

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema de Gestão de Receitas para Indústria de Processo

Paulo Martins Fontes

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Pedro Alexandre Rodrigues João

23 de Julho de 2018

Resumo

Como consequência de uma encomenda realizada por parte de um cliente, a JPM Industry irá instalar uma fábrica ligada à indústria de processo onde os produtos resultantes estarão relacionados com as áreas dos detergentes e cosméticos. Como a produção neste tipo de indústria está ligada à execução de receitas, foi feito um pedido por parte desse mesmo cliente para que o sistema de criação, edição e gestão das receitas fosse desenvolvido pela própria JPM Industry em vez de se utilizar software já desenvolvido por terceiros. Desse modo surgiu esta Dissertação com o objetivo de desenvolver um Sistema de Gestão de Receitas para Indústria de Processo de forma a que fosse satisfeito o pedido realizado pelo cliente. O sistema deve ser modular, parametrizável e evolutivo e é expectável que respeite as boas práticas e referenciais normativos aplicáveis. O sistema deve também ser acedido através de um navegador web permitindo que seja utilizado em vários tipos de dispositivos como computadores, dispositivos móveis ou mesmo tecnologias do tipo 'SCADA', não necessitando assim de qualquer tipo de instalação nesses dispositivos.

Inicialmente é feito um estudo aprofundado sobre as normas existentes que sejam aplicáveis a este tipo de sistema e também um estudo sobre as soluções atualmente disponíveis no mercado. Com o conhecimento das imposições existentes nas normas, com a informação recolhida das funcionalidades disponíveis nos sistemas atualmente no mercado e com os requisitos para o sistema a desenvolver bem estabelecidos é então feito o projeto para a solução do sistema que se pretende desenvolver. Concluída a fase de projeto é feita uma implementação de um protótipo da mesma e também realizado um conjunto de testes com o objetivo de validação a mesma.

Do projeto do sistema resulta uma proposta de uma estrutura base tanto para a forma como a informação relevante ao sistema é armazenada como para a interface mais aconselhada para o mesmo. É ainda feito um conjunto de sugestões quanto a alguns softwares de terceiros existentes que visam facilitar em certo modo a implementação de certas funcionalidades que um sistema deste género necessita. Da implementação do protótipo resulta um sistema que cumpre com a maioria dos requisitos impostos, mas que carece de uma implementação mais profunda do processo de criação das receitas onde falta um ambiente gráfico para desenvolvimento e também carece de um conjunto de testes mais rigorosos e pela mão de quem poderá vir a utilizar o sistema.

Do principal objetivo de ter um Sistema de Gestão de Receitas para Industria de Processo resulta uma estrutura para o sistema capaz de suportar uma implementação real, comprovado pela implementação do protótipo, e resulta também dessa implementação protótipo um sistema que com mais de tempo dedicado ao seu desenvolvimento é capaz de se tornar numa primeira versão funcional do mesmo.

Abstract

As a result of an order placed by a customer, JPM Industry will install a factory linked to the process industry where the resulting products will be related to the areas of detergents and cosmetics. Since the production in this type of industry is linked to the execution of recipes, a request was made by the same client that the system for the creation, editing and management of recipes should be developed by JPM Industry itself instead of using software already developed by other companies. Therefore, this dissertation arose with the objective of developing a Recipe Management System for Process Industry in order to satisfy the request made by the client. The system must be modular, parameterizable and evolutionary and it is expected that it respects the good practices and applicable normative references. The system must also be accessed through a web browser allowing it to be used in various types of devices such as computers, mobile devices or even 'SCADA' technologies, without requiring any type of installation on these devices.

Initially an in-depth study on the existing standards that are applicable to this type of system is made as well as a study on the solutions currently available in the market. With the knowledge of the existing constraints in the standards, with the information gathered from the functionalities available in the systems currently in the market and with the requirements for the system to be developed well established, the project for the solution of the system to be developed is then made. Once the design phase has been completed, a prototype implementation has been implemented and a set of tests has been carried out with the purpose of validating it.

From the system design results a proposal of a base structure for both the way the relevant information to the system is stored and for the most recommended interface for it. A set of suggestions is also made regarding some existing third party software aimed at facilitating in a certain way the implementation of certain functionalities that a system like this needs. The implementation of the prototype results in a system that meets most of the imposed requirements but lacks a deeper implementation of the recipes creation process where there is a lack of a graphical environment for development and also lacks from a more rigorous set of tests and by the hand who will use the system.

From the main objective of having a Recipe Management System for Process Industry results a structure for the system capable of supporting a real implementation, proven by the implementation of the prototype, and also results from this prototype implementation a system that with more time dedicated to its development is capable of becoming a first functional version of it.

Agradecimentos

Nunca teria conseguido alcançar esta etapa da minha vida sozinho e sem o apoio que muitas pessoas demonstraram para comigo e que ao longo do tempo se mostrou crucial para que este objetivo da minha vida fosse cumprido. Assim gostaria de demonstrar a minha eterna gratidão para com elas.

Aos meus pais agradeço por me terem sempre apoiado nas minhas escolhas, por me terem dado sempre todas as condições necessárias para que conseguisse realizar o curso com sucesso e por terem sido sempre um modelo de amor, humildade e persistência para mim.

Ao meu irmão Hélder tenho a agradecer toda a paciência que ao longo da vida tem tido para comigo, por me ter sempre orientado pelas melhores escolhas e por ser para mim um modelo de esforço e dedicação.

À minha namorada Célia agradeço a compreensão por todos aqueles momentos em que não estivemos juntos porque tinha coisas importantes para a faculdade para fazer e agradeço toda a força que me foi dando ao longo do curso sem nunca me deixar desistir dos meus objetivos.

Aos meus amigos da FEUP, Cláudio e Pedro, ficarei para sempre grato por todos os momentos que acompanharam a parte académica do curso e que foram sempre importantes para balancear o trabalho com o lazer e o prazer da companhia deles.

Ao meu amigo Zé um obrigado por todos os momentos de desabafo partilhados no decorrer do curso e que permitiram um alívio da pressão que o curso exerce sobre os alunos.

Às minhas amigas Ana Namorado e Ana Pina agradeço por terem sempre ouvido todos os meus desabafos nos momentos em que as coisas não corriam tão bem quanto o esperado e por terem sempre conseguido devolver-me o ânimo para encarar os problemas de frente.

Ao colaborador da JPM Luís Soares um obrigado por todo o apoio demonstrado para comigo ao longo da Dissertação e por, apesar da sobrecarga de trabalho, arranjar sempre um bocado do seu tempo para atender aos meus pedidos. Ao colaborador da JPM Luís Pires um grande obrigado por acompanhar sempre de perto esta minha luta final e por se ter mostrado sempre disponível para me ajudar no que fosse preciso.

E por fim um grande obrigado ao meu orientador Pedro Alexandre Rodrigues João por toda a preocupação e apoio demonstrados desde o primeiro dia desta minha última aventura na realização do meu curso.

Paulo Martins Fontes

“No one ever made a difference by being like everyone else.”

P.T. Barnum

Conteúdo

1	Introdução	1
1.1	A Empresa	1
1.2	Enquadramento e Descrição do Projeto	2
1.3	Metodologia Seguida no Projeto	4
1.4	Estrutura da Dissertação	4
2	Estado da Arte	5
2.1	Indústria de Processo	5
2.2	Erros de Intervenção Humana	6
2.3	Sistema de Gestão de Receitas	6
2.4	Interação de um Sistema de Gestão de Receitas com Outros Sistemas	7
2.5	Normas	8
2.5.1	ISA88 / IEC 61512	8
2.5.2	EC-GMP Annex 11 e FDA CFR 21 Part 11	14
2.6	Softwares no Mercado	15
2.6.1	Gestão de Lotes	16
2.6.2	Gestão de Receitas	17
2.6.3	Gestão de Materiais, Rastreabilidade e Registo	18
2.6.4	Relatórios	19
2.6.5	Interface	19
2.6.6	Servidor	20
2.6.7	Plataformas Compatíveis	20
2.7	Desenvolvimento Web	21
2.7.1	Lado do Cliente	21
2.7.2	Lado do Servidor	22
2.7.3	Base de Dados	23
2.7.4	Segurança da Rede	23
2.8	Resumo	25
3	Descrição da Solução Proposta	27
3.1	Arquitetura Funcional	27
3.2	Base de Dados	28
3.2.1	Utilizadores	29
3.2.2	Equipamentos	29
3.2.3	Receitas	30
3.2.4	Materiais	31
3.2.5	Relatórios	32
3.3	Interface	32

3.3.1	Login	33
3.3.2	Menu	34
3.3.3	Equipamentos	34
3.3.4	Materiais	35
3.3.5	Receitas	36
3.3.6	Relatórios	37
3.3.7	Utilizadores	37
3.3.8	Ajuda	37
3.4	Software de Terceiros	38
3.4.1	Smarty	38
3.4.2	jQuery	39
3.4.3	W3.CSS e Bootstrap CSS	40
3.4.4	jQuery Bootgrid	40
3.4.5	Bootstrap-select	41
3.4.6	Bootstrap Notify e Bootbox.js	41
3.4.7	Bootstrap Tree View	42
3.4.8	GoJS	42
3.4.9	FPDF	42
3.5	Resumo	43
4	Implementação e Resultados	45
4.1	Softwares Utilizados	45
4.2	Sistema Implementado	46
4.2.1	Login	48
4.2.2	Menu	48
4.2.3	Equipments	49
4.2.4	Materials	51
4.2.5	Recipes	53
4.2.6	Users	57
4.3	Testes e Resultados	58
4.3.1	Correta Inserção de Dados na Base de Dados	58
4.3.2	Introdução Errada de Dados no Sistema	59
4.3.3	Restrições de Ações no Sistema	59
4.3.4	Velocidade do Sistema	60
5	Conclusões e Trabalho Futuro	63
5.1	Conclusões e Satisfação dos Objetivos	63
5.2	Trabalho Futuro	64
A	Linguagem Gráfica PFC	67
A.1	Utilizadores	67
B	Código SQL das Tabelas	71
B.1	Utilizadores	71
B.2	Equipamentos	71
B.3	Materiais	74
B.4	Receitas	75
	Referências	79

Lista de Figuras

2.1	<i>Computer Integrated Manufacturing (CIM)</i>	8
2.2	<i>Physical Model</i> [1]	10
2.3	<i>Procedural Control Model</i> [1]	11
2.4	<i>Process model</i> [1]	12
2.5	Elementos Constituintes de uma Receita [2]	13
2.6	Níveis de Complexidade dos sistemas disponíveis	15
2.7	Comunicação entre Cliente, Servidor e Base de Dados	22
2.8	Comunicação Utilizando PHP	23
3.1	Interface do Sistema	28
3.2	Funcionamento Interno do Sistema	29
3.3	Transferência de informação entre Sistema de Gestão de Receitas e controlador	30
3.4	Principais grupos de dados na Base de Dados	31
3.5	Modelo Relacional das Tabelas dos Utilizadores	32
3.6	Modelo Relacional das Tabelas dos Equipamentos	33
3.7	Modelo Relacional das Tabelas das Receitas	34
3.8	Modelo Relacional das Tabelas dos Materiais	35
3.9	Mockup da Página de Login	36
3.10	Mockup da Página do Menu	37
3.11	Mockup da Página de Equipamentos	38
3.12	Mockup da Página de Materiais	39
3.13	Mockup da Página de Receitas	40
3.14	Mockup da Página do Editor de Receitas	41
3.15	Mockup da Página de Relatórios	42
3.16	Mockup da Página de Utilizadores	43
4.1	Estrutura de Pastas do Sistema Implementado	46
4.2	Janela de Inserção de Dados e Janela de Alertas Implementadas	47
4.3	Página de Login Implementada	48
4.4	Página de Menu Implementada e Menu Lateral Oculto	49
4.5	Página Inicial da Secção Equipments Implementada	50
4.6	Página Inicial da Secção Materials Implementada	52
4.7	Página Inicial da Secção Recipes Implementada	53
4.8	Página com Formulário de adição de Equipment Requirments	55
4.9	Página do Editor de Receitas Implementada	56
4.10	Página Inicial da Secção Users Implementada	58
4.11	Consulta de Dados de uma Tabela Usando o phpPgAdmin	59

4.12	Informação da Secção Performance das Ferramentas de Programador do Google Chrome	60
A.1	Símbolos Representativos de Elementos de uma Receita [2]	67
A.2	Símbolo de Início [2]	67
A.3	Símbolo de Fim [2]	68
A.4	Transição Implícita [2]	68
A.5	Transição Explícita [2]	68
A.6	Início de uma Sequência de Escolhas [2]	69
A.7	Fim de uma Sequência de Escolhas [2]	69
A.8	Início de Várias Escolhas em Simultâneo [2]	70
A.9	Fim de Várias Escolhas em Simultâneo [2]	70

Abreviaturas e Símbolos

AJAX	Asynchronous JavaScript and XML
ANSI	American National Standards Institute
APICS	American Production and Inventory Control Society
BatchML	Batch Markup Language
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CIM	Computer Integrated Manufacturing
CSS	Cascading Style Sheets
DCS	Distributed Control System
ERP	Enterprise Resource Planning
EU	European Union
FDA	Food and Drug Administration
FIFO	First-In, First-Out
GMP	Good Manufacturing Practices
GRAFCET	GRAPHe de Commande Etape-Transition
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
ISA	International Society of Automation
ISO	International Organization for Standardization
JS	JavaScript
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MES	Manufacturing Execution System
PDF	Portable Document Format
PFC	Procedure Function Charts
PHP	Hypertext Preprocessor
RDBMS	Relational Database Management System
SCADA	Supervisory Control and Data Acquisition
SFC	Sequential Function Chart
SHA	Secure Hash Algorithms
SSL	Secure Sockets Layer
URL	Uniform Resource Locator
WIP	Work In Progress
XML	Extensible Markup Language
XSS	Cross-site Scripting

Capítulo 1

Introdução

Neste primeiro capítulo é feita uma pequena apresentação da empresa proponente da Dissertação, é também feita uma breve descrição do projeto, dos seus principais objetivos e metodologia seguida no seu desenvolvimento e é ainda abordada a estrutura da Dissertação.

1.1 A Empresa

O desenvolvimento desta Dissertação será realizado nas instalações da empresa JPM Industry, tanto em Vale de Cambra onde se localiza a sua sede bem como nas instalações no Edifício Central do Parque de Ciência e Tecnologia da Universidade do Porto (UPTEC) onde possui um escritório.

A empresa foi fundada com o nome JPM - Automação e Equipamentos Industriais, S.A. em 1994, em Vale de Cambra, com o objetivo de oferecer serviços de eletrificação e automação para a indústria, especializando-se no mercado da intralogística. Em 2009 foi criado o Grupo JPM onde a empresa integrou, passando a ser conhecida como JPM Industry, e do qual também fazem parte a JPM Renováveis que tem como missão oferecer soluções de energia sustentável e renovável para indústrias e a Jointsteel com a missão de fabricar equipamentos e soluções de processo em aço inoxidável [3].

Atualmente a JPM Industry tem como indústrias alvo aeroportos, correios, armazéns, manufatura e processo. Os aeroportos, correios e armazéns permitem à empresa aplicar todo o conhecimento que dispõe ao nível da intralogística fazendo com que sejam implementadas soluções onde se responde às necessidades e exigências que existem no transporte de cargas e encomendas nestes tipos de indústrias. Já a indústria da manufatura, que converte matérias primas em produto acabado e que está normalmente associada à indústria de componentes eletrónicos, maquinaria, papel, impressão, metais, madeira, materiais de construção, entre outros, permite à JPM Industry que através da sua experiência na gestão e execução de projetos associados a este tipo de indústria implemente soluções eficientes, eficazes e que vão de encontro às especificações do cliente. Por fim a indústria de processo que trata da conversão de fórmulas e receitas em produto acabado e que está associada a indústrias alimentares, cosméticos, moda, farmacêuticas, química, ração

alimentar para animais, entre outros, permite à JPM Industry fazer uso da sua experiência combinada com as tecnologias das empresas do mesmo grupo, a Jointsteel e a JPM Renováveis, de modo a garantir a fiabilidade, produtividade e eficiência das unidades de processo instaladas nos clientes [4].

Na indústria de processo a empresa oferece soluções nas áreas dos transportadores industriais, da robótica, da automação e do software. A nível dos transportadores industriais são responsáveis pelo seu desenho, fabrico e montagem. Já na área da robótica são responsáveis pelo projeto, fabrico e implementação final da solução proposta. Estas soluções vão desde a despaletização/paletização à integração num ecossistema de uma máquina. Na área da automação desenvolvem e implementam soluções de modo a aumentar a produtividade e competitividade dos seus clientes e na área do software implementam soluções standard ou personalizadas a pedido do cliente [5].

Em resposta a um pedido desses a JPM Industry depara-se atualmente com a necessidade de implementar um software personalizado para um cliente ligado à indústria de processo que necessita de fazer a gestão de receitas dos produtos que pretende produzir nas suas instalações que serão elas realizadas também pela JPM Industry.

1.2 Enquadramento e Descrição do Projeto

Devido há existência no mercado de várias empresas a fabricar o mesmo tipo de produto, surgem desafios a nível da produção que obrigam as empresas a ter os seus processos a funcionar da maneira mais eficiente e eficaz possível. Só assim as empresas conseguem produtos que vão de encontro aos padrões de qualidade impostos e ao mesmo tempo ter custos para o produto final que sejam tanto competitivos no mercado como lucrativos para a empresa.

Com estes objetivos em vista, empresas que operam no domínio da indústria de processo tendem a automatizar os seus processos aumentando assim a cadência de produção e diminuindo o número de lotes produzidos com defeito. Dentro desta automatização aparece a necessidade de diminuir o tempo de configuração de todo o processo quando se pretende utilizar os recursos disponíveis na fábrica para produzir produtos diferentes. Assim surgem os sistemas de gestão de receitas que permitem a criação de receitas distintas que são destinadas ao fabrico de um produto em específico. Isto permite que quando surge a necessidade de mudar a configuração do processo de fabrico para se produzir um produto diferente apenas seja preciso fazer o descarregamento do novo processo pretendido para os equipamentos de controlo em questão.

Esta Dissertação surge como resposta a um caso destes onde um cliente encomendou à JPM Industry o projeto e instalação de uma fábrica para produtos na indústria de processo de detergentes e cosméticos e pediu para que o sistema de gestão das receitas para produção fosse desenvolvido pela própria empresa visto não querer utilizar softwares de terceiros para esse efeito. Como quem está a projetar as instalações é a JPM Industry é esta quem impõe os requisitos para o sistema que pretende instalar no cliente.

Assim o principal objetivo da Dissertação passa por desenvolver um Sistema de Gestão de Receitas para Indústria de Processo para ser instalado nesse cliente e este sistema tem como requisitos funcionais:

- Criar, editar e apagar receitas;
- Guardar as receitas numa base de dados;
- Permitir consulta dos materiais disponíveis;
- Gerar relatórios da produção de cada lote;
- Permitir consulta dos relatórios gerados;
- Ser executado num navegador web;
- Deve ser modular.

O sistema de controlo que será utilizado na fábrica para se descarregar o processo de produção para os controladores terá de ir buscar à base de dados a informação gerada e armazenada pelo Sistema de Gestão de Receitas. Além do desenvolvimento do sistema que cumpra com os requisitos anteriormente descritos e impostos pela empresa, são também objetivos desta Dissertação:

- Estudo sobre a indústria de processo;
- Estudo das normas aplicáveis a este tipo de sistema;
- Estudo das soluções disponíveis no mercado capazes de responder aos requisitos impostos para o sistema;
- Estudo de soluções a implementar para o desenvolvimento de um sistema deste tipo;
- Implementação de uma solução que cumpra com os requisitos impostos;
- Testes ao sistema criado;
- Documentar o sistema criado;
- Apresentação do sistema criado à empresa.

De salientar é o facto de este sistema não visar ser certificado nem reconhecido por qualquer entidade externa, podendo assim não vir a cumprir com todas as normas passíveis de serem aplicadas a este tipo de sistema caso a empresa não veja necessidade para tal.

1.3 Metodologia Seguida no Projeto

Esta Dissertação terá três fases distintas, uma fase de análise, uma de planejamento e outra de execução. Na primeira fase será feito um estudo aprofundado sobre as normas aplicáveis a este tipo de sistema de modo a que o sistema criado vá de encontro às mesmas e será também feito um estudo das soluções disponíveis no mercado que consigam executar as funções pretendidas para o sistema. Na fase de planejamento será projetada uma solução para o sistema em questão onde será detalhado o funcionamento de cada parte do sistema e como estas interagem entre si. Por fim na fase de execução será implementada a solução apresentada para o sistema na fase anterior. Além da implementação será feita ainda uma série de testes ao sistema de modo a validar o bom funcionamento do mesmo.

1.4 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 4 capítulos. No capítulo 2, é descrito o estado da arte, é feito um estudo de normas que sejam relevantes para este tipo de sistemas, dos sistemas que existem atualmente no mercado que sejam semelhantes ao que se pretende desenvolver, é feito um levantamento das suas principais funcionalidades e também um levantamento das tecnologias que permitam fazer uma implementação do sistema de modo a que este tenha uma interface web. No capítulo 3 está presente todo o projeto do sistema desde a sua arquitetura funcional até aos mockups das diversas páginas da sua interface. No capítulo 4 é descrita a implementação protótipo que foi feita do Sistema de Gestão de Receitas para Indústria de Processo. Por fim no capítulo 5 são descritas as principais conclusões da implementação do sistema que foi projetado, são abordados os objetivos que ficaram por cumprir, possíveis melhorias ao sistema projetado e implementado e são ainda abordados os principais cuidados a ter para quem continuar com o desenvolvimento e implementação do sistema.

Capítulo 2

Estado da Arte

Neste capítulo é feito um levantamento sobre o estado da arte. Isto inclui um estudo sobre o que é a indústria de processo, uma análise dos problemas existentes neste tipo de indústria, um estudo sobre as normas relevantes para este tipo de indústria e um estudo sobre os diversos Sistemas de Gestão de Receitas que o mercado atualmente disponibiliza capazes de cumprir com os requisitos que o sistema que se pretende desenvolver possui. Além disso é ainda feito um levantamento das ferramentas web que permitem uma implementação de um sistema destes que seja capaz de correr num navegador web.

2.1 Indústria de Processo

Segundo a *American Production and Inventory Control Society* (APICS) [6] a indústria de processo está ligada a um tipo de produção que adiciona valor através da mistura, da separação ou de reações químicas e pode ser executada tanto em lotes como em modo contínuo. Entenda-se por produção em lotes casos onde a matéria é processada em grupo e só passa para a etapa ou estação seguinte quando todo o lote especificado está processado e completou a etapa a que estava associado. Normalmente ligados a este tipo de indústria estão produtos tais como comida, bebida, químicos, fármacos ou outros bens para consumo. Estes produtos têm todos em comum o facto de não se conseguir reverter o processo e voltar a obter os ingredientes originais.

Também associados a esta indústria surgem dois termos muito importantes, sendo eles receita e fórmula. Entenda-se por receita um conjunto de instruções que especificam passo a passo como uma tarefa deve ser desempenhada e por fórmula o conjunto de ingredientes, com as respetivas quantidades, que são necessários para a produção de um determinado produto. Como esta indústria baseia a sua produção em fórmulas e receitas, todo o processo de produção é escalável podendo assim variar a quantidade final do produto que se pretende obter ajustando apenas as quantidades dos ingredientes na fórmula sem alterar o procedimento da receita.

2.2 Erros de Intervenção Humana

Normalmente quando falamos em operações não automatizadas na indústria de processo estamos a falar em procedimentos iniciados manualmente por operadores que depois vão iterando entre os diversos passos que a receita que dispõem, muitas vezes em papel, lhes dá e vão realizando as ações necessárias à concretização da receita. Numa fábrica que execute várias receitas por dia, que envolvem diferentes passos na sua produção e diferentes condições de segurança, é normal que acabem por surgir erros e enganos por parte do operador que levem a que depois o produto final não corresponda às especificações desejadas e consequentemente o lote seja dado como perdido.

Segundo o *Center for Chemical Process Safety* (CCPS) [7] erros que podem surgir do engano ou da falta de atenção do operador são:

- Passos da receita não serem executado pela ordem correta, na altura correta e com a duração correta;
- Saltar passos na receita;
- Ações com taxas diferentes das especificadas;
- Adição de material errado ou quantidade incorreta;
- Resposta errada perante falhas de equipamentos.

