



Sistema de gestão de indicadores de produção

TIAGO MARQUES VILAÇA

Outubro de 2020

Sistema de gestão de indicadores de produção

Tiago Marques Vilaça

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

Orientador: Ricardo Almeida

Porto, Outubro 2020

Dedicatória

Dedico esta dissertação à minha família que sempre me apoiou.

Resumo

O departamento de produção da Continental Mabor utiliza uma base de dados desenvolvida em MsAccess para gerir os seus indicadores de produção. Esta aplicação, no início, era adequada, mas à medida que o tempo foi passando e quantidade de informação foi aumentando a aplicação começou a ter crashes frequentes e falhas prolongadas.

Assim, foi desenvolvida uma aplicação Web com as mesmas funcionalidades da aplicação existente, uma base de dados SQL com os dados migrados, acesso a um servidor de geração de relatórios e preenchimento automático de alguns campos através da integração de sistemas internos da Continental Mabor.

Para validar as funcionalidades da aplicação desenvolvida foram elaborados questionários de satisfação aos utilizadores e testes de performance. De forma geral os resultados foram positivos e os utilizadores classificaram a aplicação como uma melhoria em relação à anterior.

Palavras-chave: Gestão de dados, Desenvolvimento Web, MsAccess, Reporting Server

Abstract

Continental Mabor's production department uses a MsAccess database to manage their production indicators. This application, in the beginning was enough to satisfy the needs of the department but as time went on and the database grew bigger the began to have frequent crashes and long failure times.

So, a Web application was developed with the same functionalities as the existent application, an SQL database with migrated data, access to a report server and automatic fill in of fields due to the integration with Continental Mabor's internal systems.

Satisfaction questionnaires and performance tests were conducted to validate the functionalities of the developed application. The results were positive, and the users defined the application as an improvement over the previous one.

Keywords: Data management, Web Development, MsAccess, Reporting Server

Agradecimentos

Agradeço ao meu orientador Ricardo Almeida pela disponibilidade e esforço prestado no esclarecimento de dúvidas e revisão do documento.

Agradeço aos meus amigos e família pelos conselhos prestados.

Agradeço aos meus colegas da Continental Mabor pelo fantástico acolhimento e ajuda no desenvolvimento da aplicação.

Índice

1	Introdução	1
1.1	Contexto	1
1.1.1	Apresentação da empresa	1
1.1.2	Processo de construção de um pneu	3
1.2	Problema	4
1.3	Objetivos	5
1.4	Organização do documento	5
2	Análise de valor	7
2.1	Modelo NCD (New Concept Development)	7
2.1.1	Identificação de oportunidade	8
2.1.2	Análise de oportunidade	8
2.1.3	Geração e enriquecimento de ideias	8
2.1.4	Seleção de ideias	9
2.1.5	Definição de conceito	9
2.2	Valor percebido e valor para o cliente	9
2.2.1	Valor percebido	10
2.2.2	Valor para o cliente	10
2.3	Proposta de valor	10
2.4	Modelo Canvas	11
3	Estado de Arte	13
3.1	Conceitos e tecnologias	13
3.1.1	Conceitos Aplicações Web	13
3.1.2	Tecnologias de desenvolvimento Web	14
3.2	Aplicação preexistente	16
3.3	Aplicações semelhantes	22
3.3.1	Critérios de seleção	22
3.3.2	Critérios de avaliação	23
3.3.3	HyperOffice Atlas	23
3.3.4	Knack	28
3.3.5	Zoho Creator	33
3.3.6	Comparação e Conclusão	36
4	Design	39
4.1	Requisitos	39
4.1.1	Stakeholders	39
4.1.2	Requisitos funcionais	40
4.1.3	Requisitos não funcionais	44
4.2	Modelo de domínio	44

4.3	Arquitetura	46
4.3.1	Arquiteturas propostas	46
4.3.2	Arquitetura escolhida.....	48
4.3.3	Especificação da arquitetura selecionada	49
5	Desenvolvimento	53
5.1	Use case realization.....	53
5.1.1	UC1 e UC3 - Criar e editar utilizadores.....	53
5.1.2	UC2 e UC3 - Efetuar login e mudança de password	59
5.1.3	UC5/UC6/UC7/UC8/UC9/UC10 - Criar e editar turnos, máquinas e tipos de stock	64
5.1.4	UC11/UC12/UC26 - Criar e editar dados de produção e sugestões do sistema.....	69
5.1.5	UC13/UC14/UC15/UC16/UC17/UC18/UC25/UC26 - Criar e editar dados de produção diária, dados de programação, APA e Workoff	76
5.1.6	UC19 e UC20 - Criar e editar dados de SCRAP.....	80
5.1.7	UC21/UC22/UC23/UC24- Criar e editar dados do relatório de turno e de relatório da reunião da supervisão	87
5.1.8	UC25- Gerar relatórios	91
5.2	Migração de dados.....	93
5.3	Testes.....	94
5.3.1	Testes unitários.....	94
5.3.2	Testes de integração.....	95
5.3.3	Testes de aceitação.....	96
6	Avaliação	99
6.1	Métricas	99
6.2	Hipóteses.....	100
6.3	Metodologia.....	101
6.4	Análise de Resultados.....	101
6.4.1	Inquérito de satisfação ao utilizador	102
6.4.2	Tempo de resposta.....	103
7	Conclusão	105
7.1	Objetivos alcançados	106
7.2	Trabalho futuro	107

Lista de Figuras

Figura 1 - Processo de construção de um pneu	3
Figura 2 – Modelo NCD (Koen et al., 2001)	7
Figura 3 – Modelo Canvas	11
Figura 4 – Modelo Cliente-Servidor (Oluwatosin, 2014)	14
Figura 5 - Exemplo de formulários de produção DP111	17
Figura 6 - Pré-visualização parcial de um relatório de produção.....	18
Figura 7 – Dados programados para o dia 09/07/2019	19
Figura 8 – Dados produzidos no dia 08/07/2019	19
Figura 9 – Percentagem de lixo (SCRAP) dos vários departamentos (I a IV)	20
Figura 10 – Relatório Diário.....	20
Figura 11 - Relatório de turno	21
Figura 12 - Relatório diário de reunião da Supervisão	22
Figura 13 – <i>Setup</i> inicial.....	24
Figura 14 – Design da tabela de teste	25
Figura 15 – <i>Form Designer</i>	25
Figura 16 – Criação de uma view/report	26
Figura 17 – Aplicação desenvolvida.....	26
Figura 18 – Formulário	27
Figura 19 – View	27
Figura 20- Relatório PDF gerado	28
Figura 21 – <i>Data</i>	29
Figura 22 – Criação de um objeto	29
Figura 23 – Pages	30
Figura 24 – Criação do formulário	30
Figura 25 –Tabela do relatório	31
Figura 26 – Gráfico de barras	32
Figura 27 – Aplicação final	33
Figura 28 – Criação do formulário	34
Figura 29 – Criação do relatório	34
Figura 30 – Formulário final	35
Figura 31 – Relatório final	35
Figura 32 – Relatório PDF	35
Figura 33 – Diagrama de casos de uso	40
Figura 34 – Modelo de domínio	45
Figura 35 – Arquitetura simples.	47
Figura 36 – Arquitetura baseada em Web APIs.....	47
Figura 37 – Arquitetura baseada em micro serviços.	48
Figura 38 – Diagrama de componentes <i>Frontend</i>	50
Figura 39 – Diagrama de componentes <i>Backend</i>	51
Figura 40 – Diagrama de implantação.....	52

Figura 41 – Diagrama de atividades de criar um utilizador	54
Figura 42 – Diagrama de classes de criar e editar utilizadores	55
Figura 43 – Diagrama de sequência de criação de um utilizador	56
Figura 44- Diagrama de sequencia de edição de utilizador	57
Figura 45 – Lista de utilizadores e botões de adição e edição	58
Figura 46 – Formulário de registo e edição de utilizadores.....	58
Figura 47 – Diagrama de atividades do processo de login e alteração de password	59
Figura 48 – Diagrama de classes do processo de login e mudança de password	60
Figura 49 – Diagrama de sequência <i>reset</i> de password.....	61
Figura 50 – Listagem de utilizadores com botão de reset password	61
Figura 51- Diagrama de sequência do processo de login e alteração de password.....	62
Figura 52 – Sequencia de introdução de login e alteração de password	62
Figura 53 – Excerto de código de comparação de password.....	63
Figura 54 – Diagrama de atividades da gestão de turnos, máquinas e tipos de stock.....	64
Figura 55 – Diagrama de classes de Máquinas, Stocks e Turnos	65
Figura 56- Diagrama de sequência da criação de máquinas/turnos/tipo de stock.....	66
Figura 57- Diagrama de sequência de edição de máquinas/turnos/tipo de stock.....	67
Figura 58 – Processo de criação de turnos automática.	68
Figura 59 – Página de gestão de máquinas	68
Figura 60 – Formulário de registo e edição de uma máquina	69
Figura 61 – Diagrama de atividades da criação e edição dados de produção e sugestões do sistema	70
Figura 62- Diagrama de classes da criação e edição dados de produção e sugestões do sistema	71
Figura 63 - Diagrama de sequencia da criação e edição dados de produção e sugestões do sistema manual	73
Figura 64 – Diagrama de sequência sistema de sugestão automática	74
Figura 65 – Interface de registo de dados de produção para o Departamento IV.....	75
Figura 66 – Interface de registo de dados de produção para o Departamento III	75
Figura 67 – Diagrama de atividades de criação e edição dos dados diários	76
Figura 68 – Diagrama de classes dos dados diários	77
Figura 69 - Diagrama de sequencia do registo e edição de dados diários	79
Figura 70 – Página web de registo de dados de produção diários.....	80
Figura 71 – Diagrama de atividades de criação e edição dos dados de SCRAP.....	81
Figura 72 – Diagrama de classes do processo de gestão de SCRAP e constantes de Metas SCRAP	82
Figura 73 – Diagrama de sequencia do registo e edição de SCRAP	84
Figura 74 - Diagrama de sequencia de gestão de constantes do tipo Scrap.....	85
Figura 75- Página de registo de dados SCRAP	86
Figura 76- Página de gestão de metas de SCRAP.....	86
Figura 77 – Diagrama de atividades da gestão do relatório de turno e da reunião da supervisão	87
Figura 78 -Diagrama de classes com as classes responsáveis pela gestão dos UCs 21 a 24	88

Figura 79 – Diagrama de sequencia da criação e edição de um <i>ShiftReport</i>	90
Figura 80 – Registo de uma reunião da supervisão	91
Figura 81 – Diagrama de atividades da geração de relatórios.....	92
Figura 82 – Excerto de código de migração dos turnos	93
Figura 83 – Testes unitários aos métodos <i>get</i> e <i>set</i> da classe <i>User</i>	94
Figura 84- Inicialização do Cliente HTTP de teste	95
Figura 85- Teste de integração da função de login.....	95

Lista de Tabelas

Tabela 1 – Resultados dos critérios de avaliação	36
Tabela 2 – Casos de uso e prioridade	41
Tabela 3 – Testes de Aceitação	96
Tabela 4 - Metodologias.....	101
Tabela 5 - Escala de classificação dos inquéritos.....	101
Tabela 6 – Questionário de feedback do frontend com percentagem de respostas	102
Tabela 7- Questionário de feedback dos relatórios com percentagem de respostas.....	103
Tabela 8 – Resultados dos testes de integração.....	104
Tabela 9 – Objetivos do projeto	106
Tabela 10 – Casos de uso implementados	106

Acrónimos e Símbolos

Lista de Acrónimos

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
CSV	<i>Comma-separated Values</i>
DTO	<i>Data Transfer Object</i>
FFE	<i>Fuzzy Front-End</i>
GRPD	<i>General Data Protection Regulation</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
HTTPS	<i>Hyper Text Transfer Protocol Secured</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model view Controller</i>
NCD	<i>New Concept Development</i>
NPD	<i>New Product Development</i>
PDF	<i>Portable Document Format</i>
SQL	<i>Structured Query Language</i>
TSV	<i>Tab-separated Values</i>
UI	<i>User Interface</i>

Lista de Símbolos

1 Introdução

Neste capítulo, é fornecido contexto relativo ao projeto desenvolvido como tema da dissertação, o problema e os objetivos requeridos pela organização. São também mencionados os motivos da realização do projeto e a metodologia de desenvolvimento deste. Finalmente, é identificada a estrutura do documento, fornecendo uma pequena descrição de cada capítulo.

1.1 Contexto

O projeto descrito neste documento consiste na descrição do processo de construção de uma aplicação de gestão de indicadores de produção desenvolvida durante um estágio académico realizado na Continental Mabor (Continental, 2019), localizada em Lousado. O estágio possui uma duração de 6 meses, tendo este sido iniciado no dia 11/11/2019 e com termino previsto no dia 11/05/2020, no entanto devido à pandemia COVID-19 esta data foi alterada para o final de junho de 2020. Nas seguintes subsecções é feita uma apresentação breve da Continental Mabor e é descrito o processo de construção de um pneu para dar seguimento à descrição do problema.

1.1.1 Apresentação da empresa

Fundada no dia 8 de outubro de 1871 em Hanôver na Alemanha, a Continental iniciou as suas atividades como uma empresa de produção de borracha. Desde a sua inepção é possível destacar alguns momentos com maior importância (Continental, 2020a):

- 1898- Iniciada a produção de pneus pneumáticos para automóveis sem padrão do piso;
- 1900- O primeiro dirigível alemão, o LZ 1, utiliza material para balões da Continental para selar os sacos de gás;
- 1901- Carro que vence a corrida Nice-Salon-Nice utiliza pneus Continental;

- 1935 - Série ininterrupta de sucessos em corridas com os pneus de corrida Continental a equipar carros Mercedes e Auto-Union;
- 1950- A Continental tem um efetivo de 13 500 trabalhadores na fábrica principal.
- 1979- A aquisição da empresa europeia de pneus Uniroyal, Inc., EUA, fornece à Continental uma maior base na Europa;
- 1987- Aquisição do fabricante norte-americano de pneus General Tire, Inc.

Em 1989, foi criada uma *joint venture* com a empresa portuguesa Mabor, iniciando a produção em Lousado, Famalicão, Portugal. Atualmente, a Continental Mabor emprega cerca de 2000 funcionários e, contando com as restantes fábricas em Portugal o número de empregados sobe para cerca de 2500. Estas fábricas são:

- Continental Mabor (Lousado): Focada na construção de Pneus;
- Continental Pneus Portugal (Lousado): Focada na venda de Pneus em Portugal;
- Continental Indústria Têxtil do Ave (Lousado): Produção de artigos têxteis para a indústria de borracha;
- Continental Lemmerz (Palmela): Produz componentes para automóveis;
- Continental Teves: Produção de sistemas de travagem;
- Continental Engineering Services (Porto): Desenvolve sistemas e software para uso automóvel;

Os valores do Grupo Continental são(Continental, 2020b):

- Confiança: A Continental incentiva uma cultura de confiança entre os colaboradores e *stakeholders*;
- Paixão por vencer: “A competição é o nosso mundo; o desempenho de topo é a nossa meta”;
- Liberdade para agir: A Continental incentiva os colaboradores a serem livres na organização do seu trabalho;
- Uns pelos outros: Elevado espírito de cooperação ao nível global.

A visão do Grupo Continental é “A vossa mobilidade. A vossa liberdade. A nossa assinatura.”. Esta visão é descrita como “Somos uma empresa comercial, pelo que a nossa principal tarefa é fornecer, de forma eficiente, bens e serviços para total satisfação dos nossos clientes e *stakeholder*, tais como colaboradores, investidores, parceiros de negócios, políticos e sociedade.”(Continental, 2020c).

Finalmente a missão da Continental é “O Futuro Começa Mais Cedamente com a Continental.” (Continental, 2020d). A partir desta missão é possível deduzir os seguintes tópicos:

- Segurança – o nosso objetivo: zero acidentes. Proteger a vida e preservar os recursos;
- Informação – o nosso objetivo: poupar tempo, aumentar o conforto. Mobilidade mais inteligente através da condução constantemente conectada;
- Ambiente – o nosso objetivo: ar limpo. Condução sem emissões e eficiente em termos de recursos;
- Mobilidade acessível – o nosso objetivo: mobilidade individual para todos. Proporcionar mais liberdade e oportunidades.

1.1.2 Processo de construção de um pneu

A fábrica da Continental Mabor em Lousado, funciona utilizando um horário de turnos. Cada dia útil possui 3 turnos com os seguintes espaços de tempo:

- Turno 1 - Das 8 às 16 horas;
- Turno 2 - Das 16 às 24 horas;
- Turno 3 - Das 24 às 8 horas.

O turno 3, apesar de ocorrer no dia seguinte, é considerado um turno do mesmo dia que o turno 1 e 2. Por exemplo, no dia 20/10/2020, o turno 1 e 2 decorrem das 8 às 24 horas e o turno 3 das 24 às 8 horas do dia 21/10/2020, mas é registado como decorrido no dia 20.

Durante o fim-de-semana existem 5 turnos com os seguintes horários:

- Turno 5 – Das 8 às 19 horas de sábado;
- Turno 6 – Das 19 às 6 horas de sábado;
- Turno 7 – Das 6 às 14 horas de domingo;
- Turno 8 – Das 14 às 24 horas de domingo;
- Turno 9 – Das 24 às 8 horas de domingo.

De forma similar ao turno 3, apesar de decorrer durante uma segunda feira, o turno 9 é considerado um turno de fim-de-semana.

Como é possível verificar, a fábrica produz pneus 24 horas por dia durante os 7 dias semanais, havendo paragens durante feriados ou períodos de férias (Natal, etc.). Durante cada turno existem 6 processos pertencentes à linha de produção de um pneu que decorrem em simultâneo. Estes processos são apresentados na Figura 1.

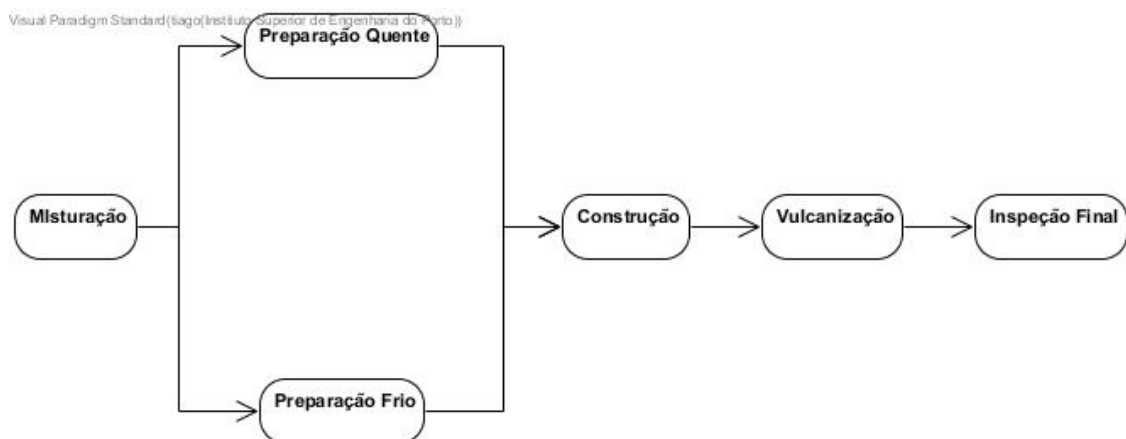


Figura 1 - Processo de construção de um pneu

1.1.2.1 Misturação (DPI)

Na área de misturação, são misturadas matérias primas como a borracha natural, sílica e outros produtos químicos para gerar vários compostos a utilizar no processo de montagem dos pneus.

1.1.2.2 *Preparação Quente (DPIIQ)*

Na área de preparação Quente os compostos gerados na área anterior são processados a elevadas temperaturas para criar componentes da parte exterior do pneu.

1.1.2.3 *Preparação Frio (DPIIF)*

Na área de preparação a frio, vários compostos são tratados a frio para criar componentes da parte interior do pneu.

1.1.2.4 *Construção (DPIII)*

No processo de construção, os componentes produzidos nas áreas de preparação são assimilados, em duas fases, para formar o Pneu Verde. Este pneu apresenta uma estrutura cilíndrica lisa e não está pronto para ser utilizado na estrada.

1.1.2.5 *Vulcanização (DPIV)*

O Pneu Verde é vulcanizado numa prensa a altas temperaturas para formar o pneu finalizado. Este processo pode demorar entre 10 a 45 minutos por pneu.

1.1.2.6 *Inspeção final (DPV)*

Finalmente, o pneu vulcanizado passa por várias inspeções de qualidade para validar a construção correta do pneu e se está a par dos standards de qualidade exigido pela Continental. Se o pneu passar por esta fase é guardado em armazém para depois ser vendido ao cliente.

1.2 Problema

O departamento de produção da Continental Mabor é responsável por gerir toda a quantidade de produtos resultantes do processo de desenvolvimento de um pneu pelas diversas máquinas presentes na fábrica, em cada turno, e a produção total de um dia e gerar relatórios com estes valores.

Para suportar esta tarefa é utilizada uma aplicação desenvolvida utilizando MSAccess (Microsoft, 2020a) disponibilizada pela Microsoft (Microsoft, 2020b). Esta tecnologia permite desenvolver aplicações com base de dados, formulários e geração de relatórios, em relativamente pouco tempo e mínimo conhecimento de programação. Assim, o MSAccess pode ser considerado uma boa opção para negócios de pequena dimensão devido ao baixo investimento necessário para gerir os dados de negócio.

No seu estado atual esta aplicação apresenta vários problemas:

- Lentidão: A base de dados possui registos com mais de 10 anos e algumas tabelas têm possuem milhares de registos, tornado o acesso aos dados um processo demorado. Como prevenção, a cada ano são removidos os registos do ano mais antigo da base de dados principal e estes são salvaguardados numa copia separada.
- Falhas de sistema: Devido ao tamanho e idade da aplicação, esta encontra-se bastante desatualizada levando à ocorrência diária de *crashes*.

- Elevado tempo de manutenção: Resultante das falhas diárias, tanto o departamento de Produção como o de IT perdem bastante tempo a manter a aplicação operacional.

1.3 Objetivos

Com a realização pretende-se substituir o atual sistema de gestão de indicadores de produção por uma solução tecnologicamente atualizada. Durante o percurso do estágio, o estagiário deverá:

- Analisar os requisitos da aplicação através da escrita de um documento SRS (*Software requirements specification*)(Hécio A. Soares and Raimundo S. Moura, 2015)
- A modelação de base de dados tendo em conta que os dados presentes na aplicação preexistente terão de ser migrados;
- Implementar uma solução WEB multiutilizador com todas as funcionalidades disponíveis na aplicação MSAccess. Esta solução deverá estar tecnologicamente a par com as diversas aplicações da Continental Mabor e standards de desenvolvimento que providenciem escalabilidade;
- Geração de relatórios;
- Documentar a aplicação;
- Integrar a nova solução no sistema da Continental Mabor para providenciar preenchimento automático de alguns campos dos relatórios recebendo informações diretamente de outras aplicações ou base de dados da organização;

1.4 Organização do documento

O relatório encontra-se estruturado da seguinte forma:

- Capítulo 1 - Introdução: Apresentação do Contexto, problema e objetivos e a estrutura do documento;
- Capítulo 2 - Análise de Valor: Definição da análise de valor do projeto
- Capítulo 3 - Estado de Arte: Contextualização tecnológico do tema de aplicações web, análise da aplicação pré-existente e levantamento e comparação de soluções semelhantes disponíveis no mercado;
- Capítulo 4 - Design: Descrição dos requisitos funcionais e não funcionais e a arquitetura escolhida para a implementação
- Capítulo 5 - Implementação: Processo de desenvolvimento do sistema;
- Capítulo 6 - Avaliação: Descrita o processo de avaliação da solução obtida;
- Capítulo 7 - Conclusão: Conclusões relativas ao projeto desenvolvido.

2 Análise de valor

Neste capítulo é definido o módulo de análise de valor. Primeiramente é apresentado o Modelo de New Concept Development (NCD) e os seus cinco elementos de *Front End*. De seguida, são definidos os conceitos de valor, valor percebido e valor para o cliente. Finalmente, é declarada a proposta de valor e o modelo Canvas.

2.1 Modelo NCD (New Concept Development)

Segundo Peter Koen, o processo de inovação está dividido em 3 áreas: o Fuzzy Front End (FFE), o New Product Development(NPD) e a comercialização (Koen et al., 2002).Neste capítulo é estudado o FFE através da aplicação do modelo New Concept Development (NCD).

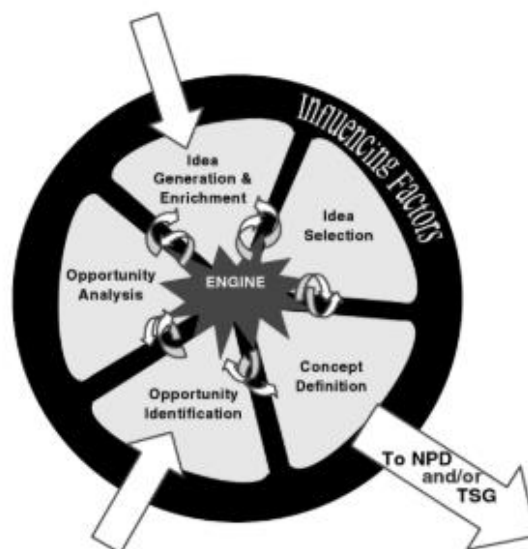


Figura 2 – Modelo NCD (Koen et al., 2001)

O modelo NCD, apresentado no Figura 2, define uma linguagem comum e uma definição dos componentes chave do FFE. No centro está o motor, que usa como combustível a liderança e cultura da organização, que potencia os cinco elementos do NCD, a identificação de oportunidade, a análise de oportunidade, a geração de ideias, a seleção de ideias e a definição do conceito. Na área exterior estão os fatores externos que influenciam o processo de inovação (Koen et al., 2001).

2.1.1 Identificação de oportunidade

Nesta fase a organização identifica as oportunidades disponíveis no ambiente empresarial. Estas oportunidades podem ser para obter uma vantagem competitiva, reduzir custos de operações, uma nova direção do negócio ou até uma melhoria de um produto existente. Os fatores externos possuem um papel importante na identificação de oportunidades, por exemplo, estudos de mercados, avanços tecnológicos e feedback de clientes (Koen et al., 2001).

Assim, a oportunidade para a realização deste estágio surge devido às falhas constantes do sistema de gestão de indicadores de produção e devido aos avanços tecnológicos na área do desenvolvimento *web*, é possível oferecer uma melhoria na aplicação de gestão de indicadores preexistente.

2.1.2 Análise de oportunidade

Nesta fase é necessário recolher informação adicional para determinar se a oportunidade é viável para ser transformada numa oportunidade de negócio. Assim, são necessários estudos de mercado, análise de tecnologias e recolha de feedback de utilizadores. (Koen et al., 2001).

Neste caso, os utilizadores já apresentavam queixas pela fraca qualidade do software de gestão de indicadores. Ao realizar este estágio, a Continental Mabor pode proporcionar um upgrade ao software sem investir muito nos custos associados ao desenvolvimento de uma aplicação nova e ao mesmo tempo providenciar experiência profissional ao estagiário.

2.1.3 Geração e enriquecimento de ideias

É nesta fase que se dá a génese, desenvolvimento e maturação da oportunidade para uma ideia concreta. É um processo iterativo onde as ideias podem ser atualizadas, modificadas ou até mesmo destruídas. As ideias são geradas através de contacto direto com os utilizadores/clientes com sessões de *brainstorming* e colaborações com outras empresas/instituições. (Koen et al., 2001).

O projeto consiste no desenvolvimento de uma nova solução de gestão de indicadores de produção para substituir uma já existente. Assim, foi realizada uma reunião para com os

proponentes do estágio e os principais utilizadores desta aplicação com o objetivo de gerar algumas ideias:

- Desenvolvimento de uma aplicação Web;
- Desenvolvimento de uma aplicação multiutilizador;
- Aplicação deverá manter e melhorar funcionalidades da aplicação preexistente;
- Geração de relatórios;
- Técnicas de *machine learning* para preencher os valores planeados para cada dia;
- Integração com os outros sistemas da organização;
- Assistente virtual para esclarecimento de dúvidas.

2.1.4 Seleção de ideias

Nesta fase as ideias descritas na geração anterior são escolhidas de acordo com as limitações tecnológicas e temporais. É de se notar que não existe um método único para selecionar que ideias manter e quais remover. (Koen et al., 2001)

Desta forma, foram selecionadas as ideias:

- Desenvolvimento de uma aplicação Web;
- Desenvolvimento de uma aplicação multiutilizador;
- Aplicação deverá manter e melhorar funcionalidades da aplicação preexistente;
- Geração de relatórios;
- Integração com os outros sistemas da organização.

2.1.5 Definição de conceito

O último modelo do NCD envolve o desenvolvimento de um caso de negócio baseado em potencial de mercado, necessidades do consumidor, requerimentos de investimento, incógnitas tecnológicas e o risco do projeto (Koen et al., 2001).

O objetivo deste estágio é a implementação de uma aplicação Web de gestão de indicadores de produção que permita substituir a aplicação preexistente desatualizada e defeituosa. A aplicação deve manter todas as funcionalidades existentes à priori e deverá, também, integrar-se nos outros sistemas da Continental.

2.2 Valor percebido e valor para o cliente

Neste capítulo serão definidos os conceitos de valor percebido, valor para o cliente e os benefícios e sacrifícios do projeto.

2.2.1 Valor percebido

Raquel Sánchez-Fernández, identifica varias definições para o valor percebido sendo que estes dependem dos valores individuais de valor, preço, utilidade e qualidade (Sánchez-Fernández and Iniesta-Bonillo, 2007). Assim, é possível verificar que o valor percebido varia de individualmente e cada pessoa pode ter definições diferentes de valor para o mesmo produto.

Os utilizadores pretendem uma aplicação que seja considerada uma melhoria em relação à aplicação preexistente, com baixo número de falhas e de fácil utilização. A organização, ao iniciar este projeto, pretende providenciar aos seus colaboradores melhores condições de trabalho e reduzir o tempo perdido com a manutenção do sistema a descartar.

2.2.2 Valor para o cliente

“A percepção do valor para o cliente envolve uma troca entre o que o cliente recebe (qualidade, benefícios, valor, utilidades) e o que ele oferece (preços, sacrifícios) para adquirir o produto”(Spiteri and Dion, 2004). Portanto é necessário identificar os benefícios e os sacrifícios para identificar o valor para o cliente.

Os Benefícios são:

- **Aplicação atualizada e com baixo número de falhas;**
- **Aplicação fácil de utilizar;**
- **Aplicação acessível em qualquer lugar;**
- **Suporte.**

Os Sacrifícios são:

- **Tempo da aprendizagem e habituação à nova aplicação.**

2.3 Proposta de valor

A proposta de valor é uma declaração que permite apresentar a ligação entre os benefícios da utilização do produto com as necessidades dos clientes. (Osterwalder and Pigneur, 2003) Assim, foi definida a seguinte proposta de valor:

A solução desenvolvida no âmbito deste estágio é uma aplicação web que permite aos colaboradores da Continental Mabor gerir os indicadores de produção da fábrica e gerar relatórios com esses indicadores apresentados de uma forma simples e legível. O objetivo deste projeto é substituir a plataforma anterior de gestão de indicadores pois se encontra desatualizada e apresenta falhas diárias de sistema. Ao realizar este *upgrade* os colaboradores

podem usufruir de uma nova plataforma atualizada, rápida, eficaz e com baixa quantidade de falhas.

2.4 Modelo Canvas

O modelo Canvas é desenvolvido por Alexander Osterwalder and Yves Pigneur, descreve como é que uma organização gera valor a partir do seu produto (Osterwalder et al., 2011). Está dividido em 9 partes: Parcerias Chave, Atividades Chave, Recursos Chave, Proposta de Valor, Relações com os clientes, Canais de comunicação, Segmentos de Clientes, Custo de estrutura e Receitas. A Figura 3 apresenta o Modelo Canvas para o projeto desenvolvido.

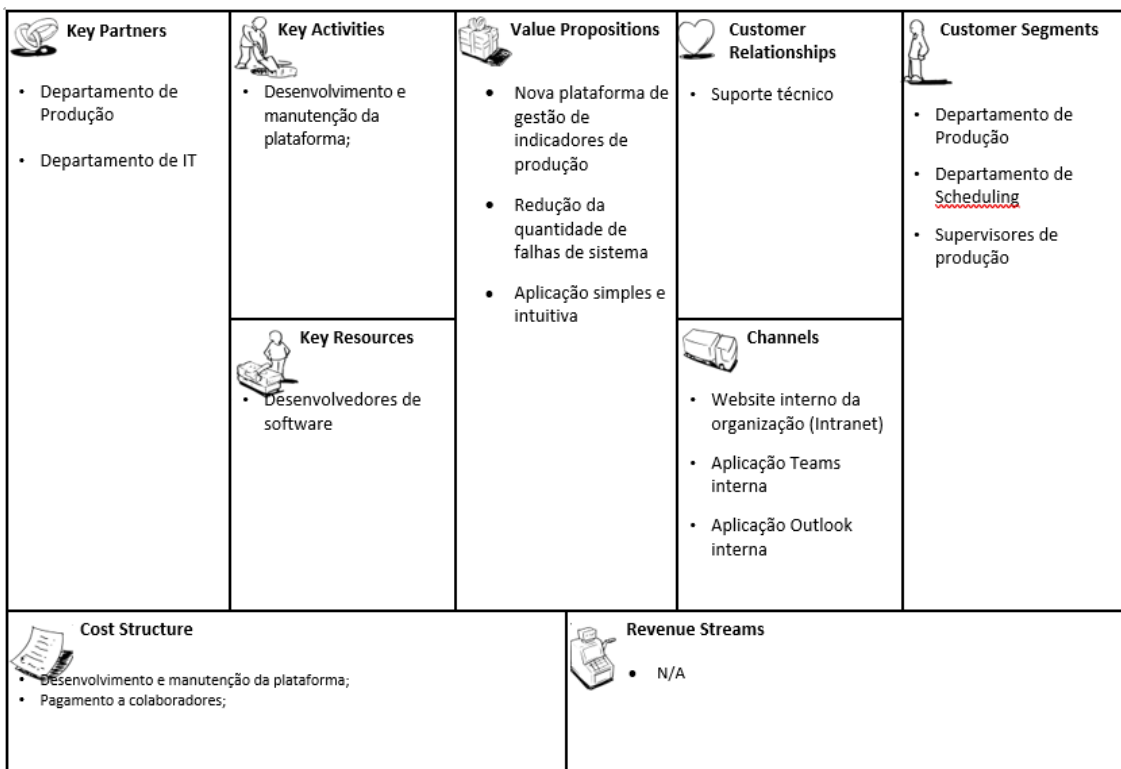


Figura 3 – Modelo Canvas

É possível visualizar que o componente de Receitas do modelo Canvas acima mencionado se encontra vazio. Sendo uma aplicação para uso interno, esta é fornecida gratuitamente aos colaboradores.

3 Estado de Arte

Neste capítulo é realizada uma introdução teórica aos conceitos de Aplicações Web. São também mencionadas as principais tecnologias utilizadas atualmente. De seguida, é feita uma análise à aplicação preexistente. No final, é efetuado um levantamento de aplicações semelhantes à solução pretendida e é realizada uma comparação entre estas e a aplicação preexistente.

3.1 Conceitos e tecnologias

Neste capítulo é apresentada uma curta introdução teórica do funcionamento das aplicações Web e, de seguida são listadas algumas tecnologias e linguagens de programação mais utilizadas, fazendo referência às suas principais características e sites que as utilizam.

3.1.1 Conceitos Aplicações Web

As aplicações Web são programas instalados em servidores Web, acessíveis por qualquer utilizador que tenha acesso à internet e a um *Web Browser*. O exemplo mais comum deste tipo de aplicações são os “sites” que acedemos no dia-a-dia. Este tipo de aplicações seguem o modelo Cliente-Servidor(Oluwatosin, 2014).

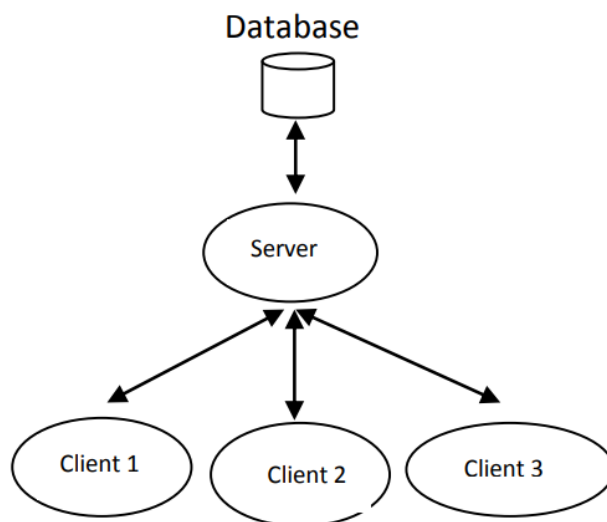


Figura 4 – Modelo Cliente-Servidor (Oluwatosin, 2014)

Na Figura 4 é apresentado o Modelo Cliente-Servidor que representa o funcionamento geral das aplicações Web. Os Clientes realizam pedidos ao Servidor, usando normalmente o padrão *Hypertext Transfer Protocol (HTTP)/ Hypertext Transfer Protocol Secure (HTTPS)*. Estes padrões consistem no envio de texto ou imagens entre o cliente e servidor, utilizando um modelo de pedido/resposta, ou seja, o cliente efetua um pedido HTTP ao servidor e este responde utilizando uma resposta HTTP com um código que informa do sucesso ou insucesso do pedido e com o conteúdo requerido pelo cliente. (J. Chen and W. Cheng, 2016).

Jm Riós define as aplicações web como “um sistema que geralmente é constituído por um back-end e páginas web (o front-end)”(Ríos and Souto, 2019) . De facto, as aplicações Web são normalmente constituídas por:

- **Back-end:** responsável por aceder à base de dados e gerir a camada de negócio. Normalmente responde aos pedidos HTTP com mensagens no formato JSON (Pezoa et al., 2016), XML (Bray et al., 2000) e HTML (Alawar and Naser, 2017)
- **Front-end:** responsável por receber dados do *back-end* e torná-los visíveis para o cliente.

3.1.2 Tecnologias de desenvolvimento Web

Nesta seção são apresentadas algumas ferramentas para o desenvolvimento de aplicações Web. As ferramentas estão organizadas por funcionalidade, se são utilizadas para implementar *front-end* ou *back-end*.

3.1.2.1 Back-end: Node.js/Express

O Node.js é um ambiente de execução baseado em *javascript open-source* desenvolvido pela *OpenJs Foundation* (OpenJSFoundation, 2020). O Node.js utiliza uma arquitetura orientada a eventos, ou seja, o sistema deteta uma mudança de estado despoletada por uma ação e age

sobre esta (Michelson, 2006). Este ambiente de execução apresenta uma implementação assíncrona e não-bloqueante sendo assim mais fácil desenvolver sistemas escaláveis com o Node.js (Node.js, 2020). O Node.js foi desenhado com baixa latência em mente sendo bem adequado para ser utilizado como fundação de *frameworks* como o Express.

O Express é uma *framework web* que adiciona funcionalidades fundamentais de aplicações web sem ofuscar as funcionalidades existentes do Node.js. O Express é utilizado por algumas aplicações tais como **PayPal** (Paypal, 2020), **Uber**(Uber, 2020) e **IBM**(IBM, 2020)

3.1.2.2 Back-end e Front-end: Asp.net

O Asp.NET é uma framework de desenvolvimento web open-source baseada em C# desenvolvida pela Microsoft (Microsoft, 2020c). De maneira similar à relação entre Node.js e Express definida no ponto 3.1.2.1 o Asp.Net estende a plataforma .Net adicionando-lhe:

- Framework para processamento de pedidos web;
- Templates para construir páginas web dinâmicas;
- Bibliotecas para padrões web como *Model View Controller* (MVC)
- Sistema de autenticação que inclui bibliotecas, uma base de dados e páginas de modelos para manuseamento de logins, incluindo autenticação e autenticação externa com Google, Twitter entre outras.

Aplicações que utilizam o Asp.NET são o **StackOverflow** (StackOverflow, 2020), **UPS** (UPS, 2020) e o *website* da **Microsoft**(Microsoft, 2020b).

3.1.2.3 Back-end: Laravel

O Laravel é uma *framework web open-source* baseada em PHP (Laravel, 2020). Oferece as seguintes funcionalidades:

- Desenvolvimento para padrão MVC;
- Gestor de dependências;
- Suporte para testes unitários;
- Gestão de serviços de pagamento;
- Suporte de migrações de bases de dados.

Sites que utilizam o Laravel são **Yulair** (Yulair, 2020), **Startups** (Startups, 2020) e **Fox47** (Fox, 2020).

3.1.2.4 Front-end: React.js

O React.js é uma *framework web baseada em Javascript*, desenvolvida pelo Facebook para a construção de interfaces gráficas. Apresenta uma arquitetura por componentes, ou seja, cada componente gere o seu próprio estado e as interfaces são construídas de modo declarativo (Facebook, 2020a).

Os principais sites que usam React.js são: **Facebook** (Facebook, 2020b), **Instagram**(Instagram, 2020) e **Netflix** (Netflix, 2020).

3.1.2.5 Front-end: Angular

O Angular é uma *framework* web baseada em *TypeScript* desenvolvida pela Google (Google, 2020). Desenvolvida para a construção de interfaces gráfica que providencia as seguintes funcionalidades:

- Carrega apenas o código necessário para mostrar a interface;
- *Code completion e feedback* com a utilização dos editores de texto e IDEs mais populares;
- Geração de código;
- Templates;
- Linha de comandos para geração de novos componentes;
- Testing.

Algumas das aplicações mais conhecidas que usam Angular são: **Samsung Forward**(Samsung Forward, 2020) , **Forbes** (Forbes, 2020) e a **Delta** (Delta, 2020).

3.2 Aplicação preexistente

A aplicação para gerir os indicadores de produção da Continental Mabor é utilizada por vários utilizadores de vários departamentos em simultâneo:

- DPI – Departamento da área de misturação;
- DPII – Departamento da área de preparação. No contexto desta aplicação o departamento de preparação quente e frio são mencionados em conjunto como DPII;
- DPIII – Departamento da área de construção;
- DPIV – Departamento da área de vulcanização;
- DPV – Departamento da área de inspeção de qualidade.

Estes departamentos pertencem à área de produção e são responsáveis por introduzir dados relativos ao que cada máquina produziu e valor em stock de cada material num determinado turno. O programa apresenta diferentes formulários para os utilizadores de distintos departamentos.

Produção Dia Relatório Individual

Produção diária

Data
Turno

Equipa
Departamento

Máquina	Quantidade
Módulo 01	385
Módulo 02	521
Módulo 03	394
Módulo 04	514
Módulo 05	385
Módulo 06	579
Módulo 07	502
Módulo 08	422
Módulo 09	490
Módulo 10	364
Módulo 11	378
Módulo 12	510
Módulo 13	482
Módulo 14	409
Módulo 15	380
Módulo 16	468

Figura 5 - Exemplo de formulários de produção DPIII

Se um utilizador aceder a dados já preenchidos, a aplicação permite a edição destes com o objetivo de corrigir erros de introdução. Após a introdução de dados de todos os departamentos forem introduzidos é possível imprimir um relatório de produção (Figura 6) onde estão incluídos todos os valores registados pelos múltiplos utilizadores. Neste relatório estão apresentados para todas as máquinas (ordenadas por departamento ou área de produção a que pertencem) quanto é que cada uma produziu para cada turno desse dia.

