curso.

Por esse facto, foi necessário desenvolver uma ferramenta que lhe confira uma melhor capacidade da gestão dos vários recursos humanos e materiais.

Neste trabalho foi elaborado um conjunto de estudos e trabalho de análise para o desenvolvimento de uma aplicação capaz de dar resposta às necessidades da PDF.

Neste trabalho foi feito um estudo de mercado para verificar a existência de uma aplicação que conseguisse dar resposta ao problema da PDF, de uma forma global.

foi realizado um estudo que engloba as tecnologias, *frameworks* e linguagens de programação, existentes capazes de dar resposta ao desenvolvimento de uma aplicação.

## 2 Análise de valor

Pretendeu-se com este trabalho, gerir os principais recursos materiais e intangíveis com os quais a PDF tem a constante necessidade de interagir e saber o seu estado.

A análise de valor permitiu explorar todo o mundo conceptual e auxiliou a direccionar a produção de uma aplicação orientada para as grandes necessidades que eram sentidas pelos vários utilizadores existentes na PDF.

Foi necessário obter informação sobre os fundamentos inovadores e diferenciadores que podiam ser agrupados neste trabalho. Nesse sentido, foi analisado um conjunto de trabalhos cujos pontos de vista ajudaram na idealização e maturação conceptual de valor [1] [2] [3] [4]. Estes trabalhos permitiram fazer a filtragem das ideias que foram tidas, focando-se numa saída de um produto que estivesse dotado de um valor real para o cliente e por conseguinte com valor acrescentado para a entidade, além de se manter a componente inovadora.

Foi realizada a análise no ponto de vista do negócio, de forma a se conseguir reduzir custos e aumentar os lucros. Nesse sentido e numa perspectiva mais virada para a área dos negócios, foi analisado e criada a análise o modelo CANVAS.

No modelo identificou-se os parceiros, as principais atividades e a prestação de serviços a empresas. Identificaram-se os principais custos para o projeto e a sua manutenção.

Da análise de valor em conjuntos com o modelo CANVAS, foi extraído que a entidade PDF pode beneficiar com a utilização de um sistema de gestão de todos os domínios que estão associados ao problema. Só assim, a PDF consegue ter múltiplos projetos em execução, dar respostas em tempo real às requisições de serviços e manter bem presente o estado dos seus recursos materiais, equipamentos, salas, ajudas técnicas, gestão de custos, gestão de pessoas e gestão de projetos.

Ao nível das necessidades identificadas no problema apresentado, entre outras, podem ser referidas, a necessidade de promover melhores resultados nos projetos, a diferenciação das interfaces para os vários utilizadores, o acesso à aplicação de qualquer localização do globo e o suporte também para dispositivos móveis (facilitando a mobilidade do utilizador).

## 3 Estado da Arte

Foram analisadas as aplicações que o mercado tem disponível que pudessem satisfazer as necessidades deste projeto e que são apresentadas na tabela 1. Na tabela 1, o símbolo “-” representa que a aplicação não é do domínio, enquanto o símbolo “X” significa que a aplicação é do domínio.

Nesta análise concluiu-se que algumas das aplicações conseguiam dar resposta a alguns dos domínios necessários, mas não foi possível encontrar uma aplicação que pudesse dar resposta à totalidade dos requisitos do projeto. Como tal, verificou-se a seria

necessário desenvolver um produto novo, à medida do modelo de negócio e que fosse capaz de dar resposta às necessidades existentes. Para o desenvolvimento da aplicação foi necessário fazer-se um estudo para identificar tecnologias apropriadas para esse desenvolvimento.

Foi analisado também um conjunto de frameworks desenvolvidas com recurso a um conjunto de linguagens e que se apresentam como boas ferramentas para a produção de código, reutilização de código.

Aplicação	Equipamentos	Materiais	Projetos	Gestão Pessoas	Gestão Custos
Slack	-	-	-	X	-
Avaliação	-	-	-	X	-
Hondana Indica	-	-	-	X	-
Runrun.it	-	-	-	X	-
Wrike	-	-	X	X	-
Pontomais	-	-	-	X	-
Gupy	-	-	-	X	-
Biz.u	-	-	-	X	-
Recruta simples	-	-	-	X	-
RH1000	-	-	-	X	-
Felizz	-	-	-	X	-
Folha Certa	-	-	-	X	-
ERP Centralgest	X	X	-	X	X
ERP SBG	-	-	-	X	-
Bitrix 24	-	-	-	X	-
Social Base	-	-	-	X	-
Trello	-	-	-	X	-
Wunderlist	-	-	-	X	-
Latpass	-	-	-	-	-
Dropbox	-	-	-	-	-
Drive Fx	X	X	-	-	-
Sistrade	-	-	-	-	X
Quick Books	X	X	-	-	X
Nibo	-	-	-	X	X
Ganatum	X	X	-	-	X
Bkper	-	-	-	-	X

Tabela 1 Relação das aplicações aos domínios do problema [5][6]

Framework	Web	Móvel	Comunidade Ativa	Javascript	CSS	HTML
Springboot	X	X	-	X	-	-
Botstrap	X	-	X	X	X	X
Cordova	-	X	-	-	-	-
Angular	X	-	-	-	-	-
React	-	-	-	X	-	-
Ionic	X	X	-	-	X	X
Pure	X	-	-	X	X	X

Tabela 2 Comparação de Frameworks [7][8]

Esse conjunto de *frameworks* está resumida e apresentada na tabela 2. Na tabela 2 pode-se notar que o símbolo “-” indica que a *framework* não tem essa característica, enquanto o símbolo “X” indica que satisfaz a característica analisada.

Concluiu-se que a *framework* mais adequada para ser usada para um projeto deste tipo seria o Bootstrap, já que seria compatível com um conjunto de linguagens para o desenvolvimento web, como é o caso do JavaScript CSS e o HTML.

Foi realizado um estudo sobre as linguagens de programação para perceber se eram compatíveis com tecnologias web, se tinham aptidão para o desenvolvimento de bases de dados e se tinham potencial para serem usadas num ambiente CRUD. As informações recolhidas para o estudo provieram das fonte [9][10], [11], [12], [13], [14]. Na tabela 3, o símbolo “-” indica que a mesma não é adequada enquanto o símbolo “X” indica que é adequada.

Linguagens	Web	BD	CRUD
Javascript	X	-	X
PHP	X	-	X
SQL	-	X	-
C	-	-	-
C++	-	-	-
C#	X	-	X
Python	X	-	-
Ruby	X	-	-
Java	X	-	-

Tabela 3 Comparação de linguagens.

Das linguagens de programação analisadas somente o SQL é adequada para o desenvolvimento de um sistema de base de dados relacional e por isso foi usada no projeto. O JavaScript, o PHP e o C# permitiam o desenvolvimento de *frontend*. Não se optou por fazer uso do C# uma vez que projetos deste tipo exigem uma constante compilação do projeto. O JavaScript levantava uma questão problemática, já que coloca a lógica do negócio do lado cliente. O PHP contrariamente ao JavaScript, consegue manter a lógica do negócio contida no servidor e como tal foi a escolhida.

Neste trabalho foram identificados 6 grupos distintos de utilizadores, sendo eles o sistema, diretor, técnico de suporte, gestor de stock, aluno e o *teaching assistant*.

Na análise do problema foi identificado o domínio “salas”, “equipamentos”, “materiais”, “projetos”, “gestão de pessoas”, “ajuda técnica”, “gestão de custos” e “softwares”. Para cada um dos domínios foi desenvolvido um CRUD personalizado para cada um dos grupos de utilizadores.

## 4 Arquitetura

Ao nível da arquitetura foi escolhido para este projeto, o modelo de arquitetura cliente servidor. Este adequa-se perfeitamente à necessidade de se ter vários clientes que acedem a um ponto centralizado onde está contida toda a informação.

A figura 1 apresenta o diagrama de componentes da aplicação. Nesta aplicação os vários componentes escondem toda a complexidade desses domínios. É possível verificar que componentes dependem de outros e quais consomem serviços dos outros. É possível também perceber que o componente que representa o projeto, é o ponto de interligação de todos os outros que tem acesso a toda a informação.

Sobre o componente “salas” recai todas as atividades relacionadas com salas tal como criar, editar e eliminar sala. Recai também atividades, como a requisição de “salas” e a consulta do estado das “salas”, entre outros.

No caso do componente “equipamentos” tem sobre a sua responsabilidade, atividades como criar, editar e eliminar equipamento, requisitar equipamentos e o estado dos pedidos, entre outros.

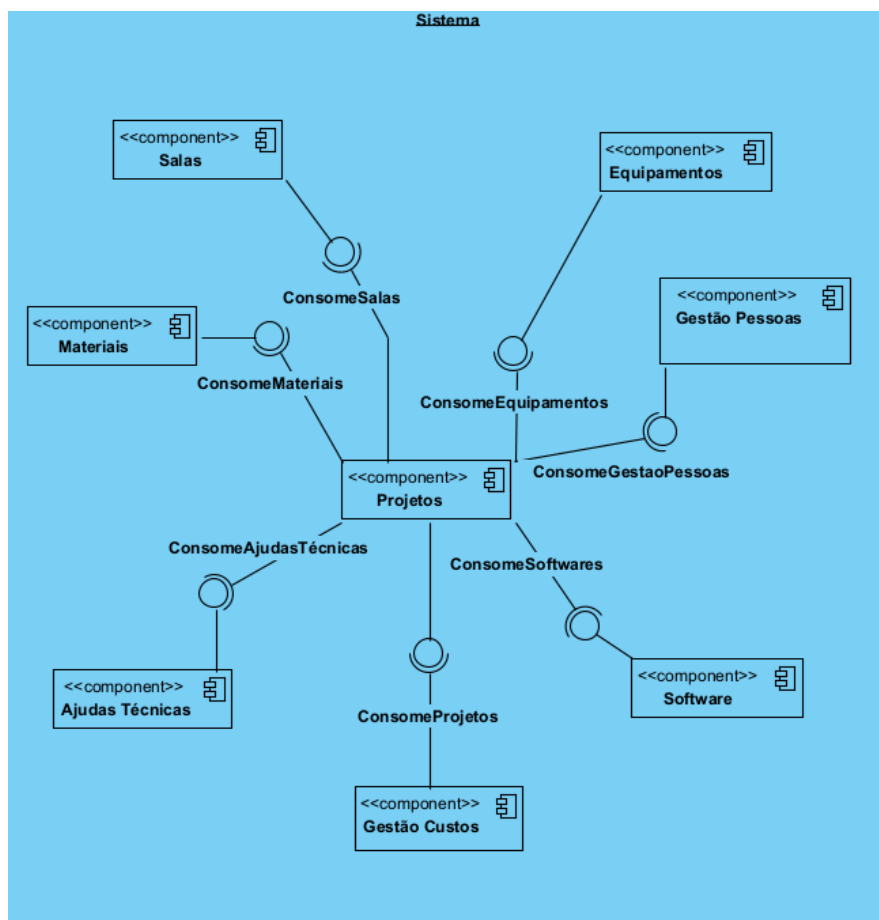


Figura 1 Diagrama de Componentes.

