



IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

ESTÁGIO NO GRUPO VALGRUPO
Desenvolvimento e testes de seguranças às plataformas do Grupo

ESTUDANTE GONÇALO MIGUEL CONCEIÇÃO VICENTE

Leiria, Março de 2024



IPL

escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

ESTÁGIO NO GRUPO VALGRUPO

ESTUDANTE GONÇALO MIGUEL CONCEIÇÃO VICENTE
Número: 2210510

Estágio realizado sob orientação do Professor Doutor Carlos Manuel Gonçalves Antunes (carlos.antunes@ipleiria.pt).

Leiria, Março de 2024

AGRADECIMENTOS

Gostaria de agradecer ao meu orientador, Professor Doutor Carlos Manuel Gonçalves Antunes pelo acompanhamento e conhecimento transmitido que me encorajaram durante todo o processo de estágio.

Deixo também uma nota de agradecimento à empresa Valsabor S.A. por me ter proporcionado a realização do estágio, aos meus familiares, amigos e colegas de trabalho que de alguma forma me ajudaram e acompanharam nesta etapa académica.

RESUMO

O presente documento descreve a realização de um estágio em contexto de trabalho no âmbito do curso de Mestrado em Cibersegurança e Informática Forense, realizado no Instituto Politécnico de Leiria, e foi efetivado na empresa Valsabor, S.A. que pertence ao grupo Valgrupo que tem como objeto de trabalho principal a produção de alimentos para animais, produção de suínos e perus e o abate dos mesmos em matadouros especializados.

Durante o período em que funcionou o estágio curricular a minha área de responsabilidade foi no departamento de informática do grupo, nomeadamente na área do desenvolvimento em virtude de conhecimentos adquiridos na Licenciatura em Engenharia Informática do Politécnico de Leiria que frequentei durante os anos 2017 a 2021. Durante todo o processo de estágio foquei-me essencialmente no desenvolvimento *web*, testes e implementação do desenvolvido bem como apoio na migração dos cenários de produção e de pré-produção da Google Cloud para um ambiente local.

Ao longo de todo o estágio em contexto de trabalho conseguiu-se implementar melhorias tanto a nível de sistemas, portal *web* interno bem como aperfeiçoamento de processos que já existiam como daqueles que foram implementados e aprimorados durante todo o processo, bem como soluções transversais de segurança começando na implementação do *datacenter*, passando pela segurança da plataforma e terminando na garantia de segurança dos acessos.

ABSTRACT

This document describes the completion of an internship in a work context within the scope of the Master's degree in Cybersecurity and Forensic Informatics, carried out at the Instituto Politécnico de Leiria, and was carried out at the company Valsabor, S.A., which belongs to the Valgrupo group whose purpose is main work in the production of animal feed, production of pigs and turkeys and their slaughter in specialized slaughterhouses.

During the period in which the curricular internship took place, my area of responsibility was in the group's IT department, namely in the area of development due to the knowledge acquired in the Degree in Computer Engineering at the Polytechnic of Leiria that I attended during the years 2017 to 2021. During the entire internship process, I focused essentially on web development, testing and implementation of what was developed, as well as supporting the migration of production and pre-production scenarios from Google Cloud to a local environment.

Throughout the internship in a work context, it was possible to implement improvements both at the systems level, internal web portal as well as improvement of processes that already existed and those that were implemented and improved throughout the process, as well as transversal security solutions starting at the implementation of the datacenter, going through platform security and finishing ensuring access security.