Mapa diário de Produção -					06/09/2019				
Turno	1	2	3	Total	Turno	1	2	3	Total
Equipa	A	B	C		Equipa	A	B	C	
Misturador 0	142	182	174	498	Módulo 01	54	137	2	193
Misturador 1	177	196	146	519	Módulo 02	408	528	500	1432
Misturador 2	106	0	0	106	Módulo 03	6	0	0	6
Misturador 3	140	128	115	383	Módulo 04	461	139	388	988
Misturador 4	114	154	151	419	Módulo 05	249	373	33	655
Misturador 5	144	129	161	434	Módulo 06	454	555	454	1463
Misturador 6	41	105	114	260	Módulo 07	0	0	0	0
Misturador 7	112	147	97	356	Módulo 08	382	461	448	1291
Misturador 8	72	125	117	314	Módulo 09	418	407	500	1325
Misturador 9	79	101	141	321	Módulo 10	515	459	500	1474
Misturador 10	155	163	124	442	Módulo 11	491	508	505	1504
Misturador 11	44	50	45	139	Módulo 12	411	455	328	1194
Strainer 1	64	80	75	219	Módulo 13	445	421	506	1372
Strainer 2	1390	1560	1480	4410	Módulo 14	537	399	495	1431
CARGAS	0	0	0	0 pn	Módulo 15	278	378	479	1135
Pisos E02	1675	4587	4600	10862 pn	Módulo 16	403	444	372	1219
Pisos E03	2725	5245	4950	12920 pn	Módulo 17	400	383	412	1195
Pisos E04	3725	0	1604	5329 pn	Módulo 18	334	440	452	1226
Pisos E05	4400	3855	4045	12300 pn	Módulo 19	214	371	485	1070
Pisos E06	4470	4380	4300	13150 pn	Módulo 20	492	439	515	1446
PISOS	16995	19087	19499	54581 pn	Módulo 21	420	450	411	1281
Paredes E01	11840	12660	11700	26200 pn	Módulo 22	347	422	413	1182
Paredes E02	7150	9325	0	12204 pn	Módulo 23	371	353	379	1103
Paredes E03	11700	11500	12000	26074 pn	Módulo 24	385	490	470	1345
PAREDES	30690	33485	23700	87875 pn	Módulo 25	500	440	511	1451
Borracha refinada (Kg)	1500	1442	0	2942	Módulo 26	385	343	280	1008
IL 1	9475	11127	10276	25732 pn	Módulo 27	361	317	342	1020
IL 2	12873	14310	12995	33482 pn	Módulo 28	289	331	292	912
IL 3	0	0	6669	5557 pn	Módulo 29	360	306	395	1061
CAMADAS	22348	25437	29940	64771 pn	Módulo 30	307	370	397	1044
CT1	6500	12300	10000	14400 pn	Módulo 31	380	115	383	878
CT2	14000	3300	5700	11500 pn	Módulo 32	328	217	301	844
CT3	19000	22000	17500	29250 pn	Módulo 33	335	315	369	1019
NUCLEOS TALOES	39500	37600	33200	55150 pn	Módulo 34	289	320	238	847
Apex 1	0	0	0	0 pn	Módulo 35	331	173	220	724
Apex 2	2800	3600	3800	5100 pn	Módulo 36	397	474	347	1218
Apex 3	3700	4300	2600	5300 pn	Módulo 37	305	260	252	817

Figura 6 - Pré-visualização parcial de um relatório de produção

De seguida, existem utilizadores do departamento de *Scheduling*, responsáveis por recolher o valor diário de vários indicadores de produção e programar as produções do dia seguinte.

Dados diários Data: 09/07/2019 **Produções** Menu FS

Programação Stocks às 8h Observações

Programados 100%	58 226
Moldes Planeados	466
Mix Greens	198
SUV's Planeados	3 499
Nr SUVs moldes	28
Nr Moldes U	3
Winter Planeados	31 436
UHP's Planeados	23 934
Nr Moldes UHP's	202
Mix UHP's	95
Stock Pneus Verde	23 018

Record: 4504 of 4690 No Filter Search

Figura 7 – Dados programados para o dia 09/07/2019

Após introduzir os valores planeados para o dia 09/07/2019 (Figura 7) o utilizador pressiona o botão “Produções” onde preenche as quantidades de produção (Figura 8) e de percentagem de lixo (SCRAP) do dia anterior (Figura 9).

Produção/Scrap 08/07/2019

Produção Scrap

Construidos	53 017	BudgetEntregas	52 200
Vulcanizados	53 590	Entregues	51 457
UHP Produzidos	22 708	Vendidos	60 055
Suvs Produzidos	3 451	Rework	952
Winter Produzidos	27 804	%Rework	1,78
		Yield	98,64
		Size Achievement	84,00

Figura 8 – Dados produzidos no dia 08/07/2019

Produção/Scrap

08/07/2019

Produção Scrap

Scrap I	0,06
Scrap II	0,51
Scrap III	0,17
Scrap IV	0,80
Scrap I-IV	1,55

Figura 9 – Percentagem de lixo (SCRAP) dos vários departamentos (I a IV)

Após o preenchimento destes dados o utilizador pode imprimir um relatório com toda a informação introduzida num certo dia.

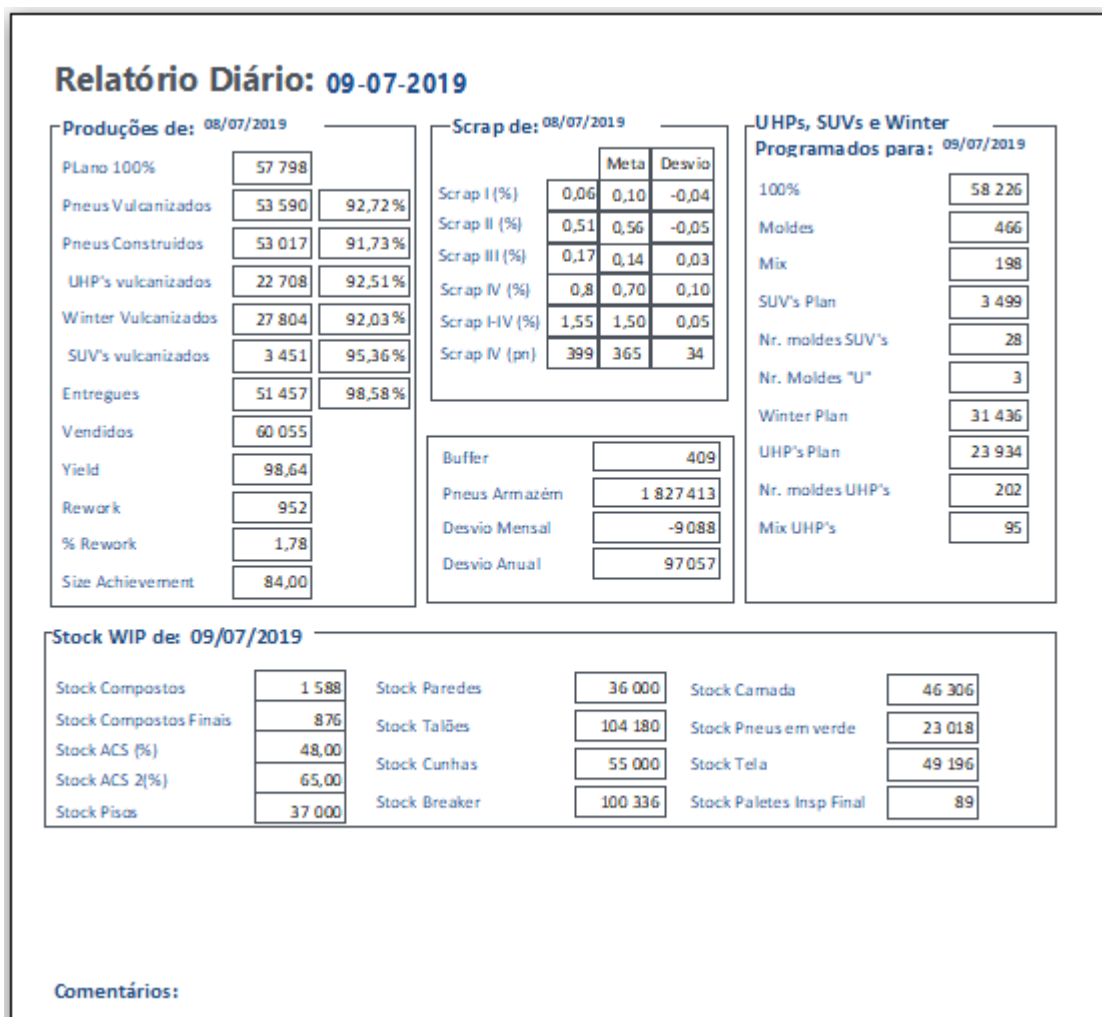


Figura 10 – Relatório Diário

Os utilizadores pertencentes ao *Scheduling* também preenchem, no final de um turno, um relatório de turno (Figura 11), onde são registados dados sobre o processo de misturação, vulcanização, construção e outras medidas importantes.

Relatório de Turno - Scheduling

Colaborador
4121 Bruno Sousa

Data
04/09/2019

Turno
2
B

Obj. Vulcanização (94% OEE)

52679

Mix

192

Average Stock

9,8

Standard Deviation

6,95

Estimated Losses

420

Standard Deviation (Pisos)

0

Average Stock (Pisos)

0

Relatório Vulcanização vulcanizados:17322

Med não programada NP:
PA:

Mud Previstas

0

Medidas não planead

Mud jante previstas

1

Mud jante realizadas

1

Relatório Preparação Várias retenções de pisos com conseqüente feiras para ajustar, exemplos: 1583; 409, 375; 847; 854; 306 - provocando várias entropias e ajustes na programação de forma a minimizar workoff.

Relatório Construção Stock:20670+37
- Vários ajustes na programação provocados pelas retenções de pisos.

Relatório SFI Nada a reportar.

Medidas				Módulos Críticos	Jantes Críticas
Prioritárias	Com atraso	Suspensas	Críticas		
				32, 33	16"

SA=

Suspensas=

UHP

Moldes

228

Medidas

109

⏪
⏴
▶
⏩
🏠
▶*
✖

Imprimir

📄

Figura 11 - Relatório de turno

Os utilizadores com o papel de Supervisor podem efetuar um registo da reunião da supervisão, onde os vários supervisores de cada área de produção se juntam no final de cada turno e registam dados relativos ao absentismo (numero de trabalhadores que não estavam presentes), avarias, diversos (paragens, entre outros), falta de materiais e ensaios. Ao mesmo tempo são mostrados dados relativos aos stocks, dados planeados e medidas, que são

introduzidas pelos outros utilizadores nas funcionalidades previamente mencionadas. (Figura 12)

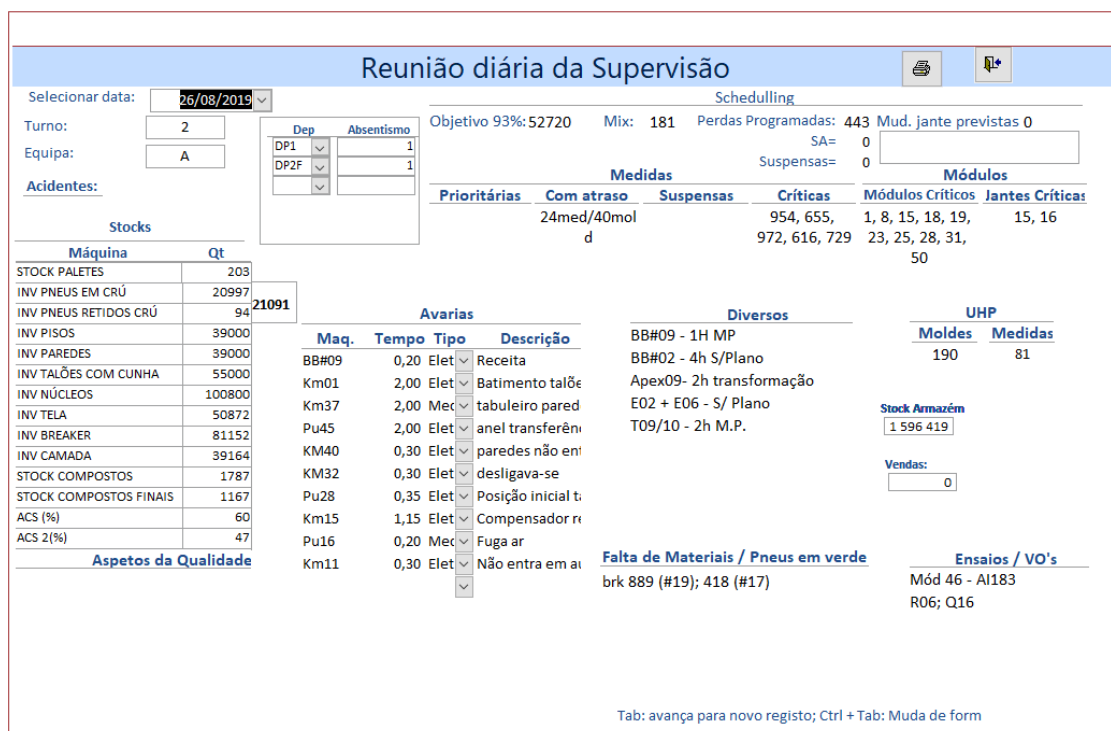


Figura 12 - Relatório diário de reunião da Supervisão

3.3 Aplicações semelhantes

Neste capítulo é feito um levantamento de soluções disponíveis no mercado com funcionalidades similares ao da aplicação desenvolvida para este projeto. Nas duas secções seguintes são apresentados os critérios de seleção e os critérios de avaliação das aplicações encontradas. De seguida são descritas as diferentes apps e no final estas são comparadas.

3.3.1 Critérios de seleção

Foram escolhidas aplicações de gestão de dados e geração de relatórios de acordo com os seguintes critérios:

- Popularidade de pesquisa no Google
- *Reviews* de blogs (Bello, 2019; Santos, 2020)
- Gratuita ou período de experimentação

Foram efetuadas pesquisas no Google utilizando os critérios acima referidos. Nas pesquisas foram utilizadas as palavras "best databse builder apps" e "top online database software".

Estas procuras resultaram na leitura de alguns blogues e websites que enumeravam algumas soluções e a partir destas e dos principais resultados do motor de busca as aplicações foram selecionadas.

3.3.2 Critérios de avaliação

Após o registo para demonstrações gratuitas foram determinados os seguintes critérios de avaliação:

- Criação de registos de dados;
- Criação de relatórios;
- Exportação para vários formatos;
- Imprimir relatórios;
- Gestão de utilizadores;
- Tem plataforma web;
- Qualidade da interface;
- Facilidade de uso;
- Preço;

3.3.3 HyperOffice Atlas

O Hyper Office Atlas(HyperOffice, 2020) é uma aplicação com o objetivo de fornecer espaços colaborativos para empresas. A app oferece, através de uma interface de *drag and drop*, a possibilidade de desenhar rapidamente espaços internos e externos. Nestes espaços é possível:

- Gerir documentos e ficheiros online;
- Partilhar calendários;
- Listas de Contacto;
- Coordenar Projetos;
- Criar Formulários, base de dados e relatórios.

Para esta descrição será mais focada na área de criar formulários e relatórios. Ao aceder à área de criação de aplicações é criada uma base dados por defeito. Para testar a aplicação foi criada uma tabela, um formulário e uma *view* (relatório).

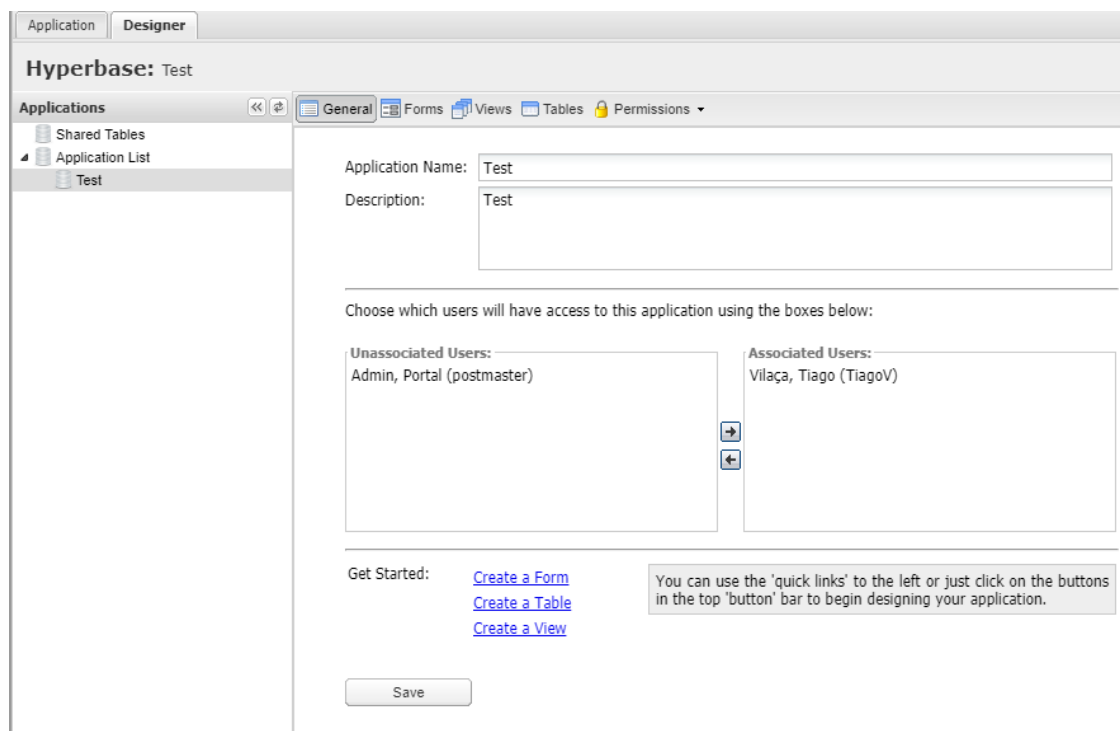


Figura 13 – Setup inicial

Ao criar a tabela (Figura 14) os campos *generatedID*, *createdBy*, *creationDate*, *LastUpdateDate* e *lastUpdatedBy* já se encontravam preenchidos por defeito. Foram adicionados os campos numéricos para registar valores produzidos pelas máquinas 1 a 4 e um texto para comentários. Na *tab* de permissões é possível gerir que utilizadores podem aceder a esta tabela.

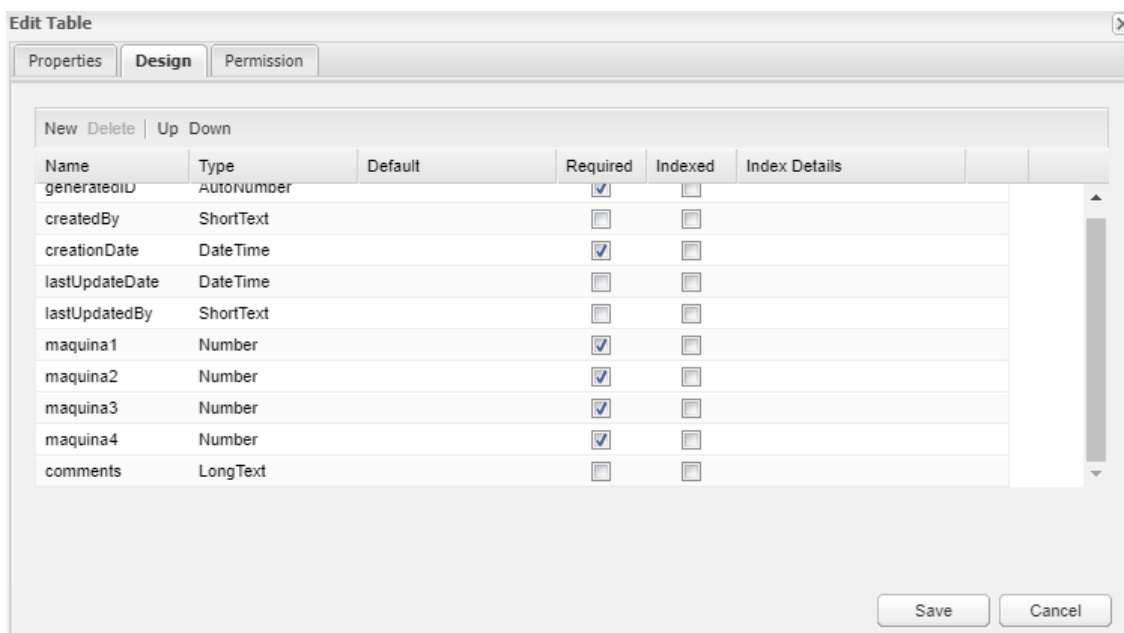


Figura 14 – Design da tabela de teste

De seguida, foi gerado um formulário para preenchimento da tabela com registos necessários.

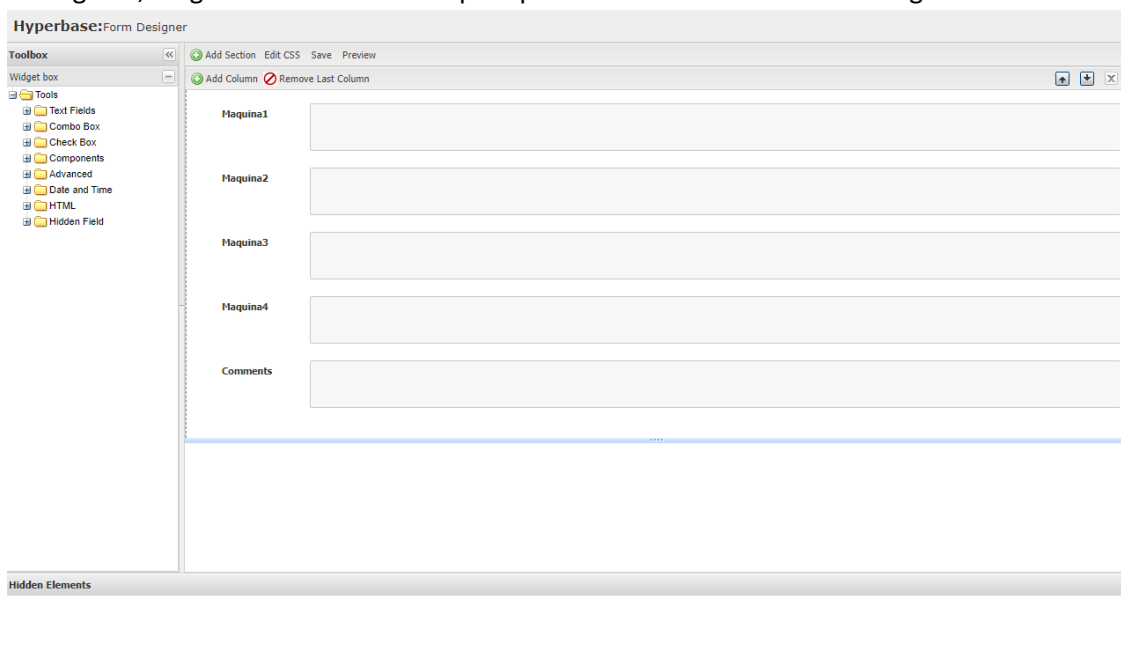


Figura 15 – Form Designer

Na Figura 15, é possível visualizar a ferramenta de desenho de um formulário. Trata-se de uma interface *drag-and-drop* onde é possível arrastar vários componentes diferentes a partir da sidebar "Toolbox" para o painel principal. Foram criados 5 campos de introdução para cada um dos elementos da tabela criada anteriormente.

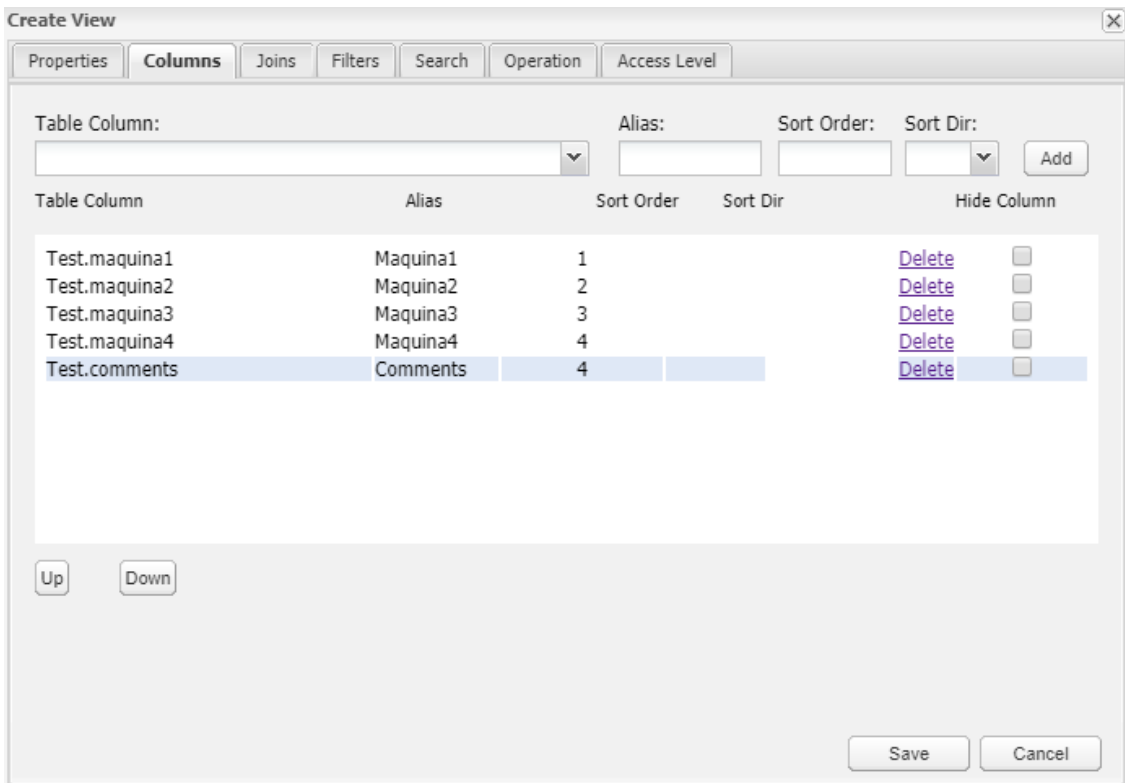


Figura 16 – Criação de uma view/report

Na parte de criação do relatório (Figura 16) é necessário indicar que tabela é que irá ser usada pelo relatório e que colunas incluir e a sua ordem. Também é possível criar *joins* entre tabelas para juntar dados de registos diferentes, aplicar filtros, pesquisas, operações de edição e remoção de registos e controle os acessos. Após estes passos a aplicação passa a estar acessível na lista de aplicações.

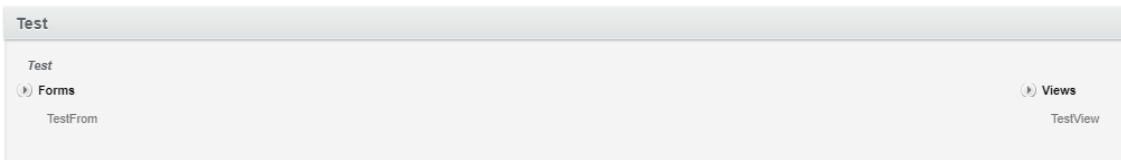


Figura 17 – Aplicação desenvolvida

Form: TestFrom

*maquina1: 10

*maquina2: 15

*maquina3: 20

*maquina4: 25

comments: All work well!

Page 0 of 0

Record ID: ** Clone Record

Share Pdf Submit

Figura 18 – Formulário

New Clone Edit Delete View Update | Printer Friendly Export Pdf Columns

Maquina1	Maquina2	Maquina3	Maquina4	Comments
10	15	20	25	All work well!
25	30	35	40	ALL OK!
50	55	60	0	Machine 4 failed!

Page 1 of 1 | Records on Page: 3 Total Records: 3

Close

Figura 19 – View

TestView

Maquina1	Maquina2	Maquina3	Maquina4	Comments
10	15	20	25	All work well!
25	30	35	40	ALL OK!
50	55	60	0	Machine 4 failed!

Figura 20- Relatório PDF gerado

A aplicação possui ferramentas que proporcionam ao utilizador liberdade para criar os formulários e relatórios de acordo com as suas necessidades. Permite exportar e imprimir os relatórios gerados (Figura 20). Como pontos negativos, nota-se que sem aplicar alguns conhecimentos *HTML* em algumas partes do processo, a interface gráfica apresenta um aspeto bastante básico. Este pode ser negativo para o cliente se este não tiver conhecimentos de programação.

3.3.4 Knack

Knack (Knack, 2020) é uma aplicação que fornece ferramentas para gerir dados numa base de dados online. Permite criar base de dados a partir de *templates* ou do princípio sem ser necessário qualquer tipo de código. A aplicação suporta:

- Procura de dados
- Formulários
- Mapas
- Relatórios
- Tabelas
- E-Commerce
- Calendários

A aplicação encontra-se dividida em duas partes. Uma chamada "*Data*" onde é possível gerir os dados registados na base de dados e outra denominada "*Pages*" onde se trata de gerir o processo de apresentação dos dados.

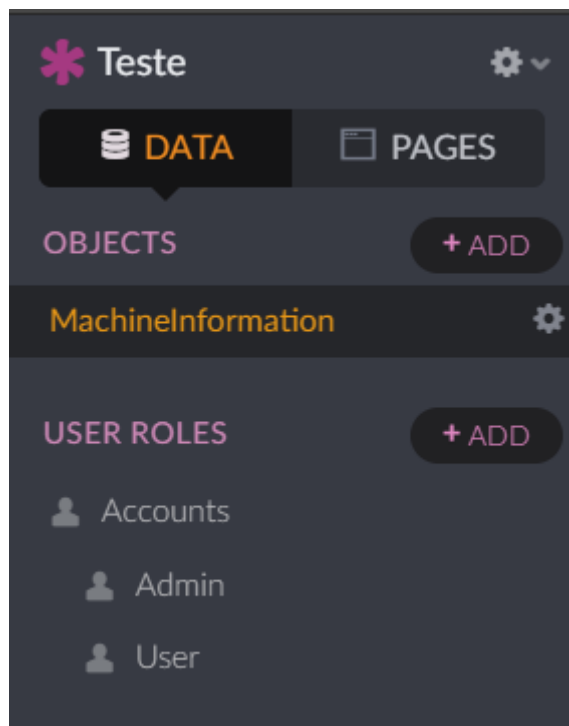


Figura 21 – Data

Na “Data” é possível criar objetos que representam registos na base de dados. Assim, foi gerado um objeto “MachineInformation” com as definições visíveis na Figura 22.

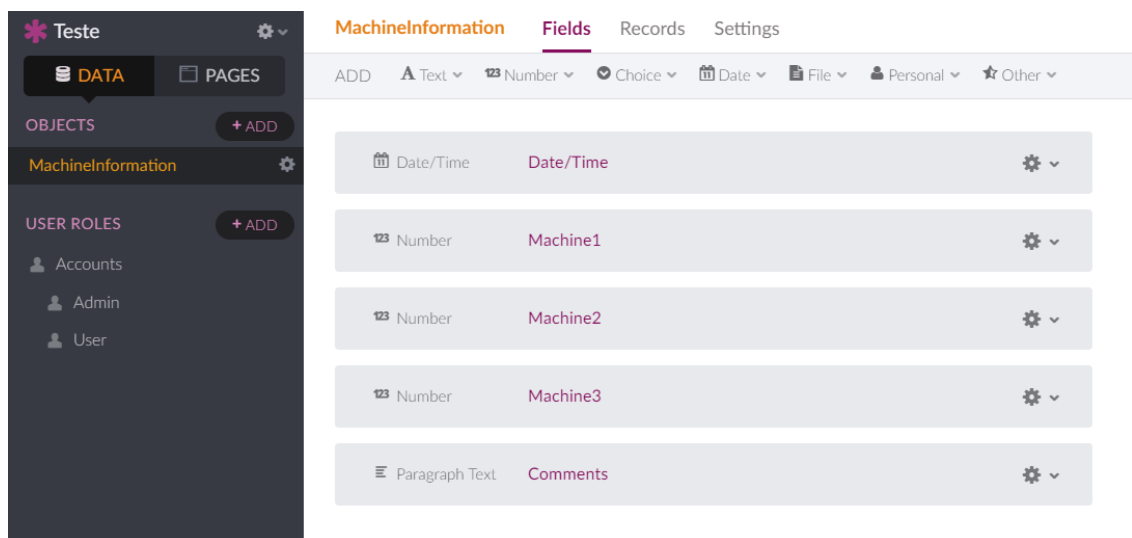


Figura 22 – Criação de um objeto

De seguida, na área “Pages” foi adicionada um novo formulário para introdução de dados.

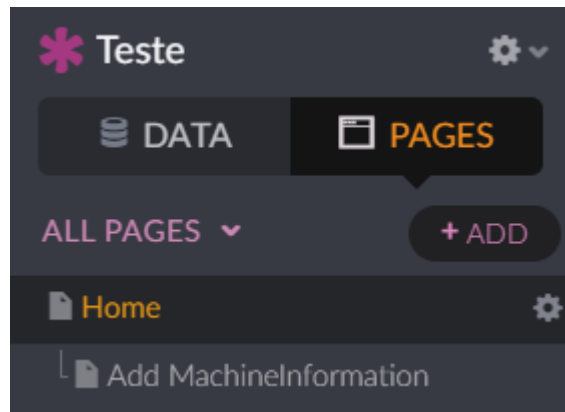


Figura 23 – Pages

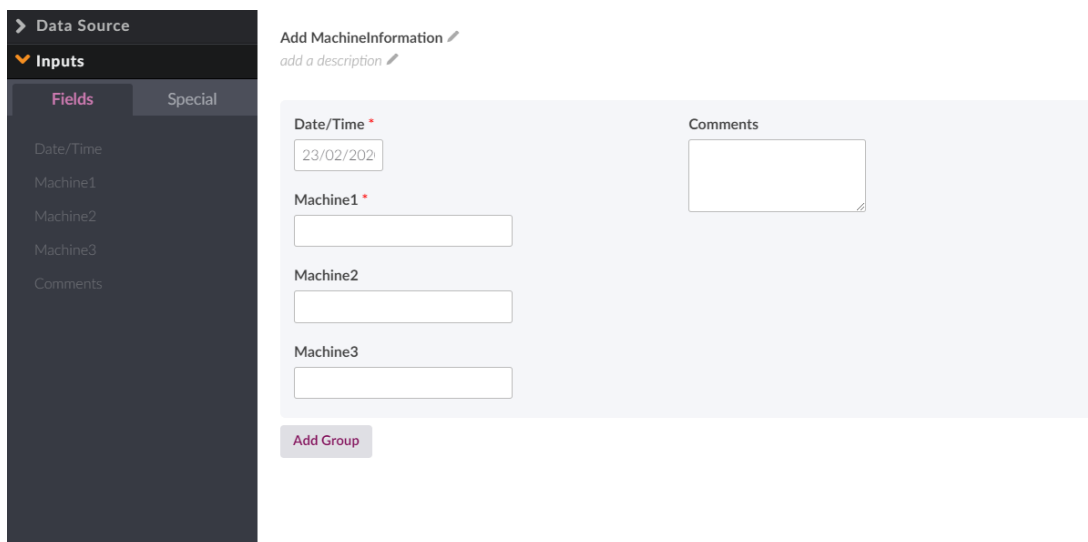


Figura 24 – Criação do formulário

A criação de formulário utiliza uma interface *drag-and-drop* onde os conteúdos da *sidebar* representam os componentes do objeto criado anteriormente. É possível seleccionar que campos adicionar a partir da lista “*Fields*”.

Quanto à criação de relatórios é possível adicionar tabelas, gráficos de barras, gráficos de área, *pie-charts*, entre outros. Para esta *demo*, foi criado um relatório com uma tabela e um gráfico de barras.

Edit Report ✕

Source **Report** Options

Data Layout

Table Rows

Date/Time options +

Pivot Columns

add column

Calculations

Machine1 Sum options - +

Machine2 Sum options - +

Machine3 Sum options - +

Row Summaries

add summary

Column Summaries

add summary

Preview
data is approximate [refresh](#)

Date/Time	Machine 1	Machine 2	Machine 3
23/02	10	15	
24/02	25	30	

Figura 25 –Tabela do relatório

Na Figura 25 foi criada uma tabela que utiliza o atributo data para mostrar os dados das máquinas introduzidos naquele dia.

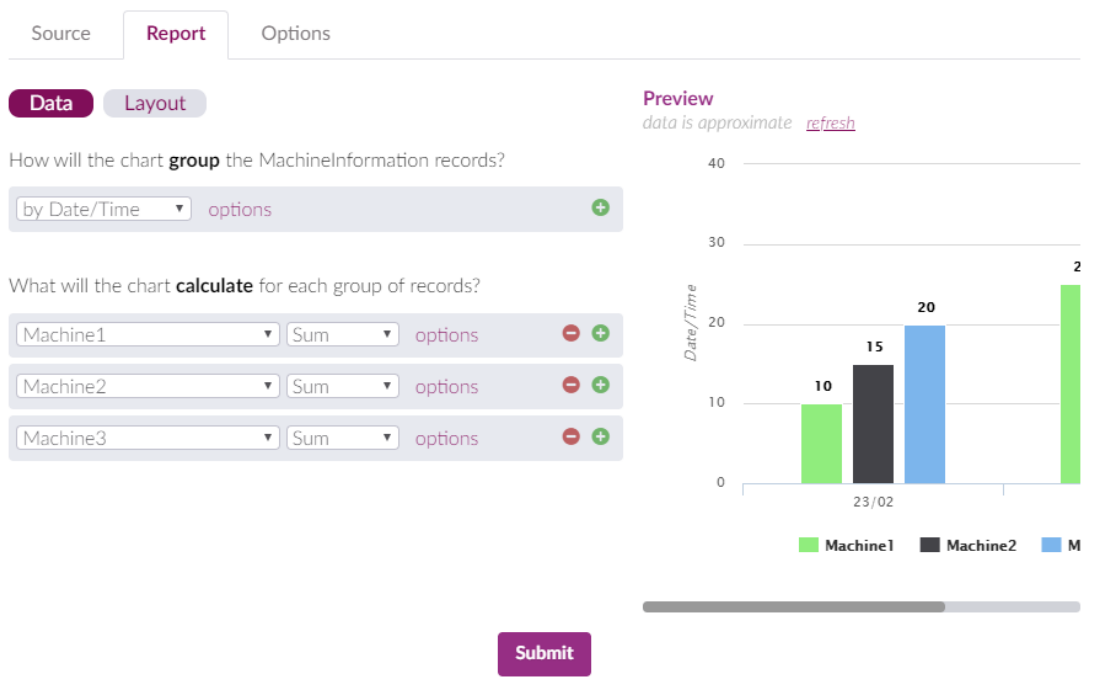


Figura 26 – Gráfico de barras

Na Figura 26, o gráfico mostra para cada data os valores registados de cada máquina.

Teste

[Home](#) > [Add MachineInformation](#) [Print](#)

Add MachineInformation

Date/Time *

Machine1 *

Machine2

Machine3

Comments

MachineInformation Report

Date/Time	Machine 1	Machine 2	Machine 3
23/02	10	15	20
24/02	25	30	35

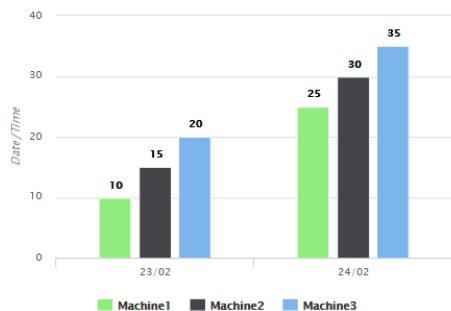


Figura 27 – Aplicação final

Na figura 27 é possível verificar o resultado da aplicação configurada anteriormente. Na parte superior, encontra-se o formulário de introdução de dados e na parte inferior o relatório de dados. No canto superior direito encontra-se um *link* para imprimir o relatório.

Esta aplicação apresenta uma interface apelativa e intuitiva durante todo o processo de *setup* da base de dados, formulário e relatório. No início do processo é apresentado um vídeo que introduz as funcionalidades da aplicação que ajudam bastante os utilizadores com menos experiência. Um fator negativo foi que não permite exportar o relatório para qualquer formato, apenas pode ser impresso.

3.3.5 Zoho Creator

O Zoho (Zoho, 2020) providencia aos seus clientes mais de 40 serviços diferentes para a gestão de empresas, entre eles, o Zoho Creator. Este serviço permite a criação de aplicações para gestão de dados e criação de relatórios.

Ao iniciar o processo de criação de uma aplicação, a primeira interface que surge é a de criação de um formulário. Neste caso, como resultado desta operação é criada o formulário de introdução de dados e uma tabela na base de dados para suportar o formulário.

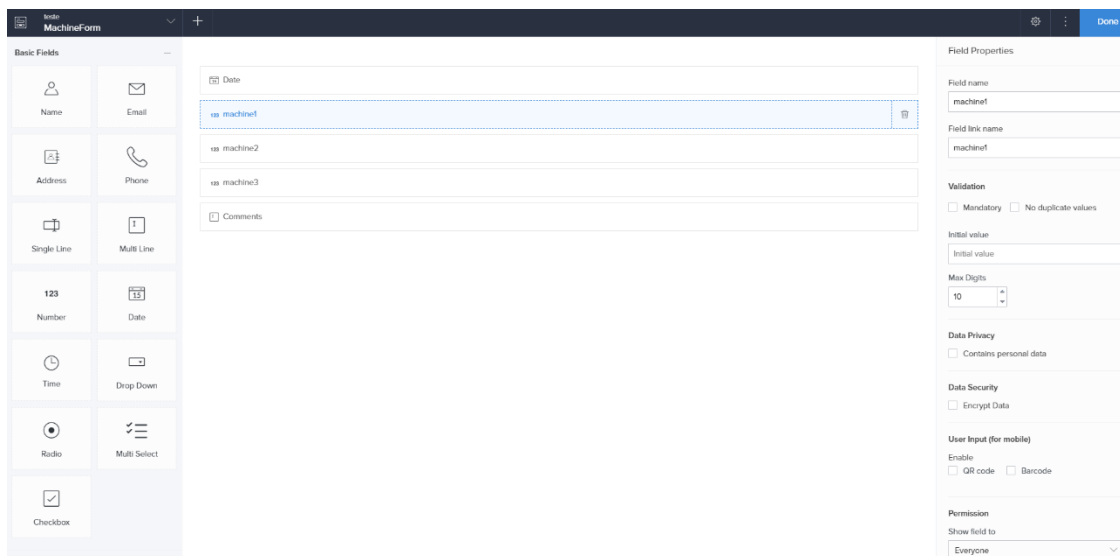


Figura 28 – Criação do formulário

Na Figura 28 é apresentada a interface de geração do formulário. É uma interface *drag-and-drop* onde na *sidebar* da esquerda estão disponíveis vários tipos diferentes de componentes de introdução de dados. Na *sidebar* da direita, são apresentadas as várias propriedades de um campo de introdução, tais como, se é um campo obrigatório, restrições, se é necessário aplicar encriptação, gestão de permissões entre outros.

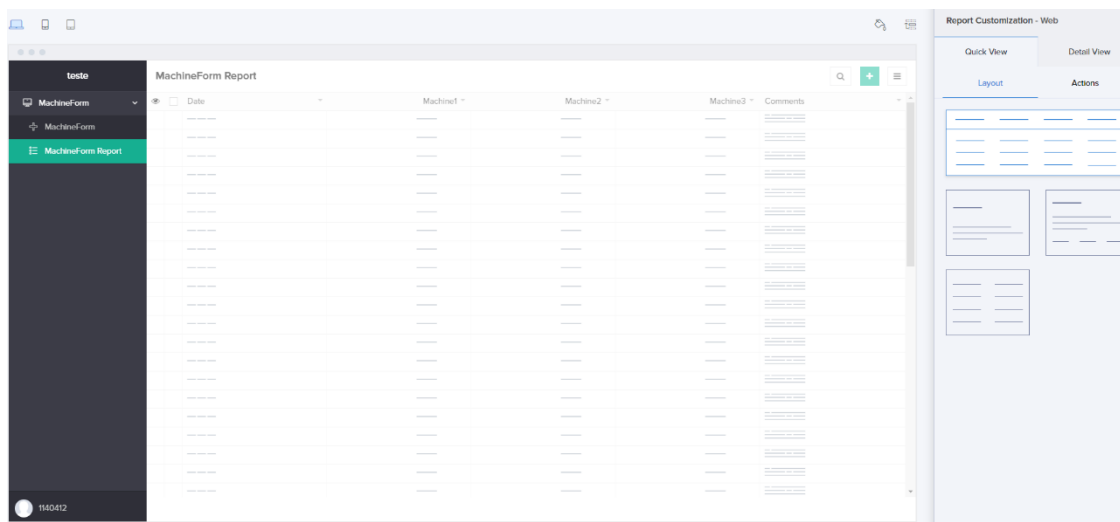


Figura 29 – Criação do relatório

Quanto à geração do relatório, como é possível verificar na Figura 29, apenas é necessário seleccionar o modelo a seguir para representar dados, localizados na *sidebar* da direita. Nesta *sidebar* também é possível definir ações de consulta, edição ou eliminação de dados. Também

é possível definir uma vista detalhada que aparece quando é efetuado um clique em cima de um registo. O sistema suporta temas e seleção de ícones próprios do Zoho.

