



Plataforma para gerar e executar diálogos de negócio no contexto de recolha automática de dados

JOEL MÁRCIO LEITÃO DA SILVA

novembro de 2017

Plataforma para gerar e executar diálogos de negócio no contexto de recolha automática de dados

Joel Márcio Leitão da Silva

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas de Informação e Conhecimento**

Orientador: Professor Doutor Paulo Maio

Porto, Outubro de 2017

Resumo

O presente documento descreve um estudo dos sistemas de informação atuais das seguintes áreas: 1) Controlo e Gestão de Tempos de Presença, 2) Controlo de Acessos e 3) Controlo e Gestão dos Tempos de Produção e Qualidade.

Os sistemas atuais destas áreas são tipicamente compostos pelos seguintes dois componentes:

- **Unidade de Aquisição de Dados (UAD):** dispositivo móvel que possibilita a recolha dos dados pretendidos;
- **Sistema de Gestão de Informação:** estes sistemas permitem processar a informação recolhida pelas UADs.

Existem alguns problemas nos sistemas atuais, nomeadamente a existência de uma forte dependência entre a UAD e o sistema de gestão em questão. Se o objetivo da recolha dos dados mudar ou se a UAD inicialmente configurada for alterada por outro equipamento, é necessário realizar um esforço de implementação ou reimplementação para que seja possível recolher os dados pretendidos. Esta implementação além de demorar tempo vai ter custos associados, uma vez que, só profissionais com conhecimentos tecnológicos no sistema em questão é que conseguem realizar as alterações necessárias.

O objetivo deste trabalho é desenvolver uma solução que incentive a reutilização de diálogos de negócio em diferentes UADs. Para ser possível a reutilização, esta solução apresenta o conceito de divisão em partes distintas, permitindo a separação de responsabilidades. Esta divisão consiste nos seguintes três processos:

- **Processo de modelação:** através de uma plataforma de modelação é possível modelar diálogos de negócio com base numa notação;
- **Processo de aquisição:** o processo de aquisição é o momento em que numa determinada UAD é executada uma recolha dos dados com base num diálogo de negócio;
- **Processo de gestão da informação:** este processo trata-se dos sistemas de gestão atuais.

Com base na proposta de solução é apresentada uma proposta de *design* e o respetivo desenvolvimento dos protótipos com base nesta proposta de *design*, um para o processo de modelação e outro para o processo de aquisição. Por fim é realizada uma avaliação à solução apresentada.

Palavras-chave: Diálogo de Negócio, Plataforma de Modelação, Interpretador, UAD, BPMN, BPMN.IO

Abstract

This document describes a study of current information systems in the following areas: 1) Attendance and Punctuality Management, 2) Access Control and 3) Control and Management of Production Times and Quality.

The current systems of these areas are typically composed by the following two components:

- **Data Acquisition Unit (UAD):** a mobile device that enables the collection of the desired data;
- **Information Management System:** these systems allow to process the information collected by the UADs.

There are some troubles in the current systems, namely the existence of a strong dependency between the UAD and the information management system. If the purpose of the data collection is changed or if UAD what is it initially configured is replaced for another device, an implementation or reimplementation effort is required before the desired data can be collected. This implementation besides taking time will have associated costs, because only professionals with technological knowledge about the system can do the necessary changes.

The goal of this work is to develop a solution that encourages the reuse of business dialogues in different UADs. To be possible to be reused, this solution presents the concept of division into distinct parts, allowing the separation of responsibilities.

- **Modeling process:** through a modeling platform it is possible to model business dialogues based on a notation;
- **Acquisition process:** the acquisition process is the moment in one UAD is executed one collection data based on business dialogue;
- **Information management process:** this process is the same of the current information management systems.

Based on the proposed solution is presented a design proposal and the respective development of the prototypes based on this design proposal, one for the modeling process and another for the procurement process.

Keywords: Business Dialogue, Modeling Platform, Interpreter, UAD, BPMN, BPMN.IO

Agradecimentos

Quero começar por agradecer ao meu orientador, ao Professor Doutor Paulo Maio, por todo o apoio, estímulo e disponibilidade demonstrada ao longo deste trabalho, bem como todo o interesse que sempre lhe dedicou.

Ao Departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto, e aos seus docentes, pelos conhecimentos transmitidos, quer ao longo da licenciatura como do mestrado, que sem eles este trabalho não era possível.

Aquela que para o Mundo é a Vera, mas que para mim é todo o meu Mundo, dedico em especial este trabalho, por todo o seu apoio, pela sua grandeza e pelo seu amor.

A todos vocês, um muito OBRIGADO!

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema.....	2
1.3	Objetivos.....	3
1.4	Análise de Valor	4
1.5	Estrutura	5
2	Contexto	7
2.1	Áreas de Negócio	7
2.1.1	Controlo e Gestão de Tempos de Presença	7
2.1.2	Controlo de Acessos.....	10
2.1.3	Controlo e Gestão dos Tempos de Produção e Qualidade.....	11
2.1.4	Sumário	13
2.2	Soluções Existentes	13
2.2.1	Sistemas de Controlo e Gestão de Tempos de Presença	13
2.2.2	Sistemas de Controlo de Acessos	19
2.2.3	Sistemas de Controlo e Gestão dos Tempos de Produção e Qualidade	21
2.2.4	Sumário	23
2.3	Caracterização das UADs.....	23
2.3.1	Características das UADs.....	24
2.3.2	Sumário	27
3	Especificação de Diálogos (e Processos) de Negócio.....	29
3.1	Caracterização dos Diálogos de Negócio	29
3.2	Fluxo de Atividades de Aquisição.....	30
3.2.1	Dados de entrada (<i>input</i>)	31
3.2.2	Dados de saída (<i>output</i>).....	31
3.2.3	Decisões.....	31
3.2.4	Validações	32
3.2.5	Armazenamento dos dados recolhidos	32
3.2.6	Sumário	33
3.3	Modelação de Processos de Negócio	33
3.3.1	Avaliação de ferramentas BPMN	37
3.4	Análise de Valor	40
3.4.1	Problema.....	41
3.4.2	Processo de Negócio e de Inovação	42
3.4.3	Proposta de Valor	45
3.4.4	Modelo de CANVAS	47
3.5	Visão da Solução	49

4	Análise e Requisitos	53
4.1	Processo de Modelação	53
4.1.1	Análise.....	54
4.1.2	Requisitos.....	58
4.2	Processo de Aquisição	64
4.2.1	Análise.....	64
4.2.2	Requisitos.....	66
5	Design	69
5.1	Visão Geral	69
5.2	Notação de Especificação de Diálogos.....	71
5.3	Plataforma de Modelação	74
5.3.1	Ferramenta BPMN.IO.....	74
5.3.2	Extensão da Ferramenta BPMN.IO.....	80
5.4	Interpretador.....	83
6	Implementação	87
6.1	Notação de Especificação de Diálogos.....	88
6.2	Plataforma de Modelação	89
6.3	Interpretador.....	97
6.4	Sistema de Gestão de Informação	102
7	Avaliação	103
7.1	Diálogos de Ensaio.....	103
7.2	Modelação de Diálogos	107
7.3	Execução de Diálogos.....	110
7.3.1	Testes de Validação aos Dados.....	110
7.3.2	Testes de Armazenamento de Dados	111
7.3.3	Teste de Execução do Fluxo de Aquisição	111
7.4	Resultados Atingidos	113
8	Conclusão	115
8.1	Resumo	115
8.2	Objetivos Alcançados	117
8.3	Trabalho Futuro.....	117
8.4	Considerações Finais	117
	Referências	119
	Anexos	123
A.1	Notação BPMN	123

8.4.1	Objetos de Fluxo	124
8.4.2	Dados	134
8.4.3	Objetos de ligação	134
8.4.4	Swinlanes	135
8.4.5	Artefactos	136

Lista de Figuras

Figura 1 – Exemplo de uma folha de um livro de ponto tradicional (adaptado)	8
Figura 2 - Cenário do registo de assiduidade	10
Figura 3 – Cenário possível no controlo de acessos.....	11
Figura 4 – Cenário de aquisição de dados na montagem de um computador	13
Figura 5 - Funcionamento Padrão dos Sistemas de Controlo e Gestão de Tempos de Presença	15
Figura 6 - Backoffice do InnuxTime (Innux Technologies, Lda, 2016b).....	17
Figura 7 – Arquitetura do BioStar 2 SDK (Suprema, 2016b)	18
Figura 8 – BioStar 2 SDK Workflow (Suprema, 2016b)	18
Figura 9 - Sistema Backoffice BioStar 2 (Suprema, 2016a)	19
Figura 10 – Configuração do Sistema BioStar 2 (Suprema, 2016a).....	20
Figura 11 – Configuração do Sistema InnuxAccess (Innux Technologies, Lda, 2016d).....	20
Figura 12 - Funcionamento Padrão dos Sistemas de Controlo de Acessos	21
Figura 13 - UAD do sistema Task Manager, modelo terminal da empresa X64 (X64, 2016a) ...	21
Figura 14 – UAD do sistema ZEUS® Plant Data Collection solution, modelo IT 8001(ISGUS, 2016c).....	22
Figura 15 – UADs com teclas de função (KABA, 2016a).....	25
Figura 16 – Conceitos de um diálogo de negócio	30
Figura 17 – Objetos que constituem um negócio (Eriksson and Penker, 2000)	34
Figura 18 - Interface Gráfico do BPMN.IO	40
Figura 19 - Modelo <i>New Concept Development</i> NCD (adaptado (Koen et al., 2001))	44
Figura 20 – Visão da Solução Pretendida	49
Figura 21 - Modelo de Dómino: Regras do Diálogo de Negócio	57
Figura 22 – Diagrama de Casos de Uso da Plataforma de Modelação	59
Figura 23 – Instanciação de um diálogo de negócio	66
Figura 24 – Diagrama de Casos de Uso do “Processo de Aquisição”	66
Figura 25 - Diagrama de Componentes Geral do Sistema	70
Figura 26 – Diagrama de Classes do BPMN (adaptado).....	71
Figura 27 - Diagrama de Classes “Extensão do BPMN”	73
Figura 28 - Diagrama de Componentes do BPMN.IO.....	74
Figura 29 – Arquitetura do bpmn-js: Componentes e responsabilidades (adaptado (“BPMN.IO,” 2016)).....	75
Figura 30 – Diagrama de Componente e Serviços do <i>diagram-js</i>	77
Figura 31 – Bibliotecas do Componente <i>bpmn-moddle</i>	78
Figura 32 – Alterações no Componente do <i>bpmn-js-properties-panel</i>	80
Figura 33 – Extensão do bpmn-js-properties-panel.....	82
Figura 34 – Arquitetura do Interpretador	85
Figura 35 – Exemplo da Extensão Realizada ao BPMN	89
Figura 36 – Tarefas de Aquisição Representadas em JSON	91
Figura 37 – Representação Gráfica das Tarefas de Aquisição no BPMN.IO.....	92

Figura 38 – Extensão Realizada do Componente <i>Draw</i>	92
Figura 39 – Extensão do Serviço <i>Pallette</i>	93
Figura 40 – Implementação da Extensão do <i>ContextPad</i>	94
Figura 41 – Extensão do Classe <i>PropertiesActivator</i>	95
Figura 42 – Configuração dos Componentes Desenvolvidos no BPMN.IO	96
Figura 43 – Interface Gráfico da Plataforma de Modelação	97
Figura 44 – Lógica de Instanciação do Interpretador	98
Figura 45 – XSLT: transformação da regra “Introdução” para HTML.....	100
Figura 46 – Interface Gráfico do Interpretador	101
Figura 47 - Interpretador utilizando o método de aquisição código de barras	101
Figura 48 – XML do Diálogo de Ensaio Nº3	109
Figura 49 – Resultado das Validações na UAD	111
Figura 50 – Diálogo de Ensaio Nº 7	112
Figura 51 – Dados Armazenados em Formato JSON	113

Lista de Tabelas

Tabela 1 – Métodos de transferência de dados entre as UADs e as aplicações	23
Tabela 2 – UADs analisadas.....	26
Tabela 3 – Métodos de aquisição suportados pelas UADs	27
Tabela 4 – Método/Protocolo de transmissão utilizado nas UADs.....	27
Tabela 5 – Questões possíveis nos diálogos de negócio	30
Tabela 6 – Notações para a modelação de processos (ABPMP, 2013)	35
Tabela 7 – Alguns elementos das notações para a modelação de diálogos de negócio.....	36
Tabela 8 – Critérios de avaliação de ferramentas de modelação BPMN	38
Tabela 9 – Critérios de Seleção da Ferramenta BPMN	39
Tabela 10 - Modelo de Negócio de CANVAS	48
Tabela 11 – Caso de uso “Criar Diálogo de Negócio”	59
Tabela 12 – Caso de Uso “Alterar Diálogo de Negócio”	59
Tabela 13 – Caso de Uso “Inserir Tarefa de Aquisição”	60
Tabela 14 – Caso de Uso “Questão ao utilizador”	60
Tabela 15 – Caso de Uso “Inserir Questão ao Sistema”	61
Tabela 16 – Caso de Uso “Definir Resposta ao Estática (pré-definida) ”	61
Tabela 17 – Caso de Uso “Configurar validação de sobre os dados recolhidos”	62
Tabela 18 – Caso de Uso “Configurar processo decisão”	62
Tabela 19 – Caso de Uso “Configurar armazenamento dos dados recolhidos”	63
Tabela 20 – Caso de Uso “Executar diálogo de negócio”	67
Tabela 21 – Correlação entre o BPMN a Notação de Especificação de Diálogos	72
Tabela 22 – Método da Sistema de Gestão.....	102
Tabela 23 – Diálogo de Ensaio Nº 1.....	104
Tabela 24 – Diálogo de Ensaio Nº 2.....	104
Tabela 25 – Diálogo de Ensaio Nº 3.....	104
Tabela 26 – Diálogo de Ensaio Nº 4.....	104
Tabela 27 – Diálogo de Ensaio Nº 5.....	105
Tabela 28 – Diálogo de Ensaio Nº 6.....	105
Tabela 29 – Diálogo de Ensaio Nº 7.....	105
Tabela 30 – Diálogo de Ensaio Nº 8.....	106
Tabela 31 – Diálogo de Ensaio Nº 9.....	106
Tabela 32 – Diálogo de Ensaio Nº 10.....	107
Tabela 33 – Complexidade do Diálogo Vs. Tempo de Modelação	108
Tabela 34 – BPMN Objetos de Fluxo – Tipos de Eventos.....	124
Tabela 35 – Tipos de Eventos Iniciais de Alto Nível.....	125
Tabela 36 – Tipos de Eventos de Subprocessos	126
Tabela 37 - Tipos de Eventos Intermediário.....	128
Tabela 38 – Tipos de Eventos Finais.....	130
Tabela 39 – Tipos de Atividades no BPMN.....	131
Tabela 40 – Tipos de Tarefas	131

Tabela 41 – Tipos de Subprocessos.....	132
Tabela 42 – Tipos de <i>Gateway</i>	133
Tabela 43 – Elementos Dados BPMN.....	134
Tabela 44 – Objetos de Ligação BPMN	135
Tabela 45 – Swimlanes BPMN.....	135
Tabela 46 – Artefactos BPMN	136

Acrónimos e Símbolos

Lista de Acrónimos

ABPMP	<i>Association of Business Process Management Professionals</i>
API	<i>Application Programming Interface</i>
BPM	<i>Business Process Management</i>
BPMN	<i>Business Process Model Notation</i>
BPR	<i>Business Process Reengineering</i>
DLL	<i>Dynamic-link library</i>
ERP	<i>Enterprise Resource Planning</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile</i>
HTML	<i>HyperText Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
REST	<i>Representational State Transfer</i>
RFID	<i>Radio-Frequency IDentification</i>
SDK	<i>Software Development Kit</i>
TQM	<i>Total Quality Management</i>
UAD	<i>Unidade de Aquisição de Dados</i>
XML	<i>eXtensible Markup Language</i>
XSD	<i>XML Schema Definition</i>
XSLT	<i>eXtensible Stylesheet Language for Transformation</i>

1 Introdução

1.1 Contexto

A informação constitui uma fonte de grande importância para a gestão das organizações. Segundo Drucker (Drucker, 1993), quem dispõe de informação de boa qualidade, fidedigna, em quantidade adequada e no momento certo, adquire vantagens competitivas. Por outro lado, a falta de informação de qualidade possibilita brechas que podem conduzir a erros e/ou à perda de oportunidades.

Com a emergência da internet, James e Marakas (James and Marakas, 2011) afirmam que ela é a plataforma global de comunicação para as organizações e que os seus gestores foram obrigados a mudar a forma como raciocinam, para conseguirem alterar e adaptar os seus processos de negócio a esta nova realidade. Estas mudanças surgem com a necessidade de os processos de gestão terem que ser executados mais rapidamente, para não se perderem oportunidades. Para se atingir este objetivo é necessário obter num momento exato e com precisão o estado do negócio. Ainda (James and Marakas, 2011) asseguram, que as tecnologias de informação são um apoio essencial para atingir estes objetivos. Esta relevância criou a necessidade de que a informação tenha que ser adquirida através de métodos mais simples, rápidos e eficazes.

Atualmente existem diversos sistemas de informação desenvolvidos para corresponder às necessidades processuais e organizacionais das organizações. Neste trabalho, o objetivo é abordar os sistemas das seguintes áreas de negócio: 1) Controlo e Gestão de Tempos de Presença, 2) Controlo de Acessos a Áreas Restritas e 3) Controlo e Gestão de Tempos de Produção e Qualidade.

Os sistemas existentes nas áreas apresentadas são normalmente compostos pelos seguintes dois módulos:

1. **Recolha automática de dados:** a recolha automática dos dados pretendidos é efetuada através de Unidades de Aquisição de Dados (UAD).
2. **Sistemas de gestão:** são responsáveis pela validação e processamento dos dados recolhidos pelas UADs.

Para uma mais fácil compreensão destes sistemas, como exemplo, pode-se observar os sistemas de controlo de acesso a uma sala de espetáculos. Nas portas de entradas destas salas, existe um controlo através de uma leitura ao código de barras existente nos bilhetes. O equipamento que efetua esta leitura é uma UAD e o seu objetivo é validar ou não a entrada. No momento da validação, a UAD comunica com o sistema de gestão enviando um conjunto de informação (e.g. uad_id, data/hora, código barras) para determinar se deve ou não permitir a entrada. Após receber o pedido, o sistema de gestão vai efetuar uma validação para verificar se os dados recolhidos pela UAD são ou não válidos, caso sejam válidos, informa a UAD que pode permitir a entrada e regista o acesso deste bilhete. Este registo é importante para um processamento posterior que numa próxima validação, estes sistemas de gestão não permitam o acesso a um bilhete com a mesma informação.

O exemplo anterior apresenta uma validação remota (*online*), no entanto, existe o tipo de validação local (*offline*) que é realizada na UAD.

As empresas que comercializam estes sistemas (Elo Sistemas de Informação, 2014) denominam o processo de aquisição/recolha de dados nas UADs por diálogo de negócio. Onde, as UADs permitem que seja possível executar um ou mais diálogos de negócio, com o objetivo de recolher a informação necessária para o processo em questão. Um diálogo de negócio representa um conjunto de questões, cujas respostas uma vez recolhidas representam um conjunto de informação estruturada e sistematizada que pode ser disponibilizada para um processamento posterior.

1.2 Problema

Nas áreas de negócio abordadas, um dos problemas existentes nas soluções relaciona-se com as necessidades da captura dos dados ser específica em cada cliente ou projeto. Se o objetivo do processo de aquisição de dados mudar, obrigatoriamente terá de existir um novo processo de implementação ou reimplementação para alterar os diálogos de negócio e a forma como estes são executados nas UADs e/ou o processamento efetuado pelos sistemas de gestão sobre os dados adquiridos.

Estes sistemas obrigam a um esforço quer de recursos tecnológicos como humanos, sempre que existe a necessidade de se criar ou alterar um processo de aquisição de dados. A utilização destes recursos aumentam os custos e tempo na adaptação dos sistemas atuais. Onde, os recursos humanos têm que ser profissionais com conhecimentos tecnológicos e conceptuais no sistema em questão. A quantidade de tempo despendido na adaptação, não permite que se obtenha de imediato os dados pretendidos no processo de aquisição.

É importante realçar que não é possível adaptar um processo de aquisição existente, sem existir a necessidade de se adaptar todo o sistema, uma vez que, existe uma forte ligação/dependência

entre as UADs utilizadas e os sistemas de gestão. Normalmente, são sistemas de adaptação por implementação ou reimplementação imperativa.

Pode-se enunciar os problemas dos sistemas atuais da seguinte forma:

- Correspondem as necessidades de um projeto ou cliente;
- Alta dependência entre as UADs e os sistemas de gestão;
- Obrigam a efetuar alterações quer nas UADs como nos sistemas de gestão sempre que o objetivo da recolha de dados mudar;
- Apenas profissionais qualificados e com conhecimento nos sistemas em questão conseguem criar ou alterar diálogos de negócio;
- Custos de implementação/adaptação elevados.

Para alterar os diálogos de negócio e a forma como estes são executados nas UADs atualmente, existe a possibilidade de termos dispositivos móveis (e.g. telemóveis, tablets) que são autênticos computadores. Estes equipamentos juntamente com os novos sistemas operativos, como o iOS (Apple, 2016), Android (Google Inc., 2016) e o Windows Phone (Microsoft Corporation, 2016), permitem que seja possível desenvolver aplicações para serem executadas nestes equipamentos.

Alguns dos dispositivos móveis possuem características semelhantes às UADs utilizadas nas áreas de negócio abordadas (e.g. leitura de cartões com base na tecnologia RFID, leitura de dados biométricos, leitura de códigos de barra). Estas características e a possibilidade de desenvolver aplicações para estes dispositivos, possibilitam que seja possível desenvolver aplicações para as áreas de negócio apresentadas e diminuir os custos, recursos, tempo de implementação, produção e adaptação das UAD existentes.

A utilização destes dispositivos móveis como equipamentos de aquisição de dados permitem inúmeras vantagens, como por exemplo, o processamento de informação, o processo de validação de dados e o protocolo de comunicação tornarem-se mais simples de implementar e adaptar as necessidades de cada área de negócio. O custo destes equipamentos é inferior à das UADs tradicionais.

1.3 Objetivos

Este trabalho tem como objetivo principal agilizar e facilitar, nomeadamente na configuração e execução, o processo de recolha de dados em diferentes UADs nas seguintes áreas de negócio: 1) Controlo e Gestão de Tempos de Presença, 2) Controlo de Acessos e 3) Controlo e Gestão de Tempos de Produção e Qualidade.

Para cumprir este objetivo é necessário efetuar várias tarefas que podem por si só serem consideradas como subobjetivos do objetivo principal. Consequentemente, os subobjetivos identificados são:

- Descrever e sistematizar as áreas de negócios;
- Descrever e sistematizar as soluções atuais e, em particular, as existentes no mercado português;
- Compreender, descrever, sistematizar e caracterizar os diferentes diálogos de negócio associados à recolha automática de dados nas áreas de negócios abordadas;
- Identificar e sistematizar as abordagens existentes que potencialmente são aplicáveis à resolução do problema abordado;
- Avaliar as abordagens identificadas com o intuito de as selecionar (ou não) para a solução a propor;
- Desenhar uma solução para o problema descrito explorando algumas das abordagens previamente identificadas;
- Implementar um protótipo da solução desenhada/proposta com vista à sua avaliação;
- Avaliar a solução proposta tendo em consideração cenários de aplicação.

A subsecção seguinte identifica resumidamente o valor de negócio resultante dos objetivos deste trabalho.

1.4 Análise de Valor

O núcleo deste trabalho é o desenvolvimento de uma plataforma que permita gerar diálogos de negócio. Um diálogo de negócio representa um modelo que define um conjunto de regras, que possibilita uma correta aquisição e transmissão dos dados pretendidos através das UADs.

A utilização de uma plataforma para gerar diálogos de negócio, facilita a definição de processos de aquisição mais rapidamente, simplificando o processo de modelação. Onde, pessoas sem quaisquer conhecimentos na área de engenharia de *software*, mas com conhecimentos da área de negócio, conseguem modelar processos de aquisição de dados. Esta simplicidade aliada à rapidez na modelação dos diálogos possibilita a redução de custos, tempo e recursos na implementação de um novo projeto/sistema ou, na adaptação de um existente. Esta plataforma visa desenvolver uma adaptabilidade/compatibilização, que atualmente não existe, entre os sistemas de gestão e as diversas UADs existentes no mercado.

É possível afirmar que o uso desta plataforma, possibilita mudar o conceito de implementação ou reimplementação imperativa utilizada nos sistemas atuais, para uma abordagem por configuração declarativa.

Pode-se concluir, que o uso da solução pretendida permite aumentar o valor para as empresas que desenvolvem e comercializam estes sistemas nas seguintes áreas:

- **Produção de sistemas:** através da metodologia de configuração estes sistemas serão implementados mais depressa e com um custo inferior. Outro fator importante é que

será possível, pessoas com apenas conhecimentos das áreas de negócio modelar/configurar os diálogos de negócio.

- **Comercialização dos sistemas:** no momento da comercialização/apresentação, este conceito permite adaptar/configurar rapidamente os sistemas às reais necessidades dos clientes. Desta forma, vai ser possível apresentar em menos tempo, com menos custos e até no momento da apresentação, o processo de aquisição desejado pelo cliente.

Por outro lado, os clientes no momento da compra, conseguem perceber que este sistema satisfaz as suas necessidades atuais e que permite de forma simples e rápida uma adaptação a novas/futuras necessidades. Um exemplo desta adaptação, num processo de aquisição de dados é a necessidade de alterar os dados de aquisição, método de aquisição ou método de comunicação/transmissão com aplicação de gestão.

1.5 Estrutura

A estrutura deste documento encontra-se dividida em oito capítulos.

O primeiro capítulo introduz a temática do trabalho, apresenta algumas definições de conceitos fundamentais, a motivação, os objetivos do trabalho, uma análise de valor, e a estrutura do documento.

O segundo capítulo apresenta uma contextualização, onde são descritas as áreas de negócio, os sistemas existentes e as UADs utilizadas.

O terceiro capítulo descreve a caracterização dos diálogos de negócio. Como os diálogos de negócio são processos de aquisição de dados, foi necessário fazer um enquadramento sobre as notações de modelação de processos de negócio e das suas ferramentas de modelação existentes. Ainda, é apresentado uma análise de valor onde, é descrito o problema e o raciocínio detalhado para chegar à visão da solução.

No quarto capítulo é apresentada a análise e requisitos do processo de modelação e do processo de aquisição.

O quinto capítulo descreve um *design* possível para a visão da solução.

No sexto capítulo é descrito as etapas realizadas na implementação dos componentes notação, plataforma e interpretador.

O sétimo capítulo apresenta a avaliação da solução proposta.

Finalmente no oitavo capítulo, é apresentados as conclusões da realização deste trabalho, objetivos alcançados, trabalho futuro e considerações finais.

2 Contexto

Este capítulo pretende efetuar uma apresentação das áreas de negócio e das respetivas soluções existente em cada área.

2.1 Áreas de Negócio

Esta seção apresenta uma descrição e sistematização das áreas de negócio abordadas neste trabalho:

- Controlo e Gestão de Tempos de Presença (cf. 2.1.1)
- Controlo de Acessos (cf. 2.1.2)
- Controlo e Gestão dos Tempos de Produção e Qualidade (cf. 2.1.3).

Cada uma destas áreas de negócio encontra-se descrita nas subsecções seguintes.

2.1.1 Controlo e Gestão de Tempos de Presença

As organizações têm a necessidade de obter os tempos de trabalho dos seus colaboradores, não só por uma questão de controlo interno da organização mas também porque a lei em muitos países assim o obriga. Esta necessidade e obrigação levou à criação dos denominados livros de ponto, com a finalidade de registar o tempo que um colaborador despende no seu posto de trabalho. Este registo é importante para diversos processos nas organizações, como por exemplo, o processamento do vencimento, para perceber se é necessário conceder dias de folga ao colaborador, controlar períodos de férias, entre outros.

O método tradicional de controlo e gestão de tempos de presença consiste no preenchimento manual dos livros de ponto (ver Figura 1), onde é necessário preencher o tempo que um

colaborador efetuou num determinado dia no seu posto de trabalho. O preenchimento manual deste livro pode muitas vezes ser feito de forma incorreta, uma vez que, de forma manual encontra-se sujeito aos erros cometidos pelas pessoas.

Posteriormente surgiram os relógios de ponto onde eram utilizados os “cartões de ponto”. Estes relógios possuem uma abertura onde é necessário colocar os cartões para imprimir a data e a hora, este procedimento tem que ser realizado pelos colaboradores na entrada e saída do seu serviço. Depois os cartões de ponto são recolhidos pelo departamento de recursos humanos para efetuarem o controlo das horas de trabalho de cada colaborador.

Livro de Ponto					
Funcionário: Joel Márcio Leitão da Silva				Mês: Maio	
DIA	Hora/Entrada	Hora/Saída	Hora/Entrada	Hora/Saída	Assinatura
1	Feriado				
2	8:58	12:33	14:00	18:03	<i>Joel Silva</i>
3	9:01	12:32	13:57	18:03	<i>Joel Silva</i>
4	8:58	12:33	13:57	18:02	<i>Joel Silva</i>
5	9:01	12:34	13:58	18:03	<i>Joel Silva</i>
6	Fim-de-semana				
7	Fim-de-semana				
8	8:58	12:33	13:57	18:09	<i>Joel Silva</i>
9	9:01	12:34	13:58	18:03	<i>Joel Silva</i>
10	9:00	12:32	13:57	18:02	<i>Joel Silva</i>
11	9:01	12:34	13:59	17:59	<i>Joel Silva</i>
12	8:58	12:33	13:58	18:03	<i>Joel Silva</i>
13	Fim-de-semana				
14	Fim-de-semana				
15	9:01	12:34	13:58	18:00	<i>Joel Silva</i>
16	8:58	12:33	13:59	18:02	<i>Joel Silva</i>
17	9:01	12:34	13:57	18:06	<i>Joel Silva</i>
18	8:58	12:32	13:58	18:04	<i>Joel Silva</i>
19	9:00	12:34	14:00	18:02	<i>Joel Silva</i>
20	Fim-de-semana				
21	Fim-de-semana				
22	8:58	12:34	13:57	18:02	<i>Joel Silva</i>
23	9:01	12:33	14:00	18:03	<i>Joel Silva</i>
24	8:58	12:34	13:58	18:04	<i>Joel Silva</i>
25	9:00	12:32	13:57	18:02	<i>Joel Silva</i>
26	9:01	12:34	13:58	18:05	<i>Joel Silva</i>
27	Fim-de-semana				
28	Fim-de-semana				
29	9:00	12:32	14:00	18:02	<i>Joel Silva</i>
30	9:01	12:34	13:57	18:00	<i>Joel Silva</i>
31	9:00	12:33	13:58	18:02	<i>Joel Silva</i>

Figura 1 – Exemplo de uma folha de um livro de ponto tradicional (adaptado)

Com a evolução das tecnologias de informação e de forma a simplificar a gestão dos processos nas organizações, as empresas de *software* desenvolveram soluções que permitem simplificar

o processo de controlo e gestão de tempos de presença. O objetivo destas soluções é recolher de forma automática os dados necessários, tais como:

- 1. Identificação do colaborador;**
- 2. Data e hora do registo;**
- 3. Tipo de registo:** este tipo de registo é de múltipla escolha, alguns dos tipos mais usados para este registo são: se o registo é de entrada ou de saída, se a ausência ou entrada é em serviço.

Após a recolha de informação, estes sistemas de forma automática ou semiautomática efetuam o processamento da informação recolhida, substituindo de forma eficaz os métodos manuais de recolha e processamento da informação.

As empresas IDONIC (IDONIC, 2017), SSE (SSE, 2017), Innux (Innux Technologies, Lda, 2016a) que desenvolvem estes sistemas, afirmam que a sua utilização possibilita obter as seguintes vantagens para as organizações:

- Aumento da produtividade;
- Controlo da assiduidade e pontualidade dos colaboradores;
- Redução de custos administrativos;
- Aumento da assiduidade e rentabilidade dos Recursos Humanos;
- Facilidade no processamento do vencimento;
- Gestão de faltas (e.g. doença, justificadas, não justificadas);
- Gestão de horas-extra;
- Controlo dos horários;
- Elimina a circulação de papel;
- Aumento da qualidade da informação e a organização dos Recursos Humanos.

A Figura 2 ilustra um cenário possível, representado em UML, do registo de assiduidade nestes sistemas. O objetivo deste diálogo é guardar a informação do tipo de registo (e.g. entrada, saída), identificação do colaborador e a data e hora do registo. Este diálogo de negócio é o mais simples nesta área pois, existem outros mais complexos, como a consulta de saldos de horas, justificações de faltas, ausências/saídas em serviço entre outros.

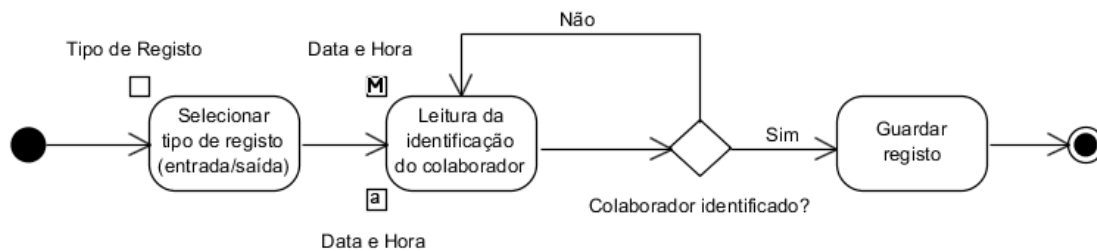


Figura 2 - Cenário do registo de assiduidade

2.1.2 Controlo de Acessos

Atualmente existem diversas situações nas organizações em que é necessário controlar quem e quando é que se esteve numa determinada localização. Tradicionalmente utiliza-se uma pessoa com a função de porteiro, para evitar a entradas de estranhos ou de pessoas não autorizadas a estas localizações e ainda, para registar quem e quando é que as pessoas autorizadas se deslocaram a estas locais.

Com a evolução das tecnologias de informação, apareceram sistemas de informação que permitem de forma automática efetuar o trabalho do porteiro com mais exatidão. Estes sistemas permitem que sejam criados perfis de acesso, que definem em que horários os colaboradores têm acesso a cada zona. Por exemplo, pode-se definir que determinados colaboradores têm acesso à porta de entrada de uma organização apenas entre as 8h às 19h, mas aos escritórios apenas poderão entrar das 9h às 18h.

Estes novos sistemas, tipicamente permitem definir quem e quando é que determinadas pessoas podem aceder a uma determinada área/zona/sala e o posterior registo do acesso. Normalmente à entrada e em alguns casos também à saída de cada zona, junto à porta, existe uma UAD para efetuar a aquisição da identificação da pessoa/quem pretende entrar ou sair. A abertura da porta, barreira ou elevador encontra-se pendente de uma validação que verifica as informações das permissões pré-definidas, que especificam quem e quando é que se tem acesso a uma determinada zona. Caso a validação seja positiva, existe um registo da seguinte informação: 1) Identificação do colaborador/pessoa, 2) Zona que teve acesso, 3) Data e hora de acesso e 4) Tipo de acesso (e.g. entrada ou saída). Grande parte destes sistemas efetuam também o registo de tentativa de acesso não autorizado.

Em determinadas áreas de negócio, como parque de estacionamento, estes sistemas muitas das vezes efetuam o controlo de restrições dos lugares/recursos disponíveis permitindo ou não o seu acesso. Outra restrição é só permitir o acesso a uma determinada área/zona/sala em um determinado período/horário.