ÍNDICE

Agradecimentos	i
Resumo	iii
Abstract	v
Índice	vii
Lista de Figuras	ix
1 A empresa	1
2 Contexto do Estágio	5
3 Trabalho Realizado	13
3.1 Metodologia de Desenvolvimento	13
3.1.1 Metodologias de Desenvolvimento com Plano	13
3.1.2 Metodologias de Desenvolvimento Ágeis	14
3.2 Metodologia Utilizada	16
3.2.1 <i>Scrum</i>	16
3.2.2 <i>Kanban</i>	17
3.3 <i>Ferramentas utilizadas</i>	17
3.3.1 <i>Jira</i>	18
3.3.2 <i>Bitbucket</i>	18
3.4 <i>Implementação</i>	19
3.4.1 <i>Tecnologias</i>	19
3.4.2 <i>Linguagens de Programação</i>	20
3.4.3 <i>API</i>	22
3.4.4 <i>Base de dados</i>	22
3.4.5 <i>Frameworks</i>	23
3.4.6 <i>Datacenter e Hardware</i>	26
3.4.7 <i>Projetos</i>	28
4 Implementação e testes realizados	33
4.1 Metodologias e Ferramentas utilizadas nos testes	33
4.1.1 Hardware	33
4.1.2 Desenvolvimento de aplicações Web	36
4.2 Testes de Segurança	42
4.3 Resultados	45

ÍNDICE

5	Conclusão	49
5.1	Trabalhos Futuros	49
6	Bibliografia	51
	Declaração	53

LISTA DE FIGURAS

Figura 1	Valgrupo: O Grupo	2
Figura 2	Metodologias de desenvolvimento com plano	14
Figura 3	Metodologias de desenvolvimento ágeis	15
Figura 4	Scrum	17
Figura 5	Jira	18
Figura 6	Bitbucket	19
Figura 7	Linguagens Compiladas vs Linguagens Interpretadas	20
Figura 8	Modelo Lógico Gestão Contratos Matérias Primas	30
Figura 9	Modelo Lógico Entrada e Saída de Viaturas	32
Figura 10	Replicação entre servidores	35
Figura 11	Esquema de rede	36
Figura 12	Fluxograma Testes de Desenvolvimento	38
Figura 13	Níveis de testes	39
Figura 14	Adição do campo do lote	39
Figura 15	Janela de inserção de dados pelas fábricas	40
Figura 16	Resultado da ferramenta <i>NIKTO</i> através da rede interna	43
Figura 17	Resultado da ferramenta <i>NIKTO</i> através da rede externa	43
Figura 18	Resultado da ferramenta <i>DIRB</i> através da rede interna	46
Figura 19	Resultado da ferramenta <i>DIRB</i> através da rede externa	47
Figura 20	Resultado da ferramenta <i>ZaProxy</i>	48

A EMPRESA

O presente relatório de estágio enquadra-se num estágio do Mestrado em Cibersegurança e Informática Forense do Instituto Politécnico de Leiria e referente ao ano letivo 2022/2023. O estágio referido foi elaborado na empresa Valsabor S.A.

A empresa Valsabor S.A. enquadra-se no grupo de empresas denominado Valgrupo. Este grupo de empresas opera nas mais diversas áreas de negócio tendo como principal foco a área da produção animal e abate dos mesmos em matadouros próprios.

O Valgrupo [14] teve início no ano de 1989 e a sua atividade centra-se na produção animal: suínos, bovinos e aves, e no abate, transformação e comercialização de produtos alimentares. Conta com um centro de inseminação artificial, distribuição de medicamentos veterinários, produção de rações e distribuição de todos os produtos e matérias-primas. O grupo atua ainda em outros setores como a construção de infraestruturas, produção e comercialização de vinhos e realização de eventos.

O Valgrupo conta com cerca de 1500 colaboradores diretos e 400 indiretos, contém um conjunto de 32 empresas, sendo destas 23 próprias e 9 participadas. Todas estas empresas encontram-se organizadas em grupos de empresas. Assim sendo, a Valnutri é o grupo de empresas de onde fazem parte as nove fábricas de ração que alimentam o grupo, produzindo anualmente 530 000 toneladas de ração. A Valciaro é um centro de inseminação artificial de suínos que fornece todas as empresas que compõem a Valpor, que representa o conjunto de empresas relacionadas com a produção animal (15 empresas) que contém 350 explorações de criação de animais que origina uma criação anual de 1 600 000 suínos, 400 bovinos e 750 000 perus. De forma a transportar os animais e os produtos das diversas áreas de negócio existe a ValTTL, que é composta por uma frota de 320 viaturas divididas entre veículos ligeiros e pesados de mercadorias. De forma a gerir os investimentos imobiliários do grupo existe a ValMob, composta por duas empresas imobiliárias. A ValWine dedica-se à produção e comercialização de vinhos através da empresa Quinta da Atela. A Valcivil é o grupo de empresas dedicadas à construção civil, que faz obras de manutenção nas várias explorações de criação de animais. Existe ainda uma vertente do grupo direcionada para a zootécnica, análises e medicamentos veterinários.