No caso do componente “materiais” recai sobre ele algumas atividades como criar, editar e eliminar material, pedidos de material, estado de pedido de material, entre outros. Para o componente de “gestão de pessoas” recaem atividades como criar pessoas e criar equipas, entre outras. Para o componente das “ajudas técnicas” recai toda a

complexidade associada a este domínio, como por exemplo, requisitar uma ajuda técnica ou o ver o estado do pedido de ajuda técnica, entre outras.

No componente “software” recaem responsabilidades como criar software, editar software, fazer um pedido de software e consultar o estado de um pedido de software, entre outros. Para o componente “gestão de custos” recai a responsabilidade de gerir os custos associados a todas as atividades que ocorreram durante a vida de um projeto.

## 5 Principais funcionalidades

Este projeto englobou 172 casos de uso. Como principais funcionalidades implementadas, podem ser referidas algumas, a título de exemplo.

Estão também presentes funcionalidades para “criar um projeto” e “criar uma equipa” que pode ser, entretanto, atribuída a um projeto.

As funcionalidades “pedir material”, “pedir sala” e “pedir equipamento” permitem ao utilizador dar encaminhamento ao seu projeto. Estão também presentes funcionalidades que permitem “anular um pedido de material”, “anular um pedido de sala” e de “anular um pedido de equipamento”, bem como estão presentes funcionalidades que permitem acompanhar o estado dos vários pedidos realizados.

Existem funcionalidades que permitem “bloquear material”, “bloquear sala” e “bloquear equipamento” para que eles não possam ser requisitados. Estão também presentes as funcionalidades que permitem “reativar equipamento”, “reativar sala” e “reativar equipamento” que estavam bloqueados.

Existem funcionalidades que permite “adicionar um estudante a uma equipa” e “remover um estudante de uma equipa”.

A funcionalidade “consultar os custos do projeto” permite ter a noção do estado dos custos do projeto, de forma a permitir, caso necessário, perspetivar um eventual financiamento para a conclusão do projeto.

Existem também um conjunto de funcionalidades que permitem, numa fase final do projeto, obter toda a informação dos recursos e respetivos custos para a execução do projeto.

## 6 Resultados

Através de inquéritos de entrevista semiestruturados, foi testado junto dos utilizadores se estes conseguiam de uma forma expedita, localizar e testar as seguintes funcionalidades: “criar material”, “criar equipamento”, “pedir material”, “pedir equipamento”, “pedir software”, “pedir uma sala” e “consultar os custos do projeto”. Foi possível verificar que os utilizadores conseguiram com sucesso e expeditamente localizar e testar essas funcionalidades.

## 7 Conclusões

No estudo realizado concluiu-se que ao nível das aplicações estudadas, nenhuma tinha a capacidade de responder a todas as necessidades identificadas.

O desenvolvimento de uma solução personalizada foi a opção seguida para dar resposta à grande abrangência do problema.

Das *frameworks* analisadas, o BootStrap era a mais adequada para o ambiente pretendido e como tal foi adotada. Relativamente às linguagens de programação, as linguagens adotadas por serem as mais adequadas foram CSS, PHP e HTML. O CSS e o HTML fazem parte da composição da *framework* BootStrap.

Ao nível de resultados obtidos foram bastantes positivos já que mediante entrevistas semiestruturadas realizadas aos utilizadores da aplicação, conseguiu-se perceber a sua aptidão para localizar e testar algumas das funcionalidades de forma expedita.

## Referências

- [1] P. a. A. G. a. B. R. a. C. A. a. D. J. a. D. R. a. E. C. a. H. K. a. I. M. a. J. A. a. K. R. a. S. R. a. S. A. a. Koen, “Providing Clarity and Common Language to the Fuzzy Front End,” *Research-Technology Management*, vol. 44, pp. 46 - 55, 2001.
- [2] T. Woodall, “Conceptualising ‘ Value for the Customer ’ : A Structural , Attributional and Dispositional Perspective Why ‘ Value for the Customer ’ ?,” 2003.
- [3] B. M. M. H. D.-. W. M. Nick Rich, “INNOREGIO: dissemination of innovation and knowledge management techniques”.
- [4] J. Lapierre, “Customer-perceived value in industrial contexts,” *Journal of Business & Industrial Marketing*, vol. 15, pp. 122 - 145, 2000.
- [5] G. Dias, “Os 18 melhores softwares e aplicativos para RH que facilitam sua rotina,” 2018. [Online]. Available: <https://www.gupy.io/blog/melhores-aplicativos-para-rh>. [Acedido em 16 09 2019].
- [6] M. Ribeiro, “6 opções de software de gestão financeira para pequenas empresas,” [Online]. Available: <https://pluga.co/blog/financeiro/software-de-gestao-financeira/>. [Acedido em 17 09 2019].
- [7] E. Cronapp, “Conheça 7 frameworks que tornam mais rápido o desenvolvimento de softwares,” 21 02 2019. [Online]. Available: <https://www.cronapp.io/pt-br/conheca-7-frameworks-que-tornam-mais-rapido-o-desenvolvimento-de-sofware/>. [Acedido em 21 02 2019].
- [8] E. Scudero, “Os 6 frameworks front-end mais amados no mundo (segundo o GitHub),” 19 02 2018. [Online]. Available:

- <https://becode.com.br/frameworks-front-end-mais-amados-segundo-github/>. [Acedido em 21 02 2019].
- [9] T. iee.org, “Interactive: The Top Programming Languages,” 2014. [Online]. Available: <https://spectrum.ieee.org/static/interactive-the-top-programming-languages/#index/1/0/1/0/1/50/1/50/1/50/1/30/1/30/1/30/1/20/1/20/1/5/1/5/1/20/1/100/>. [Acedido em 10 09 2019].
- [10] T. TIOBE, “TIOBE Index for September 2019,” 2019. [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Acedido em 11 09 2019].
- [11] P. Carbonnelle, “PYPL PopularitY of Programming Language,” 2019. [Online]. Available: <http://pypl.github.io/PYPL.html>. [Acedido em 11 09 2019].
- [12] E. sistrade, “Melhore as actividades financeiras da sua empresa,” [Online]. Available: <https://www.sistrade.com/pt/solucoes/area-administrativa-financeira/gestao-financeira/>. [Acedido em 16 09 2019].
- [13] E. Pontomais, “Softwares de gestão de pessoas: 4 sistemas que você precisa conhecer!,” 2018. [Online]. Available: <https://pontomais.com.br/blog/softwares-de-gestao-de-pessoas-que-voce-precisa-conhecer/>. [Acedido em 13 09 2019].
- [14] T. G. Trends, “{<https://trends.google.pt>,” 2019. [Online]. [Acedido em 11 09 2019].
- [15] H. M. J. B. e. E. K. Peter A. Koen, “Managing the Front End of Innovation Part I: Results From a Three-Year Study,” *Research-Technology Management* 57.2., pp. 34-43, 2014 a.
- [16] H. M. J. B. e. E. K. Peter A. Koen, “Managing the Front End of Innovation Part II: Results from a Three-Year Study,” *Research-Technology Management* 57.3, pp. 25-35, 2014 b.
- [17] P. a. M. A. G. a. B. S. a. C. A. a. F. E. a. F. S. a. J. A. a. P. P. a. S. R. Koen, “1 Fuzzy Front End: Effective Methods, Tools, and Techniques,” 2019.

---

# SMARTHEALTH - Sistema de Recomendação para Cuidados de Saúde Personalizados ao Longo da Vida

João Santos, Constantino Martins, and Luiz Faria

<sup>1</sup> Instituto Superior de Engenharia do Porto (ISEP) 1150639@isep.ipp.pt

<sup>2</sup> Grupo de Investigação em Engenharia e Computação Inteligente para a Inovação e o Desenvolvimento (GECAD)

**Resumo** Nos últimos anos as perturbações mentais tornaram-se um dos principais fatores na diminuição da qualidade de vida, tendo um grande impacto no quotidiano das pessoas afetadas e naqueles que a rodeiam. Portugal é um dos países mais afetados na Europa, onde mais de um quinto da população sofre de perturbações psiquiátricas.

Esta dissertação documentará a conceção de uma aplicação móvel, desenvolvida para ajudar as pessoas com perturbações de ansiedade ou que sofram de sintomas ligeiros gerais de ansiedade.

A aplicação móvel desenvolvida possui várias funcionalidades úteis na gestão da perturbação de ansiedade, entre as quais se destacam a realização de exercícios de relaxamento, registo diário de autoreflexões acerca do estado emocional, análise de sentimentos e recomendação personalizada de atividades. Tais ferramentas encontram-se disponíveis sempre que o utilizador necessitar, nomeadamente em situações com maiores picos de ansiedade e stress. Pretende-se com a utilização da aplicação auxiliar o utilizador a compreender melhor a sua perturbação psicológica, os sintomas a ela associados e os impactos resultantes no seu quotidiano, e a desenvolver estratégias de como melhorar a sua sintomatologia.

O processo de elaboração da aplicação referida consistiu na implementação de um sistema de recomendação de atividades, adaptado e personalizado a um determinado tipo de utilizador. As funcionalidades do sistema de recomendação permitem ao utilizador recorrer à realização de atividades de relaxamento, tendo em conta o impacto dos exercícios na melhoria da qualidade de vida, o feedback fornecido pelo utilizador na aplicação e os resultados na diminuição da perturbação, medidos através da realização de questionários e do registo diário de emoções.

**Palavras-chaves:** Aplicação Móvel, Sistema de Recomendação, Perturbação de Ansiedade, React Native, Javascript.

## 1 Introdução

### 1.1 Definição do Problema

Cada vez mais o conceito “ansiedade” constituiu uma palavra corrente no nosso vocabulário, tornando-se hoje em dia a perturbação mental mais comum em todo o mundo. Estima-se que mundialmente 33,7% das pessoas poderão ser afetadas pela doença em alguma fase da sua vida e Portugal inclui-se como um dos países mais afetados no que concerne a perturbações da ansiedade [1]. Segundo o Estado da Saúde Mental em 2016, as perturbações psiquiátricas afetam mais de um quinto da população portuguesa, sendo 6,06% desse valor, relativo a perturbações da ansiedade [2].

Com o novo contexto pandémico do vírus SARS-CoV-2, a prevalência da ansiedade aumentou e a qualidade de vida diminuiu, nos indivíduos sujeitos a isolamento profilático [3].