A Figura 3 apresenta utilizando a representação UML, um cenário possível no controlo de acessos.

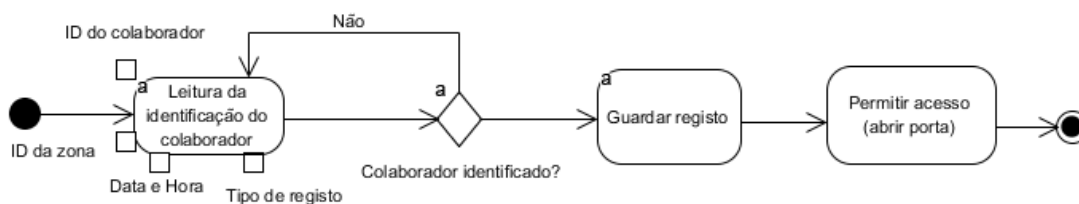


Figura 3 – Cenário possível no controlo de acessos

2.1.3 Controlo e Gestão dos Tempos de Produção e Qualidade

Os tempos de produção são fatores primordiais na gestão do processo produtivo de um produto ou serviço. Na generalidade dos casos, os tempos de produção são determinantes para a competitividade e sobrevivências das organizações.

A realização do controlo e gestão dos tempos de produção nas organizações aumenta o valor das mesmas, uma vez que o principal objetivo da produção de um produto ou serviço é o planeamento da utilização dos recursos disponíveis. Este planeamento proporciona a necessidade da existência de um controlo e gestão dos tempos de produção. Na prática a programação do tempo de produção de um produto resulta num plano de atividades/processos faseados no tempo.

Nas empresas, a produção de um produto ou serviço implica a realização de um processo que nas generalidades das vezes, é constituído por vários subprocessos. Estes subprocessos podem ser divididos em processos que gerem a informação da produção (e.g. informação sobre o stock de um matéria-prima) e por processos que permitam perceber/adquirir a informação referente à produção (e.g. qual a ordem, qual a matéria-prima utilizada, quem efetua a produção, tempo de início e tempo de fim). É sobre este último tipo de processos, processos de aquisição de informação no momento de produção, que se pretende abordar.

Os objetivos do controlo e gestão dos tempos de produção são perceber quais os recursos, o tempo e os tipos de paragem que envolvem a produção de um determinado produto. Atualmente, graças à transição para os novos processos de produção criados pela revolução industrial, muitos produtos são produzidos em serie, onde cada produção pode estar fragmentada em várias etapas/processos. O controlo do tempo de produção implica que seja necessário obter o tempo despendido na produção de cada etapa/processo, desta forma obtém-se dados que podem levar a alterações estratégicas nas linhas de produção. Como por exemplo, uma etapa/processo que seja mais demorado, pode ser alterada para ser realizada em paralelo. Ainda, perceber quais foram os tipos de paragem existentes e respetiva causa numa produção é crucial para o planeamento da produção de um produto. Pode-se entender como tipos de paragem as seguintes ocorrências:

- **Falta de matéria-prima:** nesta ocorrência é necessário especificar qual a matéria em causa;
- **Avaria nos recurso:** este acontecimento requer a identificação do recurso que originou a paragem (e.g. qual máquina/equipamento, falta de mão de obra).

A informação obtida pela utilização de um sistema que permita controlar os tempos de produção, pode viabilizar que seja possível controlar quem realizou e quais os materiais/matéria-prima que foram consumidos para a criação de um produto. Com esta informação sempre atualizada num sistema de gestão, que represente os processos de negócio é possível gerar alertas (eventos) que permitam auxiliar a gestão da logística/encomendas dos materiais de produção.

Como exemplo, para simplificar a compreensão dos processos de aquisição de informação, também denominados de diálogos de negócio, pretende-se abordar o sistema de montagem de um computador.

Na montagem de um computador pretende-se considerar necessárias os seguintes componentes: 1) caixa/torre, 2) fonte de alimentação, 3) placa mãe, 4) processador, 5) cooler da CPU, 6) memória RAM e 7) disco rígido. O diálogo deverá informar ao colaborador os passos que deve realizar por uma determinada ordem, deve ainda em cada passo requisitar qual o componente utilizado para ser possível validar o processo passo a passo. É necessário que em cada passo o colaborador introduza o número de serie do componente que utilizou (pode-se utilizar leitura de código de barras para agilizar este processo). A aquisição da informação do componente que é colocado em cada computador pode ser fundamental para o controlo de qualidade do produto, pois posteriormente a informação recolhida pode ser muito relevante no serviço de assistência. Com esta informação armazenada e com a informação futuramente adquirida no serviço de assistência é possível fazer uma análise sobre a qualidade dos componentes utilizados. Esta análise pode ser importante para as estratégias e posições futuras das organizações.

Para entendermos melhor o processo de produção de um computador vamos considerar o seguinte cenário: é necessário iniciar o processo introduzindo na UAD o número da ordem de serviço. A introdução deste número vai permitir que em cada etapa/passos da montagem seja possível validar se o componente utilizado pertence ao modelo em questão. Após a introdução da ordem de serviço, é necessário efetuar as seguintes etapas na UAD: 1) selecionar ordem de serviço, 2) identificar ordem de serviço, 3) identificar caixa, 4) identificar fonte de alimentação, 5) identificar placa mãe, 6) identificar processador, 7) identificar cooler, 8) identificar memória RAM, 9) identificar disco rígido e 10) concluir/registar processo de montagem. A Figura 4 apresenta um diagrama de atividades, representado em UML, que ilustra este processo de produção.

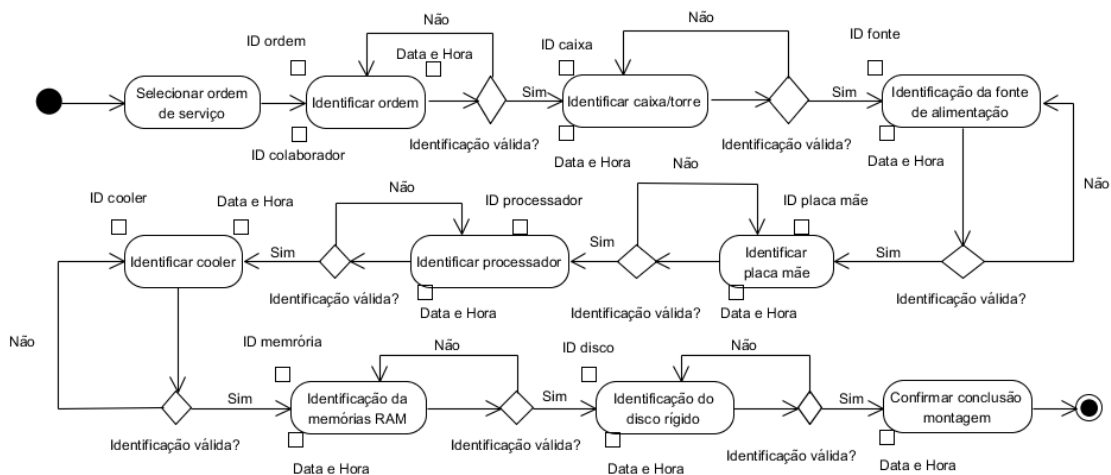


Figura 4 – Cenário de aquisição de dados na montagem de um computador

2.1.4 Sumário

Nas áreas de negócio apresentadas, encontra-se grandes semelhanças nos diálogos necessários para a aquisição de dados, nas áreas de controlo e gestão de tempos de presença e no controlo de acessos. Nestes processos a única diferença é a abertura de portas, barreiras ou elevadores que é necessário na área de controlo de acessos. Os dados adquiridos nestas duas áreas são praticamente os mesmos, a única exceção é que área de controlo de acessos é necessário recolher a identificação de qual a área/zona/sala que foi acedida. Em alguns casos o acesso a uma área/zona/sala (e.g. parque de estacionamento) efetua também o registo de presença.

Por sua vez, a área de controlo e gestão dos tempos de produção e qualidade, os dados de aquisição dependem do tipo de produção. Normalmente os dados de aquisição comuns na produção de um produto são: registo do colaborador que efetuou-a o serviço, número de ordem de serviço, registo de quais as matérias/matérias-primas utilizadas, registo das paragens de serviço e respetiva causa, entre outros dados.

Com tantos dados comuns nestas áreas, é possível reutilizar/adaptar diálogos de negócio existentes para outros clientes.

2.2 Soluções Existentes

Esta subseção tem o objetivo de apresentar e caracterizar as soluções existentes nos mercados, nacional e internacional, para as áreas de negócio abordadas.

2.2.1 Sistemas de Controlo e Gestão de Tempos de Presença

Neste trabalho foram estudados os seguintes sistemas de gestão e controlo de tempos de presença: 1) BioStar 2 (Suprema, 2016a), 2) ZEUS® Time & Attendance (ISGUS, 2016a), 3) Time

HR (Elo Sistemas de Informação, 2016a) e 4) InnuxTime (Innux Technologies, Lda, 2016b). O Time HR e o InnuxTime são sistemas de empresas portuguesas.

Através do estudo realizado, conclui-se que o funcionamento em todos os sistemas é muito semelhante. Todos eles utilizam o seguinte padrão de funcionamento:

Utilização de UADs: distribuição de UADs pela organização. Estes equipamentos têm como funcionalidade adquirir os registos de assiduidade dos colaboradores e armazená-los em memória própria, armazenamento *offline*, ou transmitir os dados recolhidos para as aplicações *backoffice*, armazenamento *online*.

Aplicação *Backoffice*: estas aplicações são responsáveis pela gestão dos dados necessários, quer dos colaboradores, dos seus respetivos horários, entre outros dados. Ainda, é responsável por atualizar e recolher as informações relativas à autenticação/validação dos colaboradores nos terminais e do processamento da informação recolhida.

Comunicação entre a aplicação e as UADs: a comunicação entre as aplicações e os terminais pode ser realizada das seguintes formas:

- Protocolo RS232;
- Protocolo RS422;
- Protocolo RS485;
- Protocolo TCP/IP;
- GSM;
- GPRS;
- ISDN;
- Flash disk.

Os sistemas de gestão (*backoffice*), como o InnuxTime e o BioStar 2, possuem rotinas que efetuam a comunicação com as UADs com o objetivo de efetuar recolha e atualização dos dados existentes neles. O sistema InnuxTime permite que determinadas UADs (Innux Technologies, Lda, 2016c) utilizem o protocolo de comunicação GSM, em que é a UAD que envia os dados utilizando o serviço de SMS como forma de transferi-los à aplicação. Outro método usado na recolha de dados nos sistemas *offline*, é a recolha de informação através do uso do cartão de memória existente no terminal, as UADs da Kaba [XXX] e da Innux permitem esta opção.

A Figura 5 apresenta o funcionamento padrão de um sistema de controlo e gestão de tempos de presença. Nesta figura é possível perceber que a aplicação é que é responsável por gerir a

informação e recolhe-la das UADs. Nestes sistemas, a aquisição de informação nas UADs está dependente de uma sincronização com os dados existentes no sistema de gestão.

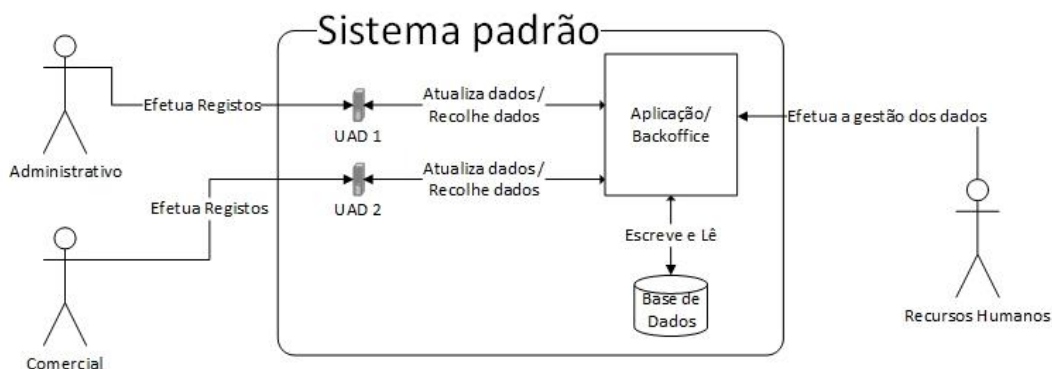


Figura 5 - Funcionamento Padrão dos Sistemas de Controlo e Gestão de Tempos de Presença

Com a análise das funcionalidades dos sistemas apresentados, pode-se concluir que todos os sistemas possuem funcionalidades similares. Assim, apresenta-se as principais funcionalidades comuns:

1. **Recolha automática de marcações:** os sistemas é que são responsáveis pela recolha das marcações existente nas UADs. O sistema InnuxTime dispõe de uma área de configurações, onde é possível configurar rotinas que efetuam a recolha dos registos de assiduidade existente nas UADs.
2. **Configurar terminais de ponto (UADs):** nesta funcionalidade, é possível acrescentar e eliminar UADs.
3. **Ficha de colaborador:** esta opção permite que seja possível definir as informações dos colaboradores (e.g. nome, morada, telefone, telemóvel, email, foto e password de acesso) e o seu respetivo perfil;
4. **Gerir Controlo de acessos e perfis de acesso:** permite configurar os tipos de perfis de acesso (dias e horário) e distribuir estes perfis pelas respetivas UADs.
5. **Configurar códigos de ausência:** permite criar os diferentes tipos de ausências.
6. **Códigos de incidência:** possibilita a definição de diferentes tipos de códigos de saída em serviço.
7. **Horários:** possibilita a definição de vários horários de trabalho.
8. **Configurar planos de trabalho:** permite registar as regras que são aplicadas às marcações efetuadas num único dia, os planos de trabalho contêm o padrão de

horários a serem efetuados dentro de um conjunto de dias. Inclui distribuição de horários e substituições.

9. **Registar feriados:** possibilita a introdução das datas dos feriados.
10. **Registar ausências:** permite registar o tipo de ausência. Estes tipos podem ser dias de férias, baixa médica, consultas médicas, formação, entre outros.
11. **Processar marcações:** este módulo permite listar as marcações dos colaboradores e ainda que se efetuem eventuais ajustes dos dados.
12. **Consultar marcações de ponto:** permite consultar as marcações efetuadas nos terminais de ponto para efeito de controlo de presença.
13. **Consultar marcações irregulares:** possibilita consultar irregularidades cometidas no uso dos terminais de controlo ponto e acessos. Os tipos de irregularidades registados pelos terminais são os seguintes: 1) Leitura de cartão não registado, 2) Leitura de cartão inválido, 3) Leitura de cartão fora do seu horário de acesso, 4) Abertura forçada do terminal e 5) Impressão digital inválida.
14. **Consultar acumulados:** permite visualizar os totais de trabalho ou ausência calculados pela aplicação.
15. **Consultas de ausente e presente:** permite visualizar no momento instantâneo qual o estado de um funcionário a nível de presente ou ausente.

Com a análise das funcionalidades anteriormente descritas, é possível perceber que os dados de configuração e parametrização são realizados em uma aplicação, que funciona como *backoffice* para as UADs. Esta aplicação efetua regularmente a leitura dos dados existentes nas UADs, com o objetivo de efetuar o processamento e registo desta informação numa base de dados.

A Figura 6 apresenta a janela inicial do *backoffice* do sistema InnuxTime.

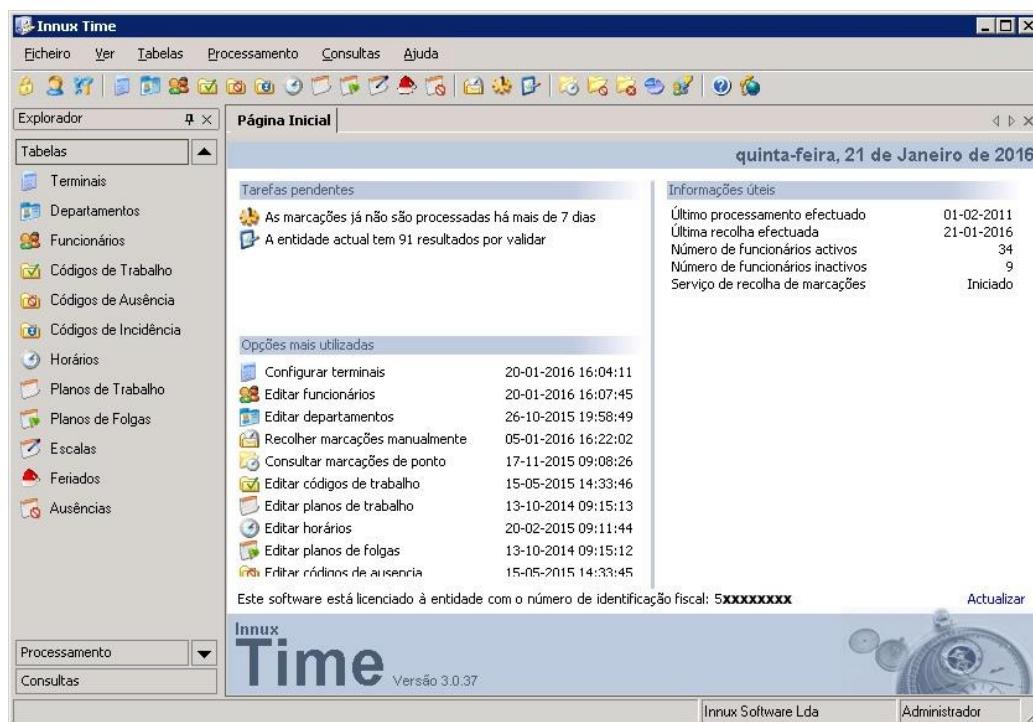


Figura 6 - Backoffice do InnuxTime (Innux Technologies, Lda, 2016b)

A empresa Suprema (“Suprema,” 2016g) é um fornecedor internacional de UADs. Desenvolveu um SDK, o BioStar 2 SDK (Suprema, 2016b), com o objetivo de simplificar a comunicação entre os sistemas *backoffice* que gerem a informação e os seus terminais/UADs que recolhem a informação.

A Figura 7 ilustra a arquitetura deste SDK, onde é possível verificar que dispõe de um conjunto de APIs que permitem facilmente interagir diretamente com os terminais. Além disso, suporta várias linguagens de programação, como o C ou o C# com o objetivo de funcionar em diferentes sistemas operativos. Empresas como a Elo (“Elo - Sistemas De Informação, Lda,” 2016) utilizam os terminais desenvolvidos pela Suprema, onde este SDK é a ponte entre as suas aplicações de gestão e os respetivos equipamentos.

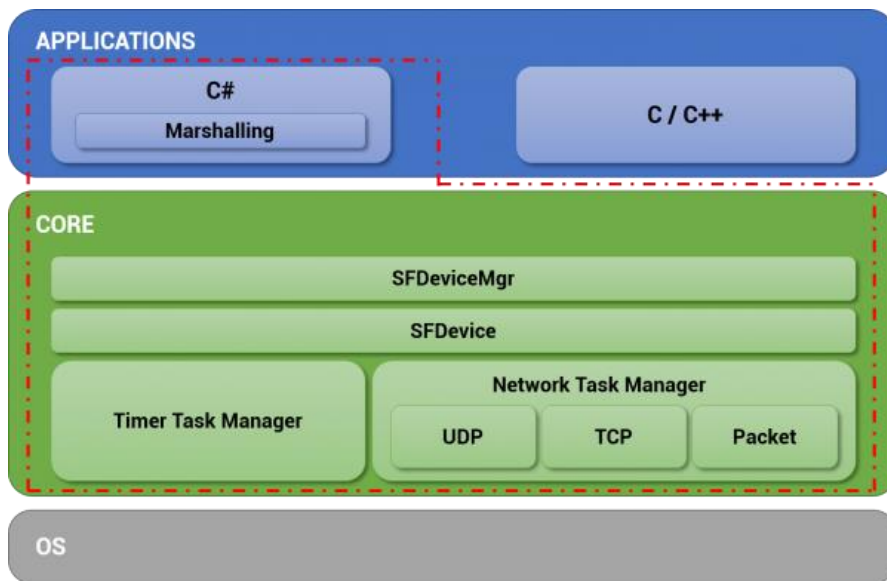


Figura 7 – Arquitetura do BioStar 2 SDK (Suprema, 2016b)

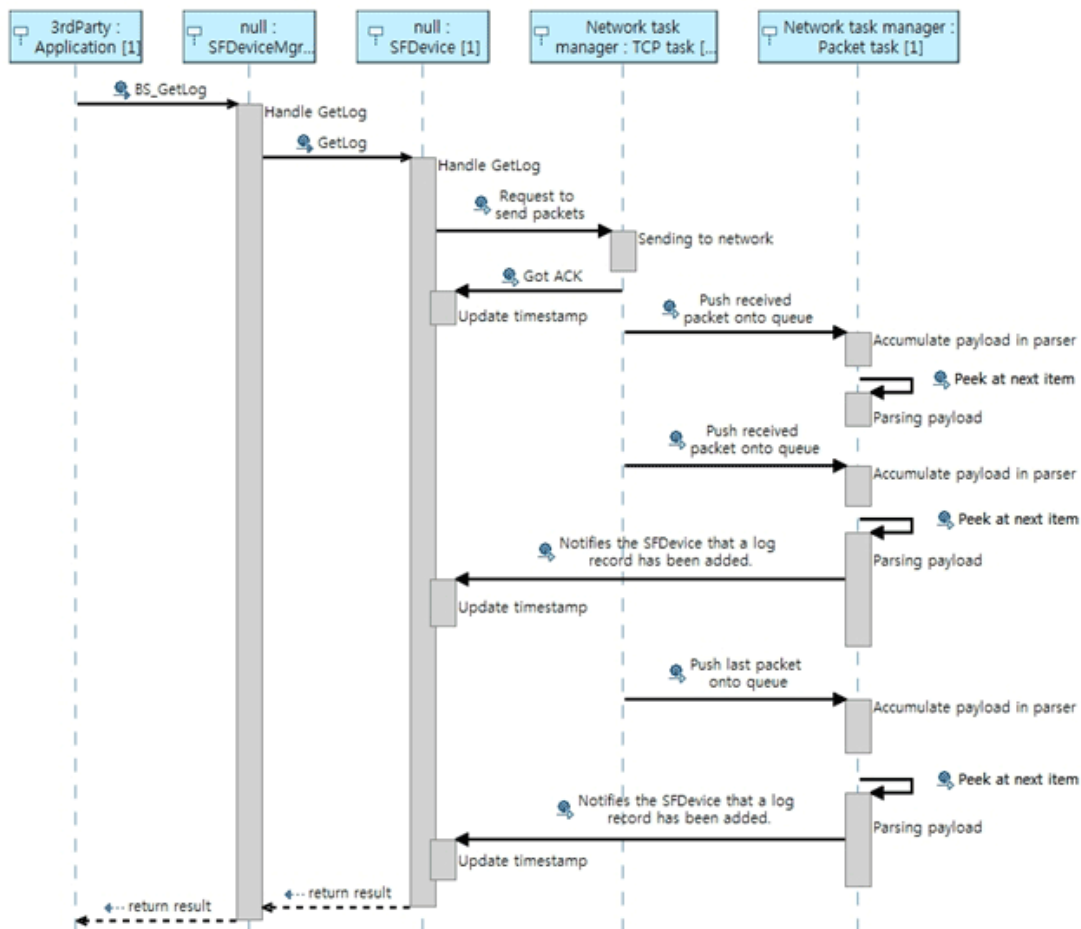


Figura 8 – BioStar 2 SDK Workflow (Suprema, 2016b)

A Suprema também disponibiliza BioStar 2 API (Suprema, 2016c), que consiste numa aplicação web, que fornece os mecanismos necessários para os desenvolvedores criarem aplicações ou integrar o BioStar 2. Esta API permite realizar a gestão da informação necessária nos seus terminais, como os dados dos utilizadores. O BioStar 2 API encontra-se desenvolvido numa arquitetura REST utilizando o formato de dados JSON, para efetuar solicitações e resposta de forma a facilitar a sua compreensão. O objetivo desta aplicação, é que a utilizem como base de forma a facilitar a integração com os terminais que comercializam.

2.2.2 Sistemas de Controlo de Acessos

Após a análise dos sistemas que permitem efetuar o controlo de acessos a áreas restritas, como o InnuxAccess (Innux Technologies, Lda, 2016d), BioStar 2 (Suprema, 2016a) e ZEUS® Access Control (ISGUS, 2016a), compreende-se que estes sistemas são muito semelhantes aos sistemas de controlo e gestão de tempos de presença. Em todos os casos, são uma expansão das funcionalidades dos sistemas de controlo e gestão dos tempos de presença. A principal diferença destes sistemas é que os terminais tem a responsabilidade de efetuar uma abertura automática de portas, barreiras ou elevadores.

A Figura 9 apresenta o menu inicial do sistema de gestão BioStar 2 da empresa Sul Coreana a Suprema (Suprema, 2016b).

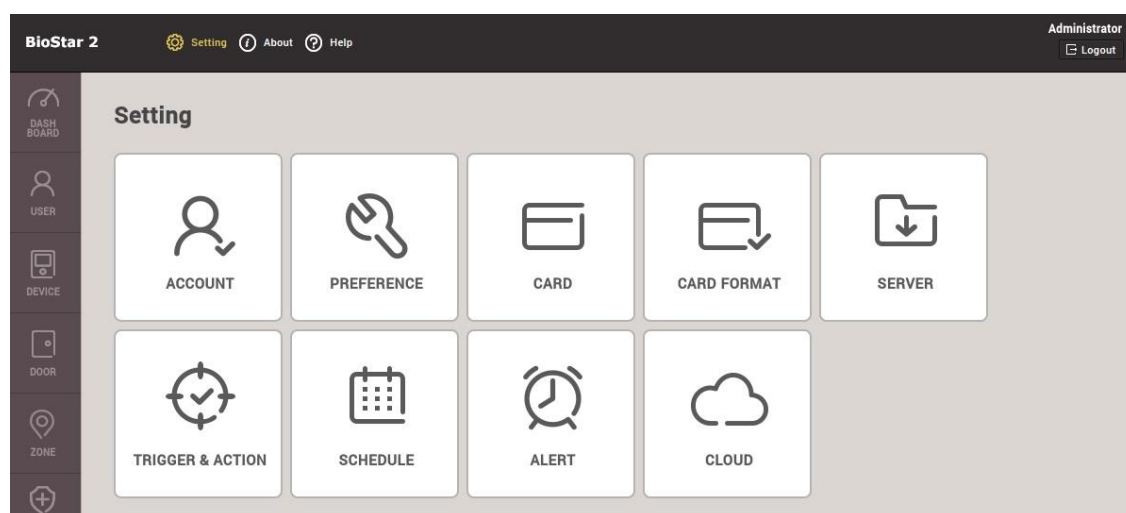


Figura 9 - Sistema Backoffice BioStar 2 (Suprema, 2016a)

Ao analisarmos a Figura 10 e a Figura 11, é perceptível a compreensão que ambos os sistemas apresentados são configurado da mesma forma. Ambos utilizam um computador como servidor, um terminal/UAD para a aquisição dos dados, que posteriormente efetua a abertura da porta e a comunicação entre os terminais e o servidor é efetuada através da rede existente, utilizando o protocolo TCP/IP.

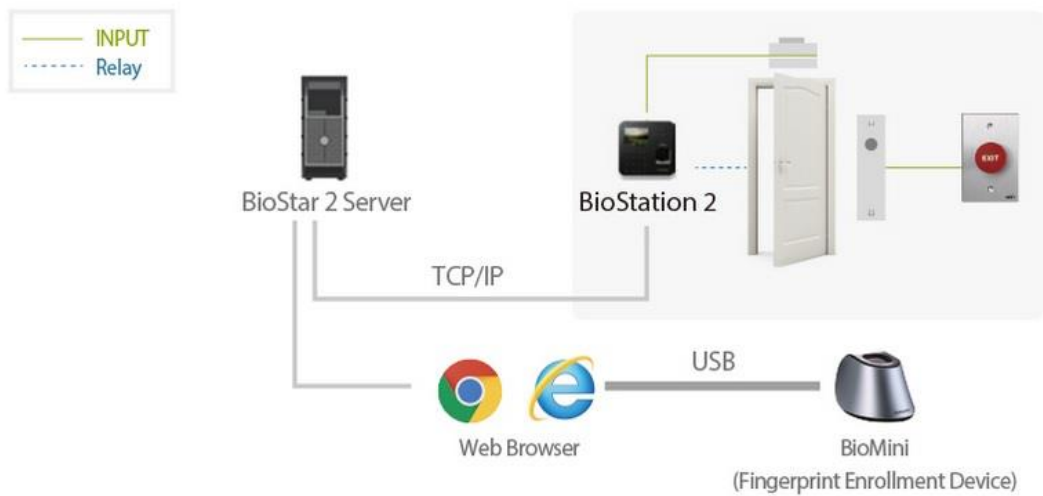


Figura 10 – Configuração do Sistema BioStar 2 (Suprema, 2016a)

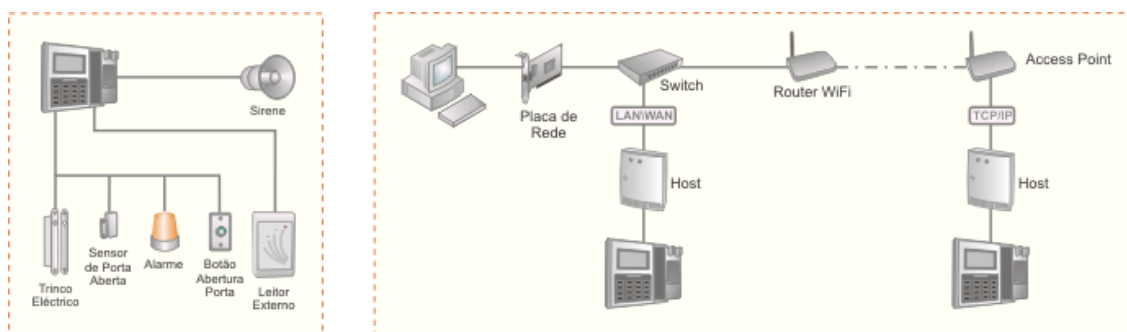


Figura 11 – Configuração do Sistema InnuxAccess (Innux Technologies, Lda, 2016d)

O funcionamento dos sistemas de controlo de acesso, apresentado na Figura 12, é semelhante ao funcionamento dos sistemas de controlo e gestão de tempos de presença. A única diferença é que, no controlo de acessos estes sistemas são responsáveis pela abertura de porta, barreira ou elevadores.

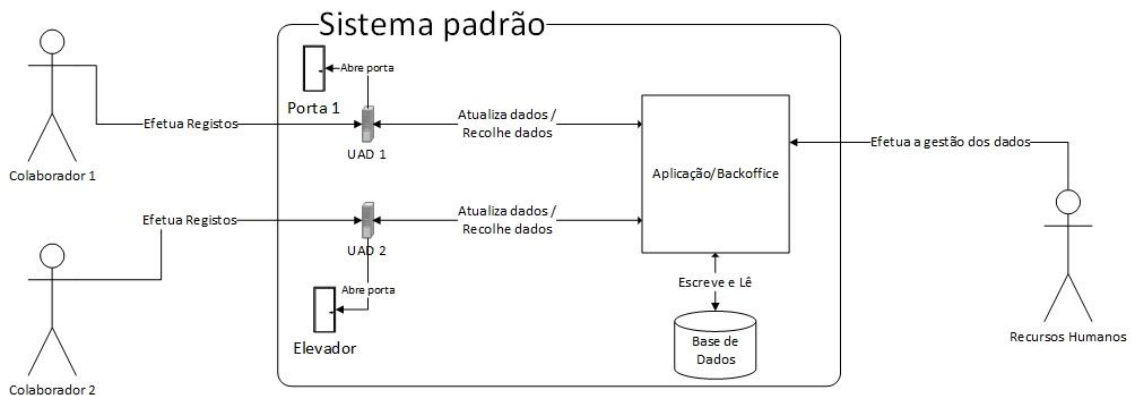


Figura 12 - Funcionamento Padrão dos Sistemas de Controlo de Acessos

2.2.3 Sistemas de Controlo e Gestão dos Tempos de Produção e Qualidade

Na realização deste trabalho foram analisados os seguintes sistemas de gestão e controlo de produção: 1) Task Manager (Elo Sistemas de Informação, 2016b) e 2) ZEUS® Plant Data Collection solution (ISGUS, 2016b).

O Task Manager (Elo Sistemas de Informação, 2016b) é caracterizado como um sistema que permite desenhar os processos de negócio de aquisição de informação e de interação com os utilizadores. Possibilita definir quais as consultas, relatórios e alarmes automáticos que resultem de ocorrências de fatores dinâmicos nos processos, como por exemplo, desvios, avarias ou não conformidades. A recolha de informação neste sistema, dependendo das situações pode ou não estar sujeita a validações. Estas validações podem ser efetuadas *online* ou *offline*. As validações *online* são caracterizadas por interações aos sistemas de gestão em tempo real, já as validações *offline* são as que são efetuadas nas UADs. A validação *offline* em ambientes onde não é possível a comunicação com os sistemas de gestão, torna-se num requisito crucial para a correta aquisição de dados. O Task Manager foi desenvolvido numa filosofia Cliente/Servidor. As UADs utilizadas são equipamentos específicos para a recolha de dados, e utilizam os seguintes protocolos de comunicação: 1) RS485, 2) TCP/IP, 3) GSM e 4) GPRS.



Figura 13 - UAD do sistema Task Manager, modelo terminal da empresa X64 (X64, 2016a)

O sistema ZEUS® Plant Data Collection solution (ISGUS, 2016b) é descrito como um sistema que aumenta a produtividade e reduz os custos de produção. Caracterizado por conseguir

implementar os processos de aquisição, que definem as sequências operacionais da produção de um produto. Utiliza os terminais que a empresa comercializa, como UADs para efetuar de forma rápida e eficaz a aquisição de dados. Estas UADs possibilitam que após a recolha de informação, esta seja transferida para os sistemas de gestão usando os seguintes métodos:

- Gravação em dispositivos móveis (e.g. cartões de memória);
- Através da rede Ethernet, é efetuada a transmissão de dados de forma direta entre as UADs e o sistema de gestão.

Este sistema dispõe de métodos de validação para uma correta aquisição de dados. Um dos grandes objetivos descritos neste sistema é que a sua utilização, permite em tempo real e com exatidão identificar quais os processos mais cruciais na produção de um produto. Assim, os gestores conseguem otimizar melhorias significativas nas suas produções. As vantagens enumeradas na sua utilização são as seguintes:

- Aceleração dos processos;
- Cumprimento dos prazos programados;
- Controle dos custos de produção;
- Otimização do cronograma de planeamento;
- Descrevem ainda, que este sistema permite a fácil integração com um sistema ERP.



Figura 14 – UAD do sistema ZEUS® Plant Data Collection solution, modelo IT 8001(ISGUS, 2016c)

Após a análise dos sistemas apresentados, é perceptível as semelhanças entre ambos. Em ambos é necessário UADs para uma fácil e eficaz aquisição dos dados, ambos utilização métodos de validação. O sistema Task Manager evidencia a capacidade de conseguir efetuar validações nas UADs. A troca de informação entre as UADs e os sistemas de gestão são efetuadas usando alguns métodos idênticos, como ilustra a Tabela 1.