Figura 30 – Formulário final

Date	Machine1	Machine2	Machine3	Comments
25-Feb-2020	50	55	0	Machine 3 failed!
24-Feb-2020	25	30	35	OK!
23-Feb-2020	10	15	20	OK!

Figura 31 – Relatório final

MachineForm Report

Date	Machine1	Machine2	Machine3	Comments
25-Feb-2020	50	55	0	Machine 3 failed!
24-Feb-2020	25	30	35	OK!
23-Feb-2020	10	15	20	OK!

Figura 32 – Relatório PDF

Nas figuras x e y são apresentados os resultados da aplicação construída durante a demo. No relatório, ao clicar no botão de opções para poder exportar dados para os formatos PDF, HTML, JSON, CSV e TSV. Também é possível importar e imprimir dados.

A solução da Zoho apresenta uma interface intuitiva e de fácil utilização. O processo de criação do formulário pode ser efetuado por um utilizador sem experiência de programação. Como ponto negativo, nota-se a falta de liberdade na criação do relatório. É apenas possível escolher dos *templates* pré-definidos o que pode levar a que um cliente não possa satisfazer totalmente as suas necessidades.

3.3.6 Comparação e Conclusão

Após a análise das várias soluções foi preenchida a Tabela 1 de acordo com os critérios de avaliação definidos em 3.2.2.

Tabela 1 – Resultados dos critérios de avaliação

	HyperOffice Atlas	Knack	Zoho Creator
Gestão de base de dados	Sim	Sim	Não
Criação de formulários	Sim	Sim	Sim
Criação de relatórios	Sim	Sim	Sim
Exportação para vários formatos	Sim	Não	Sim
Imprimir relatórios	Sim	Sim	Sim
Gestão de utilizadores	Sim	Sim	Sim
Qualidade da interface	Média	Boa	Boa
Plataforma web	Sim	Sim	Sim
Facilidade de uso	Médio	Boa	Boa
Preço/mês	50\$	39€	10€

Analisando as funcionalidades, a app da Zoho é a única que não apresenta ferramentas para gestão da base de dados enquanto que a aplicação da Knack é a única que não suporta exportação dos relatórios para vários formatos. Quanto à qualidade da interface e facilidade de uso, a HyperOffice foi classificada com um médio devido à complexidade de opções

disponíveis, tornando o processo de *setup* mais difícil. Em termos de preço a HyperOffice Atlas é a mais cara e a Zoho Creator a mais barata. Isto deve-se à quantidade de serviços oferecidos por cada uma das aplicações, sendo este número muito maior no caso da HyperOffice.

Comparando com a aplicação preexistente, seria possível replicar as funcionalidades desta nas aplicações encontradas no mercado, visto que todas são plataformas dedicadas à gestão de dados. A aplicação preexistente, sendo desenvolvida utilizando MSAccess apresenta algumas vantagens:

- Fácil integração com outras aplicações da Microsoft. O grupo Continental utiliza maioritariamente sistemas fornecidos pela Microsoft tais como Office365 e Teams;
- Maior controlo dos dados. A aplicação preexistente permite ao utilizador acesso direto às tabelas onde estão guardados os dados, enquanto que nas aplicações disponíveis no mercado as bases de dados se encontram fora do alcance da empresa;

A aplicação possui também algumas desvantagens:

- Com o aumento da quantidade de informação nas tabelas a aplicações começa a mostrar alguma degradação na velocidade de resposta;
- Aplicação desenvolvida por alguém sem conhecimentos de desenvolvimento de software pode levar a falhas no futuro.

Concluindo, este processo de análise das soluções foi fundamental para perceber as principais funcionalidades das aplicações do mercado de gestão de dados empresariais. Apesar de se ter decidido desenvolver uma nova aplicação internamente, ao identificar os pontos fracos e fortes de cada aplicação disponível no mercado e preexistente, esta análise é crucial para a implementação da solução pretendida pela Continental Mabor.

4 Design

Neste capítulo serão apresentados os requisitos funcionais e não funcionais levantados. A partir deste levantamento resultarão as funcionalidades a implementar da aplicação. Serão também apresentadas as várias arquiteturas consideradas e finalmente, o modelo de domínio.

4.1 Requisitos

O levantamento de requisitos foi efetuado após a análise extensa da aplicação preexistente (descrita no capítulo 1.2.1), em termos de funcionalidades disponíveis para o utilizador e também dos dados guardados nesta. Foram também realizadas reuniões com o principal utilizador da aplicação, o responsável do departamento de produção, para descrever as funcionalidades a adicionar, manter e alterar. Todos os requisitos estão presentes num documento SRS (Hécio A. Soares and Raimundo S. Moura, 2015)(Anexo 1).

Nesta secção são identificados os *stakeholders* e os atores do sistema e são descritos os requisitos funcionais e não funcionais. Estes requisitos encontram-se definidos de forma resumida e podem ser consultados com maior detalhe no documento disponível no Anexo 1.

4.1.1 Stakeholders

Foram identificados os seguintes *stakeholders*:

- **Continental Mabor:** Fornecer uma plataforma de gestão de indicadores de produção atualizada aos seus colaboradores
- **Utilizador da aplicação:** Interesse em usufruir de uma nova e atualizada aplicação
- **Administrador do sistema:** Interesse em gerir o sistema e os dados guardados neste

- **Estagiário/Desenvolvedor:** Interesses em desenvolver uma nova aplicação e aprofundar o seu conhecimento

4.1.2 Requisitos funcionais

Nesta secção são apresentados todos os casos de uso e os seus atores. É também definida a prioridade e uma curta descrição de cada caso. Na Figura 33 é apresentado o diagrama de casos de uso desta aplicação.

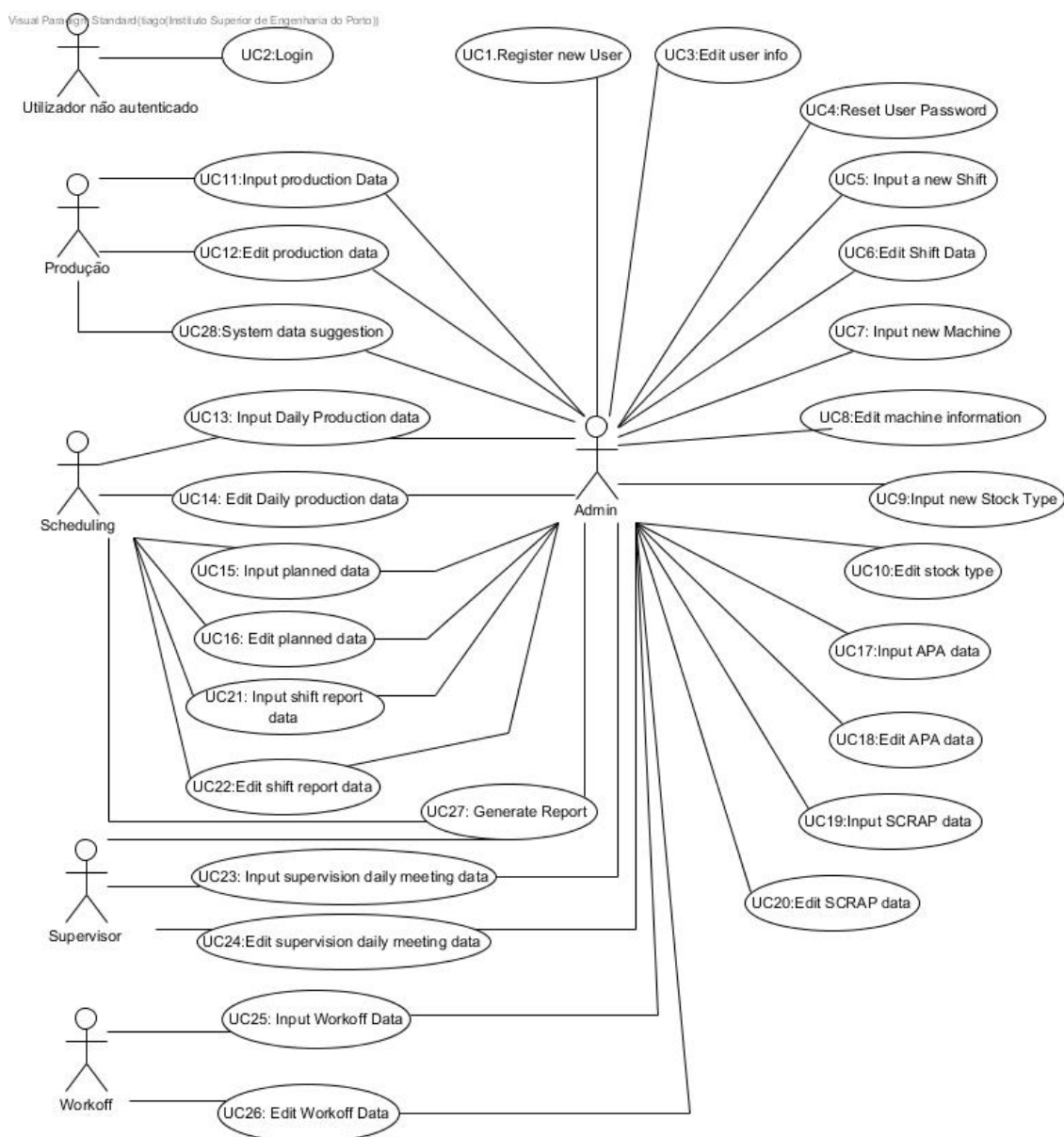


Figura 33 – Diagrama de casos de uso

Na tabela seguinte são apresentados todos os casos de uso e a sua prioridade em termos de o que deve ser implementado em primeiro lugar.

Tabela 2 – Casos de uso e prioridade

Caso de uso	Prioridade
UC1: Registrar novo utilizador	1
UC2: Efetuar login	1
UC3: Editar dados de utilizadores	2
UC4: Mudança de password	2
UC5: Introduzir novo turno	3
UC6: Editar dados de turnos	1
UC7: Registrar nova máquina	1
UC8: Editar dados de máquinas	1
UC9: Registrar novo tipo de Stock	1
UC10: Editar tipo de Stock	1
UC11: Introduzir dados de produção	1
UC12: Editar dados de produção	2
UC13: Introduzir dados de produção diária	1
UC14: Editar dados de produção diária	2
UC15: Introduzir dados de programação	1
UC16: Editar dados de programação	2
UC17: Introduzir dados de APA	2
UC18: Editar dados da APA	2
UC19: Introduzir dados de SCRAP	1
UC20: Editar dados de SCRAP	2
UC21: Introduzir dados do relatório de turno	1
UC22: Editar dados de relatório de turno	2
UC23: Introduzir dados da reunião de supervisão	1
UC24: Editar dados da reunião de supervisão	2
UC25: Introduz dados de Workoff	2
UC26: Editar dados de Workoff	2
UC27: Gerar relatórios	2
UC28: Sugerir dados de preenchimento	3

4.1.2.1 UC1: Registrar novo utilizador

O utilizador administrador efetua um registo de um novo utilizador, introduzindo o *username*, email interno, *role* e password por defeito. Esta password será alterada da próxima vez que o utilizador efetuar o login.

4.1.2.2 UC2: Efetuar login

O utilizador não autenticado efetua o login com as credenciais providenciadas pelo administrador. Se este for o seu primeiro login, o utilizador terá de alterar a sua palavra passe.

4.1.2.3 UC3: Editar dados de utilizadores

O administrador pode gerir as informações dos utilizadores do sistema. Este pode alterar os dados e permissões dos outros utilizadores.

4.1.2.4 UC4: Mudança de password

O administrador pode alterar a *password* de um utilizador, em caso de esquecimento, para uma com um valor por defeito definido anteriormente. O utilizador, ao efetuar o processo de login, é obrigado a alterar a *password*.

4.1.2.5 UC5: Introduzir novo turno

O utilizador administrador pode registar um novo turno no sistema. O administrador introduz a data, número do turno e equipa. O registo do turno também pode ser efetuado de forma automática pelo sistema quando um utilizador efetua o processo de login.

4.1.2.6 UC6: Editar dados de turnos

O administrador pode gerir os dados relativos aos turnos registados no sistema. O utilizador pode criar um turno no sistema e editar turnos existentes.

4.1.2.7 UC7: Introduzir nova máquina

O utilizador administrador regista uma nova máquina no sistema. A administrador introduz o nome da máquina, tipo, valor de divisão e o departamento a que esta pertence.

4.1.2.8 UC8: Editar dados das máquinas

O administrador pode editar os dados das máquinas registadas no sistema. Neste caso de uso o utilizador pode desativar uma máquina para que esta não seja utilizada em processos de registo de dados de produção e relatórios.

4.1.2.9 UC9: Introduzir novo tipo de Stock

O utilizador administrador regista um novo tipo de stock no sistema. A administrador introduz o nome e o departamento a que esta pertence.

4.1.2.10 UC10: Editar novo tipo de Stock

O administrador pode editar os dados das máquinas registadas no sistema. Neste caso de uso o utilizador pode desativar uma máquina para que esta não seja utilizada em processos de registo de dados de produção e relatórios.

4.1.2.11 UC11: Introduzir dados de produção

O utilizador de produção/administrador introduz dados produzidos pelas máquinas e stock gerado num dado turno.

4.1.2.12 UC12: Editar dados de produção

O utilizador de produção/administrador pode editar dados de produção para um turno.

4.1.2.13 UC13: Introduzir dados de produção diária

O utilizador de *Scheduling*/administrador introduz dados de produção diária. Estes dados incluem os valores planeados para o dia presente, valores produzidos e de SCRAP do dia anterior.

4.1.2.14 UC14: Editar dados de produção diária

O utilizador de *Scheduling*/administrador pode editar dados de produção diária.

4.1.2.15 UC15: Introduzir dados de programação

O utilizador de *Scheduling*/administrador introduz dados relativos à programação da quantidade de pneus a produzir.

4.1.2.16 UC16: Editar dados de programação

O utilizador de *Scheduling*/administrador pode editar dados de programação.

4.1.2.17 UC17: Introduzir dados de APA

O utilizador administrador introduz os dados relativos ao stock no armazém APA e Buffer.

4.1.2.18 UC18: Editar dados de APA

O utilizador administrador pode editar os dados relativos ao stock no armazém APA e Buffer.

4.1.2.19 UC19: Introduzir dados de SCRAP

O utilizador administrador introduz os dados relativos ao SCRAP (lixo) produzido por cada departamento. O utilizador pode registar as constantes de relativas às metas de SCRAP de cada departamento para um ano.

4.1.2.20 UC20: Editar dados de SCRAP

O utilizador administrador edita os dados relativos ao SCRAP (lixo) produzido por cada departamento. O utilizador pode também editar as constantes de relativas às metas de SCRAP de cada departamento para um ano.

4.1.2.21 UC21: Introduzir dados do relatório de turno

O utilizador de *Scheduling*/administrador introduz dados referentes ao relatório de turno. Estes dados incluem relatórios de ocorrências dos vários departamentos de produção.

4.1.2.22 UC22: Editar dados de relatório de turno

O utilizador de *Scheduling*/administrador pode editar dados do relatório de turno.

4.1.2.23 UC23: Introduzir dados da reunião de supervisão

O utilizador supervisor/administrador introduz dados da reunião da supervisão. Estes dados são registos de ocorrências como, acidentes, avarias, etc.

4.1.2.24 UC24: Editar dados da reunião de supervisão

O utilizador supervisor /administrador pode editar dados da reunião da supervisão.

4.1.2.25 UC25: Introduz dados de Workoff

O utilizador da *Workoff*/administrador introduz dados relativo ao consumo de *Workoff*.

4.1.2.26 UC26: Editar dados de Workoff

O utilizador da *Workoff* /administrador pode editar dados de *Workoff*.

4.1.2.27 UC27: Gerar relatórios

O utilizador administrador/superintendente/*schedulling* seleciona a data e o tipo de relatório o servidor de *reports* gera o relatório pretendido.

4.1.2.28 UC28: Sugerir dados de sistema

O utilizador da produção/administrador pode, antes de ser efetuado qualquer registo definitivo no sistema, ativar uma opção onde o sistema acede a uma base de dados dos sistemas internos da Continental Mabor para preencher automaticamente os campos do registo dos dados de produção. (4.1.2.11). O utilizador pode aceitar ou alterar estes dados antes destes serem gravados na base de dados.

4.1.3 Requisitos não funcionais

Os requisitos não funcionais são requisitos que não são funcionalidades diretamente relacionadas com o utilizador, mas são igualmente importantes para o bom funcionamento da aplicação.

- **Performance:** O sistema deve ter um tempo de resposta curto e não deve ser afetado pela carga presente no sistema;
- **Segurança:** O sistema deve providenciar um sistema de autenticação e autorização. Deve também garantir a confidencialidade dos dados introduzidos pelo utilizador, garantindo o GRPD;
- **Adaptabilidade:** O sistema deve apresentar uma interface adaptável a diferentes utilizadores.
- **Usabilidade:** O sistema deverá possuir uma interface intuitiva. Deverá também prevenir erros, pedindo sempre confirmação em operações importantes;
- **Confiabilidade:** O sistema deverá possuir tolerância a falhas;
- **Suportabilidade:** O sistema deverá suportar múltiplas linguagens, numa primeira fase em português e inglês;
- **Escalabilidade:** O sistema deverá ser implementado tendo em conta futuras alterações e novas funcionalidades.

4.2 Modelo de domínio

O modelo de domínio é um diagrama onde é apresentada a lógica de negócio através das várias entidades do sistema. A aplicação trata-se de um sistema de registo de dados de produção e outros indicadores importantes para medir o desempenho da fábrica. Na Figura 34 está apresentado o modelo de domínio para esta projeto, seguido de uma descrição de cada entidade representada.

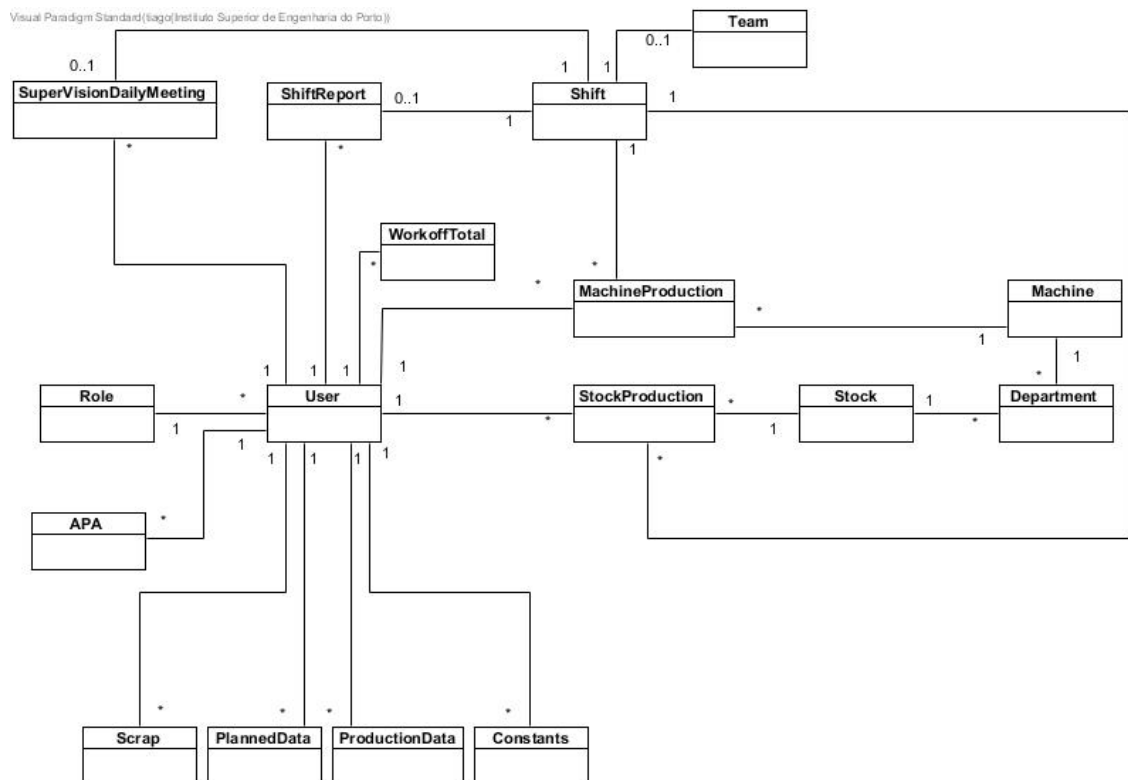


Figura 34 – Modelo de domínio

Na figura 34 é possível identificar as seguintes entidades:

- **User:** Utilizador do sistema que efetua registos no sistema. O *user* é registado sempre que este introduz ou edita dados no sistema;
- **Role:** Papel do utilizador no sistema;
- **Shift:** Representa um turno no sistema;
- **Scrap:** Valor percentual de material residual do processo de produção de um pneu produzido por cada departamento num determinado dia;
- **Planned Data:** Conjunto de valores planeados de produção para um certo dia. Exemplo: número de pneus a produzir para amanhã;
- **Production Data:** Conjunto de valores produzidos para um certo dia; Exemplo: Número de pneus vendidos;
- **Constants:** Definição de constantes usadas nos relatórios;
- **Shift Report:** Contem todos os dados de um relatório de turno. Este relatório inclui dados de quanto pneus foram vulcanizados e algumas ocorrências nos vários processos de produção de um *Shift*;
- **SuperVision Daily Meeting:** Contem todos os dados do relatório de reunião diária da supervisão. Esta entidade possui dados referentes a avarias, faltas de materiais ensaios para um determinado *Shift*;
- **Team:** Representa a equipa que está a trabalhar num certo *Shift*.
- **Workoff:** Representa dados de *workoff* de um determinado dia;

- **Stock:** Representa um tipo de stock controlado por um departamento. Exemplo: Pneus contruídos no DPIII, pneus vulcanizados no DPV;
- **StockProduction:** Quantidade de um tipo de stock produzida num *Shift*. Exemplo: 2000 pneus contruídos no turno 1;
- **Machine:** Representa uma máquina pertencente a um departamento. Exemplo: Máquina Misturadora1 do departamento DPI;
- **Machine production:** Quantidade produzida por uma máquina num certo *Shift*. Exemplo: Máquina misturadora1 produziu 100 Kg no turno 2;
- **Departament:** Representa um departamento ao qual são associados máquinas e tipos de stock.

4.3 Arquitetura

Nesta secção são definidas várias alternativas para a arquitetura do sistema e após análise de cada uma será apresentada a escolhida e esta definida em maior detalhe.

4.3.1 Arquiteturas propostas

Nas seguintes secções são identificadas as alternativas definidas para a arquitetura, através de diagramas de componentes de alto nível e é fornecida uma explicação para cada uma.

4.3.1.1 Alternativa 1 – Arquitetura Simples

Na Figura 35 está representada a primeira alternativa. Trata-se de uma arquitetura simples, onde a camada de negócio e a interface gráfica se encontram localizadas no mesmo componente e este liga-se diretamente à base de dados.

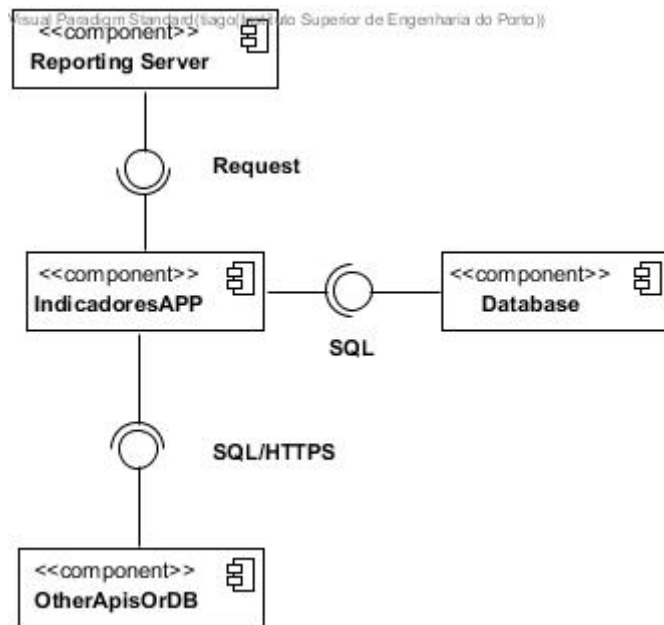


Figura 35 – Arquitetura simples.

4.3.1.2 Alternativa 2 – Baseada em Web APIs

Na Figura 36 encontra-se definida a segunda alternativa.

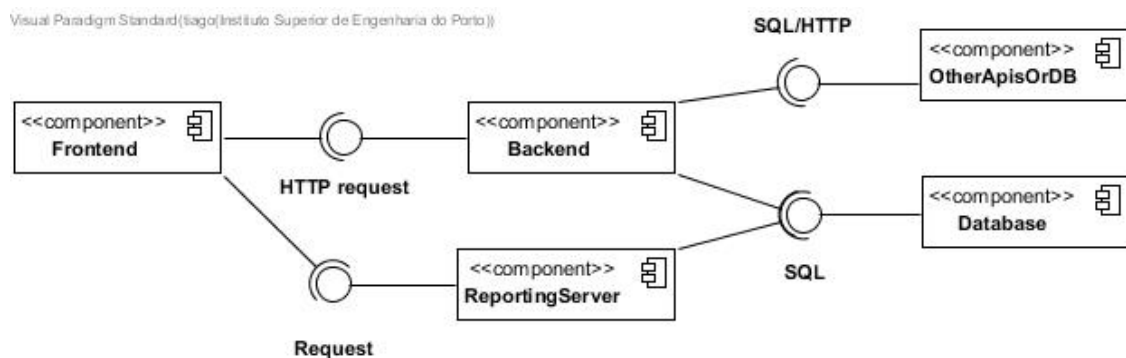


Figura 36 – Arquitetura baseada em Web APIs.

É possível diferenciar 5 componentes diferentes:

- **Frontend:** Aplicação que serve de interface com o utilizador;
- **Backend:** Aplicação responsável pela lógica de negócio. É também responsável por comunicar com a base de dados (SQL) e por providenciar uma interface HTTP para comunicar com o *frontend*;
- **Reporting Server:** Aplicação responsável por aceder à base de dados e gerar relatórios. Estes relatórios serão apresentados no *frontend*;
- **Database:** Base de dados responsável por garantir a persistência de dados

- **OtherApisOrDb:** APIs ou Bases de Dados internas da organização para consulta de dados

4.3.1.3 Alternativa 3 – Baseada em micro serviços

Na Figura 38 encontra-se definida a terceira alternativa. Esta alternativa é baseada em micro serviços. Ao contrário da segunda alternativa, o componente *BackendAPI* é dividida em vários serviços, um para cada grande funcionalidade do sistema (*SupervisionMeetingService*, *DailyProductionReportService*, *ProductionService*, *ShiftReportService*, *AuthService*). Seguindo o padrão *Database per Service*, cada serviço possui a sua própria base de dados onde apenas serão guardados dados relativos à função definida para o serviço. Por exemplo o *ProductionService* apenas regista dados relativos à produção das máquinas de cada turno e o *AuthService* é responsável por gerir os dados dos utilizadores e os processos de login. Segundo o padrão *API Gateway*, o componente *ServiceGateway* coordena todos os serviços a partir de um único ponto de entrada.

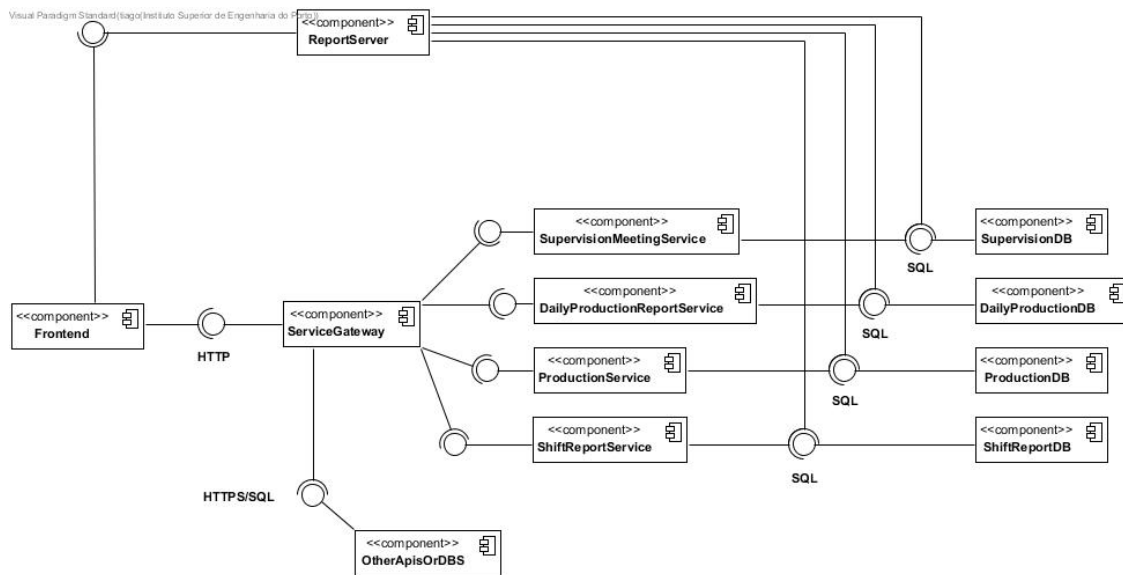


Figura 37 – Arquitetura baseada em micro serviços.

4.3.2 Arquitetura escolhida

Analisando a alternativa 1, trata-se de uma arquitetura monolítica. Este tipo de arquitetura apesar de apresentar vantagens em termos de baixo tempo de desenvolvimento e baixa complexidade de implementação, foi rejeitada devido ao elevado número de desvantagens. A camada da lógica de negócio e a interface gráfica estariam acopladas na mesma aplicação

tornando manutenção, alteração e adição de novas funcionalidades um processo mais complicado.

De seguida, a alternativa 2, trata-se de uma arquitetura baseada em APIs. Esta arquitetura apresenta algumas vantagens em relação à alternativa 1. Remove a lógica de negócio da camada da interface gráfica, fácil implementação e várias *frameworks* e IDEs suportam este tipo de desenvolvimento. Contudo, apresenta algumas desvantagens, tal como, quando a API atinge uma quantidade elevada de código é mais difícil efetuar manutenção e alterações e, pode também, levar a uma sobrecarga da API servidora. Com o maior tamanho da aplicação pode vir também a necessidade de dividir a equipa de desenvolvimento em várias com diferentes responsabilidades, o que pode resultar numa maior dependência na colaboração entre *developers* para desenvolver novas funcionalidades. Em caso de falha da API servidora todo o sistema fica inoperacional até a recuperação deste.

A alternativa 3 apresenta uma arquitetura baseada em micro serviços. Ao utilizar este tipo de arquitetura são removidas algumas desvantagens apresentadas na alternativa 2. Ao dividir as funcionalidades da aplicação servidora em vários micro serviços, descentraliza-se a carga no sistema e torna-se mais fácil a manutenção e adição de novas funcionalidades. Em caso de falha de um serviço, os outros mantem-se operacionais mantendo sempre o sistema operacional. Algumas desvantagens deste tipo de arquitetura incluem a maior complexidade de implementação, principalmente notável quando um serviço depende de outro. É necessário a implementar um método de comunicação entre serviços e seria necessária uma coordenação planeada entre as equipas responsáveis pelos serviços.

Assim, rejeitando a alternativa 1, resta apenas decidir entre a 2 e a 3. Não existe uma solução perfeita para este tipo de problema, ambas as opções apresentam vantagens e desvantagens válidas que levem à sua aceitação ou rejeição. No entanto foi rejeitada parcialmente a alternativa 3, pois não se justificava uma implementação de micro serviços para um projeto deste tamanho. Embora, com o passar do tempo, a aplicação possa crescer em tamanho poderá ser considerada uma mudança para uma implementação com micro serviços.

4.3.3 Especificação da arquitetura selecionada

Após a definição de alto nível da arquitetura do sistema, é necessário aprofundar a arquitetura a um nível mais baixo. Nesta secção são apresentados os diagramas de componentes para o *Backend e Frontend*, e o diagrama de implementação da solução.

4.3.3.1 Frontend

A estrutura do *frontend* encontra-se apresentada na Figura 38. A principal função do componente de *frontend* é a interação com o utilizador através de uma interface gráfica e comunicação com servidor para visualização e introdução dos dados pelo utilizador.

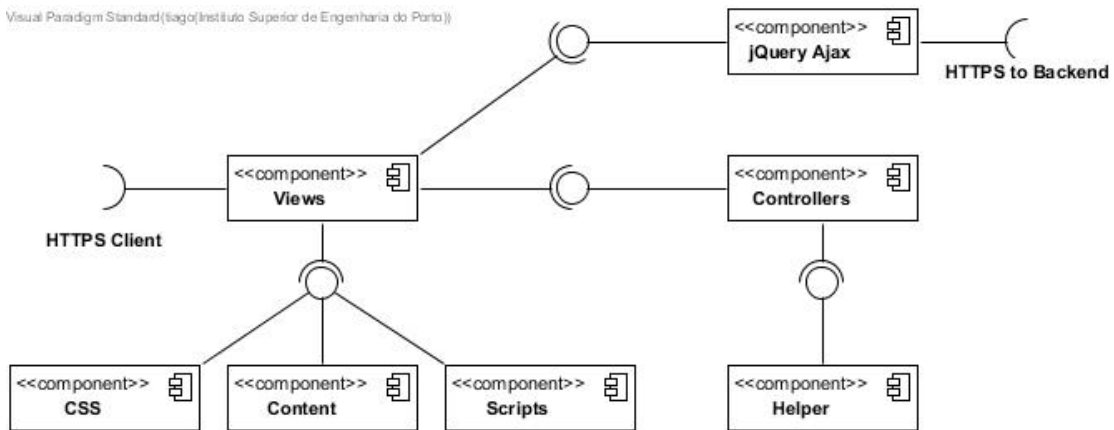


Figura 38 – Diagrama de componentes *Frontend*

Na Figura 38 estão presentes os seguintes componentes:

- **View:** Responsável por fornecer a interface gráfica ao utilizador;
- **CSS, Content, Scripts:** Assistem a *View* no processo de construção da interface gráfica
- **Controller:** Gerir dados para a *View*
- **Helper:** Assistir o *Controller*
- **Jquery Ajax:** Responsável por enviar pedidos para a API *Backend*

4.3.3.2 *Backend*

Na Figura 39 é apresentada a estrutura do *backend*. O *backend* é responsável por gerir os dados recebidos do *frontend* e da base de dados.

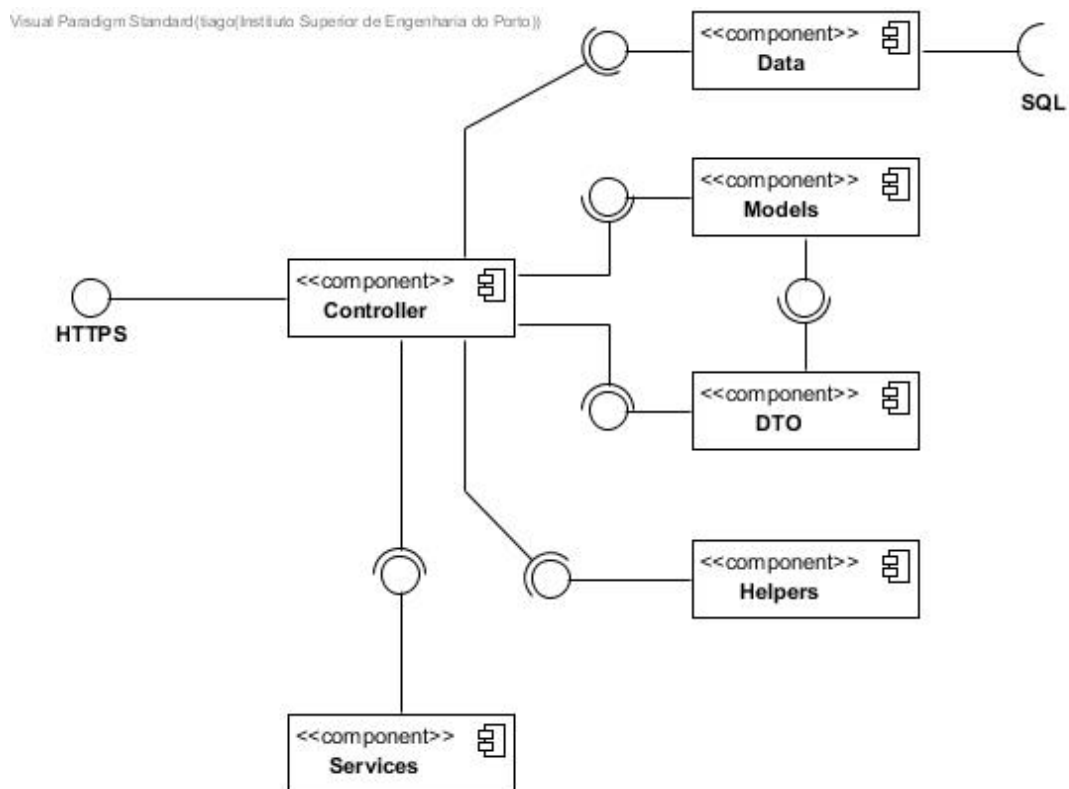


Figura 39 – Diagrama de componentes *Backend*

Na Figura 39 são definidos os seguintes componentes:

- **Controller:** Responsável por responder aos pedidos HTTP feitos à API e onde maior parte da lógica de gestão dos dados é aplicada;
- **Services:** Responsável pelo processo de login;
- **Data:** Responsável por providenciar uma ligação à base de dados onde o *controller* pode guardar dados;
- **DTO (Data transfer object):** Utilizados pelo *controller* para o envio de dados de uma forma filtrada das instâncias das classes existentes no componente *models* para o *frontend* para evitar o envio de informação desnecessária;
- **Models:** Representação das entidades do domínio e logica de negócio. O *controller* pode aceder a estas classes diretamente ou através dos DTO;
- **Helpers:** Componente onde são implementadas funções que são utilizadas várias vezes pelas várias classes do *controller*.

4.3.3.3 Diagrama de implantação

O diagrama de implantação permite visualizar onde será instalada aplicação final. O diagrama de implantação encontra-se apresentado na Figura 40.

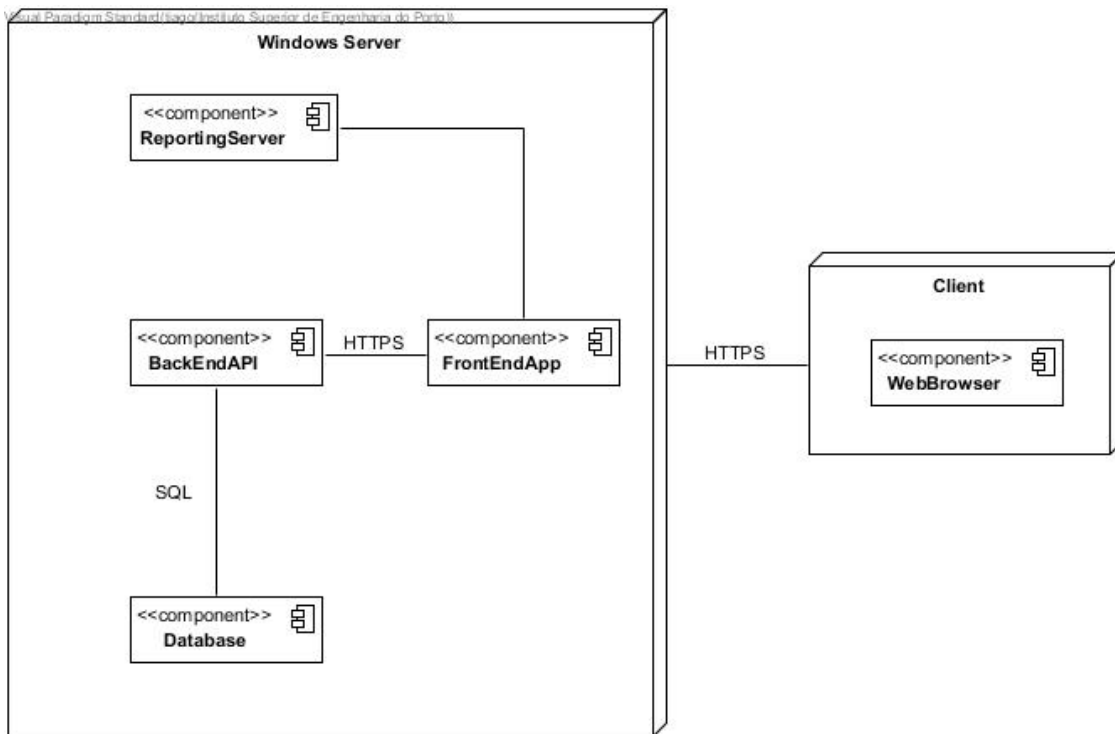


Figura 40 – Diagrama de implantação

Segundo a Figura 40 a solução será implantada num servidor *Windows Server* (Microsoft, 2019) interno da organização. Foi definido pelo Departamento de Tecnologias de Informação da Continental Mabor que a aplicação e a base de dados serão implantadas no mesmo servidor. Os utilizadores podem aceder à aplicação através qualquer Web Browser que esteja instalado nas suas máquinas de trabalho.

5 Desenvolvimento

Neste capítulo é descrito o processo de desenvolvimento da aplicação. Este processo é apresentado através de uma *Use case realization*, onde o desenvolvimento de todos os casos de uso são descritos com o uso de diagramas. De seguida, são também descritos os processos de migração de dados e de testes.

5.1 *Use case realization*

Nos subcapítulos são descritos os casos de uso mencionados no ponto 4.1.3, com o uso de vários diagramas. Diagramas de atividade, onde é explicado o processo de execução do caso de uso da perspetiva do utilizador e do sistema em geral, diagramas de classe que identificam as classes envolvidas no caso de uso e diagramas de atividade, onde é explicado em detalhe o caso de uso ao nível da execução do código. São também apresentados alguns *screenshots* da interface final da aplicação. Alguns casos de uso estão agrupados se o seu funcionamento for similar ou se fazem parte do mesmo processo, por exemplo, o caso de uso de mudança de password ocorre quando o utilizador efetua o login no sistema.

Para desenvolver o *backend* e o frontend foi utilizado o ASP.NET (Microsoft, 2020c). O *frontend* utiliza *HTML* e *jQuery* (jQuery, 2020). O ambiente de desenvolvimento foi o Visual Studio 2017 (Microsoft, 2020d).

5.1.1 UC1 e UC3 – Criar e editar utilizadores

Neste capítulo é apresentado o processo de gestão de utilizadores realizado pelo utilizador com permissões de administrador. O administrador pode criar novos utilizadores, definir os seus dados pessoais, departamentos a que pertencem e *roles* e alterá-los a qualquer momento.