Atualmente vivemos na “Era da informação”, onde a tecnologia se torna a cada ano, mais rápida, amigável, acessível e emergente no nosso quotidiano, independente da faixa etária do seu consumidor. O mercado dos *smartphones* é um dos mais beneficiados com a evolução da tecnologia e em Portugal já são cerca de 96,8% dos portugueses que possuem telemóvel entre os quais 75% com acesso à Internet, indicam os dados da Anacom do estudo realizado em 2018 [4]. Desta forma, surgiu a oportunidade de recorrer à implementação de aplicações móveis orientadas para auto-intervenções, com promoção de redução da perturbação de ansiedade aos seus utilizadores e melhoria da qualidade de vida.

## 2 Perturbação de Ansiedade

Perturbações de ansiedade incluem transtornos que compartilham características de medo e ansiedade excessivos e distúrbios comportamentais relacionados [5].

A maioria das Perturbações de Ansiedade são desenvolvidas na infância, sendo normal a sua continuidade até à adolescência nos indivíduos que não tenham qualquer tipo de acompanhamento psicológico e/ou médico. A prevalência de Perturbações de Ansiedade é maior no sexo feminino quando comparado com o sexo masculino [6].

Este tipo de doença têm um impacto muito grande nas vidas das pessoas afetadas, prejudicando a sua vida de relação e vida profissional. Estudos revelaram, que as próprias famílias dos indivíduos afetados são sobrecarregadas com o peso e o impacto negativo da perturbação de ansiedade de forma indireta [7].

### 2.1 O que é a ansiedade?

O sentimento de ansiedade é comum no quotidiano de qualquer indivíduo, perante situações de medo ou expectativa. Esta pode até mesmo estar associada a um melhor desempenho humano, segundo a lei *Yerkes-Dodson* [8].

Podemos concluir que, até determinado nível de ativação, existe uma correlação com a melhoria da concentração, traduzindo-se num melhor desempenho

intelectual. Contudo à medida que o aumento da ativação e excitação ultrapassam valores considerados moderados e representam valores elevados, o nível de desempenho diminui progressivamente em relação à ativação. Somente neste período a ansiedade se torna deletéria para o indivíduo, contribuindo para um pior desempenho e manifestando-se através de sintomas físicos e psicológicos associados à doença.

Considera-se que a ansiedade corresponde a uma perturbação mental, quando esta prejudica o indivíduo, através de mal-estar físico e psicológico.

Sendo a Perturbação de Ansiedade considerada uma Perturbação Mental, e dada a variabilidade individual da manifestação da doença e da resposta às intervenções terapêuticas, torna-se mais difícil aplicar regras gerais de abordagem a toda a população. Logo diferentes pessoas tem diferentes níveis de ansiedade, das quais algumas terão mais tolerância face aos elevados níveis de ansiedade, enquanto outras pessoas podem ser mais suscetibilidade em desenvolver uma Perturbação de Ansiedade, sofrimento e mau estar geral e constante [9].

## **2.2 Tipos de ansiedade**

De forma a garantir uma melhor representação e sistematização dos sintomas associados, os especialistas definiram grupos específicos para determinados tipos de ansiedade e tipo de tratamentos. Os transtornos de ansiedade diferem uns dos outros nos tipos de objetos ou situações que induzem medo, ansiedade ou comportamento de evitação e a ideação cognitiva associada [5].

Uma pessoa pode ser diagnosticada com vários tipos de ansiedade, tendo por base diferentes abordagens de tratamento e intervenções clínicas.

1. Perturbação de Ansiedade Generalizada (PAG)
2. Perturbação de pânico
3. Fobias
4. Perturbação Obsessivo-Compulsiva (POC)
5. Perturbação de Stress Pós-Traumático (PSPT)

Este projeto desenvolvido tem como principal objetivo lidar e auxiliar o primeiro tipo de ansiedade, PAG.

## **3 Aplicação SmartHealth**

### **3.1 Vista Lógica**

Na vista lógica são abordados os principais componentes da aplicação e as relações entre si.

Permite ao Utilizador o uso de certas funcionalidades, como alguns exercícios e algumas atividades consideradas mais úteis e importantes, dependente do acesso ao Servidor. Isto dá-se a implementar alguns dos componentes diretamente na aplicação a desenvolver. A aplicação quando conectado ao Servidor, analisa e processa todos os dados, que por sua vez, comunica com a base de dados, de forma a armazenar todas as informações necessárias do utilizador e suas preferências.

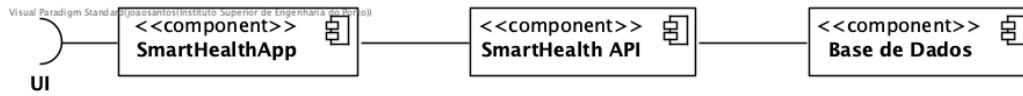


Figura 1. Vista Lógica da aplicação desenvolvida.

### 3.2 Funcionalidades da Aplicação

A aplicação móvel foi desenvolvida através do *React Native*, que se trata de uma *Framework* utilizada para desenvolvimento de aplicativos móveis multiplataforma. Dessa forma, tornou-se possível a criação de uma aplicação que suporta tanto o sistema do *Android* como o *iOS*, recorrendo-se apenas a linguagem de programação *Javascript*.

**FP1 – Iniciar Sessão:** Utilizador tem de estar registado previamente no sistema para que a funcionalidade seja efetuada com sucesso. Na página principal da aplicação são solicitados os dados relativos à autenticação do utilizador, com base no seu email e a palavra-passe correspondente.

De realçar a necessidade de utilizar uma biblioteca externa para processar os métodos de criptografia, tais necessidades provêm para que todas as palavras-chaves armazenadas na base de dados, estejam no seu formato protegido, preservando a segurança do sistema.

Recorreu-se igualmente à utilização de outra biblioteca, a qual tem como principal objetivo criar um *token* de sessão que reforça a segurança no processo de autenticação e validação dos utilizadores no sistema.

**FP2 – Registrar:** Se o utilizador não tiver registado no sistema é necessário efetuar o registo da sua conta, selecionando a opção *register*, que redireciona o utilizador para uma página diferente onde se procede à recolha da informação necessária para criar uma conta nova.

Após inserir os dados necessários, é efetuada uma validação da palavra-passe, de forma a respeitar um conjunto de regras de segurança e proteção para o utilizador.

**FP4 – Definir Objetivos Pessoais:** O utilizador pode definir uma lista de objetivos e manter o acompanhamento dos objetivos concluídos no decorrer do uso da aplicação.

**FA1 – Escrever Diário / FA2 – Autoavaliar Sentimentos:** O utilizador pode optar por autoavaliar diariamente os seus sentimentos relacionados com o humor, vida social, sono e hábitos saudáveis. É também possível escrever pequenas anotações em forma de diário na aplicação.

**FP5 – Responder Questionários:** Após o registo no sistema estar concluído, é apresentado ao utilizador um questionário que tem de responder sobre Escalas de Ansiedade Depressão e Stress (EADS-21) composta por 21 itens.

Para além do questionário inicial apresentado após registo, o utilizador pode também optar por realizar outros questionários de diferentes áreas relacionadas com a qualidade de vida dentro da aplicação.

**FS6 – Avaliar Questionários:** Através de uma função na aplicação é possível obter os resultados das subcategorias do questionário utilizado. Já o resultado global, quando obtido, é armazenado na base de dados para futuras comparações e análises, sendo apresentado ao utilizador antes de entrar na página principal da aplicação.

### 3.3 Sistema de Recomendação

Nesta subsecção é abordado o mecanismo implementado no sistema de recomendação, assim como o papel desempenhado pelas atividades no sistema.

**FS2 – Recomendar Atividades** Estas atividades distribuem-se por três categorias distintas, no âmbito de competências a nível desportivo, social e de relaxamento, figura 3. É imperativo que a soma das três categorias apresentadas corresponda a um valor total de 100 valores. Se uma atividade obtiver mais valorização numa determinada categoria, apenas significa que esta é mais predominante nessa mesma categoria. A valorização das categorias e atividades, foram preenchidas respeitando um critério pessoal, o objetivo será comprovar o seu caso de uso no presente trabalho apresentado e não priorizar a atribuição numérica da classificação das atividades, conforme visível na figura 2.

id_exercise	name	sport_category	social_category	relaxing_category	description	duration
1	Bowling	50	30	20	Jogar Bowling	180
2	Futebol	80	20	0	Jogar uma partida de Futebol	90
3	Pescar	50	15	35	Ir à Pesca	180
4	Xadrez	70	15	15	Jogar uma partida de Xadrez	30
5	Golfe	50	25	25	Dar umas tacadas no campo de golfe	60
6	Piquenique	0	75	25	Fazer um piquenique	90
7	Praia	10	60	30	Ir à Praia	210
8	Refeição Fora	0	80	20	Fazer uma Refeição Fora	80
9	Jogos de Tabuleiros	0	90	10	Participar em jogos de tabuleiros	90
10	Leitura	0	0	100	Ler um livro	20
11	Dançar	30	20	50	Dançar ao som da tua música	30
12	Caminhar Ar Livre	25	25	50	Dar um passeio na Natureza	15
13	Música	0	0	100	Ouvir Música	45
14	Natação	80	0	20	Ir à Piscina	60

**Figura 2.** Atividades (Base de Dados)

Utilizar-se-á um modelo tridimensional de coordenadas, no qual cada atividade se encontra representada como um ponto relativo no espaço, e, em substituição das coordenadas espaciais conhecidas (x,y,z), apresentam-se os valores correspondentes das categorias (desporto, social, relaxamento).



**Figura 3.** ChooseCategoryScreen (Aplicação)

Utilizando como ponto inicial o valor do ponto do utilizador inserido, torna-se possível calcular desse modo para cada ponto de atividade a sua distância no espaço a esse mesmo ponto de referência.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

Com esta fórmula, é possível determinar os pontos mais próximos da coordenada do utilizador, o que se traduz nas atividades que estão mais relacionadas com os valores inseridos pelo utilizador.

**FE4 - Avaliar Exercício** O utilizador é capaz de selecionar qual das atividade do sistema foi realizada, conforme a sua preferência, e avaliá-la conforme dois parâmetros:

1. Escala de 0 a 10 valores, que avalia o impacto da diminuição da ansiedade;
2. Escala de 0 a 5 valores, que avalia a satisfação geral da atividade realizada;

As referidas avaliações têm um papel fulcral na adaptação do ponto de referência de categoria do utilizador no espaço tridimensional, conforme é possível ver na figura 4. Se uma avaliação de uma atividade selecionada for elevada, esta obedece a uma aproximação da coordenada referência do utilizador para os valores representados pela atividade escolhida.

Caso contrário, o impacto da alteração de valores no seu ponto de referência não será significativo e, ao fim de umas interações seguindo o mesmo padrão de avaliação, por parte do utilizador, o sistema solicitará que este volte a preencher os valores das categorias, de forma a recomendar melhores atividades para aquele tipo de utilizador. A isto chama-se, algoritmo de atualização.