Tabela 1 – Métodos de transferência de dados entre as UADs e as aplicações

	ZEUS® Plant Data Collection solution	Task Manager
RS485		x
TCP/IP	x	x
GSM		x
Cartão de memória	x	
GPRS		x

2.2.4 Sumário

Através do estudo realizado, é perceptível perceber que os sistemas existentes para as áreas do controlo e gestão de tempos de presença, controlo de acessos e controlo e gestão dos tempos de produção e qualidade são tipicamente constituídos pelos seguintes dois tipos de componentes:

- **Unidades de Aquisição de Dados (UADs):** equipamentos desenvolvidos propositadamente com características específicas, para que de forma simples e rápida consigam proceder à recolha da informação necessária em cada área de negócio. Podem funcionar em modo *online*, com comunicação aos sistemas de gestão, ou em modo *offline*, guardando os dados recolhidos em memória própria que posteriormente são recolhidos manualmente.
- **Aplicação que permite a gestão da informação (*backoffice*):** esta aplicação permite a configuração dos parâmetros necessários para o funcionamento destes sistemas. Ainda, é a responsável pela sincronização dos dados necessários nas UADs, em modo *online*, para que estes equipamentos consigam efetuar a recolha correta dos dados. Posteriormente os dados recolhidos são processado por esta aplicação.

2.3 Caracterização das UADs

As UADs são essenciais para o funcionamento das soluções apresentados na seção 2.2. Estes equipamentos são desenvolvidos propositadamente com o objetivo de simplificar a recolha da informação necessária, nas respetivas áreas de negócio. O seu preço pode variar conforme as necessidades do cliente, uma vez que são equipamentos modulares. Por exemplo, um equipamento pré-configurado que traga só teclado como método de aquisição de dados, se o cliente desejar que tenha leitor de código de barras o equipamento pode ser adaptado.

Na realização deste trabalho, foram analisadas as UADs dos seguintes fabricantes: Suprema (“Suprema,” 2016g), Kaba (“KABA,” 2016c), ISGUS (“ISGUS,” 2016g), X64 (X64, 2016a) e Innux (“Innux Technologies, Lda,” 2016g). A Innux é a única empresa portuguesa estudada que desenvolve as próprias UADs e que as comercializa para outras empresas, como a Projecttime

("Projecttime," 2016) e a Smartstep ("Smartstep," 2016), juntamente com os sistemas de gestão que desenvolve. A empresa Elo ("Elo - Sistemas De Informação, Lda," 2016) tem parcerias com as empresas Suprema, X64 e ISGUS para utilizar as suas UADs como equipamentos de aquisição de dados para os sistemas que desenvolve. A empresa STILC ("STILC, Lda.," 2016) utiliza as UAD da empresa Kaba.

A aquisição da informação nas UADs requer que estas disponham de métodos de aquisição, que consigam processar e armazenar a informação recolhida (modo *offline*) e métodos de comunicação para enviarem a informação para os respetivos sistemas de gestão (modo *online*), que efetuam a gestão da informação.

2.3.1 Características das UADs

As UADs são tipicamente caracterizadas pelas seguintes características:

- **Sistema Operativo:** estes equipamentos têm, na generalidade dos casos, sistemas operativos proprietários criados para a área de negócio em questão.
- **Capacidade de processamento:** tempo que demoraram a recolher e a processar a informação recolhida.
- **Capacidade de armazenamento:** número de registos que permitem armazenar.
- **Teclado:** o teclado permite uma interação com o utilizador, disponibilizando um método de entrada de dados, onde muitas vezes determinadas teclas (e.g. teclas de função, ver Figura 15) possibilitam o acesso a funcionalidades distintas. Combinação de várias teclas de função permitem funcionalidades diferentes.
- **Ecrã:** em quase todas as UADs o ecrã que permite apenas transmitir informação ao utilizador. No entanto, a empresa Suprema já disponibiliza equipamentos como o "BioStation A2" (Suprema, 2016d) em que o ecrã é sensível ao toque, onde é possível utiliza-lo como dispositivo de entrada de dados.
- **Método de aquisição de dados:** define a forma como consegue obter a informação. Existem diversos métodos de aquisição, como leitura de cartões, leitura de dados biométricos ou introdução de um código.
- **Métodos/Protocolo de transmissão de dados:** algumas UADs armazenam os dados localmente em cartões de memória, que posteriormente tem que ser recolhidos e incluídos nos sistemas de gestão manualmente. Outras UADs, já possibilitam que a informação seja enviada ou recolhida automaticamente pelos sistemas de gestão, onde é necessário a utilização de um método/protocolo de transmissão de dados.



Figura 15 – UADs com teclas de função (KABA, 2016a)

As UADs atualmente utilizadas permitem, adquirir a identificação dos colaboradores através dos seguintes métodos: 1) Código PIN, introdução de um código que identifica o colaborador, 2) uso de cartões (e.g. RFID ou magnéticos), ou 3) Biometria (e.g. impressão digital, veias, reconhecimento facial, etc.). Alguns destas UADs permitem que se efetuem combinações de vários tipos de validação, como por exemplo o cartão mais biometria ou biometria mais código PIN.

A Tabela 2 apresenta algumas das características das UADs analisadas.

Tabela 2 – UADs analisadas

Equipamento	Hardware	Métodos de aquisição	Protocolo de transmissão	Capacidade de armazenamento
BioStation 2 (Suprema, 2016e)	CPU 1.0GHz Memoria 128MB RAM + 8GB Flash Ecrã LCD 2.8" Som 16bit Hi-Fi Teclado	Biométrico RFID Código PIN	TCP/IP RS-485 RS-232 USB	Max. 500 000 registos.
Xpass S2 (Suprema, 2016f)	CPU 533MHz Memoria 16MB RAM + 16MB Flash Som Multi-tone Buzzer	RFID	TCP/IP RS-485	Max. 50 000 registos.
IT 8200 FP (ISGUS, 2016d)	Ecrã TFT 3,5" Teclado	Biométrico RFID Código PIN	TCP/IP	Max. 50 000 registos.
IT 4110 (ISGUS, 2016e)	Teclado	RFID Código PIN	RS-485	Não especificado.
NuxFinger (Innux Technologies, Lda, 2016e)	Ecrã 160x80 pixels com retroiluminação Teclado 16 teclas	Biométrico RFID Código PIN	RS-232 RS-485 TCP/IP GSM	Max. 20 000 registos.
NuxIn II (Innux Technologies, Lda, 2016f)	Ecrã 160x80 pixels com retroiluminação Teclado 16 teclas	RFID Código PIN	RS-232 RS-485 TCP/IP GSM	Max. 20 000 registos.
B-web 93 00 (KABA, 2016a)	Ecrã 320x240 pixels Teclado	Biométrico RFID Código PIN Cartão magnético	TCP/IP RS-232 RS-485 GSM	Até 200 colaboradores.
Reader 9110 (KABA, 2016b)	Não especificado.	RFID	RS-485	Não especificado.
IT 8001 (ISGUS, 2016c)	Ecrã Teclado Leitor de código de barras	RFID Código PIN Cartão magnético Código de barras	TCP/IP	Max. 50 000 registos.
IT 900 (ISGUS, 2016f)	Ecrã 6,5" Suporta JAVA	Teclado com teclas de função.	TCP/IP	Não especificado.
XRT707 (X64, 2016b)	Leitor de código Bateria 650 mAh	RFID Código de barras Teclado com teclas de função.	433 MHz wireless	Não especificado.

Tabela 3 – Métodos de aquisição suportados pelas UADs

Métodos de aquisição	Fabricantes				
	Kaba	ISGUS	Innux	Suprema	X64
Biometria - dedo	X	X	X	X	
Código PIN	X	X	X		X
Cartão Cidadão			X		
Biometria - palma da mão			X		
Cartão Magnético	X				
Códigos barras	X	X			X
RFID - Mifare	X	X		X	
RFID - EM				X	
RFID - HID	X	X		X	
RFID - Hitag		X			
RFID - LEGIC	X	X			
RFID - Indutivo					

Tabela 4 – Método/Protocolo de transmissão utilizado nas UADs

Método/Protocolo de transmissão	Fabricantes				
	Kaba	ISGUS	Innux	Suprema	X64
RS232	X	X	X	X	X
RS422	X	X		X	
RS485	X	X	X	X	
TCP/IP	X	X	X	X	
GSM	X		X		
Cartão de memória			x	X	
ISDN		X			
GPRS				X	

2.3.2 Sumário

Analisando as principais características das UADs existentes, é perceptível que as UADs são desenvolvidas propositadamente com as características necessárias para recolher a informação necessária. Estas características são semelhantes às utilizadas num computador, representam um conjunto de componente de *hardware* específicos (e.g. processador, memória, ecrã, etc.) e possuem em alguns casos, um sistema operativo proprietário otimizado para controlar o *hardware* com o objetivo da correta aquisição dos dados pretendidos.

3 Especificação de Diálogos (e Processos) de Negócio

Este capítulo tem como objetivo descrever uma especificação detalhada sobre os diálogos de negócio e o seu relacionamento com os processos de negócio. Ainda, apresenta uma análise de valor onde, é descrito o problema existente nos sistemas atuais e qual o raciocínio que foi seguido para apresentar a visão da solução.

3.1 Caracterização dos Diálogos de Negócio

Um diálogo de negócio neste trabalho representa um subprocesso de negócio, com o objetivo de simplificar a aquisição de dados através de uma interação homem-máquina.

Pode-se caracterizar os diálogos de negócio nos seguintes conceitos:

- Representam fluxos de atividades de aquisição;
- Definem quais os métodos de recolha dos dados;
- Percebem quais os dados de entrada que são necessários em cada atividade;
- Têm o conhecimento necessário para efetuar ou solicitar validações nos dados;
- Com capacidades de decisão para determinar qual a atividade seguinte;
- Com o objetivo de recolher um conjunto de informação estruturada e sistematizada, os dados de saída;
- Descrevem quais os métodos de transmissão que devem ser usados, para transmitir os dados para as aplicações de gestão;
- Permitem guardar os dados recolhidos em armazenamento local ou remoto.

Por outras palavras, pode-se afirmar que um diálogo de negócio representa um modelo que define um conjunto de regras, para permitir uma correta aquisição e transmissão dos dados

pretendidos. Este modelo deve ser instanciado nas UADs, de modo que, estes equipamentos consigam interpretar e transmitir a informação necessária a recolher.

A Figura 16 representa um diálogo de negócio, onde os números representam os seguintes conceitos: 1) entrada de dados, 2) atividade de aquisição, 3) decisão ou validação, 4) transmissão de dados e 5) dados de saída.

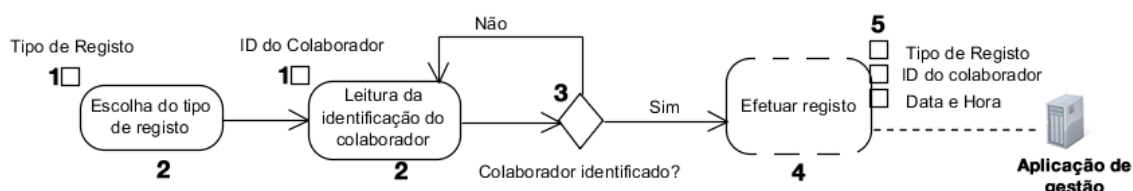


Figura 16 – Conceitos de um diálogo de negócio

3.2 Fluxo de Atividades de Aquisição

O fluxo de atividades de aquisição representa as atividades que definem as questões e os métodos necessários para a aquisição dos dados. Este fluxo é que determina as diferentes etapas na aquisição dos dados definindo quais os métodos (e.g. teclado, biometria, leitura de código de barras) possíveis para a recolha dos dados. Este fluxo pode definir ciclos de aquisição através do uso de condições.

Nas áreas de negócio apresentadas, as questões possíveis (Elo Sistemas de Informação, 2014), existem outras, nas atividade de aquisição são as seguintes:

Tabela 5 – Questões possíveis nos diálogos de negócio

Questão	Controlo e Gestão de Tempos de Presença	Controlo de Acessos	Sistemas de Controlo e Gestão da Produção
Código da entidade			X
Código de defeito			X
Código de paragem	X		X
Código do armazém			X
Código do componente			X
Confirmação	X	X	X
Critério			X
Data de fim	X	X	X
Data de início	X	X	X
Data de ocorrência	X	X	X
Funcionário	X	X	X
Hora de fim	X	X	X
Hora de início	X	X	X

Hora de ocorrência	X	X	X
Informação			X
Local	X	X	X
Máquina			X
Número de serie			X
Operação			X
Ordem de fabrico			X
Posto de trabalho	X	X	X
Quantidade			X
Subordem			X
Tipo de registo	X	X	X

3.2.1 Dados de entrada (*input*)

Os dados de entrada representam os dados que devem ser adquiridos em cada atividade através de respostas às questões. Estes dados podem ser adquiridos de diversas formas. Os dados de entrada, em alguns cenários, podem ter dependências de outros diálogos de negócio executados por outros intervenientes, no processo de negócio. Um exemplo desta dependência, na produção de um produto com linha de montagem em serie, em que a referência do material produzido na etapa anterior é necessária para uma etapa posterior. Estes dados podem ser recolhidos através de diferentes métodos (e.g. teclado, leitura de código de barras, leitura de cartões, etc.).

3.2.2 Dados de saída (*output*)

Os dados de saída são os dados que se esperam alcançar com a execução de um diálogo de negócio. Em alguns casos, estes dados são os dados de entrada de outras atividades ou dados necessários para outros diálogos de negócio. A execução de um diálogo de negócio pode ter como objetivo um conjunto de dados de saída (ver Figura 4). Existem dados de saída que são adicionados automaticamente como, por exemplo, a data/hora da recolha, a identificação da UAD, etc.

3.2.3 Decisões

A aquisição de informação, dependendo dos dados recolhidos, muitas das vezes determina o caminho que o fluxo de aquisição deve tomar. Para se determinar, é necessário que existam métodos de decisão. Os métodos de decisão utilizam o conhecimento do estado atual do diálogo de negócio (e.g. atividade atual, dados adquiridos) para decidir qual é o passo seguinte. É possível através das decisões definir ciclos de aquisição.

Os métodos de decisão podem ser realizados das seguintes duas formas:

1. **Decisão local (*offline*):** a UAD é responsável em determinados estados do processo de aquisição, com base na informação recolhida, decidir qual a atividade que se deve realizar. O processo de decisão local é um método que determina, através de um processamento executado na UAD, qual o próximo passo que o diálogo deve realizar.
2. **Decisão remota (*online*):** em determinadas situações, é necessário que exista um serviço externo (e.g. Web service, REST, etc.) para decidir qual o próximo passo que o fluxo deve tomar. Um exemplo deste tipo de decisão, é quando numa produção termina um *stock* de uma matéria-prima e é necessário uma consulta ao sistema de gestão de produção, para determinar qual o próximo lote da matéria-prima que se deve utilizar.

3.2.4 Validações

As validações nos diálogos de negócio permitem perceber se os dados de entrada ou de saída em cada fluxo de atividades são os desejados. Após uma resposta pelo utilizador a uma questão, poderá ser necessário verificar se os dados inseridos (resposta) correspondem ao desejado.

Para validar os dados no momento de aquisição, é necessário os seguintes dois tipos de validações:

1. **Validação local (*offline*):** esta validação é executada na UAD. O diálogo descreve as informações necessárias para que a UAD consiga validar os dados. Estas informações normalmente são regras de negócio, que determinam o formato dos dados, ou restringem os dados através de listas de seleção estáticas. Um exemplo desta validação, no controlo de acessos, após a aquisição da identificação do colaborador é necessário verificar se ele tem acesso à zona em questão.
2. **Validação remota (*online*):** a validação remota utiliza um serviço externo (e.g. Web service, REST, etc.), onde é necessário que a UAD consiga interpretar e executar o método/protocolo de comunicação definido no diálogo. O serviço externo pode possibilitar uma consulta a um motor de regras de negócio (e.g. aplicação de gestão).

3.2.5 Armazenamento dos dados recolhidos

O diálogo de negócio descreve como é que os dados devem ser armazenados. Tipicamente é possível armazenar os dados nos seguintes tipos de repositórios:

1. **Armazenamento local (*offline*):** descreve como guardar os dados recolhidos em armazenamento local da própria UAD.
2. **Armazenamento remoto (*online*):** descreve como transmitir os dados recolhidos para o sistema de gestão de informação (*backoffice*).

3.2.6 Sumário

Após a caracterização de um diálogo de negócio, é possível afirmar que um diálogo de negócio representa um fluxo de aquisição, onde o objetivo é a recolha de determinados dados. Os dados recolhidos podem obedecer a determinadas restrições/formatos, para assegurar este objetivo existem as validações sobre os dados recolhidos. Em alguns casos, os dados recolhidos podem determinar através de processos de decisão qual é o próximo passo do fluxo de aquisição. Finalmente, os dados recolhidos podem ser armazenados em repositórios locais ou transmitidos para repositórios remotos existentes nos sistemas de gestão.

Por outras palavras, pode-se afirmar que um diálogo de negócio representa um subprocesso de negócio, com o objetivo é recolher determinados dados. Esta representação requer que se efetue um estudo sobre a modelação de processos de negócio e de quais as notações existentes para representar estes processos.

3.3 Modelação de Processos de Negócio

A modelação de processos de negócio é um conjunto de atividades interligadas que permitem representar os processos de negócio das organizações. De acordo com Eriksson e Penker (Eriksson and Penker, 2000), os métodos tradicionais de representação dos negócios das organizações, como o organograma, dividem as organizações em departamentos ou seções, efetuando uma representação vertical. Este método de documentação é limitado à forma como o negócio é construído e organizado, porque não documentam os processos transversais/horizontais das organizações que interagem com vários departamentos.

Ainda, Eriksson e Penker (Eriksson and Penker, 2000) definem que o modelo de negócio é uma abstração do funcionamento do próprio negócio. Esclarecem que este modelo deve ser composto pelos seguintes objetos: 1) objetivos, 2) recursos, 3) processos e 4) regras, conforme ilustrado na Figura 17.

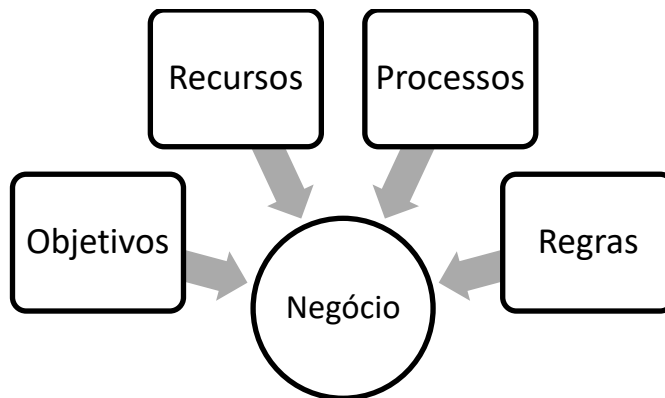


Figura 17 – Objetos que constituem um negócio (Eriksson and Penker, 2000)

Para a ABPMP (*Association of Business Process Management Professionals*) (ABPMP, 2013), a modelação dos processos de negócio constitui um conjunto de atividades envolvidas na criação de representações dos processos de negócios das organizações.

Atualmente existem diversas notações que permitem representar os processos de negócio. Entende-se como notação, um conjunto padronizado de símbolos e regras que determinam o significado desses símbolos.

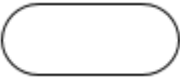



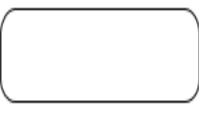













A Tabela 6 apresenta três das notações existentes, para modelar processos. Estas notações são as mais utilizadas na modelação de processo.

Tabela 6 – Notações para a modelação de processos (ABPMP, 2013)

Notação	Descrição	Características	Vantagens	Desvantagens
Fluxograma	Aprovado como um padrão ANSI (American National Standards Institute), permite esquematizar representações de processo de negócio.	<ul style="list-style-type: none"> • Usado com pistas ou sem pistas. • Várias variações para diferentes propósitos. • Conjunto central de símbolos simples. • Precursor das notações modernas. 	<ul style="list-style-type: none"> • Bem entendido por engenheiros. • Em alto nível, ajuda a criar consenso. • Fácil aprendizagem. 	<ul style="list-style-type: none"> • A existência de diversas variações. • Pode ser impreciso na definição de processos complexos. • Os objetos não possuem um conjunto de atributos descritivos.
UML Diagrama de Atividades	O UML é mantido pela OMG, o mesmo grupo que mantém o BPMN. Permite representar fluxos conduzidos por processamentos.	<ul style="list-style-type: none"> • Conjunto de técnicas de diagramas e notações relacionadas. • O conjunto de símbolos varia de acordo com o tipo de modelo. • Uso da extensão SysML (OMG, 2017). 	<ul style="list-style-type: none"> • Tipo de utilizadores bem estabelecidos. • Utilizada em muitas organizações. • Existem diversas referências bibliográficas. 	<ul style="list-style-type: none"> • As representações da notação podem variar de ferramenta.
BPMN (Business Process Model Notation)	Definido pela Object Management Group (OMG, 2011), permite projetar e modelar processos de negócio e suas transformações na linguagem de execução, a Process Modeling Language (PML).	<ul style="list-style-type: none"> • Ícones organizados em conjuntos descritivos e analíticos para atender a diferentes necessidades de utilização. • Possibilita a indicação de eventos de início, intermédios e fim. • Fluxo de atividades e mensagens. • Comunicação entre organizações e colaboração entre organizações. 	<ul style="list-style-type: none"> • Uso e entendimento difundido em muitas organizações. • Polivalência para modelar as diversas situações de um processo. • Suportado por ferramentas BPMS. 	<ul style="list-style-type: none"> • Exige treino e experiência para o seu correto uso. • Dificulta visualização do relacionamento entre vários níveis de um processo.

A Tabela 7 apresenta os principais elementos usados na modelação de diálogos de negócio nas três notações estudadas.

Tabela 7 – Alguns elementos das notações para a modelação de diálogos de negócio

Elemento	Fluxograma	UML Diagrama de Atividades	BPMN
Início do processo			
Processo/ Ação/ Atividade	 Processo	 Ação	 Atividade
Subprocesso		Não especifica.	
Decisão			
Condições	Não especifica.	Não especifica.	
Representação de dados			 Dados/Dados de entrada/ Dados de saída
Fim do processo			

Após o estudo das notações de modelação, é perceptível que a notação BPMN (*Business Process Model Notation*) é mais rica em elementos de representação e mais extensa nos conceitos de modelação. Desta forma, optou-se por usar a notação de modelação BPMN. No anexo A1, é possível visualizar melhor a diversidade elementos que o BPMN possui.

3.3.1 Avaliação de ferramentas BPMN

O objetivo desta subsecção é identificar qual a ferramenta BPMN que mais se adequa às necessidades deste projeto. A escolha desta ferramenta obriga a identificar quais os critérios necessários, para selecionar a correta ferramenta de modelação.

Para determinarmos quais as possíveis ferramentas de modelação, estabelecemos as seguintes quatro categorias de critérios:

1. **Modelação:** esta categoria representa a capacidade que a ferramenta possui para descrever os processos de negócio;
2. **Ambiente de desenvolvimento:** são as características que a ferramenta dispõe para o desenvolvimento de um processo de negócio, com o objetivo de agilizar e simplificar o processo;
3. **Integração:** capacidade de facilitar a integração através do uso de subprocessos;
4. **Termos de utilização:** tipo de licenciamento da ferramenta.

A Tabela 8 descreve quais os critérios fundamentais para cada categoria, para que seja possível escolher a ferramenta de modelação, classificando-os por importância.

Tabela 8 – Critérios de avaliação de ferramentas de modelação BPMN

Categoria	Critério	Classificação	Descrição
Modelação	Suporte ao padrão BPMN 2.0.	Fundamental	A ferramenta de modelação deverá permitir descrever os processos através da notação BPMN 2.0
	Estender componentes	Fundamental	A ferramenta deve permitir estender os componentes atuais, permitindo assim especificar novos componentes.
Ambiente de desenvolvimento	Visualização gráfica dos processos.	Fundamental	A ferramenta deverá permitir uma visualização gráfica do fluxo dos processos.
	HTML	Fundamental	A ferramenta deverá ser 100% HTML, sem a necessidade de instalação de aplicações adicionais.
	Permitir importar	Importante	A ferramenta deverá permitir abrir fluxos de processos representados através da notação BPMN 2.0, para posterior edição/alteração.
	Permitir exportar	Fundamental	A ferramenta deverá permitir exportar o fluxo de processo para um ficheiro XML.
Integração	Utilização de subprocessos.	Importante	A ferramenta deverá permitir reutilizar processos como subprocessos.
Termos de utilização	<i>Software</i> de código aberto	Fundamental	A ferramenta deverá permitir a sua utilização e alteração sem quaisquer restrições.

Foi necessário identificar diversas ferramentas que permitissem a modelação do BPMN. Para se escolher qual a ferramenta que mais se adequa às necessidades do projeto, elaborou-se a Tabela 9 onde, apresenta-se as vantagens e as desvantagens do uso de cada ferramenta identificada, através de uma comparação binária.

Tabela 9 – Critérios de Seleção da Ferramenta BPMN

	Suporte ao padrão BPMN 2.0.	Permite estender componentes	Visualização gráfica dos processos.	100% HTML	Permitir importar	Permitir exportar	Utilização de subprocessos	Software de código aberto	Comentários
BPMN.IO ("BPMN.IO," 2016)	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim	Existe a representação gráfica do subprocesso mas, é necessário estender o elemento.
GoJS ("GoJS," 2016)	Sim	Sim	Sim	Sim	Não	Não	Sim	Não	
HELFO ("HELFO," 2017)	Sim	Não	Sim	Sim	Sim	Sim	Sim	Não	Versão completa tem que ser paga. Funciona em ambiente nuvem.
Activiti ("Activiti," 2016)	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim	
Modelio ("Modelio," 2016)	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim	
JOINTJS ("JOINTJS," 2017)	Não	Não	Sim	Sim	Não	Não	Não	Não	Ferramenta tem que ser comprada Notação BPMN incompleta Necessidade de estender o componente do subprocesso
draw.io (Gaudenz, 2017)	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Não importa diagramas complexos.
SYDLE ("SYDLE BPM," 2017)	Sim	Não	Sim	Sim	Não	Não	Sim	Não	Ferramenta tem que ser comprada Permite a modelação e execução de processos de negócio.

Após a análise às ferramentas de modelação existentes e com base nos critérios de seleção definidos, optou-se por utilizar o BPMN.IO, uma vez que, é a ferramenta que mais corresponde aos critérios definidos. A Figura 18 mostra parcialmente o interface gráfico do BPMN.IO onde, é possível visualizar a área de modelação.

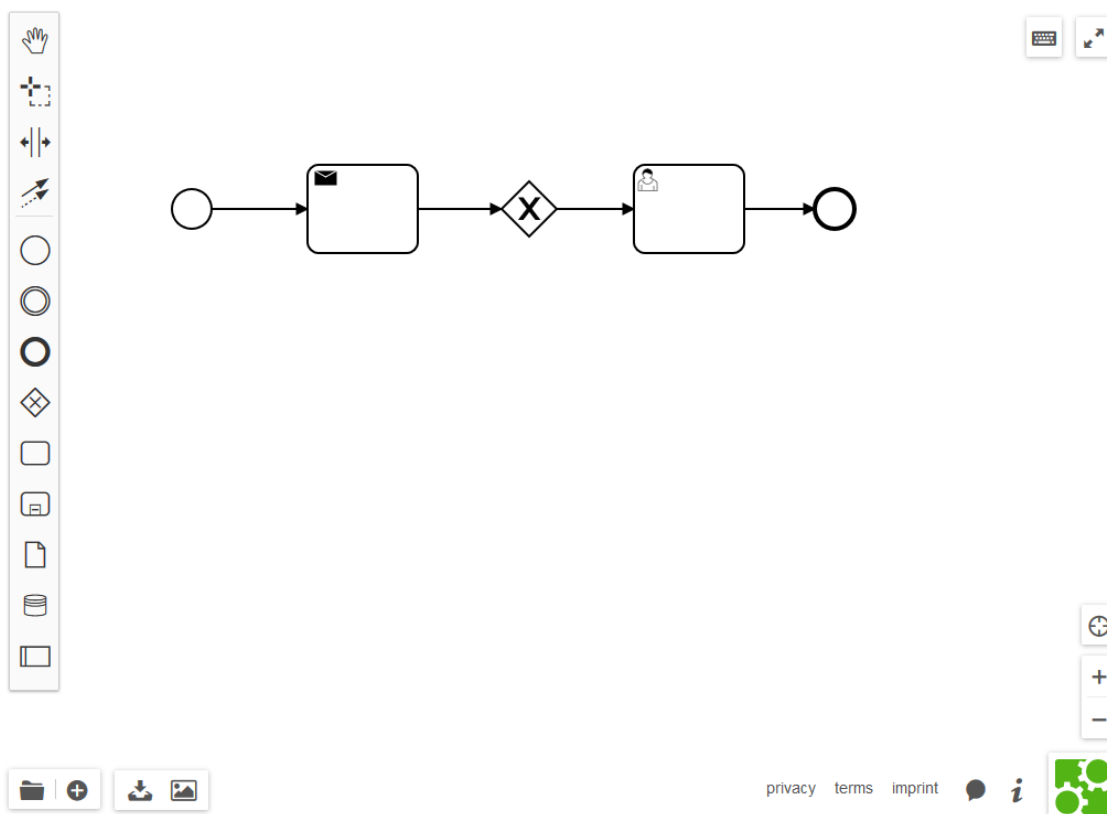


Figura 18 - Interface Gráfico do BPMN.IO

3.4 Análise de Valor

Esta secção expõe na primeira subsecção uma descrição do problema existente nos sistemas atuais. Posteriormente é apresentada uma análise de valor bem explícita com base no problema. O objetivo de uma análise de valor é esclarecer aquilo que se pode fazer para aumentar o valor num produto (bem ou serviço), com o menor custo possível, sem diminuir a sua qualidade (Nicola, 2016).

3.4.1 Problema

Através do estudo e análise realizado nas secções anteriores, sobre os sistemas existentes nas seguintes três áreas:

- 1) Controlo e Gestão de Tempos de Presença;
- 2) Controlo de Acessos a Áreas Restritas;
- 3) Controlo e Gestão de Tempos de Produção e Qualidade.

Em ambas as áreas, os sistemas são tipicamente compostos pelos seguintes dois componentes:

1. Unidades de Aquisição de Dados (UADs);
2. Aplicação que permite a gestão da informação (*backoffice*).

Um dos problemas existentes nestes sistemas relaciona-se com as necessidades da captura dos dados ser específica em cada cliente ou projeto. Por outras palavras, pode-se afirmar que se o objetivo do processo de aquisição mudar, obrigatoriamente terá de existir um novo processo de implementação ou reimplementação para alterar as UADs e as aplicações de gestão. Este novo esforço de desenvolvimento além de demorar tempo terá custos associados. Onde, o desenvolvimento apenas pode ser realizado, por profissionais qualificados e com conhecimento nos sistemas em questão.

O segundo problema, mas não menos importante, é que os sistemas de gestão, na maioria dos casos, são os responsáveis por comunicar com as UADs. Estes sistemas têm que ser programados para conseguir funcionar/comunicar com as UADs pretendidas. Existem já algumas APIs, como o BioStar 2 API (Suprema, 2016c), que possui os mecanismo necessário de interação com as UADs. No entanto, existe uma grande dependência no uso destas APIs, uma vez que, os sistemas de gestão tem que obrigatoriamente utiliza-las para comunicar com as UADs. Desta forma, é possível concluir, que a troca de UAD num novo projeto/cliente obriga a uma reimplementação/adaptação da aplicação de gestão.

Através dos problemas apresentados pode-se enumera-los da seguinte forma:

1. Captura de dados específica por cada cliente ou projeto;
2. Grande dependência entre as UADs e as aplicações de gestão existentes;
3. Implementações ou reimplementações apenas pode ser realizada por profissionais qualificados;
4. Troca do modelo da UAD obriga um novo processo de reimplementação/adaptação para a UAD conseguir comunicar com a aplicação de gestão;

3.4.2 Processo de Negócio e de Inovação

Atualmente, a inovação nos processos de negócio é fundamental para as organizações. Neste sentido, (Flynn, 2003) afirma que as empresas só podem alcançar melhorias nos seus produtos ou serviços por meio da inovação. O processo de inovação consente a criação de ideias e colocá-las em prática. Este processo permite criar oportunidades num determinado contexto. Desta forma pode-se afirmar, que a inovação nas organizações tornou-se um fator fundamental para o seu sucesso.

A importância da inovação levou à definição de vários modelos de inovação. Em 1991 (Smith and Reinertsen, 1991), definem que a inovação pode ser caracterizada como um processo que se encontra dividido nos seguintes três subprocessos: 1) *Fuzzy Front End* (FFE), 2) processo de desenvolvimento de um novo produto (*new product development* - NPD) e 3) processo de comercialização.

O FFE é o processo inicial da inovação, contendo as atividades e o tempo necessário até o desenvolvimento de um conceito de produto. O termo *fuzzy* (imprevisível) quer dizer que, este subprocesso pode ser incerto, sem existir um modelo comum de como efetua-lo. É responsável pela geração de ideias, identificação de oportunidades e a definição do conceito no processo de inovação. Desta forma, podemos individualizar os seguintes três elementos chave no FFE:

- **Oportunidade:** é uma lacuna detetada num negócio ou tecnologia, com a finalidade de capturar uma vantagem competitiva;
- **Ideia:** pode ser entendida como a forma mais embrionária de um novo produto (bem ou serviço), consiste numa vista de nível elevado da solução para resolver o problema identificado pela oportunidade;
- **Conceito:** trata-se de uma forma bem definida, pode ser uma descrição escrita ou visual, das características e benefícios de um novo produto (bem ou serviço).

No início do século XXI, Koen (Koen et al., 2001) (Koen et al., 2002), propôs um modelo de inovação para permitir uma linguagem comum sobre as atividades do FFE. Este modelo é denominado de "*The New Concept Model*" (NCD) (Figura 19) e é constituído pelas seguintes três partes principais:

- **Motor:** responsável por dirigir a área interior e é estimulado pela liderança, cultura e estratégia das organizações
- **Fatores de influência:** são as capacidades organizacionais, estratégias de negócio e ambiente externo (e.g. canais de distribuição, concorrentes, clientes, etc.);
- **Área interior:** divide-se nos seguintes 5 elementos:
 1. **Identificação de oportunidades:** trata-se da procura de lacunas em negócios ou tecnologias. Envolvem métodos de previsão do futuro, de modo que, as oportunidades possam ser escolhidas para uma análise mais aprofundada. Os métodos mais utilizados são: *roadmapping* (mapa que permite visualizar os

objetivos do desenvolvimento), análise e previsão de tendências tecnológicas, análise de inteligência competitiva, análise de tendência de clientes, pesquisa de mercado e planeamento de cenários. Estes métodos permitem verificar se a oportunidade existe;

2. **Análise de oportunidades:** a oportunidade é avaliada para confirmar que é válida. Alguns dos métodos utilizados na identificação de oportunidade são também utilizados neste elemento. Como, *roadmapping*, análise e previsão de tendências tecnológicas, análise de inteligência competitiva, análise de tendência de clientes e planeamento de cenários. Nesta análise são despendidos mais recursos que no elemento anterior;
3. **Geração e aperfeiçoamento de ideias:** é o nascimento, desenvolvimento e amadurecimento da ideia concreta. O objetivo é obrigar a organização a gerar ou modificar oportunidades identificadas. Pode ser formado através de um processo formal, incluindo sessões de *brainstorming* e banco de ideias. Outro método utilizado é o TRIZ (teoria da resolução inventiva de problemas), pois permite uma abordagem sistemática para resolver problema permitindo criar soluções com múltiplas alternativas.
4. **Seleção de ideias:** o objetivo é selecionar quais as ideias que vão ser submetidas a critérios de avaliação. As organizações deve dispor de processos formais que avaliem as ideias, recursos, como tempo e custos e quais as pessoas aptas para efetuar esta seleção, ainda numa fase inicial. Pode-se verificar a probabilidade técnica ou comercial de sucesso, qual será a recompensa,
5. **Definição de conceito:** este é o elemento final do NCD. Deve definir um argumento convincente sobre um novo conceito. Podemos utilizar critérios de avaliação para termos orientação na seleção de conceitos.