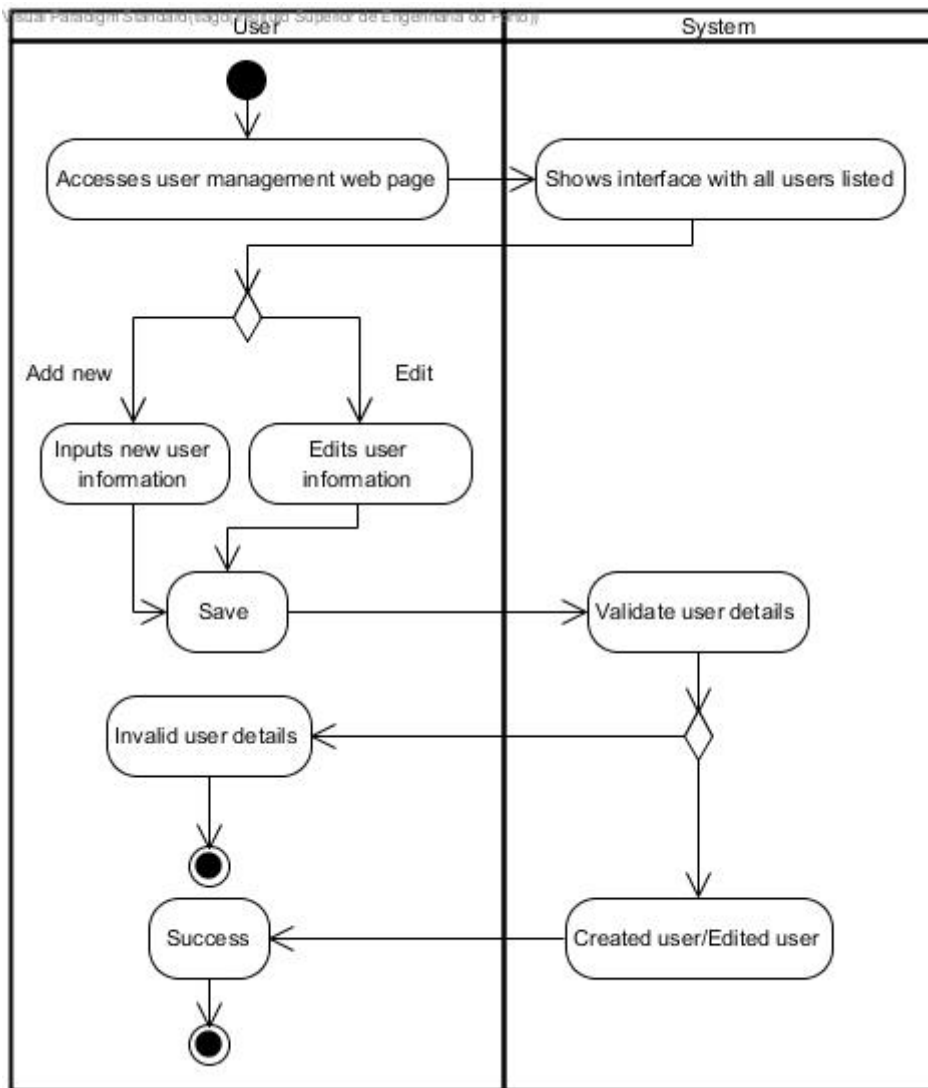


Figura 41 – Diagrama de atividades de criar um utilizador

Na Figura 41, o administrador acede à página web de gestão de utilizadores e o sistema mostra uma lista de todos os utilizadores. O administrador decide se quer criar um utilizador ou editar um existente. O administrador preenche um formulário do novo utilizador ou edita dados existentes e guarda. O sistema valida os dados e guarda o registo na base de dados e informa o administrador do sucesso da operação.

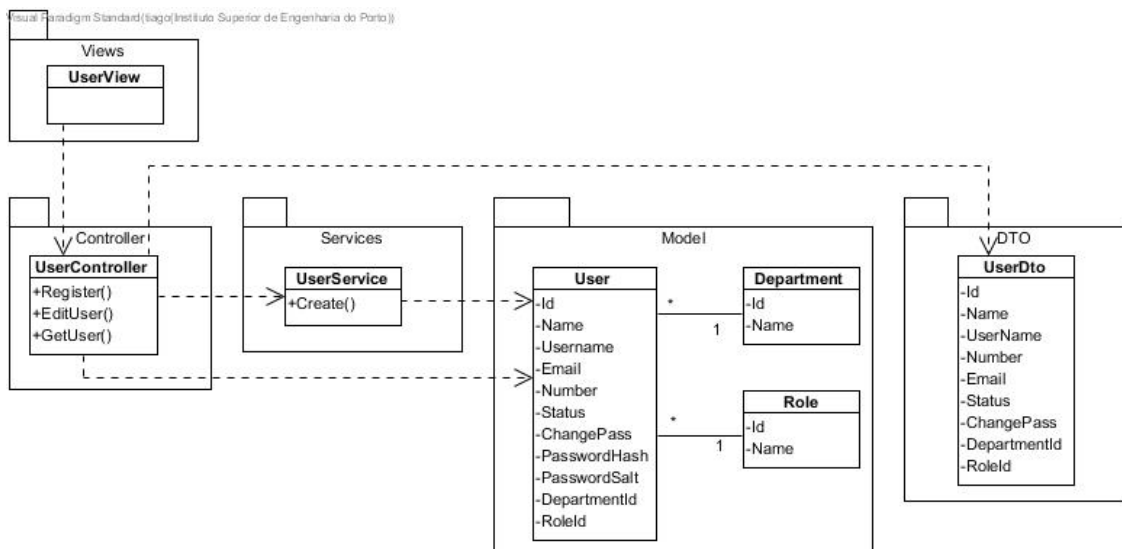


Figura 42 – Diagrama de classes de criar e editar utilizadores

Na Figura 42 é possível identificar as seguintes classes:

- **UserView**: Interface gráfica de gestão de utilizadores;
- **UserController**: Classe que recebe pedidos HTTP e responde com a informação pretendida. Responsável por gerir os dados das instâncias de *User*:
 - **Register**: Método que responde a pedidos do tipo POST para o registo de um novo utilizador;
 - **EditUser**: Método que responde a pedidos do tipo PUT para a edição de um utilizador existente;
 - **GetUsers**: Método que responde a pedidos do tipo GET para fornecer a informação de todos os utilizadores;
 - **GetUser**: Método que responde a pedidos do tipo GET para procura de informação de um utilizador.
- **UserService**: Classe auxiliar para criação de um novo utilizador
 - **Create**: Método que valida o utilizador e atribui-lhe uma password encriptada
- **User**: Classe que representa um utilizador no sistema
 - **Id**: Variável do tipo *long* que identifica do utilizador na base de dados.
 - **Name**: String com o nome do utilizador;
 - **Username**: String com o *username* usado para o processo de login;
 - **Email**: String com o email interno do utilizador
 - **Number**: String com o identificador interno do utilizador. Pode ser só numérico ou alfanumérico;
 - **Status**: Booleano que representa de o utilizador está ativado ou desativado no sistema;
 - **ChangePass**: Booleano que representa se o utilizador tem de mudar de password ao efetuar o login;
 - **PasswordHash**: *Array* de *bytes* com a password encriptada

- **PasswordSalt**: Array de bytes com a chave criptográfica para verificar se a password introduzida no momento de login é a correta;
- **DepartmentId**: Identificador do departamento do utilizador;
- **RoleId**: Identificador do *role* do utilizador;
- **Department**: Classe que representa um Departamento no sistema;
- **Role**: Classe que representa um Role no sistema. Os roles do sistema são “Admin”, “Produção”, “Scheduling”, “Workoff”, “Supervisão”;
- **UserDTO**: Classe *Data Transfer Object* com estrutura semelhante à classe *User* mas que omite a informação da password.

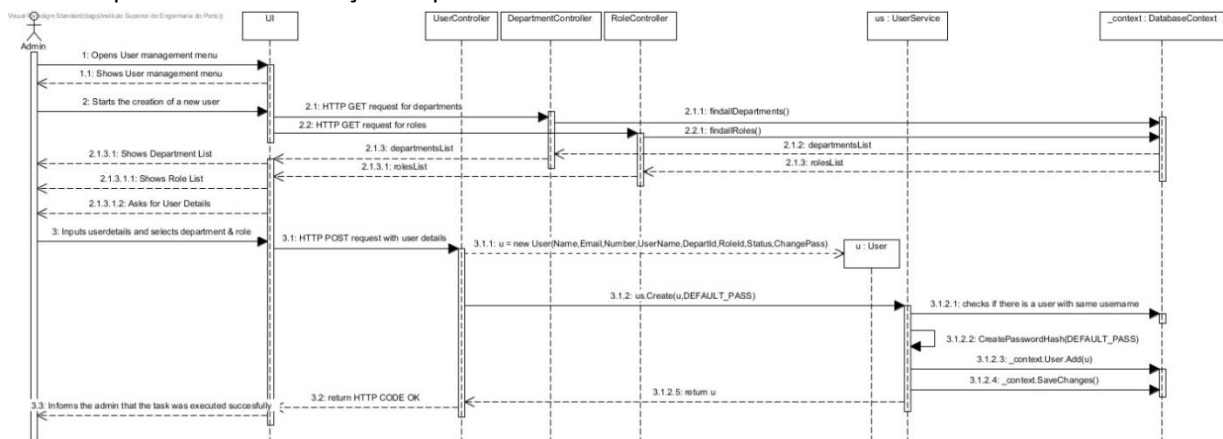


Figura 43 – Diagrama de sequência de criação de um utilizador

Na Figura 43 é apresentado o processo de registo de um utilizador. O administrador inicia o processo de registo. A interface envia um pedido HTTP GET aos controladores dos departamentos e *roles* para receber a informação de todos os registos disponíveis. Os controladores efetuam pedidos à base de dados e recebem as listas com todos os departamentos e *roles* registados. O controlador responde ao pedido com as listas e a interface mostra ao utilizador um formulário para preencher e duas listas para o administrador escolher o departamento e *role* do novo utilizador. O administrador introduz e guarda os dados. A interface envia um pedido HTTP POST para o controlador com a informação introduzida e o controlador envia essa informação para o *UserService*. O serviço verifica se existe um utilizador com o mesmo *username* e encripta a palavra chave definida por defeito. Esta encriptação é feita através da biblioteca *Security* do ASP.NET usando a encriptação HMACSHA512 (Microsoft, 2020e). De seguida, o *UserService* insere o utilizador na base de dados. O controlador responde à interface com o código HTTP OK 200 se o processo correu com sucesso ou HTTP BADREQUEST 400. A interface informa o utilizador do sucesso ou insucesso da operação.

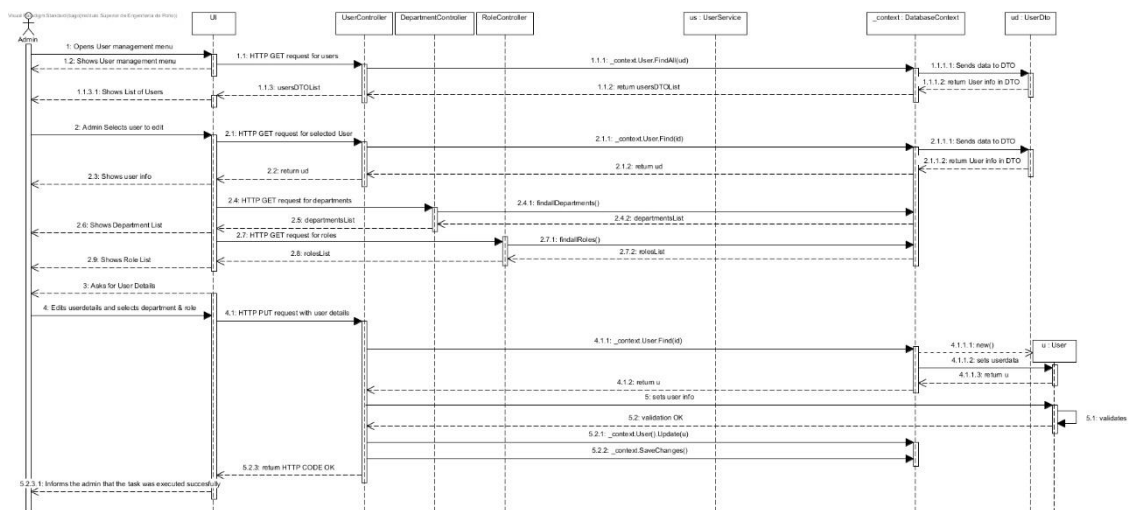


Figura 44- Diagrama de sequencia de edição de utilizador

O administrador para iniciar processo de edição de edição dos dados de um utilizador acede à página web de gestão de utilizadores (Figura 44). A interface envia um pedido *HTTP GET* para o controlador no *backend* com o objetivo de receber os dados de todos os utilizadores no sistema. O controlador envia uma *query* para a base de dados e esta responde com a lista de todos os *users*. O controlador envia essa lista para a interface e esta converte-a numa tabela onde são listados todos os utilizadores e onde é adicionado um botão para editar os dados de cada um. O administrador pressiona o botão de editar. A interface envia um pedido *HTTP GET* ao controlador a pedir todas as informações do utilizador selecionado. O controlador envia comunicação com a base de dados para encontrar essa informação e retorna essa informação para a interface. A interface também envia outros pedidos ao controlador para receber informação de todos os departamentos e *roles*. A interface apresenta um formulário com a informação recebida preenchida para o administrador editar. As listas de departamentos e *roles* são também apresentadas com os valores definidos para o utilizador pré-selecionados para poderem ser alterados. O administrador edita os dados necessários e guarda os dados. A interface envia um pedido *HTTP PUT* com os novos dados do utilizador para o controlador. O controlador acede aos dados desse utilizador, através da base de dados, altera os novos valores e após a validação destes, atualiza o registo. O controlador responde à interface e esta informa o utilizador do sucesso da operação.

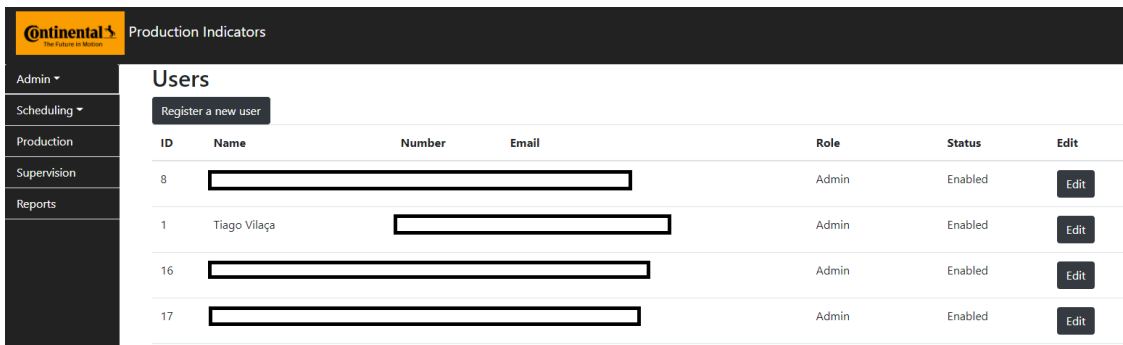


Figura 45 – Lista de utilizadores e botões de adição e edição

Edit User Info [Close]

Username:

Name:

Number:

Email:

Role:

Department:

Active User?

Figura 46 – Formulário de registo e edição de utilizadores

Na Figura 45 é apresentada a interface gráfica da página de gestão de utilizadores. Os utilizadores encontram-se listados numa tabela ordenados por *role*. A imagem encontra-se censurada para proteger a informação dos utilizadores da aplicação. Ao pressionar o botão “Register new user” ou “Edit” é apresentado um formulário representado na Figura 46, onde é possível efetuar o registo de um utilizador ou editar um existente.

5.1.2 UC2 e UC3 - Efetuar login e mudança de password

Nesta subsecção são apresentados os casos de uso de efetuar o processo de login e mudança de password. O utilizador introduz o seu *username* e *password* para obter acesso à aplicação. Se for necessário, o utilizador terá de alterar a sua *password* após efetuar o login. Esta mudança acontece quando o administrador efetua o *reset* de password.

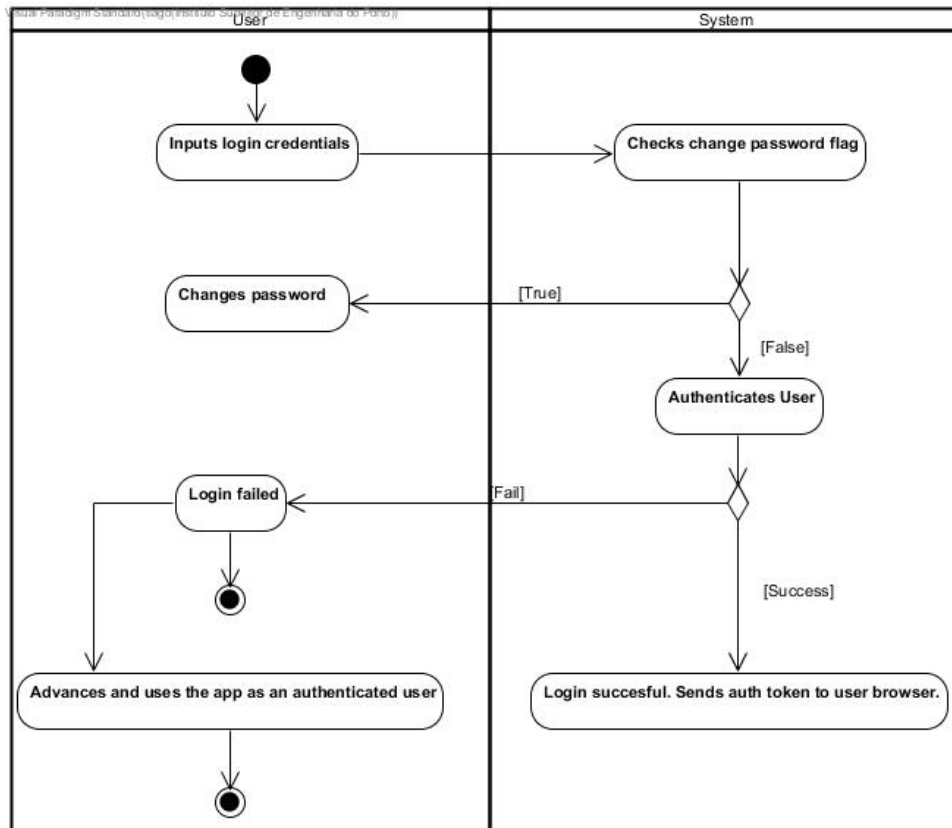


Figura 47 – Diagrama de atividades do processo de login e alteração de password

O utilizador acede à aplicação onde é requerido que este efetue o processo de login e, se necessário, mudar de password (Figura 47). O sistema apresenta um formulário de login e pede ao utilizador para introduzir o seu *username* e *password*. O utilizador introduz o *username* e *password*. O sistema autentica o utilizador e verifica se este necessita de alterar a password. Se for necessário alterar a password, o sistema apresenta um formulário onde o utilizador introduz a sua nova password. O sistema encripta a nova *password* e regista a alteração. De seguida, o sistema permite que o utilizador aceda à aplicação.

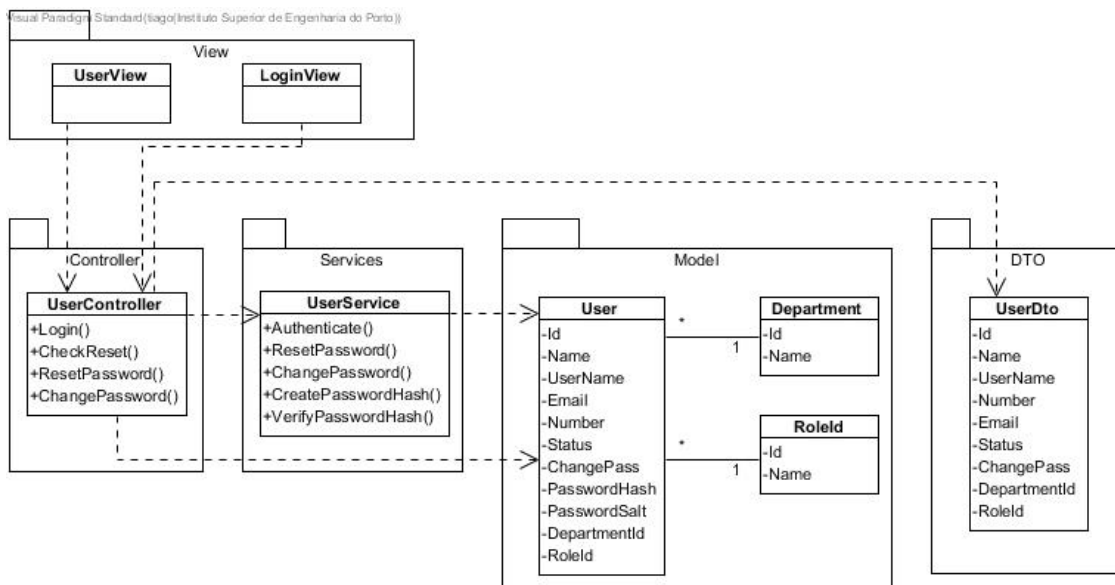


Figura 48 – Diagrama de classes do processo de login e mudança de password

Na Figura 48 estão representadas as seguintes classes:

- **LoginView**: Interface gráfica para login;
- **UserView**: Interface gráfica de gestão de utilizadores;
- **UserController**: Classe que recebe pedidos HTTP e responde com a informação pretendida
 - **Login**: Método que responde a pedidos do tipo POST para autenticar o utilizador;
 - **CheckReset**: Método que verifica se é necessário alterar a *password*;
 - **ResetPassword**: Método que responde a pedidos do tipo PUT para reinicializar a *password* para o valor por defeito;
 - **ChangePassword**: Método que responde a pedidos do tipo PUT para alterar a *password* para o valor recebido por parâmetro.
- **UserService**: Classe auxiliar para criação de um novo utilizador
 - **Authenticate**: Método que valida o utilizador e atribui-lhe uma password encriptada;
 - **ResetPassword**: Altera a password de volta para o valor por defeito;
 - **ChangePassword**: Altera a password para um valor recebido por parâmetro;
 - **CreatePasswordHash**: Encripta a password recebida por parâmetro;
 - **VerifyPasswordHash**: Compara a password introduzida pelo utilizador com a password encriptada no sistema para verificação no processo de autenticação
- **User**: Classe que representa um utilizador no sistema
- **Department**: Classe que representa um Departamento no sistema
- **Role**: Classe que representa um Role no sistema
- **UserDTO**: Classe *Data Transfer Object* com estrutura similar à classe *User* mas que omite a informação da password.

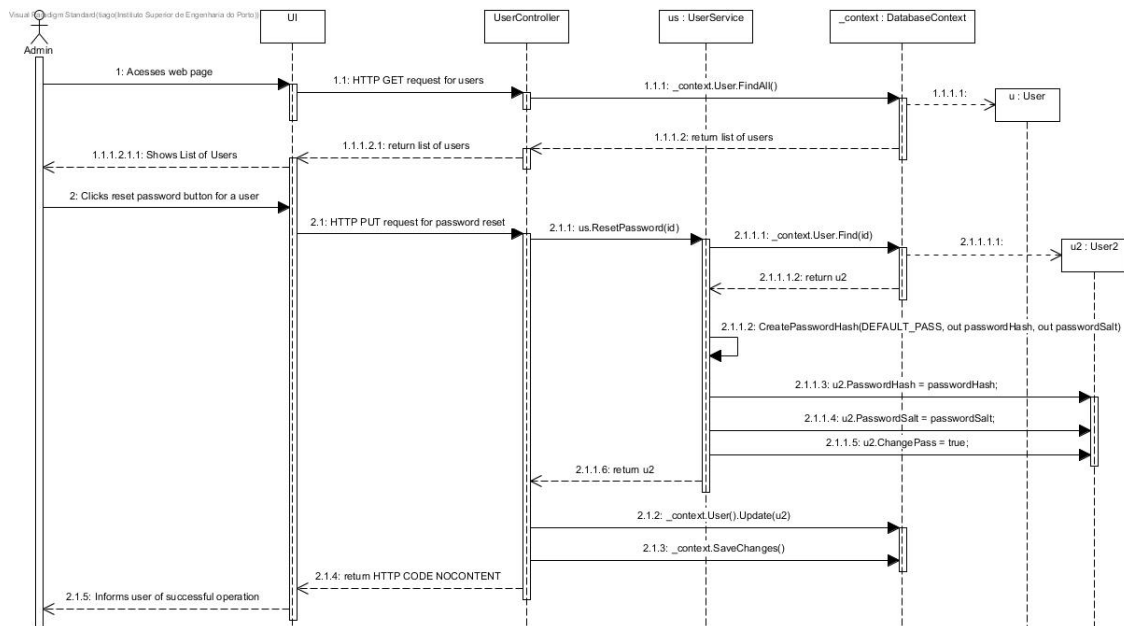


Figura 49 – Diagrama de sequência *reset* de password

ID	Name	Number	Email	Role	Status	Edit	Password
8	[Redacted]	[Redacted]	[Redacted]	Admin	Enabled	Edit	Reset Password
1	Tiago Vilaça	[Redacted]	[Redacted]	Admin	Enabled	Edit	Reset Password
16	[Redacted]	[Redacted]	[Redacted]	Admin	Enabled	Edit	Reset Password
17	[Redacted]	[Redacted]	[Redacted]	Admin	Enabled	Edit	Reset Password

Figura 50 – Listagem de utilizadores com botão de reset password

Na Figura 49, é mostrado o processo de *reset* da password. O administrador acede à página de gestão de utilizadores e o sistema lista todos os utilizadores registados no sistema (Figura 50). O administrador escolhe um utilizador e pressiona o botão de *reset password* para esse utilizador. A interface envia o pedido para o *UserController* e este invoca o método *ResetPassword* do *UserService*. Este método procura o utilizador selecionado na base de dados e utiliza o método *CreatePassword* para gerar uma password e chave criptográfica através de um valor por defeito. O serviço altera o *PasswordHash*, *PasswordSalt* e a *flag ChangePass* na instância do utilizador e retorna-o ao controlador. O controlador atualiza o registo do utilizador na base de dados e responde ao pedido efetuado pela *UI*.

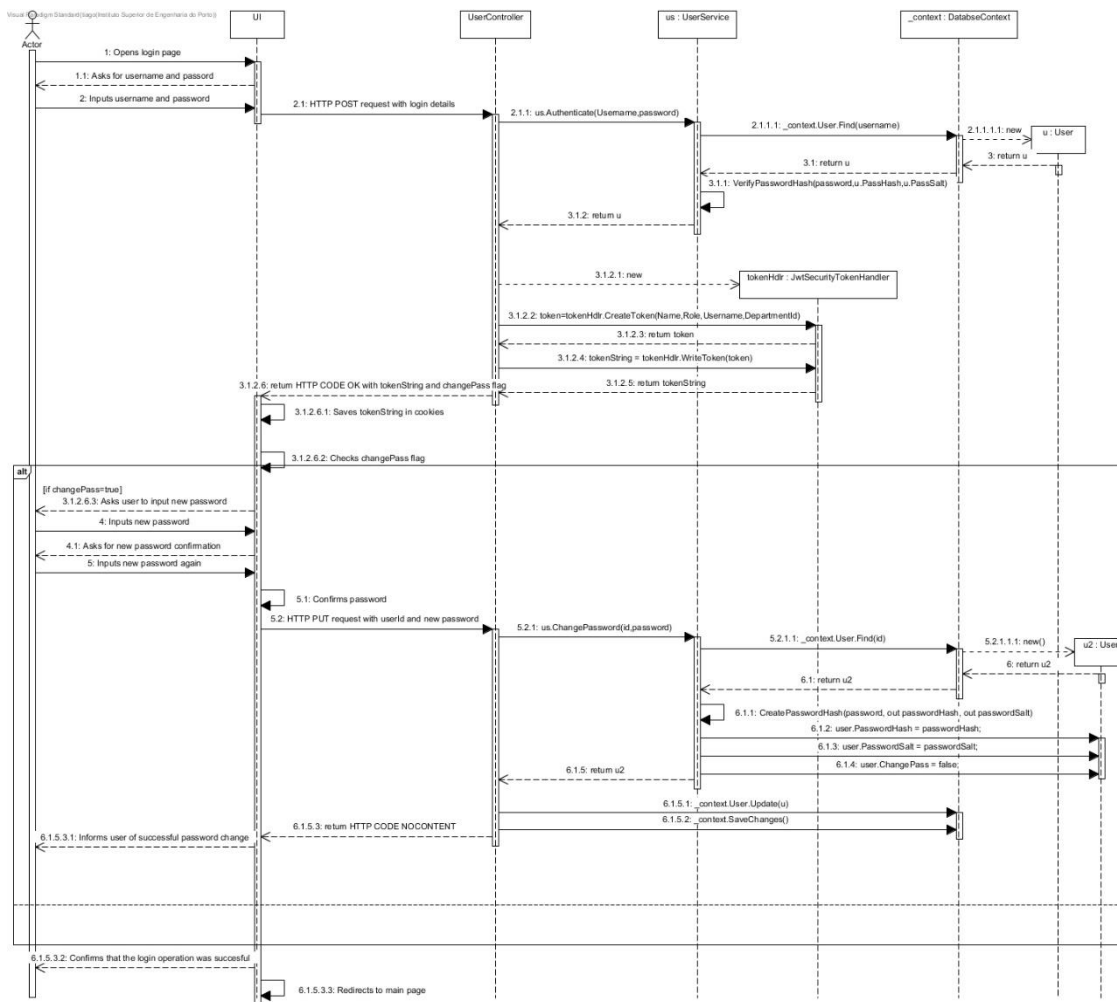


Figura 51- Diagrama de seqüência do processo de login e alteração de password

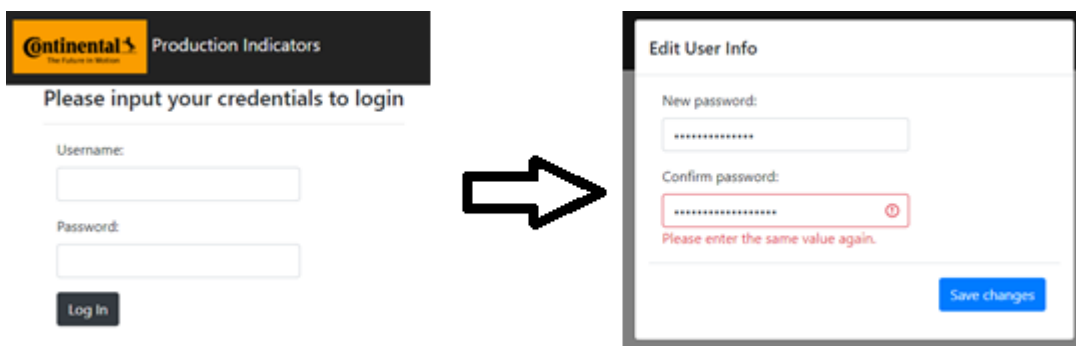


Figura 52 – Sequencia de introdução de login e alteração de password

Na Figura 51, está apresentado o processo de login e de possível mudança de password em detalhe. O utilizador acede à página web e a interface apresenta um formulário onde este introduz o seu *username* e *password*. A interface envia os dados de login através de um pedido *HTTP GET* para o *UserController* e este redireciona a informação para o *UserService*

que vai tratar do processo de autenticação. O serviço pesquisa na base de dados se o utilizador existe e, de seguida, verifica se a password é correta. O método *VerifyPasswordHash* aplica a chave de encriptação à password introduzida e compara com a password encriptada disponível no registo do utilizador na base de dados. Após a verificação da *password*, o controlador cria um *JSON Web Token* com o nome, *role*, *userName* e departamento. O *token* é encriptado e enviado, bem como a *flag ChangePass*, para a interface como resposta ao pedido. A interface guarda o *token* nas *Cookies* e verifica a *flag ChangePass*. Se este valor booleano estiver a *true*, a interface apresenta ao utilizador um formulário de alteração de password onde este introduz a nova *password* duas vezes, sendo a segunda vez para confirmação, processo representado na Figura 52. A password tem de ter o mesmo comprimento que a password de confirmação e tem de ter mais de 8 caracteres. A interface envia um pedido *HTTP PUT* para o controlador com o id do utilizador e a nova password e o *UserController* envia esses dados para o método *ChangePassword* do *UserService*. Este método procura o utilizador na base de dados, encripta a nova *password* e altera as variáveis de *PasswordHash* e *PasswordSalt*, altera o valor da *flag ChangePass* para *false* e retorna a instância do utilizador para o controlador. O controlador atualiza este registo na base de dados e responde ao pedido com o código *HTTP NO CONTENT*. A interface informa o utilizador do sucesso da operação e permite-lhe acesso ao menu principal da aplicação. Caso a *flag ChangePass* esteja definida como *false* após o processo de login, é garantido acesso à aplicação diretamente.

```
I reference | 0 exceptions
private static bool VerifyPasswordHash(string password, byte[] storedHash, byte[] storedSalt)
{
    if (password == null) throw new ArgumentNullException("password");
    if (string.IsNullOrWhiteSpace(password)) throw new ArgumentException("Value cannot be empty or whitespace only string.", "password");
    if (storedHash.Length != 64) throw new ArgumentException("Invalid length of password hash (64 bytes expected).", "passwordHash");
    if (storedSalt.Length != 128) throw new ArgumentException("Invalid length of password salt (128 bytes expected).", "passwordHash");

    using (var hmac = new System.Security.Cryptography.HMACSHA512(storedSalt))
    {
        var computedHash = hmac.ComputeHash(System.Text.Encoding.UTF8.GetBytes(password));
        for (int i = 0; i < computedHash.Length; i++)
        {
            if (computedHash[i] != storedHash[i]) return false;
        }
    }

    return true;
}
```

Figura 53 – Excerto de código de comparação de password

Na Figura 53 está apresentado o código que verifica se a password introduzida é a correta. O método recebe a password introduzida no processo de login, a password e chave de encriptação do utilizador a autenticar. Após validar os dados recebidos, o sistema cria uma instância da classe de criptografia utilizando a chave do utilizador (*StoredSalt*). De seguida, a password introduzida é encriptada utilizando a instância mencionada anteriormente e compara os dois *arrays* de *bytes*. Se os *arrays* forem iguais o método retorn o valor *true* e o utilizador é autenticado. Este método foi adaptado do blog de desenvolvimento web de Jason Watmore (Watmore, 2018).

5.1.3 UC5/UC6/UC7/UC8/UC9/UC10 – Criar e editar turnos, máquinas e tipos de stock

Neste capítulo são descritos os casos de uso de criar e editar a informação relativa a máquinas, tipos de stock e turnos. Normalmente os turnos são criados automaticamente, mas é possível criar e editá-los manualmente.

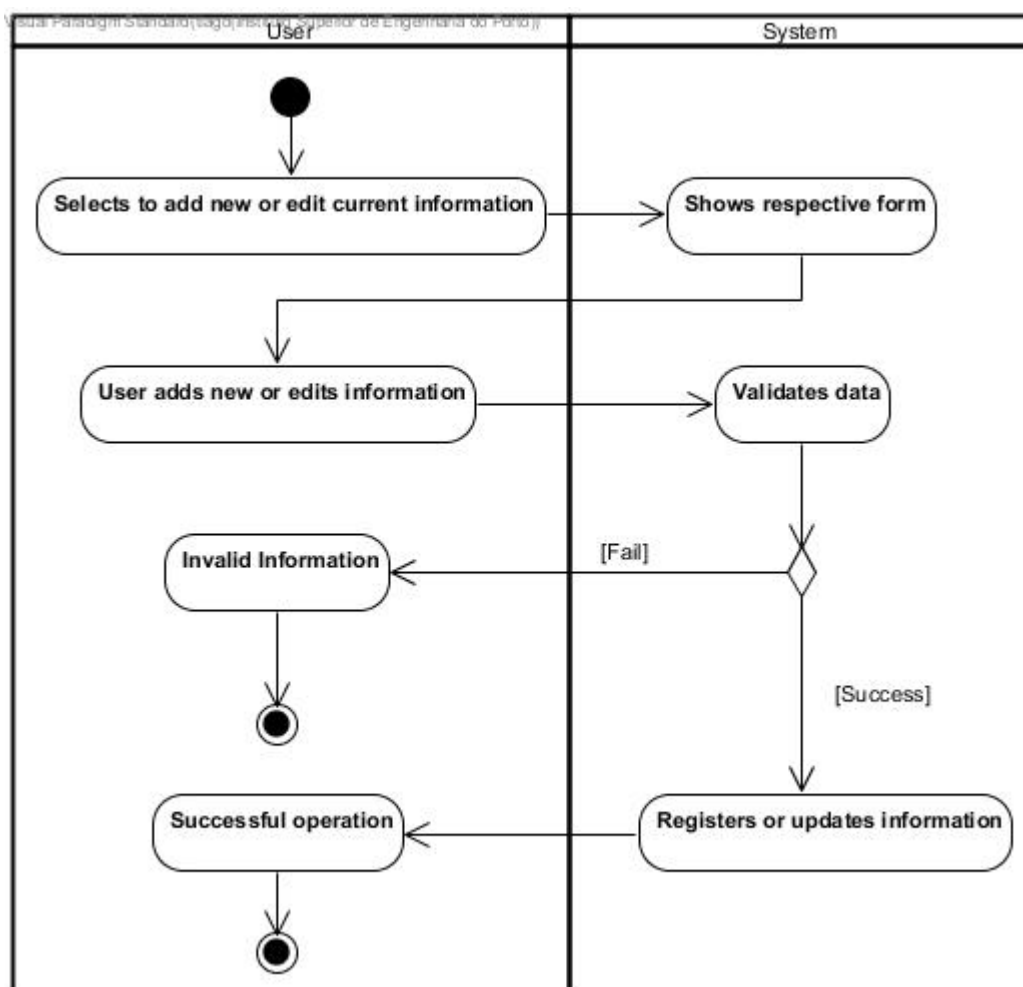


Figura 54 – Diagrama de atividades da gestão de turnos, máquinas e tipos de stock

O utilizador administrador acede à página web de gestão de turno/máquinas/stocks (Figura 54). O sistema apresenta os registos presentes numa lista e a possibilidade de efetuar um novo registo ou editar um dos existentes. O utilizador introduz ou edita a informação e guarda. O sistema valida os dados e guarda a informação na base de dados e informa o utilizador do sucesso da operação.

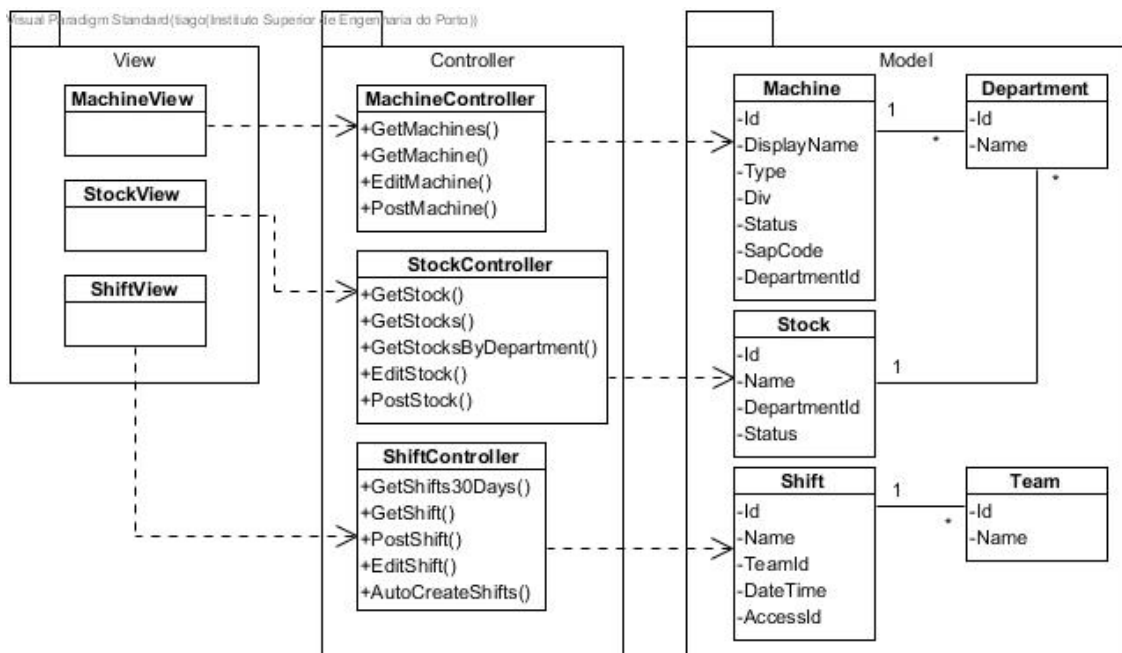


Figura 55 – Diagrama de classes de Máquinas, Stocks e Turnos

Na Figura 55 é possível identificar as seguintes classes:

- **MachineView:** Interface gráfica de gestão de máquinas;
- **StockView:** Interface gráfica de gestão de tipos de stock;
- **ShiftView:** Interface gráfica de gestão de turnos;
- **MachineController:** Classe que recebe pedidos HTTP e gere a informação sobre máquinas;
- **StockController:** Classe que recebe pedidos HTTP e gere a informação sobre *stocks*;
- **ShiftController:** Classe que recebe pedidos HTTP e gere a informação sobre turnos;
- **Machine:** Classe que representa uma máquina da fábrica;
 - **Id:** Variável do tipo *long* que identifica a máquina na base de dados;
 - **DisplayName:** *String* com o nome a mostrar para identificar a máquina;
 - **Type:** *String* que identifica o tipo de máquina;
 - **Div:** Valor decimal utilizado nos relatórios para cálculos relativos à produção;
 - **Status:** Valor booleano que representa se a máquina está ativa ou não;
 - **SapCode:** *String* com a identificação da máquina nos sistemas internos da Continental Mabor;
 - **DepartmentId:** Variável do tipo *long* que identifica o departamento responsável pela máquina;
- **Stock:** Classe que representa um tipo de stock, por exemplo o *stock* de pneus em verde ou *stock* de tela têxtil;
 - **Id:** Variável do tipo *long* que identifica do tipo de *stock* na base de dados;
 - **Name:** *String* com o nome do tipo de *stock*
 - **Status:** Valor booleano que representa se o tipo de *stock* está ativo ou não;

- **DepartmentId:** Variável do tipo *long* que identifica o departamento responsável pelo tipo de stock;
- **Shift:** Classe que representa um turno de trabalho na fábrica;
 - **Id:** Valor numérico do tipo *long* que identifica o turno na base de dados;
 - **Name:** *String* com o nome do turno. Este valor pode ser Turno 1, 2, 3 durante a semana e 4 e 5 durante o fim-de-semana
 - **DateTime:** Data de realização do turno;
 - **TeamId:** Valor booleano que representa se a equipa que está a trabalhar no turno;
 - **AccessId:** Variável do tipo *Int* utilizada para auxiliar a migração dos dados da aplicação Access;
- **Department:** Classe que representa um departamento;
- **Team:** Classe que representa uma equipa que está a trabalhar num turno;
 - **Id:** Variável do tipo *long* que identifica a equipa na base de dados;
 - **Name:** *String* com o nome da equipa.

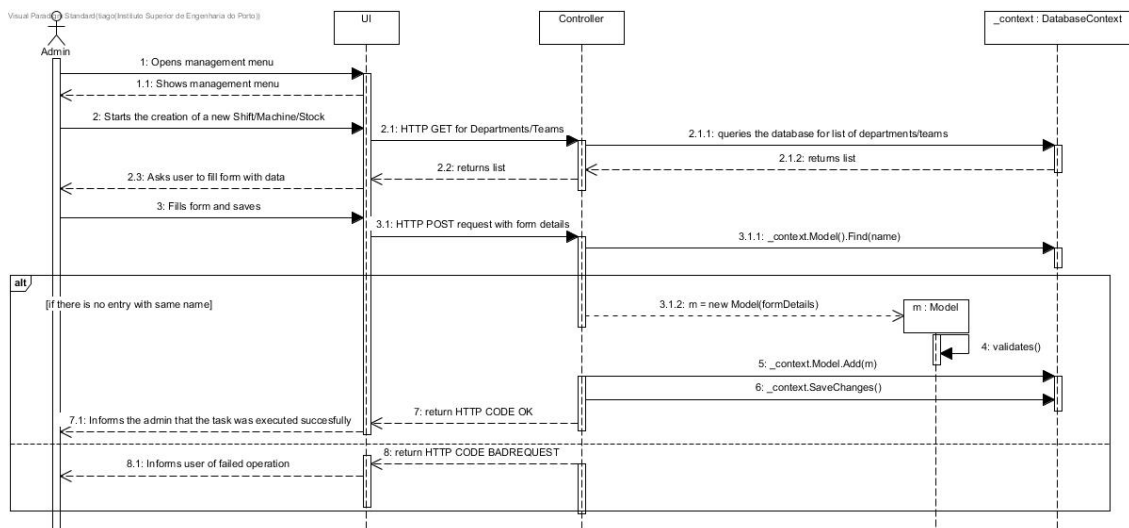


Figura 56- Diagrama de sequência da criação de máquinas/turnos/tipo de stock

Na Figura 56 encontra-se descrito o processo de criação de máquinas/turnos/tipo de stock, de uma forma geral, devido à semelhança entre estes casos de uso. O administrador abre a página de gestão de dados e inicia o processo de criação de um novo registo. A interface envia um pedido *HTTP GET* para o controlador para preparar os dados necessários para efetuar este novo registo, no caso de máquina e tipo de stock é preciso a lista dos departamentos e para um turno é necessário a lista das equipas. O administrador introduz e guarda os dados. A interface envia um pedido *HTTP POST* para o controlador com a informação introduzida e este verifica se não existe um registo com o mesmo nome na base de dados. Caso não exista o controlador cria uma instância de máquinas/turnos/tipo de stock com os dados introduzidos e guarda-os na base de dados. O controlador responde ao pedido com o código OK e a interface informa o utilizador do sucesso da operação.