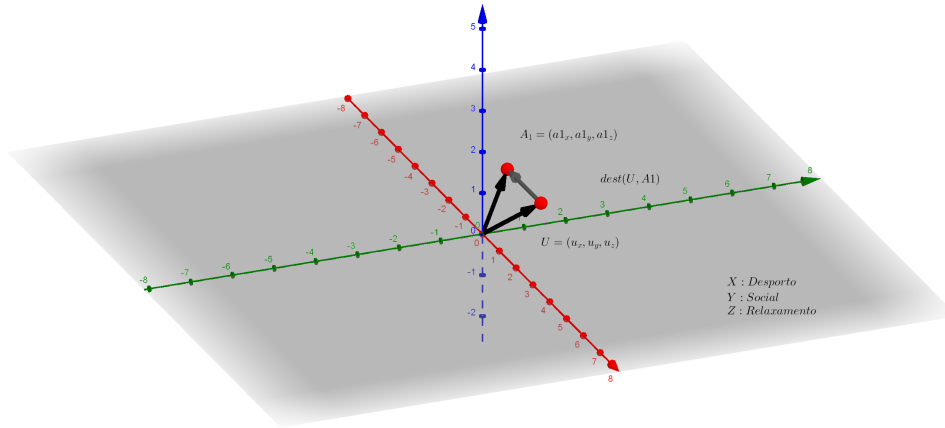


Figura 4. Gráfico Tridimensional da solução desenvolvida.

**Algoritmo de atualização** Resposta do utilizador para avaliar resultado da atividade selecionada:

- R1 : Diminuição da ansiedade;
- R2 : Satisfação geral da atividade;

O algoritmo de atualização do perfil do utilizador após submissão da avaliação, segue a seguinte fórmula:

$$rating = \frac{R1 + R2}{R1max + R2max} * \alpha \quad (2)$$

$$\alpha = \frac{6}{10} \quad (3)$$

O valor do ritmo de atualização,  $\alpha$ , foi determinado através de várias interações e análise dos resultados obtidos a partir do algoritmo em execução. Este conceito constitui o principal responsável pelo crescimento da atualização dos valores.

A taxa de atualização,  $rating$ , é posteriormente utilizada para calcular os novos valores da categoria do utilizador, respeitando a seguinte fórmula:

- $u'$  : nova coordenada de perfil das categorias do utilizador;
- $u$  : coordenada atual de perfil das categorias do utilizador;
- $a1$  : coordenada das categorias da atividade selecionada;

$$u' = u + (a1 - u) * rating \quad (4)$$

## 4 Conclusões

A presente artigo teve como principal objetivo o desenvolvimento de uma aplicação móvel, com as suas respetivas componentes, capaz de auxiliar os utilizadores com perturbação de ansiedade a compreenderem a sua condição e desenvolverem ferramentas para lidarem com os seus sentimentos, pensamentos e emoções, a fim de reduzir o impacto da perturbação da ansiedade no seu quotidiano.

Este tipo de aplicações já foram analisadas em diversas circunstâncias e comprovadas com o seu valor de melhoria e diminuição da perturbação da ansiedade.

De forma avaliar a solução obtida esta será disponibilizada a um grupo de utilizadores, utilização de uma técnica de amostragem, que terão de responder a um questionário no final de um tempo experimental definido, sendo posteriormente analisados os benefícios do uso da aplicação e os fatores da usabilidade da aplicação. Será também realizada uma série de entrevistas, para aprovar os resultados adquiridos e obtenção do *feedback* dos utilizadores da aplicação. No final, será elaborado um relatório dos resultados e respetivas conclusões.

## 5 Agradecimentos

Este trabalho conta com o apoio da FCT (Fundação para a Ciência e a Tecnologia) nomeadamente através do financiamento da Unidade de Investigação UI/DB/00760/2020 e com o apoio do Fundo Europeu de Desenvolvimento Regional (FEDER) através do Programa Operacional Regional do Norte – NORTE 2020, entidade financiadora do projeto SmartHealth (Inteligência Artificial para Cuidados de Saúde Personalizados ao Longo da Vida), ref<sup>ª</sup> NORTE-01-0145-FEDER-000045.

## Referências

1. Bandelow, B., Michaelis, S.: Epidemiology of anxiety disorders in the 21st century (2015)
2. DGS.: Programa Nacional para a Saúde Mental (2017)
3. Ferreira, L., Pereira, L., da Fé Brás, M., Iichuk, K.: Quality of life under the COVID-19 quarantine (2021)
4. ANACOM.: O Sector das Comunicações (2018)
5. American Psychiatric Association.: Diagnostic and statistical manual of mental disorders : DSM-5 (2013)
6. Carmen P., McLean, Anu A., Brett T., Litz B., Stefan H.: Gender differences in anxiety disorders: Prevalence, course of illness, comorbidity and burden of illness (2011)
7. Senaratne, R., Van Ameringen M., Mancini, C., Patterson B.: The burden of anxiety disorders on the family (2010)
8. Teigen, K.: Yerkes-Dodson: A Law for all Seasons (1994)
9. Correia, D., Brites, J.: Guia Prático para Vencer a Ansiedade (2020)

# *Software as a Service: Modelo de Distribuição e Comercialização de Software Aplicado a Ingest*

Alexandra Branco<sup>1</sup> e Nuno Bettencourt<sup>2</sup>

<sup>1</sup> Instituto Superior de Engenharia do Porto, Porto, Portugal  
1182161@isep.ipp.pt

<sup>2</sup> Instituto Superior de Engenharia do Porto, Porto, Portugal  
nmb@isep.ipp.pt

**Resumo** Nos últimos anos, os avanços da computação na *cloud* e das tecnologias de contentorização vieram facilitar a distribuição de *software*. Assim surge o conceito de *Software as a Service* (SaaS), que se traduz na disponibilização do *software* através da internet. Porém, a adoção desta prática permanece complexa, uma vez que envolve alterações no quotidiano das organizações e no processo de desenvolvimento e entrega do *software*. Com este estudo, pretende-se demonstrar a aplicabilidade deste conceito através do provisionamento de uma infraestrutura capaz de suportar a implantação de um *software* de *ingest* de nível empresarial. Foi estudado o estado atual da arte e das tecnologias que permitem concretizar os objetivos propostos. A solução foi desenhada com base no estudo realizado e posteriormente implementada. Para avaliar a solução e perceber o impacto que esta trouxe realizaram-se testes funcionais à solução recolhendo métricas sobre os tempos de duração do processo de provisionamento e implantação de *software*. Em suma, os objetivos deste estudo foram alcançados permitindo a distribuição do *software* como um serviço.

**Palavras-chave:** Contentores · *Infrastructure as Code* · *Ingest* · Kubernetes · *Software as a Service*.

## 1 Introdução

Nos últimos anos assistimos a uma transformação digital em vários negócios e setores de atividade, como é o caso da saúde e da educação. No entanto, a transformação digital não é específica de uma indústria<sup>3</sup>, esta pode ser aplicada em qualquer organização que procure integrar tecnologia nas várias áreas do negócio<sup>4</sup>. Nos ambientes de produção e pós-produção audiovisual esta transformação foi visível ao longo dos anos, e atualmente a televisão e o cinema já funcionam

<sup>3</sup> *Disrupting Traditional Broadcasting Through Digital Transformation and Social/Digital Production*, [https://www.grassvalley.com/media/0cona5rd/gv\\_alyve\\_wp-pub-2-0881a-en.pdf](https://www.grassvalley.com/media/0cona5rd/gv_alyve_wp-pub-2-0881a-en.pdf) (accedido em 02/12/2022)

<sup>4</sup> *What is digital transformation?*, <https://enterprisersproject.com/what-is-digital-transformation> (accedido em 02/12/2022)

com conteúdos totalmente digitais [1]. O surgimento de dispositivos tecnológicos cada vez mais capazes contribuíram para esta transformação, e nos dias de hoje, computadores pessoais, *smartphones* ou *tablets* funcionam também como dispositivos de exibição e captura [2].

O *ingest* faz parte do cotidiano das empresas que trabalham com grandes volumes de *media* e é o termo técnico utilizado na indústria do *broadcast* para definir o processo de preparação dos conteúdos multimídia para edição [2]. Em particular, os sistemas de *ingest* auxiliam os operadores nas suas tarefas diárias através do processamento automático de fluxos, que podem incluir a captura de vários sinais de vídeo, a extração de metadados, transcodificação, gestão e distribuição do conteúdo.

A evolução tecnológica da última década, juntamente com as melhorias da velocidade de conexão à internet, a capacidade de processamento de dados na *cloud*, o surgimento de arquiteturas de micro serviços e os utilizadores cada vez mais exigentes, permitiram evoluir a forma como o *software* é distribuído, surgindo o conceito de *SaaS* [3]. Embora seja uma forma de distribuição do *software* que está a ser amplamente adotada e em crescente desenvolvimento na indústria, existem organizações e áreas de negócio em que existe alguma dependência de *hardware* e por isso torna-se difícil uma total disponibilização do *software* como um serviço.

Este estudo tem como principal objetivo melhorar a distribuição de um *software* de *ingest* de nível empresarial. Assim, procedeu-se ao provisionamento de uma infraestrutura, através de técnicas de *Infrastructure as Code* (IaC), capaz de suportar a implantação do *software*.

## 2 Estado da Arte

No âmbito do estudo realizado foram estudados os conceitos teóricos subjacentes de forma a perceber o estado atual da arte e das tecnologias envolvidas que permitem cumprir com os objetivos definidos.

### 2.1 *Software as a Service*

O avanço tecnológico dos últimos anos permitiram evoluir a forma como o *software* é distribuído, surgindo o conceito de *SaaS*. Ao invés de o *software* ser comprado e instalado nos computadores dos utilizadores, agora encontra-se disponível na internet [4].

A disponibilização do *software* como um serviço pode ser visto como uma prioridade, em particular nas áreas de *software* de *Customer Relationship Management* (CRM) e *Enterprise Resource Planning* (ERP) [5].

Apesar das novas aplicações e sistemas serem projetados desde o início para disponibilização como um serviço, existem organizações que procuram transformar e adaptar as arquiteturas dos seus sistemas atuais. A transformação arquitetural de um sistema é algo trabalhoso e demorado.

## 2.2 *Infrastructure as Code*

A gestão e implementação de servidores com alta credibilidade e disponibilidade sempre foi um procedimento longo e desafiador, especialmente no passado em que era um processo maioritariamente manual [6].

A introdução da computação na *cloud* juntamente com a evolução das tecnologias de virtualização surgem como uma solução promissora para muitos destes problemas. A combinação destas tecnologias oferece uma forma mais eficiente de gestão de configurações, que resolve os problemas de escalabilidade e agilidade da infraestrutura. No entanto, muitas organizações enfrentam ainda problemas com a inconsistência destes sistemas, pois tendem a usar processos e ferramentas legadas que não conseguem acompanhar a evolução exigida [7].