O processo FFE neste modelo inicia-se, pela identificação de oportunidades ou pela geração de ideias. Estas oportunidades ou ideias interagem com os outros elementos do modelo, interagindo também como fatores de influência e empurrados pelo motor. Apenas existe uma saída, a definição de conceito.

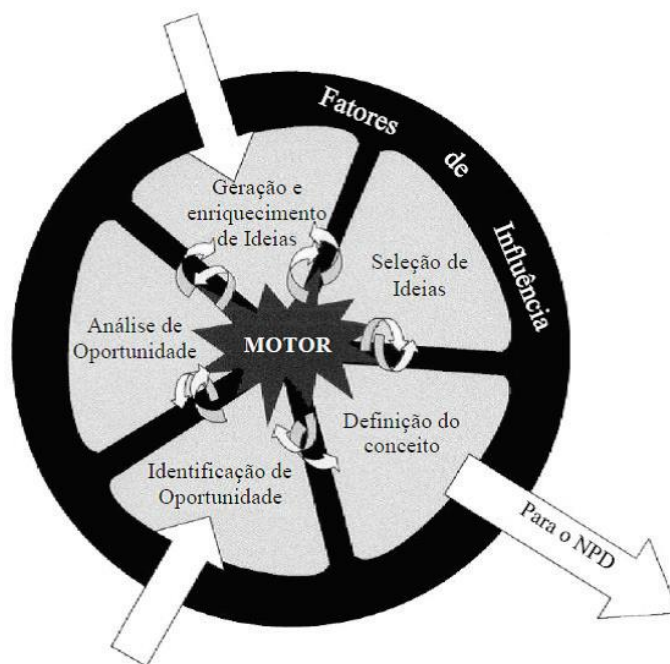


Figura 19 - Modelo *New Concept Development* NCD (adaptado (Koen et al., 2001))

De acordo com a modelo NCD, vamos enquadrar com base neste projeto os cinco elementos referentes à área interior.

1. **Identificação de oportunidades:** através de pesquisas de mercado, conseguimos verificar as seguintes lacunas existentes nos sistemas atuais: alta dependência entre as UADs e os sistemas de gestão, esta dependência impossibilita a reutilização dos diálogos de negócio e sempre que é necessário efetuar uma alteração em um diálogo de negócio, é obrigatório adaptar as UADs e os sistemas em função dos dados pretendidos. Conclui-se que seria muito benéfico, a reutilização dos diálogos de negócio, pois poderia existir uma poupança de recursos, custos e tempo de implementação. Esta reutilização potência a realização da sua venda, pois permite que o cliente consiga ver mais rapidamente as suas necessidades satisfeitas.
2. **Análise de oportunidades:** ao analisar-se a oportunidade identificada e verificando as tendências tecnológicas atuais, percebe-se que ao usar-se os dispositivos móveis (*e.g. smartphone* ou *tablets*) como alternativa às UADs tradicionais, pode-se efetuar poupanças significativas de custos nos equipamentos. Os dispositivos móveis possuem características semelhantes às UADs utilizadas.
3. **Geração e aperfeiçoamento de ideias:** nesta fase começou-se a debater como é que poderíamos usar os dispositivos móveis como UADs e como fazer a reutilização dos diálogos de negócio. Chegou-se à conclusão que seria necessário desenvolver uma plataforma que possibilitasse a criação/esquematização de diálogos de negócio. Deste modo, a plataforma é a responsável pela configuração do processo/fluxo de aquisição de dados. Para ser possível a aquisição dos diálogos de negócio através do modelo definido na plataforma, é necessário criar uma ponte entre o processo de modelação e o processo de aquisição de dados, para isso definiu-se o conceito do interpretador. Um

interpretador seria o *software* responsável por interpretar e executar os diálogos de negócio nos dispositivos móveis.

4. **Seleção de ideias:** decidiu-se que a ideia da plataforma e do interpretador eram credíveis.
5. **Definição de conceito:** como saída para o processo de desenvolvimento definiu-se o conceito de criar uma plataforma de geração de diálogos de negócio. Esta plataforma deve permitir a criação de diálogo de negócio de forma gráfica. Estes diálogos posteriormente devem ser instanciados nas UADs (dispositivos móveis), através do interpretador, de forma que seja possível efetuar a aquisição dos dados pretendidos.

3.4.3 Proposta de Valor

Segundo Nicola (Nicola et al., 2012), o valor tem vindo a ser definido por diversas teorias dependendo do contexto como necessidade, interesse, padrão, crenças, atitudes e preferências.

O valor para o cliente, trata-se da relação que o valor do produto (bem ou serviço) possui, baseado na relação entre os benefícios e os custos percebidos para a sua aquisição. Se o cliente não perceber o valor do produto não vai adquiri-lo. O cliente deve perceber o valor do produto, este conhecimento conduz-nos à definição de valor percebido.

No contexto do marketing, existe várias definições de valor percebido. Segundo Zeithaml (Zeithaml, 1988), o valor percebido por parte dos clientes é a conexão para a sua satisfação. Já Kotler (Kotler, 1998) definiu que o valor percebido assinala que uma oferta de *marketing* tem valor associado entre o fornecedor e o cliente, resultando da diferença entre o valor total esperado (benefícios) e o custo (sacrifícios) para o cliente. Desta forma, pode-se definir o valor percebido, como a percepção por parte do cliente, de quais os benefícios que retira de um determinado produto (bem ou serviço), em relação aos sacrifícios, como o valor que paga por ele. O cliente considera o valor percebido como uma vantagem competitiva.

Através de uma perspetiva longitudinal de valor é possível compreender o valor percebido que um cliente pode ter de um produto, tendo em conta uma linha temporal. Desta forma, é possível dividir a linha temporal nas seguintes quatro fases distintas (Nicola, 2016):

1. **Pré-compra:** o objetivo é prever como possíveis clientes percebem o produto (bem ou serviço);
2. **Compra:** o valor percebido para o cliente no momento da compra;
3. **Pós-compra:** obtenção de resultados baseados em experiências de clientes ou fornecedores;
4. **Depois da utilização:** fase que reflete o ponto de eliminação da venda.

Uma proposta de valor tem como objetivo explicar como um conjunto de produtos (bem ou serviços), podem criar valor para um segmento específico de clientes (OSTERWALDER and PIGNEUR, 2010). Esta análise é importante, uma vez que, possibilita demonstrar como um produto (bem ou serviço) se pode diferenciar da concorrência. Para definir-se uma proposta de

valor de um produto (bem ou serviço), é necessário perceber quais são as expectativas do cliente, através dos benefícios/sacrifícios que o produto oferece.

Como proposta de valor para este trabalho, pretende-se apresentar uma plataforma que permita mudar o paradigma dos sistemas atuais nas áreas de negócio apresentadas. A utilização desta plataforma facilita a definição de processos de aquisição mais rapidamente, simplificando o processo de modulação e de implementação. Onde, pessoas sem quaisquer conhecimentos na área de engenharia de *software*, mas com conhecimentos da área de negócio, conseguem modelar processos de aquisição de dados. Esta simplicidade aliada à rapidez na modulação dos diálogos possibilita a redução de custos, tempo e recursos na implementação de um novo projeto/sistema ou, na adaptação de um existente. Ainda, permitira desenvolver uma adaptabilidade/compatibilização, que atualmente não existe, entre os sistemas de gestão e as diversas UADs existentes no mercado.

Analisando a solução proposta numa perspetiva longitudinal de valor percebemos, que os clientes vão compreender que este conceito satisfaz mais rapidamente as suas necessidades, graças à sua simplicidade e adaptabilidade. Esta compreensão pode acontecer no momento da compra, onde, é possível apresentar aos clientes os processos de aquisição pretendidos em funcionamento e adapta-los às suas exigências em tempo real. Desta forma, os clientes percebem que após a compra, se pretenderem efetuar alterações nos seus processos de aquisição, é possível através deste conceito que potencia a reutilização/adaptação de diálogos de negócio. Esta rapidez permite que as adaptações após a compra sejam realizadas com um custo inferior ao das soluções atuais.

Este novo sistema destina-se aos seguintes dois segmentos de clientes:

1. **Empresas de *software*** que já possuem sistemas nestas áreas de negócio e que pretendam migrar os seus sistemas, para a metodologia que estamos a propor.
2. **Empresas ou pessoas** que pretendam a aquisição de dados dos seus processos de negócio. Este segmento de clientes tem a necessidade de obter dados dos seus processos de formas mais simples e organizadas. Através desta plataforma vão conseguir definir os diálogos de negócio em questão, que posteriormente serão instanciados nos seus dispositivos móveis (funcionando como UADs), para recolher os dados. Os dados adquiridos serão processados pelas aplicações de gestão existentes nestes clientes.

Relativamente aos sacrifícios, as empresas de *software* têm de se adaptarem a este conceito. Para esta adaptação será necessário que primeiro os seus colaboradores despendam tempo com formação, com o objetivo de compreenderem como configurar diálogos de negócio através da plataforma. Segundo, existe a obrigação de se efetuar uma compatibilização entre os seus sistemas de gestão e as suas UADs, implementando o conceito do interpretador.

Já as empresas ou pessoas que pretendam a aquisição de dados dos seus processos de negócio, como sacrificio apenas terão que compreender como configurar diálogos de negócio. Uma vez

que, usaram os seus dispositivos moveis como UAD através da utilização do nosso interpretador para estes equipamentos.

3.4.4 Modelo de CANVAS

Um negócio para ter sucesso, é necessário compreender e visualizar o seu modelo de negócio. Um método simples e fácil de entender a influência que determinadas áreas possuem sobre o negócio é definir o modelo de CANVAS. Segundo Osterwalder (OSTERWALDER and PIGNEUR, 2010), o modelo de CANVAS é uma ferramenta de gestão estratégica para as organizações. Permitindo desenvolver novos modelos de negócios ou descrever os existentes. O seu objetivo é a construção de um mapa visual pré-formatado com nove grupos fundamentais para o negócio.

Segundo Osterwalder (OSTERWALDER and PIGNEUR, 2010), o modelo de CANVAS divide-se nos seguintes grupos:

- **Customer Segments (Segmentos de Clientes):** grupo ou grupos de clientes alvo para quem se pretende criar valor.
- **Value Proposition (Proposta de Valor):** representa o valor que o produto (bem ou serviço) vai oferecer aos clientes alvo. Explicando quais os problemas que vão ser resolvidos com base nas necessidades dos clientes alvo.
- **Channels (Canais):** como é que os clientes vão conhecer o valor do produto (bem ou serviço). Define os métodos de comunicação que a organização usa para entregar valor aos clientes.
- **Customer Relationships (Relacionamento com o Cliente):** forma como a empresa se vai posicionar para relacionar com os clientes.
- **Revenue Streams (Fontes de Receita):** é como a organização vai obter receitas através da proposta de valor.
- **Key Resources (Recurso-Chave):** trata-se dos ativos necessários para que o modelo de negócio funcionar.
- **Key Activities (Atividades-Chave):** são as atividades fundamentais que se deve realizar de modo que o modelo de negócio funcione.
- **Key Partners (Parcerias-Chave):** quais os fornecedores e parceiros essenciais ao funcionamento do modelo de negócio;
- **Cost Structure (Estrutura de Custos):** descreve quais os principais custos necessários para na operação do modelo de negócio.

Na Tabela 10, encontra-se definido o modelo de negócio de CANVAS, é importante realçar, que as empresas de *software* encontram-se nos seguintes grupos:

- **Parcerias-chave:** pela possibilidade de comercializarem a nossa solução (revendedores).
- **Segmento de clientes:** pela possibilidade da migração dos seus sistemas atuais ou a modelação de novos sistemas utilizando o conceito apresentado.

Tabela 10 - Modelo de Negócio de CANVAS

Parcerias-Chave	Atividades-Chave	Proposta de Valor	Relacionamento com o Cliente	Segmentos de Clientes
Empresas de <i>software</i> (<i>revendedores</i>).	Desenvolvimento da plataforma.	Redução de custos.	Acompanhamento e apoio na modelação de diálogos de negócio.	Empresas de <i>software</i> .
Fornecedores de UADs.	Desenvolvimento de interpretadores.	Redução de tempo.	Manuais de utilização.	Empresas ou pessoas que pretendam a aquisição de dados dos seus processos de negócio.
	Definição de diálogos de negócio usando a plataforma.	Redução de recursos.		
	Marketing.	Simplificação.		
	Recurso-Chave	Maior adaptabilidade.	Canais	
	Engenheiros de <i>software</i> . Servidores.		Reuniões. Canal no YOUTUBE. Email.	
Estrutura de Custos			Fontes de Receita	
Custos de implementação do <i>software</i> . Aluguer de servidores. Marketing.			Modelação de diálogos de negócio Formação/Suporte. Licença de utilização. Adaptação de sistemas existentes.	

3.5 Visão da Solução

Com base na análise efetuada nas secções anteriores, o objetivo desta subsecção é apresentar uma visão de solução para o problema descrito, utilizando as abordagens e conceitos identificados anteriormente.

Através da análise do problema é possível definir uma série de alterações à arquitetura utilizada nos sistemas atuais, para que seja possível eliminar a alta dependência entre as UADs e os sistemas de gestão. O objetivo destas alterações é permitir a reutilização dos diálogos de negócio em UADs distintas (e.g. telemóveis, tablets). Entende-se ser necessário dividir estes sistemas em três processos distintos (cf. Figura 20):

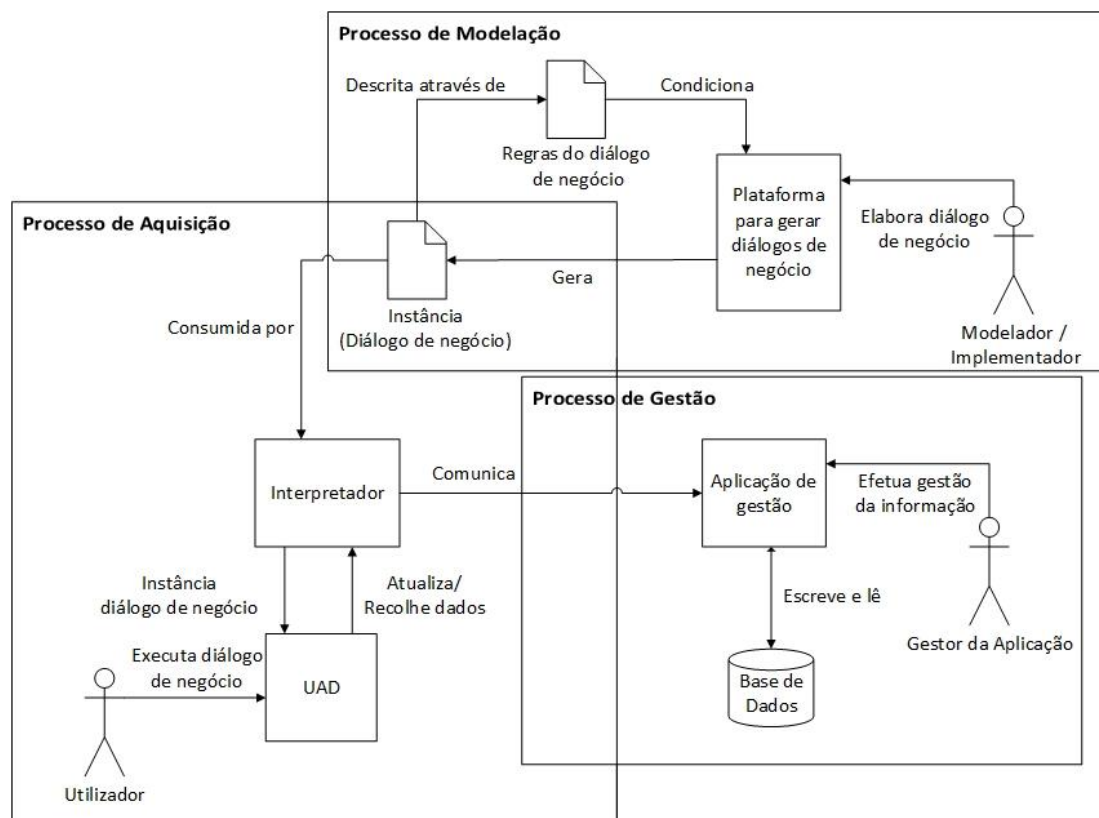


Figura 20 – Visão da Solução Pretendida

- 1. Processo de modelação:** este processo é o responsável por permitir de forma simples e gráfica, criar ou alterar, através de uma configuração, diálogos de negócio. Por outras palavras, pode-se afirmar que é o responsável por permitir a especificação dos passos necessários ao processo de aquisição.
- 2. Processo de aquisição:** responsável pela execução dos diálogos de negócio pelas UADs. O seu objetivo é que consiga interpretar o fluxo de aquisição descrito no diálogo de

negócio, de forma que, possibilite a recolha dos dados desejados. Este processo deve proporcionar uma simples interação homem-máquina.

3. **Processo de gestão de informação:** este processo trata-se das aplicações de gestão existentes.

Com a divisão dos sistemas em três processos distintos, é possível definir responsabilidades para cada uma das deles e apresenta-las da seguinte forma:

- **Processo de Modelação**
 - **Regras do diálogo de negócio:** define os tipos de regras possíveis do fluxo de aquisição (e.g. tarefas de aquisição, dados de entrada, dados de saída, tipos de validações, quais as decisões, como guardar os dados, como comunicar com o processo de gestão, entre outros).
 - **Plataforma para modelar/gerar diálogo de negócio:** esta plataforma através de uma notação de modelação de processos de negócio deve permitir de modo simples, a definição de um modelo de aquisição necessário para a recolha dos dados pretendidos. Desta forma, esta plataforma é responsável por gerar uma instância de um diálogo de negócio que representa um fluxo de aquisição.
 - **Instância de um diálogo de negócio:** também pode ser denominado como diálogo de negócio. Define como deve ser realizado o fluxo de aquisição de dados. Este documento/artefacto descreve todas as etapas do diálogo de negócio. Criado pela plataforma de modelação, com o objetivo de posteriormente ser consumida pelo interpretador, no processo de aquisição.
- **Processo de Aquisição**
 - **Interpretador:** este componente consome o diálogo de negócio, gerado pelo processo de modelação. Desta forma, o interpretador compreende como instanciar na UAD os passos necessários para a aquisição dos dados. O interpretador deve possuir o conhecimento necessário para interagir com uma determinada UAD. Por outras palavras, pode-se afirmar que o interpretador vai interpretar o fluxo de aquisição definido no diálogo e transforma-lo num processo de recolha de dados estruturado para uma UAD.
 - **UAD:** este componente trata-se dos dispositivos móveis (e.g. tablet, smart phone, etc.).

- **Processo de Gestão de Informação**
 - **Aplicações de Gestão:** estes são os sistemas existentes que efetuam a gestão da informação necessária para cada área.

É importante realçar, que o uso desta solução permite recolher os dados pretendidos quase de imediato. Para se perceber melhor esta afirmação, observe-se o seguinte exemplo: existe a necessidade de se alterar um diálogo para que recolha mais informações. É possível alterar o diálogo rapidamente através da plataforma, para que no momento da aquisição, seja possível recolher e guardar os dados desejados. O processamento dos dados, realizado pelo processo de gestão, pode ser adaptado/reimplementado posteriormente mas, os dados são recolhidos de imediato após a alteração do diálogo.

Os benefícios que se pretende alcançar com desenvolvimento desta solução são:

- **Redução de custos:** como o processo de modelação passa a ser por configuração declarativa e não por implementação ou reimplementação imperativa, deixa de ser obrigatório que tenha que ser realizado por profissionais qualificados e com o conhecimento no sistema em questão. Esta proposta de solução aumenta a simplicidade graças aos conceitos do processo de modelação.
- **Simplificação:** possibilita que pessoas com conhecimento da área de negócio em questão consigam modelar os diálogos desejados. Estas pessoas devem ter conhecimentos em informática apenas na ótica de utilizador.
- **Redução de tempo:** a plataforma permite agilizar o processo de modelação, reduzindo substancialmente o tempo de criação ou adaptação de um novo diálogo de negócio.
- **Redução de recursos:** permite a redução de recursos humanos (profissionais qualificados).
- **Maior adaptabilidade:** com a utilização do conceito interpretador, é possível utilizar o mesmo diálogo de negócio em diferentes UADs. Deste modo, comprova-se que este novo conceito permite implementar uma maior adaptabilidade do que a existente nos sistemas atuais.

4 Análise e Requisitos

Este capítulo visa analisar o problema e conseqüentemente enunciar sob a forma de requisitos o que se pretende alcançar tendo em consideração a visão da solução apresentada na subsecção 3.5. Esta visão apresenta o conceito de divisão dos sistemas atuais em três partes distintas: 1) Processo de modelação, 2) Processo de aquisição e 3) Processo de gestão de informação.

Tendo em consideração que o processo de gestão de informação se trata essencialmente dos sistemas de gestão de informação (*backoffice*) atuais (i.e. já existentes), estes não foram analisados sob o ponto de vista de requisitos. Assim, relativamente a estes apenas é abordado como é que se interligam com o processo de aquisição.

Para uma maior facilidade de compressão, esta secção encontra-se subdividida nas seguintes subsecções:

1. Processo de Modelação.
2. Processo de Aquisição.

4.1 Processo de Modelação

Com base na visão da solução realizada através da identificação dos problemas dos sistemas atuais, o processo de modelação deve permitir modelar, de forma rápida e simples, diálogos de negócio. Para se alcançar este objetivo, é necessário criar uma plataforma que permita modelar diálogos de negócio.

O processo de modelação consiste numa plataforma que permite modelar diálogos de negócio, com base num conjunto de regras que limitam o processo de modelação. Nas próximas subsecções é analisado em detalhe o processo de modelação.

4.1.1 Análise

A caracterização efetuada na secção 3 permite perceber que um diálogo de negócio obedece a um conjunto de regras. Desta forma, a plataforma de modelação deve possibilitar modelar diálogos de negócio, com base em regras que permitem descrever fluxos de aquisição de dados. Com base neste requisito, é necessário adotar uma notação (ou vocabulário ou esquema) capaz de capturar/represente a lógica (ou regras) de aquisição de dados, nomeadamente do fluxo que essa aquisição implica. **Um fluxo de aquisição de dados** representa a sequência de tarefas que permitem a recolha dos dados desejados. Este fluxo é composto por elementos que realizam três categorias distintas de tarefas:

- **Tarefas de aquisição:** permitem descrever quais e como os dados são recolhidos.
- **Tarefas de validação:** permitem validar os dados no momento da sua recolha. Desta forma, garante-se a qualidade dos dados recolhidos.
- **Tarefas de decisão:** têm a responsabilidade de escolher, com base nos dados recolhidos, qual a próxima tarefa do fluxo de aquisição a ser executada.

Para um melhor entendimento do fluxo de aquisição, de seguida é apresentado cada um dos três conceitos e a sua respetiva enumeração através das suas regras de aquisição.

As **tarefas de aquisição** são responsáveis por descrever quais os dados que devem ser recolhidos. Encontram-se divididas nos seguintes três tipos:

- **Questões ao utilizador:** permite descrever quais são os dados que o utilizador deve introduzir no momento da sua execução. A mesma tarefa pode conter uma ou mais questões ao utilizador.
- **Questões ao sistema:** permite descrever quais as questões que são efetuadas à UAD no momento de execução. (e.g. data atual, data e hora atual, identificação da UAD, etc.);
- **Respostas estáticas:** respostas pré-configuradas no momento de modelação do diálogo de negócio. Como exemplo para este tipo de tarefas, imagine-se uma UAD que regista os dados de acesso a uma determinada zona onde, o registo dessa entrada guarda um valor pré-configurado que identifica essa zona.

Uma **questão** tem associada como seu objetivo uma **resposta**. Uma resposta pode ser dos seguintes dois tipos:

- **Introdução:** permite definir o tipo de dados que se pretende (e.g. número inteiro, número decimal, data, data e hora, verdadeiro ou falso, entre outros). Este tipo de respostas pode ser realizada através de um método de introdução existente na UAD (e.g. teclado, leitor de código de barras, leitor RFID, etc.);
- **Escolha:** a resposta é selecionada de uma lista de possíveis respostas. A lista de respostas pode ser pré-configurada, através de uma lista estática ou, solicitada através de um pedido ao sistema de gestão.

A recolha de uma resposta com os dados desejados pode implicar que seja necessário configurar **validações** às respostas. Estas validações devem ser realizadas localmente na UAD (validações *offline*) ou através de uma solicitação ao sistema de gestão (*online*). Pode-se desta forma, dividir as validações nos seguintes dois tipos:

- **Validação local:** executa na UAD, com a possibilidade de ser realizada através dos seguintes métodos:
 - **Condicional:** permite definir restrições nos dados recolhidos dependendo do tipo de dados esperado da resposta: (e.g. valor igual, valor maior ou igual, valor maior, valor menor ou igual, valor menor, valor num intervalo, entre outros);
 - **Expressão regular:** descreve uma restrição sobre o tipo de dados que se pretende adquirir através de uma expressão regular;
 - **Execução de um script:** execução de uma função em script que permite validar o valor recolhido.
- **Validação remota:** possibilita solicitar remotamente uma validação aos dados recolhidos. Utiliza um protocolo de comunicação para delegar a validação ao processo de gestão.

Uma decisão permite definir o rumo que o fluxo de aquisição deve tomar em determinado estado, conforme os dados recolhidos. Através das decisões deve ser possível definir ciclos de aquisição. Para cumprir este objetivo, uma decisão tem uma relação com uma ou mais condições. **Uma condição** permitir descrever um conjunto de verificações sobre os dados recolhidos. Caso as verificações sejam satisfeitas, o fluxo de aquisição deve seguir com a sequência definida na condição. Ainda, deve ser possível definir qual é a condição por omissão, desta forma, é possível configurar qual o fluxo a seguir, quando não é possível satisfazer nenhuma condição.

O resultado de cada execução de um diálogo de negócio (i.e. após a aquisição dos dados) precisa de ser armazenado de forma persistente. Este objetivo pode ser conseguido através de dois tipos de **armazenamento de dados** distintos:

- **Armazenamento local:** permite armazenar os dados recolhidos localmente na UAD.
- **Armazenamento remoto:** permite descrever uma solicitação/pedido ao sistema de gestão, para lhe transmitir a responsabilidade de armazenar os dados recolhidos. Em caso de falha, a informação deve ser armazenada localmente e enviado posteriormente.

A interação com os sistemas de gestão atuais (processo de gestão) é realizada através de **pedidos**. Um dos seus objetivos pode ser validar a qualidade dos dados recolhidos, solicitar dados necessários ao sistema de gestão (e.g. itens para a resposta do tipo escolha) ou transmitir a responsabilidade de armazenar os dados.

Por fim, para potencializar a reutilização de diálogos de negócio permite-se que um diálogo possa conter/remeter para um outro diálogo (subdiálogo).

Para que se consiga perceber melhor as regras do diálogo de negócio e os seus relacionamentos, a Figura 21 apresenta um modelo de domínio que representa estas regras.

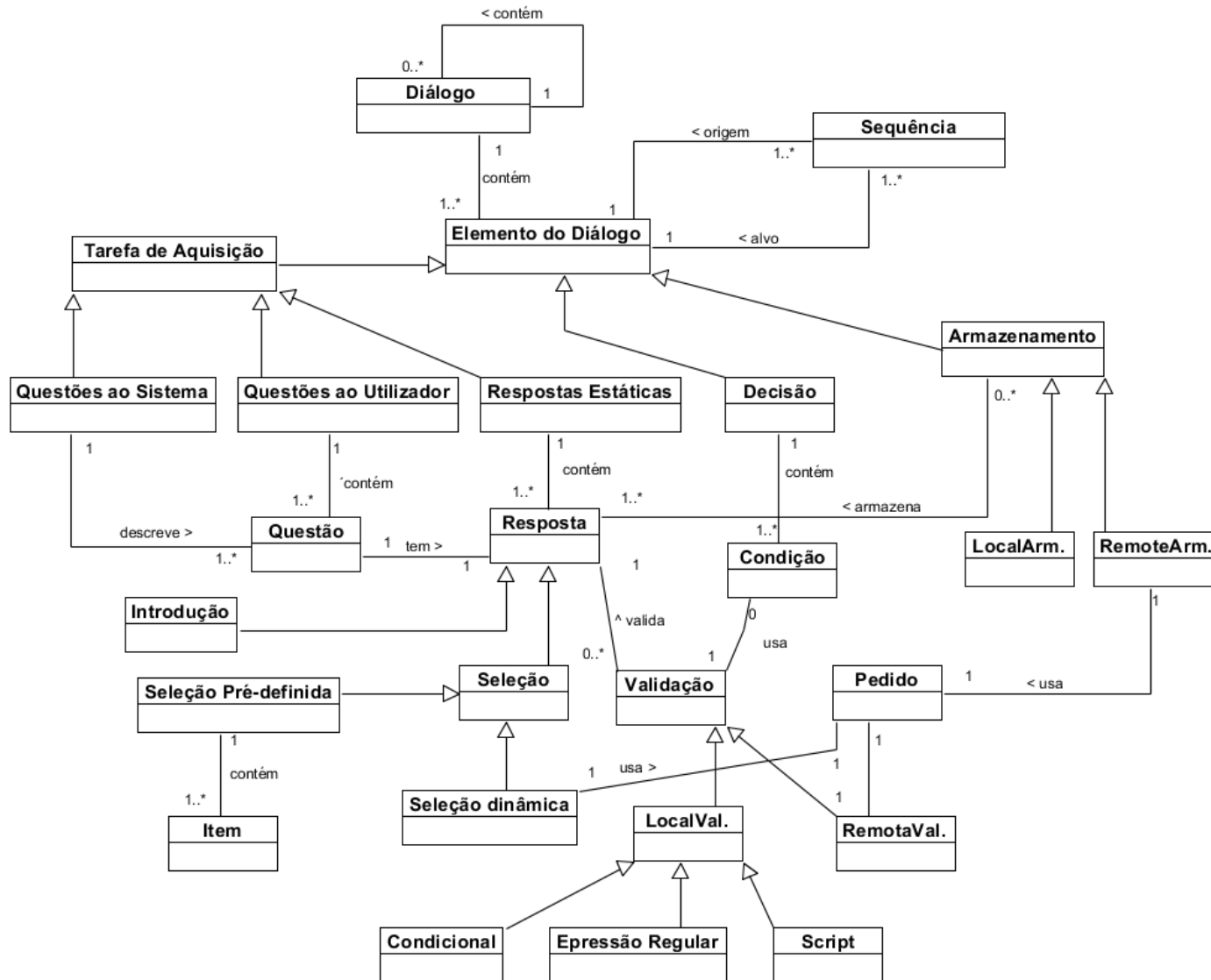


Figura 21 - Modelo de Dómino: Regras do Diálogo de Negócio

4.1.2 Requisitos

O ator no processo de aquisição trata-se de um modelador. Este modelador possui o conhecimento da área de negócio em questão, de forma que consiga modelar os diálogos de negócio desejados. Por norma, o modelador apenas possui conhecimentos de informática na ótica de utilizador não possuindo qualquer conhecimento de desenvolvimento de aplicações.

4.1.2.1 Requisitos Funcionais

Esta plataforma deve possibilitar as seguintes funcionalidades essenciais:

- Criar, alterar e visualizar diálogos de negócio: o processo de modelação deve, de forma simples, permitir modelar diálogos de negócio através de uma notação de modelação de processos de negócio. Este processo de modelação deve seguir um conjunto de regras, designadas de regras do diálogo de negócio.
- Importar e Exportar diálogo de negócio: a plataforma de modelação deve permitir exportar e importar diálogos de negócio. Através da importação de, a disponibilização de alterações de um diálogo de negócio serão realizadas mais depressa, pois não existe a necessidade de se criar de novo o diálogo.

Estas funcionalidades genéricas são representadas (e até detalhadas) através de um diagrama casos de uso na Figura 22.

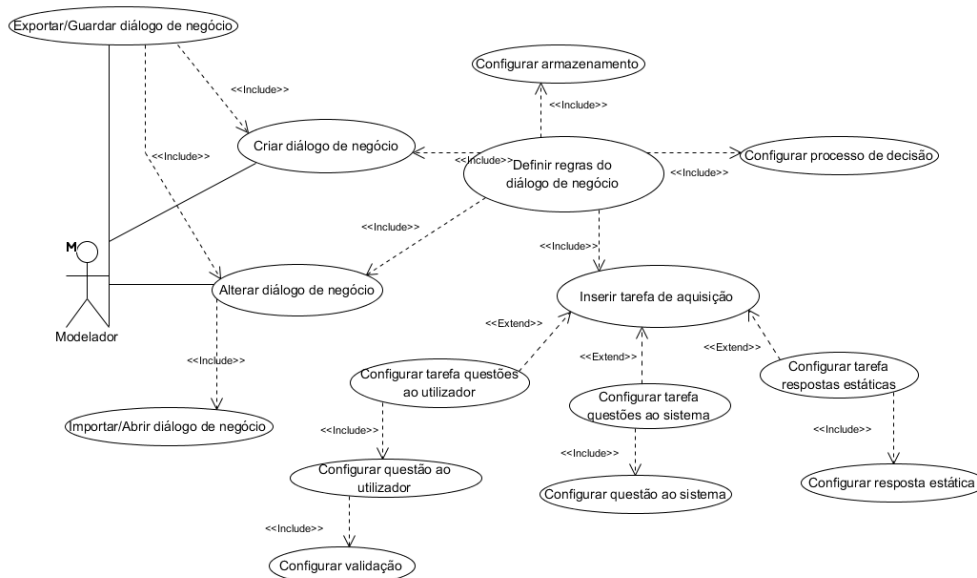


Figura 22 – Diagrama de Casos de Uso da Plataforma de Modelação

De seguida, os casos de uso mais relevantes são descritos através de tabelas onde consta o nome do caso de uso, o ator do mesmo, as partes interessadas, as pré e pós condições para a realização do caso de uso bem como o seu fluxo principal e alguns fluxos alternativos (Tabela 11 a 19) .