De forma a garantir o abate dos animais produzidos, o grupo possui 4 matadouros espalhados pela zona centro e sul de Portugal. Nesse sentido, existe o matadouro da Valsabor, no concelho de Santarém, que tem uma capacidade de abate de suínos de 250 animais e de desmancha de 350 animais por hora; o matadouro da Maporal, no concelho de Reguengos de Monsaraz, é o mais moderno matadouro nacional e dedica-se à exportação de carne de suínos, principalmente para a China, e tem uma capacidade de abate e desmancha de 350 suínos por hora; o matadouro da Raporal, no concelho do Montijo, dedica-se ao abate de suínos e bovinos, tendo a capacidade para abater e desmanchar 200 suínos por hora e de abater 400 bovinos; o matadouro da Nutriaves, no concelho de Óbidos, dedica-se ao abate de perus, tendo capacidade de abater e desmanchar 500 animais por hora (figura 1).



Figura 1: Valgrupo: O Grupo

Com todas estas áreas de negócio o grupo tem um volume de negócios de 750 milhões de euros.

A nível do meu estágio, este decorreu na sede do grupo, nas instalações da Valsabor, onde existe o departamento informático do grupo. Este departamento é composto por 20 elementos divididos pelas áreas de hardware, software e implementação, redes

e desenvolvimento de software. Assim, a área do hardware é composta por 6 elementos, software e implementação 5 elementos, redes 1 elemento e desenvolvimento de software 8 elementos.

A área do hardware é uma área que abrange um variado número de sub-áreas, nomeadamente computadores, telemóveis, balanças, controlos de acesso, relógios de ponto, pistolas HENNESSY (utilizadas para medição da percentagem da carne animal). A área do software e implementação é uma área relacionada com software de gestão empresarial, nomeadamente SAP e ERP Eticadata bem como softwares utilizados em áreas de produção, tal como TraceOne, Flow e Inova utilizados em matadouros e MultiDos utilizado em fábricas de ração.

A área das redes é responsável por garantir que são aplicadas as melhores práticas de segurança nos vários servidores do grupo. Uma vez que todas as empresas do grupo se encontram interligadas com a sede via uma VPN Central, os servidores que suportam as diversas empresas encontram-se localizados na Valsabor, que possuiu uma linha de redundância de fibra e backups de rede 4G, contudo existem servidores em mais dois pontos estratégicos do grupo de forma que, caso exista uma falha de comunicação com a sede, sejam levantados esses servidores e que todo o grupo não fique sem sistema informático.

A área do desenvolvimento de software tem como principal foco a programação de um portal interno do grupo, a elaboração de uma aplicação móvel que é alimentada com dados do portal, contudo com a particularidade de funcionar offline guardando os dados em base de dados local e quando detetar que está com ligação à internet sincroniza os dados. Existe ainda uma vertente da área de desenvolvimento que se dedica à implementação de customizações no ERP Eticadata que permite melhorar a usabilidade do mesmo e diminuir o erro humano bem como a integração de documentos no ERP através do portal, uma vez que o grupo tem como objetivo tornar o portal o seu sistema mandatário de trabalho ficando apenas os ERP's para tratamento de dados contabilísticos.

Aquando do início do estágio curricular na empresa Valsabor S.A., eu já me encontrava na empresa como colaborador à cerca de um ano, o que me permitiu já estar enquadrado com a empresa, reduzindo o tempo necessário de adaptação à empresa e a todo o seu meio envolvente.

Durante o período em que funcionou o estágio curricular encontrei-me inserido no departamento de informática do grupo, nomeadamente na área do desenvolvimento em virtude de conhecimentos adquiridos na Licenciatura em Engenharia Informática

A EMPRESA

do Politécnico de Leiria que frequentei durante os anos 2017 a 2021.