A ideia por trás do conceito de IaC é escrever e executar código para definir, implantar, atualizar e destruir a infraestrutura. Isto representa uma mudança importante na mentalidade, agora tudo passa a ser tratado como *software*, mesmo os aspetos que representam *hardware* [8]. A possibilidade de escrever código para definir a infraestrutura é um mecanismo poderoso. No entanto, implica que os programadores aprendam novas linguagens de programação e ferramentas. Apesar deste investimento inicial, a conversão das práticas manuais de criação da infraestrutura em código permite obter melhorias drásticas na capacidade de entregar *software*.

## 2.3 Orquestração de Contentores

Nas aplicações contentorizadas é possível gerir e manter os contentores de forma manual, através das interfaces gráficas e APIs disponibilizadas pelos fabricantes. Porém, num sistema com um número elevado de contentores e com necessidades específicas não é viável efetuar esse trabalho manualmente<sup>5</sup>. A orquestração de contentores permite definir como implantar, monitorizar e controlar dinamicamente as aplicações. Normalmente, as tecnologias de orquestração de contentores oferecem as seguintes funcionalidades [9]:

**Controlo do limite de recursos (*resource limit control*)** permite reservar uma quantidade específica de recursos para um determinado contentor.

**Agendamento (*scheduling*)** define a política usada para colocar a quantidade desejada de contentores nos nós desejados num determinado instante de tempo.

**Balanceadores de carga (*load balancing*)** é responsável por distribuir a carga entre as várias instâncias.

**Verificação de integridade (*health check*)** a integridade pode ser obtida controlando e verificando se um contentor é capaz de responder a solicitações.

**Tolerância a falhas (*fault tolerance*)** permite especificar e manter um número desejado de contentores.

**Escalonamento automático (*auto-scaling*)** permite adicionar e remover contentores automaticamente.

---

<sup>5</sup> *What is container orchestration?*, <https://www.vmware.com/topics/glossary/content/container-orchestration.html> (acedido em 02/12/2022)

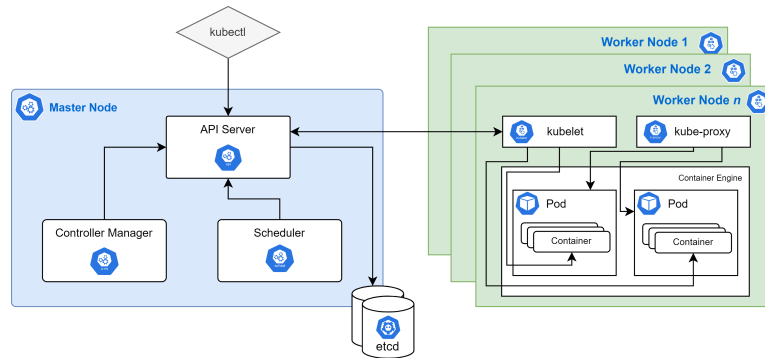


Fig. 1. Arquitetura de um *cluster* K8s<sup>6</sup>

Atualmente existem várias tecnologias que permitem realizar esta orquestração de contentores, nomeadamente Kubernetes (K8s), Docker Swarm, OpenShift, Nomad e Marathon. Dadas as características das várias tecnologias e com a aplicação de um método de apoio à decisão, o *Analytic Hierarchy Process* (AHP), escolheu-se a tecnologia Kubernetes como a mais adequada para resolução do problema.

O K8s, foi originalmente desenvolvido pela Google, depois de uma década de experiência na implantação de sistemas escaláveis e confiáveis. Desde a sua introdução em 2014, o K8s cresceu e tornou-se um dos maiores e mais populares projetos *open-source* do mundo.

É uma plataforma de orquestração de contentores que automatiza muitos dos processos manuais envolvidos na implantação, gestão e dimensionamento de aplicações em contentores. Atualmente é considerada a API padrão para a construção de aplicações na *cloud*. Está presente em quase todas as *clouds* públicas e neste momento está sobre a alçada da *Cloud Native Computing Foundation* (CNCF).

O Kubernetes garante que todos os contentores estão a ser executados, ficando atento a contentores inativos, sem resposta ou falhados, substituindo-os por novos contentores em correto funcionamento. Esta tecnologia foi criada para mudar radicalmente a forma como as aplicações são implantadas e foi desenhada para oferecer mais velocidade, eficiência e agilidade.

Na arquitetura de um K8s, um *cluster* é um conjunto de *hosts* de armazenamento e recursos de rede que o K8s usa para executar as várias tarefas. Os principais componentes (Figura 1) de um *cluster*, são:

**Etcd** é uma base de dados distribuída que armazena pares de chave-valor e fornece uma forma de armazenar informação consistente num *cluster*;

**API Server** é responsável por expor uma interface REST através da qual todas as interações relacionadas com os objetos do K8s são validadas e efetuadas;

**Scheduler** é responsável por selecionar o *node* que preenche os requisitos de um *pod* a ser lançado;

<sup>6</sup> *Kubernetes Components*, <https://kubernetes.io/docs/concepts/overview/components/> (acedido em 02/12/2022)

- Kubelet** é o elemento primário de um *node* que recebe instruções do *API Server* e a sua principal tarefa é instruir como operar os contentores dos *Pods*. Ademais, este elemento tem a tarefa de provisionar os *volumes* necessários ao *Pod* e de reportar o estado do seu *node* e dos seus *Pods* ao resto do sistema;
- Kube Proxy** responsável por criar a abstração fornecida pelos *services*. Para isso este elemento cria e gere um conjunto de regras de rede permitindo o redirecionamento e balanceamento dos pedidos pelos vários *Pods*;
- Kubectl** é a interface de linha de comandos que executa comandos no *cluster* K8s;
- Controller Manager** tem a tarefa de controlar e para isso conta com um conjunto de controladores individuais que realizam tarefas em rotina, sendo os mais relevantes: *Node Controller*, *Replication Controller* e *Service Account*.

### 2.4 Gestão de Aplicações

As implantações com K8s são realizadas através de ficheiros YAML, e normalmente, uma aplicação requer mais do que um objeto e isso traduz-se em vários ficheiros com centenas de linhas. À semelhança das ferramentas de gestão de configuração, que auxiliam na gestão e configuração de sistemas computacionais, existem também ferramentas de gestão de aplicações, cujo objetivo é facilitar a implantação e gestão das aplicações.

A ferramenta Helm é apresentada, desde o seu primeiro *commit* no repositório Git em 2015, como um gestor de pacotes do K8s. O seu principal objetivo é empacotar os objetos K8s necessários para implantação de uma aplicação, garantindo que podem ser instalados, atualizados e destruídos juntos.

Helm fornece opções para configurar pacotes no momento de instalação e reconfigurar instalações durante as atualizações. Apesar disso, Helm é um gestor de pacotes e não uma ferramenta de gestão de configurações.

## 3 Proposta de Solução

A solução implementada traduziu-se, na prática, à implantação de um *cluster* K8s provisionado de forma automática através de uma *pipeline* de *Continuous Integration/Continuous Delivery* (CI/CD) com recurso ao GitLab<sup>7</sup>. Para a criação do *cluster* foi utilizado o Terraform<sup>8</sup> que permitiu aplicar técnicas de IaC e tirar partido de todas as vantagens que esta tecnologia oferece.

Embora exista uma arquitetura base normalmente adotada para a implantação de *clusters* K8s, cada caso de uso tem as suas especificações que obrigam a escolher uma topologia para o *cluster*. No caso deste estudo, foi implantado um *cluster* com três *master nodes* e implementados mecanismos de redundância e alta disponibilidade, para garantir que a falha de um componente não implica a indisponibilidade do *software*.

---

<sup>7</sup> *CI/CD Pipelines*, <https://docs.gitlab.com/ee/ci/pipelines/> (acedido em 02/12/2022)

<sup>8</sup> *Terraform by Hashicorp*, <https://www.terraform.io/> (acedido em 02/12/2022)

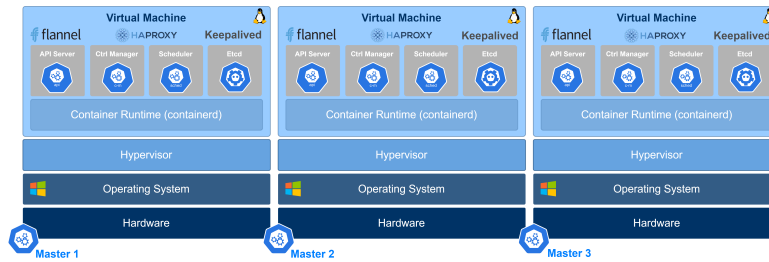


Fig. 2. Representação de alto nível dos três *masters* que compõe o *cluster*

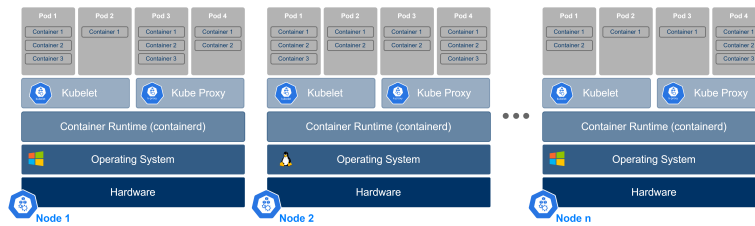


Fig. 3. Representação de alto nível dos *nodes* que compõe o *cluster*

Foram utilizadas as tecnologias *HAProxy* (para balancear carga), *Keepalived* (para verificar a cada período de tempo a disponibilidade de cada *master node*) e *Flannel* (para implementação do modelo de rede sugerido pelo K8s).

Cada *master* é composto por vários componentes e cada um deles com responsabilidades bem definidas. A figura 2 ilustra a arquitetura dos três *masters* da solução. Cada servidor Windows hospeda uma VM Linux onde são instanciados os componentes do K8s (*i.e.*, *API Server*, *Controller Manager*, *Scheduler* e *Etc*), o *Flannel*, o *HAProxy* e o *Keepalived*.

Cada *node* será instanciado diretamente num servidor físico. Dado o requisito de o *cluster* a implementar ser híbrido, isto é, possibilitar a execução de conteúdos Windows e Linux, os sistemas operativos dos *nodes* podem ser distintos entre eles. Cada *node* possui o *Kubelet* e o *Kube Proxy* instalado que permitem gerir e manter os vários *pods* em execução no *cluster*.

A *pipeline* CI/CD implementada é responsável pela criação e destruição de todos os componentes referidos. É composta por quatro *stages*: *validate*, *plan*, *apply* e *destroy*, e existem dependências entres os vários *jobs*, isto é, só faz sentido provisionar os *nodes* do *cluster* caso os *masters* já tenham sido provisionados.

## 4 Avaliação

De seguida apresentam-se os resultados da avaliação obtidos de acordo com os objetivos formulados.