Tabela 11 – Caso de uso “Criar Diálogo de Negócio”

Caso de Uso	Criar Diálogo de Negócio
Ator	Utilizador
Partes Interessadas	Utilizador: criação do diálogo de negócio de forma simples e prática.
Pré Condições	
Pós Condições	
Fluxo Principal (sucesso)	1- O utilizador escolhe criar um novo diálogo de negócio. 2- A plataforma de modelação apresenta o ambiente de modelação. 3- O utilizador descreve o fluxo de aquisição. 4- O Utilizador guarda o diálogo de negócio. 5- A plataforma de modelação válida o fluxo de aquisição e guarda o diálogo de negócio no formato XML.
Extensões (fluxos alternativos)	5- Fluxo de aquisição incompleto. 1. A plataforma informa quais os dados em falta. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 12 – Caso de Uso “Alterar Diálogo de Negócio”

Caso de Uso	Alterar Diálogo de Negócio
Ator	Utilizador
Partes Interessadas	Utilizador: alteração de um diálogo de negócio existente de forma simples e prática.
Pré Condições	Existir um diálogo de negócio já modelado.
Pós Condições	
Fluxo Principal (sucesso)	1- O utilizador escolhe qual o diálogo de negócio que pretende alterar. 2- A plataforma de modelação apresenta o ambiente de modelação com a visualização do diálogo de negócio escolhido. 3- O utilizador descreve novos fluxos de aquisição. 4- O Utilizador guarda o diálogo de negócio. 5- A plataforma de modelação válida o fluxo de aquisição e guarda o diálogo de negócio no formato XML.
Extensões (fluxos alternativos)	5- Fluxo de aquisição incompleto. 1. A plataforma informa quais os dados em falta. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 13 – Caso de Uso “Inserir Tarefa de Aquisição”

Caso de Uso	Inserir Tarefa de Aquisição
Ator	Utilizador
Partes Interessadas	Utilizador: criar nova tarefa de aquisição.
Pré Condições	Criar ou alterar um diálogo de negócio.
Pós Condições	O fluxo de aquisição do diálogo de negócio possui uma nova tarefa de aquisição.
Fluxo Principal (sucesso)	<p>1- O utilizador escolhe qual o tipo de tarefa de aquisição pretende criar (questões ao utilizador, questões ao sistema ou respostas estáticas) e conecta-o ao fluxo de aquisição.</p> <p>2- A plataforma de modelação apresenta a tarefa escolhida.</p> <p>3- O utilizador configura a tarefa escolhida.</p> <p>4- A plataforma de modelação válida a configuração do utilizador.</p>
Extensões (fluxos alternativos)	<p>4- Fluxo de aquisição incompleto.</p> <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 14 – Caso de Uso “Questão ao utilizador”

Caso de Uso	Configurar Questão ao Utilizador
Ator	Utilizador
Partes Interessadas	Utilizador: definir nova questão ao utilizador.
Pré Condições	Criar ou alterar um diálogo de negócio. Inserir uma tarefa de aquisição do tipo questões ao utilizador.
Pós Condições	A tarefa questões ao utilizador possui a informação de uma nova questão.
Fluxo Principal (sucesso)	<p>1- O utilizador seleciona qual a tarefa do tipo questões ao utilizador que pretende alterar.</p> <p>2- A plataforma de modelação apresenta a configuração dessa tarefa.</p> <p>3- O utilizador escolhe inserir uma questão.</p> <p>4- A plataforma apresenta o formulário para se configurar a questão.</p> <p>5- O utilizador preenche o formulário apresentado.</p> <p>6- A plataforma válida se os dados estão corretamente preenchido.</p>
Extensões (fluxos alternativos)	<p>6- Dados incorretos ou insuficientes.</p> <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 15 – Caso de Uso “Inserir Questão ao Sistema”

Caso de Uso	Configurar Questão ao Sistema
Ator	Utilizador
Partes Interessadas	Utilizador: definir nova questão ao sistema.
Pré Condições	Criar ou alterar um diálogo de negócio. Inserir uma tarefa de aquisição do tipo questões ao sistema.
Pós Condições	A tarefa questões ao sistema possui a informação de uma nova questão.
Fluxo Principal (sucesso)	<p>1- O utilizador escolhe qual a tarefa do tipo questões ao sistema que pretende alterar.</p> <p>2- A plataforma de modelação apresenta a configuração dessa tarefa.</p> <p>3- O utilizador escolhe inserir uma nova questão.</p> <p>4- A plataforma apresenta o formulário para se configurar a questão.</p> <p>5- O utilizador seleciona qual a questão que pretende fazer ao sistema preenchendo o formulário.</p> <p>6- A plataforma válida se os dados estão corretamente preenchido.</p>
Extensões (fluxos alternativos)	<p>6- Dados incorretos ou insuficientes.</p> <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 16 – Caso de Uso “Definir Resposta ao Estática (pré-definida)”

Caso de Uso	Configurar Resposta ao Estática (pré-definida)
Ator	Utilizador
Partes Interessadas	Utilizador: definir uma nova resposta estática.
Pré Condições	Criar ou alterar um diálogo de negócio. Inserir uma tarefa de aquisição do tipo respostas estáticas.
Pós Condições	A tarefa respostas estáticas possui a informação de uma nova resposta pré-definida.
Fluxo Principal (sucesso)	<p>1- O utilizador escolhe qual a tarefa do tipo questões ao sistema que pretende alterar.</p> <p>2- A plataforma de modelação apresenta a configuração dessa tarefa.</p> <p>3- O utilizador escolhe inserir uma nova resposta.</p> <p>4- A plataforma apresenta os campos necessários para se configurar a resposta.</p> <p>5- O utilizador configura os dados da resposta.</p> <p>6- A plataforma válida se os dados estão corretamente preenchido.</p>
Extensões (fluxos alternativos)	<p>6- Dados incorretos ou insuficientes.</p> <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 17 – Caso de Uso “Configurar validação de sobre os dados recolhidos”

Caso de Uso	Configurar validação de sobre os dados recolhidos
Ator	Utilizador
Partes Interessadas	Utilizador: restringir o formato dos dados que pretende obter.
Pré Condições	Criar ou alterar um diálogo de negócio. Inserir uma tarefa de aquisição do tipo questões ao utilizador. Inserir uma nova questão ao utilizador.
Pós Condições	O diálogo possui o conhecimento necessário para efetuar uma validação sobre os dados recolhidos.
Fluxo Principal (sucesso)	<ol style="list-style-type: none"> 1- O utilizador escolhe qual a tarefa do tipo questões ao utilizador que pretende alterar. 2- A plataforma de modelação apresenta a lista de questões dessa tarefa. 3- O utilizador seleciona qual a questão que pretende configurar uma validação. 4- A plataforma apresenta o formulário de validação. 5- O utilizador configura os dados referentes à validação. 6- A plataforma válida se os dados estão corretamente preenchido.
Extensões (fluxos alternativos)	<ol style="list-style-type: none"> 6- Dados incorretos ou insuficientes. <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 18 – Caso de Uso “Configurar processo decisão”

Caso de Uso	Configurar processo decisão
Ator	Utilizador
Partes Interessadas	Utilizador: definir processo de decisão.
Pré Condições	Criar ou alterar um diálogo de negócio. Inserir uma tarefa de aquisição.
Pós Condições	O diálogo possui as regras necessárias para poder decidir qual o próximo fluxo de aquisição que deve tomar.
Fluxo Principal (sucesso)	<ol style="list-style-type: none"> 1- O utilizador escolhe introduzir um novo processo de decisão. 2- A plataforma de modelação graficamente apresenta os parâmetros necessários para a sua configuração da decisão. 3- O utilizador define qual é o fluxo por defeito. 4- O utilizador define uma ou mais condições e qual o fluxo que deve ser tomado se estas condições forem satisfeitas. 5- A plataforma válida se os dados estão corretamente preenchido.
Extensões (fluxos alternativos)	<ol style="list-style-type: none"> 5- Dados incorretos ou insuficientes. <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

Tabela 19 – Caso de Uso “Configurar armazenamento dos dados recolhidos”

Caso de Uso	Configurar armazenamento dos dados recolhidos
Ator	Utilizador
Partes Interessadas	Utilizador: definir como guardar a informação recolhida.
Pré Condições	Criar ou alterar um diálogo de negócio. Inserir uma tarefa de aquisição.
Pós Condições	O diálogo possui o conhecimento de quando e como deve guardar os dados recolhidos.
Fluxo Principal (sucesso)	<ol style="list-style-type: none"> 1- O utilizador escolhe introduzir um novo repositório de dados e conecta-o ao fluxo de aquisição. 2- A plataforma de modelação graficamente apresenta o repositório e os parâmetros necessários para a sua configuração. 3- O utilizador define qual o tipo de repositório (local ou remoto). 4- A plataforma apresenta s formulário do preenchimento conforme o tipo de repositório escolhido. 5- O utilizador preenche o formulário. 6- A plataforma válida se os dados estão corretamente preenchido.
Extensões (fluxos alternativos)	<ol style="list-style-type: none"> 6- Dados incorretos ou insuficientes. <ol style="list-style-type: none"> 1. A plataforma informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

4.1.2.2 Requisitos Não Funcionais

Esta subsecção tem como objetivo apresentar os requisitos não funcionais que são relevantes para o sucesso e aceitação da plataforma de modelação. Desta forma, pode-se enumera-los da seguinte forma através do modelo FURPS+ (Grady, 1992):

- **Usabilidade:** o plataforma de modelação, deve de forma simples e intuitiva, possibilitar que os utilizadores consigam modelar os diálogos de negócio desejados Deste modo, os utilizadores deverão conseguir através de elementos gráficos, especificar quais os dados que devem ser recolhidos no processo de aquisição. A simplicidade do processo de modelação tem como principal objetivo a aceitação por parte dos utilizadores.
- **Fiabilidade/Confiabilidade:** a plataforma de modelação deve apresentar os elementos gráficos conforme a especificação das regras do diálogo de negócio. A importação de

um diálogo de negócio deve ser apresentada graficamente da mesma forma de como quando a sua exportação.

- **Desempenho:** a plataforma de modelação deve possibilitar importar, modelar e exportar diálogos de negócio imediatamente. Assim, o tempo de resposta deve ser instantâneo após uma solicitação por parte do modelador.
- **Suportabilidade:** a plataforma de modelação deve ser 100% compatível com a linguagem de marcação HTML. Assim, é possível ser usada em qualquer dispositivo através de um navegador web sem a necessidade de instalação. Deve permitir facilmente a sua extensão em caso de ser necessário permitir estender a notação de modelação.

4.2 Processo de Aquisição

Esta subsecção apresenta uma análise e requisitos do processo de aquisição. O processo de aquisição deve conhecer a notação de especificação de diálogos de negócio e, através desse conhecimento ser capaz de interpretar e executar em colaboração com uma UAD um qualquer diálogo de negócio especificado segundo a mesma notação.

4.2.1 Análise

O processo de aquisição refere-se ao momento em que o utilizador, interagindo com uma UAD, executa um diálogo de negócio. Para ser possível executar diálogos numa determinada UAD, é necessário desenvolver um interpretador dos diálogos de negócio para o tipo de UAD em questão. Para se perceber melhor o conceito de um interpretador e interação que deve ter com uma UAD, os próximos parágrafos apresentam uma análise que descreve as responsabilidades e a interação destes dois componentes.

O interpretador é um componente que permite converter um diálogo de negócio num fluxo de aquisição para uma determinada UAD. Por outras palavras, pode-se afirmar que o objetivo do interpretador é instanciar um diálogo de negócio na UAD pretendida. Através desta instanciação, o interpretador deve conduzir o fluxo de aquisição definido pelo diálogo de negócio, de modo que, seja possível recolher a informação desejada.

Pode-se dizer que o interpretador possui as seguintes responsabilidades:

- **Gerir o fluxo de aquisição:** deve gerir o fluxo de aquisição com base nas regras definidas no diálogo de negócio. Através destas regras, deve ser responsável por realizar as seguintes tarefas:

- **Realizar tarefas de aquisição:** deve apresentar as tarefas de aquisição pela sequência definida no diálogo de negócio na UAD;
 - **Validações:** executar validações locais e/ou, comunicar com o sistema de gestão para solicitar uma validação remota;
 - **Decisões:** executar decisões locais e/ou, comunicar com o sistema de gestão para solicitar uma decisão remota;
 - **Armazenar dados:** armazenar dados localmente e/ou, solicitar o armazenamento de dados ao sistema de gestão.
- **Especificação da interface de aquisição:** através de um diálogo de negócio, o interpretador deve construir dinamicamente uma interface gráfica, que possibilite uma interação com o utilizador. As características das UADs, definidas na subsecção 2.3, restringem como os dados podem ser recolhidos. Como exemplo de uma restrição, pode-se apresentar uma UAD que apenas possibilita a visualização de três linhas de texto (e.g. UAD *NuxFinger* da Innux Technologies, Lda, 2016e). No momento da criação da tarefa de aquisição, o interpretador deve ter em conta a característica do ecrã e possibilitar um diálogo com o utilizador usando apenas três linha do ecrã para apresentar os dados.
 - **Interação com a UAD:** o interpretador deve conseguir interagir com a UAD. O objetivo desta interação é que o interpretador deve conseguir orientar as tarefas de aquisição na UAD. Um exemplo desta orientação é que o interpretador como responsável por gerir o fluxo de aquisição deve, informar a UAD de quais as informações que deve apresentar em determinado estado do processo de aquisição.
 - **Comunicação com o sistema de gestão (*backoffice*):** conforme a especificação do diálogo de negócio, o interpretador deve conseguir comunicar com o sistema de gestão. Esta comunicação consiste nos seguintes objetivos: solicitar dados, validações ou decisões ao sistema de gestão.

A Figura 23 representa a instanciação do diálogo de negócio através do interpretador. Onde, o diálogo de negócio é representado semanticamente através da notação de especificação de diálogos. O diálogo de negócio é o requisito de entrada do interpretador, que vai interpretar o conteúdo deste diálogo, de forma que consiga instanciar o processo de aquisição definido no diálogo de negócio na UAD.

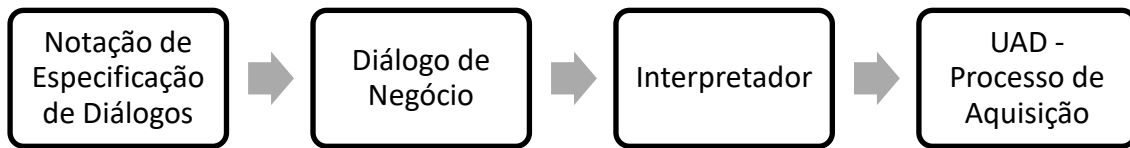


Figura 23 – Instanciação de um diálogo de negócio

É importante realçar que o interpretador tem a responsabilidade de através da do conteúdo do diálogo de negócio, conseguir comunicar corretamente com o sistema de gestão de informação. De forma, que consiga pedir os dados necessários, solicitar validações ou registar a informação recolhida num armazenamento remoto. Esta relação deve ser realizada através de um protocolo de comunicação.

4.2.2 Requisitos

Nesta subsecção encontra-se descritos os requisitos funcionais e não funcionais do processo de aquisição.

4.2.2.1 Requisitos Funcionais

A Figura 24 apresenta o diagrama de casos de uso do processo de aquisição. Onde, é possível verificar que apenas existe um ator (utilizador) e que a sua função é executar diálogos de negócio.



Figura 24 – Diagrama de Casos de Uso do “Processo de Aquisição”

A próxima tabela, Tabela 20, apresenta o caso de uso “Executar um diálogo de negócio”, no formato completo.

Tabela 20 – Caso de Uso “Executar diálogo de negócio”

Caso de Uso	Executar diálogo de negócio
Ator	Utilizador
Partes Interessadas	Utilizador: executar diálogo de negócio.
Pré Condições	Utilização de uma UAD
Pós Condições	São recolhidos os dados definidos no diálogo de negócio.
Fluxo Principal (sucesso)	<ol style="list-style-type: none"> 1- O utilizador escolhe executar o diálogo. 2- O UAD apresenta as questões do diálogo. 3- O utilizador responde às questões. 4- A UAD valida se os dados estão corretamente preenchido.
Extensões (fluxos alternativos)	<ol style="list-style-type: none"> 4- Dados incorretos ou insuficientes. <ol style="list-style-type: none"> 1. A UAD informa quais os dados incorretamente preenchido ou obrigatórios. 2. O utilizador preenche esses dados. 3. O fluxo principal é retomado.

4.2.2.2 Requisitos Não Funcionais

De seguida, nesta subsecção, são enumerados os requisitos não funcionais para o processo de aquisição através do modelo FURPS+.

- **Usabilidade:** a recolha de dados no processo de aquisição requer uma interação entre o utilizador e a UAD (homem-máquina). Esta interação deve ser o mais simples e intuitiva para uma fácil compreensão por parte do utilizador. No entanto, esta interação encontra-se limitada pelas características de cada UAD (e.g. tipo de ecrã, tipo de métodos de aquisição, etc.) assim, pode-se afirmar que o interface de aquisição deve ser adequado para cada UAD.
- **Fiabilidade/Confiabilidade:** o interpretador deve realizar o fluxo de aquisição com base nas regras definido no diálogo de negócio sem existir qualquer falha. Os dados recolhidos devem ser os esperados.
- **Desempenho:** a recolha dos dados pretendidos deve ser realizada o mais rápido possível. A comunicação entre a UAD e o sistema de gestão não devem demorar mais do que 5 segundos.
- **Suportabilidade:** o interpretador deve possibilitar que seja possível estender os seus processos de aquisição.

5 Design

Este capítulo, tem como o objetivo transformar a visão da solução, análise e requisitos numa proposta de *design*. Encontra-se dividido nas seguintes subsecções:

- **Visão geral:** apresenta uma visão geral do sistema onde, é descrito todos os componentes dos diferentes processos;
- **Notação de especificação de diálogos:** enuncia uma proposta de *design* para o componente notação.
- **Plataforma de modelação:** descreve com maior pormenor a proposta de *design* para a plataforma de modelação.
- **Interpretador:** encontra-se descrito uma proposta de *design* para o interpretador.

5.1 Visão Geral

Tendo em consideração a visão da solução (cf. secção 3.5), é apresentado a Figura 25, que representa a estrutura do sistema pretendido através de um diagrama de componentes. Nesta figura encontram-se identificados os componentes necessários e relações (comunicações) existentes entre os diferentes componentes.

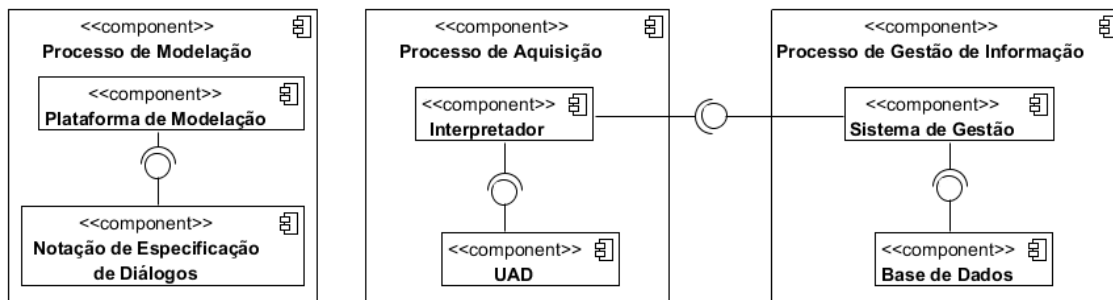


Figura 25 - Diagrama de Componentes Geral do Sistema

O processo de modelação tem como objetivo permitir criar e alterar diálogos de negócio segundo as regras do diálogo de negócio (notação). Enquanto, o processo de aquisição tem como objetivo permitir executar diálogos de negócio segundo as regras do diálogo de negócio (notação). Assim, não existe diretamente qualquer interação entre o processo de modelação e o processo de aquisição, ambos funcionam sem o conhecimento um do outro. No entanto, o resultado do processo de modelação (diálogo de negócio) é o ponto de partida do processo de aquisição e ambos estão dependentes da notação.

Por sua vez, o processo de aquisição necessita interagir com determinadas UADs. Esta interação consiste em realizar um conjunto de tarefas/funções necessárias na UAD (e.g. desenhar interface de aquisição, recolher dados, entre outras). O interpretador deve ainda, conseguir comunicar com o sistema de gestão referente ao processo de gestão de informação. O método/protocolo de comunicação é descrito no diálogo e definido no momento da modelação, para um determinado sistema de gestão (*backoffice*). No momento da execução o interpretador deve através da informação contida no diálogo, conseguir comunicar com o sistema de gestão desejado.

O processo de gestão de informação refere-se aos sistemas atuais. A compatibilização dos sistemas de gestão atuais, em alguns casos, com a proposta apresentada, pode ser realizada através da construção de uma nova camada. Esta camada será a responsável por permitir a interação entre o processo de aquisição e o processo de gestão de informação.

Como o processo de gestão de informação e as UADs já existem nos sistemas atuais, a perspetiva deste trabalho foca-se nos seguintes componentes:

1. Notação de Especificação de Diálogos
2. Plataforma de Modelação
3. Interpretador

As próximas subsecções abordam com mais detalhes estes três componentes, apresentando uma proposta de *design* para cada um.

5.2 Notação de Especificação de Diálogos

Como já foi analisado na subsecção 3.3, a notação de modelação escolhida para se representar as regras do diálogo de negócio é o BPMN (*Business Process Model and Notation*).

A escolha do BPMN para notação de modelação que descreve os diálogos de negócio, motiva que se compreenda como o BPMN pode representar os diálogos de negócio. É importante realçar que a estrutura do BPMN é definida através de um XSD (*XML Schema Definition*). Assim, a Figura 26 apresenta o diagrama de classes (adaptado) com base no XSD que descreve o BPMN onde, são apresentados os elementos que se assemelham às necessidades do diálogo de negócio.

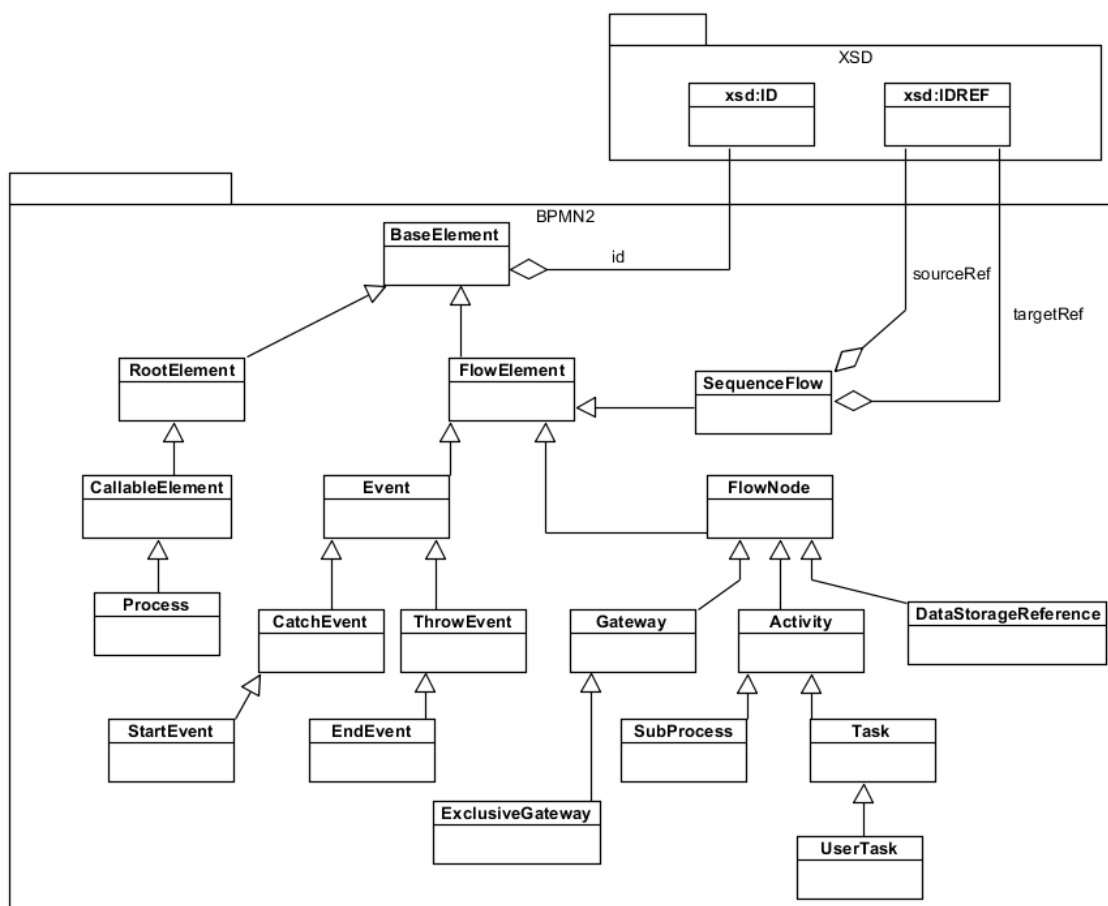


Figura 26 – Diagrama de Classes do BPMN (adaptado)

Como apresentado na Figura 26, o BPMN possui diversos elementos que permitem representar diferentes conceitos, nomeadamente tarefas (*Task*, *UserTask*), armazenamento de dados (*DataStorageReference*), ligação entre elementos (*SequenceFlow*), etc.

A Tabela 21 representa uma correlação possível da estrutura do BPMN às necessidades da representação de um diálogo de negócio, correspondendo os elementos do BPMN com a notação de especificação de diálogos.

Tabela 21 – Correlação entre o BPMN a Notação de Especificação de Diálogos

Elemento BPMN	Elemento da notação de especificação de diálogos	Descrição
Process	Diálogo	O elemento <i>Process</i> deve ser estendido para que seja possível representar o conceito do diálogo.
<i>Task</i>	Questões ao Sistema Respostas Estática	Pode-se estender esta regras usando o conceito tarefa (<i>Task</i>) já existente no BPMN.
<i>UserTask</i>	Questões ao Utilizador	Existindo já o conceito tarefa ao utilizador (<i>UserTask</i>) no BPMN, faz sentido estender a regra “Questões ao Utilizador” sobre esta tarefa.
<i>ExclusiveGateway</i>	Decisão	As tarefas de decisão podem ser representadas por este elemento BPMN.
<i>DataStorageReference</i>	Armazenamento	Este elemento do BPMN pode ser estendido para se poder representar o armazenamento.

Como é visível através da Tabela 21, os elementos da notação BPMN não representam diretamente os conceitos de modelação do diálogo de negócios. É necessário que seja possível representar todos elementos definidos nos requisitos descritos na subsecção 4.1.1. Desta forma, é necessário realizar uma extensão sobre o BPMN, para ser possível representar diálogos de negócio.

A Figura 27 através de um diagrama de classes representa a extensão que é necessária realizar ao BPMN, de forma que seja possível criar uma notação de especificação de diálogos. Nesta figura é perceptível quais são os elementos do BPMN que vão ser estendido através da utilização do conceito herança (e.g. Process, Task, User Task, Exclusive Gateway, Data Storage Reference). No entanto, é necessário criar outros elementos (cf. secção 4.1.1) para ser possível caracterizar diálogos de negócio.

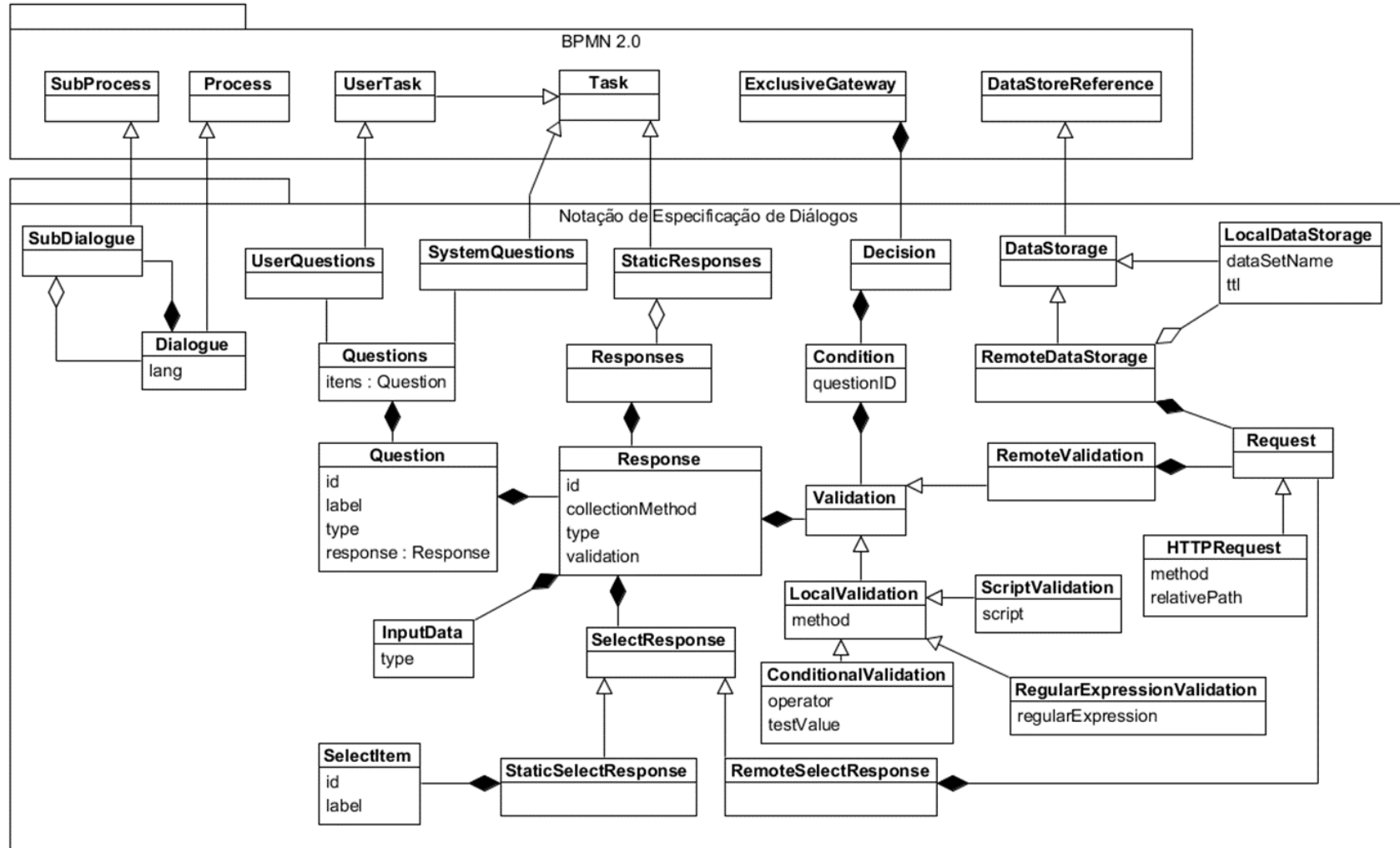


Figura 27 - Diagrama de Classes “Extensão do BPMN”

A utilização do BPMN como base da notação de especificação de diálogos restringe que se tenha que estender o BPMN, através de um ficheiro XSD (*XML Schema Definition*).

5.3 Plataforma de Modelação

Tendo em consideração o *design* do componente anterior e a existência de ferramentas *open-source* de modelação de processo em BPMN (cf. secção 3.3.1) considerou-se que não era uma opção adequada desenvolver o mesmo de raiz (i.e. do nada). Assim, escolheu-se a ferramenta BPMN.IO (“BPMN.IO,” 2016) como base de partida. É importante realçar que um dos critérios fundamentais é que a ferramenta deve ser desenvolvida para funcionar em navegadores web, sem o recurso à instalação de outras aplicações.

5.3.1 Ferramenta BPMN.IO

Para se perceber quais as alterações necessárias a realizar no BPMN.IO, de modo que seja possível adaptar esta ferramenta aos requisitos desejados, a Figura 28, apresenta um diagrama de componentes da ferramenta BPMN.IO.

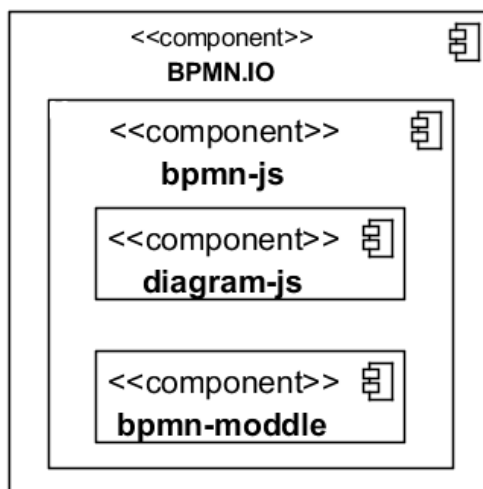


Figura 28 - Diagrama de Componentes do BPMN.IO

A seguir, para se perceber melhor o funcionamento do BPMN.IO é descrito individualmente cada componente e as suas responsabilidades.

O *bpmn-js* é descrito como uma biblioteca *JavaScript* que consegue apresentar a notação BPMN através de elementos gráficos nos *browsers*, sem a necessidade de existir um servidor web. É o componente principal do BPMN.IO, responsável por interagir e configurar os restantes componentes através do seu sistema por módulos que permite encapsular serviços. O objeto que deve ser instanciado para se iniciar o uso deste componente é o *BpmnModeler*.

BpmnModeler permite no momento da sua instanciação, através do construtor, definir quais os módulos ou serviços adicionais (extensões) que se pretende usar.

A Figura 29 representa a arquitetura do BPMN.IO descrevendo a relação entre os diferentes componentes e quais as suas responsabilidades. Nesta figura, é perceptível que o **bpmn-js** é responsável por ligar os componentes **diagram-js** e **bpmn-moddle**.

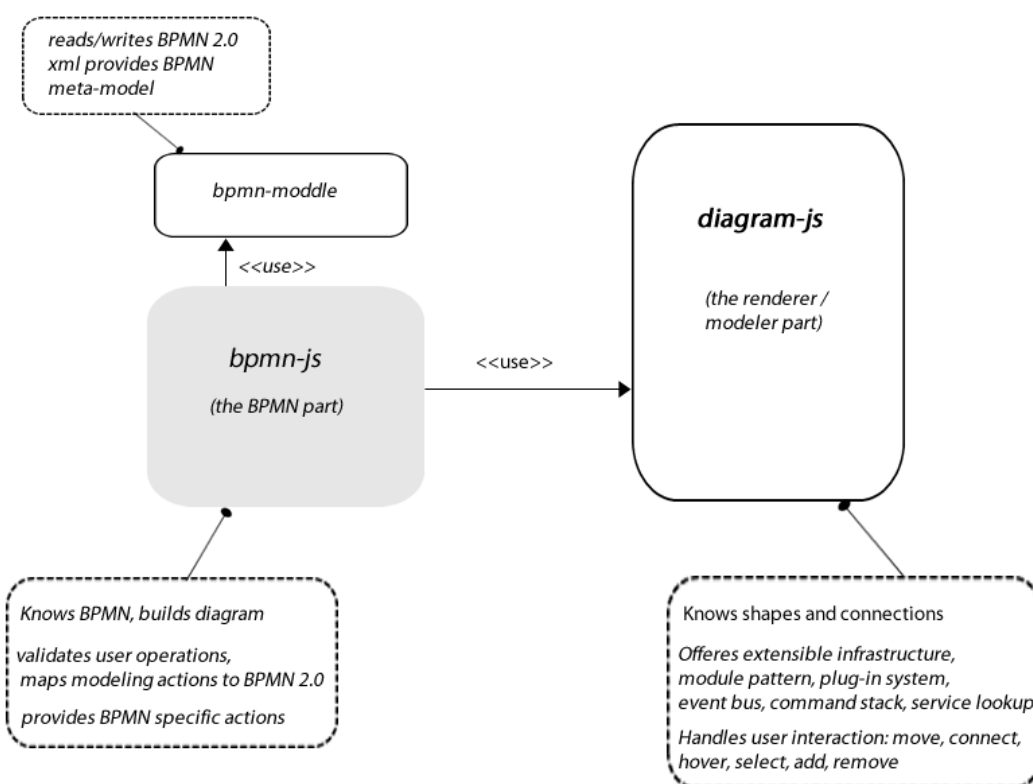


Figura 29 – Arquitetura do bpmn-js: Componentes e responsabilidades (adaptado (“BPMN.IO,” 2016))

O componente **diagram-js** (Camunda Services GmbH, 2017a) consiste num conjunto de bibliotecas *JavaScript* que possibilitam desenhar elementos e conexões, disponibilizando ainda métodos para se interagir com esses elementos gráficos. Pode-se afirmar que é uma caixa de ferramentas, com métodos para exibir e modificar diagramas em navegadores web. Desta forma, permite desenhar/renderizar elementos visuais e criar experiências interativas sobre eles. É importante realçar que este componente possui um modelo de dados interno (*data model*) que descreve quais os elementos que podem ser criados e como se relacionam.

A Figura 30 apresenta o diagrama de componentes do **diagram-js** permitindo visualizar os seguintes componentes:

- **Draw:** este componente é o responsável por desenhar/apresentar os elementos gráficos pretendidos no momento da modelação. Por outras palavras, pode-se dizer que este componente possui o conhecimento de como representar graficamente cada elemento do modelo de dados interno.