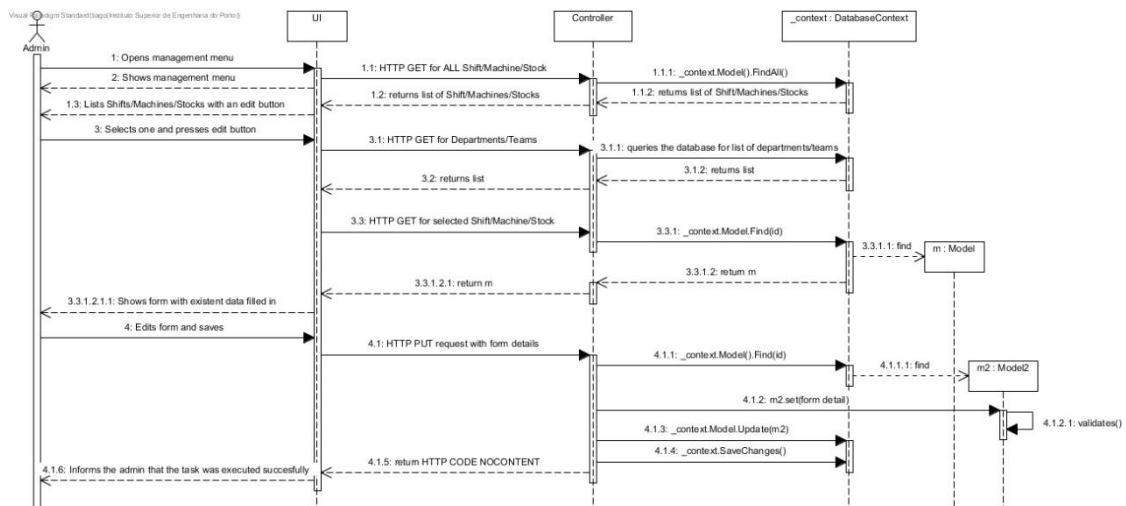


Figura 57- Diagrama de sequência de edição de máquinas/turnos/tipo de stock

O processo de edição encontra-se representado no diagrama da Figura 57. O administrador abre a página de gestão de máquinas/turnos/tipo de stock. A interface envia um pedido HTTP GET para o controlador para procurar todos os registos disponíveis. O controlador efetua uma *query* na base de dados e retorna a lista para a interface. A UI apresenta todos os registos numa lista e adiciona um botão de edição a cada uma das instâncias listadas. O utilizador seleciona uma e pressiona o botão de editar. De maneira semelhante ao processo de criação de um novo registo, é necessário receber a lista de departamentos/equipas. A interface apresenta o formulário com os dados do utilizador preenchidos e o administrador edita e guarda a informação. A UI envia os dados alterados através de um pedido HTTP PUT para o controlador. O controlador recebe o pedido e procura na base de dados pela instância a editar, aplica as alterações e atualiza o registo na base de dados. O controlador responde ao pedido com o código HTTP NOCONTENT e a interface informa o utilizador do sucesso da operação.

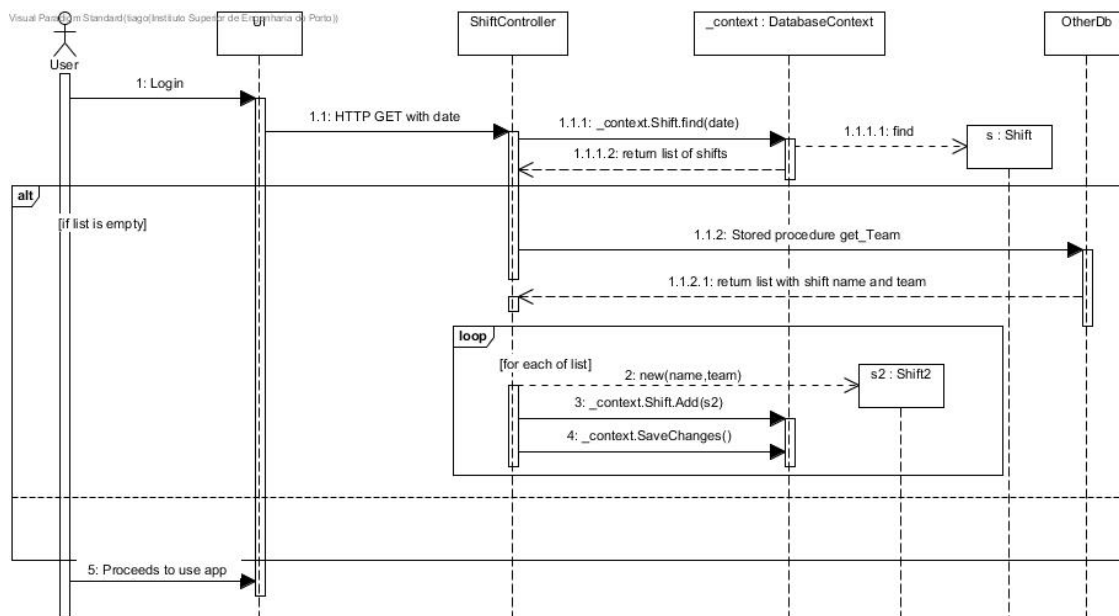


Figura 58 – Processo de criação de turnos automática.

No caso do registo de turnos, estes podem ser gerados pelo sistema automaticamente. Quando um utilizador efetua um *login* na aplicação, a interface envia dois pedidos HTTP para o *ShiftController*, um a pedir a criação automática de turnos para o dia atual e outro para o dia seguinte. O controlador pesquisa na base de dados se já existem turnos criados e, caso não existam, executa um *stored procedure* disponibilizado pela empresa que recebe uma data como parâmetro e retorna a equipa que está responsável pelo turno. O controlador cria as instâncias do turno e guarda-as na base de dados.

Machines

Register a new machine

ID	Machines	Type	SapCode	Div	Status	Edit
DPI						
1	Misturador 0	CARGAS	N/A	1	Enabled	Edit
2	Misturador 1	CARGAS	N/A	1	Enabled	Edit
3	Misturador 2	CARGAS	N/A	1	Enabled	Edit
4	Misturador 3	CARGAS	N/A	1	Enabled	Edit
5	Misturador 4	CARGAS	N/A	1	Enabled	Edit

Figura 59 – Página de gestão de máquinas

The image shows a web-based form titled "Edit Machine Info". The form has a white background and a thin border. It contains several input fields and a checkbox. The fields are: "Display Name" (text input with "Misturador 0"), "Type" (text input with "CARGAS"), "Sap Code" (empty text input), "Div" (text input with "1"), and "Department" (dropdown menu with "DPI" selected). Below these is a checkbox labeled "Active Machine?" which is checked. At the bottom right, there are two buttons: "Save" (blue) and "Close" (grey). The form is set against a dark background.

Figura 60 – Formulário de registo e edição de uma máquina

Na Figura 59 é apresentada a interface de gestão de máquinas. Todas as máquinas registadas no sistema estão listadas numa tabela e ordenadas por departamento primeiro e depois por ordem alfabética. O utilizador pode pressionar o botão “New machine” ou “Edit” para efetuar um novo registo ou editar um existente (Figura 60). As interfaces de gestão de stocks e de turnos foram implementadas de forma semelhante.

5.1.4 UC11/UC12/UC26 - Criar e editar dados de produção e sugestões do sistema

Neste capítulo são descritos os casos de uso de introdução e edição dos dados de produção de cada turno. O utilizador da produção irá introduzir as quantidades produzidas por cada máquina e de cada tipo de stock, dependendo do departamento selecionado. Durante a criação de um novo registo é possível receber as produções de registadas nos sistemas internos e preencher automaticamente os dados necessários.

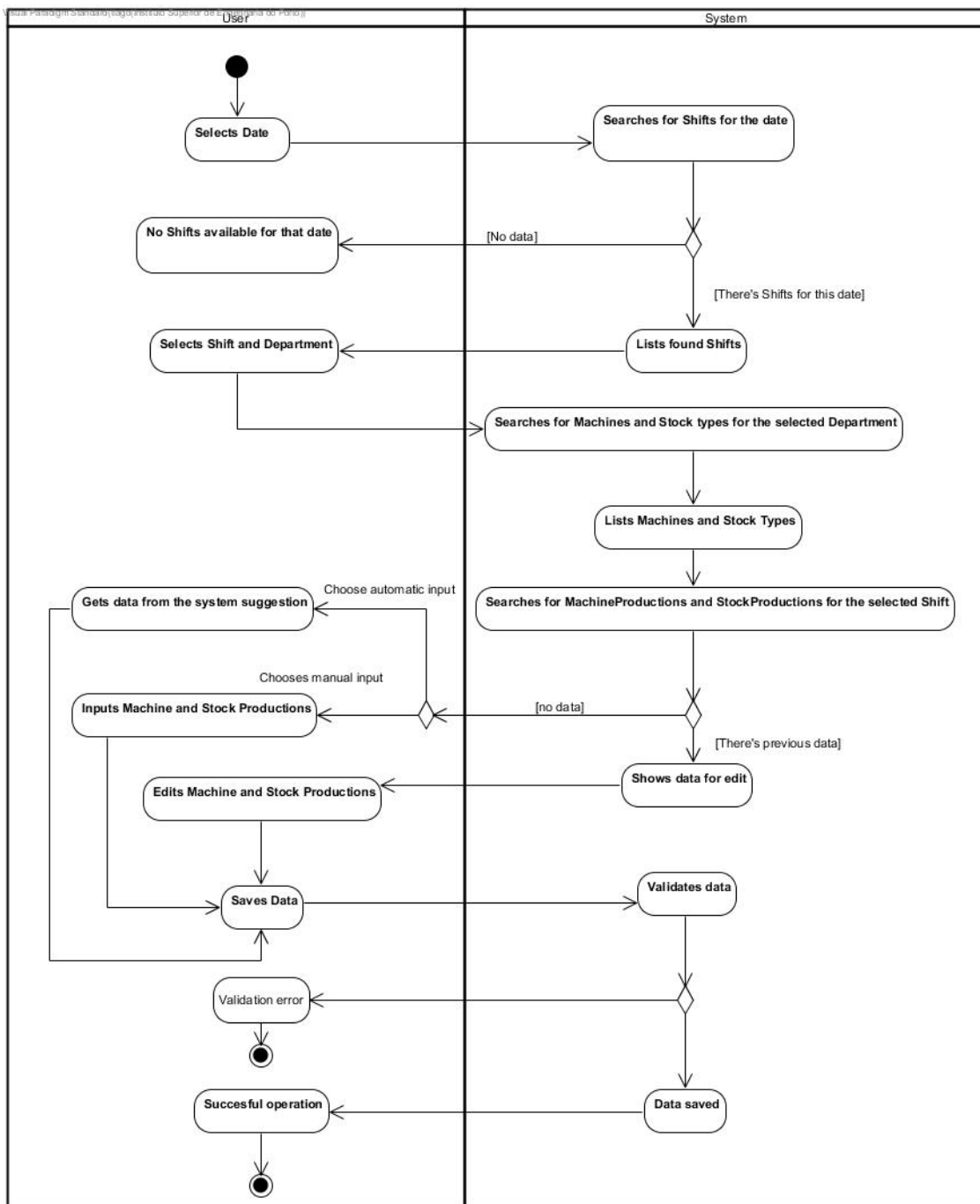


Figura 61 – Diagrama de atividades da criação e edição dados de produção e sugestões do sistema

Na Figura 61 é definido o diagrama de atividades do processo de registo de produção no final de cada turno. O utilizador acede à página web e o sistema pede ao utilizador para seleccionar uma data e requiere também que seja definido o turno. Após a introdução destes dois parâmetros, o utilizador selecciona o departamento a que pertence. O sistema procura por registos com os parâmetros definidos e, se existirem, apresenta o formulário ao utilizador com os dados preenchidos e o utilizador edita os dados necessários. No caso de não existir o

registo, o formulário apresentado encontra-se vazio e, o utilizador pode introduzir os dados para um novo registo de forma manual ou automática. O sistema recebe os dados, valida-os, cria um registo ou atualiza o registo existente e informa o utilizador do sucesso da operação.

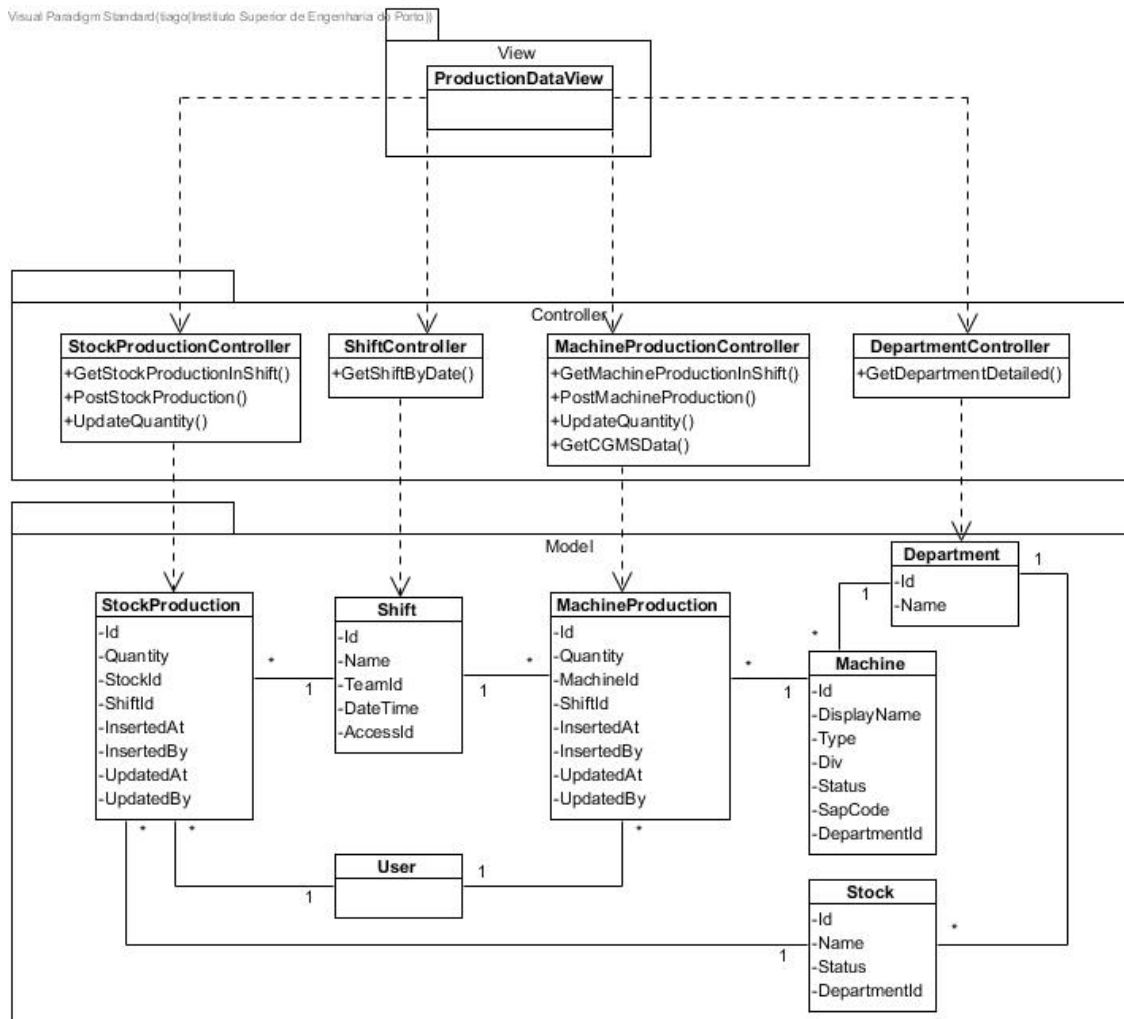


Figura 62- Diagrama de classes da criação e edição dados de produção e sugestões do sistema

Na Figura 62 encontram-se definidas as seguintes classes:

- **ProductionDataView:** Interface gráfica para registo de produções do turno
- **ShiftController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *Shift*
- **MachineController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *Machine*
- **MachineProductionController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *MachineProduction*
- **StockController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *Stock*

- **StockProductionController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *StockProduction*;
- **DepartmentController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *Department*;
- **Shift:** Classe que representa um turno;
- **Machine:** Classe que representa uma máquina;
- **Stock:** Representa um tipo de stock;
- **User:** Representa o utilizador do sistema;
- **MachineProduction/StockProduction:** Classe que representa a quantidade produzida por uma máquina/stock num determinado turno;
 - **Id:** Variável do tipo long única que identifica a instância na base de dados
 - **Quantity:** Variável do tipo decimal que representa a quantidade produzida
 - **MachineId/StockId:** Variável do tipo long que identifica a máquina/stock a qual corresponde a *MachineProduction/StockProduction*;
 - **ShiftId:** Variável do tipo long que identifica o turno que corresponde a *MachineProduction/StockProduction*;
 - **InsertedAt:** Data de introdução do registo na base de dados;
 - **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
 - **UpdatedAt:** Data da última atualização do registo na base de dados;
 - **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;

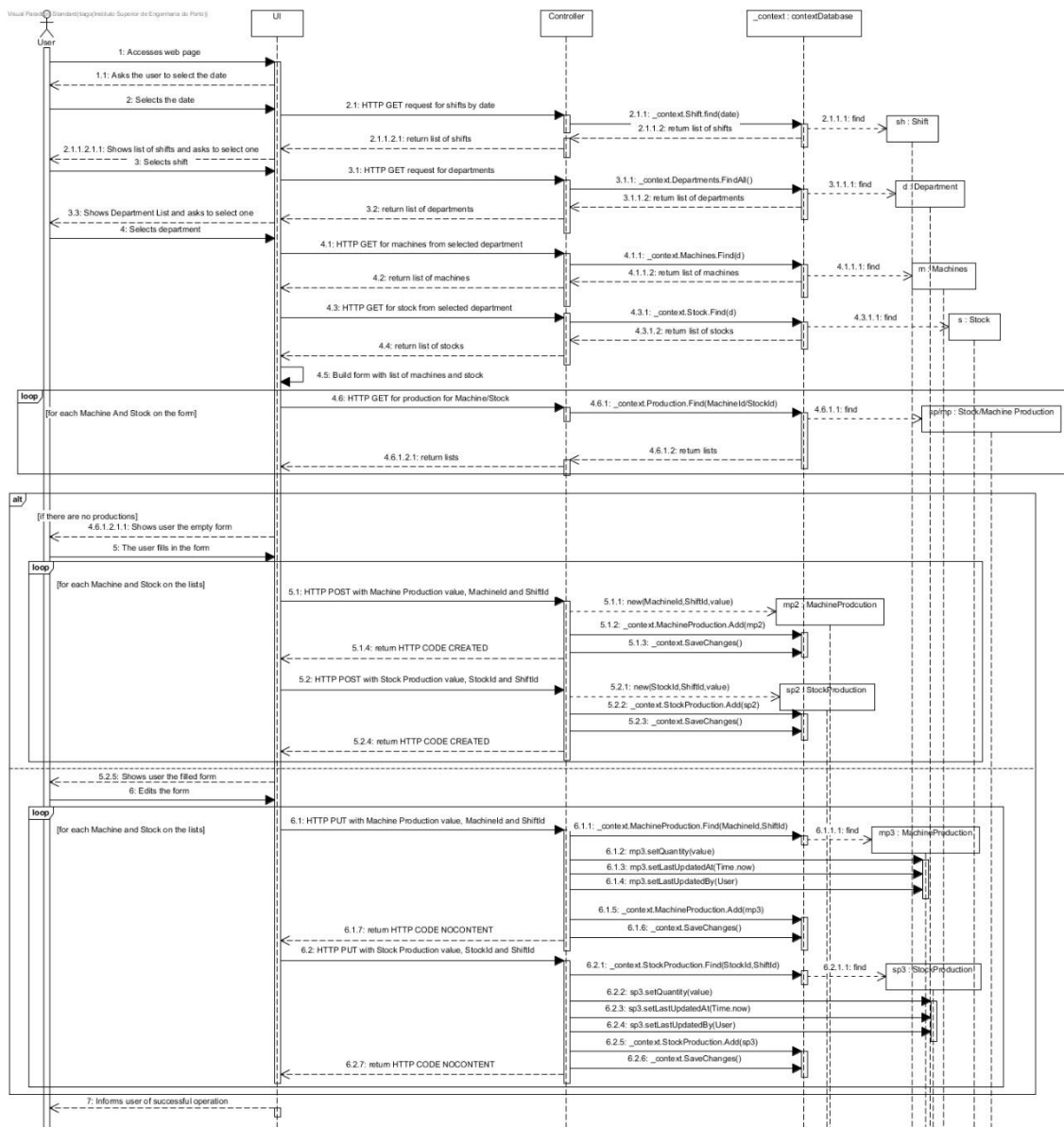


Figura 63 - Diagrama de sequencia da criação e edição dados de produção e sugestões do sistema manual

Na Figura 63 encontra-se definido em detalhe o processo de registo de produções de máquinas e stocks durante um turno. O utilizador acede à página web e seleciona uma data. A interface efetua um pedido HTTP GET para procurar os turnos para a data selecionada. O controlador procura na base de dados e retorna a lista de turnos para a interface. O utilizador seleciona o turno e a UI repete o processo de procura para os departamentos. O utilizador seleciona o departamento. Após a seleção destes parâmetros, a interface envia dois pedidos HTTP GET para o controlador para receber informação sobre as máquinas e tipos de stock do departamento selecionado. O controlador realiza as *queries* na base de dados e retorna a lista de máquinas e tipos de stock. A UI constrói o formulário e para cada máquina e tipo de stock envia um pedido HTTP GET para o controlador para receber informação sobre as

MachineProductions e *StockProductions* de cada máquina/stock do turno escolhido. Se não houver dados de produção a interface apresenta o formulário vazio e o utilizador insere os dados. Se existirem registos anteriores, a interface preenche o formulário e permite ao utilizador editar os dados. O utilizador introduz/edita e guarda a informação. A interface envia, para cada campo do formulário, um pedido HTTP POST ou HTTP PUT com a quantidade, id da máquina/stock e id do turno. O controlador recebe os pedidos e regista as novas produções ou atualiza as existentes. Após a finalização deste processo a interface informa o utilizador do sucesso da operação.

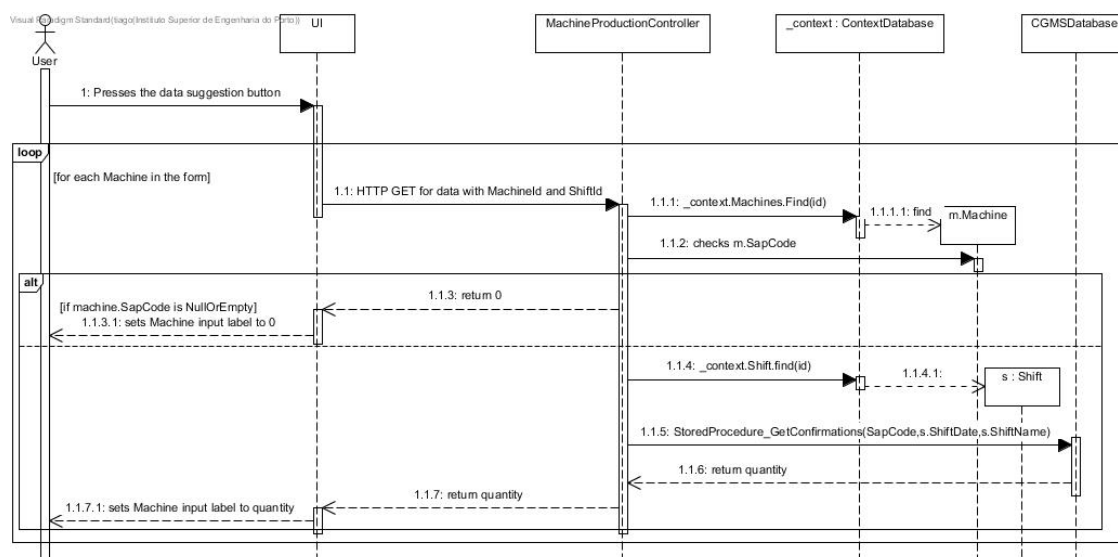


Figura 64 – Diagrama de sequência sistema de sugestão automática

Durante o processo de registo de novos dados de produção o utilizador pode utilizar pedir ao sistema para introduzir os dados automaticamente no formulário (Figura 64). Esta funcionalidade só está disponível para as máquinas. O utilizador pressiona o botão de sugestão de dados e a interface envia um pedido HTTP GET por cada máquina do departamento selecionado com o Id da máquina e Id do turno. O controlador usa o Id da máquina para procurar na base de dados a máquina e obter o SapCode. Se este estiver vazio o controlador retorna o valor 0 e a interface preenche o campo correspondente com esse valor. Caso contrário, utiliza o Id do turno para procurar na base de dados o turno. De seguida, utiliza os dados da máquina e do turno para invocar um *StoredProcedure* numa base de dados da Continental Mabor onde está guardada as produções de cada máquina em cada turno. O *StoredProcedure* retorna à quantidade e este valor é enviado para a UI. A interface preenche o valor no formulário. O utilizador pode continuar o processo de registo descrito para a Figura 55 sem ter de preencher manualmente as produções das máquinas.

Production Indicators | Tiago Vileça | Log off

Productions

Search

Select a Date: 04/05/2020 | Select a Shift: Shift 1 Team C | Select a Department: DPIV

DPIV Shift 1 Team C 04/05/2020

Machine		Stock	
Pintura 1:	1750	INV PNEUS EM CRÚ:	11552
Pintura 2:	1430	INV PNEUS RETIDOS CRÚ:	0
Pintura 3:	0		
Pintura 4:	2336		
Pintura 5:	154		
VULCANIZADOS:	9037		

Save

Figura 65 – Interface de registo de dados de produção para o Departamento IV

Production Indicators

Productions

Search

Select a Date: 08/10/2020 | Select a Shift: Shift 1 Team A | Select a Department: DPIII

Get Machine Data from CGMS

DPIII Shift 1 Team A 08/10/2020

Machine	
Módulo 01:	109
Módulo 02:	0
Módulo 03:	450
Módulo 04:	415
Módulo 05:	284
Módulo 06:	465
Módulo 07:	154
Módulo 08:	361

Figura 66 – Interface de registo de dados de produção para o Departamento III

Nas Figuras 65 e 66, são apresentadas duas versões da mesma interface de registo dos dados de produção. Dependendo do departamento selecionado, são mostradas máquinas e tipos de stock diferentes. Na primeira imagem foi acedido a um registo do turno 1 do dia 4 de maio de 2020 e o sistema apresentou os dados registados desse dia e permite a edição destes. Na segunda imagem, foi selecionado o turno 1 do dia 8 de outubro de 2020. Para este turno, ainda não tinha ocorrido um registo de dados, por isso a interface apresentou a interface com os campos de introdução vazios, mas foi utilizado o botão “Get Machine Data from CGMS” para preencher automaticamente os campos.

5.1.5 UC13/UC14/UC15/UC16/UC17/UC18/UC25/UC26 - Criar e editar dados de produção diária, dados de programação, APA e Workoff

Neste capítulo são descritos os casos de uso de registo e edição de produção diária, dados de programação, APA e *Workoff*. O registo de produção diária consiste de indicadores de produção total de um certo dia, normalmente registados no dia seguinte. Os dados de programação são dados de planeamento da quantidade total a ser produzida para um certo dia. Este registo costuma ser efetuado no início do dia. Os dados de APA referem ao valor de pneus no armazém APA e Buffer, para um dia. Finalmente os registos de *Workoff* representam material que irá ser reintroduzido na linha de produção para evitar desperdícios, num certo dia. Todos estes registos têm em comum o facto de serem informação diária, daí serem agrupados neste capítulo.

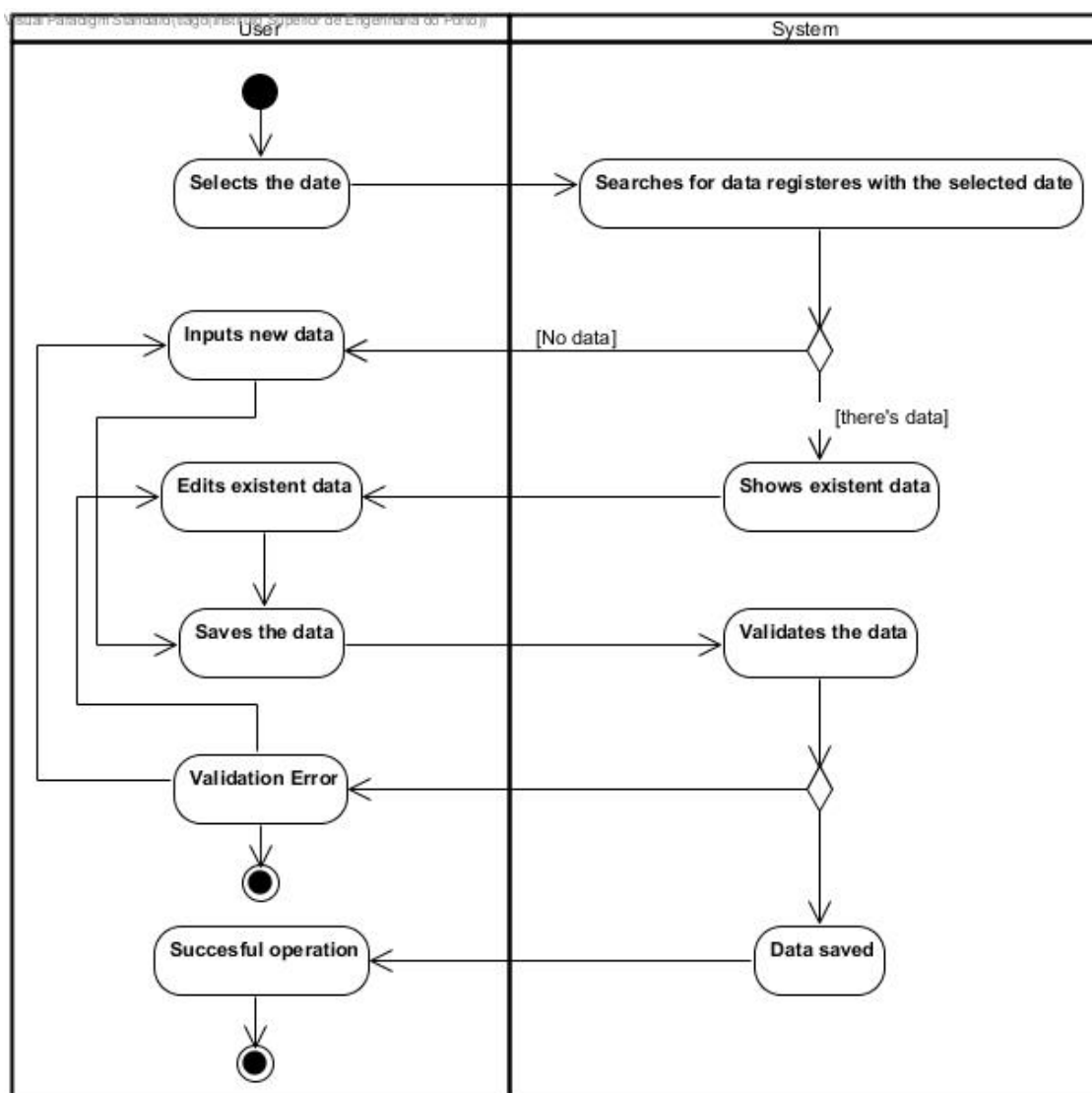


Figura 67 – Diagrama de atividades de criação e edição dos dados diários

Na Figura 67 está representado o diagrama de atividades geral destes casos de uso. O utilizador acede à página web e seleciona a data. O sistema procura por registos para essa data e, se existirem apresenta o formulário preenchido para o utilizador editar. Caso contrário o utilizador efetua um novo registo. Após introduzir ou editar os dados, o utilizador guarda-os. O sistema valida os dados e guarda-os na base de dados e informa o utilizador do sucesso da operação.

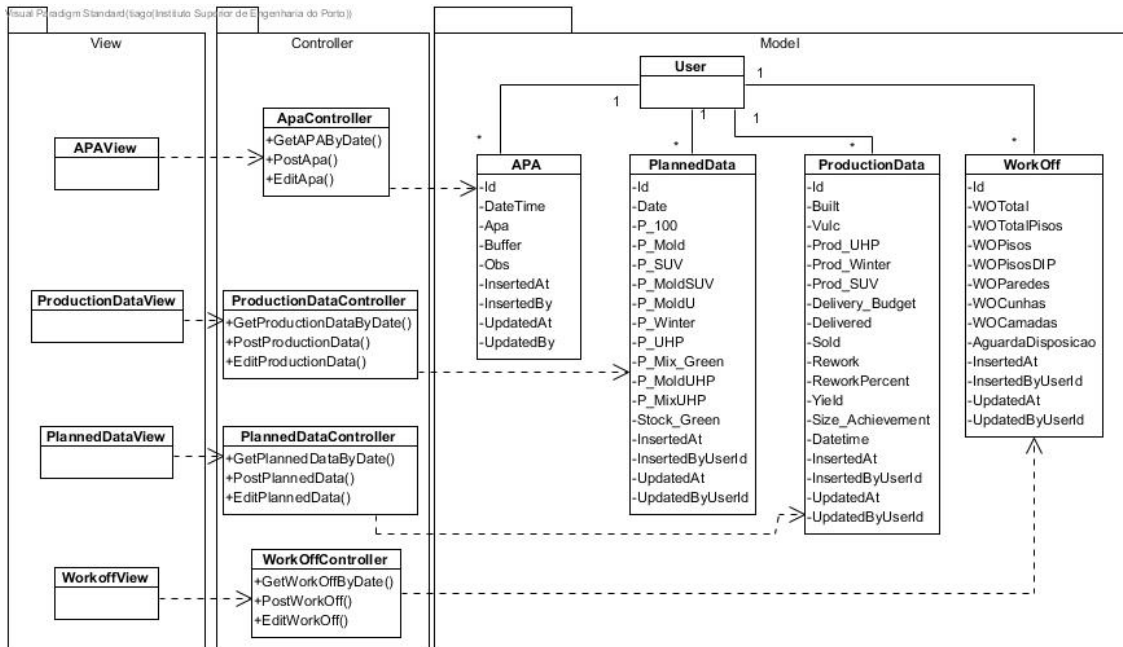


Figura 68 – Diagrama de classes dos dados diários

Na Figura 68 é possível identificar as seguintes classes:

- **Views:** Interfaces gráficas para cada tipo de dados;
- **Controller:** Classes que recebem pedidos HTTP e que são responsáveis por gerir dados das classes do Modelo correspondentes;
- **APA:** Classe que representa dados de armazém;
 - **Id:** Valor numérico do tipo *long* que identifica a classe na base de dados;
 - **DateTime:** Data do registo;
 - **Apa:** Valor numérico inteiro que representa o valor armazenado no APA;
 - **Buffer:** Valor numérico inteiro Buffer;
 - **Obs:** *String* para observações;
 - **InsertedAt:** Data de introdução do registo na base de dados;
 - **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
 - **UpdatedAt:** Data da última atualização do registo na base de dados;
 - **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **PlannedData:** Dados de planeamento de produção de um dia;

- **Id:** Valor numérico do tipo *long* que identifica a classe na base de dados;
- **Date:** Data do registo;
- **P_100 ... Stock_Green:** Valores numéricos do tipo inteiro;
- **InsertedAt:** Data de introdução do registo na base de dados;
- **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
- **UpdatedAt:** Data da última atualização do registo na base de dados;
- **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **ProductionData:** Classe que representa os valores produzidos num dia;
 - **Id:** Valor numérico do tipo *long* que identifica a classe na base de dados;
 - **DateTime:** Data do registo;
 - **Built ... Size_Achievement:** Valores numéricos do tipo inteiro;
 - **InsertedAt:** Data de introdução do registo na base de dados;
 - **InsertedBy:** Variavel do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
 - **UpdatedAt:** Data da última atualização do registo na base de dados;
 - **UpdatedBy:** Variavel do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **Workoff:** Classe que representa material a ser reintroduzido no processo de fabrico;
 - **Id:** Valor numérico do tipo *long* que identifica a classe na base de dados;
 - **DateTime:** Data do registo;
 - **WO_Total ... Aguarda_Disposicao:** Valores numéricos do tipo inteiro;
 - **InsertedAt:** Data de introdução do registo na base de dados;
 - **InsertedBy:** Variavel do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
 - **UpdatedAt:** Data da última atualização do registo na base de dados;
 - **UpdatedBy:** Variavel do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **User:** Utilizador do sistema;

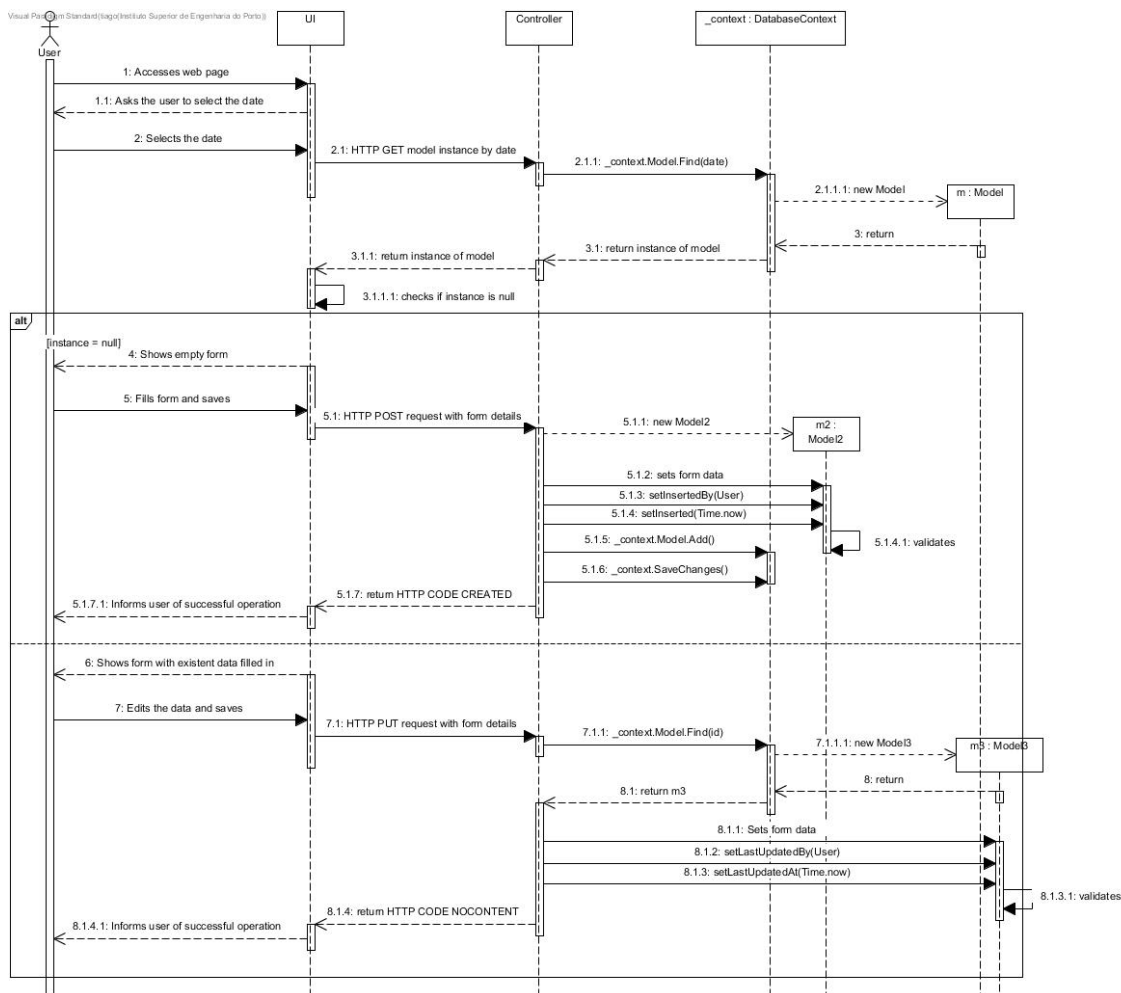


Figura 69 - Diagrama de sequencia do registo e edição de dados diários

Na Figura 69 encontra-se definido o processo de registo e edição de produção diária, dados de programação, APA e *Workoff*. O utilizador acede à página web e a interface pede para escolher uma data. O utilizador seleciona a data e a interface efetua um pedido HTTP GET ao controlador para verificar se existe registos para a data escolhida. O controlador pesquisa na base de dados e retorna para a interface o resultado da pesquisa. Se não existir um registo prévio, a interface apresenta um formulário vazio e, em caso contrário, apresenta um formulário preenchido. O utilizador introduz novos dados ou edita os existentes e guarda a informação. A interface envia um pedido HTTP POST, no caso de um novo registo, ou um HTTP PUT, em caso de atualização, com a informação do formulário. O controlador ao receber o pedido POST cria uma instância de classe e adiciona-a à base de dados. Ao receber o pedido PUT vai procurar na base de dados pelo registo existente, atualiza com a nova informação e guarda as alterações. O controlador responde à interface com o código correspondente ao pedido e a UI informa o utilizador do sucesso da operação.

Production

Search

Select a Date

05/05/2020

Built:	UHP Produced:	Winter Produced:	Delivered:	Rework:	Yield:
29765	22417	2932	29776	1174	97.66
Curing:	SUVs Produced:	Budget Delivered:	Sold:	%Rework:	Size Achievement:
29036	569	30000	21154	4.04	69

Figura 70 – Página web de registo de dados de produção diários

Na Figura 70 foi acedido ao registo do dia 5 de maio de 2020 e, como já existia um registo para esse dia, o sistema permite editar automaticamente os dados introduzidos. As interfaces para os dados de programação, APA e Workoff foram implementadas de maneira similar.

5.1.6 UC19 e UC20 – Criar e editar dados de SCRAP

Neste capítulo são definidos os casos de uso de criação e edição de dados relativos ao *SCRAP* produzido por cada departamento. Cada departamento possui uma meta, definida em percentagem, da quantidade de *SCRAP* produzida que não deve ser ultrapassada. O utilizador pode também definir estas metas durante o processo de registo ou edição dos valores de *SCRAP*.