A criação da infraestrutura demora em média ( $\bar{x}$ ) cerca de 12 minutos e 47 segundos. Nos três servidores utilizados os valores foram semelhantes, no entanto o desvio padrão ( $\sigma$ ) calculado é de 53 segundos, justificado pelas características dos servidores (*i.e.*, velocidade de processamento, memória RAM, armazenamento)

**Tabela 1.** Tempo de criação de toda a infraestrutura

<i>Stage</i>	<i>Job</i>	Servidor 1	Servidor 2	Servidor 3
<i>Validate</i>	validate	00:00:18	00:00:14	00:00:14
<i>Plan</i>	lin-nodes	00:00:09	00:00:09	00:00:07
	masters	00:00:10	00:00:10	00:00:08
	win-nodes	00:00:10	00:00:11	00:00:09
<i>Apply</i>	lin-nodes	00:01:44	00:03:38	00:01:28
	masters	00:07:29	00:07:11	00:07:02
	win-nodes	00:02:42	00:02:36	00:02:37
<b>Total</b>		00:12:42	00:13:55	00:11:45
$\bar{x}$		<b>00:12:47</b>		
$\sigma$		00:00:53		

**Tabela 2.** Tempo de destruição da infraestrutura

<i>Stage</i>	<i>Job</i>	Servidor 1	Servidor 2	Servidor 3	$\bar{x}$	$\sigma$
<i>Destroy</i>	all	00:02:26	00:02:27	00:02:30	<b>00:02:28</b>	00:00:02
	lin-nodes	00:01:13	00:01:10	00:01:15	<b>00:01:13</b>	00:00:02
	win-nodes	00:00:23	00:00:19	00:00:21	<b>00:00:21</b>	00:00:02

e pela velocidade mais lenta de conexão à internet. No geral, o valor obtido (12 minutos e 47 segundos) parece bastante promissor, dada a complexidade da infraestrutura criada e que provisionada de forma manual demoraria bastante mais tempo uma vez que seria necessário conectar a cada um dos servidores e executar todos os comandos de provisionamento em cada um deles. Além disso, devido ao fator humano poderiam ocorrer erros e os resultados não seriam previsíveis.

A destruição da infraestrutura pode ser realizada de uma única vez, através da execução do *job* `all`, ou parcialmente através da execução dos *jobs* `lin-nodes` e/ou `win-nodes`. A tabela 2 apresenta os valores obtidos na execução dos vários *jobs* que compõe o *stage* `destroy`. Pela análise da tabela 2 verifica-se que dos três testes realizados em três servidores diferentes, a destruição completa da infraestrutura dura em média ( $\bar{x}$ ) 2 minutos e 28 segundos (`destroy:all`). Nos três servidores utilizados os valores são semelhantes, obtendo-se um desvio padrão ( $\sigma$ ) de 2 segundos.

Analisando a tabela 3 verifica-se que os tempos de implantação do *software* através da solução implementada se revelam bastante interessantes e promissores. Uma implantação do *software* dura, em média, 22 segundos. Para os três servidores utilizados os tempos revelaram-se bastante semelhantes, tendo-se obtido um desvio padrão irrelevante.

É de salientar que estes valores refletem o tempo de execução da *pipeline* de CI/CD, no entanto existe um espaço temporal necessário para que o K8s consiga iniciar os contentores. Este tempo pode ser maior ou menor dependendo das imagens do contentor utilizadas, isto é, se no *node* onde executar o *pod* já possuir a imagem do contentor este processo será significativamente mais rápido uma vez que o *download* já não precisa de ser realizado.

**Tabela 3.** Tempo de destruição da infraestrutura

	servidor 1	Servidor 2	Servidor 3
<b>Teste 1</b>	00:00:21	00:00:23	00:00:20
<b>Teste 2</b>	00:00:23	00:00:25	00:00:22
<b>Teste 3</b>	00:00:20	00:00:21	00:00:19
$\bar{x}_r$	00:00:21	00:00:23	00:00:20
$\sigma_r$	00:00:01	00:00:02	00:00:11
$\bar{x}$	<b>00:00:22</b>		
$\sigma$	00:00:00		

## 5 Conclusão

O projeto desenvolvido obteve resultados promissores, permitindo evoluir a forma de distribuição do *software*. Com a solução implementada é possível aceder ao *software* de forma instantânea sem a necessidade de instalações, reduzir os custos com *hardware* e disponibilizar atualizações constantes e oportunas. O estudo e desenvolvimento conseguido permitiu alavancar e servir como ponto de partida para disponibilização do *software* numa *cloud* pública.

## Referências

1. Tekalp, A. Murat: Digital Video Processing. 2nd edn. Prentice-Hall, New York (2015).
2. Taveira, B., Virtualização de Estúdios Móveis na Produção de Conteúdos Audiovisuais em Direto, Tese de mestrado da FEUP, Porto (2017).
3. Sun, W. and Zhang, X. and Guo, C. J and Sun, P. and Su, H.: Software as a Service: Configuration and Customization Perspectives. Em: 2008 IEEE Congress on Services Part II (services-2 2008). <https://doi.org/10.1109/SERVICES-2.2008.29>.
4. Wang, L. and von Laszewski, G. and Younge, A. and He, X. and Kunze, M. and Tao, J. and Fu, C.: Cloud Computing: a Perspective Study. New Generation Computing, 137-146 (2010).
5. Buxmann, P. and Hess, T. and Lehmann, S.: Software as a Service. Em: WIRTSCHAFTSINFORMATIK. (2008). <https://doi.org/10.1007/s11576-008-0095-0>.
6. Noll, J. L and Pretto, F.: Implementação de infraestrutura como código para provisionamento e deploy de aplicações. Em: Revista Destaques Acadêmicos. (2020). <https://doi.org/10.22410/issn.2176-3070.v12i4a2020.2760>.
7. Naziris, S.: Infrastructure as Code - Towards Dynamic and Programmable IT Systems, Master's Thesis of Faculty of Electrical Engineering, Mathematics and Computer Science of University of Twente, Netherlands (2019).
8. Brikman, Y.: Terraform: Up & Running: Writing Infrastructure as Code. 2nd ed. O'Reilly Media, Inc., United States of America (2019)
9. Casalicchio, E. and Iannucci, S.: The state of the art in container technologies: Application, orchestration and security. Em: Concurrency and Computation: Practice and Experience. (2020). <https://doi.org/https://doi.org/10.1002/cpe.5668>

---

## Test Results Framework

Miguel Cabral<sup>1</sup>, Pedro Silva<sup>2</sup>, Vítor Cardoso<sup>1</sup>, Pedro Reis<sup>1</sup>, Sandra Aires<sup>1,3</sup>,  
Fernando Carvalho<sup>1</sup>, and António Sousa<sup>1,3\*</sup>

<sup>1</sup> Instituto Superior de Engenharia do Porto, Porto, Portugal

<sup>2</sup> Capgemini Engineering, Porto, Portugal

<sup>3</sup> Laboratório de Engenharia Matemática, Porto, Portugal  
ats@isep.ipp.pt

**Resumo** Atualmente, os testes de automatização são prática comum em empresas que desenvolvem soluções baseadas em software. Estes testes têm a função de avaliar o desempenho das tecnologias utilizadas pelas empresas. Este documento apresenta um projecto desenvolvido numa empresa e com o qual se altera a forma de apresentação dos resultados de testes efetuados a software. Esse software destina-se ao sector automóvel, o qual representa uma das áreas de atividade mais importante na empresa.

Para realizar estes testes, a empresa utiliza uma aplicação desenvolvida internamente, *Engineer Test Automation Framework (ETAF)*. Embora seja possível ajudar os seus clientes através da utilização desta ferramenta, a empresa considerou que esta poderia tornar-se ainda mais vantajosa, nomeadamente na visualização de resultados, na medida em que o método utilizado pela empresa não é visto como intuitivo. Nesse sentido, surge este projecto, que pretende ser um complemento útil ao ETAF, otimizando a apresentação dos resultados dos testes realizados.

Este projecto centrou-se no desenvolvimento de uma aplicação Python, dividida em duas partes: uma que extrai a informação mais relevante dos testes realizados pela ETAF e outra responsável pela visualização desses resultados numa aplicação *web*. Este projecto pode desempenhar um papel útil na vida quotidiana da empresa, uma vez que permite a visualização intuitiva dos resultados obtidos no momento da execução dos testes de automatização, tanto pelos seus funcionários como pelos seus clientes.

**Palavras-chave:** Automatização de testes, XML, Extração de dados, *Dashboard* gráfica

## 1 Introdução

Nos dias que correm, a tecnologia, bem como a evolução e inovação da mesma, são bases imprescindíveis do mundo profissional, tendo as relativas à área da engenharia um dos mais elevados impactos. Os avanços na engenharia estão

---

\* Corresponding author

inevitavelmente dependentes de um bom usufruto dos meios tecnológicos e vice-versa. Desta forma, é de elevada importância que haja um acompanhamento contínuo das mudanças tecnológicas, com o intuito de aumentar a sua eficácia.

A fim de avaliar as funcionalidades das várias tecnologias desenvolvidas, a empresa na qual se desenvolveu este projeto, realiza um vasto conjunto de testes que fornecem informações úteis sobre o objeto em estudo (desenvolvimento e teste de *software* ou *hardware*). As informações recolhidas incidem principalmente na verificação da existência de possíveis anomalias no *software* ou *hardware* estudado e na robustez do código implementado, isto é, se este é capaz de ser executado em diferentes condições, um elevado número vezes e num curto período de tempo, por exemplo. A informação gerada por cada teste realizado é bem organizada, mas é extremamente vasta, o que exige um esforço adicional ao operador na sua análise. É neste contexto que surge a necessidade da realização deste projeto, visando fornecer uma ferramenta que melhore a análise e tratamento dos dados provenientes dos testes que são realizados e simultaneamente apresente de forma adequada - tabular ou gráfica - um resumo dos objetos que o utilizador deseja estudar.

Neste documento, composto por 5 seções, detalha-se a aplicação desenvolvida para solucionar o problema apresentado. Na secção seguinte são introduzidas as ferramentas utilizadas para a sua resolução; na secção 3 apresenta-se o modelo desenvolvido, cujos resultados são detalhados na secção 4. Por último, na secção 5 apresentam-se as conclusões sobre a eficácia da metodologia adotada e esboçam-se propostas de melhoria.