- **Serviços auxiliares:**
 - **CommandStack:** o funcionamento deste serviço representa a utilização do padrão comando (*command*). Durante o processo de modelação existe uma pilha (*stack*) que guarda as alterações efetuadas, possibilitando que seja possível anular alterações (*undo*) ou cancelar uma anulação através do avançar (*redo*).
 - **ContextPad:** fornece ações contextuais para um determinado elemento. Um exemplo de uma ação é permite ligar os componentes, através de conexões, com base nas regras do modelo de dados interno.
 - **Palette:** este serviço consiste na especificação de quais os elementos gráficos que podem ser escolhidos através de um menu.
 - **Modeling:** fornece APIs para atualizar elementos no momento da modelação (e.g. mover, excluir).
 - **Overlays:** disponibiliza APIs que permitem anexar informações adicionais a cada elemento gráfico do diagrama. Assim, é possível adicionar através de sobreposições elementos HTML a cada elemento gráfico.

- **Núcleo de Serviços (Core):** este componente visa assegurar a realização dos seguintes serviços essenciais:
 - **Canvas:** disponibiliza um conjunto de APIs para adicionar e remover elementos gráficos. Gere o ciclo de vida de cada elemento.
 - **EventBus:** é o canal de comunicações entres as diferentes componentes. Os outros componentes podem criar, remover ou inscreverem-se em eventos de outras bibliotecas, de forma que, são notificadas assim que estes acontecerem. Desta forma, potencializa a modularização e permite desacoplar responsabilidades. Este método é semelhante ao padrão *Publish-Subscribe*.
 - **ElementFactory:** é uma fábrica que permite criar formas e conexões de acordo com o modelo de dados interno definido pelo *diagram-js*.
 - **ElementRegistry:** conhece todos os elementos adicionados ao diagrama e fornece APIs que possibilitam recuperar os elementos e/ou sua representação gráfica pelo id de cada elemento. É responsável por validar das conexões entres os elementos, através do conhecimento definido no modelo de dados do *diagram-js*.

- **GraphicsFactory:** responsável por criar representações gráficas dos elementos como formas e conexões.

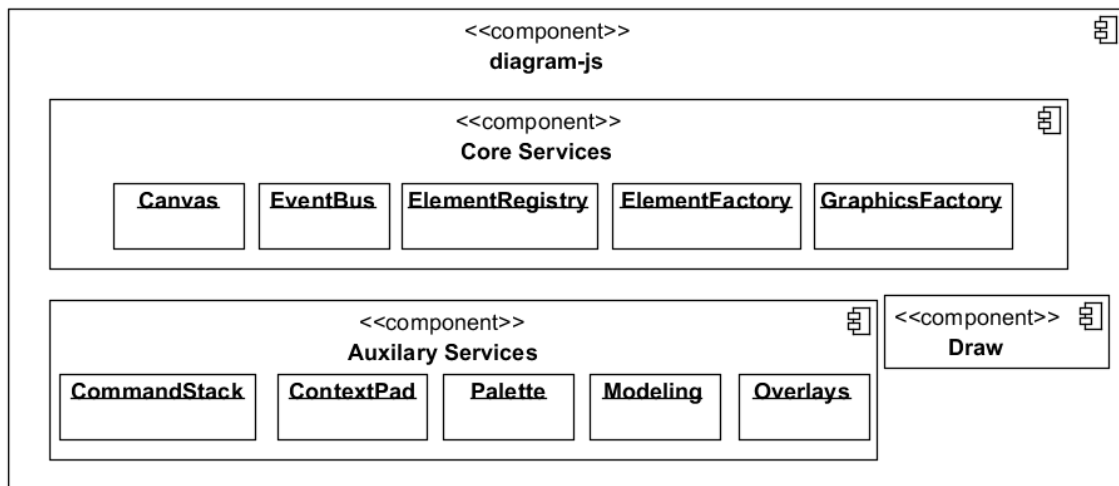


Figura 30 – Diagrama de Componente e Serviços do *diagram-js*

O outro componente usado pelo *bpmn-js* é o *bpmn-moddle* que é responsável por conhecer a estrutura do BPMN definido pelo padrão BPMN 2.0 (OMG, 2011). Permite ler e escrever documentos XML compatíveis com o esquema BPMN 2.0 e obter as informações relacionadas por de trás das formas e conexões desenhadas no diagrama. Pode-se afirmar que encapsula o conhecimento sobre BPMN, podendo assim durante o processo de modelação ou importação, efetuar validações. Os fundamentos do *bpmn-moddle* são baseados nas seguintes duas bibliotecas (cf. Figura 31):

- **moddle:** oferece uma forma concisa de definir os elementos que representam o BPMN em objetos *Java Script*. Pode-se dizer que o *moddle* tem o conhecimento de como os objetos em *JavaScript* representam a notação BPMN. Por outras palavras, o *moodle* é responsável por especificar o BPMN em objetos *JavaScript*. Deve-se realçar que esta biblioteca para entender a estrutura do BPMN, efetua uma leitura a um ficheiro JSON que representa esta notação.
- **moddle-xml:** permite ler e escreve documentos XML com base no *moodle*.

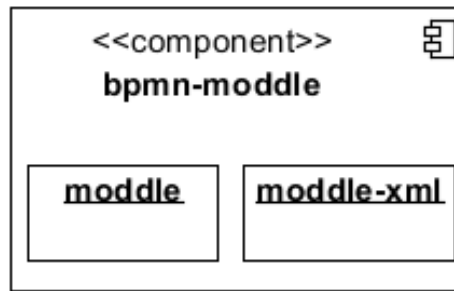


Figura 31 – Bibliotecas do Componente *bpmn-moddle*

Pode-se concluir que o componente *bpmn-moddle* usa a estrutura do BPMN como forma de validar os elementos criados no momento da modelação.

Uma das extensões (módulo) já existentes do BPMN.IO é o projeto *bpmn-js-properties-panel* (Camunda Services GmbH, 2017). Este componente permite criar um painel no interface gráfico do BPMN.IO onde, é possível editar as propriedades do elemento gráfico selecionado.

A componente *PropertiesPanel* é responsável no *bpmn-js-properties-panel*, por gerir os painéis de propriedades (*PropertiesActivator*). Este componente é notificado sobre o elemento gráfico que foi selecionado, após ter-se inscrito nos seguintes eventos do serviço *EventBus* do componente *diagram-js*:

- **selection.changed**: quando um elemento gráfico é selecionado é disparado este evento. Assim, a classe *PropertiesPanel* sabe qual o elemento selecionado.
- **diagram.destroy**: quando um elemento gráfico é eliminado do diagrama o componente *PropertiesPanel* é informado, assim pode destruir o painel (*PropertiesActivator*) que permite editar as propriedades desse elemento.

Após um elemento gráfico ter sido selecionado o *PropertiesPanel* delega a responsabilidade de apresentar o formulário de configurações ao respetivo *PropertiesActivator* e aguarda que o *PropertiesActivator* lhe transmita as alterações através de comandos. Este comando deve ser criado pelo serviço *CmdHelper*, de forma que se respeite a seguinte informação: qual o elemento e o respetivo atributo alterado.

O *PropertiesActivator* é a uma classe abstrata que permite definir um painel de propriedades onde, é possível especificar quais os atributos de determinados elementos que podem ser editados. Esta classe é notificada pelo *PropertiesPanel* que informa qual o elemento gráfico selecionado, se esta classe tiver um serviço de configuração para este elemento, é apresentado o formulário de configurações definido no serviço. Relaciona-se com o componente

PropertiesPanel através de uma inscrição no seguinte evento **propertiesPanel.isEntryVisible** através do serviço **EventBus**. Este evento notifica qual o elemento gráfico selecionado.

Após esta explicação, é perceptível que o elemento que deve ser estendido é o **PropertiesActivator**. Com esta extensão consegue-se apresentar os formulários de configuração para o elemento da notação desejado. Na realização desta extensão é necessário trabalhar com as seguintes serviços:

- **CmdHelper**: formulário permite criar comandos que representam as alterações após a edição de um atributo. Ao usar-se este serviço é possível validar se a edição
- **ElementHelper**: este serviço permite criar o elemento desejado com base na descrição existente no componente **bpmn-moodle**.

Ainda, para se poder agilizar o processo de construção da interface gráfica dos formulários, pode-se utilizar as seguintes serviços do componente **bpmn-js-properties-panel**:

- **IdProps**: no momento da edição, permite verificar se o ID é válido.
- **EntryFactory**: esta biblioteca permite criar os elementos HTML necessários, para se configurar os atributos do elemento gráfico selecionado (e.g. *label*, *input*, *select*, *textarea*, entre outros).

A Figura 32 apresenta as alterações necessárias para se poder estender o projeto **bpmn-js-properties-panel**, para com base nos elementos que são necessários configurar das regras do diálogo de negócio.

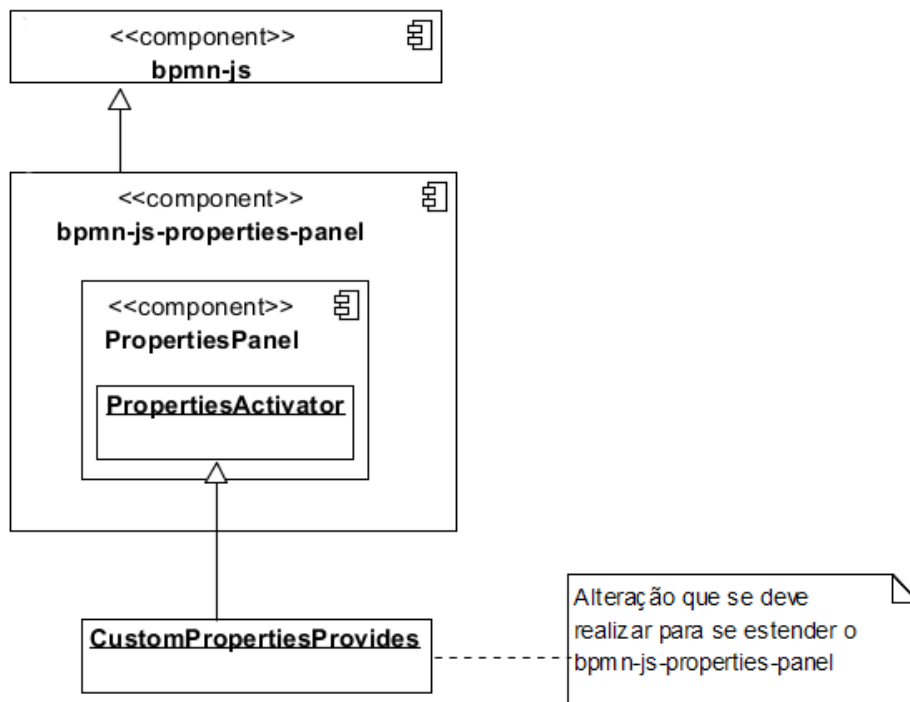


Figura 32 – Alterações no Componente do *bpmn-js-properties-panel*

A escolha do *bpmn-js-properties-panel* como ponto de partida, para satisfazer as necessidades de configuração dos diálogos de negócio, parece ser uma solução aceitável, uma vez que permite reduzir substancialmente o tempo de implementação.

5.3.2 Extensão da Ferramenta BPMN.IO

O objetivo desta subsecção é que com base na especificação realizado na subsecção anterior, sobre a ferramenta BPMN.IO, se enunciem as extensões necessárias realizar nesta ferramenta, para que seja possível modelar diálogos de negócio.

Com base no estudo realizado sobre a arquitetura da ferramenta BPMN.IO concluiu-se, que as extensões necessárias realizar são nos componentes *bpmn-moddle* e *diagram-js*.

Para ser possível modelar os diálogos de negócio segundo a notação de especificação de diálogos, é necessário estender o componente que valida a estrutura BPMN no BPMN.IO. Desta forma, o componente que deve ser estendido é *bpmn-moddle* uma vez que, é o responsável por ler e escrever documentos XML com base na estrutura da notação BPMN. A Figura 30 apresenta um diagrama de componentes descrevendo a extensão necessária.

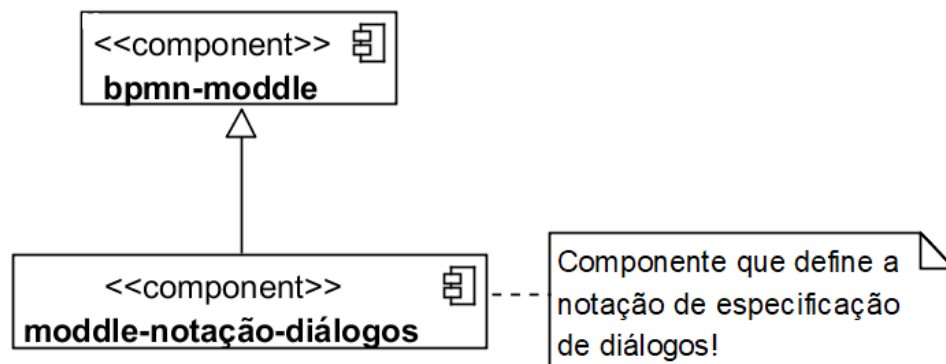


Figura 30 – Extensão do BPMN no BPMN.IO

O componente **bpmn-moddle** requer que seja necessário criar um ficheiro no formato JSON, que descreva a extensão que se pretende realizar sobre o BPMN. O objetivo do ficheiro JSON é que seja consumido pela biblioteca **moddle**, de forma que esta biblioteca perceba como representar a notação de especificação de diálogos no momento da modelação, em objetos *JavaScript*. Este requisito obriga a converter do XSD que representa a notação de especificação de diálogos para o formato JSON.

O outro componente do BPMN.IO que deve ser estendido é o **diagram-js**. Neste componente é necessário estender o subcomponente **Draw**, uma vez que é o componente responsável por apresentar e desenhar os elementos gráficos. Com esta extensão, é possível representar graficamente os seguintes elementos: 1) questões ao utilizador, 2) questões ao sistema e 3) respostas estáticas. Só é necessário representar estes elementos graficamente uma vez que, os restantes elementos necessários já fazem parte do BPMN.IO (e.g. *start event, end event, exclusive gateway, data storage reference, etc.*).

Ainda, no componente **diagram-js** é necessário alterar os seguintes serviços auxiliares:

- Estender o serviço **Pallette** para que no menu apareça as seguintes opções: 1) questões ao utilizador, 2) questões ao sistema e 3) respostas estáticas.
- Estender o serviço **ContextPad** para que no momento da modelação seja possível criar conexões entre as seguintes tarefas de aquisição: 1) questões ao utilizador, 2) questões ao sistema e 3) respostas estáticas.

Outra necessidade é estender o interface gráfico, de forma que seja possível efetuar as configurações necessárias no diálogo de negócio. O objetivo desta adaptação consiste que no momento da modelação, quando se seleciona um elemento gráfico que represente uma regra do diálogo de negócio, seja possível configurar os seus atributos através de um formulário.

Como referido na subsecção na subsecção 5.3.1, para a definição dos componentes gráficos que permitem a configuração dos elementos da notação de especificação de diálogos, deve-se

estender o módulo *bpmn-js-properties-panel*, mais concretamente a classe *PropertiesActivator* e criar serviços para os elementos que se pretende configurar. A Figura 33 apresenta como se deve realizar esta extensão e quais os serviços essenciais que deve ser criados.

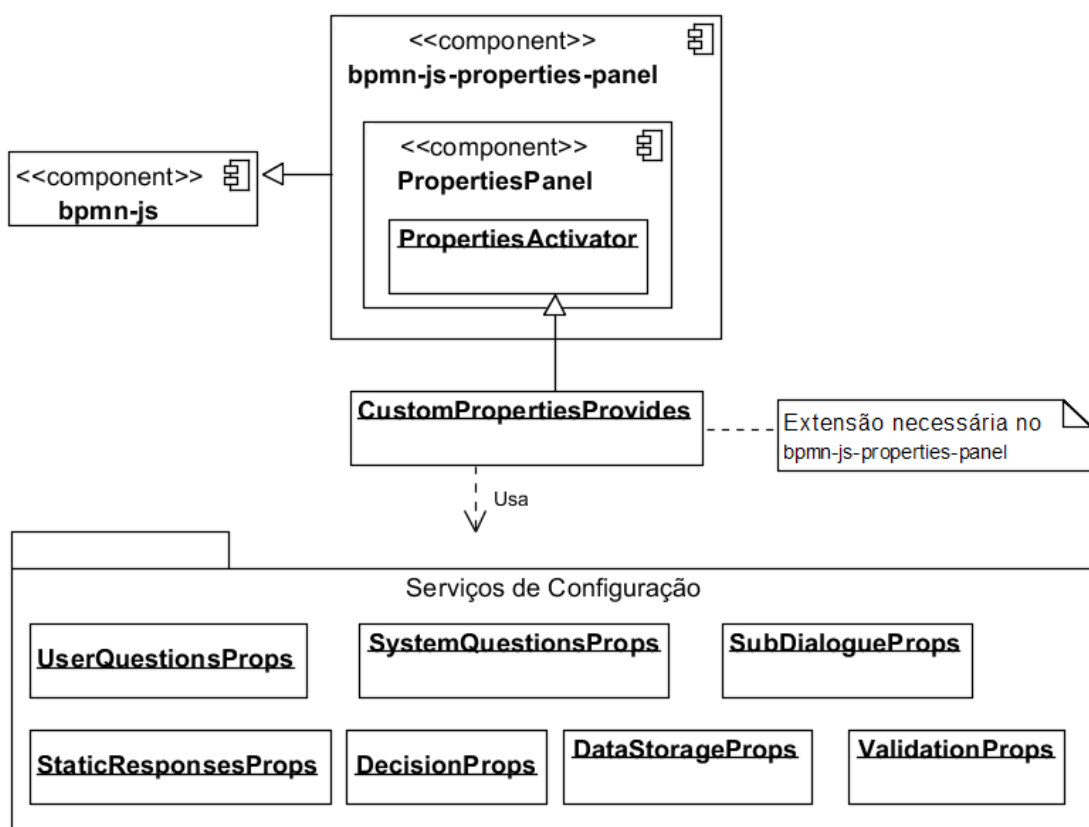


Figura 33 – Extensão do bpmn-js-properties-panel

A extensão *CustomPropertiesProvides* necessita de utilizar diferentes serviços para cada um dos elementos da notação que se pretende configurar. A seguir são enumeradas os serviços que devem ser implementados, que em alguns casos são utilizadas para configurar diferentes elementos:

- **UserQuestionsProps**: este serviço é responsável por permitir configurar uma lista de questões ao utilizador (tarefa de aquisição questões ao utilizador).
- **SystemQuestionsProps**: este serviço é responsável por permitir configurar uma lista de questões ao sistema (tarefa de aquisição questões ao sistema).
- **StaticResponseProps**: este serviço é responsável por permitir configurar uma lista de respostas pré-definidas (tarefa de aquisição respostas estáticas).
- **DecisionProps**: este serviço é responsável por permitir configurar decisões.

- **ValidationProps:** este serviço é responsável por permitir configurar validações sobre os dados recolhidos (local ou remota). Pode ser usada pelos serviços **UserQuestionsProps** e **DecisionProps**.
- **SubDialogueProps:** este serviço é responsável por permitir configurar um subdiálogo.

5.4 Interpretador

O processo de aquisição refere-se ao momento em que um determinado utilizador, com a ajuda de uma UAD executa um diálogo de negócio. Nesta subsecção, é apresentada uma proposta de *design* para o componente interpretador, seguindo a análise e requisitos realizada na subsecção 4.2.

A Figura 34 apresenta uma arquitetura possível para o interpretador, onde se encontra representado os seguintes subcomponentes:

- **Gestor do Fluxo de Aquisição:** é o componente responsável por controlar o fluxo de aquisição e delegar as diferentes responsabilidades aos outros componentes, dependendo da regra do diálogo de negócio que esteja a ser executada.
- **Transformação do diálogo:** é responsável pelas seguintes duas tarefas:
 1. Ler o ficheiro que representa o diálogo de negócio;
 2. Interpreta o diálogo de negócio com base na notação de especificação de diálogos.
- **Armazenamento:** permite armazenar os dados recolhidos nos diferentes tipos de armazenamento. Em armazenamento local através dos recursos da UAD ou, usando o componente **Pedidos** para que possa guardar os dados remotamente no sistema de gestão.
- **Decisão:** responsável por decidir qual o caminho que o fluxo deve tomar quando o gestor do fluxo executar uma decisão. Como a decisão contém validações, este componente deve usar o componente **Validações**.
- **Validações:** possibilita realizar validações sobre os dados recolhidos. Estas validações podem ser locais ou remotas. É possível realizar validações remotas através do uso do componente **Pedidos**, para se delegar a validação ao sistema de gestão. A implementação deste componente pode ser realizada através da utilização do padrão de projeto Strategy (Gamma et al., 1994), desta forma

dependendo do sistema de gestão ou da UAD o comportamento deste componente pode mudar.

- **Pedidos:** permite interagir com a aplicação de gestão do processo de gestão de informação. A implementação deste componente pode ser realizada através da utilização do padrão de projeto *Strategy*, desta forma dependendo do sistema de gestão o comportamento deste componente pode mudar.
- **Interação com a UAD:** este componente deve disponibilizar um interface que represente as interações necessárias que o interpretador deve conseguir realizar na UAD. Os diferentes modelos de UADs obrigam que se desenvolva um adaptador para cada modelo, que permita realizar as tarefas necessárias em cada UAD. Este componente deve ser implementado através da utilização do padrão de projeto *Adapter* (Gamma et al., 1994).
- **Adaptador UAD 1:** representa o componente necessário que se deve desenvolver, de modo que seja possível converter o interface que define as interações necessárias com o tipo de UAD para este modelo (UAD 1).

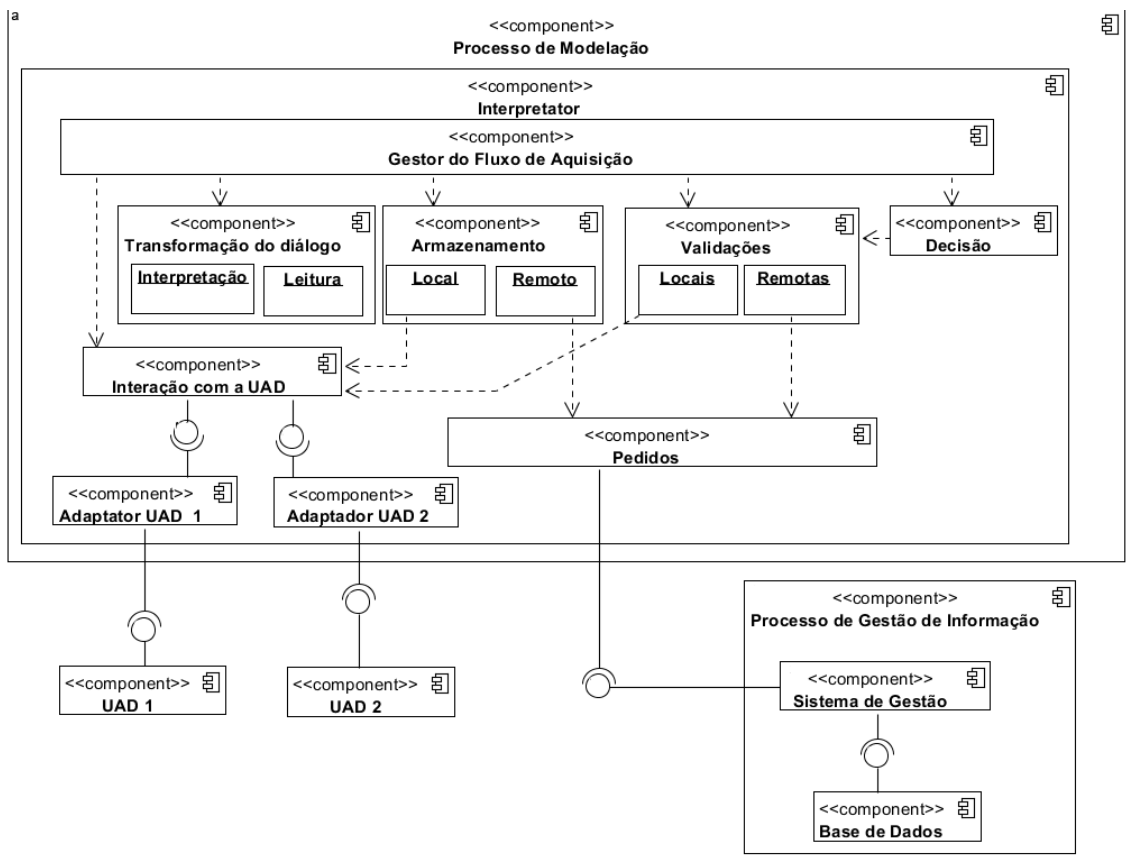


Figura 34 – Arquitetura do Interpretador

6 Implementação

Este capítulo destina-se à descrição do processo de implementação dos componentes que fazem parte da proposta de solução. Encontra-se dividido nas seguintes subsecções:

1. **Notação de especificação de diálogos:** descreve a implementação do processo de modelação.
2. **Plataforma de modelação:** enuncia o processo de desenvolvimento e avaliação do processo de aquisição.
3. **Interpretador:** para validar a interação entre o processo de aquisição e o processo de gestão, existe a necessidade de se criar uma aplicação que simule um sistema de gestão. Esta secção apresenta uma aplicação protótipo que faz essa simulação.
4. **Sistema de Gestão de informação:** para o interpretador poder interagir com o processo de gestão, implementou-se um sistema que simula um sistema de gestão de informação.

É relevante salientar que a implementação da plataforma de modelação e do interpretador, são apresentados protótipos com base na análise de *design* realizada no capítulo 4. Desta forma, é possível validar o *design* efetuado. Ambos os componentes foram desenvolvidos para funcionarem em navegadores web, sem a necessidade de servidor (*server-side*), utilizou-se o Node.js (Dahl, 2017) para validar o *JavaScript* e proporcionar um processo de desenvolvimento mais escalável. Assim, foi possível o uso do NPM (*Node Package Manager*) para instalar e gerir as dependências das bibliotecas utilizadas.

6.1 Notação de Especificação de Diálogos

Como descrito na subsecção 5.2, a notação de modelação que se deve estender para ser possível representar os diálogos de negócio, é a notação BPMN.

Como a estrutura do BPMN é definida por um ficheiro XSD e para ser possível reutilizar esta estrutura como base, é necessário descrever as regras do diálogo de negócio num ficheiro XSD, de forma que seja possível estender os conceitos base do BPMN.

A Figura 35 apresenta um excerto do XSD da extensão realizada ao BPMN onde, é apresentado apenas as tarefas de aquisição. Nesta figura, é possível observar que o diálogo (*Dialogue*) é uma extensão do elemento processo do BPMN (*Process*), a tarefa questões ao utilizador (*QuestionsTask*) estende o elemento tarefa ao utilizador (*User Task*) do BPMN e a tarefa questões ao sistema (*SystemTask*) e respostas estáticas (*StaticResponses*) estendem o elemento Tarefa (*Task*) do BPMN.

É possível visualizar o ficheiro XSD completo que contém a descrição da notação de especificação de diálogos, através deste endereço: <http://1070812-tmdei.azurewebsites.net/>

```

<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
  elementFormDefault="qualified"
  xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
  xmlns:tmdei="http://tempuri.org/XMLSchema.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.omg.org/spec/BPMN/20100524/MODEL" schemaLocation="BPMN20.xsd"/>

  <xs:complexType name="Dialogue">
    <xs:complexContent>
      <xs:extension base="bpmn2:tProcess">
        <xs:attribute name="lang" use="required" type="xs:string"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="UserQuestions">
    <xs:complexContent>
      <xs:extension base="bpmn2:tUserTask">
        <xs:sequence>
          <xs:element name="label" type="xs:string" />
          <xs:element name="description" type="xs:string" />
          <xs:element ref="tmdei:Question" minOccurs="1" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SystemQuestions">
    <xs:complexContent>
      <xs:extension base="bpmn2:tTask">
        <xs:sequence>
          <xs:element name="label" type="xs:string" />
          <xs:element name="description" type="xs:string" />
          <xs:element ref="tmdei:Question" minOccurs="1" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="StaticResponses">
    <xs:complexContent>
      <xs:extension base="bpmn2:tTask">
        <xs:sequence>
          <xs:element ref="tmdei:Response" minOccurs="1" maxOccurs="unbounded" />
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

Figura 35 – Exemplo da Extensão Realizada ao BPMN

6.2 Plataforma de Modelação

Esta subsecção descreve a implementação da plataforma de modelação, seguindo as alterações que foram descritas como necessárias no capítulo do *design*, nomeadamente na subsecção 5.3.2.

Com a escolha da ferramenta de modelação foi BPMN.IO, condicionou que a primeira alteração que se realizou, fosse a transformação do XSD que define a notação de especificação de

diálogos para o formato JSON. A Figura 36 apresenta a transformação do XSD efetuada para o formato JSON, apenas das tarefas de aquisição para não ser muito exaustivo. O objetivo deste ficheiro é que seja consumido pelo componente **bpmn-moddle**, de forma que este componente consiga validar as regras do diálogo de negócio e criar os diálogos de negócio (XML) com base no ficheiro JSON.

A representação gráfica dos elementos que representam as tarefas de aquisição deve ser realizada através da extensão do componente **Draw**. A Figura 37 apresenta o aspeto gráfico definido para cada tarefa de aquisição, que aparece no momento da modelação do diálogo de negócio. Já a Figura 38 apresenta o código *JavaScript* da extensão realizada ao componente **Draw**.

```

{
  "name": "TMDEI",
  "prefix": "tmdei",
  "uri": "http://tmdei",
  "xml": {
    "tagAlias": "lowerCase"
  },
  "associations": [],
  "types": [
    {
      "name": "Dialogue",
      "superClass": [ "bpmn:Process" ],
      "properties": [
        {
          "name": "lang",
          "type": "String",
          "isAttr": true
        }
      ]
    },
    {
      "name": "UserQuestions",
      "superClass": [ "bpmn:UserTask" ],
      "properties": [
        {
          "name": "questions",
          "type": "Questions"
        }
      ]
    },
    {
      "name": "SystemQuestions",
      "superClass": [ "bpmn:Task" ],
      "properties": [
        {
          "name": "questions",
          "type": "Questions"
        }
      ]
    },
    {
      "name": "StaticResponses",
      "superClass": [ "bpmn:Task" ],
      "properties": [
        {
          "name": "responses",
          "type": "Responses"
        }
      ]
    }
  ]
},

```

Figura 36 – Tarefas de Aquisição Representadas em JSON

É possível visualizar o ficheiro JSON completo que contém a descrição da notação de especificação de diálogos, através deste endereço: <http://1070812-tmdei.azurewebsites.net/>



Figura 37 – Representação Gráfica das Tarefas de Aquisição no BPMN.IO

```

var inherits = require('inherits');
var BaseRenderer = require('diagram-js/lib/draw/BaseRenderer');
var isAny = require('bpmn-js/lib/features/modeling/util/ModelingUtil').isAny;
var images = require('./images');
var svgAppend = require('tiny-svg/lib/append'),
    svgCreate = require('tiny-svg/lib/create');

function ExtRender(eventBus) {
  BaseRenderer.call(this, eventBus, 1500);

  this.canRender = function(element) {
    return isAny(element, ['tmdei:UserQuestions', 'tmdei:SystemQuestions', 'tmdei:StaticResponses']);
  };

  this.drawShape = function(parent, shape) {
    var url = undefined;
    if(shape.type === "tmdei:UserQuestions")
    {
      url = images.userQuestionsURL;
    }else if(shape.type === "tmdei:SystemQuestions")
    {
      url = images.systemQuestionsURL;
    }else if(shape.type === "tmdei:StaticResponses")
    {
      url = images.staticResponsesURL;
    }
    var gfx = svgCreate('image', {
      x: 0,
      y: 0,
      width: shape.width,
      height: shape.height,
      href: url
    });
    svgAppend(parent, gfx);

    return gfx;
  };
}
inherits(ExtRender, BaseRenderer);
module.exports = ExtRender;

```

Figura 38 – Extensão Realizada do Componente *Draw*

A seguir é apresentada a implementação realizada de modo a estender-se o componente *diagram-js*, nomeadamente os serviços *Palette* e *ContextPad*.

O BPMN.IO possui um menu lateral onde contém os elementos gráficos do processo de modelação. Este menu é definido pelo serviço *Palette*. Para que neste menu seja possível

aparecer as tarefas de aquisição é necessário estender este serviço. A Figura 39 apresenta a extensão realizada ao serviço **Palette**.

```
var images = require('../images');
function ExtPaletteProvider(palette, create, elementFactory) {
  this._create = create;
  this._elementFactory = elementFactory;
  palette.registerProvider(this);
}

ExtPaletteProvider.prototype.getPaletteEntries = function() {
  var elementFactory = this._elementFactory,
      create = this._create;
  function startCreateUserQuestions(event) {
    var serviceTaskShape = elementFactory.create('shape', { type: 'tmdei:UserQuestions' });
    create.start(event, serviceTaskShape);
  }
  function startSystemQuestions(event) {
    var serviceTaskShape = elementFactory.create('shape', { type: 'tmdei:SystemQuestions' });
    create.start(event, serviceTaskShape);
  }
  function startStaticResponses(event) {
    var serviceTaskShape = elementFactory.create('shape', { type: 'tmdei:StaticResponses' });
    create.start(event, serviceTaskShape);
  }
  return {
    'create-services-userQuestions': {
      group: 'activity',
      title: 'Create a new TMDEI User Questions!',
      imageUrl: images.userQuestionsURL,
      action: {
        dragstart: startCreateUserQuestions,
        click: startCreateUserQuestions
      }
    },
    'create-services-systemQuestions': {
      group: 'activity',
      title: 'Create a new TMDEI System Questions!',
      imageUrl: images.systemQuestionsURL,
      action: {
        dragstart: startSystemQuestions,
        click: startSystemQuestions
      }
    },
    'create-services-staticResponses': {
      group: 'activity',
      title: 'Create a new TMDEI Static Responses!',
      imageUrl: images.staticResponsesURL,
      action: {
        dragstart: startStaticResponses,
        click: startStaticResponses
      }
    }
  };
};
module.exports = ExtPaletteProvider;
```

Figura 39 – Extensão do Serviço *Palette*

Para ser possível interligar através de conexões as tarefas de aquisição é necessário estender o serviço **ContextPad**. Esta extensão permitir conectar as tarefas de aquisição com os elementos já existentes no BPMN (e.g. *start event*, *exclusive gateway*, *end event*, etc.). A Figura 40 apresenta a extensão implementada.

```

var inherits = require('inherits');
var ContextPadProvider = require('bpmn-js/lib/features/context-pad/ContextPadProvider');
var isAny = require('bpmn-js/lib/features/modeling/util/ModelingUtil').isAny;
var assign = require('lodash/object/assign'),
    bind = require('lodash/function/bind');

function CustomContextPadProvider(eventBus, contextPad, modeling, elementFactory, connect,
    create, popupMenu, canvas, rules, translate) {
  ContextPadProvider.call(this, eventBus, contextPad, modeling, elementFactory, connect, create,
    popupMenu, canvas, rules, translate);

  var cached = bind(this.getContextPadEntries, this);

  this.getContextPadEntries = function(element) {
    var actions = cached(element);
    var businessObject = element.businessObject;
    function startConnect(event, element, autoActivate) {
      connect.start(event, element, autoActivate);
    }
    if (isAny(businessObject, [ 'tmdei:UserTask' ])) {
      assign(actions, {
        'connect': {
          group: 'connect',
          className: 'bpmn-icon-connection-multi',
          title: translate('Connect using custom connection'),
          action: {
            click: startConnect,
            dragstart: startConnect
          }
        }
      });
    }
    return actions;
  };
}

inherits(CustomContextPadProvider, ContextPadProvider);

```

Figura 40 – Implementação da Extensão do *ContextPad*

Ainda como referido na subsecção Extensão da Ferramenta BPMN.IO, os formulários que permitem a configuração são implementados através de uma extensão da classe ***PropertiesActivator***. A Figura 41 apresenta a extensão realizada.