A automatização do processo de produção permite um maior controlo do processo em si e permite assim fazer um controlo de qualidade superior e reduzir o número de erros introduzidos no processo por parte do operador. É nesta automatização que surgem os Sistemas de Gestão de Receitas. Um exemplo apresentado por Joel Haight [8] que nos permite verificar isso é o seguinte: Um passo do processo da receita em que é preciso fazer a contagem de 2 horas e depois aumentar a velocidade de rotação para 40 rotações por minuto (rpm) e terminado este passo tem de adicionar 240 litros (L) de água desionizada. Neste exemplo, caso o controlo seja feito por um sistema automatizado é mais fácil manter o registo do tempo que passou até chegar às 2 horas e pode manter um maior controlo sobre a velocidade de rotação exigida. Há ainda também a vantagem de não haver o risco de adicionar outro produto que não a água desionizada pois esse controlo é feito pelo sistema. Agora existe também vantagens em deixar o processo aberto a interações por parte dos operadores porque neste mesmo exemplo caso queiramos que a agitação não crie espuma, é mais fácil ao operador baixar à velocidade de rotação do que o sistema de controlo se aperceber da espuma.

2.3 Sistema de Gestão de Receitas

Um Sistema de Gestão de Receitas permite que o operador apenas tenha de seleccionar a receita do produto que pretende produzir, introduzir as alterações necessárias à mesma para ajustar a

produção a algum equipamento em específico, caso seja necessário, e depois fazer a transferência da mesma para os controladores do processo de fabrico. Mesmo com o processo automatizado, como já referido, pode haver receitas que precisem da intervenção do operador para a realização de ajustes à produção. Normalmente são criados níveis diferentes de permissões para quem pode criar, alterar, seleccionar e operar sobre uma receita tendo em vista a redução dos erros que também poderão surgir neste tipo de processo. Estas ações podem ser executadas previamente à execução da receita ou mesmo durante a execução dependendo das permissões.

Por norma este tipo de sistema é responsável por fazer verificações de segurança às receitas criadas, como por exemplo ver se a execução da receita é compatível com os equipamentos seleccionados. É também responsável por analisar, rever e manter toda a informação existente das receitas na sua base de dados como por exemplo proteger alterações não autorizadas, manter registo de cópias e receitas editadas ou mesmo determinar quais os equipamentos a que as receitas estão destinadas. Não nos podemos esquecer que normalmente também são responsáveis pela criação de documentação com informação da produção que permite uma validação por parte dos engenheiros e dos laboratórios do lote produzido [7].

Um sistema de Gestão de Receitas para Indústria de Processo que possua estas funcionalidades permite manter consistência na qualidade do material dos lotes produzidos, reduzir aos tempos de configuração dos equipamentos, reduzir o tempo de desenvolvimento de novas receitas, reduzir o tempo que demora a colocar novos produtos no mercado, permite aumentar a cadência de produção melhorando a utilização dos equipamentos disponíveis e com isto a empresa consegue também aumentar os seus lucros.

2.4 Interação de um Sistema de Gestão de Receitas com Outros Sistemas

De modo a facilitar a compreensão de onde se integra um sistema deste tipo entre os diversos sistemas de controlo e de informação de uma empresa surge a arquitetura *Computer Integrated Manufacturing* (CIM). Esta permite aplicar uma melhor gestão da produção dividindo todo o processo em 4 camadas onde é definido o que contem cada camada, que processos ocorrem dentro dela e como estes sistemas se integram e comunicam entre si. Esta separação é apresentada na Figura 2.1 da página 8.

Segundo Luís Almeida [9] as camadas pelas quais a CIM é definida são: a camada corporativa, a camada da fábrica, a camada de controlo do processo e a camada do processo. Na camada corporativa surgem os sistemas *Enterprise Resource Planning* (ERP) que são sistemas de informação que integram dados sobre a perspectiva funcional da organização como finanças, vendas, recursos humanos e podem também incluir sistemas que ajudem na tomada de decisões relativas à produção e visam integrar todos os departamentos da empresa. Na camada da fábrica surgem os sistemas *Manufacturing Execution System* (MES) que permitem ter acesso em tempo real a toda a informação disponível do processo de transformação a decorrer. Esta informação concede a possibilidade de realizar ajustes à produção, tanto ao nível dos equipamentos a utilizar bem como

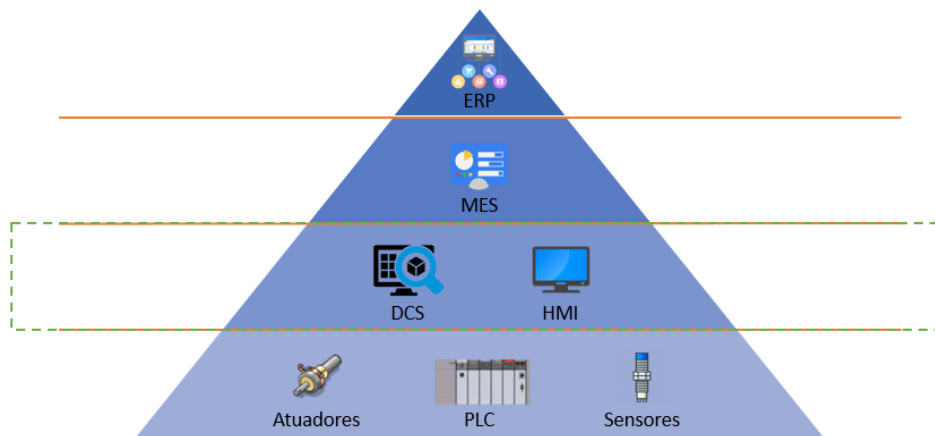


Figura 2.1: *Computer Integrated Manufacturing (CIM)*

o agendamento da produção. Na camada de controlo do processo surgem os sistemas *Distributed Control System (DCS)*. Estes sistemas permitem dividir os vários elementos de controlo ao longo da fábrica e controla-los através de uma interface gráfica *Human Machine Interface (HMI)*. Com esta divisão é possível garantir que em caso de falha de algum controlador os restantes conseguem continuar a produção independentemente do que falhou. Na camada do processo surgem os elementos de controlo, os atuadores e os sensores.

Tendo em consideração esta análise, podemos ver que um Sistema de Gestão de Receitas integra-se na camada de controlo do processo.

2.5 Normas

Uma empresa que tenha como objetivo produzir algo, de maneira a que possa colocar no mercado os seus produtos, tem de cumprir com normas e regras de boas práticas impostas por algumas entidades internacionais, ou locais, de modo a que os seus produtos possam ser certificados para tal. Algumas dessas entidades a ter em conta para este caso em específico de um Sistema de Gestão de Receitas para Indústria de Processo são: a *International Society of Automation (ISA)*, a *International Electrotechnical Commission (IEC)*, a *Food and Drug Administration (FDA)* e a *European Union (EU)* com as *Guidelines to Good Manufacturing Practice (GMP)*.

2.5.1 ISA88 / IEC 61512

De maneira a normalizar a implementação de sistemas de controlo de produção por lotes surgiu em 1995 a norma ISA88. Segundo a *International Society of Automation (ISA)* [10] esta norma fornece uma orientação tanto a nível da estrutura bem como da terminologia utilizada na criação do sistema deste género permitindo assim criar um sistema de controlo modular. A ISA88 acabaria por ser adotada mais tarde pelo IEC, em 1997, adquirindo o nome de IEC 61512.

Atualmente a norma possui um total de 4 partes, que foram sendo publicadas conforme a norma ia evoluindo, sendo elas:

- ANSI/ISA-88.00.01-2010 Batch Control Part 1: Models and terminology;
- ANSI/ISA-88.00.02-2001 Batch Control Part 2: Data structures and guidelines for languages;
- ANSI/ISA-88.00.03-2003 Batch Control Part 3: General and site recipe models and representation;
- ANSI/ISA-88.00.04-2006 Batch Control Part 4: Batch Production Records.

Tal como o nome indica e de acordo com a *International Society of Automation (ISA)* [1], a primeira parte da norma define os modelos e as terminologias a utilizar quando se trabalha com a gestão e controlo de processamento e produção de lotes. Oferece assim a oportunidade de adotar um modelo que seja partilhado entre vários utilizadores e fornecedores e oferece também um conjunto de termos transversais a estes. De qualquer maneira a norma é meramente sugestiva não sendo obrigatório seguir o que esta aconselha. Já a segunda parte da norma, novamente segundo a *International Society of Automation (ISA)* [2], define os modelos e estruturas para os diversos conjuntos de dados, define também as tabelas da base de dados através das quais as informações são trocadas entre diversos sistemas e além disso define ainda a linguagem gráfica para a representação das receitas. A terceira parte fornece informações mais específicas relacionadas com os modelos e as representações das *General Recipes* e *Site Recipes*. Já a quarta parte orienta a nível de registos que serão feitos durante a produção e execução das receitas.

A norma ISA88 tem como um dos seus objetivos fazer uma distinção clara entre os equipamentos disponíveis para a produção e o procedimento da receita em si para a produção do lote pretendido. Esta distinção permite a execução de receitas diferentes nos mesmos equipamentos ou então a execução da mesma receita em equipamentos diferentes, mas que partilhem as mesmas características. Para isso define os sistemas de controlo de produção em lotes em três modelos distintos:

- ***Physical model*** — onde define a configuração física dos equipamentos;
- ***Procedural model*** — onde define as instruções utilizadas para o controlo da produção em lotes;
- ***Process model*** — onde define as ações de processamento para cada *Unit*.

2.5.1.1 *Physical Model*

O *Physical Model* surge como uma maneira de dividir hierarquicamente a estrutura dos equipamentos disponíveis na fábrica, ou seja, a cada nível entramos mais no detalhe do equipamento. Este modelo divide-se em sete níveis, mas os primeiros três não são relevantes para este tipo de

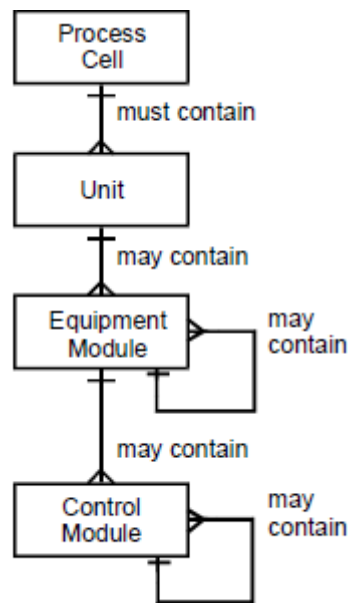


Figura 2.2: *Physical Model* [1]

sistemas e não são utilizados pela norma. Assim sendo, tal como se pode verificar na Figura 2.2, os quatro níveis relevantes são:

- **Process Cell** — conjunto de *Units*, *Equipment Modules* e *Control Modules* que contém todos os equipamentos necessários para a produção de um lote;
- **Unit** — conjunto de *Equipment Modules* e *Control Modules*. Normalmente operam independentemente umas das outras e só conseguem processar um lote de cada vez. Estamos a falar por exemplo em tanques de mistura ou em reatores;
- **Equipment Module** — conjunto de *Control Modules* e possivelmente de outros *Equipment Modules* que executam um conjunto de ações de produção. Algumas dessas ações podem ser misturar ou pesar, por exemplo;
- **Control Module** — conjunto de sensores, atuadores e outros *Control Modules* que são orientados ao estado do processo e seu controlo. Alguns exemplos passam por motores, válvulas, controladores PID, etc.

2.5.1.2 Procedural Model

Quanto ao *Procedural Model* ele define o processo usado na criação de um produto e este, conforme se pode verificar na Figura 2.3 da página 11, divide-se nos seguintes níveis:

- **Procedure** — procedimento composto por um conjunto de vários *Unit Procedures* e que é responsável pela criação de um produto. Um exemplo é “Produção de cosmético A”;

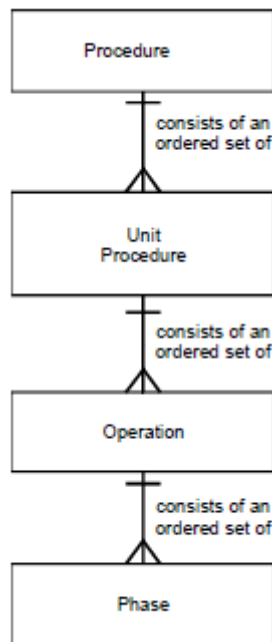


Figura 2.3: *Procedural Control Model* [1]

- **Unit Procedure** — conjunto de várias *Operations* devidamente ordenadas e que é responsável por executar as ações numa *Unit* em específico que foi definida no modelo físico. Um possível exemplo é “Preparação de aditivo B”;
- **Operation** — conjunto de *Phases* devidamente ordenadas que é responsável pelo processamento dos ingredientes. Normalmente correspondem a mudanças físicas ou químicas nos materiais e um exemplo será “Adicionar água, aquecer e manter temperatura”;
- **Phase** — ações elementares no processo de fabrico do produto. Um exemplo é “Adicionar açúcar”.

2.5.1.3 Process model

O *Process model* é em tudo semelhante ao *Procedural Model*, mas especifica os vários parâmetros necessários à execução dos procedimentos escolhidos. Assim neste modelo, como se pode verificar na Figura 2.4 da página 12, temos os níveis:

- **Process** — semelhante a um *Procedure*;
- **Process Stage** — semelhante a um *Unit Procedure*;
- **Process Operation** — semelhante a uma *Operation*, mas fornecendo mais detalhes, sendo um exemplo “Adicionar 20l de água, aquecer até 50°C, manter temperatura 10 minutos”;
- **Process Actions** — semelhante a uma *Phase*, mas mais detalhado, como por exemplo “Adicionar 50Kg de açúcar”.

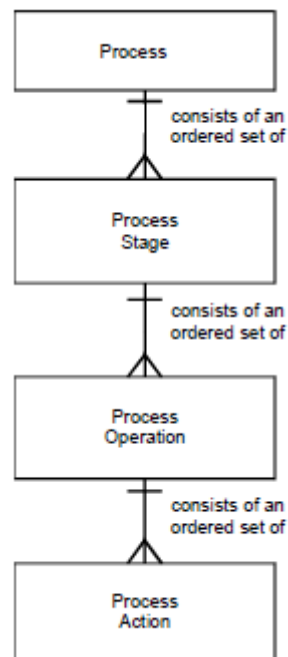


Figura 2.4: *Process model* [1]

2.5.1.4 Receitas

Como elo de ligação entre estes três modelos surgem as receitas. A norma especifica os seguintes tipos de receitas:

- **General Recipe** — esta é a receita que serve de base para os restantes tipos de receita. Neste tipo de receita é definido apenas que matérias primas usar, quantidades e que ações são necessárias executar. Não é referido em que equipamentos deve ser executada nem qualquer outra informação extra;
- **Site Recipe** — este tipo de receita deriva da *General Recipe* e adiciona informações referentes ao local onde podem ser executadas caso, por exemplo, a empresa disponha de várias instalações diferentes;
- **Master Recipe** — este tipo de receita deriva da *Site Recipe* e adiciona informação referente ao tipo de equipamentos necessários para a execução da mesma;
- **Control Recipe** — este tipo de receita deriva da *Master Recipe* e é criada aquando a execução de uma *Master Recipe*. Pode ser igual ou então ter informação personalizada para a execução de um lote em específico.

Independentemente do tipo de receita, estas são sempre constituídas por um *Header*, uma *Formula*, *Equipment Requirements* e *Recipe Procedure*. O *Header* corresponde a informação específica da receita como por exemplo o seu autor, data de criação, informação referente ao produto

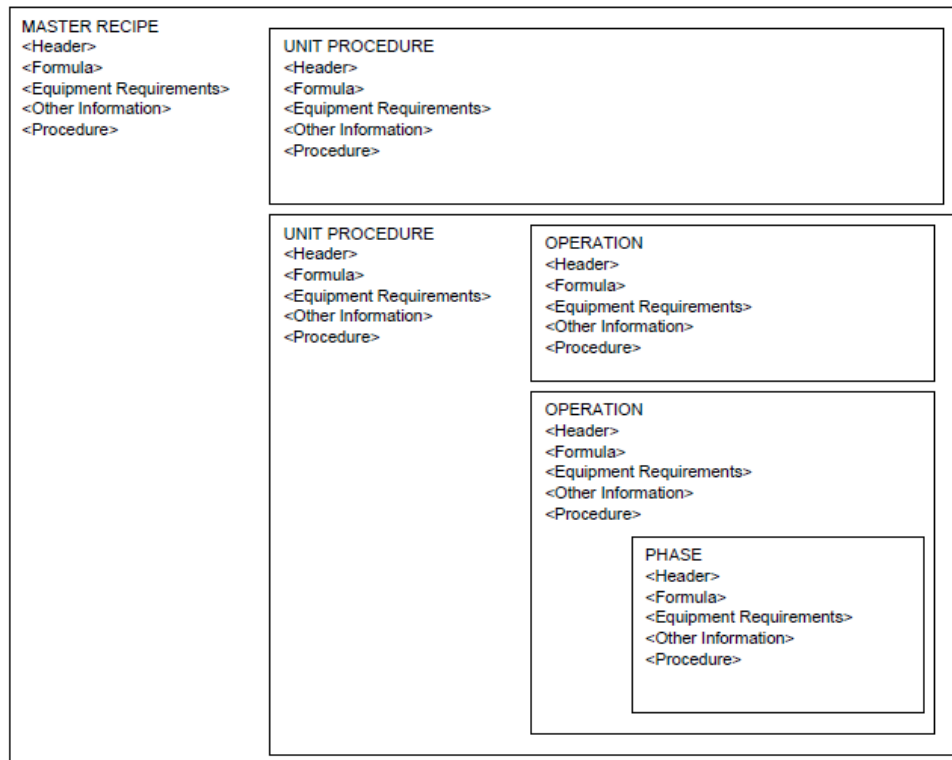


Figura 2.5: Elementos Constituintes de uma Receita [2]

que esta produz, etc. Já a *Formula* refere quais são as entradas, saídas e variáveis de processo, ou parâmetros, da receita. Os *Equipment Requirements* fazem referência aos equipamentos que a receita precisa para poder ser executada e se obter o produto final pretendido. O *Recipe Procedure* é todo o conjunto de passos devidamente ordenados responsáveis pela transformação dos ingredientes no produto final. Cada receita pode posteriormente ser guardada, caso assim se queira, num formato XML como o representado na Figura 2.5.

Estas receitas são representadas graficamente através de uma linguagem chamada *Procedure Function Charts* (PFC). A sua execução ocorre do topo para baixo e permite a execução em paralelo de vários elementos. Esta linguagem é definida por um conjunto de símbolos que definem os vários passos da receita. Apenas a simbologia é imposta sendo que a sua dimensão ou tamanho do texto são deixados à escolha do utilizador.

Nestes símbolos surgem elementos base como os representados na Figura A.1 da página 67 onde cada um deles representa um dos quatro níveis apresentados no modelo processual na primeira parte da norma ISA88, não esquecendo que estes estão ligados hierarquicamente entre si.

Depois é exigido que cada PFC, independentemente do nível processual que represente, precisa de ter um símbolo de início e um símbolo de fim como os representados na Figura A.2 da página 67 e na Figura A.3 da página 68 respetivamente. Deste modo a receita não pode correr em modo contínuo, em contraste do que acontece num *Sequential Function Chart* (SFC).

Entre dois elementos base pode surgir, ou não, uma transição. Um exemplo de uma ligação

direta sem transição poder ser visto na Figura A.4 da página 68 e um exemplo com uma transição pode ser visto na Figura A.5 da página 68. Caso exista, a transição é representada por dois traços e será definida por uma expressão que ao ser avaliada terá de ser ou verdadeira ou falsa de modo a avaliar a possibilidade de continuação da evolução da receita.

Quando um elemento tem possibilidades distintas para dar continuação ao processo da receita este terá uma sequência onde será feita a seleção do caminho a seguir tendo em consideração as transições colocadas. Um exemplo representativo deste caso pode ser encontrado na Figura A.6 da página 69 e num caso destes a escolha só poderá cair sobre uma das alternativas. O fim destas possíveis alternativas deve ser representado como demonstrado na Figura A.7 da página 69.

Caso se pretenda executar vários passos em paralelo também há essa possibilidade, e se assim o pretendermos devemos representar a união entre a transição e os passos a serem executados através de dois traços paralelos conforme demonstrado na Figura A.8 da página 70. Uma sequência deste género só termina quando todos os ramos da mesma terminarem também. Assim sendo devemos representar um caso destes conforme demonstrado na Figura A.9 da página 70, onde a transição só é tomada em consideração quando todos os ramos da sequência anterior estão finalizados.

2.5.2 EC-GMP Annex 11 e FDA CFR 21 Part 11

Quando falámos na produção de bens nas áreas da medicina e da biotecnologia que estejam diretamente relacionados com seres vivos, surgem as diretivas EC-GMP Annex 11 e a FDA CFR 21 Part 11 que têm como principal objetivo regular a produção e validar os sistemas computadorizados das empresas neste tipo de indústria.

Através da análise de um documento disponibilizado pela EduQuest [11] onde é feita a comparação entre as duas diretivas podemos retirar as seguintes conclusões. A EC-GMP Annex 11 cobre os princípios e orientações regulamentadas nas GPM para sistemas computadorizados. Este anexo visa qualquer fabricante, do tipo de indústrias referido anteriormente, que pretenda produzir ou vender no mercado da União Europeia. O seu foco passa por fazer um controlo de qualidade dos sistemas computadorizados e tem como principal objetivo assegurar que o sistema computadorizado a usar consegue obter o mesmo nível de qualidade que um sistema manual sem aumento de risco no geral. A FDA CFR 21 Part 11 estabelece os requisitos a nível técnico e de controlo da produção de maneira a que sejam mantidos registos eletrónicos da produção. O seu foco passa por usar registos eletrónicos em todos os sistemas computadorizados e tem como principal objetivo garantir que estes registos eletrónicos tenham a mesma fiabilidade como se fossem feitos à mão.

Esta tipo de diretivas permitir um maior controlo sobre a produção através dos registos eletrónicos e de assinaturas eletrónicas e estes registos permitem um maior controlo da qualidade do produto produzido e permitem também fazer uma rastreabilidade do produto através do seu histórico de produção.

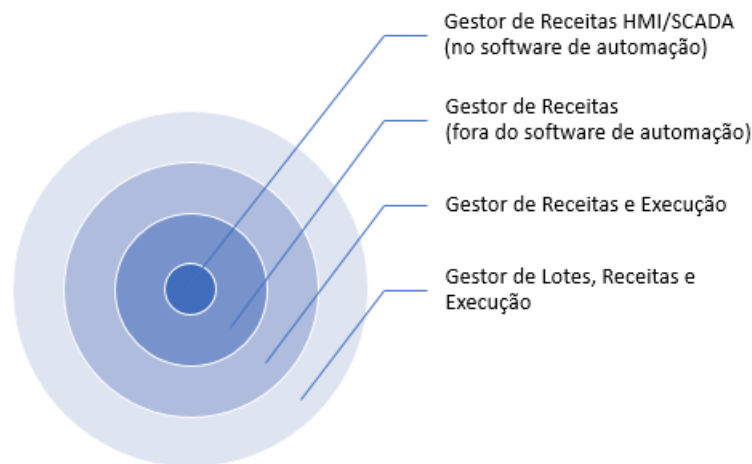


Figura 2.6: Níveis de Complexidade dos sistemas disponíveis

2.6 Softwares no Mercado

Atualmente o mercado tem uma oferta de produtos que têm diferentes níveis de complexidade, e por complexidade entenda-se número de funcionalidades e configurações disponíveis para o utilizador. Após uma filtragem inicial, o estudo vai incidir sobre os seguintes Sistemas de Gestão de Receitas:

- Wonderware InBatch [12];
- Wonderware Recipe Manager Plus Standard Edition e Professional Edition [13];
- Rockwell Software FactoryTalk Batch [14];
- Iconics Recipe + Batch Management [15];
- Mar-Kov Recipe Manager Professional [16];
- NovaTech Flexbatch (Batch and Recipe Management software) [17].

Após realizada uma análise destes sistemas facilmente se chega à conclusão que este tipo de sistemas se dividem em quatro níveis distintos como demonstrado na Figura 2.6.

No primeiro nível surgem os sistemas de gestão de receitas que correm no software de automação, onde no local, através de uma interface HMI / Supervisory Control and Data Acquisition (SCADA), podemos fazer uma configuração dos equipamentos e tanto gerir como alterar parâmetros das fórmulas a utilizar. Permitem também fazer a transferência desses parâmetros para o sistema de controlo sendo que o sistema de controlo é um sistema independente do gestor de receitas.