## 2 Situação atual

Atualmente, com o intuito de auxiliar a automatização de testes dos seus clientes, a empresa utiliza uma aplicação baseada em *Robot Framework* [1] que desenvolveu internamente a partir da *framework* ETAF. Esta ferramenta gera de forma automática vários ficheiros com os resultados dos testes: um documento em formato *Extensible Markup Language* (XML) com informação detalhada dos testes, um ficheiro `log.html` e um `report.html`. Estes dois últimos constituem relatórios em formato *HyperText Markup Language* (HTML), onde são listadas todas as informações relativas aos testes executados. A análise de resultados exige a visualização de vários relatórios pouco intuitivos, não permitindo a comparação automática entre diversos relatórios gerados em tempos diferentes, e como tal, o estudo da evolução dos dados. Por outro lado, o arquivamento destes relatórios tem limitações porque não foi implementado um procedimento automático que guarde a informação relevante após a análise dos testes. Na Fig. 1 apresenta-se parte de um relatório gerado pela aplicação ETAF sobre um software em estudo e, através da análise do mesmo, é possível verificar a necessidade de se obter resultados mais intuitivos.

Pretende-se assim desenvolver uma ferramenta que extraia a informação relevante dos relatórios gerados pela ETAF, mas que também guarde a informação

de acordo com as necessidades do utilizador e permita uma análise temporal sobre a evolução das tecnologias desenvolvidas.

Com este projeto, a empresa tenta otimizar a *framework* ETAF, desenvolvida e utilizada internamente, focando-se em encontrar soluções para as suas limitações atuais. Nesse sentido, realizou-se uma análise às melhorias que seria possível implementar, tentando manter a atual metodologia de trabalho. Após essa análise, decidiu-se que a fase de geração dos ficheiros XML mantém-se, sendo adicionada uma nova funcionalidade para armazenar todas as informações de cada execução da ferramenta de teste num ficheiro formato XLS (do Excel) e numa base de dados MongoDB, de modo a permitir a sua posterior análise e visualização. Para esse efeito, desenvolveu-se uma *dashboard* gráfica onde são sumariados os resultados dos testes executados, bem como outros dados estatísticos que contribuam para uma boa análise a esses mesmos testes.

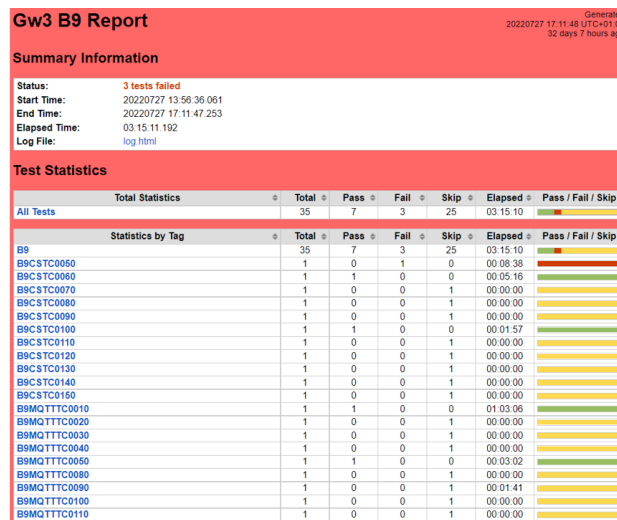


Fig. 1. Exemplo de um ficheiro *report.html* gerado pela ETAF.

## 2.1 Ferramentas utilizadas

*Robot Framework* é uma ferramenta que tem como finalidade a automatização de testes e, por isso, é utilizada por empresas da área da engenharia para esse fim. Algumas das suas principais vantagens são o facto de se tratar de uma tecnologia *open source* e de ter uma sintaxe relativamente simples [2]. Também a sua fácil adaptabilidade a inúmeras linguagens de programação, como Python ou Java, constitui um ponto positivo, já que esse fator torna a ferramenta ainda mais completa e abrangente. A informação gerada em cada teste pela aplicação ETAF é armazenada em ficheiros, respeitando a linguagem de marcação XML ou HTML. Ambas as linguagens delimitam a informação através de *tags*, no entanto um documento XML não tem as suas *tags* predefinidas, sendo estas escolhidas

pelo utilizador [3–6]. Os ficheiros gerados pela ETAF serão processados e arquivados pela aplicação a desenvolver em linguagem Python, sendo a informação extraída, guardada e apresentada numa *dashboard* gráfica.

Python é uma linguagem interpretada, de sintaxe fácil, orientada a objetos e repleta de bibliotecas para diversos fins que completam a implementação base [7]. Entre as suas possibilidades está o desenvolvimento, quer de aplicações, quer de interfaces que ligam diversos componentes de software. Torna-a, por isso, muito vantajosa para criar a interface entre as ferramentas a desenvolver e o utilizador [8]. Algumas bibliotecas Python utilizadas neste trabalho são Numpy, Pandas, ElementTree e Plotly. Recorreu-se ainda à *framework* Streamlit.

Em Python podem-se processar listas com o objetivo de suprir a falta de *arrays*, no entanto, a sua manipulação é um processo bastante lento. A biblioteca Numpy implementa funções para criar e operar *arrays* e matrizes de forma eficiente, colmatando essa lacuna na linguagem base. Numpy também é usada para implementar diversas operações matemáticas que operam sobre *arrays* ou sobre matrizes, sendo uma biblioteca a partir da qual se desenvolvem outras bibliotecas mais específicas, onde é necessário recorrer ao uso de operações matemáticas no tratamento e comparação de dados [9].

Pandas é uma dessas bibliotecas associadas ao Numpy e oferece funções para processar informação organizada em listas ou tabelas, sendo adequada para manipulação e análise estatística de dados, facilitando assim a sua interpretação [10].

ElementTree é uma biblioteca que fornece funções para a conexão do código com o documento XML [11].

Existem diversas bibliotecas para representação gráfica. Neste trabalho adotou-se a Plotly por possibilitar a representação gráfica de dados de forma interativa, através do uso de gráficos e tabelas de diferentes tipos [12].

Streamlit é uma *framework* do Python, *open source*, compatível com as bibliotecas Numpy e Pandas. As aplicações correm num navegador *web* de forma simples [13].

Neste projeto, a implementação da *dashboard* gráfica, recorreu extensivamente ao Streamlit e às bibliotecas referidas previamente. Para salvaguardar toda a informação, foi utilizada uma base de dados implementada sobre o MongoDB [14]. É uma base de dados do tipo NoSQL e destinada à gestão e armazenamento de documentos processados pela aplicação. Este recurso constitui uma mais-valia para a evolução e perspetiva futura de todo o trabalho.

### 3 Sistema implementado

Com este projeto procura-se otimizar a *framework* ETAF, focando-se em arranjar soluções para as suas limitações atuais no processamento de dados relativos aos testes executados. Assim, a fase de geração dos ficheiros XML, através do uso da *framework* ETAF mantém-se, sendo adicionadas novas funcionalidades para armazenar num ficheiro XLS e numa base de dados todas as informações de cada execução, sendo assim possível a sua posterior visualização e tratamento.

### 3.1 Arquitetura do sistema

A arquitetura do sistema apresenta-se no fluxograma da Fig. 2. O núcleo desta nova ferramenta é a aplicação que gere as tarefas a executar: assim, de início é solicitado ao utilizador que informações pretende extrair do ficheiro XML. Após o processamento, os resultados obtidos são armazenados num ficheiro XLS e na base de dados NoSQL. Após o processamento é gerada a *dashboard* gráfica com um resumo tabular e/ou gráfico dos resultados. Por último, o utilizador pode extrair mais informação dos testes executados ou encerrar a aplicação.

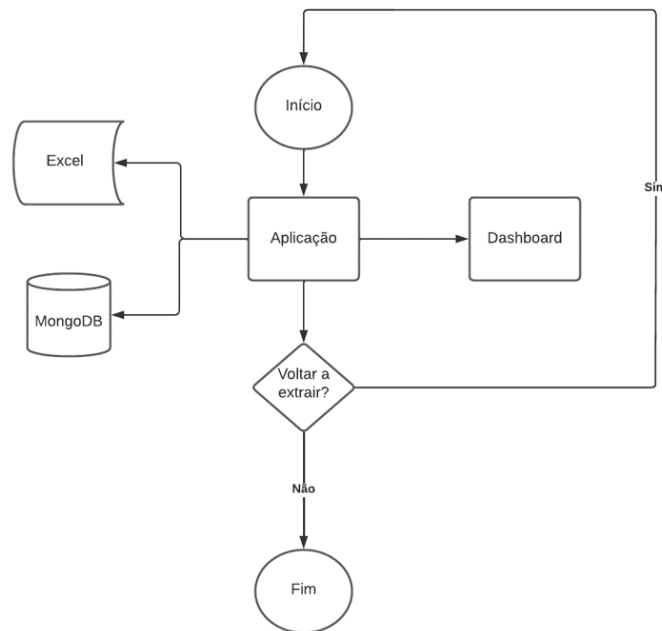


Fig. 2. Esquema geral do sistema implementado.

## 4 Resultados

Neste capítulo apresentam-se os resultados que foram obtidos com esta nova ferramenta quando aplicada a um ficheiro XML gerado pela aplicação ETAF num dos seus testes. Os resultados são armazenados num ficheiro XLS, e numa base de dados NoSQL, implementada em MongoDB, respetivamente.

### 4.1 Extração e armazenamento de dados

Os dois primeiros objetivos delineados inicialmente foram a leitura do ficheiro XML que contém as informações relativas aos testes executados e a extração e armazenamento dos dados mais relevantes presentes nesse ficheiro.

## Livro de Atas do SEI'22

Na Fig. 3a é possível observar o menu inicial criado para a aplicação, sendo solicitado ao utilizador que introduza informação relativa aos dados que pretende extrair. No exemplo ilustrado, é visível que dá entrada a opção "2", que corresponde à extração das informações *TimeStamp* e *Status* dos testes executados.

É possível observar na Fig. 3c a listagem de alguns resultados que foram extraídos do ficheiro XML, o que significa que estes mesmos dados já foram armazenados num ficheiro XLS, criado de forma automática, bem como numa base de dados, também esta criada automaticamente. As figuras 3b e 3d ilustram um extrato da informação enviada para o ficheiro XLS e para a base de dados MongoDB, respetivamente.

```
----- What data do you want to extract from the tests?-----
Enter 0 to extract the Time Stamp of the tests
Enter 1 to extract the Status of the tests
Enter 2 to extract All of the above
-----
Enter the number assigned to the action you desire: 2
```

(a) Menu da aplicação.