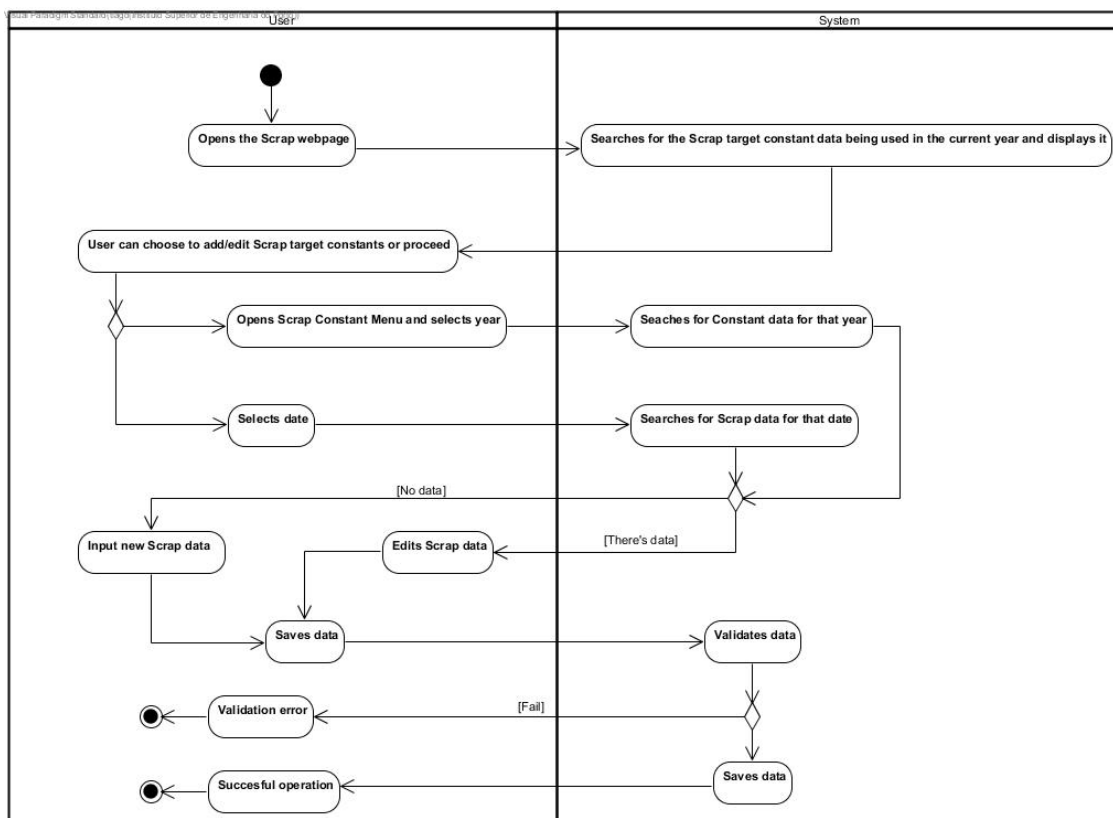


Figura 71 – Diagrama de atividades de criação e edição dos dados de SCRAP.

Na Figura 71 encontra-se representado o diagrama de atividades do processo de gestão dos dados de SCRAP. O utilizador acede à página web e a interface apresenta as metas de SCRAP para o ano presente, se estas estiverem definidas. O utilizador acede ao menu de registo das constantes para registar e editar as metas de SCRAP. O sistema mostra um formulário num *pop-up* e pede ao utilizador para escolher o ano para qual quer definir as metas de SCRAP. O utilizador selecciona o ano e o sistema procura se existem constantes definidas para esse ano. Se existirem, o utilizador pode editar os dados existentes ou, em caso contrário pode registar novas metas. O utilizador guarda os dados e o sistema valida-os, efetua o registo e informa o utilizador do sucesso da operação. O *pop-up* fecha e o utilizador selecciona a data para efetuar o registo do SCRAP produzido. O sistema procura por um registo de SCRAP da data introduzida e, caso este exista, apresenta-o ao utilizador e este edita os dados. Se este não existir, o utilizador introduz os novos dados. O sistema valida e regista os dados introduzidos. O sistema informa o utilizador do sucesso da operação.

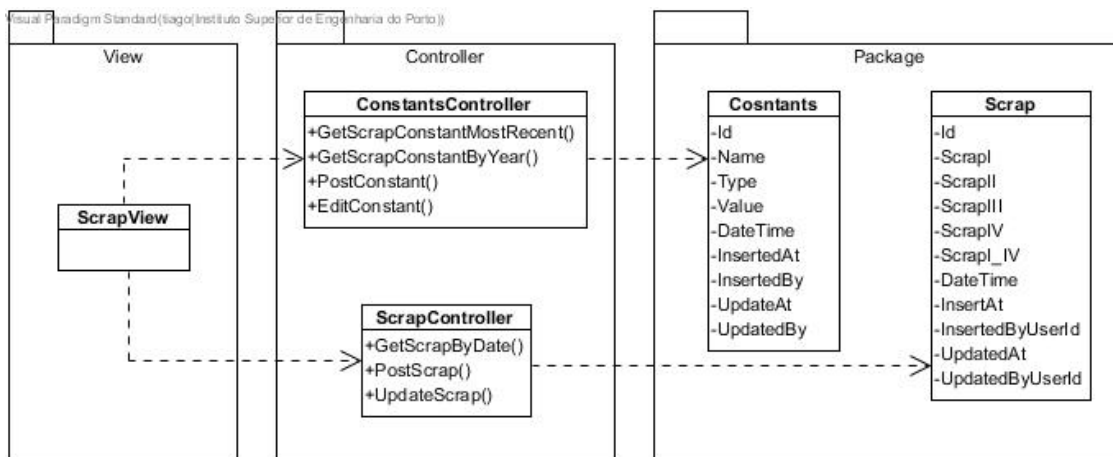


Figura 72 – Diagrama de classes do processo de gestão de SCRAP e contantes de Metas SCRAP

Na Figura 72 estão definidas as classes responsáveis pela execução dos casos de uso 19 e 20:

- **ScrapView:** Interface gráfica para os dados de SCRAP
- **ConstantsController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *Constants*;
 - **GetScrapConstantMostRecent:** Procura pelas constantes do tipo SCRAP definidas para o ano mais recente;
 - **GetScrapContantsByYear:** Procura pelas constantes do tipo SCRAP para um ano definido por parâmetro;
 - **PostConstant:** Criar um registo de constantes;
 - **EditConstant:** Edita um registo de constantes;
- **ScrapController:** Classe que recebe pedidos HTTP e gere os dados das instâncias de *SCRAP*;
 - **GetScrapByDate:** Procura pelo registo para um ano definido por parâmetro;
 - **PostScrap:** Criar um registo de SCRAP;
 - **EditScrap:** Edita um registo de SCRAP;
- **Constants:** Classes que representam uma constante no sistema;
 - **Id:** Valor numérico do tipo long que identifica a instância na base de dados;
 - **Name:** *String* com o nome da constante;
 - **Type:** *String* com o tipo de constante, neste caso este valor é “SCRAP”;
 - **Value:** Valor numérico do tipo decimal que representa o valor da constante;
 - **DateTime:** Data de validade da constante;
 - **InsertedAt:** Data de introdução do registo na base de dados;
 - **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
 - **UpdatedAt:** Data da última atualização do registo na base de dados;
 - **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **Scrap:** Classe que representa a quantidade percentual de SCRAP produzido por vários departamentos;

- **Id:** Valor numérico do tipo long que identifica a instância na base de dados;
- **SCRAPI:** Valor numérico do tipo decimal que representa o valor percentual do SCRAP produzido pelo Departamento 1;
- **SCRAPII:** Valor numérico do tipo decimal que representa o valor percentual do SCRAP produzido pelo Departamento 2;
- **SCRAPIII:** Valor numérico do tipo decimal que representa o valor percentual do SCRAP produzido pelo Departamento 3;
- **SCRAPIV:** Valor numérico do tipo decimal que representa o valor percentual do SCRAP produzido pelo Departamento 4;
- **SCRAPI_IV:** Valor numérico do tipo decimal que representa o valor percentual do SCRAP produzido por todos os departamentos;
- **DateTime:** Data de validade da constante;
- **InsertedAt:** Data de introdução do registo na base de dados;
- **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
- **UpdatedAt:** Data da última atualização do registo na base de dados;
- **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;

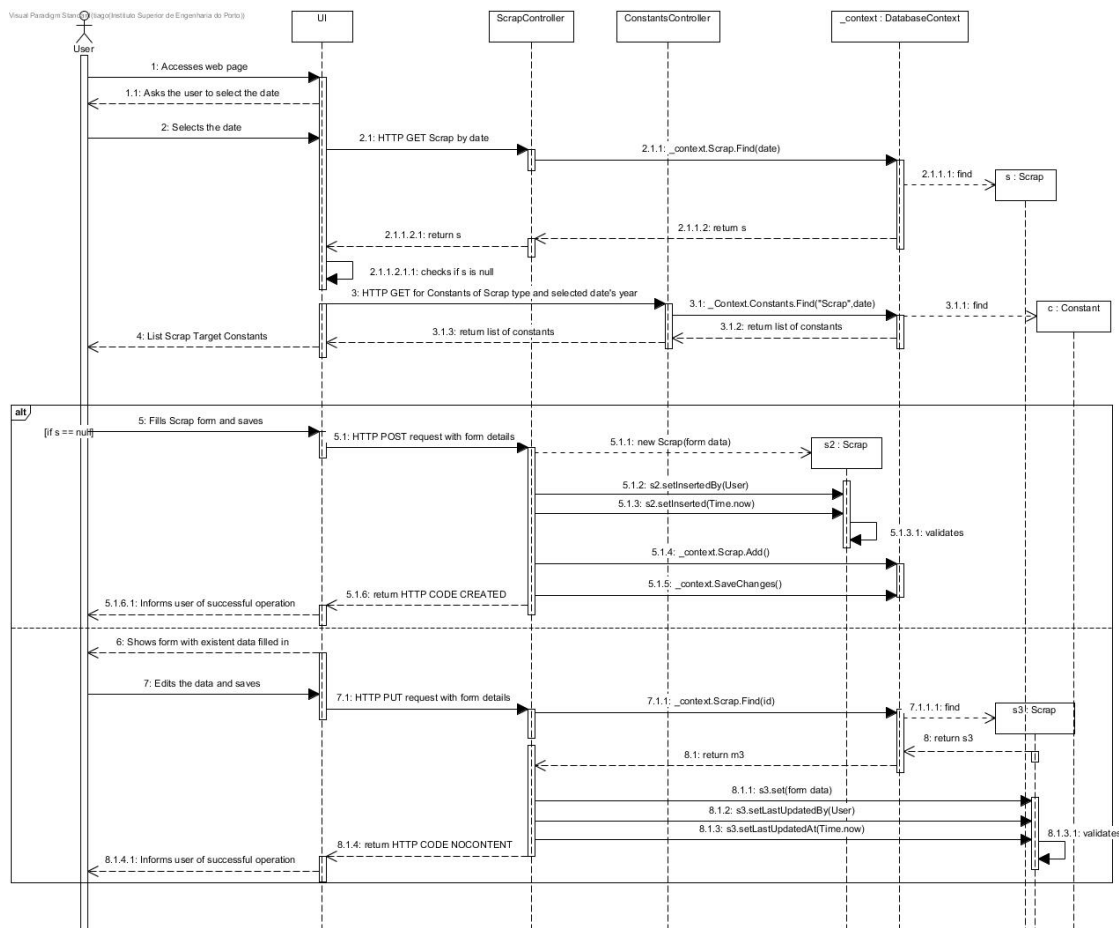


Figura 73 – Diagrama de sequencia do registo e edição de SCRAP

No diagrama apresentado na Figura 73 é descrito em detalhe o processo de gestão dos dados de *SCRAP*. O utilizador acede à página web e a interface pede para seleccionar uma data. O utilizador selecciona a data e a interface envia dois pedidos HTTP GET, um para o *ScrapController* com o objetivo de pesquisar se existem dados para a data seleccionada e, outro para o *ConstantsController* para apresentar as constantes do tipo “*Scrap*” e do ano da data seleccionada. Os controladores efetuam as respetivas pesquisas na base de dados e enviam a informação encontrada para a interface. A interface lista as constantes e apresenta um botão de registo/edição destas. Este processo encontra-se representado na Figura 63. Se existirem dados prévios de *SCRAP*, a interface apresenta o formulário preenchido e o utilizador edita os dados, caso contrário o utilizador introduz novos dados. A interface envia estes dados com um pedido *HTTP POST* ou *HTTP PUT* para o *ScrapController*. No caso de se tratar de um novo registo, controlador criar uma instância da classe *Scrap* com os dados recebidos e após validação guarda os dados na base de dados. No caso de se tratar de um processo de edição de um registo existente o controlador procura na base de dados pelo registo a editar, altera-lhe os dados e atualiza o registo na base de dados. O controlador responde à interface com o código HTTP correspondente e esta informa o utilizador do sucesso da operação.

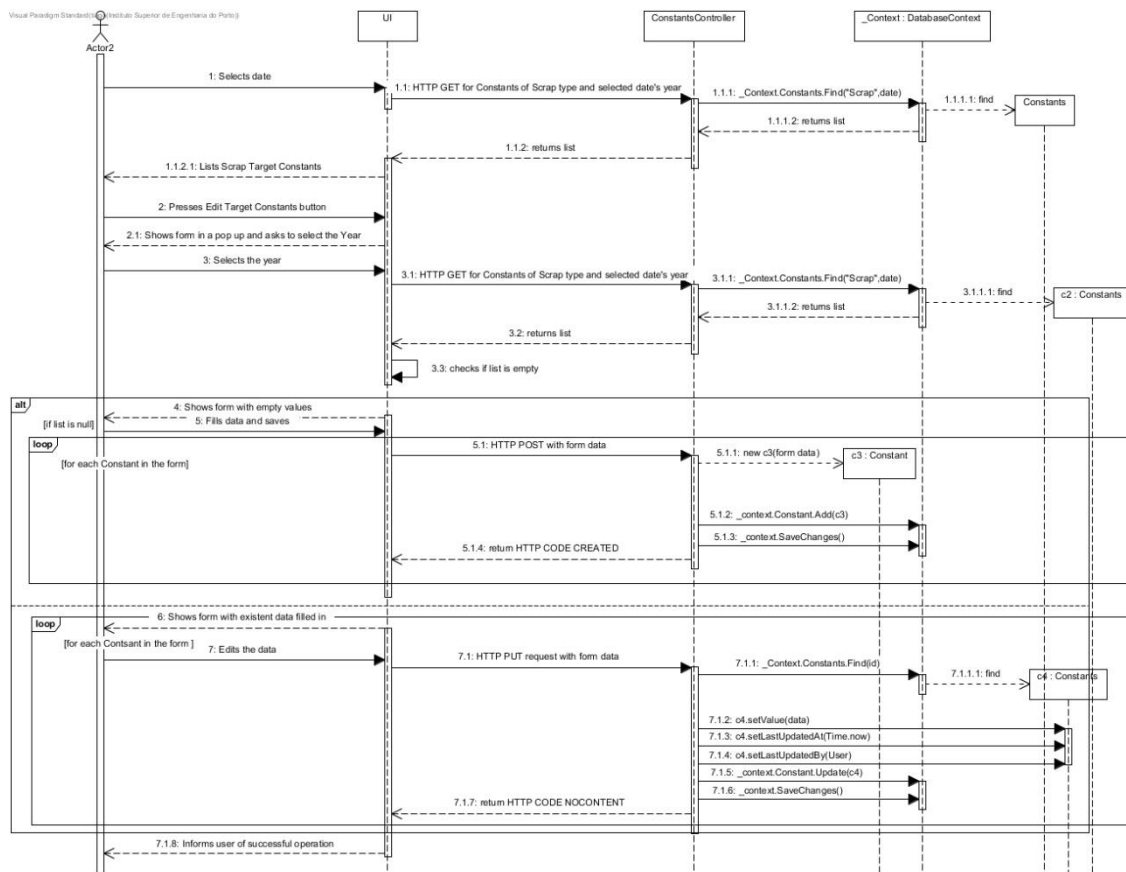


Figura 74 - Diagrama de sequencia de gestão de constantes do tipo Scrap

O utilizador, durante o processo de registo de SCRAP, pode decidir efetuar um novo registo ou editar as constantes que representam as metas de SCRAP de cada departamento. Este processo está definido no diagrama da Figura 74. O utilizador pressiona o botão de editar as constantes. A interface apresenta um formulário num *pop-up* e pede ao utilizador para selecionar o ano. O utilizador seleciona o ano e a interface envia um pedido HTTP GET para o controlador com o objetivo de pesquisar pelas constantes definidas para o ano selecionado e do tipo "Scrap". O controlador pesquisa na base de dados e envia a informação encontrada para a interface. Se não existir um registo prévio o formulário é apresentado vazio e o utilizador efetua um novo registo, caso contrário o utilizador edita os dados existentes. O utilizador conclui o processo ao guardar os dados. A UI envia um pedido HTTP POST ou HTTP PUT para cada constante definida. Neste processo são definidas as metas de SCRAPI, SCRAPII, SCRAPIII, SCRAPIV e SCRAPI_IV. O controlador introduz as novas instâncias ou atualiza as existentes na base de dados e responde ao pedido com o código HTTP correspondente. A interface informa o utilizador do sucesso da operação.

Scrap Category	Value	Scrap Target	Target Value
Scrap I:	0.14	Scrap I Target:	0.10
Scrap II:	0.74	Scrap II Target:	0.55
Scrap III:	0.14	Scrap III Target:	0.12
Scrap IV:	1.13	Scrap IV Target:	0.69
Scrap I-IV:	2.14	Scrap I-IV Target:	1.45

Figura 75- Página de registo de dados SCRAP

Figura 76- Página de gestão de metas de SCRAP

Na Figura 75 é apresentada a interface de registo de dados de SCRAP. Foi selecionado o dia 6 de maio de 2020. O sistema apresenta os valores das metas de SCRAP para o ano 2020 e o registo de SCRAP efetuado nesse dia. A qualquer momento o utilizador pode alterar as metas de SCRAP através do botão “*Edit Target Constants*”. Ao pressionar este botão é mostrada o formulário da Figura 76. O utilizador seleciona um ano e pode alterar os dados existentes ou, ao selecionar o ano 2021 pode registar as metas para esse ano.

5.1.7 UC21/UC22/UC23/UC24- Criar e editar dados do relatório de turno e de relatório da reunião da supervisão

Neste capítulo são definidos os casos de uso de criação e edição do relatório de turno. O utilizador de *Scheduling*, regista vários indicadores de produção para um determinado turno. São também definidos, devido à sua semelhança o processo de criação e edição do relatório da reunião da supervisão. No final de cada turno, os supervisores de produção têm uma reunião onde são reportados casos de absentismo (trabalhadores que faltaram ao trabalho, acidentes, avarias, materiais em falta, etc.

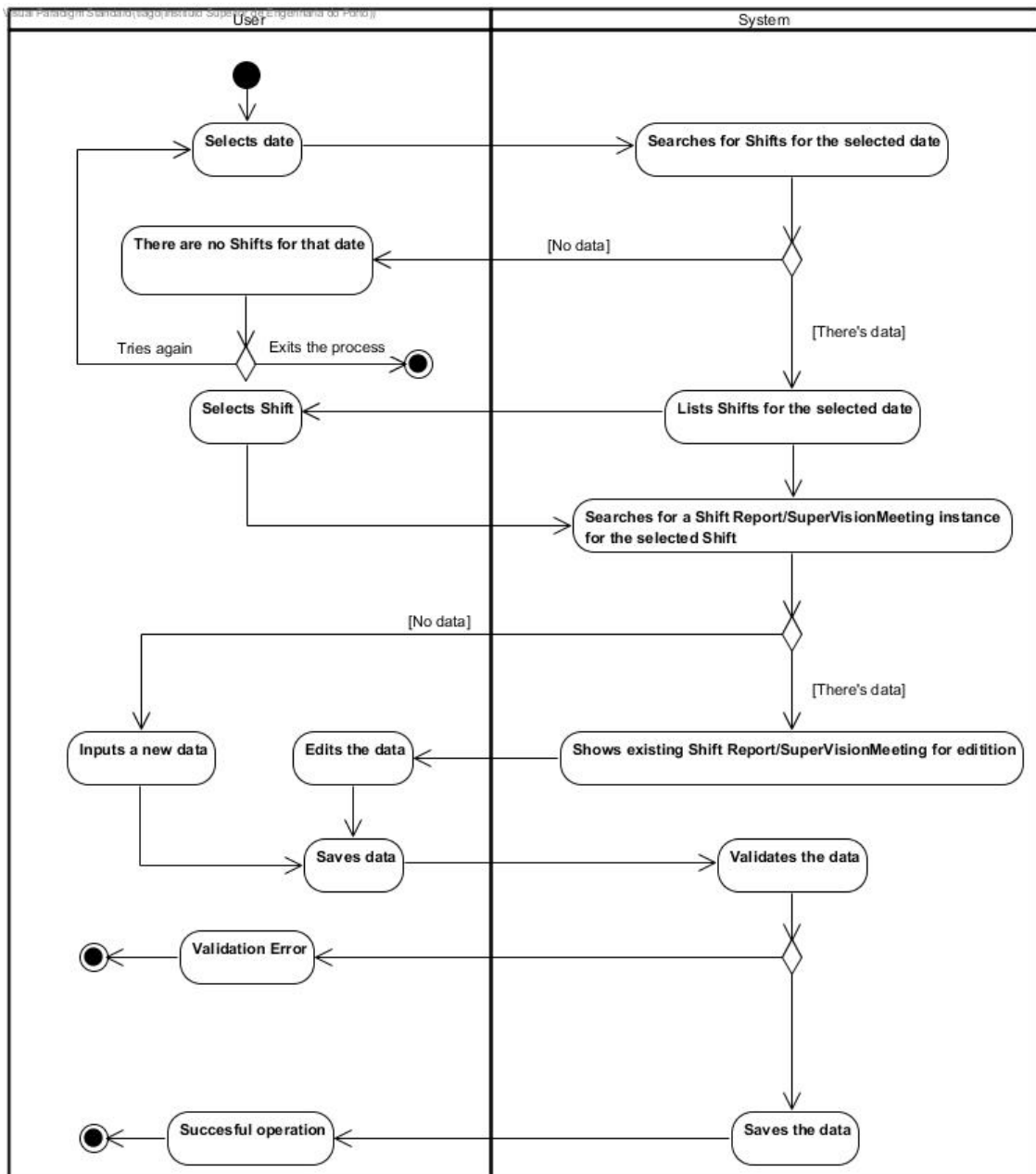


Figura 77 – Diagrama de atividades da gestão do relatório de turno e da reunião da supervisão

Na Figura 77 encontra-se representado o diagrama de atividades do processo de introdução e edição de dados do relatório de turno e da reunião da supervisão. O utilizador acede à página *web* e seleciona a data. O sistema apresenta os turnos definidos para essa data e pede ao utilizador para escolher um. O utilizador seleciona o turno e o sistema procura por dados existentes para o turno selecionado. Se existir, o sistema apresenta o formulário preenchido e, em caso contrário, apresenta-o vazio. O utilizador introduz ou edita os dados e pressiona o botão de guardar. O sistema valida e guarda os dados e informa o utilizador do sucesso da operação.

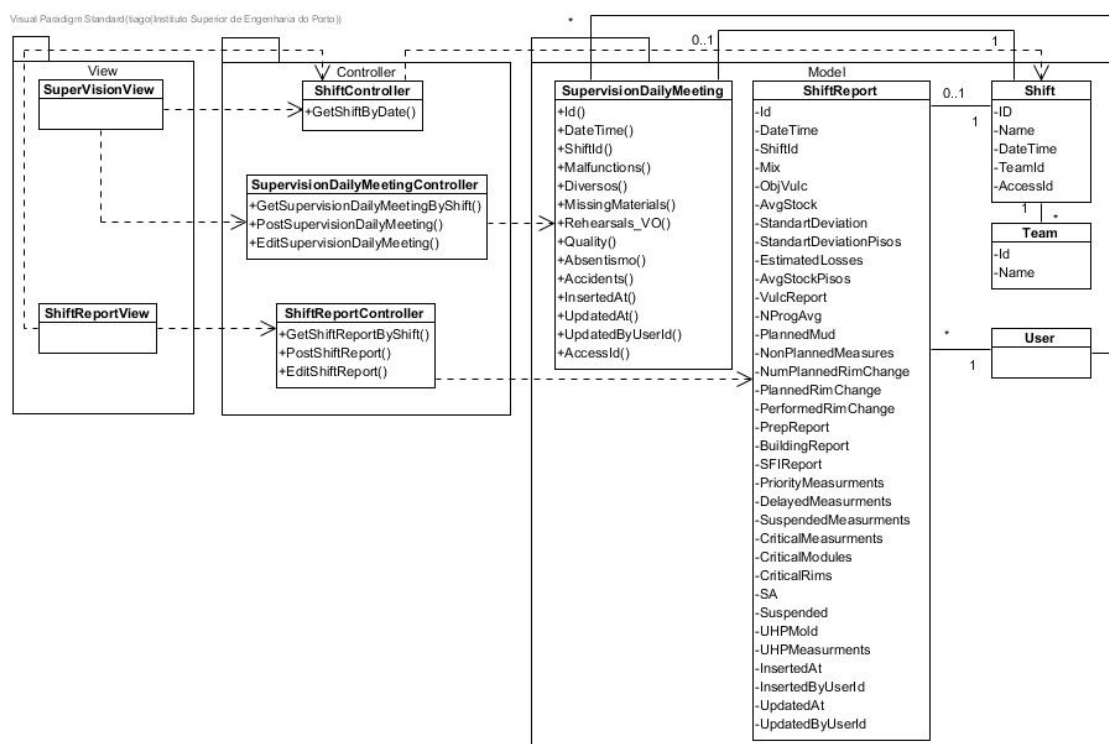


Figura 78 -Diagrama de classes com as classes reponsáveis pela gestão dos UCs 21 a 24

Na Figura 78 é possível identificar as seguintes classes:

- **SuperVisionView**: Interface gráfica para os dados da reunião da supervisão;
- **ShiftReportView**: Interface gráfica para os dados dos relatórios de turno;
- **ShiftController**: Classe que recebe pedidos HTTP e gere os dados das instâncias de *Shift*;
- **SuperVisionDailyMeetingController**: Classe que recebe pedidos HTTP e gere os dados das instâncias de *SupervisionDailyMeeting*;
- **ShiftReportController**: Classe que recebe pedidos HTTP e gere os dados das instâncias de *Shiftreport*;
- **SuperVisionDailyMeeting**: Classe que representa uma reunião da supervisão
 - **Id**: Valor numérico do tipo long que identifica a instância na base de dados;
 - **DateTime**: Data de validade da constante;
 - **ShiftId**: Valor numérico do tipo long que identifica o turno;

- **Malfunctions Accidents:** Campos de texto;
- **InsertedAt:** Data de introdução do registo na base de dados;
- **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
- **UpdatedAt:** Data da última atualização do registo na base de dados;
- **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **ShiftReport:**
 - **Id:** Valor numérico do tipo *long* que identifica a instância na base de dados;
 - **DateTime:** Data de validade da constante;
 - **ShiftId:** Valor numérico do tipo *long* que identifica o turno;
 - **Mix UHPMeasurements:** Campos de texto;
 - **InsertedAt:** Data de introdução do registo na base de dados;
 - **InsertedBy:** Variável do tipo *long* que identifica o utilizador que efetuou o registo na base de dados;
 - **UpdatedAt:** Data da última atualização do registo na base de dados;
 - **UpdatedBy:** Variável do tipo *long* que identifica o utilizador que efetuou a última atualização na base de dados;
- **Shift:** Classe que representa um turno;
- **Team:** Classe que representa uma equipa;
- **User:** Classe que representa um utilizador.

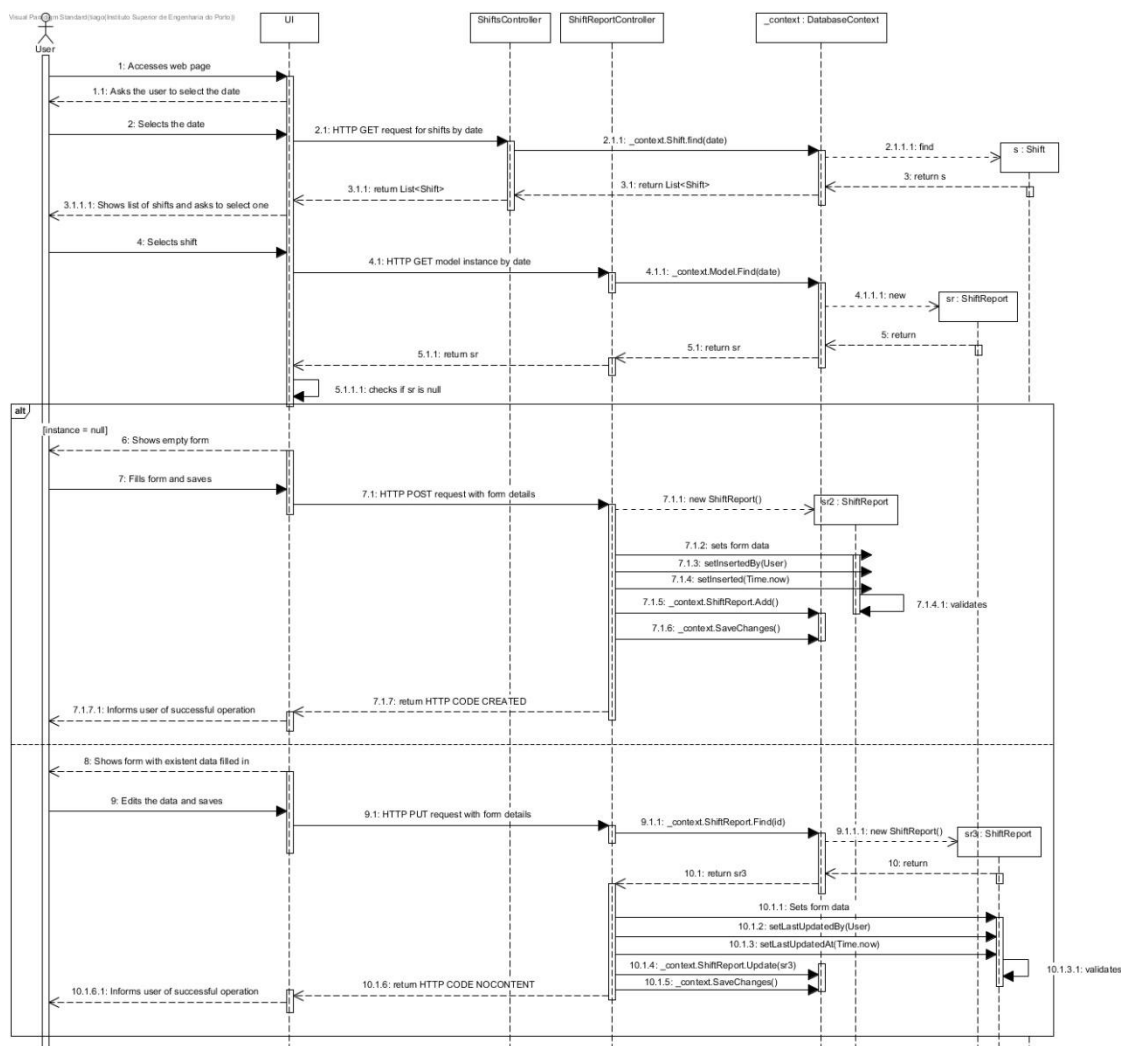


Figura 79 – Diagrama de sequencia da criação e edição de um *ShiftReport*

Na Figura 79 encontra-se definido o processo de gestão de um relatório de turno. Este diagrama pode ser usado para representar o mesmo processo para a reunião da supervisão. O utilizador acede à página web e seleciona uma data. A interface envia um pedido *HTTP GET* ao *ShiftsController* para receber os dados dos turnos para a data selecionada. O controlador pesquisa na base de dados pelos turnos correspondentes à data selecionada e envia-os para a interface. A UI lista os turnos e pede ao utilizador para selecionar um turno. Após a escolha do turno a interface envia um pedido *HTTP GET* para o controlador para pesquisar se já existe um registo prévio na base de dados. O controlador executa a *query* e responde ao pedido da interface enviando o resultado da pesquisa. Caso exista registo prévio o formulário é preenchido com a informação existente e o utilizador edita os dados e, em caso contrário o utilizador introduz nova informação. O sistema envia os dados para o controlador utilizando os pedidos *HTTP POST* ou *PUT*. O controlador recebe os dados e depende do tipo de pedido cria uma instância do *ShiftReport* ou procura na base de dados pelo registo existente e atualiza os dados. De seguida, o controlador responde ao pedido utilizando o código HTP correspondente e a interface informa o utilizador do sucesso da operação.

Supervision Daily Meeting

Search

Select a Date
09/01/2020

Select a Shift
Shift 2 Team B

Malfunctions

Machine	Time	Type	Description
km8	2	Elet.	bloco
km11	0.3	Mec.	fixador talões

+ -

Absent

Department
DP2Q

Absent
2

+ -

Topics	Quality	Missing Materials/Green tires	AI/ETO/SOP/OE's
47 - Med.1942 - sem comp T-15026 19 - 4h mtc 26 - upgrade cgms q16 - há 3 dias com flash PARADA p21/22 n35/36 mp q23/24 - 30m transf.	38 unif	pisos: 730	46 oe 1047 39 ai med 1412 20 oe med 594 q16 e o35

Save Back

Figura 80 – Registo de uma reunião da supervisão

Na Figura 80 é mostrada a interface gráfica utilizada para as reuniões da supervisão. O utilizador adiciona uma *malfunction* ou absentismo pressionando o botão “+” que adicionar uma nova linha para preencher. No caso oposto, ao pressionar o botão “-” a linha mais recente é eliminada. Estes dados são gravados como *strings* no formato JSON. Neste caso foi acedido ao registo do turno 2 do dia 9 de janeiro de 2020 e é possível editar os dados. A interface do relatório de turno foi implementada de maneira semelhante.

5.1.8 UC25- Gerar relatórios

O utilizador utiliza a informação registada no sistema dos casos de uso anteriores para gerar relatórios informativos para uma melhor gestão dos dados e partilha de informação.

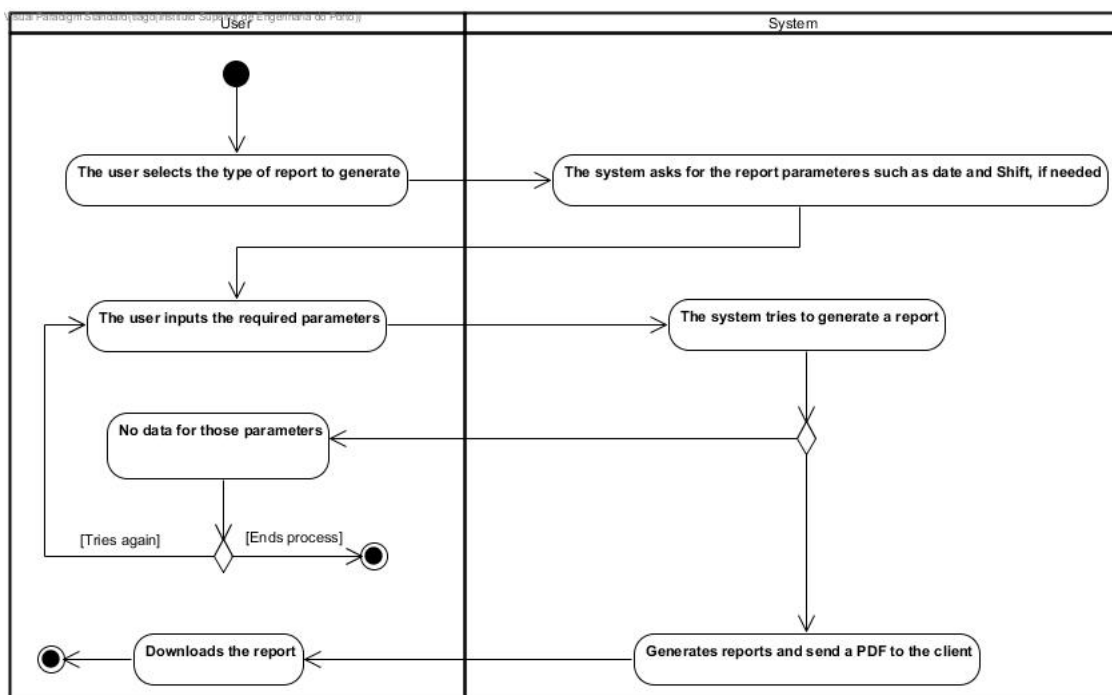


Figura 81 – Diagrama de atividades da geração de relatórios

Na Figura 81 encontra-se representado o diagrama de atividades do processo de geração de relatórios. O utilizador acede à página web e o sistema apresenta uma lista de tipos de relatórios e a data. O utilizador seleciona o tipo de relatório e a data. Se for necessário, dependendo do tipo de relatório, o sistema apresenta uma lista de turnos para a data selecionada. Após a definição dos parâmetros necessários o sistema gera o relatório e envia o ficheiro PDF para o *browser* do utilizador. O utilizador efetua o *download* do relatório e visualiza a informação gerada.

Existem 5 tipos de relatórios:

- **Mapa de Produção diário:** Relatório que lista todas as *Machines* e *Stocks* e as quantidades produzidas para cada uma para os vários turnos de um dia. Apenas necessita de receber a data com parâmetro;
- **Mapa de Produção de fim de semana:** Relatório que lista todas as *Machines* e *Stocks* e as quantidades produzidas para cada uma para os vários turnos do fim de semana. Apenas necessita de receber a data com parâmetro;
- **Resumo de Produção diário:** Relatório que, para um dia selecionado como parâmetro, apresenta os valores de produzidos, *Scrap*, *Workoff* e *APA* do dia anterior e os dados programados do dia selecionado;
- **Resumo de produção de fim de semana:** Relatório que, para uma segunda-feira como parâmetro, apresenta os valores de produzidos, *Scrap*, *Workoff* e *APA* do fim de semana anterior e os dados programados do dia selecionado;
- **Relatório de turno:** Relatório que apresenta os dados da classe *ShiftReport*; Necessita de uma data e um turno como parâmetro;

- **Reunião diária da supervisão:** Relatório que apresenta os dados da classe *SupervisionDailyMeeting*, dados de quantidade de *Stocks* e alguns dados do relatório de turno. Necessita de uma data e um turno como parâmetro;

Os relatórios foram desenvolvidos utilizando a ferramenta Microsoft Reporting Services Projects para o Visual Studio (Microsoft, 2020f). Esta ferramenta permite, através de um sistema *drag-and-drop* desenhar os relatórios desejados. Após a construção do relatório são definidos *datasets* que pesquisam na base de dados de acordo com os parâmetros definidos. Finalmente, os relatórios são publicados num SQL Server Reporting Services (SSRS)(Microsoft, 2020g) alojado no mesmo servidor que a base de dados do sistema.

O aspeto final dos relatórios encontra-se no Anexo 2.

5.2 Migração de dados

Os dados da aplicação existente encontram-se guardados nas tabelas da ferramenta Access. A aplicação tem 26 tabelas registadas. Foi necessário analisar o conteúdo de cada uma para analisar a estrutura dos dados. Das 26 tabelas, 12 foram descartadas por estarem vazias ou apenas possuírem dados relativos a alguns dias, o que pode significar que provavelmente foram utilizadas para efeitos de teste e nunca foram apagadas. As restantes tabelas possuem dados a partir do ano 2011 que teriam de ser migrados.

Para migrar os dados as tabelas restantes foram exportadas para ficheiros Excel para serem lidos pela aplicação e guardados na base de dados. Para conseguir ler os ficheiros foi usada a ferramenta ExcelDataReader (GitHub, 2020).

```
1 references | 0 exceptions
public void ImportShifts(string filePath)
{
    using (var stream = File.Open(filePath, FileMode.Open, FileAccess.Read))
    {
        using (var reader = ExcelReaderFactory.CreateReader(stream))
        {
            //skips first row
            reader.Read();

            do
            {
                while (reader.Read())
                {
                    Shift shift = new Shift();
                    shift.AccessId = (Convert.ToInt32(reader.GetValue(0)));
                    shift.DateTime = reader.GetDateTime(1);
                    shift.Name = reader.GetString(2);
                    var teamName = reader.GetString(3);
                    shift.Team = _context.Team.Where(x => x.Name == teamName).First();
                    _context.Shift.Add(shift);
                    Console.WriteLine(reader.GetValue(0));
                } while (reader.NextResult());
                _context.SaveChanges();
            }
        }
    }
}
```

Figura 82 – Excerto de código de migração dos turnos

Na Figura 82 é apresentado o código da migração da tabela que contem os dados dos turnos. O código lê cada linha do ficheiro Excel, cria uma instância de turno com os dados da linha e adiciona a instância à base de dados. Este processo foi repetido para todas as tabelas da aplicação Access. O processo demorou alguns dias devido à elevada quantidade de informação guardada nas tabelas, algumas com mais de 80 mil linhas.

5.3 Testes

Neste subcapítulo são descritos os processos de teste para verificação do correto funcionamento dos métodos e funções da aplicação. São apresentados 3 tipos de testes: unitários, de integração e de aceitação.

5.3.1 Testes unitários

Para a implementação dos testes unitários foi utilizada a ferramenta de testes xUnit(xUnit, 2020). Foram desenvolvidos testes unitários para todas as funções atómicas, ou seja, que não incluem chamadas de outros métodos ou acessos às bases de dados. Estes testes abrangem principalmente métodos *get*, *set* e métodos de validação.

```
1 reference
public class UserTest
{
    public User user;
    0 references | 0 exceptions
    public UserTest()
    {
        user = new User();
    }
    [Fact]
    ✓ | 0 references | 0 exceptions
    public void getsetId()
    {
        user.Id = 1;
        Assert.True(user.Id == 1, "should be 1");
    }
    [Fact]
    ✓ | 0 references | 0 exceptions
    public void getsetName()
    {
        string name = "test";
        user.Name = name;
        Assert.True(user.Name == name, "should be " + name);
    }
}
```

Figura 83 – Testes unitários aos métodos *get* e *set* da classe *User*

O código ilustrado na Figura 83, utiliza a *tag* “[Fact]” para identificar os casos de teste. No início é inicializado o utilizador com os seus atributos vazios e para cada um deste é alterado o valor e é verificado se a alteração foi bem efetuada.

Foram implementados 220 testes unitários que incluem os métodos das classes de modelo e classes auxiliares.

5.3.2 Testes de integração

Os testes de integração foram desenvolvidos usando a ferramenta xUnit tal como os testes unitários. Estes testes cobrem os métodos que necessitam de vários componentes para a sua execução, tais como métodos das classes do controlador e métodos que utilizem acesso à base de dados.

```
public class TestContext
{
    private TestServer _server;
    28 references | 7/14 passing | 0 exceptions
    public HttpClient Client { get; private set; }

    14 references | 7/14 passing | 0 exceptions
    public TestContext()
    {
        SetupClient();
    }

    1 reference | 0 exceptions
    private void SetupClient()
    {
        _server = new TestServer(new WebHostBuilder()
            .UseEnvironment("Development")
            .UseConfiguration(new ConfigurationBuilder()
                .SetBasePath("C:\\[redacted] Desktop\\Indicadores\\IndicadoresProducao\\IndicadoresTests")
                .AddJsonFile("appsettings.json")
                .Build()
            )
        );
        .UseStartup<Startup>());

        Client = _server.CreateClient();
    }
}
```

Figura 84- Inicialização do Cliente HTTP de teste

```
[Fact]
0 references | 0 exceptions
public async void Alogin()
{
    var context = new TestContext();
    using (var client = context.Client)
    {
        LogForm form = new LogForm()
        {
            username = "Teste1",
            password = "Teste12345"
        };

        var content = new StringContent(JsonConvert.SerializeObject(form), Encoding.UTF8, "application/json");
        var response = await client.PostAsync("/api/Users/login", content);
        dynamic jsonObject = JObject.Parse(response.Content.ReadAsStringAsync().Result);
        token = "Bearer " + (string)jsonObject.token;
        Assert.True(token.Length > 7); // if longer than Bearer+WHITESPACE
    }
}
```

Figura 85- Teste de integração da função de login

Na Figura 84 é efetuado o *setup* do cliente HTTP de teste que, como é possível verificar na Figura 85, é utilizado para efetuar pedidos HTTP aos métodos dos controladores, simulando assim a sua atividade normal.

Foram implementados 74 testes para todos os métodos dos controladores e com acesso à base de dados.