```

function TMDEIPropertiesProvider(eventBus, bpmnFactory, elementRegistry, modeling) {
  PropertiesActivator.call(this, eventBus);
  this.getTabs = function(element) {
    var generalTab = {
      id: 'general',
      label: 'General',
      groups: createGeneralTabGroups(element, bpmnFactory, elementRegistry,modeling)
    };
    // The "user questions" tab
    var userQuestionsTab = {
      id: 'userQuestions',
      label: 'User Questions',
      groups: createUserQuestionsTabGroups(element,bpmnFactory, elementRegistry,modeling)
    };
    // The "system questions" tab
    var systemQuestionsTab = {
      id: 'systemQuestions',
      label: 'System Questions',
      groups: createSystemQuestionsTabGroups(element,bpmnFactory, elementRegistry,modeling)
    };
    // The "static responses" tab
    var staticResponsesTab = {
      id: 'staticResponses',
      label: 'Static Responses',
      groups: createStaticResponsesTabGroups(element,bpmnFactory, elementRegistry,modeling)
    };
    // The "questions" tab
    var subProcessTab = {
      id: 'subProcess',
      label: 'Sub Dialogue',
      groups: createSubProcessTabGroup(element,bpmnFactory, elementRegistry,modeling)
    };
    // The exclusive gateway tab
    var exclusiveGatewayTab = {
      id: 'exclusiveGateway',
      label: 'Decisions',
      groups: createExclusiveGatewayTabGroup(element,bpmnFactory, elementRegistry,modeling)
    };
    // The exclusive Data Storage
    var dataStorageTab = {
      id: 'dataStorage',
      label: 'Data Storage',
      groups: createDataStorageTabGroup(element,bpmnFactory, elementRegistry,modeling)
    };
    // Show general + "form" tab
    return [
      generalTab,
      userQuestionsTab,
      systemQuestionsTab,
      staticResponsesTab,
      subProcessTab,
      exclusiveGatewayTab,
      dataStorageTab
    ];
  };
}

```

Figura 41 – Extensão do Classe *PropertiesActivator*

Até ao momento, foram descritas todas as alterações que se efetuaram para se adaptar esta ferramenta de modelação às necessidades do processo de modelação. Como referido no *design*, no construtor do objeto **BpmnModeler** possibilita que seja configurado módulos adicionais e

também extensões ao **bpmn-moddle**. A Figura 42 apresenta a configuração das extensões realizadas onde, os números representam as seguintes extensões:

1. Inclusão do projeto **bpmn-js-properties-panel**
2. Extensão do **bpmn-js-properties-panel**
3. Extensão do serviço **Palette** (menu)
4. Extensão do componente **Draw**
5. Extensão do serviço **ContextPad**
6. Extensão do **bpmn-moddle**

```
this.bpmnModeler = new BpmnModeler({
  container: container,
  propertiesPanel: {
    parent: parent
  },
  additionalModules: [
    propertiesPanelModule, 1
    propertiesProviderModule, 2
    extPaletteModule, 3
    extDrawModule, 4
    extPadProvider 5
  ],
  moddleExtensions: {
    tmdei: tmdeiModdleDescriptor 6
  }
});
```

Figura 42 – Configuração dos Componentes Desenvolvidos no BPMN.IO

A Figura 43 apresenta a plataforma de modelação desenvolvida, onde os números representam os seguintes conceitos:

1. Início do diálogo.
2. Tarefa de aquisição “Questões ao utilizador”.
3. Processo de decisão.
4. Sequência do fluxo por omissão, em caso das validações não serem positivas no processo de decisão.
5. Sequência do fluxo em caso de sucesso na validação do processo decisão.
6. Tarefa de aquisição “Respostas estáticas”.
7. Tarefa de aquisição “Questões ao sistema”.
8. Armazenamento de dados.
9. Fim do diálogo.
10. Painel que permite configurar o elemento gráfico selecionado.

11. Menu lateral de onde é possível criar elementos gráficos.

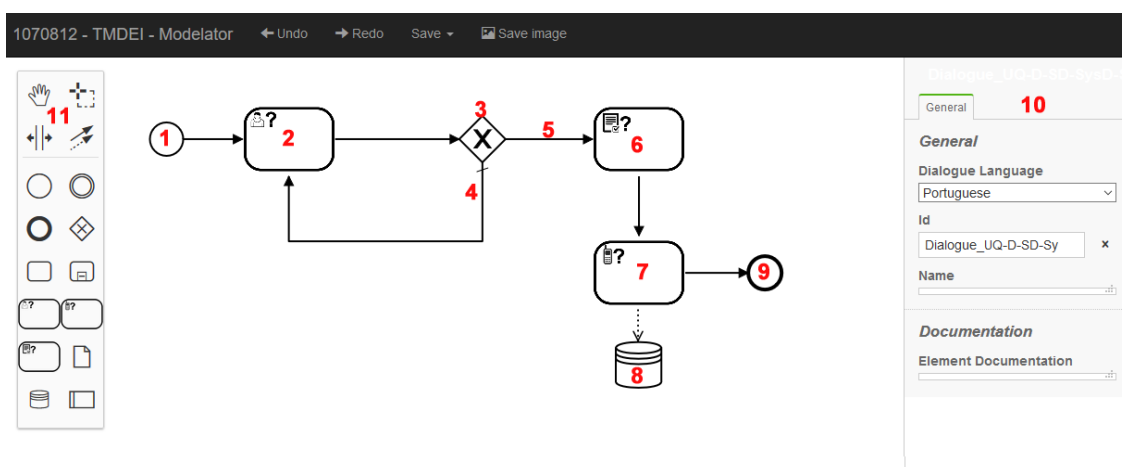


Figura 43 – Interface Gráfico da Plataforma de Modelação

A plataforma de modelação encontra-se publicada e disponível para consulta e utilização no seguinte endereço: <http://1070812-tmdei.azurewebsites.net>

6.3 Interpretador

O objetivo desta subsecção é documentar o processo de implementação do componente interpretador, referente ao processo de aquisição. Este desenvolvimento foi realizado com base no *design* descrito na subsecção 5.4.

O interpretador foi implementado com o objetivo de permitir recolher os dados pretendidos, através do uso de navegadores web (*browsers*) como UAD. Desta forma, é possível a execução de diálogos em diferentes dispositivos (e.g. computador, smart phone, tablet, etc.).

Para se realizar uma correlação entre o *design* e a implementação do interpretador, nesta subsecção, efetua-se a respetiva correspondência ao componente do interpretador descrito anteriormente no *desing* (cf. subsecção 5.4).

Uma vez que o interpretador é para ser executado em navegadores web, todos os componentes foram implementados usando a linguagem *JavaScript*.

O núcleo (core) do interpretador é o componente **gestor do fluxo de aquisição**, este componente tem a responsabilidade de controlar o fluxo de aquisição, delegando responsabilidades, dependendo do estado de aquisição, aos outros componentes. Este componente foi implementado através de uma classe *JavaScript* em que no momento da sua instanciação são configuradas as suas dependências (os restantes componentes).

O componente responsável pela instanciação do diálogo de negócio é o componente **transformação do diálogo**. Este componente para conseguir construir o interface gráfico necessário para o processo de aquisição, utiliza um ficheiro XSLT (*Extensible Stylesheet Language Transformations*). Desta forma, é possível mudar o interface gráfico alterando o XSLT. Foram criados dois ficheiros XSLT, as diferenças entre estes ficheiros, é que um recolhe números inteiros utilizando um teclado como método de aquisição. O segundo XSLT permite recolher números através da leitura de códigos de barras, usando a webcam como dispositivo de leitura. As dependências utilizadas por este componente são:

Como a UAD são os navegadores web, nesta implementação a interligação entre a UAD referente ao componente **interação com a UAD** é realizada pela transformação XSLT, que permite relacionar os elementos HTML (e.g. *input*, *select*) com o componente **gestor do fluxo de aquisição**. Deste modo, sempre que existe uma introdução por parte do utilizador, o componente **gestor do fluxo de aquisição** é notificado e executa o fluxo conforme a especificação do diálogo. A Figura 44 representa a lógica de instanciação do processo de aquisição por parte do interpretador.

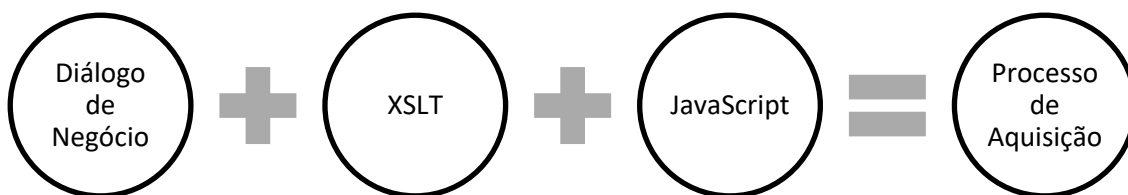


Figura 44 – Lógica de Instanciação do Interpretador

Na implementação do interpretador foram usadas diversas dependências utilizando o NPM. De seguida são enumeradas com base no componente as principais dependências usadas e os seus objetivos.

O componente **gestor do fluxo de aquisição** utiliza **Jquery** (JQuery, 2017) para conseguir definir métodos através do uso de seletores com base nos atributos de cada elemento. Desta forma, interligou-se o interface gráfico com o componente **gestor do fluxo de modelação**. Este processo ocorre no momento da transformação por parte do componente **transformação do diálogo**. Ainda utiliza a biblioteca **xml2js** (Kubica, 2017) para converter o ficheiro XML em objetos *JavaScript*.

O componente **transformação do diálogo** utiliza a biblioteca **xslt** (Murray, 2017) para conseguir transformar o ficheiro XML que descreve o diálogo de negócio em HTML, utilizando *JavaScript*.

O componente de validações utiliza a biblioteca **jquery-validation** (Staab, 2017). Estendeu-se esta biblioteca para ser possível realizar as validações personalizadas sobre os dados recolhidos. Essencialmente esta extensão, no momento da introdução dos dados pelo utilizador, transfere a responsabilidade dos dados recolhidos ao componente (**validações**) que permite fazer as validações.

O componente **validações** pode executar validações locais (condicional e expressão regular) e validações remotas. Utiliza a biblioteca **moment** (Ichernev, 2017) para conseguir validar os dados recolhidos do tipo data e data e hora. Para as validações remotas este componente delega a responsabilidade ao componente **pedidos**.

O componente **pedidos** consegue comunicar com o sistema de gestão através do protocolo HTTP. Dependendo da configuração do diálogo, consegue executar pedido POST ou GET com diferentes formatos de dados (e.g. XML ou JSON). Utiliza a biblioteca **xpath** (Rishe, 2017) para seleccionar elementos num ficheiro XML no momento que efetua um pedido ao sistema de gestão e é devolvido um ficheiro XML como resposta.

O componente **Armazenamento** utiliza a biblioteca **js-cookie** (Brack, 2017) para guardar dados no navegador web através do recurso a *cookies*. Para armazenamento remoto este componente delega a responsabilidade armazenamento remoto ao componente **pedidos**.

O componente **decisões** com base em cada condição delega o processo de validação ao componente **validações**.

Foi ainda usada a biblioteca **QuaggaJS** (Oberhofer, 2017) para possibilitar a recolha dos dados através da leitura de um código de barras. Esta leitura é realizada através do *webcam* da UAD.

A Figura 45 apresenta a transformação que o XSLT efetua sobre a regra introdução (*inputData*). Nesta figura é possível perceber que é contruído um elemento HTML do tipo *input*, e através da definição dos atributos personalizados deste elemento são descritas as regras adicionais (e.g. tipo de validação). A utilização dos atributos possibilita que no momento da execução do diálogo, o gestor do fluxo de aquisição consiga determinar qual o componente que deve usar (e.g. validações, pedidos, etc.).

```

<xsl:choose>
  <xsl:when test="tmdei:response/@type='input'">
    <xsl:choose>
      <xsl:when test="tmdei:response/tmdei:inputData/@type='date'">
        <label for="{@id}"><xsl:value-of select="@label"/></label>
        <div class="input-group date">
          <input id="{@id}" name="{@id}" placeholder="{@label}" class="form-control" required=""
            question-type="{@type}"
            input-type="{tmdei:response/tmdei:inputData/@type}"
            validation-type="{tmdei:response/tmdei:validation/@type}"
            validation-logicType="{tmdei:response/tmdei:validation/@logicType}"
            validation-local-method="{tmdei:response/tmdei:validation/@method}"
            validation-local-operator="{tmdei:response/tmdei:validation/@operator}"
            validation-local-testValue="{tmdei:response/tmdei:validation/@testValue}"
            validation-local-regularExpression="{tmdei:response/tmdei:validation/tmdei:regularExpression}"
            validation-local-javascript="{tmdei:response/tmdei:validation/tmdei:javascript}"

            validation-local-range-starts="{tmdei:response/tmdei:validation/@starts}"
            validation-local-range-ends="{tmdei:response/tmdei:validation/@ends}"

            validation-request-status=""
            validation-request-type="{tmdei:response/tmdei:validation/tmdei:request/@type}"
            validation-request-method="{tmdei:response/tmdei:validation/tmdei:request/@method}"
            validation-request-dataType="{tmdei:response/tmdei:validation/tmdei:request/@dataType}"
            validation-request-relativePath="{tmdei:response/tmdei:validation/tmdei:request/@relativePath}"
            validation-request-xpathExpression="{tmdei:response/tmdei:validation/tmdei:request/@xpathExpression}"

          />
          <span class="input-group-addon"><i class="glyphicon glyphicon-th"/></span>
        </div>
      </xsl:when>
    </xsl:choose>
  </xsl:when>

```

Figura 45 – XSLT: transformação da regra “Introdução” para HTML

A Figura 46 apresenta a interface gráfica do interpretador desenvolvido onde, os números representam os seguintes opções:

1. Permite selecionar qual o XSLT que deve fazer o processo de instanciação.
2. Permite selecionar do diálogo.
3. Permite selecionar do sistema de gestão (*backoffice*).
4. Opções do armazenamento local (ver e eliminar dados).
5. Permite adicionar um novo diálogo.
6. Inicia a execução do diálogo selecionado.
7. Área onde é apresentado o interface gráfico da execução do diálogo.
8. Consola onde é possível acompanhar o fluxo do diálogo.

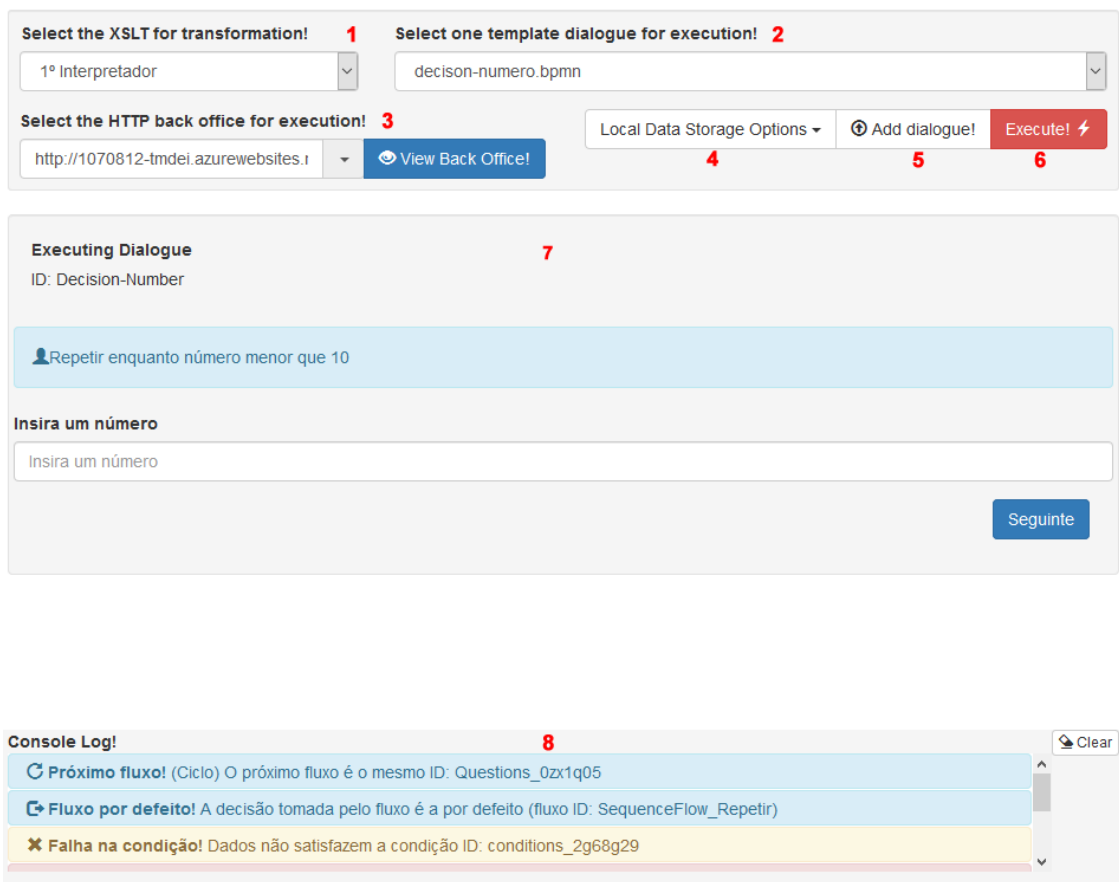


Figura 46 – Interface Gráfico do Interpretador

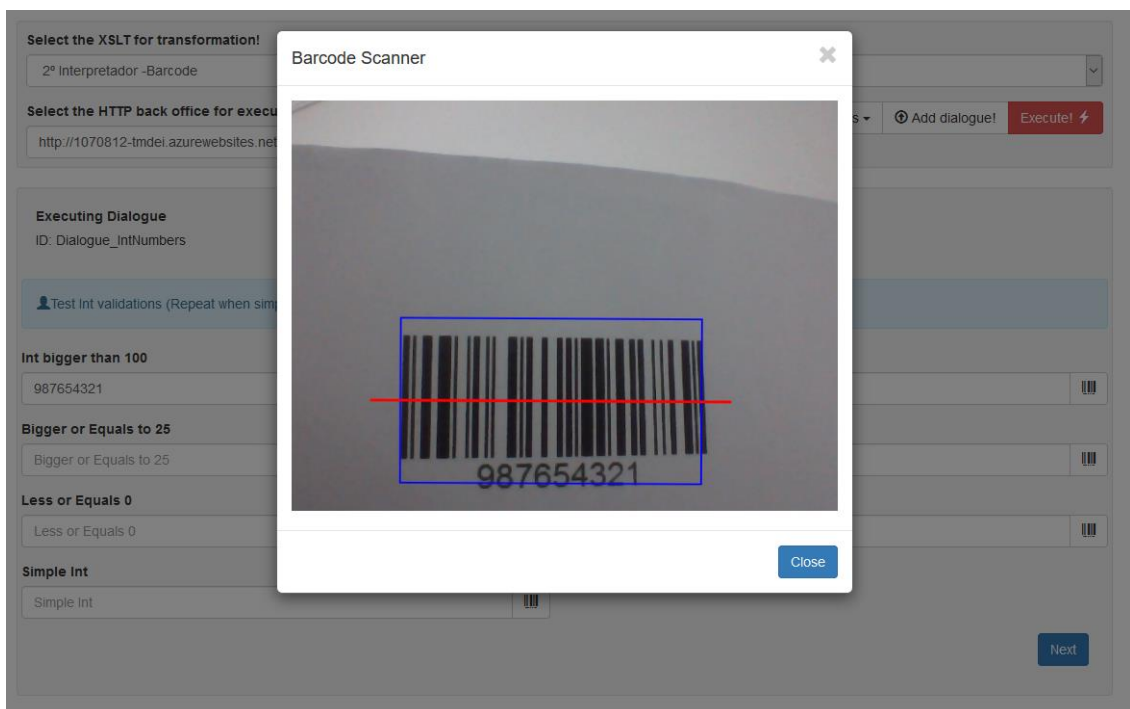


Figura 47 - Interpretador utilizando o método de aquisição código de barras

É possível testar o interpretador desenvolvido através do seguinte endereço: <http://1070812-tmdei.azurewebsites.net>

6.4 Sistema de Gestão de Informação

Para ser possível testar a integração do processo de aquisição com o processo de gestão de informação, desenvolveu-se um sistema numa arquitetura REST que simula-se um sistema de gestão. É importante realçar que dependendo do tipo de dados definido no pedido HTTP (XML ou JSON) este sistema retorna os dados nesse formato. Desta forma, é possível testar se o interpretador consegue interpretar as respostas e apresenta-las da forma correta.

A Tabela 22 apresenta os métodos desenvolvidos e qual a sua finalidade através de uma descrição.

Tabela 22 – Método da Sistema de Gestão

Descrição	Caminho relativo
Simula a validação de dados através do método POST	api/validations/PostVal
Simula a validação de dados através do método GET	api/validations/GetVal
Retorna uma lista de itens para ser usando pela resposta “Escolha” através do método GET	api/Select/Get
Retorna uma lista de itens para ser usada pela resposta “Escolha” através do método POST	api/Select/Post
Simula o armazenamento de dados remoto através do método GET	api/storage/get
Simula o armazenamento de dados remoto através do método POST	api/storage/post

7 Avaliação

Neste capítulo é apresentado uma avaliação dos resultados obtidos da implementação dos componentes, plataforma de modelação e interpretador. Encontra-se subdividida nas seguintes subsecções:

- **Diálogos de ensaio:** apresenta uma descrição de um grupo de diálogos propostos para testes.
- **Modelação de diálogos:** descreve se é ou não possível modelar os diálogos de ensaio usando a plataforma desenvolvida.
- **Execução de diálogos:** apresenta os resultados da execução dos diálogos de negócio com base num conjunto de especificações.
- **Resultados atingidos:** descreve sucintamente se os resultados foram atingidos com a implementação dos diferentes componentes.

7.1 Diálogos de Ensaio

Durante o processo de implementação foram desenvolvidos vários diálogos com diferentes complexidades, para se poder validar se é possível efetuar a sua modelação (plataforma de modelação) e posteriormente a sua execução (interpretador). A estes diálogos chamou-se de diálogos de ensaio.

As próximas tabelas apresentam a especificação dos diálogos de ensaio onde, são caracterizados da seguinte forma:

- **Identificação do diálogo:** nome do diálogo.
- **Complexidade:** grau de dificuldade quer de modelação como de execução (e.g. baixa, media, alta, muito alta, elevada).

- **Descrição do fluxo:** é descrito todos os passos do diálogo (e.g. tarefa de aquisição, validação, decisão, armazenamento).
- **Objetivos do diálogo:** correlaciona a descrição do fluxo com o seu objetivo.

Tabela 23 – Diálogo de Ensaio Nº 1

Identificação do diálogo	Nº 1
Complexidade	Baixa
Descrição do fluxo	1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador” com uma questão para cada tipo de dados (e.g. texto, inteiro, decimal, data, data e hora, verdadeiro ou falso). 2- No momento da recolha existe uma validação local para verificar se o tipo de dados corresponde ao desejado. Se a validação for positiva o fluxo termina, caso contrário deve solicitar a retificação dos dados.
Objetivo do diálogo	1- Recolher os dados dos diferentes tipos de dados. 2- Executar uma validação local conforme o tipo de dados.

Tabela 24 – Diálogo de Ensaio Nº 2

Identificação do diálogo	Nº 2
Complexidade	Média
Descrição do fluxo	1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve permitir selecionar um valor a partir de uma lista estática.
Objetivo do diálogo	1- Recolher o valor selecionado.

Tabela 25 – Diálogo de Ensaio Nº 3

Identificação do diálogo	Nº 3
Complexidade	Média
Descrição do fluxo	1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve permitir selecionar um valor a partir de uma lista dinâmica (pedido ao sistema de gestão).
Objetivo do diálogo	1- Recolher o valor selecionado.

Tabela 26 – Diálogo de Ensaio Nº 4

Identificação do diálogo	Nº 4
Complexidade	Média
Descrição do fluxo	1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro. 2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número inteiro. 3- Executa uma decisão através de uma validação condicional. Se o número recolhido é menor que 10 é executado de novo o ponto 1, caso contrário o processo de aquisição avança e o fluxo termina.
Objetivo do diálogo	1- Recolher o número.

	<p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executar uma decisão através de uma validação condicional.</p>
--	---

Tabela 27 – Diálogo de Ensaio Nº 5

Identificação do diálogo	Nº 5
Complexidade	Média
Descrição do fluxo	<p>1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro.</p> <p>2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número inteiro.</p> <p>3- No momento da recolha deve ser realizada uma validação remota. Se a validação for negativa o fluxo não deve avançar, caso contrário o fluxo termina.</p>
Objetivo do diálogo	<p>1- Recolher o número.</p> <p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executar uma validação remota.</p>

Tabela 28 – Diálogo de Ensaio Nº 6

Identificação do diálogo	Nº 6
Complexidade	Média
Descrição do fluxo	<p>1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro.</p> <p>2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número.</p> <p>3- Executada uma decisão através de uma expressão regular. Se é um número par é executado de novo o ponto 1, caso contrário o processo de aquisição avança e o fluxo termina.</p>
Objetivo do diálogo	<p>1- Recolher o número.</p> <p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executa uma decisão local através de uma expressão regular.</p>

Tabela 29 – Diálogo de Ensaio Nº 7

Identificação do diálogo	Nº 7
Complexidade	Média
Descrição do fluxo	<p>1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro.</p> <p>2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número.</p> <p>3- Executada uma decisão através de uma validação condicional. Se o número recolhido é menor que 10 é executado de novo o ponto 1, caso contrário o fluxo de aquisição avança.</p> <p>4- Executa uma tarefa de aquisição do tipo “Questões ao sistema”. Esta tarefa deve recolher a data e hora no momento da execução.</p> <p>5- Os dados recolhidos são guardados em armazenamento remoto e o fluxo termina.</p>

Objetivo do diálogo	<p>1- Recolher os dados dos diferentes tipos de dados.</p> <p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executar uma decisão através de uma validação condicional.</p> <p>4- Recolher dados do sistema.</p> <p>5- Armazenar dados remotamente.</p>
----------------------------	--

Tabela 30 – Diálogo de Ensaio Nº 8

Identificação do diálogo	Nº 8
Complexidade	Alta
Descrição do fluxo	<p>1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro.</p> <p>2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número.</p> <p>3- Executada uma decisão através de uma validação condicional. Se o número recolhido é menor que 10 é executado de novo o ponto 1, caso contrário o fluxo de aquisição avança.</p> <p>4- Executa uma tarefa de aquisição do tipo “Respostas estáticas”. O objetivo é recolher um valor pré-definido.</p> <p>5- Executa uma tarefa de aquisição do tipo “Questões ao sistema”. Esta tarefa deve recolher a data e hora no momento da execução e a identificação da UAD.</p> <p>6- Os dados recolhidos são guardados em armazenamento remoto e o fluxo termina.</p>
Objetivo do diálogo	<p>1- Recolher os dados dos diferentes tipos de dados.</p> <p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executa uma decisão local do tipo condicional.</p> <p>4- Recolher dados pré-definidos.</p> <p>5- Recolher dados do sistema.</p> <p>5- Armazenar dados remotamente.</p>

Tabela 31 – Diálogo de Ensaio Nº 9

Identificação do diálogo	Nº 9
Complexidade	Muito Alta
Descrição do fluxo	<p>1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro.</p> <p>2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número.</p> <p>3- Executada uma decisão através de uma validação condicional. Se o número for igual a 1 é executado o diálogo de ensaio nº1 de novo o ponto 1, caso contrário o fluxo de aquisição avança.</p> <p>4- Executa o diálogo de ensaio Nº 8. Após terminar a execução do diálogo é executado de novo o ponto 1.</p>
Objetivo do diálogo	<p>1- Recolher os dados dos diferentes tipos de dados.</p> <p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executa uma decisão local do tipo condicional.</p> <p>4- Executa um subdiálogo.</p>

Tabela 32 – Diálogo de Ensaio Nº 10

Identificação do diálogo	Nº 10
Complexidade	Elevada
Descrição do fluxo	<p>1- Executa uma tarefa de aquisição do tipo “Questões ao utilizador”. Esta tarefa deve recolher um número inteiro através da introdução e valor através da seleção de uma lista.</p> <p>2- No momento da recolha existe uma validação local para verificar se o valor recolhido é um número.</p> <p>3- Executada uma decisão através de uma validação condicional. Se o número for igual a 1 é executado o diálogo de ensaio nº 4, se o número for igual a 2 é executado o diálogo nº 3, caso contrário a decisão deve seguir o fluxo por defeito e executar de novo o ponto 1.</p>
Objetivo do diálogo	<p>1- Recolher os dados dos diferentes tipos de dados.</p> <p>2- Executar uma validação local para verificar o tipo de dados.</p> <p>3- Executa uma decisão local do tipo condicional para definir o próximo fluxo do diálogo.</p>

Nas próximas subsecções é possível validar se a plataforma de modelação consegue ou não modelar estes diálogos de ensaio com base na notação. Posteriormente, é verificado se estes diálogos de ensaio conseguem ser executados pelo interpretador e se o resultado da sua execução é o desejado.

7.2 Modelação de Diálogos

Esta subsecção destina-se a validar se foi ou não possível modelar através da plataforma de modelação os diálogos de ensaio.

Usando a plataforma de modelação foi possível modelar todos os diálogos de negócio. A Tabela 33 através da identificação do diálogo de ensaio, efetua uma correlação do tempo despendido durante o processo de modelação com base na complexidade do diálogo.

Tabela 33 – Complexidade do Diálogo Vs. Tempo de Modelação

Identificação do diálogo de ensaio	Complexidade	Tempo despendido no processo de modelação
Nº 1	Baixa	Menos de 90 segundos
Nº 2	Baixa	Menos de 60 segundos
Nº 3	Média	Menos de 80 segundos
Nº 4	Média	Menos de 100 segundos
Nº 5	Média	Menos de 90 segundos
Nº 6	Média	Menos de 100 segundos
Nº 7	Alta	Menos de 150 segundos
Nº 8	Alta	Menos de 150 segundos
Nº 9	Muito Alta	Menos de 60 segundos
Nº 10	Muito Alta	Menos de 75 segundos

Os diálogos de ensaio nº 9 e 10, apesar de serem os mais complexos, a sua modelação é relativamente rápida, uma vez que os subdiálogos já estão modelados e são incluídos no processo de aquisição rapidamente.

A Figura 48 apresenta o diálogo de negócio (XML) resultante do processo de modelação, referente ao diálogo de ensaio nº 3. Optou-se por colocar a figura deste diálogo para ser possível apresentar toda especificação numa só página. Nesta figura é possível visualizar que a plataforma de modelação especifica o diálogo com base na notação de especificações de diálogos.

É possível consultar todos os diálogos de negócio (XML) e até altera-los através da plataforma de modelação no seguinte endereço: <http://1070812-tmdei.azurewebsites.net>

```

<?xml version="1.0" encoding="UTF-8"?>
<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
  xmlns:dc="http://www.omg.org/spec/DD/20100524/DC" xmlns:di="http://www.omg.org/spec/DD/20100524/DI"
  xmlns:tmdei="http://tmdei" id="sample-diagram" targetNamespace="http://bpmn.io/schema/bpmn"
  xsi:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd">
<tmdei:dialogue id="Dialogo_SelecionarValor_Lista_Estatica" isExecutable="false" lang="pt-PT">
  <bpmn2:startEvent id="StartEvent_1">
    <bpmn2:outgoing>SequenceFlow_05v5sy4</bpmn2:outgoing>
  </bpmn2:startEvent>
  <bpmn2:sequenceFlow id="SequenceFlow_05v5sy4" sourceRef="StartEvent_1" targetRef="Task_04cx6f8" />
  <tmdei:userQuestions id="Task_04cx6f8">
    <bpmn2:incoming>SequenceFlow_05v5sy4</bpmn2:incoming>
    <bpmn2:outgoing>SequenceFlow_0crucm7</bpmn2:outgoing>
    <tmdei:questions>
      <tmdei:question id="valorSelecionado" label="Selecionar Valor" type="user">
        <tmdei:response type="select" collectionMethod="default">
          <tmdei:selectResponse type="request" dataType="string" valueRef="id" nameRef="text">
            <tmdei:request type="http" relativePath="api/Select/Get" method="get" dataType="xml" xpathExpression="//SelectedItem" />
          </tmdei:selectResponse>
        </tmdei:response>
      </tmdei:question>
    </tmdei:questions>
  </tmdei:userQuestions>
  <bpmn2:endEvent id="EndEvent_1w4tem7">
    <bpmn2:incoming>SequenceFlow_0crucm7</bpmn2:incoming>
  </bpmn2:endEvent>
  <bpmn2:sequenceFlow id="SequenceFlow_0crucm7" sourceRef="Task_04cx6f8" targetRef="EndEvent_1w4tem7" />
</tmdei:dialogue>

```

Figura 48 – XML do Diálogo de Ensaio N°3

7.3 Execução de Diálogos

O objetivo desta subsecção é avaliar se é ou não possível executar os diálogos de ensaio através do interpretador desenvolvido. Para alcançar esse objetivo é necessário comprovar que o interpretador consegue realizar com sucesso todas as tarefas de aquisição.

Esta subsecção encontra-se subdividida da seguinte forma:

- Testes de Validação aos Dados;
- Testes de Armazenamento de Dados;
- Teste de Execução do Fluxo de Aquisição.

7.3.1 Testes de Validação aos Dados

No processo de aquisição de dados é necessário realizar validações aos dados recolhidos. Para se testar estas validações modelou-se diálogos de negócio para satisfazer os seguintes tipos de validações:

- **Validações locais:** as validações locais devem assegurar que os dados recolhidos satisfazem o tipo de dados desejados (e.g. número inteiro, decimal, data, data e hora, etc.) ou, que cumprem uma determinada restrição (e.g. maior quê, igual a, menor, expressão regular, etc.). Para validar que o interpretador realizava corretamente as validações locais, criou-se os diferentes diálogos de negócio com os seguintes objetivos:
 1. **Validações pelo tipo de dados:** criou-se diálogos de ensaio (e.g. nº 1) com o objetivo que no momento de introdução dos dados, estes deviam ser validados conforme o tipo de dados. Os resultados da execução destes diálogos foi o esperado.
 2. **Validações através de restrição condicional:** foi executado diálogos com todas as possibilidades de restrições condicionais (e.g. nº 6, nº 7, etc) e o resultado da validação foi o esperado.
 3. **Validações através de uma expressão regular:** foi criado restrições aos dados que deveriam ser recolhidos em diferentes diálogo de ensaio (e.g. nº 6) através de expressões regulares e no momento da validação, os dados foram validados conforme esperado.
- **Validações remotas:** criou-se diversos diálogos de ensaio (e.g. nº 3, nº 5, nº 7, etc.) com as diferentes configurações de validações remotas (e.g. HTTP GET, HTTP POST).

No momento de execução deste diálogo, como esperado, existe uma comunicação ao sistema de gestão, que valida os dados recolhidos.

A Figura 49 apresenta o interface do processo de aquisição após serem executadas validações.

The screenshot shows a dialog box titled "Executing Dialogue" with the ID "Dialogue_Test_inputTypes". Below the title bar, there is a header "Test inputTypes". The main area contains several input fields with their respective data types and validation messages:

- String:** Input field contains "String". Validation message: "This field is required."
- Integer:** Input field contains "sas". Validation message: "The value (sas) must be a integer."
- Datetime:** Input field contains "25/10/2017 16:56:48".
- Decimal:** Input field contains "asa". Validation message: "The value (asa) must be a decimal."
- Boolean:** Input field contains "Boolean?".
- Date:** Input field contains "19/10/2017".

A "Next" button is located at the bottom right of the dialog box.

Figura 49 – Resultado das Validações na UAD

7.3.2 Testes de Armazenamento de Dados

O objetivo da execução de um diálogo de negócio é recolher determinadas informações. Após esta recolha é necessário armazenar esses dados.

Foram realizados diálogos de ensaio para testar se o interpretador consegue guardar os dados recolhidos nos seguintes tipos de armazenamento:

- **Armazenamento local:** os diálogos de ensaio que foram criados com este propósito foram o N.º 4 e N.º 6. Em ambos os cenários foi possível guardar os dados.
- **Armazenamento remoto:** os diálogos de ensaio que foram criados com este propósito foram o N.º 7 e N.º 8. Em ambos os cenários foi possível delegar a responsabilidade de armazenamento ao sistema de gestão.

7.3.3 Teste de Execução do Fluxo de Aquisição

Este teste é o mais completo, abordando todas as tarefas possíveis do processo de aquisição.

Como prova que os dados são recolhidos pela sequência definida e pelo tipo de tarefa de aquisição, modelou-se o diálogo de ensaio n.º 7 representado na Figura 50 através da

plataforma de modelação. Nesta figura, colou-se os números para uma mais fácil compreensão da ordem de execução do fluxo de aquisição. A ordem de execução deste diálogo é a seguinte:

1. **Início do diálogo:** ponto inicial do diálogo.
2. **Tarefa de aquisição “Questões ao utilizador”:** esta tarefa espera que o utilizador introduza um número inteiro. Nesta tarefa são realizadas as validações sobre os dados recolhidos.
3. **Processo de decisão:** este processo de decisão vai validar se o número recolhido na tarefa de aquisição é menor que dez.
4. **Sequência do fluxo por defeito:** caso número recolhido menor que dez é repetido a tarefa de aquisição.
5. **Sequência do fluxo em caso de sucesso na validação do processo decisão:** caso o número recolhido seja maior ou igual que dez, a sequência do fluxo passa para a tarefa de aquisição seguinte.
6. **Tarefa de aquisição “Respostas estáticas”:** nesta tarefa o objetivo é obter a seguinte informação, já pré-configurada, uma *string* com o seguinte valor: “teste”.
7. **Tarefa de aquisição “Questões ao sistema”:** nesta tarefa de aquisição pretende-se obter as seguintes informações do sistema: 1) data e hora da execução do diálogo e 2) identificação da UAD.
8. **Armazenamento de dados:** na execução do armazenamento neste diálogo, os dados recolhidos devem ser enviados para o sistema de gestão.
9. **Fim do diálogo:** define o fim da execução do diálogo.

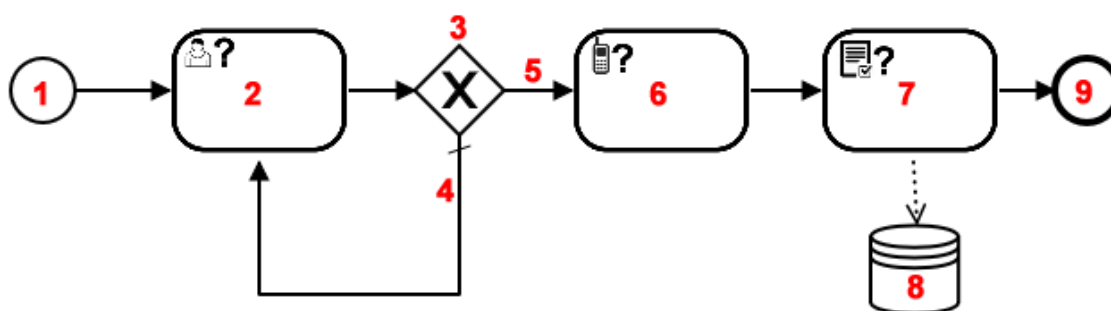


Figura 50 – Diálogo de Ensaio Nº 7

Executou-se este diálogo através do interpretador desenvolvido e conseguiu-se efetuar o processo de aquisição efetuando os passos enumerados. A Figura 50 através do formato JSON, apresenta os dados recolhidos após a execução do diálogo de ensaio nº 7. Nesta figura é possível visualizar que a ordem de recolha dos dados seguiu a que foi configurada. Ainda, é possível visualizar que a tarefa de aquisição do tipo questões ao utilizador (*ID_UserQuestions*)

foi executada 3 vezes, porque nas primeiras duas não foi possível satisfazer a condição definida no processo de decisão.