CONTEXTO DO ESTÁGIO

A equipa de desenvolvimento em que estou inserido era composta por 3 elementos que faziam desenvolvimento através da *framework Laravel* [13] para um portal web, interno do grupo, direcionado para a gestão de todo o processo de criação de animais nas diversas explorações de animais. Durante a realização de todo o estágio, a equipa de desenvolvimento cresceu para 8 elementos no total e o portal interno do grupo que era dedicado à parte animal passou a ser expandido para outras áreas de negócio da empresa.

Aquando do início do estágio o projeto onde fui inserido era já um projeto muito importante para o grupo na sua área de negócio a produção animal, contudo o objetivo seria por tornar este projeto o projeto estrutural e uma ferramenta de trabalho diária para todas as outras áreas de negócio. Contudo, o projeto até então tinha sido desenvolvido na sua totalidade por uma empresa externa, tendo passado a continuação do desenvolvimento para o departamento informático do grupo cerca de um mês antes, o que representava uma dificuldade extra pelo desconhecimento de toda a equipa de desenvolvimento face ao que se encontrava desenvolvido. Tendo em conta esse desconhecimento, foi necessário analisar todo o código existente de forma a garantir que se enquadrava nos padrões de segurança pelo qual o grupo se quer pautar.

A nível de alojamento, este portal, estava alojado nos servidores da *Google Cloud*, porém numa forma de reduzir custos e aumentar o tempo de resposta da aplicação em algumas das integrações com softwares do grupo tinha-se como objetivo transferir o alojamento do mesmo para os nossos servidores, que se encontram presentes nas instalações da Valsabor, contudo mantendo uma camada de segurança necessária a um projeto desta dimensão, que naquela fase ainda não existia.

Tendo em conta que já era funcionário da empresa e já tinha contactado com o portal, apesar de não ter interferido diretamente no seu desenvolvimento, suspeitava que existiam problemas no mesmo, nomeadamente o facto de se usar uma versão da *framework Laravel* e *PHP* já sem suporte. Contudo, de forma a ser possível ultrapassar esta limitação era necessário adaptar todas as páginas existentes para uma nova versão, uma vez que o componente base à estrutura das páginas estava *deprecated*. Ao longo do estágio iria também estudar melhor outras vulnerabilidades

que poderiam existir no portal que ainda não tinham sido identificadas.

Uma vez que era necessário adaptar as cerca de 40 páginas existentes compostas por 150 tabelas de base de dados para uma nova estrutura decidiu-se implementar uma nova tecnologia de *frontend*, *VueJS* [15]. Tendo em conta as necessidades de adaptação do que se encontrava desenvolvido para o que pretendíamos vir a desenvolver, ficou decidido em reunião que todas as páginas existentes seriam migradas sofrendo uma profunda reestruturação mesmo a nível da lógica existente uma vez que existiam funcionalidades que necessitavam de alguns ajustes. Nesse sentido foi delineado um plano de qual deveria ser a ordem de reestruturação, tendo em conta que ao mesmo tempo iriam estar a ser desenvolvidas novas funcionalidades da aplicação, contudo já seriam desenvolvidas utilizando a nova ideia de desenvolvimento.

Neste sentido, teriam de ser alocados recursos aos novos projetos que eram necessários e urgentes de implementar bem como reestruturar tudo aquilo que estava a impedir migrar a plataforma para uma nova versão.

Com esta visão em mente, foi feito um levantamento das necessidades a desenvolver de forma a se identificar onde deveriam ser alocados os recursos existentes. Tendo em conta a abrangência do grupo, foram desenvolvidas reuniões junto de vários setores do grupo para se perceber quais as suas necessidades mais urgentes. Dessas reuniões surgiram um conjunto de 10 prioridades que deveriam ser tidas em conta no desenvolvimento do ano seguinte.