5.3.3 Testes de aceitação

Testes de aceitação são testes manuais que testam as funcionalidades do sistema para perceber se a aplicação satisfaz os critérios do cliente. Estes testes são realizados por caso de uso, testando vários cenários e avaliando se o resultado obtido é o desejado. Na Tabela 3 encontram-se listados os casos de teste agrupados por caso de uso.

Tabela 3 – Testes de Aceitação

Caso de uso/Teste	Resultado esperado	Confirmação
UC1: Registrar novo utilizador		
Efetuar registo do utilizador	Regista o utilizador	Passou
Efetuar registo com um username existente	Erro de validação	Passou
UC2: Efetuar login		
Login com credenciais corretas	Login com sucesso	Passou
Login com credencias erradas	Erro de login	Passou
Login com conta desabilitada	Erro de login	Passou
UC3: Editar dados de utilizadores		
Alterar dados de utilizador	Alteração com sucesso	Passou
Alterar username para um de outro utilizador	Erro na alteração	Passou
UC4: Mudança de password		
Introduz password e confirmação corretamente	Alteração com sucesso	Passou
Password/confirmação diferentes	Erro na alteração	Passou
Password/confirmação com menos de 8 chars	Erro na alteração	Passou
UC5: Introduzir novo turno		
Introduz novo turno	Registo com sucesso	Passou
Novo turno com nome já utilizado nesse dia	Erro de validação	Passou
UC6: Editar dados de turnos		
Edita dados do turno	Edição com sucesso	Passou
Edição com o nome já utilizado no mesmo dia	Erro de validação	Passou
UC7: Registrar nova máquina		
Regista nova máquina	Registo com sucesso	Passou
Nova máquina com nome já utilizado	Erro de validação	Passou
UC8: Editar dados de máquinas		
Edita máquina	Edição com sucesso	Passou
Edita máquina com nome já utilizado	Erro de validação	Passou
UC9: Registrar novo tipo de Stock		
Regista nova máquina	Registo com sucesso	Passou
Nova máquina com nome já utilizado	Erro de validação	Passou
UC10: Editar tipo de Stock		

Edita stock	Edição com sucesso	Passou
Edita stock com nome já utilizado	Erro de validação	Passou
UC11: Introduzir dados de produção		
Introduz dados de produção	Registo com sucesso	Passou
UC12: Editar dados de produção		
Edita dados de produção	Edição com sucesso	Passou
UC13: Introduzir dados de produção diária		
Introduz dados de produção diária	Registo com sucesso	Passou
UC14: Editar dados de produção diária		
Edita dados de produção diária	Edição com sucesso	Passou
UC15: Introduzir dados de programação		
Introduz dados de programação	Registo com sucesso	Passou
UC16: Editar dados de programação		
Edita dados de programação	Edição com sucesso	Passou
UC17: Introduzir dados de APA		
Introduz dados de APA	Registo com sucesso	Passou
UC18: Editar dados da APA		
Edita dados de APA	Edição com sucesso	Passou
UC19: Introduzir dados de SCRAP		
Introduz dados de SCRAP	Registo com sucesso	Passou
Introduz dados de metas de SCRAP	Registo com sucesso	Passou
UC20: Editar dados de SCRAP		
Edita dados de APA	Registo com sucesso	Passou
Edita dados de metas de SCRAP	Registo com sucesso	Passou
UC21: Introduzir dados do relatório de turno		
Introduz dados de relatório de turno	Registo com sucesso	Passou
UC22: Editar dados de relatório de turno		
Edita dados de relatório de turno	Edição com sucesso	Passou
UC23: Introduzir dados da reunião de supervisão		
Introduz dados de APA	Registo com sucesso	Passou
UC24: Editar dados da reunião de supervisão		
Edita dados da reunião de supervisão	Edição com sucesso	Passou
UC25: Introduz dados de Workoff		
Introduz dados de Workoff	Registo com sucesso	Passou
UC26: Editar dados de Workoff		
Edita dados de Workoff	Edição com sucesso	Passou
UC27: Gerar relatórios		
Gera relatórios corretamente	Geração com sucesso	Passou
UC28: Sugerir dados de preenchimento		
Sugere dados corretamente	Sugestão com sucesso	Passou

6 Avaliação

Neste capítulo será descrito o procedimento de avaliação da solução implementada. Primeiramente serão apresentadas as métricas de avaliação. De seguida, são mencionadas as hipóteses que serão testadas e as metodologias de avaliação.

6.1 Métricas

A aplicação é constituída por três componentes:

- **Front-end:** Aplicação responsável por fornecer uma interface gráfica para interação com o utilizador;
- **Reporting Server:** Responsável por aceder à base de dados, recolher os dados necessários para qualquer gerar qualquer relatório desejado pelo utilizador;
- **Back-end:** Aplicação responsável gerir os dados gravados na base de dados e fornecê-los ao *front-end* e vice-versa;

A aplicação de *front-end* será avaliada de acordo com o *feedback* do utilizador relativamente ao uso desta. Assim foi definida a primeira métrica:

- Satisfação do utilizador quanto à utilização e performance da aplicação.

Quanto ao *reporting server*, será apreciada a qualidade visual e rapidez de geração dos relatórios. Assim foi definida a segunda métrica:

- Satisfação do utilizador quanto à qualidade do sistema de geração de relatórios.

A aplicação de *back-end* será avaliada relativamente ao tempo que a aplicação necessita para responder a qualquer pedido que lhe é feito. Esta grandeza inclui o tempo necessário para recolher/gravar dados na base de dados, aplicar logica aplicada a estes dados e responder ao pedido. Assim foi definida a terceira métrica:

- Tempo total de execução e reposta.

6.2 Hipóteses

Após a definição das métricas no capítulo anterior, é necessário formular hipóteses para as testar. Relativamente à primeira métrica, visa-se avaliar o nível de qualidade da aplicação *front-end* por parte do utilizador. Assim, foi definida a hipótese:

H0: Satisfação do utilizador (*front-end*)

H1: Questionário de satisfação possuem um nível de satisfação superior a 3 (0-5)

H2: Questionário de satisfação não possuem um nível de satisfação superior a 3 (0-5)

Para a segunda métrica, de maneira similar à primeira métrica será avaliada o nível de qualidade dos relatórios gerados pelo ponto de vista do utilizador. Foi definida a hipótese:

H0: Satisfação do utilizador (*Reporting server*)

H1: Questionário de satisfação possuem um nível de satisfação superior a 4 (0-5)

H2: Questionário de satisfação não possuem um nível de satisfação superior a 4 (0-5)

Para as duas hipóteses anteriores foi escolhido o valor superior a 3, pois representa o nível neutro de satisfação do utilizador. Os campos das métricas com este valor poderão ser os primeiros a receber mudanças ou melhorias.

Finalmente, para a última métrica, será avaliado o tempo que o sistema necessita para aceder à base de dados, gerir os dados e responder aos pedidos feito pelo *front-end*. Como tal, a hipótese a testar é:

H0: Tempo de resposta

H1: O sistema demora menos de 1.5 segundos a responder

H2: O sistema demora mais de 1.5 segundos a responder

Foi escolhido o valor de 1.5 segundos porque permite identificar que se trata de um sistema de resposta rápida e eficaz.

6.3 Metodologia

De modo a avaliar as métricas e hipóteses apresentadas anteriormente serão utilizadas metodologias diferentes que são descritas neste subcapítulo. Assim, na seguinte tabela são declaradas, para cada hipótese, o método a utilizar.

Tabela 4 - Metodologias

Métrica/Hipótese	Metodologia
Satisfação do utilizador (<i>front-end e Reporting server</i>)	Inquéritos de satisfação
Tempo de resposta	Testes de integração

De modo a avaliar a satisfação dos utilizadores, foram disponibilizados dois inquéritos de satisfação aos principais utilizadores da aplicação, um para a apreciação do *front-end* e outro para avaliação das funcionalidades de geração de relatórios (*reporting server*). Para cada utilizador, foi realizada uma demonstração e explicada todas as funcionalidades da aplicação, permitindo a estes adquirir o conhecimento total das funcionalidades da aplicação. Os inquéritos utilizam a seguinte escala de classificação.

Tabela 5 - Escala de classificação dos inquéritos

Escala	Descrição
1	Pouco satisfatório(a)
2	Insatisfatório
3	Neutro
4	Satisfatório
5	Bastante satisfatório(a)

Quanto ao tempo de resposta foram implementados testes de integração que registam o tempo decorrido desde que o sistema recebe um pedido até obter a sua resposta.

6.4 Análise de Resultados

Neste subcapítulo, é feita uma análise dos resultados obtidos para cada uma das métricas definidas na secção 6.1 e é verificado se a aplicação implementada cumpre os casos definidos nas hipóteses da secção 6.2

6.4.1 Inquérito de satisfação ao utilizador

Para perceber se a aplicação corresponde ao que foi pedido pelo cliente, foi apresentada a aplicação e as suas funcionalidades a vários indivíduos e depois foi este responderam a um questionário. Este questionário, de preferência deveria ter sido preenchido pelos utilizadores finais, mas, entre o final do estágio e escrita da presente dissertação, aplicação ainda não estava a ser usada. Assim, obteve-se uma amostra bastante reduzida de 8 elementos que incluem, o cliente principal/futuro administrador da aplicação, os responsáveis pelo estágio e alguns colegas de trabalho do departamento de informática e do *Solution Development Center*.

6.4.1.1 Feedback do frontend

O questionário é composto por 12 asserções apresentadas na tabela seguintes com o valor percentual de respostas para cada valor da escala definida no ponto 6.3.

Tabela 6 – Questionário de feedback do frontend com percentagem de respostas

Asserções	1	2	3	4	5
1- O design da aplicação é	0%	0%	0%	75%	25%
2-A velocidade da aplicação é	0%	0%	0%	25%	75%
3-A aplicação, em termos de uso intuitivo, é	0%	0%	0%	50%	50%
4-A rapidez do processo de login é	0%	0%	0%	75%	25%
5-O processo de gestão de utilizadores, máquinas e turnos é	0%	0%	0%	50%	50%
6-O processo de registo de produções das máquinas e stocks é	0%	0%	0%	100%	0%
7-O sistema de sugestão de dados a partir do CGMS é	0%	0%	0%	50%	50%
8-A processo de registo de dados diários é	0%	0%	0%	75%	25%
9-O processo de registo de SCRAP é	0%	0%	0%	75%	25%
10-O processo de registo do relatório de turno	0%	0%	0%	50%	50%
11-O processo de registo da reunião diária da supervisão é	0%	0%	25%	75%	0%
12- A aplicação como melhoria em relação à plataforma existente é	0%	0%	0%	25%	75%

As asserções de 1 a 3 estão aplicadas ao aspeto e funcionamento da aplicação em termos gerais. Para a asserção 1 existem mais repostas na escala 4 do que 5, enquanto que na asserção 2 existem mais respostas na escala 5 do que 4 e na asserção 3 existem o mesmo número para as escalas 4 e 5. Com base nestes resultados pode-se deduzir que em termos de velocidade de execução a aplicação possui um bom tempo de resposta mas, por outro lado podem ser realizadas melhorias nos campos do design e uso intuitivo. Em termos gerais, pode-se afirmar que a interface tem bom aspeto e que é intuitiva e rápida.

As asserções de 5-11 representam várias funcionalidades da aplicação. Para as asserções 6, 8 e 9 existem mais respostas na escala 4 do que 5, para as asserções 7 e 10 apresentam o mesmo valor de repostas na escala 4 e 5 e a asserção 11 apresenta 25% de respostas na escala 3 e 75% na escala 4. Assim, pode-se deduzir que os utilizadores estão satisfeitos com as funcionalidades, mas que pode haver melhorias em todas elas, mas principalmente no processo de registo de reunião da supervisão.

Finalmente a asserção 12 possui 75% de respostas na escala 5 e 25% na escala 4. Com estes resultados, maior parte dos utilizadores está totalmente de acordo que a aplicação é uma melhoria em relação à aplicação existente.

Comparando com a hipótese formulada, a média das repostas tem de ser maior que 3, a média das respostas obtidas é de aproximadamente 4. Isto demonstra que os utilizadores estão satisfeitos em relação à qualidade da aplicação.

6.4.1.2 Feedback dos relatórios

O questionário é composto por 6 asserções apresentadas na tabela seguintes com o valor percentual de respostas para cada valor da escala definida no ponto 6.3.

Tabela 7- Questionário de feedback dos relatórios com percentagem de respostas

Questão	1	2	3	4	5
1-O processo de geração de relatórios é	0%	0%	10%	90%	0%
2-A rapidez de geração dos relatórios é	0%	0%	0%	75%	25%
3-A qualidade relatório do mapa de produção é	0%	0%	0%	100%	0%
4-A qualidade do relatório do resumo diário é	0%	0%	0%	100%	0%
5-A qualidade do relatório de turno	0%	0%	0%	100%	0%
6-A qualidade do relatório da reunião da supervisão é	0%	0%	0%	100%	0%

As asserções 1 e 2 cobrem o campo do processo de geração de relatórios. A asserção 1 apresenta 10% de respostas na escala 3 e 90% de respostas na escala 4 e a asserção 2 apresenta 75% de repostas na escala 4 e 25% na escala 5. Assim, é possível deduzir que o processo de geração de relatórios é bom, mas necessita de algumas melhorias.

As asserções 3 a 4 são aplicadas à qualidade dos relatórios produzidos. Todas as asserções obtiveram 100% de respostas na escala 4 o que significa que os relatórios são bem construídos e apresentam a informação de forma clara, mas, existe espaço para melhoria.

Comparando com a hipótese formulada, a média das repostas tem de ser maior que 3, a média das respostas obtidas é de aproximadamente 4. Isto demonstra que os utilizadores estão satisfeitos em relação à qualidade do processo de geração e aspeto dos relatórios.

6.4.2 Tempo de resposta

Para avaliar o tempo de resposta, os testes de integração descritos no ponto 5.3.2, foram executados 15 vezes com um computador portátil com um processador Intel Core i7-8665U CPU @ 1.90Ghz, com uma base de dados de testes alojada num servidor interno e com ligação à rede interna da Continental Mabor por cabo Ethernet. Os resultados estão apresentados na seguinte tabela.

Tabela 8 – Resultados dos testes de integração.

Tempo médio (ms)	Menor tempo(ms)	Maior tempo
358	118	1200

Comparando com a hipótese formulada, o tempo médio deve ser menor que 1.5 segundos, o tempo médio de execução é de 358 ms. Isto demonstra que o sistema cumpre com a hipótese definida ao respeitar o tempo limite de 1.5 segundos.

7 Conclusão

A escrita desta dissertação teve como objetivo o desenvolvimento de uma aplicação de gestão de indicadores, no âmbito de um estágio hospedado pela Continental Mabor. Esta aplicação foi implementada para os vários departamentos de produção da Continental Mabor possam gerir os seus indicadores, com formulários de introdução de dados e geração de relatórios.

Primeiramente, as circunstâncias deste projeto foram contextualizadas. Foi efetuada uma curta apresentação da empresa, seguida dos problemas da solução existente e os objetivos do estágio. De seguida, foi realizada uma análise que permitiu identificar o valor percebido, valor para o cliente e uma proposta de valor.

Foi também efetuado um levantamento do estado de arte, onde foi efetuada uma pesquisa sobre o funcionamento das aplicações web e das tecnologias mais utilizadas. Foi realizado uma análise à aplicação existente com o objetivo de extrair as funcionalidades a implementar e foi realizado um levantamento de soluções disponíveis no mercado que pudessem resolver o problema. Com o levantamento destes dados foi possível proceder para a fase de planeamento, onde foram definidos os *stakeholders*, os casos de uso e a arquitetura de aplicação.

De seguida, iniciou-se a fase de implementação do projeto. Todos os casos de uso foram implementados. Numa primeira fase o *backend* foi desenvolvido parcialmente e depois a parte correspondente ao *frontend*. Com este processo foi possível adquirir *feedback* do principal cliente da aplicação para melhor gerir o desenvolvimento da aplicação. Os dados da aplicação foram migrados com sucesso e a aplicação encontra-se disponível nos servidores da Continental Mabor.

Após a fase de implementação foi necessário avaliar o grau de satisfação e de desempenho da solução obtida. Foram efetuados inquéritos de satisfação aos utilizadores e a realização de testes para medir o tempo de resposta do servidor. Estes métodos de avaliação revelaram resultados positivos e pontos a melhorar.

Conclui-se que a realização deste estágio foi bastante produtiva e educativa e, em geral, uma experiência bastante positiva para o crescimento pessoal e profissional do estagiário.

De seguida são apresentados os objetivos alcançados e o trabalho futuro.

7.1 *Objetivos alcançados*

Nesta secção são apresentados os objetivos definidos no capítulo da Introdução e os casos de uso definidos no capítulo de Planeamento e quais destes é que forma totalmente implementados.

Tabela 9 – Objetivos do projeto

Objetivo	Resultado
Analisar os requisitos da aplicação através da escrita de um documento SRS	Atingido
Modelação de base de dados tendo em conta que os dados presentes na aplicação preexistente terão de ser migrados	Atingido
Implementação de uma solução WEB multiutilizador com todas as funcionalidades disponíveis na aplicação MSAccess	Atingido
Geração de relatórios	Atingido
Documentação a aplicação	Atingido
Integração a nova solução no sistema da Continental Mabor para providenciar preenchimento automático	Atingido

Na Tabela 9 é possível verificar que todos os objetivos foram cumpridos. O documento SRS foi escrito e está disponível no Anexo 1, a base de dados foi modelada e os registos existentes foram migrados. A aplicação foi implementada com as funcionalidades todas, com geração de relatórios e integração com os sistemas da Continental Mabor. A tese presente irá servir como documentação da aplicação.

Tabela 10 – Casos de uso implementados

Caso de uso	Resultado
UC1: Registrar novo utilizador	Atingido
UC2: Efetuar login	Atingido
UC3: Editar dados de utilizadores	Atingido
UC4: Mudança de password	Atingido
UC5: Introduzir novo turno	Atingido
UC6: Editar dados de turnos	Atingido
UC7: Registrar nova máquina	Atingido
UC8: Editar dados de máquinas	Atingido
UC9: Registrar novo tipo de Stock	Atingido
UC10: Editar tipo de Stock	Atingido
UC11: Introduzir dados de produção	Atingido
UC12: Editar dados de produção	Atingido
UC13: Introduzir dados de produção diária	Atingido

UC14: Editar dados de produção diária	Atingido
UC15: Introduzir dados de programação	Atingido
UC16: Editar dados de programação	Atingido
UC17: Introduzir dados de APA	Atingido
UC18: Editar dados da APA	Atingido
UC19: Introduzir dados de SCRAP	Atingido
UC20: Editar dados de SCRAP	Atingido
UC21: Introduzir dados do relatório de turno	Atingido
UC22: Editar dados de relatório de turno	Atingido
UC23: Introduzir dados da reunião de supervisão	Atingido
UC24: Editar dados da reunião de supervisão	Atingido
UC25: Introduz dados de Workoff	Atingido
UC26: Editar dados de Workoff	Atingido
UC27: Gerar relatórios	Atingido
UC28: Sugerir dados de preenchimento	Atingido

Na Tabela 10 são apresentados os casos de uso e é possível verificar que todos foram implementados com sucesso, de acordo com os resultados obtidos no capítulo de Resultados.

7.2 Trabalho futuro

Apesar de todos os casos de usos terem sido implementados, existe sempre possibilidade de melhorar o que foi implementado. De acordo com os resultados obtidos no ponto 6.4 existem várias funcionalidades que podem ser melhoradas.

- **Frontend:**
 - Melhorias na aparência geral;
 - Simplificação de alguns processos para tornar a aplicação mais intuitiva;
 - Melhoria na funcionalidade da reunião da supervisão;
- **Backend:**
 - Análise dos métodos com maior tempo de execução para poder reduzir o seu tempo de execução;
- **Relatórios:**
 - Melhorias na aparência;
 - Simplificação do processo de geração.

Referências

- Alawar, M.W., Naser, S.S.A., 2017. CSS-Tutor: An intelligent tutoring system for CSS and HTML.
- Bello, J., 2019. 8 Microsoft Access online alternatives worth looking at [WWW Document]. URL <https://www.hyperoffice.com/blog/2019/03/04/microsoft-access-online-alternatives/> (accessed 2.23.20).
- Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F., 2000. Extensible markup language (XML) 1.0.
- Continental, 2020a. Continental História [WWW Document]. URL <https://www.continental.com/pt-pt/empresa/hist%C3%B3ria> (accessed 10.3.20).
- Continental, 2020b. Valores [WWW Document]. URL <https://www.continental.com/pt-pt/empresa/corporate-strategy/os-nossos-valores-88924> (accessed 10.3.20).
- Continental, 2020c. Visão [WWW Document]. URL <https://www.continental.com/pt-pt/empresa/corporate-strategy/vision-mission-88892> (accessed 10.3.20).
- Continental, 2020d. Missão [WWW Document]. URL <https://www.continental.com/pt-pt/empresa/corporate-strategy/our-mission-88888> (accessed 10.3.20).
- Continental, 2019. Continental [WWW Document]. Continental The Future in Motion. URL <https://www.continental-pneus.pt/ligeiros> (accessed 2.9.20).
- Delta, 2020. Delta [WWW Document]. URL [delta.com](https://www.delta.com) (accessed 2.22.20).
- Facebook, 2020a. React.js [WWW Document]. URL <https://reactjs.org/> (accessed 2.22.20).
- Facebook, 2020b. Facebook [WWW Document]. URL <https://www.facebook.com> (accessed 2.22.20).
- Forbes, 2020. Forbes [WWW Document]. URL [forbes.com](https://www.forbes.com) (accessed 2.22.20).
- Fox, 2020. Fox47 [WWW Document]. URL [Fox47.com](https://www.fox47.com) (accessed 2.22.20).
- GitHub, 2020. ExcelDataReader [WWW Document]. URL <https://github.com/ExcelDataReader/ExcelDataReader> (accessed 10.11.20).
- Google, 2020. Angular [WWW Document]. URL <https://angular.io/features> (accessed 2.22.20).
- Hécio A. Soares, Raimundo S. Moura, 2015. A methodology to guide writing Software Requirements Specification document. Presented at the 2015 Latin American Computing Conference (CLEI), IEEE. <https://doi.org/10.1109/CLEI.2015.7360001>
- HyperOffice, 2020. Hyper Office [WWW Document]. URL <https://www.hyperoffice.com/> (accessed 2.23.20).
- IBM, 2020. IBM [WWW Document]. URL <https://www.ibm.com/> (accessed 2.22.20).
- Instagram, 2020. Instagram [WWW Document]. URL <https://www.instagram.com> (accessed 2.22.20).
- J. Chen, W. Cheng, 2016. Analysis of web traffic based on HTTP protocol, in: 2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). Presented at the 2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–5. <https://doi.org/10.1109/SOFTCOM.2016.7772120>
- jQuery, 2020. jQuery [WWW Document]. URL <https://jquery.com/> (accessed 10.11.20).
- Knack, 2020. Knack [WWW Document]. URL <https://www.knack.com/tour> (accessed 2.23.20).
- 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.” *Research-Technology Management* 44, 46–55. <https://doi.org/10.1080/08956308.2001.11671418>

- Koen, P.A., Ajamian, G.M., Boyce, S., Clamen, A., Fisher, E., Fountoulakis, S., Johnson, A., Puri, P., Seibert, R., 2002. Fuzzy front end: effective methods, tools, and techniques. The PDMA toolbook 1 for new product development.
- Laravel, 2020. Laravel [WWW Document]. URL <https://laravel.com/> (accessed 2.22.20).
- Michelson, B.M., 2006. Event-driven architecture overview. Patricia Seybold Group 2, 10–1571.
- Microsoft, 2020a. MSAccess [WWW Document]. URL <https://products.office.com/en-us/access> (accessed 2.10.20).
- Microsoft, 2020b. Microsoft [WWW Document]. URL <https://www.microsoft.com/pt-pt/> (accessed 2.10.20).
- Microsoft, 2020c. ASP.NET [WWW Document]. URL <https://dotnet.microsoft.com/apps/aspnet> (accessed 2.22.20).
- Microsoft, 2020d. Visual Studio [WWW Document]. URL <https://visualstudio.microsoft.com/> (accessed 10.11.20).
- Microsoft, 2020e. HMACSHA512 Class [WWW Document]. URL <https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.hmacsha512?view=netcore-3.1> (accessed 10.10.20).
- Microsoft, 2020f. Microsoft Reporting Services Projects [WWW Document]. URL <https://marketplace.visualstudio.com/items?itemName=ProBITools.MicrosoftReportProjectsforVisualStudio> (accessed 10.11.20).
- Microsoft, 2020g. What is SQL Server Reporting Services (SSRS)? [WWW Document]. URL <https://docs.microsoft.com/en-us/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-ver15> (accessed 10.11.20).
- Microsoft, 2019. Windows Server [WWW Document]. URL <https://www.microsoft.com/en-us/cloud-platform/windows-server> (accessed 2.21.20).
- Netflix, 2020. Netflix [WWW Document]. URL <https://www.netflix.com> (accessed 2.22.20).
- Node.js, 2020. Node.js [WWW Document]. URL <https://nodejs.org/en/about/> (accessed 2.22.20).
- Oluwatosin, H.S., 2014. Client-server model. IOSRJ Comput. Eng 16, 2278–8727.
- OpenJSFoundation, 2020. OpenJSFoundation [WWW Document]. URL <https://openjsf.org/> (accessed 2.22.20).
- Osterwalder, A., Pigneur, Y., 2003. Modeling value propositions in e-Business. Presented at the Proceedings of the 5th international conference on Electronic commerce, pp. 429–436.
- Osterwalder, A., Pigneur, Y., Oliveira, M.A.-Y., Ferreira, J.J.P., 2011. Business Model Generation: A handbook for visionaries, game changers and challengers. African journal of business management 5, 22–30.
- Paypal, 2020. Paypal [WWW Document]. URL <https://paypal.com> (accessed 2.22.20).
- Pezoa, F., Reutter, J.L., Suarez, F., Ugarte, M., Vrgoč, D., 2016. Foundations of JSON schema. Presented at the Proceedings of the 25th International Conference on World Wide Web, pp. 263–273.
- Ríos, J.M., Souto, N.P., 2019. Comparison of development methodologies in Web applications. Information and Software Technology 106238.
- Samsung Forward, 2020. Samsung Forward [WWW Document]. URL <https://www.samsung-forward.com> (accessed 2.22.20).
- Sánchez-Fernández, R., Iniesta-Bonillo, M.Á., 2007. The concept of perceived value: a systematic review of the research. Marketing theory 7, 427–451.
- Santos, J., 2020. Top 10 Online Database App Builders to Create your PM Tool [WWW Document]. URL <https://project-management.com/top-10-database-application-builders-for-creating-your-own-pm-tool/>

- Spiteri, J.M., Dion, P.A., 2004. Customer value, overall satisfaction, end-user loyalty, and market performance in detail intensive industries. *Industrial marketing management* 33, 675–687.
- StackOverflow, 2020. StackOverflow [WWW Document]. URL <https://stackoverflow.com/> (accessed 2.22.20).
- Startups, 2020. Startups [WWW Document]. URL <https://www.startups.com/> (accessed 2.22.20).
- Uber, 2020. Uber [WWW Document]. URL <https://uber.com> (accessed 2.22.20).
- UPS, 2020. UPS [WWW Document]. URL <https://www.ups.com/us/en/global.page> (accessed 2.22.20).
- Watmore, J., 2018. ASP.NET Core 2.2 - Simple API for Authentication, Registration and User Management [WWW Document]. URL <https://jasonwatmore.com/post/2018/06/26/aspnet-core-21-simple-api-for-authentication-registration-and-user-management#user-service-cs> (accessed 10.12.20).
- xUnit, 2020. xUnit [WWW Document]. URL <https://xunit.net/> (accessed 10.11.20).
- Yulair, 2020. Yulair [WWW Document]. URL <http://quebec.yulair.com/> (accessed 2.22.20).
- Zoho, 2020. Zoho [WWW Document]. URL <https://www.zoho.com/> (accessed 2.23.20).

Anexo 1 -SRS

Software Requirements Specification

for

Production Indicators Management

Version 1.0 approved

Prepared by Tiago Vilaca-EXT, Tiago Marques (uic46325)

Continental Mabor

November 2019

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Product Scope	1
1.4 References	1
2. Overall Description.....	2
2.1 Product Perspective	2
2.2 User Classes and Characteristics	2
2.3 Operating Environment	3
2.4 Design and Implementation Constraints	3
2.5 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	4
4. System Features	4
4.1 Register new user	4
4.2 Manage users	5
4.3 Manage machine information	6
4.4 Load machines list from SAP/CGMS	6
4.5 Input production data	6
4.6 Correct production data	7
4.7 Input Daily Production Summary	7
4.8 Daily production summary correction	8
4.9 System data suggestion	8
4.10 Import data from excel files	9
4.11 Write shift report (Scheduling)	9
4.12 Write supervision daily meeting report	9
4.13 Workoff	10
4.14 Report generation	10
5. Other Nonfunctional Requirements.....	11
5.1 Performance Requirements	11
5.2 Security Requirements	11
5.3 Software Quality Attributes	11
5.4 Business Rules	11
Appendix A: Glossary.....	12
API.....	12
Application programming interface, set of routines, protocols, and tools for building software application	12
ASP.NET.....	12
Open-source server-side web application framework designed for web development to produce dynamic web pages developed by Microsoft	12
Bootstrap.....	12
Open-source CSS framework directed at responsive, mobile-first front-end web development. ..	12
HTML.....	12
Hypertext markup language, standard for documents designed to be displayed in a web browser.	12
MSSQL.....	12

Microsoft SQL Server is a relational database management system developed by Microsoft.....12

Frontend.....12

Component of a web application composed of a graphical interface that interacts with the user. 12

Backend.....12

Component of a web application that is responsible for manipulating and storing data. Usually doesn't interact with the user.12

Appendix B: Reports13

1. Daily Production Report13

2. Weekend Production Report14

3. Daily Report Summary.....15

4. Weekend Daily Report Summary.....16

5. Shift Report.....17

6. Daily supervision report.....18

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Production Indicators Management application. It will describe the purpose, features and constraints of the system. This document is intended for stakeholders/key users as well as the developers of the system.

1.2 Document Conventions

This document consists of 6 chapters:

- The first chapter “Introduction” where the purpose of the document and project scope are presented;
- The second chapter “Overall Description” where a more detailed description of the software attributes is showed;
- The third chapter “External Interface Requirements” details external interface requirements such as User, Hardware, Software and Communication interfaces;
- The fourth chapter “System features” lists all the system features to implement;
- The fifth chapter “Other Nonfunctional Requirements” specifies requirements that judge the operation of the system rather than specific functions.

The document uses a Times font with varying sizes for headers and text. Titles use size 18, sub titles use size 14 and regular text uses size 12. Important keywords are marked using a **Bold** highlight.

1.3 Product Scope

This software will be developed as a replacement of an outdated Production Indicators Management MS Access database. The software described in this document will be used to register all data received manually by users or automatically from central systems with updated information about the results of the production lines and to generate reports.

The main goal of this software is to increase efficiency and user experience by removing delays or crashes related to the use of a MS Access database by multiple users.

1.4 References

Daily production Report – Appendix B 1. Daily Production report
Weekend Production Report - Appendix B 2. Weekend Production report
Daily production summary - Appendix B 3. Daily Production report
Daily weekend production summary - Appendix B 4. Weekend Production report
Worker Shift Report - Appendix B 5. Weekend Production report
Daily supervision report - Appendix B 6. Weekend Production report

2. Overall Description

2.1 Product Perspective

Continental Mabor uses a small MSACCESS database that is loaded with production and operations indicators on a timely fashion way. This application records the production / operations summary of events (counts, machine events, etc.) occurred on every different area.

This application is based on a MS Access database that uses forms in order to receive data from the user. It has severe built-in limitations such as its lack of security, small maximum data storage (max of 2 GB) and limited scalability. If the current application was used by only one user it would work correctly but, because multiple users from multiple departments have access to this software, delays and crashes occur due to deadlocks in the database. Because of this, the application has major slowdowns, bad performance and it takes a long time to perform maintenance. Finally, there's no interface with the real live production systems that allows it to receive data directly from them. So, the development of a new web application with a Robust Centralized RDBMS was required. The application will allow users to fill their required production and scheduling forms with the necessary data and generate reports in a multiuser environment without the fear of delays or crashes. In order to improve efficiency, the application will also enable external interfaces to load automatically all data production possible.

2.2 User Classes and Characteristics

Production: User that belongs to the production team

- Input production data
- Allow automatic data fill up
- Import/Export with Excel files

Scheduling: User that belongs to the scheduling team

- Write shift report
- Input daily production data summary

Superintendent: User of higher status than the Production and Scheduling

- Write supervision daily meeting report

Admin: User with administration privileges

- Generate Daily Reports
- Generate Weekend Reports
- Manage machine information
- Manage users (password reset and roles management)
- Input production data
- Input daily production data summary
- Write shift report
- Write supervision daily meeting report
- Allow automatic data fill up
- Import/Export with Excel files

2.3 Operating Environment

The web application will be developed based on the following hardware and software components:

- Follows a model-view-controller architecture;
- The backend API will be developed using ASP.NET and the frontend API will use HTML, Bootstrap and jQuery;
- The database will be a relational database MSSQL;
- Implemented on local premises.

2.4 Design and Implementation Constraints

The following constraints were defined:

- The user must authenticate using their internal LDAP/MSSQL accounts;
- The interface must be similar to the other applications developed previously by the organization.

2.5 Assumptions and Dependencies

The backend API will depend on data received from other internal applications such as SAP and Continental Mabor MES (Manufacturing Execution System) application CGMS (Continental Global Manufacturing System).

3. External Interface Requirements

3.1 User Interfaces

Firstly, the user will input their login data on an email/password form and, after a successful login, all the options available to the user will be listed on the main menu. The options showed will depend on the role of the user. The user will then select the option that he desires, and a new menu will open depending on the selected feature. For every option listed there should be information available to help the user fulfil the task. At any point, the user can back out without saving. After completing a task, the user will be taken back to the main menu. If any errors occur a message should be displayed with information on how to fix and if the user can't find a fix an admin contact must be present.

3.2 Hardware Interfaces

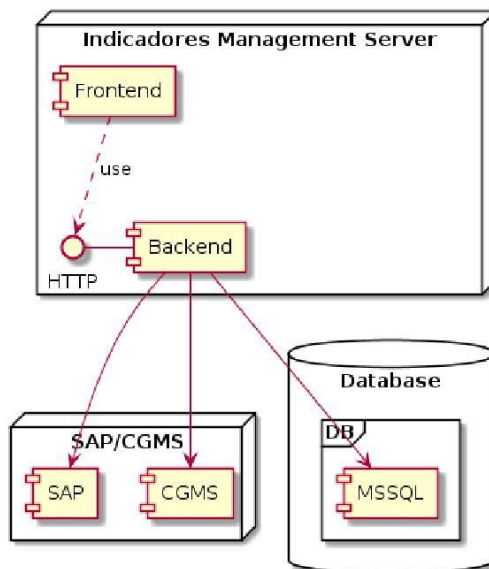
The frontend, backend and database will be hosted on a remote server meaning that any hardware can run the application as long as they have access to a web browser and internet connection.

3.3 Software Interfaces

The frontend, backend and database will be hosted on a remote server meaning that any operating system can run the application as long as they have access to a web browser and internet connection.

3.4 Communications Interfaces

The application front end will communicate with the backend through HTTP requests and the backend end will communicate with the database to save data inputted by the user. The backend will also receive information automatically from SAP and CGMS systems.



4. System Features

This chapter will describe all the system features presented in a logic sequence of execution. Each feature has 3 subchapters:

- Description: Short description of the feature
- Stimulus/Response Sequences: Listed User and System actions
- Functional requirements: Software capabilities that must be present for the user to carry out the services provided by the feature

4.1 Register new user

4.1.1 Description

The admin user can add new users and give them their respective role. **High priority.**

4.1.2 Stimulus/Response Sequences

1. The admin user accesses the add new user function;
2. The system asks for their internal LDAP account or if the user wants to create a MSSQL account;
3. The admin inputs their preferred option;
4. The system gets the necessary information if a LDAP account is used and asks for the role of the user or asks for user input if a MSSQL account is being used;
5. The admin completes the previous task;
6. The system asks to define the role of the user;
7. The user inputs the user role;
8. The system validates the choice, saves the data and informs the admin of the success of the operation.

4.1.3 Functional Requirements

REQ-1: The system must allow the admin to add new users using their internal LDAP account;
REQ-2: The system must allow the admin to create a new MSSQL account;
REQ-3: The system must allow the admin to assign a role to the new user.

4.2 Manage users

4.2.1 Description

The admin user can perform management task regarding the users. The admin can freely change the other users' permissions. The admin can, on request, reset passwords for the users. **Medium priority.**

4.2.2 Stimulus/Response Sequences

1. The admin user accesses the manage users' function;
2. The system shows a list of all the users;
3. The admin selects the user;
4. The system shows all the available options (change roles, reset passwords)
5. The admin selects the option;
6. The system asks for the necessary information for the completion of the task;
7. The user performs the task;
8. The system saves and informs the admin of the success of the operation.

4.2.3 Functional Requirements

REQ-1: The system must allow the admin to see the current role of the user;
REQ-2: The system must allow the admin to see all the available roles;
REQ-3: The system must allow the admin to change the role of the user ;
REQ-4: The system must allow the admin to reset a user's password;
REQ-5: The system must keep the password hidden through the entire process.

4.3 Manage machine information

4.3.1 Description

The system lists all the machines that are registered in the database and allows the admin to add new ones or edit machine information.

4.3.2 Stimulus/Response Sequences

1. The admin accesses the machine list option;
2. The system lists all the machines;
3. The user can add a new machine or edit information such as display names, order and division value;
4. The system validates information and saves the information.

4.3.3 Functional Requirements

REQ-1: The system must show a detailed list of all the available machines;

REQ-2: The system must allow the user to edit display name, list order or division value;

4.4 Load machines list from SAP/CGMS

4.4.1 Description

The system loads a list of machines obtained from SAP into its database.

4.4.2 Stimulus/Response Sequences

This requirement normally is automatically done by the system, but the user can force an update of the machine list.

- The user accesses the force update function;
- The system requests SAP/CGMS machines information;
- The system receives information and adds it to the database;
- The system informs the user of the success of the operation;

4.4.3 Functional Requirements

REQ-1: The system must correctly register all the machines received via SAP/CGMS;

REQ-2: The system must only add new machines to the database;

REQ-3: The system must allow the admin to manually force an update of the machine list.

4.5 Input production data

4.5.1 Description

The user accesses the production data option and inputs the required data. **High priority**.

4.5.2 Stimulus/Response Sequences

1. The user accesses the production data option;
2. The system checks which shift, team and department the user belongs too and shows the respective data to fill in;
3. The user fills in the data;
4. The system validates, saves the data and informs the user of the success of the operation.

4.5.3 Functional Requirements

REQ-1: The system must allow the user to access their respective forms depending on their shift, team and department;

REQ-2: The system must allow the user to fill in all the required data.

4.6 Correct production data

4.6.1 Description

The admin user can at any time edit the data that was introduced by the other users. **Medium priority.**

4.6.2 Stimulus/Response Sequences

1. The admin user accesses the production data;
2. The system shows all the fields that the admin can edit;
3. The admin edits the data;
4. The system validates the data and asks for confirmation;
5. The admin confirms the operation;
6. The system saves the data and informs the user of the success of the operation.

4.6.3 Functional Requirements

REQ-1: The system must allow the admin to access any production data filled in by the other users;

REQ-2: The system must allow the admin to make any necessary corrections to the production data filled in by users.

4.7 Input Daily Production Summary

4.7.1 Description

The user accesses the daily production summary option and inputs the required data. **High priority.**

4.7.2 Stimulus/Response Sequences

1. The user accesses the daily production summary option;
2. The system checks which shift, team and department the user belongs too and shows the respective data to fill in.
3. The user fills in the data.
4. The system validates, saves the data and informs the user of the success of the operation.

4.7.3 Functional Requirements

REQ-1: The system must allow the user to access their respective forms depending on their shift, team and department;
REQ-2: The system must allow the user to fill in all the required data.

4.8 Daily production summary correction

4.8.1 Description

The admin user can edit the information registered in the daily production summary. **Medium priority.**

4.8.2 Stimulus/Response Sequences

1. The admin user accesses the daily production summary data;
2. The system shows all the fields the admin can edit
3. The admin edits the data;
4. The system validates the data and asks for confirmation;
5. The admin confirms the operation;
6. The system saves the data and informs the user of the success of the operation.

4.8.3 Functional Requirements

REQ-1: The system must allow the admin to access data filled in by the other users;
REQ-2: The system must allow the admin to make any necessary corrections to the production data filled in by users.

4.9 System data suggestion

4.9.1 Description

The user can select to have the data on the formularies be partially filled up by the system in order to help the user fill their reports faster. The data comes from **CGMS/SAP**. **Medium priority.**

4.9.2 Stimulus/Response Sequences

1. The user accesses any production data form;
2. The system shows an option to automatically fill the data if possible;
3. The user chooses to automatically fill data;
4. The system fills the fields with data and allows the user to edit the new information;
5. The user verifies the data and makes changes if necessary;
6. The system validates the data and asks the user for confirmation;
7. The user confirms the operation;
8. The system saves the data and informs the user of the success of the operation.

4.9.3 Functional Requirements

REQ-1: The system must allow the user to select the option to fill the fields automatically;
REQ-2: The system must when prompted fill automatically all the possible fields.

4.10 Import data from excel files

4.10.1 Description

The user can import information on the production formularies by providing a Microsoft Excel sheet. **Medium priority.**

4.10.2 Stimulus/Response Sequence

1. The user accesses any production data form;
2. The system shows an option to upload an Excel file;
3. The user uploads an Excel file;
4. The system reads the file, inputs the data into the form and asks the user for confirm the imported data;
5. The user confirms the operation;
6. The system saves the data and informs the user of the success of the operation.

4.10.3 Functional Requirements

REQ-1: The system must allow the user to fill in any production data by uploading an Excel file.

4.11 Write shift report (Scheduling)

4.11.1 Description

The user accesses the write shift report option and writes the report. **High priority.**

4.11.2 Stimulus/Response Sequences

1. The user accesses the shift report option;
2. The system checks which shift, team and department the user belongs to and shows the report form to fill.
3. The user fills in the data.
4. The system validates and saves the data.

4.11.3 Functional Requirements

REQ-1: The system must allow the user to access their respective forms depending on their shift, team and department;

REQ-2: The system must allow the user to fill in all the required data of the shift report.

4.12 Write supervision daily meeting report

4.12.1 Description

The superintendent user writes the supervision daily meeting report. **High priority.**

4.12.2 Stimulus/Response Sequences

1. The user accesses the supervision daily meeting report option;
2. The system checks which shift, team and department the user belongs too and shows the report form to fill.
3. The user fills in the data.
4. The system validates and saves the data.

4.12.3 Functional Requirements

REQ-1: The system must allow the user to access their respective forms depending on their shift, team and department;

REQ-2: The system must allow the user to fill in all the required data of the shift report.

4.13 Workoff

4.13.1 Description

The user can access information about the workoff generated from the production line. **Medium Priority.**

4.13.2 Stimulus/Response Sequences

1. The user accesses the Workoff option;
2. The system checks which shift, team and department the user belongs too and shows the report form to fill.
3. The user fills in the data.
4. The system validates and saves the data.

4.14 Report generation

4.14.1 Description

The user accesses the report generation option and generates their desired report. This includes data from the production lines, scheduling and Workoff. **High priority.**

The most important report types are:

- Daily production report map;
- Weekend production report map;
- Daily production data;

4.14.2 Stimulus/Response Sequences

1. The user accesses the production report option;
2. The system asks for the type of report to generate (e.g. Daily Production, Weekend Production);
3. The user choses the type of report;
4. The system asks the date of the report;
5. The user inputs the date;
6. The system collects all the required data for that date and builds the report page.

4.14.3 Functional Requirements

REQ-1: The system must allow the user to choose the type of the report and the options available must be based on the role of the user.