No segundo nível surgem os sistemas de gestão de receitas que partilham as mesmas funcionalidades que os do primeiro nível, mas que não nos obrigam a estar presentes no local de fabrico.

O ajuste dos parâmetros das fórmulas e a gestão das receitas podem ser feitos fora das instalações de produção da empresa em que depois podemos fazer o download dos parâmetros para o sistema de controlo via servidor podendo assim ser aplicados mais rapidamente em diferentes instalações de produção.

No terceiro nível surgem os sistemas de gestão de receitas que partilham as funcionalidades dos anteriores e que adicionam a possibilidade de executar uma receita por equipamento remotamente e acompanhar a sua execução através da interface gráfica disponibilizada pelo software.

Por fim, no quarto nível surgem os sistemas que além da gestão de receitas permitem também a gestão de lotes. Além das funcionalidades dos outros três níveis permitem a produção de vários lotes ao mesmo tempo, com caminhos de produção partilhados e em que podemos fazer um agendamento da produção e optar pela ordem da produção que pretendermos de acordo com as encomendas existentes.

Assim, os sistemas de gestão escolhidos para o estudo encontram-se nos seguintes níveis:

- 2º nível:
 - Wonderware Recipe Manager Plus Standard Edition;
- 3º nível:
 - Wonderware Recipe Manager Plus Professional Edition;
- 4º nível:
 - Wonderware InBatch;
 - Rockwell Software FactoryTalk Batch;
 - Iconics Recipe + Batch Management;
 - Mar-Kov Recipe Manager Professional;
 - NovaTech Flexbatch (Batch and Recipe Management software).

Após concluída a análise destes Sistemas de Gestão de Receitas destacaram-se as funcionalidades descritas nos próximos subcapítulos.

2.6.1 Gestão de Lotes

Na gestão de lotes destacam-se as seguintes funcionalidades:

- Execução simultânea de vários lotes concorrentes pelos mesmos equipamentos;
- Forçar a execução do lote de forma sequencial libertando apenas para execução a etapa ativa no momento;
- Agendamento de lotes;
- Simular o processamento do lote;

- Simular o agendamento de lotes;
- Possibilidade de escolher da lista de produção de lotes aprovados que lotes queremos produzir primeiro em vez de seguir o método *first-in-first-out* (FIFO);
- Quando o tamanho do lote é superior ao máximo definido é sugerida uma divisão em vários lotes. Após validação estes lotes são adicionados à lista de produção;
- Validação do lote, isto é, verificação se a receita existe na base de dados, modelo do processo, requisitos de materiais, requisitos de equipamentos e tamanho do lote dentro dos limites pretendidos;
- Cálculo dos custos do produto acabado com base no preço das matérias primas.

2.6.2 Gestão de Receitas

Na gestão das receitas destacam-se as seguintes funcionalidades:

- Criar, copiar e modificar receitas utilizando uma interface gráfica de blocos (IEC 61131-3, linguagem gráfica SFC) não havendo necessidade de alterar o código do sistema de controlo;
- Configuração das receitas baseada na norma ISA88 definindo os equipamentos, as suas capacidades de processamento e seus requisitos de controlo;
- Simular a execução das receitas;
- Realizar alterações aos estados das fases durante a simulação;
- Acelerar o processo de *debug*, na simulação, avançando uma fase de cada vez;
- Configurar fases de transferência de materiais e ligações entre equipamentos de modo a permitir caminhos flexíveis entre equipamentos e facilitar a execução de vários lotes em simultâneo;
- Criar *templates* de receitas para todas as instalações da empresa;
- Ajustar as receitas às diferentes instalações da empresa;
- Uma receita pode ser aplicada a qualquer linha de processamento em que o equipamento cumpra os requisitos definidos na receita;
- As matérias primas, produtos intermédios e produtos acabados podem ser configurados como quantidades ou como percentagens do tamanho do lote;
- As fórmulas em que as quantidades são expressas em percentagem são escaladas automaticamente quando a receita é executada e passa a *Control Recipe*;
- Adicionar características de segurança tais como verificado-por ou feito-por de modo a garantir uma correta execução da receita;

- Validação de receitas, isto é, verificar se o modelo do processo está feito corretamente, informação do material, se os *templates* para os relatórios existem, se o lote tem tamanho mínimo, máximo e padrão e se os parâmetros da formula estão corretamente ligados;
- Durante o desenvolvimento ou execução da receita, o operador poder modificar os parâmetros da receita (ex: quantidade de material a adicionar), os procedimentos da receita (ex: não usar o agitador X neste lote), o calendário da produção (ex: fazer 10 lotes em vez de 5), as regras para o arranque do lote (ex: Arrancar o lote após a operação Y estar concluída), a utilização do equipamento (ex: Usar apenas o depósito 2 neste lote), ou escalar a quantidade do lote (ex: fazer 500Kg em vez de 200Kg);
- Fazer importação e exportação de ficheiros XML baseados no standard *Batch Markup Language* (BatchML), o que permite mover ou partilhar receitas entre vários sistemas;
- Quando são feitas alterações aos equipamentos, fórmula ou receitas são automaticamente criadas novas versões. Após as alterações terem sido feitas é necessária nova aprovação para que esta possa ser usada em produção. Depois de aprovadas as alterações, são geradas notificações e outras atualizações que estejam relacionadas;
- Guardar automaticamente novas versões de receitas em BatchML quando são guardadas ou aprovadas;
- Suporte para 5 níveis de aprovações;
- Antes de executar uma fase verifica se a fase está pronta a ser executada, se sim os parâmetros são transferidos e a fase é iniciada;
- Gerar eventos durante as execuções de modo a indicar alterações de estado ou a aplicar ações externas como assinaturas ou autorizações;
- Criar biblioteca com as classes de fases e operações que podem ser utilizadas.

2.6.3 Gestão de Materiais, Rastreabilidade e Registo

Na gestão de materiais, rastreabilidade e registos destacam-se as seguintes funcionalidades:

- Acesso à genealogia completa dos materiais/produtos através de registos eletrónicos realizados para a base de dados;
- Cumprimento das diretivas FDA 21 CFR Part 11 e EC-GMP Annex 11;
- Registo do consumo das matérias primas, seus movimentos e níveis de inventário;
- Mapeamento feito através de *tags* específicas das unidades, processos e equipamentos;
- Ver localização atual do material;

- Diferentes estados do material: ingredientes, produtos intermédios (num estado da produção), processados e produtos derivados (*by-products*);
- As matérias primas e outros materiais podem mudar a sua localização pois a receita é independente disso não afetando a sua execução;
- Base de dados dos materiais é atualizada ao longo do processamento dos lotes (incluindo produtos intermédios);
- Acesso ao *Work in Process* (WIP);
- Adicionar características de segurança tais como verificado-por ou feito-por de modo a guardar registo de quem processou o lote;
- Assinaturas armazenadas de modo a que não haja a possibilidade de as alterar indo assim de encontro à diretiva FDA 21 CFR Part 11;
- Histórico das receitas com data, hora, nome do autor e comentários opcionais.

2.6.4 Relatórios

Relacionadas com os relatórios foram encontradas as seguinte funcionalidades:

- Comparação de versões da mesma receita de maneira a analisar as suas diferenças;
- Relatórios da produção disponíveis para consulta web. Podem ser configurados de maneira a serem gerados automaticamente durante a execução do lote ou no final da sua produção. Têm alguns *templates* mas podem ser gerados outros completamente personalizáveis;
- A informação guardada na base de dados pode ser utilizada para criar gráficos ou relatórios com informações sobre o desempenho da fábrica ou mesmo para gerar *Key Performance Indicators* (KPI);
- Informação de cada execução das receitas permitindo replicar versões que apresentem melhores resultados.

2.6.5 Interface

Relacionadas com as interfaces dos sistemas foram recolhidas as seguintes funcionalidades:

- Módulos de *display* que permitem ao operador ter acesso a informação que o deixa interagir com os lotes em execução no sistema. Mostra informações tais como o estado da fase (default, parado, suspenso, em-progresso), permite realizar ações usando botões de comando (parar, suspender, resumir, cancelar) ou mesmo modificar a receita em progresso;
- Aplicação web cliente/servidor independente do sistema de controlo;

- Possibilidade de utilizar a aplicação em qualquer plataforma HMI que suporte browsers com HTML5;
- Interface web encriptada *Secure Socket Layer* (SSL);
- Como pode correr em plataformas mobile Android e iPhone permite aceder a informação em tempo real, interagir com os processos e executar as verificações de segurança/assinaturas a partir de qualquer ponto da fábrica.

2.6.6 Servidor

Normalmente a parte do sistema que corre nos servidores tem as seguintes funcionalidades:

- Vários servidores podem ser configurados de modo a usarem a mesma base de dados para histórico de informação sobre os lotes;
- Duas arquiteturas disponíveis:
 - *Warm Restart Capability*, isto é, capacidade de, após um encerramento inesperado do sistema, restaurar o sistema ao reiniciar para um estado em que tudo estava bem;
 - *Redundant Server Option*, isto é, espelha as operações do servidor principal para o de backup. Na hipótese de haver problemas de hardware no primário, o servidor de backup entra automaticamente em serviço e assume o papel de servidor primário e continua a execução do lote;
- Comunicações OPC-UA de modo a ter acesso a informação em tempo real, monitorizar alarmes ou eventos e aceder a dados históricos.

2.6.7 Plataformas Compatíveis

Este tipo de sistema normalmente é compatível com as seguintes plataformas ou versões superiores:

- Servidor:
 - Windows Server 2008 R2 Standard ou Enterprise Edition (64-Bit);
 - Windows Server 2012 Standard ou Data Center Edition (64-Bit);
 - Windows Server 2012 R2 Standard ou Data Center Edition (64-Bit);
- Cliente:
 - Windows 7 Professional ou Enterprise (32 Bit e 64 Bit);
 - Windows 8 Professional ou Enterprise (32-Bit e 64-Bit);
 - Windows 8.1 Professional ou Enterprise (32-Bit e 64-Bit);
 - Windows 10 Professional ou Enterprise na versão 1607 (64-Bit);

- Base de dados:
 - Microsoft SQL Server 2008 Standard ou Enterprise Edition (32-Bit);
 - Microsoft SQL Server 2008 R2 Standard ou Enterprise Edition (32-Bit e 64-Bit);
 - Microsoft SQL Server 2012 Standard ou Enterprise Edition (32-Bit e 64-Bit);
 - Microsoft SQL Server 2014 Standard ou Enterprise Edition (32-Bit e 64-Bit);
- Browsers (para as aplicações web):
 - Microsoft Internet Explorer, Versão 10 ou superior em Windows;
 - Google Chrome, Versão 32 ou superior em Windows;
 - Mozilla Firefox, Versão 26 ou superior em Windows;
 - Apple Safari, Versão 6.0 em aparelhos iPad.

2.7 Desenvolvimento Web

Hoje em dia devido à grande variedade de plataformas e sistemas operativos existentes onde se pretenda correr a aplicação a desenvolver, surge um enorme problema que é o de fazer o sistema compatível com todas estas plataformas. Assim sendo, segundo John Sonmez [18], a maneira mais fácil de ultrapassar este obstáculo é fazer com que o nosso sistema seja uma aplicação que corra em navegadores web permitindo assim que todos os dispositivos que tenham um navegador web possam correr o sistema desenvolvido. Outra vantagem de aplicar um sistema deste género passa por não ser necessária a instalação do programa que o utilizador pretende usar no dispositivo onde este vai correr.

O desenvolvimento web passa por criar websites que possam ser acedidos posteriormente via internet ou intranet. Este desenvolvimento normalmente tem associado a ele quatro grandes áreas:

- Programação do Lado do Cliente;
- Programação do Lado do Servidor;
- Base de Dados;
- Segurança da Rede.

As primeiras três correspondem às partes ativas do sistema e que comunicam entre si, já a quarta é referente às comunicações entre elas. A maneira como as primeiras interagem entre si pode ser verificada na Figura 2.7 da página 22.

2.7.1 Lado do Cliente

A parte da aplicação que corre no lado do cliente é executada pelo navegador web sendo que este tem de fazer um pedido ao servidor de toda a informação e ficheiros que possa precisar

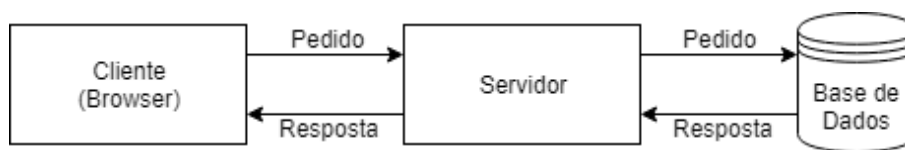


Figura 2.7: Comunicação entre Cliente, Servidor e Base de Dados

para carregar a página pretendida. Esta transferência de informação é feita através do protocolo *Hypertext Transfer Protocol* (HTTP) e na resposta vêm ficheiros *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e *JavaScript* (JS) que depois são transformados numa página web pelo navegador permitindo ao utilizador interpretar e usufruir da informação recebida. Como os pedidos ao servidor podem ser diferentes consoante as entradas dadas pelo utilizador ao sistema, a informação recebida na resposta do servidor também o pode ser. Algumas das principais linguagens utilizadas na parte do cliente já foram referidas anteriormente, sendo elas HTML, CSS e JS.

2.7.1.1 HTML

O HTML é usado para especificar o formato e disposição dos vários elementos pertencentes a uma página web. Assim sendo ele é uma das partes mais importantes no desenvolvimento web. Essa formatação da página é feita através de *tags* que definem as diversas secções de cada página e o conteúdo de cada uma delas. O navegador web utilizará esta informação em conjunto com o CSS e o JS recebidos.

2.7.1.2 CSS

O CSS surgiu para que houvesse uma separação clara entre o conteúdo da página e o seu design/estilo. Com a criação de um ficheiro CSS podemos alterar o estilo de uma ou várias páginas sem ter de mudar elemento a elemento. Assim torna-se bastante útil este género de ferramenta poupando imenso tempo no desenvolvimento e edição de conteúdo web.

2.7.1.3 JS

O JS é uma linguagem executada diretamente no navegador web e que permite dar dinamismo às páginas criadas através da manipulação seu conteúdo. Esta manipulação acontece através da interação com o *Document Object Model* (DOM) podendo assim alterar além do conteúdo também o estilo da página.

2.7.2 Lado do Servidor

Os ficheiros HTML, CSS e JS que chegam ao navegador web do utilizador são previamente preparados e processados pelo servidor e só depois enviados ao cliente. Basicamente as páginas são construídas pelo servidor. Atualmente algumas das linguagens utilizadas para este tipo de

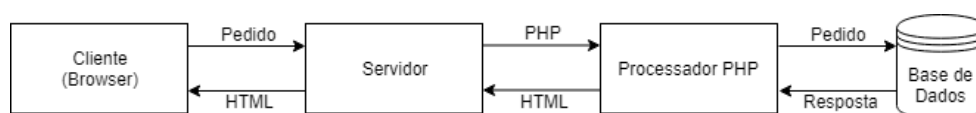


Figura 2.8: Comunicação Utilizando PHP

processamento do lado do cliente são por exemplo ASP.NET, PHP, JAVA, Python, Ruby. Um exemplo de como se processa esta troca de informação pode ser visto na Figura 2.8.

O *Hypertext Preprocessor* (PHP) é um tipo de linguagem que pode ser incorporado em documentos HTML e além disso permite processar dados introduzidos em páginas web e tratar da comunicação com bases de dados onde pode guardar ou ir buscar algumas informações importantes para a página ou para a execução do sistema. Uma das vantagens em utilizar esta linguagem para fazer o processamento da página do lado do servidor é que o código fonte nunca é mostrado ao utilizador final, apenas o servidor tem acesso a ele mantendo assim em segredo partes importantes do sistema.

2.7.3 Base de Dados

Normalmente para armazenar a informação relevante para o website e para o sistema são utilizadas bases de dados. Estes sistemas são muitas vezes referidos como *Relational Database Management System* (RDBMS) e alguns exemplos são Oracle, Microsoft SQL Server, MySQL, PostgreSQL, MariaDB. Essa informação relevante é então guardada nesse sistema em tabelas e cada tabela é um conjunto de várias entradas que consistem em colunas e linhas. Assim cada campo da tabela é referenciado pela sua coluna e pela sua linha. Para se aceder a esta informação utiliza-se PHP que realizará as comunicações com a base de dados usando uma linguagem conhecida como *Structured Query Language* (SQL).

O SQL é uma linguagem normalizada para o acesso a base de dados e manipulação das mesmas. Apesar de normalizada pelo *American National Standards Institute* (ANSI) e pela *International Organization for Standardization* (ISO) existem algumas variantes da mesma fazendo com que diferentes programas tenham depois alguns comandos adicionais específicos.

2.7.4 Segurança da Rede

Hoje em dia um dos aspetos muito importantes no desenvolvimento de um website ou sistema para funcionar através de um navegador web é a segurança. Os ataques podem ser feitos para que consigam ter acesso a informação relevante que o proprietário do website disponha e que seja confidencial ou então simplesmente para interferir com o bom funcionamento do sistema. Normalmente este género de ataques é feito por *scripts* programados para procurar na internet falhas de segurança nos websites acessíveis e tirar partido delas. Algumas dessas falhas e possíveis soluções a ter em consideração são as seguintes.

2.7.4.1 Injeção de SQL

Ataques realizados através da injeção de SQL ocorrem quando são utilizados os campos dos formulários ou mesmo os *Uniform Resource Locator* (URL) das páginas para ganhar acesso às bases de dados. Isto acontece através da injeção de código SQL dentro das *queries* que o sistema irá executar. Isto permite que se adicione, mude ou apague informação na base de dados sem que fosse suposto ter permissão para tal. Uma maneira simples de prevenir isto, se for corretamente implementada, é parametrizar as *queries*. Isto é, separar a *query* dos parâmetros de entrada da mesma. Isto permite validar ou não os parâmetros verificando se são valores válidos ou tentativas de inserir código na *query*.

2.7.4.2 XSS – Cross-site scripting

Ataques do tipo *Cross-site scripting* (XSS) são tentativas de adição de código JS malicioso nas páginas que o utilizador abre para utilizar o website. Assim é preciso prevenir a adição de código JS na página. O principal cuidado a ter passa por fazer uma verificação e validação dos dados introduzidos pelo utilizador, nos campos que lhe é permitido, antes de guardá-los e adicioná-los ao código HTML da página.

2.7.4.3 Encriptação de Palavras Passe

Relativamente às palavras passas devem ser tomadas algumas medidas de segurança. Além de forçar, por exemplo, a palavra passe a ter um número mínimo de caracteres, ou a ter números e letras maiúsculas e minúsculas, é também importante guardá-las na base de dados de forma encriptada. Esta encriptação pode acontecer, por exemplo, utilizando um algoritmo Secure Hash Algorithm (SHA). A grande vantagem de usar uma encriptação deste género é que mesmo que as passwords sejam roubadas fazer o processo inverso não é possível, ou seja, fazer a desencriptação. Sendo que a única alternativa que os atacantes teriam seria fazer uma tentativa de todas as combinações possíveis até conseguirem acesso. Um ataque deste género é conhecido como Ataque por Força Bruta.

2.7.4.4 Ataque por Força Bruta

Como referido anteriormente este tipo de ataque consiste numa tentativa de adivinhar o valor requerido num campo de um sistema onde são testadas todas as combinações possíveis até que seja validado pelo sistema. Assim de maneira a prevenir um ataque deste género existem três possíveis soluções. Uma delas é a limitação do número de vezes que o valor pode ser introduzido bloqueando o acesso após um número especificado de tentativas. Outra alternativa seria impedir o acesso a novas tentativas durante uma quantidade determinada de tempo, aumentando a esse tempo de espera sempre que o utilizador volta a introduzir valores errados. Por fim outra solução passa pela introdução de verificações visuais, conhecidas como *Completely Automated Public*

Turing test to tell Computers and Humans Apart (CAPTCHA) no processo de validação do valor introduzido descartando assim *scripts* que esteja a correr a tentar ganhar acesso ao sistema.

2.7.4.5 *Hypertext Transfer Protocol Secure* (HTTPS)

O protocolo HTTPS foi criado de maneira a garantir ao cliente que está a falar com o servidor pretendido e também a garantir que ninguém consegue interceptar ou mudar o conteúdo da informação existente na comunicação entre os dois. A informação é transmitida de forma encriptada entre o cliente e o servidor e veracidade e fiabilidade do site pode ser verificada pelo navegador web através do certificado que o website disponibiliza para análise.

2.8 Resumo

Neste capítulo foi abordado em que consiste a Indústria de Processo. Com essa análise viu-se que este tipo de indústria é suscetível de criar produtos que não vão de encontro às especificações nem aos padrões de qualidade pretendidos. Isto acontece muitas das vezes por culpa do operador responsável pelo processo de transformação e produção dos respetivos produtos. Assim como resposta não só a este mas a outros problemas semelhantes surgiram os Sistemas de Gestão de Receitas que vieram automatizar a execução deste processo de fabrico. Para complementar estes sistemas surgiram normas que visam normalizar a implementação deste tipo de processo usando modelos e terminologias transversais a todos os fornecedores e fabricantes. Isso permitiu aumentar a qualidade dos produtos finais e garantir um seguimento mais fiel do processo especificado para produção. Das normas analisadas, a que terá mais foco nesta Dissertação será a ISA88 pois o sistema não será responsável pela execução das receitas e assim sendo não terá de fazer o registo da mesma. Apenas precisa de registar dados relativos à criação e alteração das receitas e manter um controlo sobre o acesso aos dados do sistema.

Atualmente o mercado já oferece bastantes soluções a nível de Sistemas de Gestão de Receitas. A maior parte das soluções analisadas, a nível das funcionalidades que dispõem, já cumpre largamente com os requisitos do sistema a desenvolver. Grande parte delas até permite o controlo da execução dos lotes, sendo essa uma das maiores funcionalidades que maior destaque merece apesar de não fazer parte dos requisitos.

Já existe também um conjunto significativo de soluções a nível da implementação que permitem desenvolver um sistema deste género orientado a web. Como estamos a falar de um sistema que irá pertencer a uma empresa e que é um repositório de informação confidencial e de elevada importância, é de todo o interesse seguir algumas boas práticas ao nível da segurança de modo a garantir minimamente a fiabilidade do sistema.

Capítulo 3

Descrição da Solução Proposta

Neste capítulo é apresentada uma proposta de um Sistema de Gestão de Receitas capaz de cumprir com os requisitos necessários. Assim inicialmente é demonstrada a sua arquitetura funcional seguida da estrutura da base de dados que é utilizada para repositório de toda a informação relevante ao sistema. São ainda apresentados os modelos para as diversas páginas que o sistema dispõe e alguma informação relativa a algoritmos ou tecnologias utilizadas na construção e funcionamento do mesmo. É importante referir inicialmente que de modo a permitir uma melhor integração da terminologia da norma ISA88 o sistema será projetado utilizando a língua inglesa.

3.1 Arquitetura Funcional

Tendo em conta os requisitos especificados inicialmente, o Sistema de Gestão de Receitas a desenvolver tem de ser capaz de ser utilizado em dispositivos móveis, em computadores e em consolas SCADA. De maneira a garantir a compatibilidade entre todos estes dispositivos, tal como pode ser verificado na Figura 3.1 da página 28, a alternativa a utilizar passa por fazer com que o sistema corra num navegador web não sendo assim necessária a instalação de software nem o desenvolvimento de diversas versões para que fosse compatível com os sistemas operativos e dispositivos pretendidos.