	A	B	C	D	E
1	NAME	STATUS	START_TIME	END_TIME	
2	TC_0050 Boot CL15 on Registration	FAIL	20220727 13:56:36.388	20220727 14:05:14.256	
3	TC_0060 Boot CL15 on Reboot	PASS	20220727 14:05:14.271	20220727 14:10:30.131	
4	TC_0070 Boot CL15 on NewServiceList	SKIP	20220727 14:10:30.136	20220727 14:10:30.140	
5	TC_0080 Boot CL15 on GetRealTime	SKIP	20220727 14:10:30.150	20220727 14:10:30.159	
6	Core Service TC_0090 Ignition With GPS Registration	SKIP	20220727 14:10:30.201	20220727 14:10:30.211	
7	Core Service TC_0100 Ignition With GPS Reignite	PASS	20220727 14:10:30.220	20220727 14:12:26.907	
8	Core Service TC_0110 Ignition With GPS New Service List	SKIP	20220727 14:12:26.911	20220727 14:12:26.921	
9	Core Service TC_0120 Ignition With GPS Get Real Time	SKIP	20220727 14:12:26.930	20220727 14:12:26.934	
10	Core Service TC_0130 Ignition Without GPS Registration	SKIP	20220727 14:12:26.943	20220727 14:12:26.952	

(b) Resultados guardados em ficheiro XLS.

```
-----
NEW TEST
NAME: TC_0050 Boot CL15 on Registration
STATUS: FAIL
START_TIME: 20220727 13:56:36.388
END_TIME: 20220727 14:05:14.256

NEW TEST
NAME: TC_0060 Boot CL15 on Reboot
STATUS: PASS
START_TIME: 20220727 14:05:14.271
END_TIME: 20220727 14:10:30.131

NEW TEST
NAME: TC_0070 Boot CL15 on NewServiceList
STATUS: SKIP
START_TIME: 20220727 14:10:30.136
END_TIME: 20220727 14:10:30.140

NEW TEST
NAME: TC_0080 Boot CL15 on GetRealTime
STATUS: SKIP
START_TIME: 20220727 14:10:30.150
END_TIME: 20220727 14:10:30.159
```

(c) Listagem de resultados.

```
first.tests
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
ADD DATA VIEW
_id: ObjectId('6312470b24284cb4136cb3e1')
name: "TC_0050 Boot CL15 on Registration"
status: "FAIL"
start time: "20220727 13:56:36.388"
end time: "20220727 14:05:14.256"

_id: ObjectId('6312470b24284cb4136cb3d1')
name: "TC_0070 Boot CL15 on NewServiceList"
status: "SKIP"
start time: "20220727 14:10:30.136"
end time: "20220727 14:10:30.140"

_id: ObjectId('6312470b24284cb4136cb3e1')
name: "TC_0060 Boot CL15 on Reboot"
status: "PASS"
start time: "20220727 14:05:14.271"
end time: "20220727 14:10:30.131"
```

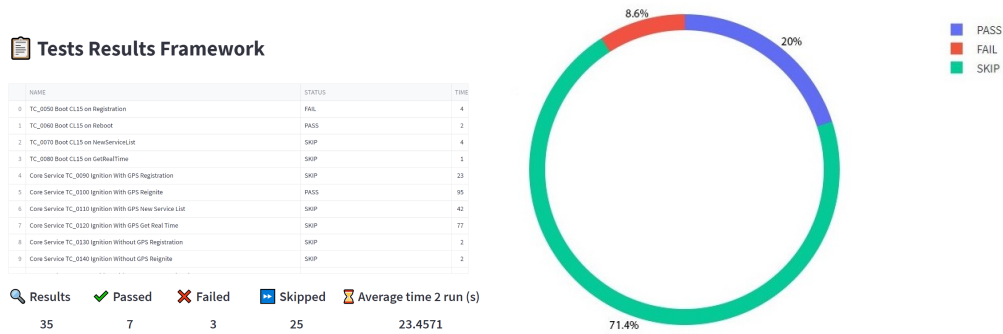
(d) Resultados arquivados em MongoDB.

**Fig. 3.** Etapas do processamento. (a) menu da aplicação, (b) resultados em ficheiro XLS, (c) visualização das etapas do processamento (d) resultados armazenados na base de dados.

## 4.2 Dashboard gráfica

Os restantes objetivos definidos para este projeto estão relacionados com o desenvolvimento da *dashboard* gráfica, havendo inicialmente a necessidade de tratamento dos dados extraídos e, por fim, o desenvolvimento de uma aplicação *web* que permita a visualização intuitiva dos resultados, aplicação esta desenvolvida com recurso à *framework* Streamlit do Python.

Na Fig. 4a observa-se o separador da página *web* com a *dashboard*, sendo visível o título do projeto, bem como uma tabela com os resultados extraídos dos testes executados e várias colunas com informações importantes relacionadas com esses mesmos testes. Os resultados presentes na tabela são condicionados por diversos filtros aplicados na pesquisa. Os filtros definidos na aplicação estão localizados numa zona lateral da página *web*. Com a aplicação do filtros consegue-se que a exposição de resultados seja restrita a apenas certos testes, como testes que falharam ou que avançaram, sempre de acordo com as necessidades do utilizador. Por fim, na Fig. 4b apresentam-se os resultados segundo um gráfico circular com a percentagem de testes extraídos que passaram, falharam ou foram avançados.



(a) *Dashboard* com resultados tabelados. (b) *Dashboard* com resultados gráficos.

**Fig. 4.** Resultados apresentados na *dashboard* da aplicação.

## 5 Conclusões e trabalho futuro

Com este trabalho foi desenvolvida uma aplicação capaz de ler e analisar o conteúdo de ficheiros XML, gerados pela *framework* ETAF da *Robot Framework*, com informações relativas aos testes de software realizados, bem como à extração e armazenamento da informação mais relevante. Assim, os dados extraídos e armazenados foram devidamente tratados, havendo já a possibilidade de comparação com outros resultados obtidos e provenientes de vários testes realizados pela empresa.

Por fim, o último ponto de desenvolvimento do projeto, que diz respeito à elaboração de uma página *web* que possibilitasse uma visualização intuitiva dos dados tratados, foi também solucionado, no entanto será alvo de melhoria futura.

Futuramente, uma medida a implementar será melhorar a forma como é apresentada a informação na página *web*, ou seja, a *dashboard* gráfica. Atualmente, tal é feito com recurso ao Streamlit mas deverá evoluir para o Grafana [15], um recurso já utilizado em diversos projetos da mesma, sendo por isso do interesse

da empresa a implementação desta ferramenta no projeto. Outra melhoria será a implementação de uma função na aplicação que seja capaz de comparar a informação que se pretende introduzir na base de dados com a informação de outros testes já armazenada nessa mesma base de dados, de forma a evitar informação repetida, o que, caso aconteça, pode condicionar de forma negativa a avaliação do progresso dos testes a longo prazo. Finalizando, uma outra possível melhoria do projeto seria a capacidade de avaliar a evolução ao longo do tempo da informação introduzida na base de dados. Isto é, verificar se a percentagem de sucessos no conjunto de testes têm a tendência a aumentar ou a diminuir, ou seja, estudar a evolução temporal.

Assim, mesmo não representando uma aplicação ótima e sem margem de melhoria, o que foi desenvolvido até ao momento constitui já uma forte base de trabalho que poderá num futuro próximo culminar numa ferramenta que representará uma mais-valia para a empresa.

## Referências

1. Robot framework. <https://robotframework.org/>. (Last accessed on 26 Aug 2022).
2. Github - robotframework/robotframework: Generic automation framework for acceptance testing and RPA. <https://github.com/robotframework/robotframework>. (Last accessed on 26 Aug 2022)
3. XML: Extensible Markup Language, MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/XML>. (Last accessed on 26 Aug 2022)
4. XML tutorial. <https://www.w3schools.com/xml/default.asp>. (Last accessed on 26 Aug 2022)
5. Markup language definition. [https://techterms.com/definition/markup\\_language](https://techterms.com/definition/markup_language). (Last accessed on 26 Aug 2022)
6. What is an XML file and how do I open one? <https://www.indeed.com/career-advice/career-development/xml-file>. (Last accessed on 26 Aug 2022)
7. What is python? <https://www.python.org/doc/essays/blurb/>. (Last accessed on 26 Aug 2022)
8. What is Python used for? A beginner's guide. <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>. (Last accessed on 26 Aug 2022)
9. Introduction to numpy. [https://www.w3schools.com/python/numpy/numpy\\_intro.asp](https://www.w3schools.com/python/numpy/numpy_intro.asp). (Last accessed on 26 Aug 2022)
10. Pandas - Python data analysis library. <https://pandas.pydata.org/>. (Last accessed on 25 Aug 2022)
11. Processing XML in Python — ElementTree, by Deepesh Nair | Towards Data Science. <https://towardsdatascience.com/processing-xml-in-python-elementtree-c8992941efd2>. (Last accessed on 26 Aug 2022)
12. Plotly python graphing library. <https://plotly.com/python/>. (Last accessed on 25 Aug 2022)
13. Streamlit documentation. <https://docs.streamlit.io/>. (Last accessed on 25 Aug 2022)
14. What is NoSQL? NoSQL databases explained. <https://www.mongodb.com/nosql-explained>. (Last accessed on 26 Aug 2022)
15. What is Grafana? <https://grafana.com/oss/grafana/> (Last accessed on 22 Oct 2022)

## Índice de Autores

Aires, Sandra, 28, 36, 68, 188

Almeida, Marta, 28

Almeida, Ricardo, 76, 140

Alves, Joaquim, 148

Barbosa, João Romano, 116

Bettencourt, Nuno, 92, 180

Botelho, André, 124

Branco, Alexandra, 180

Cabral, Miguel, 188

Canedo, Teresa, 36

Cardoso, Vítor, 28, 36, 68, 188

Carvalho, Fernando, 28, 36, 68, 188

Carvalho, Patrícia, 148

Caçote, Catarina, 76

Coelho, Luis, 84, 101

Coelho, Ricardo, 156

Correia, Afonso, 108

Costa, Joel, 164

Duarte, F. Jorge, 60, 164

Duarte, F. Jorge F., 44

Escudeiro, Nuno, 76

Faria, Luiz, 172

Fernandes, António, 60

Fernandes, Luís, 28

Ferraz, Teresa, 101

Ferreira, Ricardo, 84, 101

Fontes, Rita, 76

Gomes, Simão, 140

## **Livro de Atas do SEI'22**

---

Gouveia, Alexandre, 108  
Guedes, Pedro, 68  
Guimarães, Duarte, 148  
  
Lima, Jorge, 19  
  
Maia, Bruno, 36  
Malta, Marco, 19  
Martins, Constantino, 164, 172  
Monteiro, André, 148  
Mota, André, 76  
  
Nicola, Susana, 92  
  
Oliveira, Daniela, 101  
  
Pereira, João Paulo, 116  
Pimpita, Marly Juliana, 132  
Pinto, José, 140  
Portugal, Joana, 19  
  
Reis, Pedro, 28, 36, 68, 188  
Reis, Sara, 101  
Rodrigues, Fátima, 124  
  
Saldanha, Laura, 76  
Santos, João, 19, 172  
Silva, Emanuel, 52, 156  
Silva, Pedro, 188  
Soares, Diogo, 76  
Soares, Nuno, 52  
Soares, Rui, 44  
Sousa, António, 28, 36, 68, 188  
Sousa, Tomas, 101  
  
Terroso, Teresa, 19  
  
Vilão, Gina, 148  
  
Zickler, E. Luciano, 92