Da reunião com o departamento de compras de matérias-primas para toda a área da Valnutri, foi identificada a necessidade de informatizar a gestão de contratos de matérias-primas, uma vez que o processo existente era um processo bastante manual e falível, levantando dúvidas acerca da fiabilidade dos dados. Nesse sentido, foi proposto desenvolver uma plataforma onde seriam lançados os contratos existentes com os fornecedores externos com as quantidades contratadas e durante o intervalo de tempo que se encontrava válido, depois as fábricas iriam solicitar as suas necessidades de matérias-primas através do nosso portal interno e, a equipa do departamento de compras de matérias-primas faria a correta associação ao contrato existente garantido que seriam cumpridos os preços de compra/venda acordados. Estando feito o planeamento das cargas de matérias-primas é gerada uma encomenda de fornecedor no ERP, e as mesmas seriam recebidas diretamente na fábrica através do seu software próprio (MultiDos) e importadas para o ERP utilizado pelo grupo (Eticadata), uma vez que este mecanismo já estava implementado. Do software Eticadata, os documentos de compra da matéria-prima seriam importados para o portal Gestão Valgrupo de forma a serem finalizados pelo departamento de

compras uma vez que seria necessário confirmar, com base na fatura do fornecedor as quantidades e ou preços que deveriam ser debitados, sendo que após estar tudo em conformidade seria confirmada a receção, integrando uma guia de remessa de fornecedor no software Eticadata e mais tarde, integrando uma fatura de fornecedor de forma a lançar a mesma para pagamento. Através do portal seriam também calculados os valores pagos a transportadores com base no peso por tonelada/distância entre o lugar de carga e descarga. Com este desenvolvimento, deixaria de ser um processo completamente manual e falível e passava a ser um processo automatizado que procura diminuir o erro humano e facilitar a introdução de dados contabilísticos no ERP Eticadata através da integração direta de documentos ficando com a rastreabilidade das ações ocorridas durante aquele processo.

Uma outra área que tivemos em conta neste levantamento de necessidades foi a ValTTL, pela importância que tem na logística das diversas áreas do grupo, uma vez que são os responsáveis pela maior parte dos transportes efetuados. Foi realizada uma reunião com o departamento de frota onde se chegou à conclusão que deveria ser elaborada uma aplicação móvel onde seriam apresentados os diversos fretes que seriam feitos pelos motoristas ao longo do dia, de forma a lhes apresentar, de uma forma informatizada, o planeamento diário de tarefas e onde, os motoristas seriam responsáveis por confirmar as quantidades que estavam a transportar entre determinadas origens e destinos, devendo a aplicação trabalhar de forma offline, uma vez que em diversas explorações a rede móvel é algo limitada. A aplicação deveria ainda permitir associar anexos e tirar fotografias de forma a ser possível anexar as guias de remessa, talões de pesagem e CMR de forma a existir uma rastreabilidade completa da carga. Uma vez que os carros de ração são divididos no seu interior por tanques (compartimentos individuais) o seu planeamento é feito tanque a tanque onde cada um pode ter referências de ração diferentes, que não podem ser misturadas. Tendo em conta este cenário, a aplicação deveria permitir ao utilizador realocar a ordem da carga entre os diversos tanques do carro, respeitando sempre a quantidade aprovada pela equipa do planeamento.

Da reunião com o departamento de frota foi identificada ainda uma outra necessidade, desta feita, a requisição de viaturas por parte dos diversos colaboradores do grupo. Esta requisição advém da necessidade de deslocações entre unidades ou substituição de viaturas enquanto as viaturas alocadas se encontram em manutenção. Esta área era uma área que estava com um grande descontrolo, uma vez que as requisições eram feitas com uma antecedência muito curta dificultado o trabalho do departamento e os débitos às empresas contratantes dos funcionários que requisitaram a viatura não estavam a ser controlados corretamente. De forma a informatizar o processo deveria ser elaborada uma janela de requisição da viatura aberta para

os funcionários com base num local de recolha de viatura e num local de destino. Assim, seria possível alocar o máximo número de pessoas para o mesmo destino no número mínimo de veículos necessários. Deveria ser possível extrair uma listagem de origens/destinos e qual a empresa detentora do veículo bem como a empresa contratante do utilizador que requisitou o veículo de forma a fazer a faturação correta.