As páginas que serão apresentadas no navegador, o cliente, serão geradas pelo servidor onde o sistema estiver alojado. A transferência de informação entre cliente-servidor ocorre sobre o protocolo HTTP e nela vão ficheiros HTML e CSS com o conteúdo a mostrar e com o estilo do mesmo. Caso haja transferência de conteúdo dinâmico que não obrigue ao carregamento da página por completo, podem ainda ser transferidos ficheiros *JavaScript Object Notation* (JSON) contendo a informação a atualizar. Caso as páginas fossem constituídas por conteúdo estático não haveria necessidade de alterar os ficheiros HTML a serem transferidos para o cliente, mas como não é esse o caso, pois o sistema é dinâmico, as páginas têm de ser criadas com a linguagem PHP sendo que esta vai permitir ao servidor gerar os ficheiros HTML a enviar para o cliente com informação que inicialmente não estaria no ficheiro HTML. Esta informação introduzida será recolhida de uma base de dados onde estão todas as informações relevantes à execução do sistema. Além deste

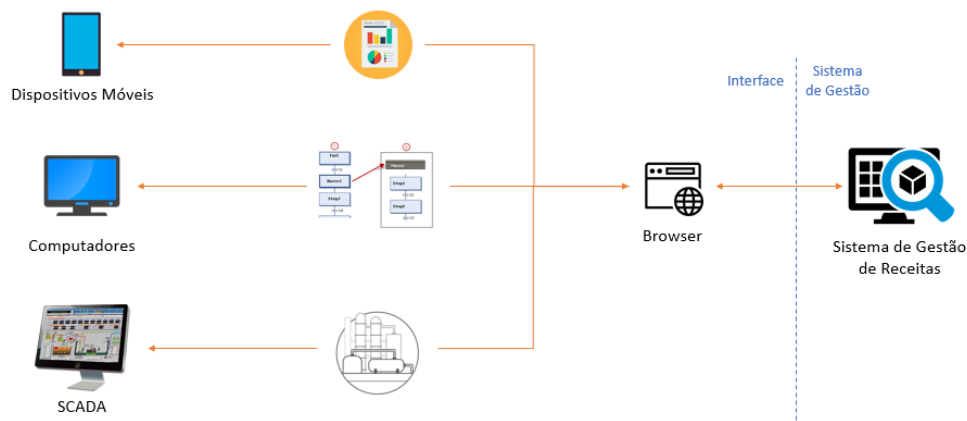


Figura 3.1: Interface do Sistema

caso, a base de dados é também utilizada para efeitos de controlo de acesso ao sistema de modo a validar acessos ou ações sobre o mesmo. Estas interações podem ser verificadas na Figura 3.2 da página 29.

Para que as receitas criadas no sistema sejam usadas para produção, o sistema de controlo irá gerar as receitas através da informação disponível na base de dados introduzida pelo Sistema de Gestão de Receitas. Este sistema de controlo da execução irá introduzir depois todos os dados referentes à produção do lote na base de dados para que o Sistema de Gestão de Receitas possa gerar os relatórios referentes àquele lote com registos de ações efetuadas na produção, registos horários, registos de erros e alarmes, etc. A transferência de informação desde o Sistema de Gestão de Receitas até ao controlador ocorre conforme demonstrado na Figura 3.3 da página 30.

3.2 Base de Dados

A informação relevante à execução do sistema é guardada num esquema de uma base de dados. Nesse esquema, apesar de na realidade não existir uma divisão entre as tabelas a usar, podemos mesmo assim identificar cinco grupos distintos de tabelas que são orientados a partes específicas do sistema. Essa divisão pode ser verificada na Figura 3.4 da página 31, sendo que os grupos são os seguintes: o grupo de dados referentes aos utilizadores, o grupo de dados referentes aos equipamentos disponíveis na fábrica, o grupo de dados quanto aos materiais e ingredientes disponíveis para a produção, o grupo de dados referente às receitas e ainda o grupo de dados da execução da receita e que são necessários para a geração dos relatórios.

A estruturação desta base de dados baseou-se nas tabelas disponibilizadas na parte 2 da norma ISA88 [2] onde são demonstradas as tabelas que se deve usar quando se troca informação relativa às receitas entre sistemas diferentes.

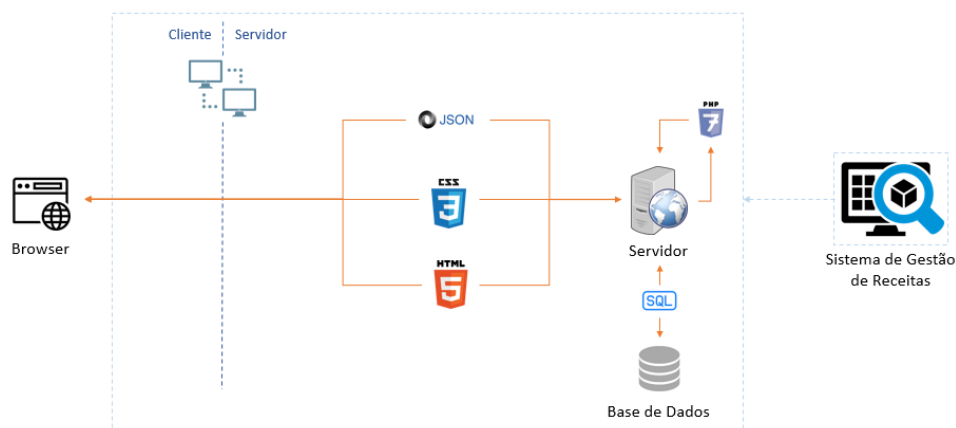


Figura 3.2: Funcionamento Interno do Sistema

3.2.1 Utilizadores

No grupo de tabelas dedicadas aos utilizadores temos duas tabelas, a `users` e a `users_logs`. O conteúdo de cada uma delas pode ser consultado na Figura 3.5 da página 32.

A primeira surge de modo a permitir manter um registo dos utilizadores do sistema e manter tanto um controlo sobre a validação do acesso dos utilizadores ao sistema como validar acesso a partes restritas do sistema onde apenas algumas pessoas possam ter permissões para tal. A segunda serve simplesmente para manter um registo sobre os acessos dos utilizadores podendo assim haver um controlo de quem utiliza o sistema.

3.2.2 Equipamentos

Na parte responsável pelos dados dos equipamentos temos sete tabelas, a `equip_element`, a `equip_property`, a `equip_phases`, a `equip_interface`, a `equip_interface_parameter`, a `equip_link` e a `units_measure`. O conteúdo de cada uma delas pode ser consultado na Figura 3.6 da página 33.

A `equip_element` é utilizada para se manter um registo de todas as classes de equipamentos existentes na fábrica e que equipamentos fazem parte da fábrica. A coluna `ee_type` pode adquirir os valores `class` ou `equipment`, sendo que quando estamos a falar de um equipamento a coluna `ee_class` indicará a que classe de equipamentos este pertence.

A `equip_property` é onde estarão guardadas todas as características físicas dos equipamentos adicionados na tabela `equip_element`. Estas características podem ser por exemplo a capacidade dos tanques ou mesmo a temperatura de trabalho dos reatores.

A `equip_phases` é a biblioteca de fases que estão previamente configuradas nos controladores e que estão disponíveis para serem utilizadas nas receitas que se pretende criar. Correspondente a cada fase também existem os parâmetros necessários à execução da mesma e cada parâmetro pode ter ou não unidades de medida associadas. Cada parâmetro pode ser de um de três tipos diferentes: `Process Variable`, `Input` ou `Output`.

Na `equip_interface` surgem as fases atribuídas aos equipamentos capazes de as executar.

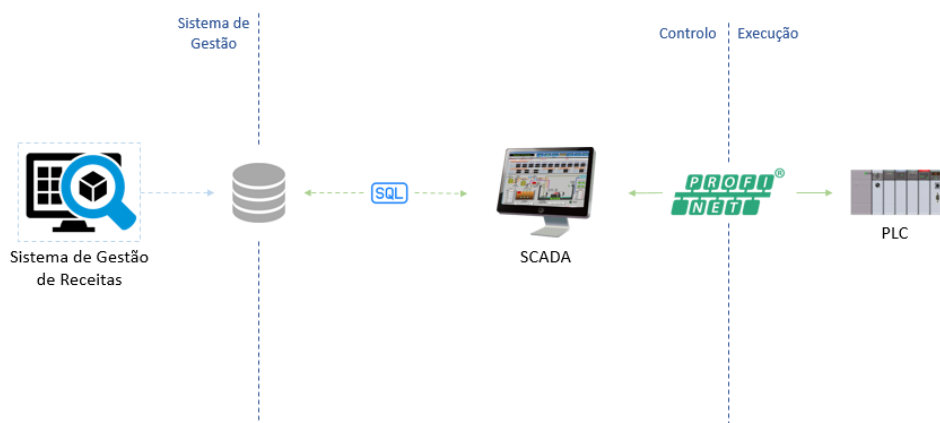


Figura 3.3: Transferência de informação entre Sistema de Gestão de Receitas e controlador

A `equip_interface_parameter` permite o armazenamento dos parâmetros das fases que foram previamente adicionadas aos equipamentos já configurados no sistema.

A `equip_links` serve para se documentar as ligações físicas existentes entre os diversos equipamentos localizados na fábrica e que foram inicialmente configurados e adicionados ao sistema. Estes dados servem para validação do processo de criação das receitas para que não se crie receitas incapazes de ser executadas.

Por fim surge a `units_measure` cujo objetivo passa por ser uma biblioteca das unidades de medida disponíveis para qualquer parte do sistema que precise deste tipo de dados.

3.2.3 Receitas

Referente ao conjunto de tabelas com a responsabilidade de manter o registo de todas as informações das receitas temos as seguintes tabelas: a `m_recipe_element`, a `m_recipe_element_parameter`, a `m_recipe_element equip`, a `m_recipe_step`, a `m_recipe_step_parameter`, a `m_recipe_link` e a `m_recipe_approvals`. É utilizada ainda informação da tabela `units_measure` que provem do grupo de tabelas dos equipamentos. O conteúdo de cada uma delas pode ser consultado na Figura 3.7 da página 34.

A tabela `m_recipe_element` serve para manter registo do *header* de uma receita, dos *Unit Procedures* e *Operations* que possam existir das diversas receitas criadas, das *Phases* que possam ser usadas nas receitas e também dos símbolos que podem ser utilizados na construção de uma receita. Cada receita pode ter um de cinco tipos de status: *Approved for Production*, *Approved for Test*, *Not Approved*, *Inactive* ou *Obsolete*. Esta tabela pode ter *recipe elements* dos seguintes tipos: *Master Recipe*, *Procedure*, *Unit Procedure*, *Operation*, *Phase*, *Begin*, *End*, *Start Parallel*, *End Parallel*, *Start Branch* ou *End Branch*. Há a possibilidade de adicionar informação relativa às quantidades máximas, mínimas ou típicas que o lote que é produzido pela receita pode ter. Essa quantidade poderá ter unidades de medida associadas.

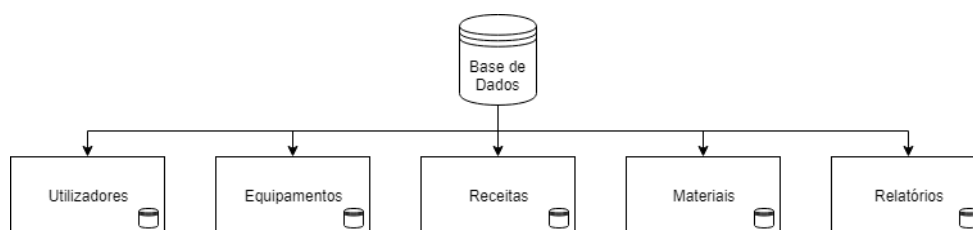


Figura 3.4: Principais grupos de dados na Base de Dados

A tabela `m_recipe_element_parameter` permite manter um registo da informação de todos os parâmetros das fases pré-configuradas e que podem ser escolhidas posteriormente para os passos das receitas a criar. Isto é, uma fase que tenha por exemplo dois parâmetros para se preencher, podemos querer que um desses parâmetro seja fixo e que na construção da receita só tenhamos de preencher o que falta. Assim podemos guardar essa fase com o devido parâmetro preenchido permitindo assim ser chamada na construção da receita.

Na tabela `m_recipe_element equip` é onde estão guardados todos os requisitos de equipamentos que uma receita tem. Isto permite filtrar depois as receitas que podem ser escolhidas para produção consoante os equipamentos que temos disponíveis e permite também filtrar as *Phases*, *Operations* ou *Unit Procedures* passíveis de serem escolhidas na criação de uma receita nova.

Uma tabela extremamente importante é a `m_recipe_step`. Esta contém todos os passos existentes numa receita seja do nível *Unit Procedure*, *Operations* ou *Phases*. Cada passo terá associado a ele um *recipe element* e a uma *Master Recipe* existentes na tabela `m_recipe_element`. Presente nesta tabela estão dados importante para a criação gráfica de toda a receita na interface do sistema a criar, sendo que as colunas específicas para esse objetivo são a `href`, a `back_color`, a `tree_priority` e a `parent_id`.

De maneira a complementar a tabela `m_recipe_step` surge a `m_recipe_step_parameter` que é responsável por reter a informação relativa aos parâmetros das fases escolhidas para alguns dos passos das receitas criadas. Estes parâmetros podem ter associadas unidades de medida.

Responsável pela ligação entre dos passos existentes nas receitas está a tabela `m_recipe_link`.

Por fim surge a tabela `m_recipe_approvals`, responsável por manter o registo de todas as aprovações ou reprovação das receitas existentes no sistema. Esta tabela permite assim manter um controlo sobre os estados das diversas receitas e quem foi responsável por elas.

Ainda terá de existir outra tabela cujo único objetivo é guardar as receitas de modo a que estas possam ser interpretadas pelas consolas SCADA de maneira a que consigam passar a informação pretendida para os controladores corretamente. Esta tabela não foi projetada visto a empresa ainda não saber exatamente como pretende que esta informação esteja guardada.

3.2.4 Materiais

Na parte responsável pelo armazenamento da informação dos materiais disponíveis na fábrica temos três tabelas distintas: a `materials`, a `materials_charac` e a `materials_loc`. O conteúdo de

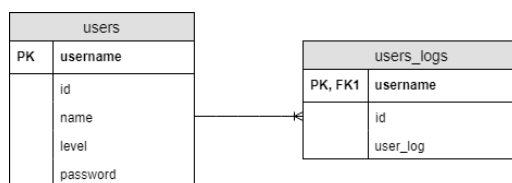


Figura 3.5: Modelo Relacional das Tabelas dos Utilizadores

cada uma delas pode ser consultado na Figura 3.8 da página 35. São ainda usadas informações constantes em duas tabelas do grupo de tabelas dos equipamentos, sendo elas a `equip_element` e a `units_measure`.

A tabela `materials` serve como biblioteca de todos os materiais que possam ter sido utilizados ou que possam vir a ser utilizados pelo sistema nas receitas. Cada elemento desta tabela pode ser de um de seis tipos: `Ingredients`, `Intermediates`, `Finished Goods`, `By-Products` ou `Other`.

De maneira a caracterizar estes materiais surge a tabela `materials_charac` onde são indicadas algumas características que distingam o material por si só ou mesmo que distingam o material de outros semelhantes ou do mesmo tipo. Alguns exemplos dessas características podem ser a viscosidade dos materiais, a temperatura a que devem ser trabalhados, etc. Cada característica pode ter uma unidade de medida associada.

Já a tabela `materials_loc` tem o objetivo de indicar que material está em cada equipamento que possa ter algum tipo de material armazenado. De salientar que em cada equipamento, ou localização, só podemos ter um material associado.

3.2.5 Relatórios

Visto ainda não ter acesso à informação que irá surgir da execução das receitas, não foi feito qualquer planeamento quanto à estrutura dos dados necessária para fazer o armazenamento dos mesmos de modo a que depois o sistema possa fazer os relatórios da execução das receitas. De qualquer modo o sistema está idealizado a haver uma parte da base de dados dedicada e orientada apenas a esse propósito.

3.3 Interface

Além de toda a parte do sistema à qual o utilizador não tem acesso (estrutura de dados, interação entre as diversas partes constituintes do sistema), e que é muito importante, há outra parte que também o é, a interface do sistema com o utilizador. Esta interface deve permitir ao utilizador usufruir de todas as funcionalidades de que o sistema dispõe e garantir uma interação o mais simples e intuitiva possível de modo a permitir uma aprendizagem e habituação do software rápida e também de modo a reduzir a probabilidade de gerar informação errada no sistema.

Assim este sistema tem oito grupos de interfaces que podemos considerar como sendo as principais: a página de login, o menu que permite acesso às diversas páginas do sistema, as páginas

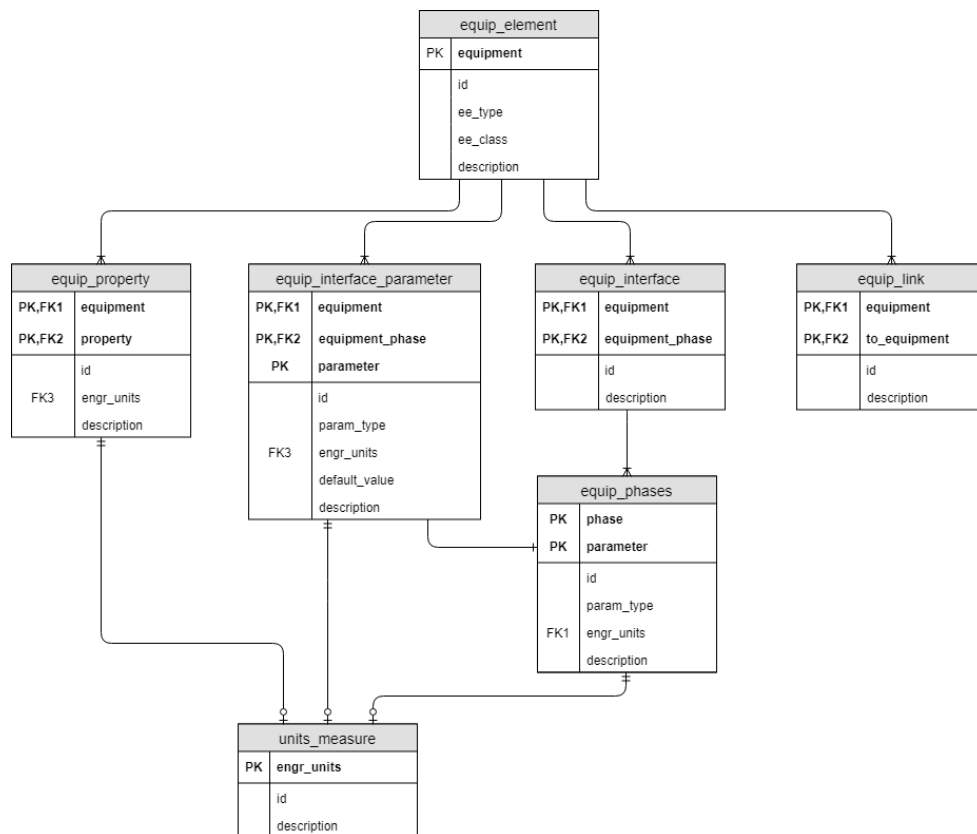


Figura 3.6: Modelo Relacional das Tabelas dos Equipamentos

referentes à configuração dos equipamentos da fábrica, as páginas relacionadas com os materiais que a empresa dispõe, as páginas que permitem uma gestão e configuração das receitas, as páginas que permitem um acesso a informação e relatórios gerados pela produção de lotes, as páginas que permitem a um administrador fazer uma gestão dos utilizadores do sistema e ainda as páginas de ajuda onde se pode encontrar informação que se considere relevante à orientação dos utilizadores na utilização do sistema.

3.3.1 Login

Logo que o utilizador tente aceder ao sistema, caso não tenha a sua sessão iniciada este não conseguirá exercer qualquer ação sobre o sistema. Assim é necessário que cada utilizador tenha a sua conta de acesso específica e que introduza essa informação de autenticação numa página semelhante à demonstrada na Figura 3.9 da página 36. Caso os dados sejam mal introduzidos é expectável que a página forneça informação de volta tal como por exemplo "Password errada" ou "Utilizador introduzido não existe". Não é expectável que exista a opção de criar contas nesta página devido à importância do sistema em si e ao que este está destinado. Assim essa função aparecerá dentro do próprio sistema quando autenticado por uma conta de alguém com permissões administrativas e essa pessoa é que poderá criar contas para os seus funcionários.

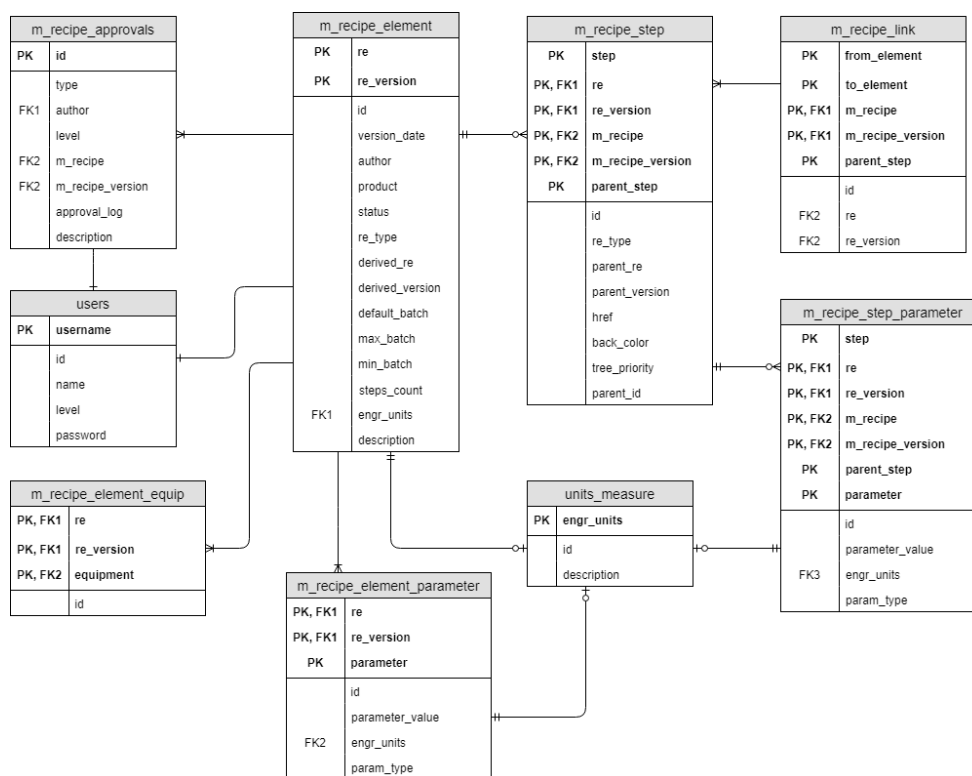


Figura 3.7: Modelo Relacional das Tabelas das Receitas

3.3.2 Menu

De maneira a permitir uma navegação entre as principais áreas do sistema, quando se inicia com sucesso a sessão no sistema a página que faz mais sentido que aparece é a do menu, ou seja, uma página onde à distância de apenas um clique podemos ir para a secção do sistema que pretendemos. De maneira a que esta navegação seja possível mesmo enquanto se está noutras páginas, há um menu lateral que em caso de o utilizador querer mudar de secção permite-o fazer isso sem ter de ir à página do menu falada anteriormente. Tanto a página do menu como o menu lateral oculto são expectáveis que sejam algo semelhante ao demonstrado na Figura 3.10 da página 37.

3.3.3 Equipamentos

Na secção do sistema dedicada à configuração dos equipamentos há uma divisão em 5 grandes subáreas: a Units Classes, a Units, a Links, a Phases e a Units of Measure. A página inicial desta secção poderá ser algo semelhante ao verificado na Figura 3.11 da página 38 onde se pode ver a divisão que acabámos de falar.

Na parte das Units Classes podemos adicionar, editar e apagar famílias de equipamentos que existam na fábrica. Depois em cada uma delas podemos fazer a configuração dos seus atributos e das fases que estas são capazes de executar. Esta atribuição de fases só é possível para fases que já existam previamente na biblioteca de fases configuradas nos controladores e disponíveis para

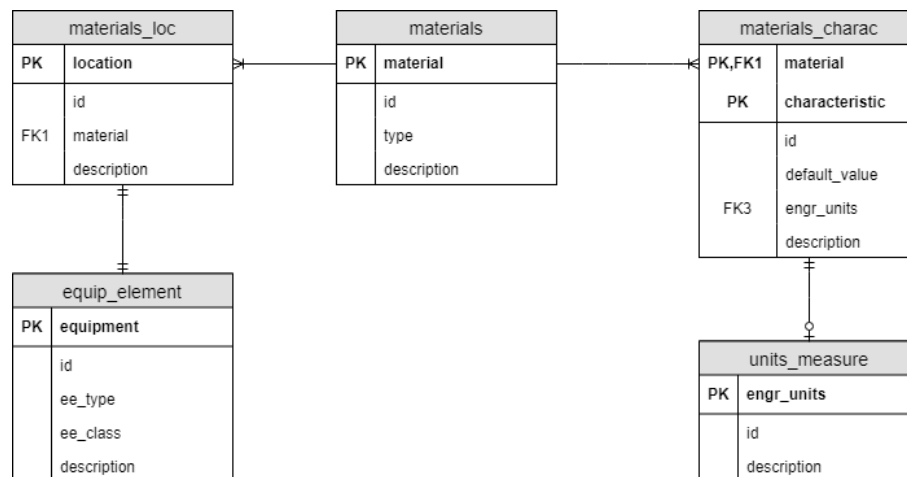


Figura 3.8: Modelo Relacional das Tabelas dos Materiais

utilização pelo sistema. Quando dentro da página das fases de um determinado equipamento, há ainda a possibilidade de configurar os parâmetros das mesmas de modo a que quando sejam chamadas na criação das receitas, já existam valores preenchidos por defeito caso assim se pretenda. Durante todos estes processos de configuração é de esperar que existam notificações do sistema para o utilizador a avisá-lo de eventuais erros, sucessos ou outra informação pertinente originária das operações.