```
{
  "ID_UserQuestions":
  [
    [{"name": "numero", "value": "1"}],
    [{"name": "numero", "value": "2"}],
    [{"name": "numero", "value": "12"}]
  ],
  "ID_StaticResponses":
  [
    [{"name": "responses_11p0hrl", "value": "staticValue"}]
  ],
  "ID_SystemQuestions":
  [
    {"name": "sysDateTime", "value": "19/10/2017 16:21:16"},
    {"name": "terminalID", "value": "terminalID-2017"}]
  ]
}
```

Figura 51 – Dados Armazenados em Formato JSON

É possível executar os diálogos de ensaios e outros, usando o interpretador desenvolvido através do seguinte endereço: <http://1070812-tmdei.azurewebsites.net>

7.4 Resultados Atingidos

Este capítulo tem como objetivo avaliar a solução proposta. Para se alcançar esse objetivo foram descritos diversos diálogos de negócio, denominados de diálogos de ensaio, com o intuito de verificar se era ou não possível modelá-los através da plataforma e posteriormente executá-los através do interpretador desenvolvido.

Foi comprovado que é possível modelar todos os diálogos de ensaio através da plataforma desenvolvida e que, os tempos despendidos na modelação destes diálogos são relativamente baixos. Apesar de não serem apresentados elementos de comparação, os tempos despendidos pelos sistemas atuais é superior aos realizados pela plataforma. Pode-se concluir que esta proposta de solução é mais rápida do que os sistemas atuais, quando existe a necessidade de criar ou alterar um diálogo de negócio. A execução de todos os diálogos de ensaio é possível através do interpretador implementado.

Os testes realizados neste capítulo, em ambos os componentes, plataforma de modelação e interpretador, comprovam a viabilidade da solução proposta.

8 Conclusão

Neste capítulo é apresentado um resumo do trabalho realizado, quais os objetivos alcançados, algumas das necessidades de trabalho futuro e por fim, são apresentadas as considerações finais.

8.1 Resumo

Atualmente existem diversos sistemas de informação que permitem recolher dados de forma automática. Neste trabalho é apresentado um estudo das seguintes áreas de negócio: 1) Controlo e Gestão de Tempos de Presença, 2) Controlo de Acessos e 3) Controlo e Gestão dos Tempos de Produção e Qualidade e dos sistemas de informação existentes.

Os sistemas existentes em ambas as áreas, sempre que existe a necessidade de se criar ou alterar um diálogo de negócio, é obrigatório realizar um esforço de implementação ou reimplementação imperativa. Esta implementação apenas pode ser efetuada por profissionais com conhecimentos tecnológicos e conceptuais no sistema em questão. Conclui-se que os problemas dos sistemas atuais são essencialmente os seguintes:

- Corresponder as necessidades de um projeto ou cliente;
- Alta dependência entre as UADs e os sistemas de gestão;
- Obrigam a efetuar alterações quer nas UADs como nos sistemas de gestão sempre que o objetivo da recolha de dados mudar;
- Apenas profissionais qualificados e com conhecimento nos sistemas em questão conseguem criar ou alterar diálogos de negócio;
- Custos de implementação/adaptação elevados.

Com base nos problemas dos sistemas atuais, é apresentada uma visão da solução que consiste em dividir o funcionamento dos sistemas atuais nas seguintes 3 partes distintas:

- **Processo de Modelação:** este processo deve permitir modelar diálogos de negócio através de uma ferramenta de modelação. Este processo é constituído por dois componentes, um é notação de especificação de diálogos, o segundo é a plataforma de modelação que possibilita modelar diálogos com base na notação.
- **Processo de Aquisição:** após ser modelado um diálogo de negócio deve ser utilizado com base na sua semântica no processo de aquisição, de forma que seja possível recolher os dados pretendidos. Este processo é constituído pelos seguintes dois componentes: 1) interpretador, possibilita instanciar diálogos de negócio numa UAD e, 2) UAD que permite uma interação homem-máquina com o objetivo de se recolher os dados pretendidos.
- **Processo de Gestão de Informação:** neste processo enquadram-se os sistemas de gestão existentes.

Os objetivos que se alcançou com esta divisões dos sistemas atuais são:

- **Redução de custos:** o processo de modelação deixa de estar limitado que apenas profissionais qualificados e com o conhecimento no sistema em questão, consigam criar ou alterar diálogos de negócio.
- **Simplificação:** possibilita que pessoas com conhecimento da área de negócio em questão consigam modelar os diálogos desejados.
- **Redução de tempo:** a plataforma permite agilizar o processo de modelação, reduzindo substancialmente o tempo de criação ou adaptação de um diálogo de negócio.
- **Redução de recursos:** permite a redução de recursos humanos (profissionais qualificados).
- **Maior adaptabilidade:** com a utilização do conceito interpretador, é possível utilizar o mesmo diálogo de negócio em diferentes UADs.

No capítulo 4 é descrito uma análise e requisitos necessários para o funcionamento da proposta de solução. Com base nesta análise e requisitos, o capítulo 5 apresenta uma proposta de *design* para resolver o problema dos sistemas atuais.

Como validação da proposta de *design*, o capítulo 6 apresenta a implementação dos diferentes componentes, notação de modelação, plataforma de modelação e interpretador. Para o processo de modelação, mais concretamente para o componente notação é desenvolvido uma notação de especificação de diálogos através da extensão do BPMN. Já a plataforma de modelação é implementada através da extensão da ferramenta BPMN.IO.

Ainda, existiu a necessidade de se desenvolver um sistema numa arquitetura REST, que simula um sistema de gestão (*backoffice*). Este desenvolvimento possibilitou comprovar e avaliar o processo de comunicação entre o interpretador e este sistema de gestão.

Finalmente como forma de avaliar os protótipos desenvolvidos foram especificados diálogos de ensaio, com o objetivo de validar se é ou não possível modela-los usando a plataforma e posteriormente executa-los usando o interpretador.

8.2 Objetivos Alcançados

Os objetivos inicialmente propostos e descritos na subsecção 1.3 foram de uma forma geral alcançados. No entanto, apenas foi contruído um interpretador que utiliza como UAD os navegadores web, não sendo possível implementar este componente com nenhuma das UAD utilizada nos sistemas atuais.

8.3 Trabalho Futuro

A plataforma de modelação permite apenas modelar a comunicação com o sistema de gestão através do protocolo HTTP, limitando a interação do processo de aquisição com o de gestão a este protocolo. Para ser possível utilizar esta plataforma num sistema existente é necessário realizar uma das seguintes tarefas:

1. Estender a notação de especificação de diálogos ao protocolo do sistema e consequente adaptar a extensão da plataforma.
2. Criar uma camada nos sistemas de gestão atuais que possibilite a comunicação através do protocolo HTTP.

O trabalho realizado nesta dissertação focou-se apenas na implementação de um interpretador que utiliza navegadores web como UAD. Não foi possível efetuar a implementação de um interpretador para uma UAD usada nos sistemas atuais. Assim, como trabalho futuro é necessário desenvolver um interpretador para validar se é possível usar este componente numa UAD usada pelos sistemas atuais.

8.4 Considerações Finais

A recolha de informações sobre o funcionamento dos sistemas atuais não foi fácil, uma vez que, após vários contactos com as empresas nacionais que comercializam estes sistemas, não foi facultado qualquer documentação. No entanto, como ponto de partida para o desenvolvimento deste trabalho haviam disponíveis ficheiros técnicos de sistemas atuais (manual de utilizador), um da Elo (Elo Sistemas de Informação, 2014) e outro da Innux (“Innux Technologies, Lda,” 2016g).

A realização deste projeto foi muito gratificante, possibilitou-me consolidar e adquirir novos conhecimentos quer nas áreas de negócio, assim como sobre a notação BPMN e as ferramentas de modelação existentes para o BPMN.

Compreendi, que atualmente estão a ser desenvolvidos diversos sistemas de informação com base na gestão de processos de negócio (BPM) onde, em muitos deles é usado a notação BPMN para se poder representar os respetivos processos.

Em suma, a realização deste trabalho deixa-me satisfeito com a realização de todos os objetivos pré-definidos e dos conhecimentos adquiridos.

Referências

- ABPMP, 2011. Guide to the Business Process Management Body of Knowledge (BPM CBOK®).
- Activiti [WWW Document], 2016. URL <https://www.activiti.org>
- Apple, 2016. iOS [WWW Document]. URL <https://developer.apple.com/ios/>
- BPMN.IO [WWW Document], 2016. URL <http://bpmn.io/>
- Brack, F., 2017. js-cookie.
- Camunda Services GmbH, 2017. bpmn-js-properties-panel [WWW Document]. URL <https://github.com/bpmn-io/bpmn-js-properties-panel>
- Dahl, R., 2017. Node.js.
- Elo - Sistemas De Informação, Lda [WWW Document], 2016. URL <http://www.elo-si.com>
- Elo Sistemas de Informação, 2016a. TIME HR [WWW Document]. URL <http://elo-si.com/pt/assiduidade/timehr>
- Elo Sistemas de Informação, 2016b. Task Manager [WWW Document]. URL <http://elo-si.com/pt/atividades/taskmanager>
- Elo Sistemas de Informação, 2014. Descrição e características da ferramenta “Gerador de Diálogos.”
- Eriksson, H.-E., Penker, M., 2000. Business Modeling with UML: Business Patterns at Work.
- Flynn, M., 2003. Idea management for organizational innovation. Int. J. Innov. Manag.
- Gamma, E., Vlissides, J., Johnson, R., Helm, R., 1994. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- Gaudenz, A., 2017. DRAW.IO [WWW Document]. URL www.draw.io
- GoJS [WWW Document], 2016. URL <http://gojs.net>
- Google Inc., 2016. Android [WWW Document]. URL <https://www.android.com/>
- Grady, R., 1992. Practical Software Metrics for Project Management and Process Improvement.
- HELFO [WWW Document], 2017. URL www.heflo.com

IDONIC, 2017. URL <http://www.idonic.com/2015/06/26/picar-o-ponto-registo-da-assiduidade-vantagens/>

Innux Technologies, Lda, 2016a. NuxBio IV [WWW Document]. URL <http://innux.com/pt/equipamentos/relogios-ponto/nuxbio-iv.html>

Innux Technologies, Lda, 2016b. InuuxTime [WWW Document]. URL <http://innux.com/pt/software/gestao-assiduidade/innuxtime.html>

Innux Technologies, Lda, 2016c. NuxBio IV [WWW Document]. URL <http://innux.com/pt/equipamentos/relogios-ponto/nuxbio-iv.html>

Innux Technologies, Lda, 2016d. InnuxAccess [WWW Document].

Innux Technologies, Lda, 2016e. NuxFinger [WWW Document]. URL <http://www.innux.com/equipamentos/relogios-ponto/nuxfinger.html>

Innux Technologies, Lda, 2016f. NuxIn II [WWW Document]. URL <http://www.innux.com/equipamentos/relogios-ponto/nuxinii.html>

Innux Technologies, Lda [WWW Document], 2016g. URL <http://innux.com/>

ISGUS, 2016a. ZEUS® Time & Attendance [WWW Document]. URL <http://www.isgus.de/eng/solutions/time-attendance/time-attendance.html>

ISGUS, 2016b. ZEUS® Plant Data Collection solution [WWW Document]. URL <http://www.isgus.de/eng/solutions/plant-data-collection/plant-data-collection.html>

ISGUS, 2016c. IT 8001.

ISGUS, 2016d. IT 8200 FP [WWW Document]. URL http://www.isgus.com/cms/upload/downloads/daten/Data_sheet_IT_8200_FP_1505.pdf

ISGUS, 2016e. IT 4100 [WWW Document]. URL http://www.isgus.de/cms/upload/downloads/daten/Datenblatt_IT_4100_IT_4110_1510.pdf

ISGUS, 2016f. Terminal IT 900.

ISGUS [WWW Document], 2016g. URL <http://www.isgus.de>

James, O., Marakas, G.M., 2011. Management Information Systems.

JOINTJS [WWW Document], 2017. URL <http://resources.jointjs.com/demos/bpmn>

JQuery, 2017. JQuery.

KABA, 2016a. Terminal B-web 93 00.

KABA, 2016b. Reader 9110 [WWW Document]. URL www.kaba.com/workforce-management/media/190780/v18/File/kaba-compact-reader-91-10.pdf

KABA [WWW Document], 2016c. URL <http://www.kaba.com>

Koen, P., Ajamian, G., Boyce, S., Clamen, A., Fisher, E., Fountoulakis, S., Johnson, A., Puri, P., Seibert, R., 2002. Fuzzy Front End: Effective Methods, Tools, and Techniques.

Koen, P., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'Amore, R., Elkins, C., Herald, K., Incorvia, M., Johnson, A., Karol, R., Seibert, R., Slavejkov, A., Wagner, K., 2001. PROVIDING CLARITY AND A COMMON LANGUAGE TO THE "FUZZY FRONT END."

Kotler, P., 1998. Marketing Management Analysis, Planning, Implementation and Control.

Kubica, M., 2017. xml2js.

Microsoft Corporation, 2016. Windows Phone [WWW Document]. URL <https://www.microsoft.com/pt-pt/windows/phones>

Modelio [WWW Document], 2016. URL www.modelio.org

Murray, J., 2017. xslt.

Nicola, S., 2016. Análise de Valor de Negócio. ISEP Instituto Superior de Engenharia do Porto.

Nicola, S., Ferreira, E.P., Pinto Ferreira, J.J., 2012. . Int. J. Inf. Technol. Decis. Mak.

Oberhofer, C., 2017. QuaggaJS.

OMG, 2017. SysML.

OMG, 2011. Business Process Model and Notation (BPMN).

OSTERWALDER, A., PIGNEUR, Y., 2010. Business Model Generation. Projecttime [WWW Document], 2016. URL <http://www.projecttime.pt>

Smartstep [WWW Document], 2016. URL <http://www.smartstep.pt/>

Smith, P.G., Reinertsen, D.G., 1991. Developing products in half the time: new rules, new tools.

SSE, 2017. GEST – Controlo e Gestão [WWW Document]. URL <http://sse.pt/gest-controlo-e-gestao/>

Staab, M., 2017. jquery-validation.

STILC, Lda. [WWW Document], 2016. URL <http://www.stilc.pt>

Suprema, 2016a. BioStar 2 [WWW Document]. URL <https://supremainc.com>

Suprema, 2016b. BioStar 2 SDK [WWW Document]. URL <http://kb.supremainc.com/bs2sdk/doku.php?id=en:start>

Suprema, 2016c. BioStar 2 API [WWW Document]. URL <https://www.supremainc.com/en/AccessControl-TimeandAttendance/Platform/BioStar-2-API>

Suprema, 2016d. BioStation A2 [WWW Document]. URL <https://supremainc.com/en/AccessControl-TimeandAttendance/Biometric-Devices/BioStation-A2>

Suprema, 2016e. BioStation 2 [WWW Document]. URL <https://www.supremainc.com/en/AccessControl-TimeandAttendance/Biometric-Devices/BioStation-2>

Suprema, 2016f. Xpass S2 [WWW Document]. URL <https://www.supremainc.com/en/AccessControl-TimeandAttendance/RFID-Devices/Xpass-S2>

Suprema [WWW Document], 2016g. URL <https://www.supremainc.com>

SYDLE BPM, 2017.

X64, 2016a. X64 [WWW Document]. URL <http://x64.com>

X64, 2016b. XRT707 [WWW Document]. URL http://www.x64.com/index.php?option=com_content&view=article&id=120&Itemid=139

Anexos

A.1 Notação BPMN

O Business Process Model and Notation (BPMN) (OMG, 2011) é um padrão que foi criado com o objetivo de modelar processos de negócio. Foi inicialmente criado pela Business Process Management Initiative (BPMI), mas atualmente é gerido pela Object Management Group (OMG) devido ao facto destas duas organizações se terem fundido em 2005.

O principal objetivo do BPMN é fornecer uma notação que seja facilmente lida e percebida por todos os intervenientes nos processos de negócios, quer por pessoal técnico quer por analistas, gestores e utilizadores, facultando uma notação intuitiva e ao mesmo tempo suficientemente completa e expressiva para a representação dos processos.

O BPMN cria uma ponte padronizada para colmatar a lacuna existente entre o processo de *design* e o processo de implementação. Outro objetivo era garantir uma linguagem definida em XML que possibilita-se execução de processos de negócios, como por exemplo a WSBPPEL (Web Services Business Process Execution Language) (OMG, 2011).

O BPMN representa o resultado de um acordo entre vários desenvolvedores de ferramentas de modelação, que tinham as suas próprias notações, em utilizar uma única notação para o benefício da compreensão do utilizador final.

A utilização do BPMN obriga a utilização de um conjunto de regras e notações gráficas para modelarmos os processos de negócio.

Existem cinco categorias base de elementos gráficos para a representação de processo de negócio. As categorias base são as seguintes:

1. **Objetos de Fluxo:** eventos, atividades e *gateways*.

2. **Dados:** objetos de dados, entrada de dados, saída de dados e repositório de dados.
3. **Objetos de Ligação:** fluxo de sequência, fluxo de mensagem, associação e associação de dados.
4. **Swimlanes:** pool e a lane.
5. **Artefactos:** grupo e anotação.



8.4.1 Objetos de Fluxo


Os objetos de fluxo são definidos como os principais elementos gráficos para definir o comportamento de um processo de negócio. Existem os seguintes três tipos de objetos de fluxo: 1) Eventos, 2) Atividades e 3) Gateways.

Eventos

Um evento é algo que acontece, são representados por círculo durante a realização de um processo de negócio. Os eventos afetam o fluxo de processos, pois normalmente tem uma causa ou impacto. Existem três tipos de eventos, baseados na fase em que afetam o fluxo de processos: início, intermédio e fim.





Tabela 34 – BPMN Objetos de Fluxo – Tipos de Eventos



Eventos		
Elemento	Descrição	Notação
Evento Inicial	Indicam onde e como o processo inicia. Existem 7 tipos de eventos iniciais (ver Tabela 35). Os eventos de alto nível podem ser iniciados em qualquer tipo de evento.	
Evento Intermédio	É um evento que ocorre entre o evento de início e o evento de fim. São utilizados para se especificar com mais exatidão os factos que acontecem durante o processo. Existem os seguintes dois tipos de eventos intermédios: 1) Eventos intermédios num fluxo normal de um processo, que respondem a dois	

	<p>objetivos, para receber (recetor) ou para emitir (emissor), como apresenta a Tabela 37.</p> <p>2) Eventos intermédios anexados a uma atividade. São usados para receber.</p>	
Evento Final	Indicam onde o processo termina.	

Os eventos de alto nível podem ser iniciados em qualquer tipo de evento. Existem 7 tipos de eventos iniciais de alto nível, como descreve a Tabela 35.








Tabela 35 – Tipos de Eventos Iniciais de Alto Nível

Eventos Iniciais		
Elemento	Descrição	Notação
Nenhum	Indicam onde o processo inicia. Utiliza-se quando o início de um processo é indefinido.	
Mensagem	Este evento acontece quando uma mensagem chega de outro interveniente e inicia o processo. Uma mensagem é uma comunicação direta entre dois intervenientes do negócio.	
Temporal (<i>Timer</i>)	O evento é iniciado num determinado ciclo de tempo, ou numa data específica (por exemplo, às 9 horas de sexta-feira).	
Condicional	Este evento acontece quando uma condição se torna verdadeira.	
Sinal	Este tipo de evento é iniciado quando se recebe sinais de outro processo. Um sinal não é uma mensagem, pois não se encaminha a nenhum destino específico.	

Múltiplo	Indica múltiplos caminhos para se iniciar um processo, só um deles é obrigatório para iniciar o processo.	
Paralelo Múltiplo	Indica múltiplos caminhos para se iniciar um processo, onde vários são obrigatórios para iniciar o processo.	

Os eventos de subprocessos podem ser muito úteis para reunir partes de fluxos que podem depois ser reutilizados noutros processos, aumentando o reuso. Este tipo de eventos são classificados em dois tipos: 1) eventos intermédio interruptores, que são aqueles que quando ocorrem provocam a interrupção do subprocesso e 2) eventos intermédios não interruptores, que quando ocorrem não provocam a interrupção do subprocesso. A próxima tabela, Tabela 36, apresenta e descreve os tipos de eventos de subprocessos.








Tabela 36 – Tipos de Eventos de Subprocessos

Eventos de Subprocessos		
Elemento	Descrição	Notação
Nenhum	Quando o início é indefinido.	
Mensagem	Este evento acontece quando uma mensagem chega de outro interveniente e inicia o processo. Se o evento é o interruptor o subprocesso terminara a sua execução.	  interruptor não interruptor
Temporal (Timer)	O evento é iniciado num determinado ciclo de tempo, ou numa data específica (por exemplo, às 9 horas de sexta-feira).	  interruptor não interruptor
Condicional	A utilização do evento interruptor permite terminar o subprocesso	 

	através da definição de uma condição. Existe também o não interruptor.	interruptor	não interruptor
Sinal	Este tipo de evento é iniciado quando se recebe sinais de outro processo		
Múltiplo	Indica múltiplos caminhos para se iniciar um processo, só um deles é obrigatório para iniciar o processo.		
Erro	Este tipo de evento inicia quando a ocorrência de erros (exceções). Só pode ser usado anexado ao subprocesso.		
Compensação	Utilizado para definir um comportamento para desfazer atividades em caso de um subprocesso ser cancelado ou necessitar de ser retomado.		

Os eventos intermédios assinalam que no decorrer do processo acontecerá uma ocorrência prevista. Este tipo de eventos podem ser de dois tipos: 1) Emissor, geram uma ocorrência de um facto e dão continuidade ao processo e 2) Recetor, esperam que ocorra uma determinada ocorrência no processo para que o processo continue. Existem duas formas de utilizar os eventos intermédios, 1) evento intermédio que são colocados dentro do fluxo do processo e 2) evento intermedio que são anexados à atividade, que apenas podem ser utilizados como recetor.

Tabela 37 - Tipos de Eventos Intermediário

Eventos de Intermédios			
Elemento	Descrição	Notação	
		Emissor	Recetor
Nenhum	Quando o início não é definido.		
Mensagem	Pode ser usado para enviar ou receber mensagens entre as entidades. Uma mensagem chega de uma entidade e dispara o evento.		
Temporal (Timer)	Atua como um mecanismo de atraso com base numa data, por exemplo às 9 horas de sexta-feira.		
Escalation	É um evento emissor. Este evento é usado somente dentro de um subprocesso. Uma escalada ocorre quando alguém em um nível mais alto de responsabilidade dentro da organização torna-se envolvido em um processo.		
Compensação	No fluxo normal, este evento intermediário indica que uma compensação é necessária. Utilizado para definir um comportamento para desfazer atividades em caso de um subprocesso ser cancelado ou necessitar de ser retomado.		
Condicional	Uma certa condição deve ser satisfeita para ativar este evento.		








Ligação	Utilizado para direcionar o fluxo para outra atividade.		
Sinal	O evento é ativado quando se envia ou se recebe um sinal. Um sinal comunica entre diferentes processos. Um sinal não se destina a nenhum interveniente, apenas os recetores do sinal é que o capturam.		
Múltiplo	Indica múltiplos caminhos para se iniciar um processo, só um deles é obrigatório para iniciar o processo.		
Paralelo Múltiplo	Indica múltiplos caminhos para se iniciar um processo, onde vários são obrigatórios para iniciar o processo.		

Tabela 38 – Tipos de Eventos Finais

Eventos de Finais		
Elemento	Descrição	Notação
Nenhum	Nenhum resultado é definido. Representa o fim normal de um processo.	
Mensagem	Significa que o processo é terminado com o envio de uma mensagem para um interveniente.	
Erro	Define que o processo foi terminado com um erro.	
Escalation	Uma escalada ocorre quando alguém em um nível mais alto de responsabilidade dentro da organização torna-se envolvido em um processo.	
Cancelar	Permite cancelar o processo em questão e aciona um processo intermédio.	
Compensação	Este tipo de evento defini que o final do processo precisa de uma compensação.	
Sinal	Este tipo de sinal determina que um sinal será enviado no fim do processo.	
Terminal	Indica que todas as atividades no processo terminam imediatamente.	
Múltiplo	Representa dois ou mais resultados de eventos finais possíveis.	

Atividades

Uma atividade é um termo genérico que permite representar o trabalho que se efetua numa organização. É representada por um retângulo com os cantos arredondados. Uma atividade

pode ser atômica ou não atômica (composta). Os tipos de atividades existentes são: tarefas e subprocessos.

Tabela 39 – Tipos de Atividades no BPMN


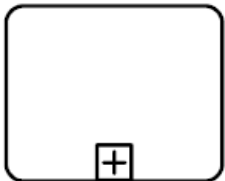
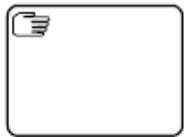



Tipos de Atividades		
Elemento	Descrição	Notação
Tarefa	A tarefa é a atividade de trabalho atômica. Ela representa uma ação no processo que pode ser executada por uma pessoa ou um sistema.	
Subprocesso	Um subprocesso é um processo que está incluído dentro de um outro processo. O símbolo mais (+), permite visualizar ou ocultar o subprocesso. Os subprocessos podem ser reutilizáveis.	

Tabela 40 – Tipos de Tarefas

Tipos de Tarefas		
Tipo	Descrição	Notação
Nenhum		
Manual	É uma tarefa que é esperada que seja executada sem o suporte de nenhuma aplicação de execução de processos de negócio ou outra aplicação.	
Recção	É uma tarefa simples que espera por uma mensagem de um participante externo.	
Script	É executada pelo motor de processos de negócio. O modelador ou implementador define um script em uma linguagem que o motor pode interpretar e executar.	
Envio	É uma tarefa simples que é projetada para enviar uma mensagem a um participante externo.	




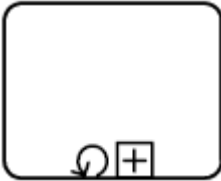
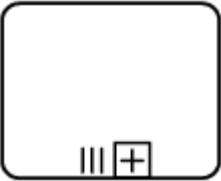
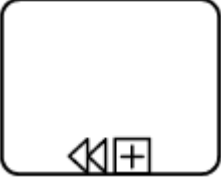
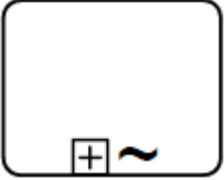
Serviço	Fornecer algum tipo de serviço, pode ser um serviço <i>web</i> ou uma aplicação automatizada.	
Utilizador	Um ator humano desempenha a tarefa com a assistência de uma aplicação de <i>software</i> .	
Regra de Negócio	Proporciona um mecanismo para o processo para enviar informações a um Business Rules Engine (motor de regras de negócio) e obter o resultado do cálculo que o motor de regras pode fornecer.	

Tabela 41 – Tipos de Subprocessos


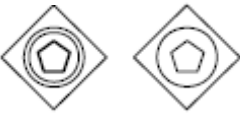



Elemento	Descrição	Notação
Ciclo (<i>Loop</i>)	Utilizada quando o número de repetições não é conhecido. A repetição será repetida enquanto a condição for atendida	
Múltiplas Instâncias	O atributo de múltiplas instâncias permite que uma atividade tenha “N” repetições, podendo ser instanciada em paralelo diversas vezes.	
Compensação	A tarefa de compensação é uma tarefa particular e não faz parte do fluxo normal de um processo. A tarefa de compensação é utilizada exclusivamente para executar a compensação de uma atividade já realizada no processo.	

Ad-Hoc	Um subprocesso ad-hoc indica um conjunto de atividades desempenhadas sem uma sequência pré-definida, pois suas tarefas não são conectadas pelo fluxo de sequência.	
--------	--	---

Gateways

Os Gateways são usados para controlar a divergência ou a convergência dos fluxos de sequência. Todos são representados por losangos, mas os marcadores internos é que definem os tipos de comportamentos. A Tabela 42 apresenta e descreve os tipos de *Gateway* utilizados na notação BPMN.






Tabela 42 – Tipos de *Gateway*

Tipos de Gateway		
Elemento	Descrição	Notação
Exclusivo	É utilizado para criar caminhos alternativos dentro de um fluxo de processo, onde apenas um pode ser tomado (representa o OU).	
Baseado no Evento	Representa um ponto de ramificação no processo em que os caminhos alternativos que seguem o gateway são baseados em eventos que ocorrem.	
Inclusivo	Permite criar caminhos alternativos, mas também em paralelo dentro de um fluxo de processo.	
Complexo	Usado para modelar um comportamento complexo de sincronização.	
Paralelo	É usado para sincronizar (combinar) fluxos paralelos e/ou criar fluxos paralelos.	

8.4.2 Dados

O elemento Dados permitem representar, transmitir, ler e escrever dados durante a execução do processo.


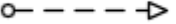
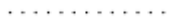
Tabela 43 – Elementos Dados BPMN

Elementos de Dados		
Elemento	Descrição	Notação
Objeto de dado	Este elemento representa a informação que é transmitida durante o processo.	
Coleção de objetos de dados	Representa uma coleção de dados, uma lista de itens.	
Dados de entrada	É uma entrada de dados externa ao processo, parâmetro de entrada. Pode ser lido por uma atividade.	
Dados de saída	É uma variável disponível com resultado da execução de um processo, parâmetro de saída.	
Repositório de dados	Este elemento representa um repositório de dados, onde o processo pode ler e escrever.	

8.4.3 Objetos de ligação

Existem objetos de ligação que permitem efetuar conexões entre os objetos de fluxo. Os quatro Objetos de Ligação são:

Tabela 44 – Objetos de Ligação BPMN


Objetos de Ligação		
Elemento	Descrição	Notação
Fluxo de Sequência	O Fluxo de Sequencia é representado por uma linha sólida com uma seta sólida na ponta e é utilizado para mostrar a ordem (sequência) em que as atividades são executadas num processo.	
Fluxo de mensagem	O Fluxo de Mensagem é representado por uma linha de tracejada com uma seta aberta na ponta e é usada para mostrar o fluxo de mensagens entre dois participantes separados do processo que as enviam e recebem.	
Associações	Uma associação é representada por uma linha de pontos com uma seta em linha na ponta e é utilizada para associar dados, texto, e outros artefactos com os objetos de fluxos. As associações são usadas para mostrar as entradas e saídas das atividades.	

8.4.4 Swimlanes

As *swimlanes* permitem dividir e organizar as atividades em diferentes categorias visuais em um diagrama. No BPMN existem dois tipos destes elementos: *pool* e *lane* como apresenta a Tabela 45.

Tabela 45 – Swimlanes BPMN


<i>Swimlanes</i>		
Elemento	Descrição	Notação
Pool	Uma Pool é uma representação gráfica que permite representar	

	um participante num processo. Permite isolar um conjunto de atividades de outras Pools e é representada por uma caixa gráfica.	
Lane	Uma Lane é uma subdivisão de uma Pool e pode ser horizontal, vertical ou ambos. As Lanes são utilizadas para organizar e categorizar as atividades.	

8.4.5 Artefactos

Os Artefactos são utilizados para providenciar informação adicional para os processos. A sua utilização permite aumentar a clareza do diagrama e expor certos pontos relevantes do processo. Existem os seguintes dois tipos Artefactos:

Tabela 46 – Artefactos BPMN

Artefactos		
Elemento	Descrição	Notação
Grupo	São usados com o objetivo de destacar, documentar ou analisar. Não afetam o fluxo de sequência do processo.	
Anotação	As anotações são um mecanismo que permite acrescentar informação em texto para facilitar a interpretação do diagrama.	