REQ-1: The system must allow the users to generate their required reports;

REQ-2: The system must allow the report to be printed;

REQ-3: The system must allow the report to be exported as a PDF.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Any action performed by the system must be executed as quickly and effectively as possible. The application must be prepared for when there are times with higher usage the response time remains almost the same or isn't greatly affected.

5.2 Security Requirements

The application will use the Internal Microsoft Office accounts for the login process. Any sensitive data, such as passwords, must be encrypted and protected.

5.3 Software Quality Attributes

Adaptability: The application will show different options depending on the role of the logged user.

Availability: The application will be implemented to allow the maximum uptime possible.

Scalability: The application must support future enhancements, changes and new features.

Usability: The user interface must be designed in order to be used easily by the user.

5.4 Business Rules

The application has 3 main roles:

- Production: Only has access to production related features (e.g. 4.6 or 4.8);
- Scheduling: Only has access to scheduling related features (e.g. 4.12);
- Superintendent: Writes the supervision daily meeting report.
- Admin: Has access to all the features including the ones used by the other roles.

Appendix A: Glossary

API	Application programming interface, set of routines, protocols, and tools for building software application.
ASP.NET	Open-source server-side web application framework designed for web development to produce dynamic web pages developed by Microsoft.
Bootstrap	Open-source CSS framework directed at responsive, mobile-first front-end web development.
HTML	Hypertext markup language, standard for documents designed to be displayed in a web browser.
MSSQL	Microsoft SQL Server is a relational database management system developed by Microsoft.
Frontend	Component of a web application composed of a graphical interface that interacts with the user.
Backend	Component of a web application that is responsible for manipulating and storing data. Usually doesn't interact with the user.

Appendix B: Reports

1. Daily Production Report

Mapa diário de Produção - 06/09/2019				
Turno	1	2	3	Total
Equipa	A	B	C	
Misturador 0	143	182	170	495
Misturador 1	177	195	146	518
Misturador 2	106	0	100	206
Misturador 3	140	128	118	386
Misturador 4	114	154	151	419
Misturador 5	144	125	151	420
Misturador 6	41	105	114	260
Misturador 7	112	147	57	316
Misturador 8	72	123	117	312
Misturador 10	75	107	141	323
Misturador 11	155	163	134	452
Strainer 1	44	50	45	139
Strainer 2	60	80	75	215
CANARIAS	1398	1880	1688	6966
Pisos ED0	0	0	0	0 pn
Pisos ED3	1678	4587	4501	10866 pn
Pisos ED4	2725	5248	4284	12257 pn
Pisos ED5	2725	0	1820	6325 pn
Pisos ED6	4403	3855	4843	13061 pn
Pisos ED8	4470	4380	4300	13150 pn
PISCOS	18998	18867	19699	67564 pn
Fraxina ED1	11540	12260	11700	35500 pn
Fraxina ED2	7150	8352	7400	22902 pn
Fraxina ED3	11300	11500	12000	26800 pn
FRAXINES	30950	32465	27000	90415 pn
Borracha refinada (kg)	1500	1442	0	2942
IL 1	5475	11127	10278	26880 pn
IL 2	12875	14310	12955	34140 pn
IL 3	0	0	6669	6669 pn
CANARIAS	32348	26437	29848	88633 pn
CT1	6500	12300	10000	28800 pn
CT2	14000	3300	6700	24000 pn
CT3	19000	23000	17500	59500 pn
NUCLEOS TALCO	38900	37900	32000	108800 pn
Apex 1	0	0	0	0 pn
Apex 2	2000	3600	2800	8400 pn
Apex 3	2700	4800	2600	10100 pn
Apex 4	2000	3300	2100	7400 pn
Apex 5	2700	3100	2300	8100 pn
Apex 6	3000	2800	2500	8300 pn
Apex 7	2800	2900	2800	8500 pn
Apex 8	2400	2700	2200	7300 pn
Apex 9	2700	2500	2800	8000 pn
Apex 10	2800	2800	2800	8400 pn
Apex 11	3300	3100	3200	9600 pn
Apex 12	1800	2200	2800	6800 pn
Apex 13	0	1900	0	1900 pn
Apex 14	0	1900	0	1900 pn
TALCO COM GUMER	32900	37800	29900	100600 pn
T D1	0	0	0	0 pn
T D2	8427	8703	8242	25372 pn
T D3	5851	5044	5754	16649 pn
T D4	7172	3972	5972	12000 pn
TELA TEXTIL	21463	21719	19108	62290 pn
CAP-STRIP D1	13700	13000	12700	39400 pn
CAP-STRIP D2	13300	16300	16000	45600 pn
CAP-STRIP D3	26400	16700	22100	65200 pn
CAP-STRIP D4	21000	19600	20800	61400 pn
Mistura	11900	13170	10071	35141 pn
ESPIRAL	78300	77370	83133	238803 pn
F B1	8443	14255	17245	39943 pn
F B2	17000	17700	13100	47800 pn
F B3	25400	17535	20101	63036 pn
F B4	25400	0	0	25400 pn
F B5	18250	13754	10753	42757 pn
TELA METALURGA	71750	55713	61317	188780 pn
IN PURSES	3850	5400	3800	13050 pn
IN TUBOS COM CIL	2400	2800	2800	8000 pn
IN TUBOS SEM CIL	2650	3600	3150	9400 pn
IN TUBOS METALURGA	2100	2100	2100	6300 pn
IN TELA	0	4500	0	4500 pn
IN CANOAS	4500	10400	4500	19400 pn
BUFFERS	20	20	20	60 pn
BUFFERS 2000	60	300	110	470 pn
STOCK FINITE	1	1	1	3 pn
STOCK COMPOSTOS	138	178	138	454 pn
STOCK COMPOSTOS FINITE	110	110	110	330 pn
Modulo 01	20	137	3	160
Modulo 02	495	558	0	1053
Modulo 03	0	0	0	0
Modulo 04	481	139	383	1003
Modulo 05	240	272	27	539
Modulo 06	454	555	454	1463
Modulo 07	0	0	0	0
Modulo 08	363	461	449	1273
Modulo 09	618	407	503	1528
Modulo 10	515	459	563	1537
Modulo 11	491	608	505	1604
Modulo 12	411	455	328	1194
Modulo 13	431	431	431	1293
Modulo 14	637	398	458	1493
Modulo 15	278	378	479	1135
Modulo 16	403	444	372	1219
Modulo 17	425	363	412	1199
Modulo 18	338	440	482	1260
Modulo 19	214	271	482	967
Modulo 20	493	439	518	1450
Modulo 21	421	459	431	1311
Modulo 22	347	425	415	1187
Modulo 23	371	353	373	1097
Modulo 24	385	448	475	1308
Modulo 25	508	440	511	1459
Modulo 26	385	343	288	1016
Modulo 27	381	517	343	1241
Modulo 28	295	331	252	878
Modulo 29	382	308	398	1088
Modulo 30	307	270	307	884
Modulo 31	382	115	383	880
Modulo 32	325	217	301	843
Modulo 33	338	315	328	981
Modulo 34	289	330	238	857
Modulo 35	331	173	228	732
Modulo 36	337	474	347	1158
Modulo 37	302	261	282	845
Modulo 38	474	300	369	1143
Modulo 39	0	260	42	302
Modulo 40	328	348	338	1014
Modulo 41	74	334	302	710
Modulo 42	251	54	345	650
Modulo 43	408	330	318	1056
Modulo 44	334	299	307	940
Modulo 45	152	233	301	686
Modulo 46	315	297	301	913
Modulo 47	328	298	188	814
Modulo 48	367	229	32	628
Modulo 49	243	288	188	719
Modulo 50	403	428	273	1104
CONSTRUTORES	16623	16961	16924	50508
Fitura 1	1920	3013	3030	8100
Fitura 2	4664	4300	4201	13165
Fitura 3	819	871	78	1768
Fitura 4	2554	1408	1200	5162
Fitura 5	125	1015	1419	2559
PINTADOS	10705	11052	10315	32072
% Pintados	#Name2	#Name2	#Name2	#Name2
VULCANIZADOS	17341	17658	17274	52273
% MAC	170	138	138	446
% CIL	1100	0	0	1100
MU11	520	515	544	1579
MU12	522	520	484	1526
MU13	1065	1044	1040	3149
MU14	565	754	729	2048
MU17	1001	1065	973	3039
MU18	517	655	736	1908
MU19	877	760	752	2389
MU20	115	570	584	1269
MU21	518	687	775	1980
MU22	905	634	620	2159
MU24	503	512	505	1520
MAC MUS LUP	8113	8255	8103	24471
TOTAL MUS	18540	18240	18610	55390
Inspecionados	15234	17187	18003	50424
MAC	34	182	214	530
Perus Respostas	197	50	72	319
Respostas	8843	8788	8750	26381
Peru Contribuidor	5000	5000	5000	15000

23 de novembro de 2019

2. Weekend Production Report

Mapa de Produção de fim-de-semana							07/09/2019		08-09-2019				
Turno	5	6	7	8	9	Total	Turno	5	6	7	8	9	Total
Misturador 0	230	222	199	194	165	910	Misturador 01	389	407	399	450	359	2112
Misturador 1	230	230	174	207	181	922	Misturador 02	510	443	395	799	350	2507
Misturador 2	0	18	0	0	18	36	Misturador 03	88	0	0	43	364	581
Misturador 3	162	172	125	168	138	665	Misturador 04	983	963	389	471	361	2427
Misturador 4	209	229	127	200	159	924	Misturador 05	156	116	88	411	298	1468
Misturador 5	184	232	164	144	172	896	Misturador 06	581	288	437	690	450	2636
Misturador 6	127	129	104	129	81	569	Misturador 07	0	0	0	0	0	0
Misturador 7	133	161	91	128	86	599	Misturador 08	700	600	612	468	2938	
Misturador 8	157	173	70	107	117	624	Misturador 09	985	531	535	589	481	2882
Misturador 10	124	100	88	106	87	505	Misturador 10	584	492	500	630	484	2690
Misturador 11	127	130	118	139	140	654	Misturador 11	573	690	391	689	118	2481
Serrador 1	80	83	72	84	73	392	Misturador 12	608	610	432	589	274	2483
Serrador 2	101	113	73	96	86	465	Misturador 13	884	730	520	700	494	3127
CANADADA	9359	2097	1361	1767	1477	5682	Misturador 14	895	896	477	679	515	3191
Placas E03	0	0	0	0	0	0	Misturador 15	595	740	540	54	528	2469
Placas E03	3940	8005	4718	3280	2700	19663	Misturador 16	987	431	449	490	434	2391
Placas E04	6485	6100	4826	3400	2927	23737	Misturador 17	528	528	402	489	354	2300
Placas E05	4275	4000	2100	5895	3075	19255	Misturador 18	987	431	449	490	434	2391
Placas E06	6310	5435	4250	6175	4575	26745	Misturador 19	891	512	495	489	424	2828
Placas E08	6020	4965	1600	5240	4475	21420	Misturador 20	648	720	411	490	482	2889
PIBORG	25930	26520	17490	24000	17752	110792	Misturador 21	650	532	400	580	273	2635
Paredes E01	84500	17317	11392	16430	10930	52237	Misturador 22	540	589	599	550	540	2918
Paredes E02	7025	2868	7290	3830	3311	18034	Misturador 23	982	539	446	589	421	2615
Paredes E03	26300	31454	14506	13100	12100	85102	Misturador 24	494	490	485	497	428	2497
Paredes E04	41785	39524	33100	33410	27333	125399	Misturador 25	474	597	384	642	384	2430
Somacho refinado (kg)	0	0	550	0	0	550	Misturador 26	416	528	368	429	339	2094
B. 1	11823	1040	0	3954	3042	26819	Misturador 27	315	284	282	481	271	1844
B. 2	19447	20177	15731	17723	13761	73839	Misturador 28	481	461	379	384	365	1870
B. 3	1330	16158	18874	5700	0	28801	Misturador 29	570	476	281	352	53	1740
CANADADA	82702	27378	26902	33916	22888	127777	Misturador 30	481	461	379	384	365	1870
CA	13156	16000	10900	11800	8000	59856	Misturador 31	481	461	379	384	365	1870
CA1	19600	13400	12400	19000	12700	67600	Misturador 32	481	461	379	384	365	1870
CA2	19600	13400	12400	19000	12700	67600	Misturador 33	481	461	379	384	365	1870
CA3	19600	13400	12400	19000	12700	67600	Misturador 34	481	461	379	384	365	1870
CA4	19600	13400	12400	19000	12700	67600	Misturador 35	481	461	379	384	365	1870
CA5	19600	13400	12400	19000	12700	67600	Misturador 36	481	461	379	384	365	1870
CA6	19600	13400	12400	19000	12700	67600	Misturador 37	481	461	379	384	365	1870
CA7	19600	13400	12400	19000	12700	67600	Misturador 38	481	461	379	384	365	1870
CA8	19600	13400	12400	19000	12700	67600	Misturador 39	481	461	379	384	365	1870
CA9	19600	13400	12400	19000	12700	67600	Misturador 40	481	461	379	384	365	1870
CA10	19600	13400	12400	19000	12700	67600	Misturador 41	481	461	379	384	365	1870
CA11	19600	13400	12400	19000	12700	67600	Misturador 42	481	461	379	384	365	1870
CA12	19600	13400	12400	19000	12700	67600	Misturador 43	481	461	379	384	365	1870
CA13	19600	13400	12400	19000	12700	67600	Misturador 44	481	461	379	384	365	1870
CA14	19600	13400	12400	19000	12700	67600	Misturador 45	481	461	379	384	365	1870
TALDES COM OLIVIA	64360	61280	41780	48450	42180	119436	Misturador 46	410	10	204	11	374	1108
T.01	0	0	0	0	0	0	Misturador 47	504	679	393	421	294	2199
T.02	13614	12868	8301	12745	8483	46701	Misturador 48	385	426	290	389	268	1734
T.03	7833	10608	8826	10611	8531	36209	Misturador 49	590	118	441	420	391	2071
T.04	11021	12088	9268	11234	8831	36286	Misturador 50	480	408	14	199	523	1349
TELA TEXTIL	32828	36889	18474	34616	24829	116716	Misturador 51	853	281	505	489	41	1880
CAP-STRIP-01	14203	17000	11870	11500	13685	28058	CONSTRUIDOR	22621	24026	17977	22698	18881	109486
CAP-STRIP-02	10117	23000	18529	17000	12400	25046	Printura 1	2630	4985	3184	2612	3144	16624
CAP-STRIP-03	19488	30000	16232	11300	6634	24333	Printura 2	5430	5801	4010	5432	3763	24490
CAP-STRIP-04	16799	26000	3080	4000	16753	24332	Printura 3	2502	198	283	36	188	3278
CAP-STRIP-05	10123	12422	6236	14000	8953	19814	Printura 4	4004	3721	2655	4788	2991	18158
MISGRIFF	71319	108620	60044	78800	54454	107949	Printura 5	300	292	430	460	392	1874
F.01	22422	29029	17898	23977	22092	31058	PRINTAGENS	14010	10788	14273	11841	10411	64341
F.02	11289	11448	13896	16216	8811	28661	% Printados	62.52	66.20	61.43	62.99	62.24	60.90
F.03	24408	26773	18091	23446	16435	38423	VULCANIZADORES	22860	23416	17310	21903	17239	108247
F.04	2874	8831	7318	8388	6248	7181	SCRAP	227	157	159	171	146	835
F.05	20701	19185	13570	14241	10464	28679	% Scrap	0.96	0.98	0.91	0.70	0.83	0.81
TELA METALICA	91684	89704	70676	82324	60540	109144	MJ811	621	403	598	504	586	2866
MV PRG0	39000	39000	39000	39000	39180		MJ812	1488	1654	490	540	830	2624
MV TALDES COM OLIVIA	39000	39000	39000	39000	36780		MJ813	1597	1256	981	1000	834	6610
MV NCL00	97000	101520	115000	107800	104650		MJ814	1081	1104	741	504	824	4634
MV PAR O RW OR	20819	20297	21884	21961	22911		MJ817	1437	1438	1022	1253	1014	6304
MV RW O RETOCC (K)	31	124	86	364	763		MJ818	1028	863	788	517	786	4423
MV TELA	42883	40288	42888	39800	41114		MJ819	1013	958	764	504	793	4481
MV MREVEN	88888	90104	91887	88800	91961		MJ820	804	802	698	601	690	3927
MV CANADA	42923	41516	42926	43100	42677		MJ821	624	1126	633	823	794	4362
BUFFER TAP	84	80	21	22	11		MJ822	1137	1133	864	921	844	6347
BUFFER RW07	954	1470	973	1250	1085		MJ824	690	696	474	652	420	2832
STOCK GRATES	88	88	77	86			MAQ. MUE. MRE	18766	18980	2672	9015	8061	66918
STOCK COMPOSTO	1765	1765	1813	1813	1813		TOTAL MJ5	25814	25880	18880	22889	18688	111818
PROD COMPOSTO RW03	1102	1102	1174	1174	1174		Inspecionados	24877	24862	18038	21830	18261	106778
ACR %	88	88	84	84	84		ECC	218	251	181	280	262	1300
ACR 5%	55	55	48	48	48		Pneus Rasoados	280	280	232	50	128	980
							Balancamento	8146	7589	5888	7033	5591	34347
							ENTRADA	22759	22879	18617	18688	18329	107068

26 de novembro de 2019

Prez. Comissao: 105648 Prez. Utilizadores: 105647

3. Daily Report Summary

Relatório Diário: 06-09-2019

Produções de: 05/09/2019			Scrap de: 05/09/2019			UHPs, SUVs e Winter Programados para: 06/09/2019		
Plano 100%	56 616			Meta	Desvio	100%	56 689	
Pneus Vulcanizados	51 646	91,22 %	Scrap I (%)	0,09	0,10	-0,01	Moldes	461
Pneus Construídos	52 370	92,50 %	Scrap II (%)	0,59	0,56	0,03	Mix	192
UHP's vulcanizados	23 833	91,01 %	Scrap III (%)	0,13	0,14	-0,01	SUV's Plan	2 424
Winter Vulcanizados	2 249	11,43 %	Scrap IV (%)	0,96	0,70	0,26	Nr. moldes SUV's	18
SUV's vulcanizados	17 772	744,22 %	Scrap I-IV (%)	1,77	1,50	0,27	Nr. Moldes "U"	1
Entregues	50 100	95,98 %	Scrap IV (pn)	467	365	102	Winter Plan	20 176
Vendidos	69 742		Buffer		140	UHP's Plan	25 234	
Yield	98,58		Pneus Armazém		1 532 553	Nr. moldes UHP's	215	
Rework	791		Desvio Mensal		-2 621	Mix UHP's	104	
% Rework	1,53		Desvio Anual		164 895			
Size Achievement	95,00							

Stock WIP de: 06/09/2019					
Stock Compostos	1 786	Stock Paredes	29 500	Stock Camada	46 500
Stock Compostos Finais	1 118	Stock Talões	90 500	Stock Pneus em verde	21 839
Stock ACS (%)	70,00	Stock Cunhas	33 000	Stock Tela	50 000
Stock ACS 2(%)	55,00	Stock Breaker	92 900	Stock Paletes Insp Final	69
Stock Pisos	37 500				

Comentários:

4. Weekend Daily Report Summary

Relatório Diário: 09-09-2019

Produções de: 07/09/2019 e 08/09/2019		Scrap de: 07/09/2019 a 08/09/2019			UHPs, SUVs e Winter Programados para: 09/09/2019	
Plano 100%	113 393	Avg	Meta	Desvio	100%	56 632
Pneus Vulcanizados	103 547 91,32 %	Scrap I (%)	0,06	0,10 -0,04	Moldes	463
Pneus Construídos	105 646 93,17 %	Scrap II (%)	0,55	0,56 -0,01	Mix	192
UHP's vulcanizados	44 693 92,37 %	Scrap III (%)	0,12	0,14 -0,02	SUV's Plan	5 857
Winter Vulcanizados	41 873 96,54 %	Scrap IV (%)	0,88	0,70 0,18	Nr. moldes SUV's	34
SUV's vulcanizados	5 806 88,41 %	Scrap I-IV (%)	1,60	1,50 0,10	Nr. Moldes "U"	2
Entregues	102 785 90,64 %	Total	Meta	Desvio	Winter Plan	21 366
Vendidos	30 146	Scrap IV (pn)	839	731 108	UHP's Plan	24 324
Yield	98,44	Buffer	48		Nr. moldes UHP's	212
Rework	1 866	Pneus em Armazém	1 601 434		Mix UHP's	107
% Rework	1,81	Desvio Mensal	-6 367			
Size Achievement	95,00	Desvio Anual	161 149			

Programados de: 07/09/2019 e 08/09/2019			
UHP's Planeados	48 383	Winter Planeados	43 373
Mix Greens	194	Mix UHP's	103
Moldes SUV	28	Moldes SUV U	1
SUV's Planeados	6 567	Moldes UHP	208

Stock WIP de: 09/09/2019					
Stock Compostos	1 811	Stock Paredes	36 700	Stock Camada	46 162
Stock Compostos Finais	1 197	Stock Talões	104 650	Stock Pneus em verde	22 590
Stock ACS (%)	66,00	Stock Cunhas	55 000	Stock Tela	44 077
Stock ACS 2(%)	57,00	Stock Breaker	86 519	Stock Paletes insp Final	66
Stock Pisos	39 100				

5. Shift Report

Relatório de turno - Scheduling

Colaborador	6070 Pedro Lemos	
Data	11/09/2019	
Turno	2 C	Obj. Vulcanização: (94% OEE) 52870
Mfx	186	Average Stock 10,28
Standard Deviation	6,05	Estimated Losses 312
Standard Deviation Pisos	0	Average Stock Pisos 0
Relatório Vulcanização	Vulcanizados: 18057	
Med não programada	NP - 0,0 PA - 14/24	
Mud Previstas	0	Medidas não planejadas 0
Mud jante previstas	0	Mud jante realizadas 0
Relatório Preparação		
Relatório Construção	Stock: 21483 + 346	
Relatório SFI	Nada a reportar	

25 de novembro de 2019

Anexo 2- Relatórios gerados

Mapa Diário Produção 02/06/2020

Turno/Equipa	1/A	2/B	3/C	Total
Misturador 0	166	150	144	460
Misturador 1	139	163	171	473
Misturador 2	0	0	0	0
Misturador 3	0	0	0	0
Misturador 4	127	153	128	408
Misturador 5	163	148	119	430
Misturador 6	105	99	128	332
Misturador 7	0	0	0	0
Misturador 9	50	0	26	76
Misturador 10	103	105	114	322
Misturador 11	113	115	134	362
Misturador 12	75	50	117	242
Strainer 1	55	50	50	155
Strainer 2	68	75	65	208
CARGAS	1 164	1 108	1 196	3 468
Pisos E02	0	0	0	0
Pisos E03	4 900	4 390	4 300	13 590
Pisos E04	0	0	0	0
Pisos E05	0	2 050	3 700	5 750
Pisos E06	3 850	720	0	4 570
Pisos E08	0	3 565	2 000	5 565
PISOS	8 750	10 725	10 000	29 475
Paredes E01	0	0	12 000	8 889
Paredes E02	0	0	0	0
Paredes E07	16 500	12 560	10 000	28 933
PAREDES	16 500	12 560	22 000	37 822
IL 1	1 392	0	0	1 160
IL 2	14 613	13 646	12 300	33 799
IL 3	0	0	0	0
CAMADAS	16 005	13 646	12 300	34 959
CT1	6 750	7 300	5 200	9 625
CT2	0	0	6 300	3 150
CT3	20 400	13 200	0	16 800
NÚCLEOS TALÕES	27 150	20 500	11 500	29 575
Apex 1	0	0	0	0
Apex 10	0	2 700	3 000	2 850
Apex 11	2 600	1 900	0	2 250
Apex 12	2 900	2 700	1 800	3 700
Apex 13	3 200	3 150	3 200	4 775
Apex 14	1 900	2 500	2 600	3 500
Apex 2	0	0	0	0
Apex 3	0	2 700	0	1 350
Apex 4	0	0	3 200	1 600
Apex 5	2 500	2 000	3 000	3 750
Apex 6	0	0	0	0
Apex 7	3 000	1 500	0	2 250
Apex 8	2 400	2 150	3 000	3 775
Apex 9	3 000	0	1 500	2 250
TALÕES COM CUNHA	21 500	21 300	21 300	32 050
T 01	0	0	0	0
T 02	7 015	7 940	9 329	19 427
T 03	0	0	0	0
T 04	6 127	8 426	5 179	15 786
TELA TÊXTIL	13 142	16 366	14 508	35 213
F 01	11 304	4 566	12 275	7 587
F 02	4 145	3 558	0	2 077
F 03	16 865	12 292	11 228	10 887
F 04	4 871	3 712	534	2 458
F 05	7 807	13 312	3 134	6 538
TELA METÁLICA	44 992	37 440	27 171	29 547
ACS (%)	73	73	73	
ACS 2 (%)	90	90	90	
BUFFER ROBOT	0	0	0	0
BUFFER TAP	0	44	24	
INV TELA	37 402	36 012	38 600	
INV PISOS	38 000	37 000	38 000	
INV CAMADA	28 406	30 738	29 900	
INV NÚCLEOS	89 000	96 000	91 000	
INV PAREDES	34 000	33 000	32 000	
INV BREAKER	87 970	89 466	92 500	
INV PNEUS EM CRÚ	13 767	13 217	12 301	
INV PNEUS RETIDOS CRÚ	0	180	180	
INV TALÕES COM CUNHA	40 000	40 000	40 000	
STOCK PALETES	85	110	138	
STOCK COMPOSTOS	1 963	1 963	1 963	
STOCK COMPOSTOS FINAIS	1 074	1 074	1 074	

Turno/Equipa	1/A	2/B	3/C	Total
Módulo 01	300	365	277	942
Módulo 02	170	16	117	303
Módulo 03	0	0	0	0
Módulo 04	350	450	383	1 183
Módulo 05	264	346	206	816
Módulo 06	436	492	480	1 408
Módulo 07	0	0	0	0
Módulo 08	0	0	0	0
Módulo 09	0	0	0	0
Módulo 10	465	245	496	1 206
Módulo 11	0	0	0	0
Módulo 12	0	0	0	0
Módulo 13	0	0	0	0
Módulo 14	0	0	0	0
Módulo 15	0	0	0	0
Módulo 16	0	0	0	0
Módulo 17	0	332	195	527
Módulo 18	374	150	460	984
Módulo 19	0	0	0	0
Módulo 20	0	0	0	0
Módulo 21	258	343	364	965
Módulo 22	435	246	380	1 061
Módulo 23	450	404	401	1 255
Módulo 24	388	361	374	1 123
Módulo 25	440	377	430	1 247
Módulo 26	313	300	321	934
Módulo 27	334	306	309	949
Módulo 28	375	6	315	696
Módulo 29	0	0	0	0
Módulo 30	300	257	342	899
Módulo 31	352	0	345	697
Módulo 32	435	468	440	1 343
Módulo 33	335	237	210	782
Módulo 34	240	302	294	836
Módulo 35	0	0	0	0
Módulo 36	106	430	507	1 043
Módulo 37	0	0	0	0
Módulo 38	0	0	0	0
Módulo 39	247	315	227	789
Módulo 40	45	300	360	705
Módulo 41	410	445	285	1 140
Módulo 42	357	50	106	513
Módulo 43	0	0	0	0
Módulo 44	0	0	0	0
Módulo 45	231	253	264	748
Módulo 46	12	3	15	30
Módulo 47	363	285	167	815
Módulo 48	307	381	284	972
Módulo 49	390	331	284	1 005
Módulo 50	229	508	442	1 179
Módulo 51	33	0	0	33
CONSTRUIDOS	9 744	9 304	10 080	29 128
Pintura 1	1 250	1 440	1 270	3 960
Pintura 2	2 195	2 045	2 430	6 670
Pintura 3	0	40	0	40
Pintura 4	2 319	2 600	2 235	7 154
Pintura 5	120	36	5	161
PINTADOS	5 884	6 161	5 940	17 985
%Pintados	60.39%	66.22%	58.93%	61.74%
VULCANIZADOS	10 096	10 220	10 070	30 386
SCRAP	62	155	81	298
%Scrap	0.61%	1.52%	0.80%	0.98%
MU#11	284	310	577	1 171
MU#12	602	367	323	1 292
MU#13	409	170	202	781
MU#14	0	0	0	0
MU#17	882	736	835	2 453
MU#18	697	736	687	2 120
MU#19	0	0	6	6
MU#20	141	456	456	1 053
MU#21	752	756	729	2 237
MU#22	456	506	627	1 589
MU#24	358	321	320	999
MAQ. MUS. UHP	4 581	4 358	4 762	13 701
Balanceamento	4 400	4 932	4 932	14 264
BCD	134	95	96	325
Inspecionados	10 094	10 150	10 746	30 990
Pneus Raspados	5	77	60	142
TOTAL_MUS	11 455	11 602	12 185	35 242
ENTREGUES	10 102	10 712	11 373	32 187

Monday, July 27, 2020

Prev Construídos 29128 Prev Vulcanizados 30386

Mapa Diario Produção 08/02/2020 a 09/02/2020

Turno/Equipa	5/E	6/D	7/E	8/D	9/E	Total
Misturador 0	246	226	185	227	171	1 055
Misturador 1	226	245	162	224	180	1 037
Misturador 2	222	220	155	206	140	943
Misturador 3	223	97	146	196	137	799
Misturador 4	221	235	150	183	143	932
Misturador 5	119	204	160	193	123	799
Misturador 6	112	116	116	130	100	574
Misturador 7	141	138	113	143	124	659
Misturador 9	156	179	117	144	118	714
Misturador 10	141	136	93	210	98	678
Misturador 11	193	192	160	188	118	851
Misturador 12	141	134	105	177	72	629
Strainer 1	100	95	55	87	70	407
Strainer 2	70	112	85	100	85	452
CARGAS	2 311	2 329	1 802	2 408	1 679	10 529
Pisos E02	0	0	0	0	0	0
Pisos E03	6 225	6 450	4 735	6 375	4 000	27 785
Pisos E04	6 325	7 100	4 880	5 900	4 155	28 360
Pisos E05	5 550	6 250	4 075	5 350	4 050	25 275
Pisos E06	6 000	5 650	4 595	5 670	4 310	26 225
Pisos E08	5 550	5 450	4 000	4 215	3 940	23 155
PISOS	29 650	30 900	22 285	27 510	20 455	130 800
Paredes E01	16 770	17 500	12 020	15 000	11 720	54 081
Paredes E02	8 300	10 250	6 600	10 750	8 150	32 630
Paredes E07	19 000	16 000	10 500	13 400	13 000	53 259
PAREDES	44 070	43 750	29 120	39 150	32 870	139 970
IL 1	1 510	3 000	10 613	13 600	10 496	32 683
IL 2	22 078	21 500	14 722	17 966	13 580	74 872
IL 3	14 683	13 390	7 133	6 907	3 628	38 118
CAMADAS	38 271	37 890	32 468	38 473	27 704	145 672
CT1	15 700	16 500	10 800	16 000	10 600	34 800
CT2	17 000	15 500	16 000	26 000	13 000	43 750
CT3	24 820	37 000	23 315	32 500	17 710	67 673
NÚCLEOS TALÕES	57 520	69 000	50 115	74 500	41 310	146 223
Apex 1	4 000	4 600	3 700	4 300	4 000	10 300
Apex 10	3 900	3 700	2 600	2 800	2 000	7 500
Apex 11	5 100	5 700	3 500	4 600	3 250	11 075
Apex 12	3 100	3 900	3 000	3 000	2 500	7 750
Apex 13	3 300	4 600	3 500	4 000	3 300	9 350
Apex 14	3 700	4 100	2 400	3 900	3 200	8 650
Apex 2	5 300	5 100	3 400	4 300	2 600	10 350
Apex 3	5 300	5 000	3 800	4 700	2 700	10 750
Apex 4	4 550	5 500	3 500	4 500	3 200	10 625
Apex 5	4 700	4 000	2 900	4 000	3 200	9 400
Apex 6	3 450	3 800	2 800	4 700	2 750	8 750
Apex 7	4 000	4 000	2 700	4 100	3 000	8 900
Apex 8	4 100	3 800	3 000	4 000	2 900	8 900
Apex 9	4 200	4 200	2 700	3 600	3 000	8 850
TALÕES COM CUNHA	58 700	62 000	43 500	56 500	41 600	131 150
T 01	0	0	0	0	0	0
T 02	12 978	12 400	7 724	10 750	8 038	41 512
T 03	13 802	12 450	9 179	11 400	9 216	44 838
T 04	14 555	14 000	8 330	12 300	9 989	47 339
TELA TÊXTIL	41 335	38 850	25 233	34 450	27 243	133 689
F 01	22 550	24 640	21 115	18 245	19 673	28 635
F 02	24 482	26 588	18 311	19 040	13 776	27 550
F 03	22 439	22 345	17 105	22 421	16 647	27 216
F 04	6 957	11 769	6 345	12 064	6 491	11 761
F 05	22 548	19 344	16 875	19 196	14 698	24 979
TELA METÁLICA	98 976	104 686	79 751	90 966	71 285	120 141
ACS (%)	48	48	47	47	47	47
ACS 2 (%)	60	60	65	65	65	65
BUFFER ROBOT	1 789	1 869	1 879	1 900	1 712	7 149
BUFFER TAP	0	20	0	40	27	87
INV BREAKER	118 253	120 901	116 785	114 417	111 741	582 107
INV TELA	62 663	66 625	66 569	64 765	63 597	324 619
INV PISOS	31 500	35 000	36 000	36 000	37 000	165 500
INV PAREDES	35 500	38 000	37 000	36 000	34 500	171 000
INV CAMADA	43 750	49 138	41 660	44 460	45 971	225 979
INV NÚCLEOS	100 700	91 000	104 500	110 000	127 100	533 300
INV PNEUS EM CRÚ	22 758	21 991	23 124	23 910	24 451	116 234
INV PNEUS RETIDOS CRÚ	0	0	0	0	0	0
INV TALÕES COM CUNHA	50 000	52 000	54 000	54 000	55 000	265 000
STOCK PALETES	47	41	47	45	50	190
STOCK COMPOSTOS	1 611	1 611	1 618	1 618	1 618	6 576
STOCK COMPOSTOS FINAIS	908	908	888	888	888	3 680

Tuesday, June 30, 2020

Prev Construidos 120609 Prev Vulcanizados 118644

Turno/Equipa	5/E	6/D	7/E	8/D	9/E	Total
Módulo 01	567	518	304	465	365	2 219
Módulo 02	741	714	458	709	597	3 219
Módulo 03	668	537	445	484	455	2 579
Módulo 04	415	693	475	589	378	2 550
Módulo 05	508	626	484	489	449	2 556
Módulo 06	612	834	700	762	573	3 481
Módulo 07	658	654	432	627	437	2 808
Módulo 08	723	640	490	519	500	2 872
Módulo 09	565	636	418	505	352	2 476
Módulo 10	510	628	577	549	521	2 785
Módulo 11	635	510	425	577	430	2 577
Módulo 12	580	596	414	508	381	2 479
Módulo 13	401	758	603	721	499	2 982
Módulo 14	730	706	518	703	480	3 137
Módulo 15	690	745	521	549	481	2 986
Módulo 16	524	648	466	540	466	2 644
Módulo 17	480	632	353	565	400	2 430
Módulo 18	509	581	535	507	395	2 527
Módulo 19	708	476	426	442	462	2 514
Módulo 20	710	659	433	644	540	2 986
Módulo 21	548	650	468	503	376	2 545
Módulo 22	480	583	439	556	438	2 496
Módulo 23	582	630	412	605	431	2 660
Módulo 24	525	478	330	600	370	2 303
Módulo 25	583	683	405	431	448	2 550
Módulo 26	463	295	271	356	319	1 704
Módulo 27	462	464	316	435	334	2 011
Módulo 28	493	436	350	440	350	2 069
Módulo 29	446	523	349	465	315	2 098
Módulo 30	447	333	411	455	356	2 002
Módulo 31	642	559	430	410	386	2 427
Módulo 32	641	660	470	610	487	2 868
Módulo 33	412	593	341	510	349	2 205
Módulo 34	420	370	338	342	340	1 810
Módulo 35	426	438	330	471	347	2 012
Módulo 36	472	600	377	472	357	2 278
Módulo 37	487	523	368	456	404	2 238
Módulo 38	554	490	367	402	397	2 210
Módulo 39	365	423	341	325	336	1 790
Módulo 40	424	443	346	423	319	1 955
Módulo 41	396	533	419	439	474	2 261
Módulo 42	470	422	307	367	300	1 866
Módulo 43	435	567	306	463	271	2 042
Módulo 44	469	439	323	408	282	1 921
Módulo 45	360	460	356	362	320	1 858
Módulo 46	462	590	422	455	379	2 308
Módulo 47	466	472	368	427	258	1 991
Módulo 48	482	454	347	460	377	2 120
Módulo 49	526	614	337	431	352	2 260
Módulo 50	621	730	500	605	488	2 944
Módulo 51	0	0	0	0	0	0
CONSTRUIDOS	26 483	28 246	20 621	25 138	20 121	120 609
Pintura 1	5 130	5 616	3 495	4 916	3 425	22 582
Pintura 2	5 039	5 366	3 914	4 349	3 765	22 433
Pintura 3	0	0	0	0	0	0
Pintura 4	3 274	4 552	3 008	3 805	3 012	17 651
Pintura 5	1 553	560	570	610	1 560	4 853
PINTADOS	14 996	16 094	10 987	13 680	11 762	67 519
%Pintados	56.63%	56.98%	53.28%	54.42%	58.46%	55.98%
VULCANIZADOS	27 250	27 113	19 835	24 597	19 849	118 644
SCRAP	174	123	128	106	115	646
%Scrap	0.64%	0.45%	0.65%	0.43%	0.58%	0.54%
MU#11	936	1 005	851	1 034	860	4 686
MU#12	999	1 122	863	976	740	4 700
MU#13	964	983	957	1 051	911	4 856
MU#14	1 102	1 104	831	1 044	785	4 866
MU#17	1 352	1 307	1 034	1 288	1 023	6 004
MU#18	1 002	881	780	974	764	4 401
MU#19	1 192	1 082	855	792	469	4 390
MU#20	1 112	1 189	910	882	669	4 762
MU#21	1 183	1 129	790	1 051	790	4 943
MU#22	1 225	1 132	805	1 119	895	5 176
MU#24	1 138	1 069	860	1 060	840	4 967
MAQ. MUS. UHP	12 195	12 003	9 536	11 271	8 746	53 751
Balancamento	9 914	9 547	7 245	8 537	6 948	42 191
BCD	120	142	126	165	188	741
Inspecionados	27 589	28 129	20 542	25 773	20 400	122 233
Pneus Raspados	280	235	270	423	262	1 470
TOTAL_MUS	29 446	29 444	22 611	25 668	21 333	128 502
ENTREGUES	26 716	26 538	20 219	24 564	19 959	117 996

Relatório Diário: 02/06/2020

Produções de: 01/06/2020

Scrap de: 01/06/2020

Programados de: 02/06/2020

Plano 100%	32 986					100%	33 044	
Pneus Vulcanizados	29 131	88.31 %	Scrap I(%)	0.08	0.10	-0.02	Moldes	267
Pneus Construidos	29 782	90.29 %	Scrap II(%)	0.72	0.55	0.17	Mix	123
UHP's Vulcanizados	13 892	88.18 %	Scrap III(%)	0.17	0.12	0.05	SUV's Plan	1 930
Winter Vulcanizados	9 546	87.19 %	Scrap IV(%)	1.40	0.69	0.71	Nr.moldes SUV's	17
SUV's Vulcanizados	1 875	98.12 %	Scrap I-IV(%)	2.37	1.45	0.92	Nr.Moldes "U"	9
Entregues	29 501	111.11 %	Scrap IV(pn)	298	186	112.1	Winter Plan	10 745
Vendidos	53 208						UHP's Plan	16 242
Yield	97.97		Buffer				Nr.moldes UHP's	135
Rework	1 564		Pneus Armazém				Mix UHP's	52
%Rework	5.37		Desvio Mensal					
Size Achievement	89.00		Desvio Anual					

Stock WIP de: 02/06/2020

ACS (%)	219
ACS 2 (%)	270
STOCK COMPOSTOS	5 889
STOCK COMPOSTOS FINAIS	3 222
INV TELA	112 014
INV PISOS	113 000
INV CAMADA	89 044
INV NÚCLEOS	276 000
INV PAREDES	99 000
INV BREAKER	269 936
INV TALÕES COM CUNHA	120 000
INV PNEUS EM CRÚ	39 285
INV PNEUS RETIDOS CRÚ	360
BUFFER ROBOT	0
BUFFER TAP	68
STOCK PALETES	333

Workoff de: 02/06/2020

Workoff Total	
Workoff Total Pisos	
Workoff Pisos	
Workoff Pisos DIP	
Workoff Paredes	
Workoff Cunhas	
Workoff Camadas	
Aguarda Disposição	

Comentários:

SHIFT REPORT - SCHEDULING

Colaborador	6070	Pedro Lemos		
Data	06/01/2020			
Turno	1	A	Objetivo Vulcanização: (94% OEE)	57437
Mix	218		Average Stock:	9.50
Standart Deviation	6.59		Estimated Losses:	482
Standart Deviation Pisos	0.00		Average Stock Pisos:	0.00
Relatório Vulcanização:	Vulcanizados: 18515			
Med não programada:	PA - 34/54			
Mud Previstas	0	Medidas não planeadas	0	
Mud jante previstas	1	Mud Jante Realizada	1	
	#45 - 21" 22"			
Relatório de preparação				
Relatório de construção	Stock: 22399 + 308			
Relatório SFI	Nada a reportar.			

7/27/2020

Supervision Daily Meeting

Data:	1/6/2020	Dept	Absentismo	Scheduling	
Turno:	A	1	DP1 2	Objetivo 93%:	57437
Stock Armazem	1498 548		Mix:	218	Perdas Programadas:
Vendas	50 052			482	Mud. jante prevista:
				SA:	0
				Suspensas:	0
					#45 - 21" 22"

Medidas				Módulos	
Prioritárias	Com atraso	Suspensas	Críticas	Módulos Críticos	Jantes Críticas
			655,734,1821,1 941,,1756,1019 ,1286,1482	3,4,5,13,15,17,18, 19,30,34,43,44	14" 15" 20"

Stock	Qt
STOCK COMPOSTOS	1 707
STOCK COMPOSTOS FINAIS	1 225
INV BREAKER	95 974
INV CAMADA	41 000
INV NÚCLEOS	121 000
INV PAREDES	40 000
INV PISOS	40 000
INV TALÕES COM CUNHA	55 000
INV TELA	47 323
ACS 2 (%)	49
ACS (%)	72
INV PNEUS EM CRÚ	23 084
INV PNEUS RETIDOS CRÚ	0
BUFFER ROBOT	2 379
BUFFER TAP	248
STOCK PALETES	93

Avarias				Diversos		UHP	
Maq	Tempo	Tipo	Desc			Moldes	Medidas
BB#07	2	Elet.	Banho (Cont)	BB#00	8h MP	251	109
BB#06	1,3	Mec.	Martelo (cont)	Raspador04	2h Qualidade		
BB#05	0,45	Elet.	Transporte negro	BCD04	Transferência-		
V116	0,2	Elet.	Comunicação.	Mód 49	CGMS		
KUKA2	2	Mec.	Casquilho	Mód 08/35	Transformação.		
Km37	0,45	Elet.	Tabuleiro tela	N29/30; P15/16	8h MP		
Km48	0,25	Mec.	Fixador talão	H01/02	8h transformação		
Pu34	0,3	Elet.	Não aplicava brk	M08	molde a lavar fora		
Pu31	0,25	Mec.	brk enroscado	P20	Molde continua fora.		
Km32	0,25	Elet.	Pente paredes	Gtal04	3h MP		
C09/10	2	Mec.	Fuga água (edifício)	6 máquinas	com validação.		
Ext 07	0,2	Elet.	Drive	Extrusora 08	4h MP		
				Apex 01	3h transformação		
				Fisher 05	3h MP		
				INL02	4h MP		
				CaM4R1	Transformação.		
				F09/10 e G01/02	Datas		

Materiais em falta / Pneus verde

Aspectos da Qualidade

Ensaios/ VO's