Já na subsecção das Units é possível adicionar os equipamentos existentes na fábrica. Caso se atribua o equipamento a alguma família de equipamentos já criada no sistema, é de esperar que este adquira todos os atributos e fases dessa família. Caso esta atribuição não seja feita, é feito todo um processo de configuração em tudo semelhante à configuração das famílias de equipamentos onde serão configurados os atributos do equipamento, as suas fases e os parâmetros das mesmas.

Na parte Links é feita a configuração das ligações físicas existentes entre os diversos equipamentos introduzidos no sistema, na parte Phases é onde a biblioteca de fases configuradas nos controladores pode ser criada e editada e na parte Units of Measure é onde as unidades de medida que podem ser utilizadas nos atributos dos equipamentos ou nos parâmetros das fases podem ser criadas e geridas.

3.3.4 Materiais

Na parte do sistema dedicada aos materiais existem duas secções distintas: a Materials e a Location. Um exemplo de como a página poderia ser encontrada na Figura 3.12 da página 39.

A secção Materials permite adicionar todos os materiais que o utilizador entender e permite ainda aceder à página de características desse material permitindo assim adicionar informação que distinga esse material em questão dos restantes.



Figura 3.9: Mockup da Página de Login

A secção Location permite indicar que material está em cada equipamento que seja capaz de servir de armazenamento para este tipo de produto. Cada equipamento só pode ter um material associado. Havendo alterações é nesta secção que o utilizador deve introduzir essa informação no sistema de modo a que os controladores depois consigam saber onde ir buscar os materiais que necessitam para a produção.

3.3.5 Receitas

Uma das partes principais deste sistema é a das receitas. Nesta existe uma página inicial onde é possível ver todos os *headers* das receitas criadas e criar novos também. Além disso é permitido ao utilizador criar novas versões ou então cópias de receitas já existentes. É ainda possível passar para a página de aprovações de uma receita em específico selecionando a receita que se pretende ou mesmo ir para a página Phases onde o utilizador pode criar um conjunto de fases pré-configuradas que poderão ser chamadas na criação das receitas. Um exemplo da maneira como esta informação seria mostrada na primeira página pode ser visto na Figura 3.13 da página 40.

Quando o utilizador escolher a receita que pretende criar passará para a parte de edição da receita. Nessa parte surgem três subáreas: a Equipment Requirments, a Procedure e a Formula.

Na parte dos Equipment Requirments o utilizador pode dizer quais os equipamentos que irá precisar para a execução da receita que pretende criar. Desta maneira, no processo de criação da receita e dos seus passos, apenas poderá escolher fases ou outros elementos que seja capazes de ser executados pelos equipamentos atribuídos e selecionados.

Na parte Procedure o objetivo passa por ser onde o utilizador configura todos os passos pelos quais a receita terá de passar. Nesta parte pode desenhar toda a execução da receita desde o nível das *Unit Procedure* até ao nível das *Phases*. É permitido ainda configurar os parâmetros das fases escolhidas para alguns dos passos e fazer a ligação entre os diversos passos criados. Um exemplo de como seria um ambiente deste género pode ser consultado na Figura 3.14 da

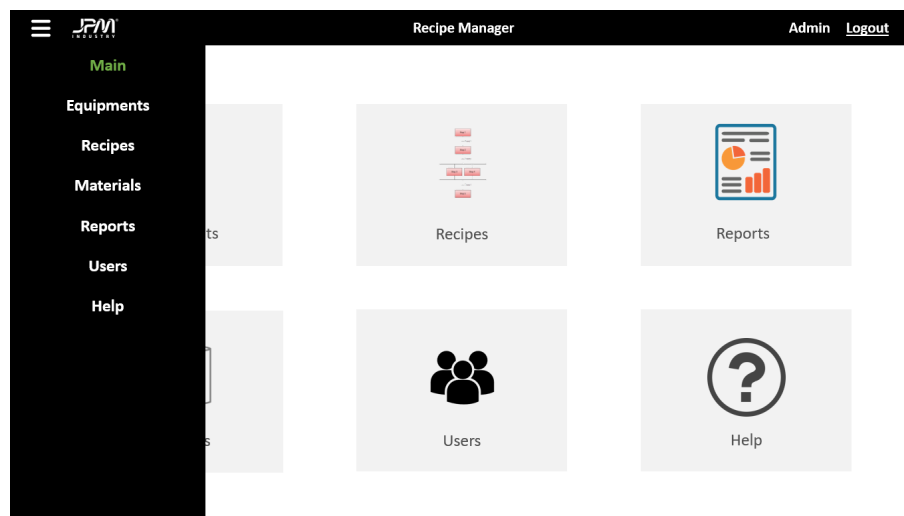


Figura 3.10: Mockup da Página do Menu

página 41. É esperado que o sistema vá dando algumas ajudas visuais conforme o processo vai sendo criado, como por exemplo sinalizar a vermelho as fases que ainda não tenham os valores dos seus parâmetros preenchidos.

Já na parte Formula é esperado que apareça um resumo de todos os *inputs* necessários à execução da receita em questão e todas as variáveis de processo e respetivos valores também elas associadas à receita a ser criada ou editada.

3.3.6 Relatórios

Na parte do sistema dedicada ao relatórios é onde o utilizador pode ter acesso aos relatórios gerados automaticamente aquando a execução de uma receita e produção de um lote. Nestes relatórios estarão informações como alarmes que ocorreram durante a produção, tempos associados à mesma, quantidades de matérias primas consumidas, quantidade de produto final produzida, etc. Caso haja interesse para tal, é nesta parte que podem surgir também alguns KPI relacionados com a produção dos lotes. Um exemplo de uma página deste género seria semelhante ao encontrado na Figura 3.15 da página 42.

3.3.7 Utilizadores

Na página dedicada à gestão dos utilizadores, o utilizador que tenha acesso com permissões administrativas sobre o sistema, pode consultar os acessos ao sistema dos diversos funcionários, criar, editar e eliminar contas e ainda mudar o tipo de permissão de acesso ao sistema.

3.3.8 Ajuda

Na página de ajuda há um guião que respeita a mesma divisão que as páginas do próprio sistema segue. Assim caso haja dúvidas enquanto ao funcionamento de alguma parte do sistema o

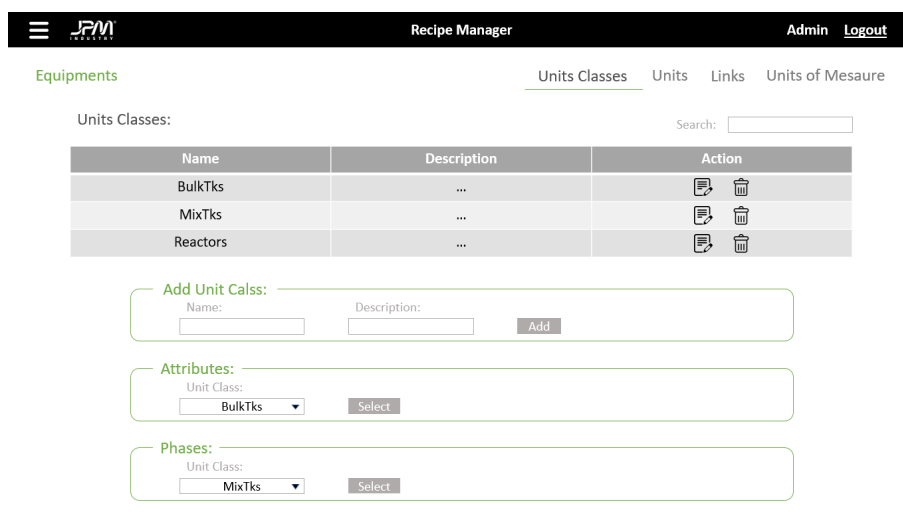


Figura 3.11: Mockup da Página de Equipamentos

utilizador pode encontrar algum tipo de instruções e ajuda nesta página. Pode encontrar informações como por exemplo quanto ao processo a seguir na configuração de equipamentos, sobre a criação de receitas e sobre o tipo de variáveis e as opções existentes nas mesmas.

3.4 Software de Terceiros

Hoje em dia já existe um grande quantidade de software, plugins ou bibliotecas criadas de maneira a que o processo de criação de algum tipo de sistema ou website seja mais simples, mais rápido e mais intuitivo. Dessa forma para a criação deste sistema a sugestão passa por usar algumas destas ferramentas já disponíveis de maneira a agilizar todo o processo de desenvolvimento. Assim as ferramentas tidas em consideração são as seguintes.

3.4.1 Smarty

Tal como descrito na página oficial do Smarty [19], este é um sistema que permite a criação de *templates* para PHP, ou seja, permite separar o desenvolvimento da parte lógica do sistema da parte da interface do mesmo. Como hoje em dia há uma distinção clara da função do designer que é responsável pelo desenvolvimento dos ficheiros HTML e CSS, para a do programador que é responsável pela parte lógica do sistema, o Smarty permite facilitar a implementação do trabalho de cada uma destas partes.

Para criar as páginas o Smarty lê os ficheiros dos *templates* que estão associados àquela página e criar os *scripts* PHP a partir deles. Os *templates* só são compilados uma vez ficando em memória para que sempre que a página seja chamada de novo o processo de leitura da mesma seja mais rápido. Os *templates* só voltam a ser compilados caso seja feita alguma alteração aos mesmos.

O uso do Smarty permite que o designer não necessite de ter conhecimentos em PHP facilitando assim a sua tarefa e como ele não precisa de mexer no código PHP não há a hipótese deste

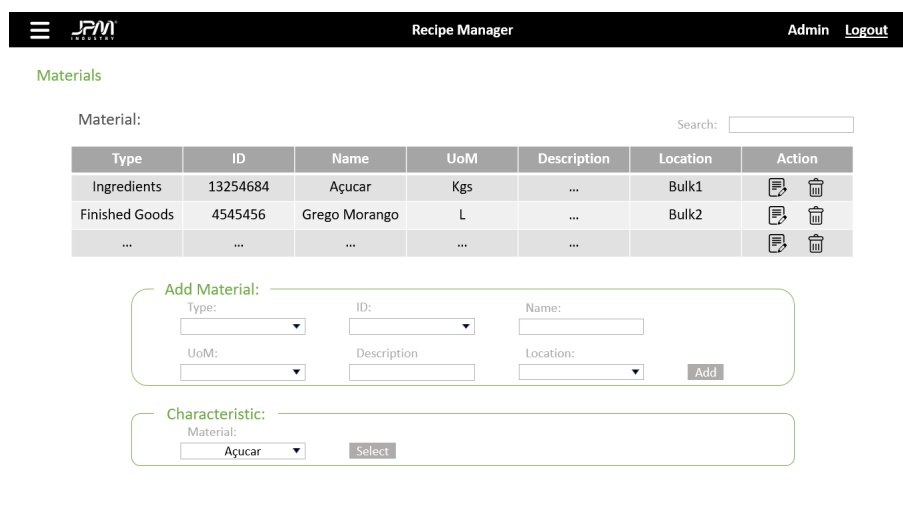


Figura 3.12: Mockup da Página de Materiais

fazer alterações que danifiquem ao código realizado pelas pessoas responsáveis pela parte lógica do sistema.

O Smarty é uma ferramenta open-source.

3.4.2 jQuery

A jQuery é uma biblioteca de JS cujo objetivo é simplificar a utilização de JS nas páginas web do sistema que se pretender desenvolver. Para isso ela é composta por métodos onde o utilizador insere os dados necessários à execução reduzindo assim a complexidade que iria haver caso tivesse de desenvolver o código JS todo para que o sistema fizesse o que pretendesse.

A biblioteca jQuery permite manipular os ficheiros CSS presentes no website, permite manipular o ficheiro HTML da página a mostrar no navegador web, permite detetar eventos na utilização da página, permite gerar efeitos e animações e ainda fazer pedidos *Asynchronous Javascript and XML* (AJAX) ao servidor.

3.4.2.1 AJAX

Os pedidos AJAX permitem que haja troca de informação entre o servidor e o cliente sem que haja alterações à página que está a ser mostrada no navegador web do cliente. Isto permite também que com essa troca de informação é possível atualizar partes específicas da página alterando o documento HTML sem que seja preciso fazer um novo carregamento da página. o AJAX pode utilizar ficheiros XML ou mesmo ficheiros JSON para transportar a informação entre o servidor e o cliente.

Recipe Manager Admin Logout

Recipes List Approvals Phases

Recipes: Search:

Name	Product	Descriptio	Version	Type	Status	DefBatSiz	MaxBatSiz	MinBatSiz	UoM	Action
Beer		...	V1.0	Actual	Inactive	100	110	90	L	
BeerTpl		...		Percent		100			L	
Red Oak	Premium Beer	...	V10.3							

New Recipe:

Recipe: Product: Description:

Type: Default Batch Size: Maximum Batch Size:

Minimum Batch Size: UoM:

Edit Recipe:

Recipe:

Copy:

Original recipe: New Recipe:

Figura 3.13: Mockup da Página de Receitas

3.4.2.2 JSON

A simplicidade do JSON permitiu que este fosse ganhando popularidade e fosse tido como uma alternativa para o tipo de ficheiros XML quando se utiliza AJAX.

O JSON é um formato que permite a troca de informações entre sistemas independente de linguagem de programação que se utiliza e que permite ainda fazer o armazenamento dados. Este tipo de ficheiros é basicamente um ficheiro texto em que se utiliza notação utilizada em objetos JS. Assim utilizando este formato de dados conseguimos fazer a troca de informação entre cliente e servidor transformando objetos JS em JSON e depois fazendo o processo inverso.

3.4.3 W3.CSS e Bootstrap CSS

Tanto a W3.CSS como a Bootstrap CSS são duas *frameworks* para CSS que permitem ao programador e designer das páginas web simplificar todo o processo de desenho da interface do sistema. Isto acontece pois elas possuem já uma extensa biblioteca de classes onde vários estilos já estão pré-configurados tanto a nível de cor, tamanho, posição e apresentação dos vários elementos constituintes da página permitindo assim ao designer atribuir estilos rapidamente aos elementos que ele pretender. Além disso estas *frameworks* já possuem ferramentas que adaptam o design das páginas tanto ao navegador web em que estão a ser abertas bem como, se o utilizador assim o pretender, em que tipo de dispositivo estão a ser abertas permitindo assim uma adaptação tanto do conteúdo como a maneira que este é mostrado no dispositivo em que está a ser consultado.

3.4.4 jQuery Bootgrid

A jQuery Bootgrid é uma interface para o utilizador criada para jQuery e Bootstrap. Esta permite mostrar um conjunto de dados em formato de tabela e permitir interações com os mesmo de modo a tornar a informação dinâmica. É possível criar paginação quando a quantidade de

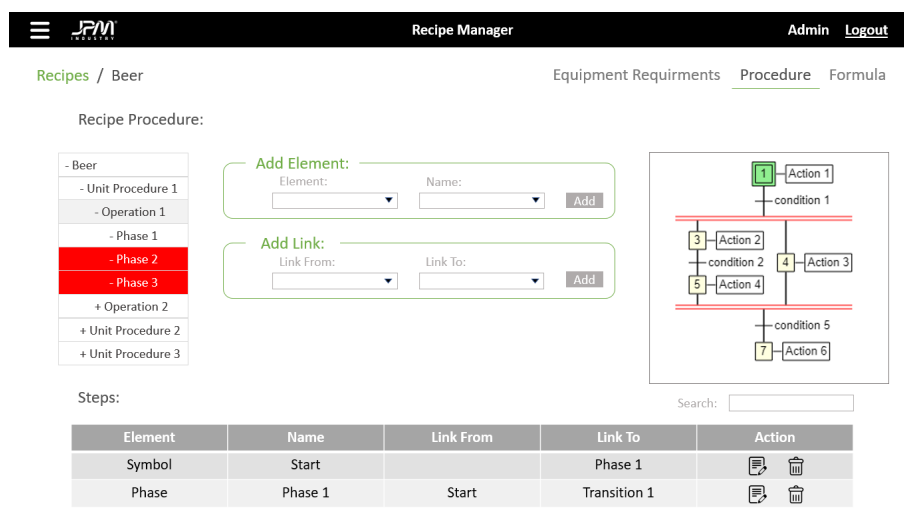


Figura 3.14: Mockup da Página do Editor de Receitas

informação passa o limite da informação que pode ser mostrada, permite filtrar os dados a mostrar e ainda fazer pesquisa sobre os dados presentes na tabela criada.

Este tipo de ferramenta pode ser extremamente útil para mostrar informação importante do sistema como por exemplo informação sobre os equipamentos ou sobre as receitas ou para representar qualquer uma das tabelas que se ache relevante existirem no sistema e que devam ser interativas e dinâmicas.

3.4.5 Bootstrap-select

O Bootstrap-select é um plugin jQuery que utiliza o Bootstrap dropdown.js de maneira a adicionar funcionalidades extra aos elementos do tipo *select* dos ficheiros HTML. Duas dessas funcionalidades extra que por exemplo são úteis para um sistema deste tipo são o facto de permitir seleccionar mais que um elemento e permite também a criação de grupos na mesma lista de selecção.

3.4.6 Bootstrap Notify e Bootbox.js

Num sistema deste género há pelos menos dois tipos de notificações que são importantes, o que tem como objetivo apenas informar o utilizador de alguma coisa que aconteceu e o que além de informar carece de algum tipo de confirmação do utilizador sobre o evento que acabou de ocorrer.

A Bootstrap Notify é uma biblioteca JS que possui métodos capazes de criar notificações que vão de encontro ao primeiro tipo de notificações falado, ou seja, do tipo que não carece de interação por parte do utilizador. Este utiliza o estilo gráfico do Bootstrap CSS para criar as notificações e permite personalizar a notificação ao gosto do programador.

Já para o segundo tipo de notificações que foi falado a Bootbox.js, que também ela é uma biblioteca JS, é capaz de fazer o que se pretende. Isto é, ela permite criar janelas de notificações em que é permitido introduzir elementos que permitam a interação do utilizador com o processo de

Batch ID	Product	Batch Size	Date	Report
125468	Beer_Black	150	19/03/2018 14:55:32	BatchID.pdf
...
...

Figura 3.15: Mockup da Página de Relatórios

alerta através de um simples clique num botão ou mesmo com a introdução de dados necessários devido ao evento que acabou de ocorrer.

3.4.7 Bootstrap Tree View

Uma ferramenta útil para criar o menu lateral onde é possível visualizar os diversos passos da receita é o plugin Bootstrap Tree View. Este permite criar um menu em árvore onde podemos demonstrar hierarquicamente o processo da receita criada. Este plugin já possui um grande conjunto de métodos que facilitam a sua utilização e pode ser personalizado sendo assim possível adaptar o seu aspeto à funcionalidade que o programador pretender.

3.4.8 GoJS

Devido à especificidade da norma ISA88 quanto à linguagem gráfica a utilizar na criação da receita, não foi possível encontrar ferramentas que permitam a criação gráfica deste processo numa página web. Foi sim encontrada uma ferramenta que permite utilizar a linguagem *GRAPHe de Commande Etape-Transition* (GRAF CET) que é em tudo semelhante à linguagem que a norma especifica e que é comumente utilizada por softwares disponíveis no mercado que permitam a criação de receitas. Assim a sugestão passa por utilizar a biblioteca JS GoJS que permite a criação de diagramas interativos em páginas HTML. Esta biblioteca tem o estilo GRAFCET entre os seus tipos de diagramas disponíveis. Este diagrama permite ainda guardar e ler os dados produzidos no mesmo através de ficheiros JSON.

3.4.9 FPDF

De forma a gerar os ficheiros *Portable Document File* (PDF) dos relatórios de produção é preciso ter alguma ferramenta complementar ao sistema que seja responsável por esse processo.

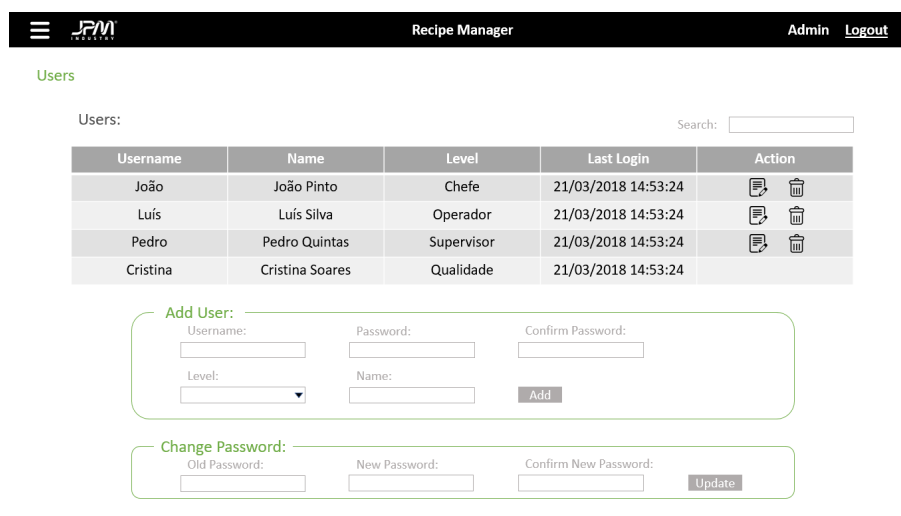


Figura 3.16: Mockup da Página de Utilizadores

A FPDF permite fazer isso. Ela é uma ferramenta gratuita que permite criar os ficheiros PDF que se pretende usando apenas código PHP. Assim é fácil de integrar esta ferramenta no resto de sistema pois ela está orientada também ao desenvolvimento web. Ela permite a criação de relatórios com todo o tipo de dados que possam ser relevantes a relatórios que este sistema possa gerar, tais como tabelas, imagens e ainda gerar gráficos. Tudo isto com informação contida na base de dados e acedida com o propósito da criação do relatório. O relatório depois de produzido é armazenado no diretório do sistema na pasta destinada a ele.

3.5 Resumo

De modo a criar um sistema o mais modular possível foi feita uma divisão do sistema em cinco grandes áreas: a dos Utilizadores, a dos Equipamentos, a das Receitas, a dos Materiais e por fim a dos Relatórios. Cada uma delas é responsável por funções específicas do sistema. A dos Utilizadores serve para gerir as contas de acesso ao sistema, a dos Equipamentos permite configurar os equipamentos da fábrica, a das Receitas permite gerir e criar receitas destinadas à produção, a dos Materiais permite fazer uma gestão do material capaz de ser escolhido nas receitas e a dos Relatórios permite a consulta de informação gerada pela produção dos lotes. A estrutura da base de dados seguiu também esta orientação quanto à sua divisão.

A criação do sistema pode ser simplificada por ferramentas já existentes desenvolvidas por outras pessoas agilizando assim todo o processo de desenvolvimento. Para ajudar a separar o design gráfico da parte lógica do sistema aconselha-se o uso do Smarty permitindo assim que o design do sistema possa ser desenvolvido à parte usando, por exemplo, as bibliotecas CSS W3.CSS e Bootstrap. Para tornar visível toda a informação pretendida nas tabelas das diversas páginas do sistema onde a informação é dinâmica é aconselhado o uso do plugin jQuery Bootgrid pois este permite a criação das tabelas e navegação nas mesmas de forma simples. Como em qualquer

sistema há necessidade de gerar notificações para o utilizador quando é preciso informa-lo de algo que ocorreu no sistema. Assim, também de maneira a simplificar a aplicação de algo deste género aconselha-se o uso das bibliotecas Bootstrap Notify e Bootstrap.js. Para a página da criação da receita existem duas bibliotecas que serão com certeza importantes para a sua implementação, a Bootstrap Tree View que permite implementar o menu lateral representativo do processo da receita e a GoJS que permite a utilização de diagramas em páginas web. Por fim para gerar os relatórios é aconselhado a ferramenta FPDF pois além de ser gratuita pode ser implementada diretamente num sistema deste género.

Capítulo 4

Implementação e Resultados

Neste capítulo é apresentada a parte do sistema que foi implementada de modo a poder haver uma prova de conceito. São também apresentadas as ferramentas utilizadas para que fosse possível implementar a solução proposta e ainda um conjunto de testes ao seu funcionamento e respetivos comentários aos resultados obtidos.

4.1 Softwares Utilizados

De maneira a ser possível implementar um sistema deste género é preciso ter uma base de dados e um servidor HTTP capaz de correr PHP e de armazenar todos os ficheiros e páginas do sistema em si. Assim sendo foi utilizada uma biblioteca de instaladores conhecida como Bitnami que está orientada a aplicações web. Mais concretamente o conjunto de instaladores denominado de Bitnami WAPP que está especificamente orientado ao desenvolvimento de aplicações web em Windows e que usem PHP. Este instala as versões mais recentes do Apache, PostgreSQL, PHP, phpPgAdmin e ainda permite a instalação de *frameworks* como a Smarty que é a que foi aconselhada e a que se pretende usar neste sistema.

O Apache é um servidor HTTP open-source para sistema UNIX e Windows. É neste servidor que irão estar alojadas as diversas páginas do sistema e os ficheiros relevantes ao mesmo. A PostgreSQL é uma base de dados também ela open-source. Esta tenta seguir ao máximo os standards existentes quanto ao SQL, mas como qualquer outra base de dados possui funções e alguma linguagem específica. Já o phpPgAdmin é uma ferramenta executada no navegador web que permite fazer a gestão de bases de dados PostgreSQL. Permite interagir com a base de dados de duas maneiras, uma delas através da execução direta de código SQL e a outra é através da interface que esta dispõe que permite fazer a configuração das tabelas, a introdução de dados e edição dos mesmos através de uma interface simples e mais intuitiva para quem esteja menos à vontade com o SQL.

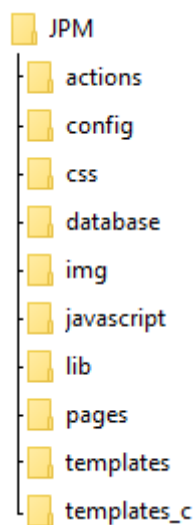


Figura 4.1: Estrutura de Pastas do Sistema Implementado

4.2 Sistema Implementado

De modo a ser seguido o objetivo de ter um sistema modular foi feita uma estruturação inicial das pastas do sistema de maneira a que o conteúdo de cada uma delas esteja orientado a uma área ou função do sistema. Assim, conforme se pode verificar na Figura 4.1, a raiz do nosso sistema tem as pastas actions, config, css, database, img, javascript, lib, pages, templates e templates_c. Nas pastas actions, javascript, pages e templates surgem ainda quatro sub-pastas específicas sendo elas equipments, materials, recipes e users. Adicionalmente nas pastas pages e templates temos ainda uma sub-pasta common.

A pasta action destina-se aos ficheiros PHP responsáveis por executar qualquer ação do sistema como por exemplo ir buscar dados ou executar ações sobre a base dados. Cada sub-pasta serve depois para criar uma divisão dos ficheiros existentes de forma a orientar cada ficheiro criado à secção do sistema a que está ligado.

A pasta config serve para albergar todos os ficheiros que possam existir de configuração do sistema como por exemplo os dados da conta de acesso à base de dados, o caminho para o diretório da raiz do sistema ou mesmo o URL base do sistema. Isto permite que quando há alterações em algum deste campos esta informação esteja separada da parte lógica do sistema não obrigando o programador a alterar todos os ficheiros onde por exemplo há uma chamada à base de dados e teríamos de inserir constantemente os dados de acesso.

A pasta css tem o propósito de albergar todos os ficheiros CSS que possam existir no sistema e que sejam responsáveis pelo design da interface do mesmo. Aqui irão estar os ficheiros CSS que podem ser desenvolvidos pelo programador ou então que são usados nas bibliotecas de terceiros que se possa vir a utilizar.

Na pasta img, tal como o nome indica, estarão todas as imagens que serão utilizadas no sistema.

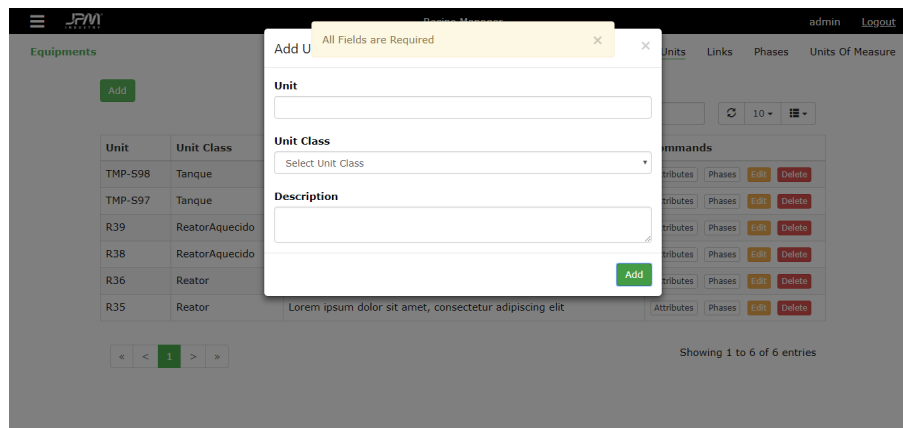


Figura 4.2: Janela de Inserção de Dados e Janela de Alertas Implementadas

A pasta javascript tem como objetivo albergar os ficheiros JS que sejam utilizados no sistema. Isto inclui tanto os ficheiros JQuery ou outras bibliotecas JavaScript de terceiros como também ficheiros deste tipo mas desenvolvidos pelo programador.

Na pasta lib estarão todas as ferramentas e software de terceiros que são utilizados no sistema. Neste caso é aqui que estará alocado o Smarty que irá ser responsável pela criação das páginas finais do sistema a mostrar no navegador web do cliente.

A pasta pages destina-se aos ficheiros PHP responsáveis pelas partes dinâmicas das diversas páginas do sistema. Nestes ficheiros está a informação necessária para que usando os *templates* que tem à sua disposição, o Smarty possa construir as páginas pretendidas. Cada sub-pasta serve para orientar os ficheiros à sua parte do sistema sendo que a sub-pasta common tem o objetivo em especial de conter todos os ficheiros que sejam responsáveis por partes das páginas que são comuns a todas elas como por exemplo o *header*, o *footer* ou mesmo o menu.

Na pasta templates estão todos os ficheiros TPL que contêm o design da página e o seu conteúdo estático. isto é, estes ficheiros é que são responsáveis pela maneira como a informação é mostrada na página do sistema. Com estes ficheiros e a informação proveniente dos ficheiros PHP da pasta pages, o Smarty constrói depois as páginas do sistema.

Por fim, a pasta templates_c serve para armazenar todas as páginas que o Smarty compila quando junta a informação dos ficheiros PHP da pasta pages com a informação dos ficheiros TPL da pasta templates. Isto permite que quando a informação não muda, o Smarty não tenha de recompilar as páginas de novo permitindo assim tornar o processo de leitura da página mais rápido pois a página já está criada.

Para introdução de dados nas diversas páginas do sistema foram usados formulários disponíveis no Bootstrap CSS e para as notificações do sistema foram usadas as bibliotecas Bootstrap Notify e Bootbox.js. Um exemplo do formulário e de uma notificação usando o Bootstrap Notify pode ser encontrado na Figura 4.2. Já para a representação dos dados nas tabelas das diversas páginas foi usado o plugin jQuery Bootgrid.

Na implementação do sistema foram usadas variáveis de sessão pois estas permitem armazenar

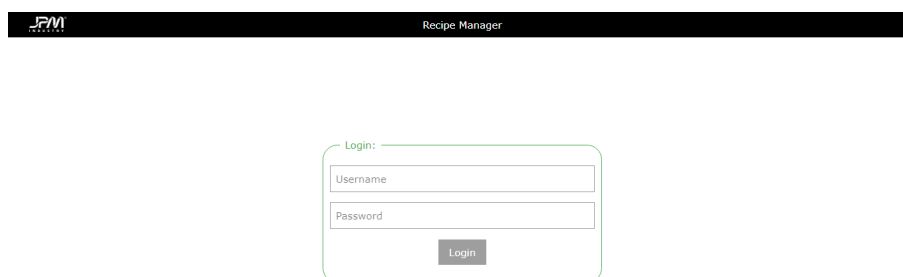


Figura 4.3: Página de Login Implementada

informações relevantes do sistema, como configurações ou informações específicas do utilizador, estando estas unicamente ligadas à sessão que foi aberta no servidor para aquele utilizador. Sempre que o utilizador encerra a sua sessão esta informação é eliminada do servidor.

4.2.1 Login

No primeiro acesso e no arranque do sistema, este barra o acesso do utilizador caso ele não tenha a sua sessão iniciada. Assim é pedido que o utilizador coloque os seus dados de acesso nos respetivos sítios, ou seja, o username e a sua password, e caso esta informação esteja correta aí sim o sistema garante o acesso do utilizador ao interior do sistema. Caso esta informação não seja introduzida corretamente o utilizador é notificado de que os dados introduzidos estão incorretos. A página de acesso implementada pode ser verificada na Figura 4.3. Não existe qualquer controlo sobre o número de tentativas permitidas nem um bloqueio do sistema quando é entendido que já foram executadas tentativas de acesso suficientes. Consoante o nível do utilizador que esteja a tentar aceder ao sistema este terá acesso limitado, ou não, ao que o sistema oferece.

4.2.2 Menu

A primeira página a ser mostrada quando se entra no sistema é a do menu. Caso o utilizador tenha permissões administrativas este pode ver e aceder a qualquer página do sistema, caso tenha permissões de laboratório só consegue aceder às páginas das receitas, dos relatórios de produção e da ajuda. É apresentada a possibilidade de navegação entre as páginas de duas maneiras, uma através da própria página do menu onde através de um clique o utilizador pode ir logo para a secção do sistema que pretender, a outra é através do menu lateral escondido que permite exatamente o mesmo tipo de navegação. Este último é acessível em qualquer parte do sistema. A implementação destes dois tipos de menu podem ser verificados na Figura 4.4 da página 49.

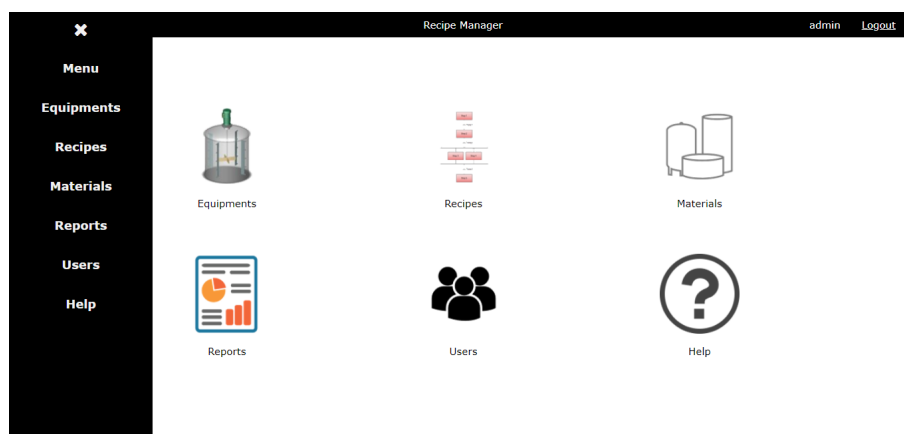


Figura 4.4: Página de Menu Implementada e Menu Lateral Oculto

4.2.3 Equipments

A parte do sistema relacionada com os equipamentos permite fazer a configuração inicial da fábrica no sistema. Isto é, permite a introdução de toda a informação relevante dos equipamentos disponíveis para que depois se possa criar as receitas em torno desta informação e destas limitações físicas. Esta secção do sistema arranca na página da configuração das Units Classes e o seu design pode ser visto na Figura 4.5 da página 50.

4.2.3.1 Units Classes

Nesta página podemos visualizar todas as famílias de equipamentos já criadas ou então adicionar novas. Quando se adiciona uma nova temos de indicar o seu nome e, opcionalmente, fazer algum tipo de comentário ou descrição relevante da mesma.

Na tabela que nos mostra as famílias de equipamentos já criadas temos a possibilidade de filtrar os dados através de pesquisa por palavras chave, ou ordenar a informação pela coluna que se pretender e ainda tornar visíveis ou não colunas que se considerem relevantes. Ainda na tabela é possível editar e apagar dados introduzidos e aceder aos atributos e às fases da família que se pretenda através dos respetivos botões apresentados nas linhas das diversas famílias.

Carregando no botão Attributes de uma das famílias de equipamentos o sistema mostra uma página onde são listados todos os atributos disponíveis. Caso aquela família ainda não tenha nenhum atributo é possível adicioná-lo, bastando indicar o seu nome e caso se pretenda, o seu valor por defeito, unidade de medida e uma pequena descrição sobre o mesmo. Estes atributos serão o que distingue este tipo de equipamentos dos restantes.

Carregando no botão Phases podemos consultar, na página que abre, todas as fases já atribuídas à família de equipamentos em questão. Caso se pretenda adicionar uma nova só temos de escolher uma de entre as fases disponíveis da nossa biblioteca e fazer a atribuição da mesma. Esta fase adicionada adquire os parâmetros de configuração que esta fase tem na biblioteca de fases do sistema permitindo que carregando no botão Parameters o utilizador seja direcionada para uma

Unit Class	Description	Commands
ReatorAquecido	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Attributes Phases Edit Delete
Reator	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Attributes Phases Edit Delete
Tanque	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Attributes Phases Edit Delete

Figura 4.5: Página Inicial da Secção Equipments Implementada

página onde é apresentada uma tabela com os diversos parâmetros da fase seleccionada. Aí o utilizador poderá alterar o valor atribuído por defeito aos parâmetros ou mesmo acrescentar esses valores caso estes não tenham nenhum atribuído ainda.

O sistema não permite a introdução de dados em duplicado por isso o utilizador nunca conseguirá introduzir famílias de equipamentos repetidas, equipamentos repetidos, e atribuir fases repetidas ao mesmo equipamento ou família de equipamentos. Quando é apagada uma família de equipamentos, todos os equipamentos atribuídos a essa família são também eles eliminados.

4.2.3.2 Units

O funcionamento da página Units é em tudo semelhante à das Units Classes diferenciando-se apenas num aspeto. Esse aspeto é o facto de quando se adiciona um equipamento ao sistema, temos de o associar a uma família de equipamentos já criada. Assim quando se faz essa associação, este equipamento criado adquire tanto os atributos como as fases e os parâmetros respetivos da família de equipamentos escolhida.

Na tabela inicial além de uma coluna dedicada ao nome do equipamento temos ainda uma coluna dedicada à família de equipamentos a que ele pertence. As outras funcionalidades de edição e gestão existentes na página Units Classes estão igualmente disponíveis nesta secção do sistema.

4.2.3.3 Links

A página dos Links implementada segue o mesmo estilo das anteriores, uma tabela onde são mostradas todas as ligações físicas introduzidas no sistema e dá ainda a possibilidade de editá-las, apaga-las ou ainda adicionar uma nova ligação.

O sistema apenas permite fazer ligações entre equipamentos que já estejam inseridos no sistema e não permite que a mesma ligação seja repetida ou inserida mais que uma vez. Caso algum

equipamento seja eliminado do sistema, as ligações aqui existentes que estejam relacionadas com ele também vão ser eliminadas.

4.2.3.4 Phases

Uma das partes mais importantes que existe no sistema e que está relacionada com os equipamentos é a página que permite a criação da biblioteca das fases que estão disponíveis e configuradas nos controladores existentes na fábrica. Isto permite dizer ao sistema que opções este tem para executar uma determinada receita e permite utilizar essa informação na configuração das famílias de equipamentos. Assim nesta página é permitido adicionar à nossa biblioteca todas as fases que desejarmos e que já estejam configuradas nos controladores. Depois de criada a fase podemos, carregando no botão Parameters, ir para a página que nos permite adicionar os parâmetros à fase que escolhemos. Estes parâmetros podem ter ou não valores e unidades de medida atribuídos.

Tal como nas outras partes do sistema não podemos criar fases em duplicado pois o sistema vai rejeitar essa introdução de dados. Não é permitido também apagar fases que já estejam atribuídas a equipamentos que existam atualmente no sistema.

4.2.3.5 Units of Measure

Não menos importante, e tal como especificado no projeto, também foi implementada a página Units of Measure. Esta permite consultar, editar e adicionar unidades de medida ao sistema. Esta informação é importante para quando o utilizador quiser atribuir unidades de medida a alguns parâmetros de fases ou mesmo a alguns atributos de equipamentos. Ao adicionar a informação aqui, tornamos esses dados disponíveis noutras partes do sistema sem ter de reintroduzir essa informação sempre que se precisa. Assim o utilizador apenas tem de escolher a unidade que pretende da biblioteca que foi criada nesta página.

O sistema não permite a eliminação de unidades de medida que estejam a ser utilizadas noutra parte do sistema.

4.2.4 Materials

A parte do sistema dedicada à gestão dos materiais foi também implementada conforme especificado no projeto. Foi criada uma separação clara entre os materiais e a localização deles. O design dessa parte do sistema pode ser consultado na Figura 4.6 da página 52.

4.2.4.1 Materials List

Na parte da página dedicada aos materiais, o sistema permite ao utilizador adicionar materiais de modo a que depois na criação das receitas o sistema os possa atribuir a algumas das fase que necessitem deste tipo de informação. Assim quando o utilizador insere um material ele deve indicar sempre que tipo de material é. Essa escolha é facilitada pois é mostrada uma lista dos tipos aceitáveis pelo sistema e o utilizador só tem de escolher um.

The screenshot shows the 'Materials' section of the 'Recipe Manager' application. The header includes the application name and user information. The main content area features a table with the following data:

Material	Type	Description	Commands
Ácido Sulfônico	Ingredients	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Characteristics Edit Delete
Amônia	Ingredients	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Characteristics Edit Delete
Butilglicol	Ingredients	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Characteristics Edit Delete
Lauriléter	Ingredients	Lorem ipsum dolor sit amet, consectetur adipiscing elit	Characteristics Edit Delete

Below the table, there is a pagination control showing 'Showing 1 to 4 of 4 entries' and a page indicator '1'.

Figura 4.6: Página Inicial da Secção Materials Implementada

Em cada material temos a possibilidade de aceder á página das suas características carregado no botão Characteristics. Esta ação leva o sistema a mostrar a página das caraterísticas do material escolhido e nessa página o utilizador pode inserir todas as características que distingam o material que escolheu dos outros semelhantes. Algumas destas características podem ser importantes para a atribuição de materiais a algumas fases de equipamentos devido ao facto de alguns materiais só poderem ser tratados em equipamentos específicos.

O sistema não aceita a introdução de materiais cujo o nome já exista no sistema e também não permite a introdução de características em duplicado num material. Não é permitido também apagar do sistema materiais que já tenham sido utilizados para a criação ou execução de receitas.

4.2.4.2 Location

Já na parte dedicada à localização dos materiais a implementação foi feita de modo a atribuir um material a uma localização e não o contrário. Se o contrário fosse feito, o controlo sobre a qualidade da informação era difícil e a probabilidade de deixar informação residual errada era muito grande. Isto é, se fossem atribuídas várias localizações ao mesmo material e tivéssemos uma entrada na tabela para cada uma delas, quando o material numa dessas localizações mudasse iria implicar eliminar aquela atribuição e e criar uma nova. O número de operações a realizar e a informação a mostrar iriam ser maiores e a utilidade dela seria a mesma. Assim foi escolhido antes atribuir um material a um dos equipamentos existentes no sistema capazes de armazenar esse tipo de produto. Quando há uma mudança do produto nesse equipamento apenas temos de mudar o material e não mudar equipamentos nos diversos materiais (eliminar associação antiga e criar a nova).

O sistema por não permite que o mesmo equipamento tenha mais do que um material associado porque também na realidade isso não pode ocorrer senão há contaminação de material por mistura de vários ingredientes.

The screenshot shows the 'Recipe Manager' application interface. At the top, there is a navigation bar with a hamburger menu, the application logo 'JRM', the title 'Recipe Manager', and user information 'admin' and 'Logout'. Below the navigation bar, the page title is 'Recipes' with links for 'List' and 'Phases'. A green '+ Add' button is visible. A search bar and a refresh button are also present. The main content is a table with the following data:

Recipe	Version	Product	Status	Default Batch	UoM	Commands
Receita de Teste	1.0	Creme	Inactive	120		Open Appr New Copy Edit Delete

At the bottom of the table, there is a pagination control showing 'Showing 1 to 1 of 1 entries'.

Figura 4.7: Página Inicial da Secção Recipes Implementada

4.2.5 Recipes

Facilmente identificado como o principal pilar deste sistema, o grupo de páginas relacionadas com as receitas foi implementado de maneira a tentar que todo o processo fosse o mais simples e intuitivo possível e de modo a garantir que toda a informação criada faz sentido e que tem um propósito. Este grupo de páginas está dividido em duas secções: a List onde são listadas todas as receitas e todas as versão de receitas alguma vez criadas e a Phases onde podemos criar fases com parâmetros já configurados como por exemplo "Adicionar Água" onde fixamos o produto e deixamos em aberto a quantidade para que seja inserida na criação da receita. Quando se entra na secção do sistema responsável pelas receitas o sistema mostra-nos a página List. Um exemplo dessa página implementada pode ser consultado na Figura 4.7.

Nesta página a única informação criada relacionada com a receita é apenas o seu *header*. Como a quantidade de informação do *header* das receitas é muito grande, inicialmente só estão visíveis o nome da receita, a versão, o produto que resulta dela, o seu estado, o tamanho típico do lote produzido relacionado com esta receita e a sua unidade de medida. Toda a restante informação está disponível mas está escondida. Assim o utilizador só precisa de tornar visível a informação que pretender visualizar caso ela não esteja já visível.

Quando o utilizador cria uma nova receita, o sistema gera uma receita pré-preenchida utilizando um template de *Unit Procedures* e *Operations* estabelecidos pela empresa. Neste caso são criados dois *Unit Procedures*: "Operações de Fabrico de" e "Operações de Descarga de". Cada nome será completado pelo produto que a receita em questão irá produzir. O primeiro e segundo *Unit Porcedure* terão uma única *Operation* cada um, sendo elas "Fabrico de" e "Descarga de" respetivamente. O facto deste template ser criado não impede a adição de novos passos ou edição dos passos criados. Este template é utilizado porque vai de encontro à maioria dos casos em que a empresa vai utilizar o sistema e assim só tem de preencher as *Phases* de cada uma das *Operations*.

Caso o utilizador pretenda entrar no editor da receita apenas precisa de carregar no botão

Open, se pretender criar uma nova versão de uma receita pode fazê-lo carregando no botão New, se pretender criar uma cópia de uma receita já existente o sistema permite fazê-lo carregando no botão Copy e se pretender ir para a página onde estão disponíveis as aprovações já realizadas sobre a receita ou onde pode realizar uma aprovação nova basta carregar no botão Appr.

O sistema só permite criar novas versões e cópias de receitas que já estejam aprovadas para produção, ou seja, que o seu estado seja Approved for Production. Uma receita só pode ser apagada e editada se o seu estado for Inactive.

Se o utilizador carregar no botão New para criar uma nova versão de uma receita é-lhe perguntado qual o número da nova versão. Depois de introduzida essa informação o sistema cria essa nova versão. Para a criar o sistema vai buscar toda a informação do *header* da receita da qual se pretende criar a nova versão. Desse *header* é criado um novo onde o estado é mudado para Inactive e a versão é alterada para a nova versão. Este novo header é então introduzido na tabela *m_recipe_element* como o *header* da nova receita. De seguida é feita a recolha de todos os requisitos de equipamentos que a receita a copiar tinha. Assim esses requisitos são replicados para a nova receita e introduzidos na tabela *m_recipe_element equip* da base de dados. De maneira a replicar o processo da receita são recolhidos todos os passos existentes na tabela *m_recipe_step* correspondentes à receita a copiar e são replicados para a nova versão. São também replicados os parâmetros existentes na tabela *m_recipe_step_parameter*, e as ligações entre os diversos passos da receita que existem na tabela *m_recipe_link*.

Se o utilizador carregar no botão Copy para criar uma cópia de uma receita é-lhe perguntado qual o nome da nova receita que pretende criar através da cópia. Quando essa informação é dada ao sistema, ele cria a cópia da receita. Para criar a cópia o sistema vai buscar toda a informação do *header* da receita da qual se pretende criar uma cópia. Desse *header* é criado um novo onde o nome da receita é mudado para o novo, o estado é mudado para Inactive e a versão é alterada para 1.0. O restante processo é em tudo semelhante ao de criar uma nova versão. São replicadas todas as informações existentes nas tabelas *m_recipe_element equip*, *m_recipe_step*, *m_recipe_step_parameter* e *m_recipe_link* que correspondam à receita a copiar e são introduzidas mas associadas à nova receita criada.

Se o utilizador pretender modificar o estado de uma receita basta carregar no botão Appr e será redirecionado para a página onde tem acesso ao histórico dos estados da receita em questão. Nesse histórico estão disponíveis informações como quem aprovou, para que estado aprovou e quando aprovou. Se uma receita estiver no estado Inactive o utilizador apenas pode mudar o estado para Waiting Approval. Se o estado ativo for Waiting Approval o utilizador pode mudar o estado para Not Approved ou Approved for Test. Se o estado atual da receita for Not Approved, o utilizador pode alterar o estado para Waiting Approval ou Inactive, sendo que o primeiro serve para submeter de novo a receita a aprovação e o segundo para permitir edições na receita. Se o estado da receita for Approved for Test ele pode ser mudado para Approved for Production ou para Not Approved. Caso a receita chegue ao estado Approved for Production esta não pode sofrer mais alterações a nível de estado nem de conteúdo.

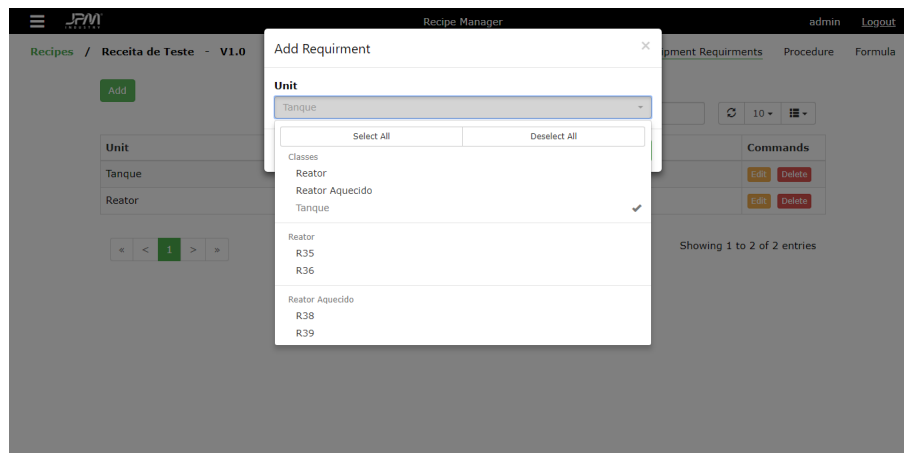


Figura 4.8: Página com Formulário de adição de Equipment Requirements

Entrando no editor da receita carregando no botão Open é apresentado ao utilizador uma primeira página, a Equipment Requirements, onde é suposto serem introduzidos os requisitos a nível de equipamento que a receita que vai criar carece. Podem ser indicadas famílias de equipamentos ou mesmo algum equipamento em específico. Este tipo de informação provém de informação já introduzida anteriormente no sistema aquando a configuração dos equipamentos e vai ajudar o sistema a limitar as *Unit Procedures*, *Operations* e *Phases* que permite o utilizador usar na construção da receita. Tal como noutras partes o sistema não permite introduzir requisitos em duplicado e a seleção é feita de uma maneira única do sistema onde podemos selecionar vários equipamentos ao mesmo tempo. Este tipo de seleção pode ser consultado na Figura 4.8.

De seguida, quando o utilizador parte para a construção do processo da receita ele terá de ir para a página Procedure. Nesta página, devido à complexidade de implementação de uma solução de desenho usando diagramas, foi optado por descrever graficamente o processo utilizando apenas o menu lateral e fazendo a representação dos diversos passos em forma de tabela auxiliados por uma representação na parte inferior do menu lateral que indica o nível da receita em que o utilizador se encontra. De maneira a facilitar a visualização do que foi descrito, esta disposição de elementos pode ser consultada na Figura 4.9 da página 56.

No diagrama auxiliar existem quatro níveis de representação: o primeiro nível está atribuído ao *Recipe Procedure*, o segundo nível aos *Unit Procedures*, o terceiro às *Operations* e o quarto às *Phases*. Já no menu estão representados só os três últimos níveis acabados de indicar. O menu, tal como a receita vai sendo gerado desde o nível superior até ao inferior. Inicialmente o utilizador tem de introduzir todos os *Units Procedures* que são constituintes da receita. Para isso tem de introduzir os vários passos desse processo. Essa ação é feita através de um campo disponível nessa página onde o utilizador escolhe que símbolo ou que *Unit Procedure* pretende introduzir e dá o nome pretendido a esse passo. Os passos de uma receita têm de ter todos nomes diferentes e o sistema faz verificações disso de modo a que o processo criado seja válido. Não podemos esquecer também que cada nível da receita tem de ter um símbolo de início e um de

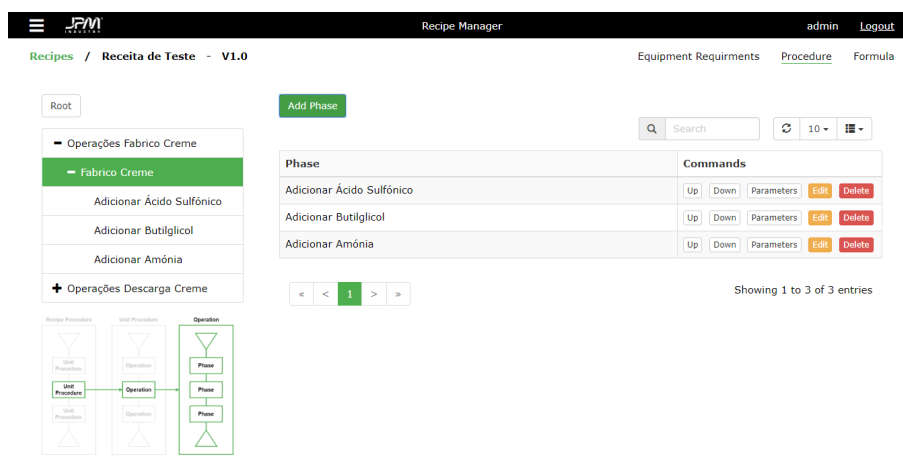


Figura 4.9: Página do Editor de Receitas Implementada

fim. Este processo é replicado para a introdução de *Operations* e de *Phases* sendo que nas *Phases* só são apresentadas os tipos de fases que os equipamentos escolhidos são capazes de executar e após esta seleção é apresentada a lista de fases pré-configuradas anteriormente e que partilhem desse mesmo tipo sendo que sempre que uma fase é introduzida são perguntados os valores dos parâmetros que essa mesma fase precisa de modo a que ela seja introduzida com toda a informação necessária. Estes parâmetros podem ser alterados posteriormente carregando no botão *Parameters* disponível na linha da tabela correspondente à fase que se pretenda alterar. Caso o utilizador pretenda movimentar algum dos passos criados pode utilizar os botões *Up* e *Down* para os mover para cima ou para baixo respetivamente.

Como o menu representativo da receita que o sistema utiliza é construído com o plugin *Bootstrap Tree View*, os dados disponíveis diretamente da criação da receita não servem para criar o menu. Desse modo têm de ser tratados de forma a que este seja capaz de os ler e utilizar. Assim, foram criados três algoritmos que são responsáveis por gerar as hiperligações de cada elemento do menu, por fazer a verificação dos parâmetros da receita alterando a cor de fundo para vermelho das fases que não estiverem com os valores devidamente preenchidos e por criar uma ordenação dos diversos passos da receita de forma a serem interpretáveis pelo plugin.

O algoritmo responsável pelas hiperligações cria o link com o caminho e informação necessária para que o sistema saiba que elemento foi carregado e que o utilizador pretende consultar. Isto é, constrói um link onde são passadas variáveis que indicam qual o tipo de elemento que foi carregado (*Unit Procedure*, *Operation* ou *Phase*) e indicam também a que passo da receita o elemento pertence. Assim o sistema sabe que tipo de página abrir, que elemento expandir no menu lateral e que informação mostrar na nova página. O link criado é então armazenado na linha do passo a que corresponde na coluna *href* da tabela *m_recipe_step*.

Para que o plugin do menu saiba que informação mostrar e por que ordem a deve mostrar, há um algoritmo que tem como dois principais objetivos ordenar verticalmente tanto os níveis como os passos de cada nível da receita e também indicar a que passo é que um grupo de passos está li-

gado, isto é, por exemplo, a que operação está ligado um grupo de passos que representam as fases dessa operação. Assim para realizar a primeira parte, o algoritmo através da informação gerada na tabela `m_recipe_link` consegue saber qual a ordem pela qual os passos de um determinado nível da receita são executados. Assim para cada nível, o algoritmo corre todas as ligações e passos existentes e ordena-os numericamente desde o símbolo de início, que corresponderá ao número 1, até ao símbolo de fim. Esta informação é então introduzida na tabela `m_recipe_element` na coluna `tree_priority`. Para a segunda parte, o algoritmo corre todos os passos da receita e verifica a que passo estes estão relacionados hierarquicamente, ou seja, a que elemento do nível superior pertencem. Depois o id desse passo correspondente do nível superior será introduzido em todos os passos que estejam relacionados com ele e existam no nível inferior. Esta informação é introduzida na coluna `parent_id` da tabela `m_recipe_step`.

Por fim a verificação se os parâmetros de todas as fases estão devidamente preenchidos é feita por um algoritmo que verifica se o campo `parameter_value` da tabela `m_recipe_step_parameter` das fases da receita aberta contém alguma informação ou não. Caso haja alguma fase que não tenha parâmetros preenchidos ela é simbolizada a vermelho no próprio menu.

Conforme o procedimento da receita vai sendo construído, na página *Formula*, existem duas tabelas, uma dedicada aos inputs e outra às variáveis de processo, onde os parâmetros das fases que vão sendo preenchidas vão sendo listados. Assim no final da construção da receita é possível verificar que materiais vão ser necessários, que quantidades e ainda os dados atribuídos a algumas variáveis do processo permitindo uma verificação final de alguns destes dados por parte do utilizador.

4.2.6 Users

Na página dos utilizadores implementada o utilizador, com permissões administrativas, pode adicionar e apagar contas de acesso ao sistema. Para adicionar uma conta basta introduzir o nome de utilizador, o nome do utilizador, o seu nível de acesso e a sua password. Caso pretenda alterar a password de acesso de alguma conta basta carregar no botão *Password* da linha do utilizador que pretender e introduzir a nova password para essa conta. Qualquer introdução de password é verificada através de uma dupla introdução da mesma, sendo que a operação só é aprovada se as duas corresponderem. Não é feita qualquer verificação ou exigência a nível da constituição da mesma ou do seu tamanho. Caso pretenda editar o `username`, o nome ou mesmo o nível de acesso de algum utilizador basta carregar no botão *Edit* e um editor aparece de modo a que essas alterações possam ser feitas. A interface desta página pode ser consultada na Figura 4.10 da página 58.

Na página *Logs* é listado todos os acessos ao sistema pelas diversas contas existentes. Esta informação provem da tabela `users_logs` e é lá introduzida sempre que um utilizador faça um login no sistema com sucesso. Esta página é puramente expositiva sendo que não é possível adicionar ou apagar dados dos sistema, apenas permite consultar informação.



Username	Name	Level	Commands
admin	Paulo Fontes	JPM	Password Edit Delete

Figura 4.10: Página Inicial da Secção Users Implementada

4.3 Testes e Resultados

De forma a validar a implementação feita do sistema projetado é necessário realizar um conjunto de testes através dos quais se consiga retirar informação que seja útil à aprovação do sistema ou não. Assim o sistema foi testado em quatro áreas distintas. A primeira passa por fazer uma verificação se ao longo da sua utilização o sistema está a inserir corretamente nas tabelas da base de dados a informação que o utilizador introduz no sistema. A segunda passa por verificar se o sistema está a bloquear a introdução de informação errada na base de dados que torne depois inviável a utilização do sistema ou mesmo a utilização das receitas nele criadas. A terceira passa por verificar se as restrições de acesso a algumas partes do sistema e algumas restrições de ações que não podem acontecer em certas situações estão mesmo bloqueadas. E por fim há ainda uma área direcionada à velocidade de execução do sistema enquanto está a ser utilizado de maneira a identificar possíveis zonas que tenha execuções mais lentas permitindo assim tentar arranjar outras formas de implementar a mesma funcionalidade que permitam uma execução mais rápida.

Algo importante também a ter em consideração durante estes testes é o documento de registos de erros que o próprio servidor Apache tem pois permite saber que ficheiro do sistema não correu como devia ou que query a base de dados não conseguiu correr e normalmente consegue dizer o porquê permitindo detetar logo onde está a falha na implementação e arranjar uma solução

4.3.1 Correta Inserção de Dados na Base de Dados

Para testar a informação que era inserida nas tabelas da base de dados foi utilizado o software phpPgAdmin que permite fazer uma consulta de todos os dados introduzidos nas tabelas. Assim é possível cruzar a informação que é esperado que lá estivesse com a que realmente está e aprovar, ou não, as partes que se estão a testar no sistema. Um exemplo da maneira como esta informação é mostrada pode ser visto na Figura 4.11 da página 59. Esta ferramenta mesmo no desenvolvimento do sistema foi muito útil pois permitiu identificar falhas na implementação e corrigi-las.

id	re	re_version	from_element	to_element	m_recipe	m_recipe_version	parent_step
80	Receita de Teste	1.0	begin_link	Inicio Receita	Receita de Teste	1.0	Receita de Teste
81	Receita de Teste	1.0	Inicio Receita	Fabrico da Receita	Receita de Teste	1.0	Receita de Teste
82	Receita de Teste	1.0	Fabrico da Receita	Descarga do Produto A	Receita de Teste	1.0	Receita de Teste
83	Receita de Teste	1.0	Descarga do Produto A	Fim da Receita	Receita de Teste	1.0	Receita de Teste
84	Fabrico	1.0	begin_link	Inicio Fabrico da Receita	Receita de Teste	1.0	Fabrico da Receita
85	Fabrico	1.0	Inicio Fabrico da Receita	Operações do Fabrico	Receita de Teste	1.0	Fabrico da Receita
86	Fabrico	1.0	Operações do Fabrico	Fim Fabrico da Receita	Receita de Teste	1.0	Fabrico da Receita
87	Operações	1.0	begin_link	Inicio Operações do Fabrico	Receita de Teste	1.0	Operações do Fabrico
88	Operações	1.0	Inicio Operações do Fabrico	Adicionar Amônia	Receita de Teste	1.0	Operações do Fabrico
89	Operações	1.0	Adicionar Amônia	Ligar Agitador	Receita de Teste	1.0	Operações do Fabrico
90	Operações	1.0	Ligar Agitador	Adicionar Butilglicol	Receita de Teste	1.0	Operações do Fabrico
91	Operações	1.0	Adicionar Butilglicol	Validar Lab	Receita de Teste	1.0	Operações do Fabrico
92	Operações	1.0	Validar Lab	Fim Operações do Fabrico	Receita de Teste	1.0	Operações do Fabrico

Figura 4.11: Consulta de Dados de uma Tabela Usando o phpPgAdmin

4.3.2 Introdução Errada de Dados no Sistema

Relativamente à introdução errada de dados esta é controlada de três maneiras, a primeira é limitando a escolha dos dados a introduzir em diversos campos apresentando apenas um lote de escolhas ao utilizador, outra maneira é através da verificação da existência de dados já com o mesmo nome evitando a introdução em duplicado de informação e por fim é uma verificação se todos os campos obrigatórios estão preenchidos.

No primeiro caso não há qualquer tipo de preocupação com a ação do utilizador, mas no segundo e terceiro caso se o utilizador tenta fazer a introdução em duplicado de informação ou então não preenche os campos todos que são necessários para a introdução daquela informação, este é avisado através de um alerta sobre qual dos erros cometeu no processo de modo a retificar o problema. Foi verificado que estes avisos estão a funcionar em todas as páginas, mas, falando no terceiro caso, podiam estar especificados ao campo que faltou preencher e não avisando apenas que os campos necessários não estão preenchidos.

4.3.3 Restrições de Ações no Sistema

Um tipo de restrição a verificar é caso não haja uma sessão iniciada no servidor, independentemente do endereço do sistema que se coloque no navegador web o utilizador é sempre redirecionado para a página de login tendo assim o seu acesso bloqueado ao sistema. Isto foi comprovado tentando o acesso a todos os endereços existentes do sistema implementado.

Outro tipo de restrição é a separação de acesso às páginas do sistema dependendo do tipo de conta com que a autenticação no sistema está feita. Foi verificado que pela interface do sistema é impossível ao utilizador com tipo de conta de laboratório aceder às páginas restritas ao administrador da empresa. Foi também verificado que mesmo que tente colocar o endereço das páginas, caso não deva ter acesso a elas, o utilizador é redirecionado para a página do menu do sistema.

Por fim o último tipo de restrição é não poder apagar materiais que já tenham sido utilizados, não poder apagar receitas que já tenham sido aprovadas para produção e não poder apagar símbolos de início quando existem passos da receita dependentes dele. Estes bloqueios foram confirmados sendo que o utilizador é sempre avisado através de um alerta de que a operação que está a tentar exercer não é possível ser executada.

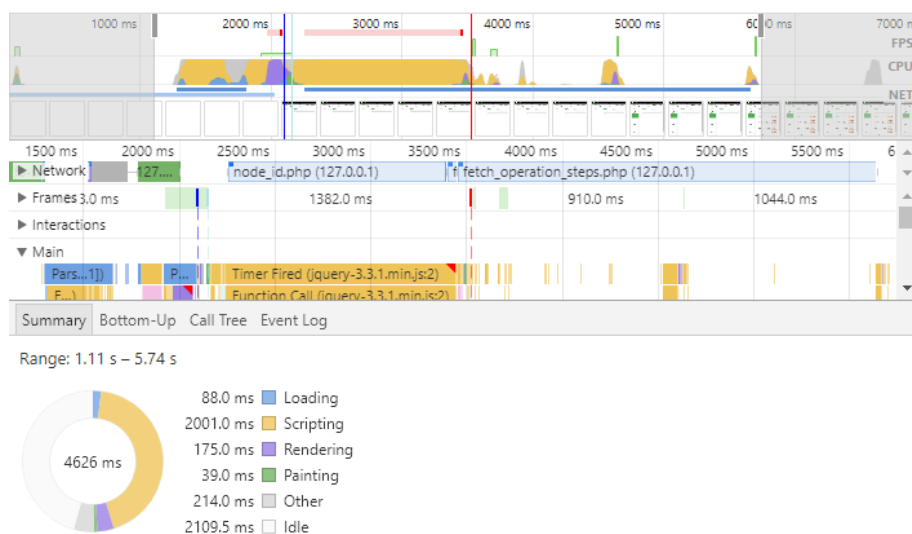


Figura 4.12: Informação da Secção Performance das Ferramentas de Programador do Google Chrome

4.3.4 Velocidade do Sistema

Antes de se ter em consideração os dados recolhidos neste teste é importante ter em conta a máquina onde o sistema foi testado. O computador onde foi realizado o teste tinha as seguintes características:

- Sistema Operativo: Windows 10 Pro 64 bits;
- Memória RAM: 4 GB DDR2 400 MHz;
- Processador: Pentium(R) Dual-Core T4400 @ 2.20GHz;
- Disco Rígido: Western Digital Scorpio Black WD5000BEKT 500GB 7200 RPM;

É preciso além disso também ter em consideração que o sistema estava a correr num servidor alojado no próprio computador.

Para realizar as medições dos tempos que as diversas operações no carregamento das páginas demoravam a ser efetuadas foram utilizadas as 'ferramentas de programador' existentes no navegador web Google Chrome Versão 66.0.3359.139, mais concretamente a secção destinada à performance da página. Esta permite fazer o carregamento de uma página à escolha e faz o registo completo de tudo o que se passa durante esse evento. Um exemplo deste ambiente pode ser visto na Figura 4.12.

Assim foi possível verificar que para as páginas com pouca informação onde só existe uma tabela com os dados referentes àquela página o navegador demora cercar de três segundos a mostrar a página, mas se fizermos o mesmo teste numa página Procedure de uma receita onde está aberta uma *Operation* e estão listadas as *Phases* todas, que é o caso onde uma página tem mais informação e mais dados para mostrar tal como o menu lateral e a tabelas das fases todas, o navegador já

demora entre seis a sete segundos a mostrar a página. Para uma implementação como a que foi feita onde sempre que se muda de *Operation* ou de *Unit Procedure* a página é carregada de novo, já estamos a falar de uma possível área a melhorar pois sete segundos já pode ser considerado por muitos utilizadores como tempo a mais de espera para que a informação seja mostrada.

Capítulo 5

Conclusões e Trabalho Futuro

Neste capítulo são apresentadas as conclusões obtidas da realização desta dissertação, são abordados os objetivos da mesma analisando se foram cumpridos ou não, e se não, porque motivo não foram cumpridos sendo ainda apresentadas perspectivas para a continuação do desenvolvimento do sistema, novas funcionalidades a implementar e abordados os principais cuidados a ter na continuação do seu desenvolvimento.

5.1 Conclusões e Satisfação dos Objetivos

Devido à grande necessidade de simplificar processos na indústria e de manter rigor na qualidade dos produtos que são produzidos surgem soluções que vão desde automatizar ao máximo possível os processos de produção até à criação de software que permita uma configuração do processo de produção de forma rápida e rigorosa. Da realização desta dissertação resulta um projeto e uma implementação de um protótipo de um Sistema de Gestão de Receitas para Indústria de Processo que tenta ir de encontro a essa necessidade atual das empresas associadas a esse tipo de indústria.

Do estudo da indústria de processo, das normas aplicáveis a este tipo de sistemas e das soluções disponíveis atualmente no mercado, resulta a conclusão de que desenvolver e implementar um sistema que cumpra com todos os requisitos impostos pelas normas e que vise algum dia ser reconhecido como cumpridor delas é extremamente difícil, complexo e exige um intervalo de tempo significativo de desenvolvimento e testes até que se tenha uma versão utilizável. Além disso os sistemas atuais no mercado já têm anos de desenvolvimento e múltiplas iterações de correções e adição de novas funcionalidades e fornecem ainda suporte em caso de mal funcionamento. Contudo, muitas das funcionalidades que o mercado oferece não são necessárias para os objetivos do sistema que a empresa propôs e necessita. Isto ajuda a simplificar o projeto e implementação do sistema realmente necessário e permite uma abordagem mais direta às funcionalidades que são realmente cruciais. Outro ponto também importante é o facto de a empresa pretender que o sistema siga o máximo possível o modelo imposto pela ISA88 mas não o precisa de seguir com

rigor. Seguindo também apenas as partes da norma que estejam diretamente relacionadas com as funcionalidades necessárias.

Dos requisitos impostos ao sistema foi projetada uma solução que permite o cumprimento de todos eles, mas na implementação da versão protótipo do sistema não foram cumpridos os objetivos de gerar os relatórios da produção dos lotes e a parte que permitia a consulta dessa informação. Estes não foram cumpridos devido ao facto de ainda não estar definido por parte da empresa que dados iam ser gerados da execução dos lotes e daí não haver informação suficiente para fazer a implementação desta parte do sistema. De salientar que no final da Dissertação o objetivo desta parte do sistema mudou, sendo que atualmente o que a empresa pretende é que sejam gerados documentos respetivos a cada receita onde seja apresentado um histórico de alterações da mesma e onde esteja disponível todo o processo da mesma para caso tenha de ser executada manualmente. Contudo e apesar desta pequena alteração todos os outros requisitos do sistema foram cumpridos.

Apesar de a solução implementada cumprir a maioria dos requisitos, sofre de grandes limitações ao nível da criação e edição das receitas. Isto porque neste momento não é capaz de demonstrar de forma clara no menu lateral no editor de receitas os casos onde existam ramificações com opções ou execuções em paralelo e porque não possui um ambiente onde permita o desenvolvimento da receita em forma de diagrama facilitando assim o processo criativo de quem gere as receitas. A primeira limitação neste momento não é muito importante pois na primeira versão implementada a empresa só pretende criar receitas em que a sua execução é verticalmente sequencial sem alternativas de execução ou execuções em paralelo. Esse tipo de receitas será para implementar em versões posteriores e se o ambiente gráfico for também ele implementado a limitação do menu lateral em representar esse tipo de processo é de certa forma ultrapassada também.

Mesmo com estas limitações todas as funcionalidades do sistema, à exceção das ligadas aos relatórios, estão implementadas e operacionais permitindo ao utilizador fazer uma configuração dos equipamentos da fábrica, fazer uma gestão dos materiais que estão disponíveis para produção, criar e gerir receitas e ainda gerir os acessos ao sistema. Consegue cumprir ainda com um dos seus maiores objetivos que era o de correr em navegadores web permitindo que fosse um sistema independente da máquina e do sistema operativo onde estivesse a correr. Ainda assim o sistema carece de testes mais profundos e diversificados de formar a garantir o correto funcionamento do mesmo.

5.2 Trabalho Futuro

O trabalho a desenvolver no futuro passará por implementar as partes do sistema que ficaram por implementar no protótipo, isto é, a parte responsável por gerar os relatórios através dos dados gerados da execução dos lotes, a parte do sistema onde esta consulta é permitida e ainda a página Help onde constam todas as ajudas necessárias à utilização do sistema. Todas as ferramentas necessárias para implementar estas partes constam no projeto do sistema.

Posteriormente à implementação completa do sistema terá de haver uma fase de testes que visa validar o sistema antes deste ser testado numa fábrica em ambiente real. Nesta fase de testes deve haver um documento onde sejam claros os diversos testes a serem feitos e o sistema deve ser apresentado a pessoas externas ao seu desenvolvimento de forma a que sejam detetadas novas melhorias através de ideias que de outra forma serão ignoradas ou então continuariam desconhecidas aos seus projetistas. Para além disso deve ser testada a sua compatibilidade com os diferentes navegadores web nos quais se entenda que haja a possibilidade do sistema se executado. Numa segunda fase de teste já nas instalações, caso passe com sucesso os testes especificados e desempenhe devidamente todas as funções a que se propõe, o sistema recebe então uma validação final ficando sujeito depois a evoluções que se acredite poderem valorizar e complementar o sistema.

Algumas evoluções para este sistema que podem vir a ser desenvolvidas e implementadas no futuro passam por editar a construção das páginas do sistema de forma a que se adaptem ao tipo de dispositivo onde estão a ser abertas, isto é, se estão a ser usadas num telemóvel, num tablet ou num computador. Passam também por desenvolver uma parte do sistema que seja responsável por apresentar indicadores de desempenho em relação à produção das receitas permitindo assim uma evolução das receitas tendo em conta as que apresentem melhores resultados, com ou sem ajustes dos operadores. Podem também passar por editar o sistema de modo a que o desenvolvimento da receita seja feito localmente e só quando é pretendido que seja guardado é que é feita a transferência para a base de dados, agilizando assim o processo de edição cortando todas as transferências de informações entre o sistema e a base de dados que ocorre na versão atualmente implementada. Uma outra evolução interessante para o sistema seria na parte das escolhas dos requisitos de equipamentos para as receitas, onde além de escolhermos os equipamentos que queremos podíamos ter um cruzamento de informação dos materiais que pretendemos usar na receita com a informação das características dos equipamentos e que tipo de materiais estes podem conter e tratar sendo assim apresentados para escolha os equipamentos que cumprissem com os requisitos impostos pelo utilizador. Também relacionado com as receitas seria importante que além da criação das receitas com quantidades fixas houvesse a possibilidade de as criar com quantidades relativas permitindo assim um escalonamento da receita em relação à quantidade de produto final que se pretender produzir no lote em questão.

Caso se pretenda continuar com a implementação do protótipo do sistema e que se tenha como objetivo chegar a uma versão completa do mesmo é importante que tenha em consideração que os algoritmos usados para a criação do menu lateral do editor da receita não têm em consideração outros símbolos além dos de início e fim de processo por isso caso haja interesse em implementar as restantes alternativas terá de haver uma revisão deste algoritmo.

Anexo A

Linguagem Gráfica PFC

Neste anexo está presente a simbologia da linguagem gráfica PFC utilizada na norma ISA88.

A.1 Utilizadores

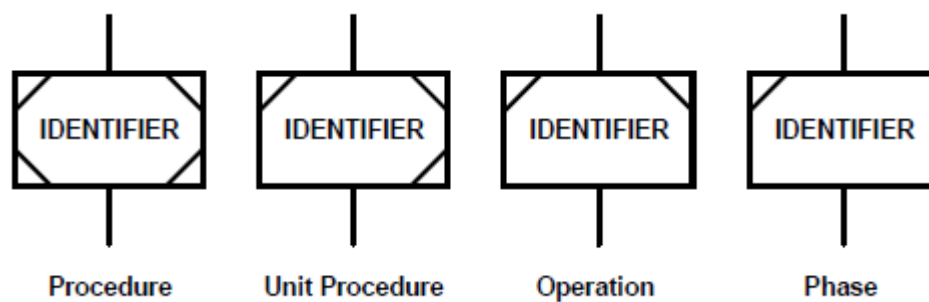


Figura A.1: Símbolos Representativos de Elementos de uma Receita [2]



Figura A.2: Símbolo de Início [2]



Figura A.3: Símbolo de Fim [2]

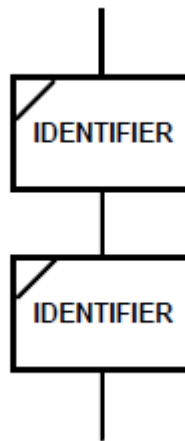


Figura A.4: Transição Implícita [2]

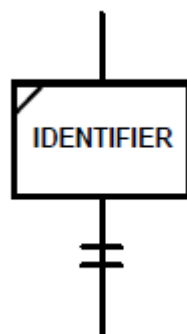


Figura A.5: Transição Explícita [2]

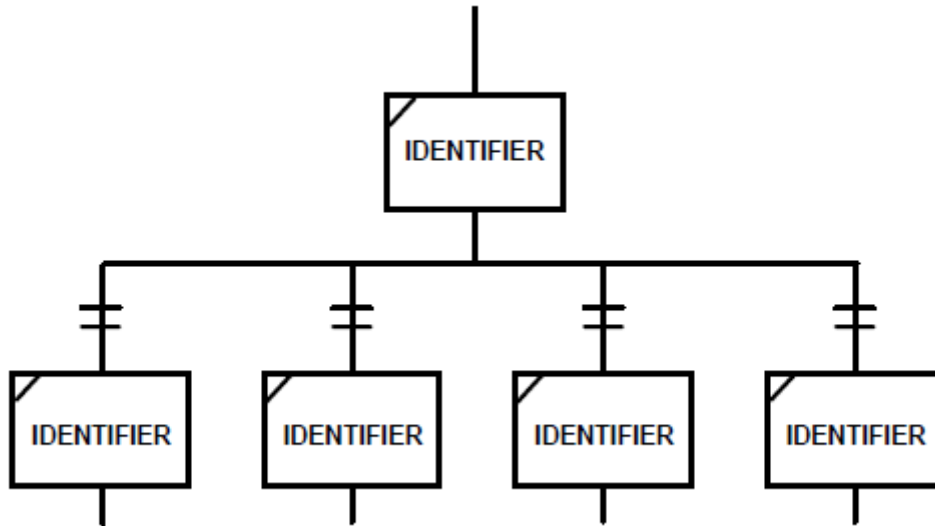


Figura A.6: Início de uma Sequência de Escolhas [2]

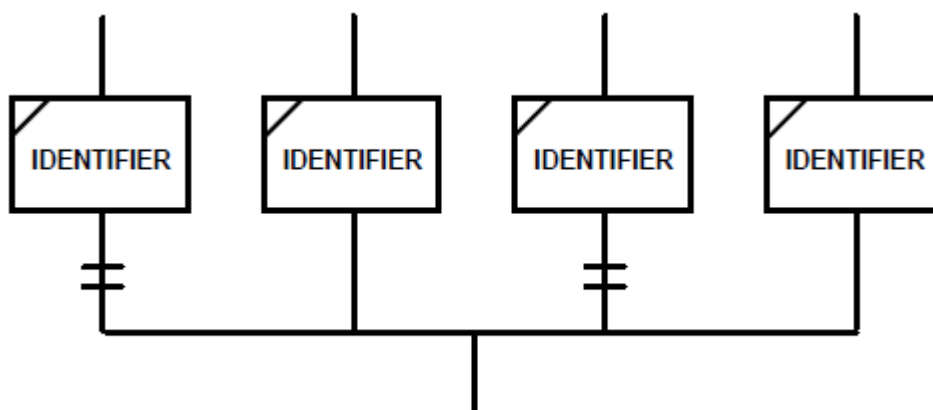


Figura A.7: Fim de uma Sequência de Escolhas [2]

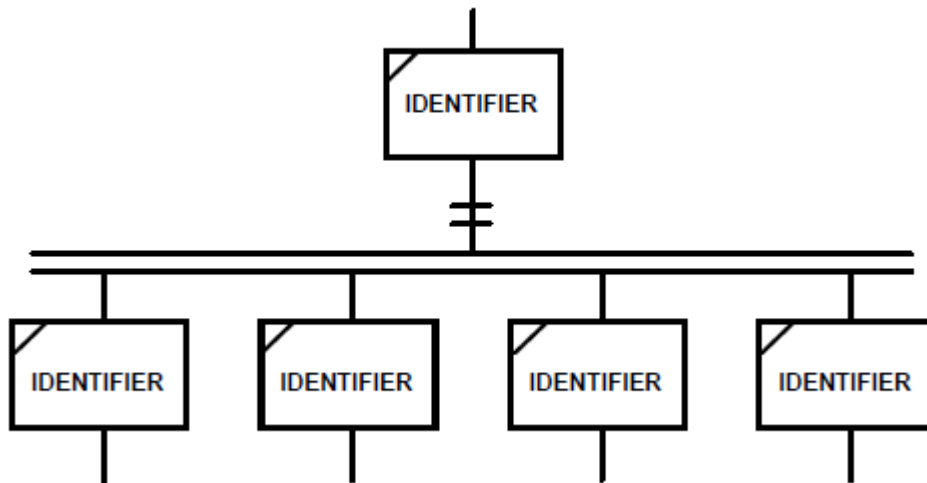


Figura A.8: Início de Várias Escolhas em Simultâneo [2]

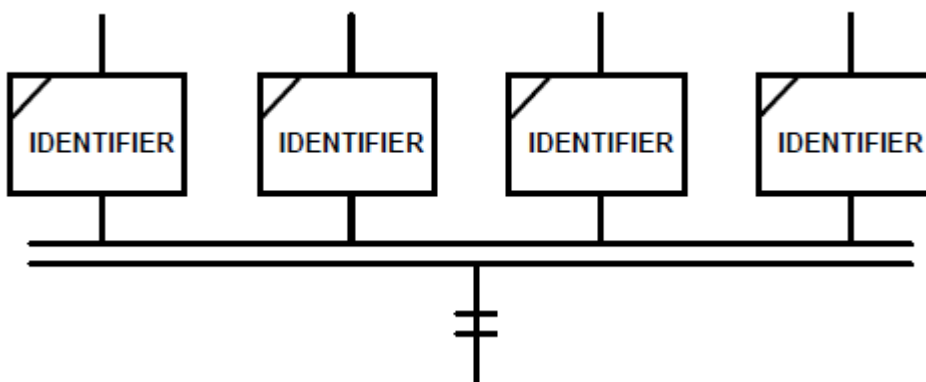


Figura A.9: Fim de Várias Escolhas em Simultâneo [2]

Anexo B

Código SQL das Tabelas

Neste anexo está presente o código SQL necessário para a criação das tabelas do sistema numa base de dados.

B.1 Utilizadores

```
CREATE TABLE users (  
    id          SERIAL,  
    username    varchar (128)    NOT NULL UNIQUE,  
    name        varchar (128)    NOT NULL,  
    password    varchar (255)    NOT NULL,  
    level       varchar (128)    NOT NULL  
);
```

```
CREATE TABLE users_logs (  
    id          SERIAL,  
    username    varchar (128)    NOT NULL,  
    user_log    timestamp        NOT NULL,  
    FOREIGN KEY (username)  
        REFERENCES users(username)  
);
```

B.2 Equipamentos

```
CREATE TABLE equip_element (  
    id          SERIAL,  
    equipment   varchar (32)     NOT NULL,  
    ee_type     varchar (32)     NOT NULL,  
    ee_class    varchar (32),
```

```
description varchar (255),
PRIMARY KEY (equipment)
);

CREATE TABLE equip_property (
    id SERIAL,
    equipment varchar (32) NOT NULL,
    property varchar (255) NOT NULL,
    property_value varchar (32),
    engr_units varchar (32),
    description varchar (255),
    PRIMARY KEY (equipment, property),
    FOREIGN KEY (equipment)
        REFERENCES equip_element(equipment)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (engr_units)
        REFERENCES units_measure(engr_units)
        ON UPDATE CASCADE
        ON DELETE SET NULL
);

CREATE TABLE equip_interface (
    id SERIAL,
    equipment varchar (32) NOT NULL,
    equipment_phase varchar (64) NOT NULL,
    description varchar (255),
    PRIMARY KEY (equipment, equipment_phase),
    FOREIGN KEY (equipment)
        REFERENCES equip_element(equipment)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (equipment_phase)
        REFERENCES equip_phases(phase)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

CREATE TABLE equip_interface_parameter (
    id SERIAL,
```

```

equipment          varchar (32)    NOT NULL,
equipment_phase    varchar (64)    NOT NULL,
parameter          varchar (32)    NOT NULL,
param_type        varchar (32)    NOT NULL,
engr_units         varchar (32),
default_value     varchar (128),
description        varchar (255),
PRIMARY KEY (equipment, equipment_phase, parameter),
FOREIGN KEY (equipment)
    REFERENCES equip_element(equipment)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (equipment_phase)
    REFERENCES equip_phases(phase)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (engr_units)
    REFERENCES units_measure(engr_units)
    ON UPDATE CASCADE
    ON DELETE SET NULL
);

```

```

CREATE TABLE equip_link (
    id                SERIAL,
    equipment         varchar (32)    NOT NULL,
    to_equipment     varchar (32)    NOT NULL,
    description       varchar (255),
    PRIMARY KEY (equipment , to_equipment),
    FOREIGN KEY (equipment)
        REFERENCES equip_element(equipment)
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    FOREIGN KEY (to_equipment)
        REFERENCES equip_element(equipment)
        ON UPDATE CASCADE
        ON DELETE CASCADE
);

```

```

CREATE TABLE units_measure (
    id                SERIAL,

```

```
    engr_units    varchar (32)    NOT NULL,  
    description   varchar (255),  
    PRIMARY KEY (enqr_units)  
);
```

```
CREATE TABLE equip_phases (  
    id SERIAL,  
    phase         varchar (64)    NOT NULL,  
    parameter     varchar (32)    NOT NULL,  
    param_type    varchar (32)    NOT NULL,  
    engr_units    varchar (32),  
    description   varchar (255),  
    PRIMARY KEY (phase, parameter)  
);
```

B.3 Materiais

```
CREATE TABLE materials (  
    id            SERIAL,  
    material      varchar (64)    NOT NULL,  
    type          varchar (32)    NOT NULL,  
    description   varchar (255),  
    PRIMARY KEY (material)  
);
```

```
CREATE TABLE materials_loc (  
    id            SERIAL,  
    location      varchar (32)    NOT NULL,  
    material      varchar (64)    NOT NULL,  
    description   varchar (255),  
    PRIMARY KEY (location),  
    FOREIGN KEY (material)  
        REFERENCES materials(material)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE,  
    FOREIGN KEY (location)  
        REFERENCES equip_element(equipment)  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
);
```

```

CREATE TABLE materials_charac (
  id SERIAL,
  material          varchar (64)    NOT NULL,
  characteristic    varchar (64)    NOT NULL,
  default_value     varchar (32),
  engr_units        varchar (32),
  description       varchar (255),
  PRIMARY KEY (material, characteristic),
  FOREIGN KEY (material)
    REFERENCES materials(material)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  FOREIGN KEY (engr_units)
    REFERENCES units_measure(engr_units)
    ON UPDATE CASCADE
    ON DELETE SET NULL
);

```

B.4 Receitas

```

CREATE TABLE m_recipe_element (
  id                SERIAL,
  re                varchar (128)  NOT NULL,
  re_version        varchar (16)   NOT NULL,
  version_date      date,
  author            varchar (32),
  product           varchar (32),
  description       varchar (255),
  status            varchar (128),
  re_type           varchar (32),
  derived_re        varchar (128),
  derived_version   varchar (16),
  default_batch     INTEGER,
  max_batch         INTEGER,
  min_batch         INTEGER,
  engr_units        varchar(32),
  steps_count       INTEGER,
  PRIMARY KEY (re, re_version),
  FOREIGN KEY (engr_units)

```

```

        REFERENCES units_measure(engr_units)
        ON UPDATE CASCADE
    );

CREATE TABLE m_recipe_element_parameter (
    ID                SERIAL,
    re                 varchar (128)    NOT NULL,
    re_version        varchar (16)     NOT NULL,
    parameter         varchar (32)     NOT NULL,
    parameter_value   varchar (128),
    engr_units        varchar (32),
    param_type        varchar (32)     NOT NULL,
    PRIMARY KEY (re, re_version, parameter),
    FOREIGN KEY (engr_units)
        REFERENCES units_measure(engr_units)
        ON UPDATE CASCADE,
    FOREIGN KEY (re, re_version)
        REFERENCES m_recipe_element (re, re_version)
        ON UPDATE CASCADE
);

CREATE TABLE m_recipe_step (
    id                SERIAL,
    step              varchar (128)    NOT NULL,
    re                 varchar (128)    NOT NULL,
    re_version        varchar (16)     NOT NULL,
    re_type           varchar (32)     NOT NULL,
    parent_re         varchar (128)    NOT NULL,
    parent_version    varchar (16)     NOT NULL,
    m_recipe           varchar (128)    NOT NULL,
    m_recipe_version  varchar (16)     NOT NULL,
    href              varchar (255),
    back_color        varchar (128),
    tree_priority     INTEGER,
    parent_id         INTEGER,
    parent_step       varchar (128)    NOT NULL,
    PRIMARY KEY (step, re, re_version, m_recipe, m_recipe_version,
    parent_step),
    FOREIGN KEY (re, re_version)
        REFERENCES m_recipe_element (re, re_version)

```

```

        ON UPDATE CASCADE,
FOREIGN KEY (m_recipe, m_recipe_version)
REFERENCES m_recipe_element (re, re_version)
ON UPDATE CASCADE
);

CREATE TABLE m_recipe_link (
    id                SERIAL,
    re                varchar (128)    NOT NULL,
    re_version        varchar (16)     NOT NULL,
    from_element      varchar (128)    NOT NULL,
    to_element        varchar (128)    NOT NULL,
    m_recipe          varchar (128)    NOT NULL,
    m_recipe_version  varchar (16)     NOT NULL,
    parent_step       varchar (128)    NOT NULL,
    PRIMARY KEY (from_element, to_element, m_recipe, m_recipe_version,
parent_step),
    FOREIGN KEY (re, re_version)
REFERENCES m_recipe_element (re, re_version)
ON UPDATE CASCADE,
    FOREIGN KEY (m_recipe, m_recipe_version)
REFERENCES m_recipe_element (re, re_version)
ON UPDATE CASCADE
);

CREATE TABLE m_recipe_step_parameter (
    id                SERIAL,
    step              varchar (128)    NOT NULL,
    re                varchar (128)    NOT NULL,
    re_version        varchar (16)     NOT NULL,
    parameter         varchar (32)     NOT NULL,
    parameter_value   varchar (128),
    engr_units        varchar (32),
    param_type        varchar (32)     NOT NULL,
    m_recipe          varchar (128)    NOT NULL,
    m_recipe_version  varchar (16)     NOT NULL,
    parent_step       varchar (128)    NOT NULL,
    PRIMARY KEY (step, re, re_version, parameter, m_recipe,
m_recipe_version, parent_step),
    FOREIGN KEY (engr_units)

```

```
REFERENCES units_measure(engr_units)
ON UPDATE CASCADE,
FOREIGN KEY (re, re_version)
REFERENCES m_recipe_element (re, re_version)
ON UPDATE CASCADE,
FOREIGN KEY (m_recipe, m_recipe_version)
REFERENCES m_recipe_element (re, re_version)
ON UPDATE CASCADE
);
```

```
CREATE TABLE m_recipe_element_equip (
  id SERIAL,
  re varchar (128) NOT NULL,
  re_version varchar (16) NOT NULL,
  equipment varchar (32) NOT NULL,
  PRIMARY KEY (re, re_version, equipment),
  FOREIGN KEY (re, re_version)
REFERENCES m_recipe_element
ON UPDATE CASCADE,
FOREIGN KEY (equipment)
REFERENCES equip_element(equipment)
ON UPDATE CASCADE
);
```

```
CREATE TABLE m_recipe_approvals (
  id SERIAL,
  type varchar (128) NOT NULL,
  author varchar (128) NOT NULL,
  level varchar (128),
  m_recipe varchar (128) NOT NULL,
  m_recipe_version varchar (16) NOT NULL,
  approval_log timestamp NOT NULL,
  description varchar (255),
  FOREIGN KEY (author)
REFERENCES users(username)
ON UPDATE CASCADE,
FOREIGN KEY (m_recipe, m_recipe_version)
REFERENCES m_recipe_element (re, re_version)
ON UPDATE CASCADE
);
```

Referências

- [1] *Batch Control: Models and Terminology (Pt.1)*. Isa, 1995.
- [2] *Batch Control: Data Structures and Guidelines for Languages ISA-88.00.02-2001 Pt. 2*. Instrument Society of America, 2001.
- [3] JPM. Empresa, 2018. Disponível em <https://www.jpm.pt/pt/empresa/>, acessado a última vez em 30 de Abril de 2018.
- [4] JPM. Indústrias, 2018. Disponível em <https://www.jpm.pt/pt/industria/>, acessado a última vez em 30 de Abril de 2018.
- [5] JPM. Soluções, 2018. Disponível em <https://www.jpm.pt/pt/solucoes/>, acessado a última vez em 30 de Abril de 2018.
- [6] APICS. *APICS Dictionary*. APICS - The Association for Op, 2008.
- [7] CCPS (Center for Chemical Process Safety). *Guidelines for Safe Automation of Chemical Processes*, chapter 6. Wiley-AIChE, 2016.
- [8] Joel Haight. Automated control systems do they reduce human error and incidents? 05 2018.
- [9] Luís Almeida. Industrial computing architectures, 2016. Acetatos Unidade Curricular Arquiteturas de Computação Industrial, Ano Letivo 2015/2016. FEUP.
- [10] ISA. Isa88, batch control, 2018. Disponível em <https://www.isa.org/isa88/>, acessado a última vez em 01 de Maio de 2018.
- [11] EduQuest. Eduquest comparison of part 11 to annex 11, 2018. Disponível em <http://www.eduquest.net/Advisories/Comparison%20of%20FDA%20Part%2011%20and%20EU%20Annex%2011.pdf>, acessado a última vez em 02 de Maio de 2018.
- [12] Schneider Electric. Wonderware inbatch 2012 r2, 2012. Disponível em <http://software.schneider-electric.com/pdf/datasheet/wonderware-inbatch-2012-r2/>, acessado a última vez em 02 de Maio de 2018.
- [13] Schneider Electric. Wonderware recipe manager plus, 2015. Disponível em <http://software.schneider-electric.com/pdf/datasheet/wonderware-recipe-manager-plus/>, acessado a última vez em 02 de Maio de 2018.
- [14] Rockwell Automation. Factorytalk® batch, 2017. Disponível em <http://literature.rockwellautomation.com/idc/groups/literature/>

- [documents/pp/proces-pp017_-en-p.pdf](#), acessado a última vez em 02 de Maio de 2018.
- [15] Iconics. Recipe + batch management, 2018. Disponível em <https://www.iconics-uk.com/solutions/hmi-scada/batching-recipe-management>, acessado a última vez em 02 de Maio de 2018.
- [16] MAR-KOV. Mar-kov recipe manager pro, 2017. Disponível em <http://www.mar-kov.com/?show=rmpro>, acessado a última vez em 02 de Maio de 2018.
- [17] NovaTech. Flexbatch - batch and recipe management, 2016. Disponível em https://www.novatechweb.com/wp-content/uploads/2012/02/OV_FlexBatch_082416_website-version.pdf, acessado a última vez em 02 de Maio de 2018.
- [18] Simple Programmer. What is web development? - a comprehensive view of web development, Dec 2017. Disponível em <https://simpleprogrammer.com/what-is-web-development/>, acessado a última vez em 03 de Maio de 2018.
- [19] Smarty Template Engine. All about smarty. Disponível em https://www.smarty.net/about_smarty, acessado a última vez em 11 de Maio de 2018.