Uma outra área a ter em conta é a área da produção de carne. Nesse sentido, foi também elaborada uma reunião com a gestão do matadouro da Valsabor de forma a identificar quais as áreas mais necessitadas de intervenção informática. Uma área que chamou à atenção é as diferenças entre os animais que os produtores dizem ter sido enviados e aqueles que chegam à fase final do processo de abate. Sendo assim, foi sugerido implementar contadores de animais ao longo da linha de forma a existir o número de animais corretos. Assim foram identificados dois sítios estratégicos onde deveriam ser contados a quantidade de ferros que erram movimentados ao longo da linha. Esses sensores deveriam ser específicos para ferro de forma a serem mais assertivos e deveriam ter a possibilidade de identificar quando a contagem ia no sentido positivo ou quando teria de ser um sentido negativo, de forma a decrementar animais caso a linha estivesse a andar no sentido inverso. De forma a fazer a contagem exploração a exploração, deveria ser desenvolvida uma janela onde seriam lançadas as explorações que iriam enviar animais bem como a ordem de abate dos mesmos e os operadores da linha através de uma janela simples seriam responsáveis por clicar num botão após a passagem do último animal de cada exploração.

Uma das preocupações do departamento de produção de carne seria a entrada, pesagens e saída de viaturas. Uma vez que nas empresas de produção de carne todo o veículo tem de ser pesado de forma a tentar evitar erros no momento de expedição, assim, o valor da pesagem após a carga tem de ser semelhante ao total dos documentos que irão acompanhar a carga. Caso exista uma diferença relativa, o processo de carregamento deve ser repetido de forma a identificar e corrigir o erro. Contudo, o processo que existia não permitia que fosse efetuada esta gestão de forma automática sendo um processo manual onde o porteiro retirava o peso do programa de pesagem e colocava num ficheiro excel para verificar as diferenças existentes. De forma a automatizar este processo seria necessário o portal conseguir comunicar com as básculas existentes e obter o peso destas de forma a registá-lo. Seria também necessário a comunicação com impressoras de talões para que fossem impressos os talões de pesagem que comprovam a pesagem. Este processo deveria ser alargado para todas as outras unidades de forma a uniformizar processos.

Uma outra necessidade do departamento de produção de carne surgiu mais tarde,

com a aquisição de uma nova empresa para o grupo. A empresa Raporal S.A., foi adquirida em novembro de 2022 e tem na sua constituição um matadouro no Montijo que tinha processos muito manuais tendo como objetivo implementar o mesmo sistema de controlo de produção que existe na Valsabor. Contudo, esse desenvolvimento teria de ser feito por uma entidade externa algo que tornou um processo mais moroso e nesse sentido, foi desenvolvido uma aba no portal para expedição da carne existente com a semelhança do que se pretendia vir a implementar, contudo numa vertente mais simplista. Foi desenvolvido uma janela onde seriam lançadas as encomendas dos clientes e uma outra janela de pesagem com a adição de leituras através de códigos de barras das etiquetas dos produtos que já tinham sido trabalhados. Após esta leitura de códigos de barras e recolha de pesos seria emitida uma etiqueta final com o peso total dos produtos.

Uma área bastante importante para o grupo é área dos medicamentos para as explorações de animais, uma vez que representam um custo significativo e que é importante ser controlado tanto por questões internas como externas. Nesse sentido reunimos com o departamento veterinário que nos alertou para a dificuldade de fazer chegar a todas as explorações os medicamentos mais atualizados ao dia atual, uma vez que as medicações vão mudando ao longo dos tempos. Assim, foi proposto desenvolver um separador no portal no onde os responsáveis das explorações deveriam solicitar quais os medicamentos que pretendiam bem como as quantidades dos mesmos, devendo os produtos apresentados ser baseados numa lista elaborada pelo departamento veterinário, podendo ainda ser feito um filtro com base no tipo de animal e exploração para restringir medicamentos específicos para certas explorações. Deveria existir também a possibilidade de dizer quais as quantidades que devem ser as ideais para ser encomendadas e caso seja uma quantidade superior, deve ser direcionado para a janela de gestão do veterinário de forma a este aprovar ou não essa requisição. Na janela de aprovação do veterinário serão, também, apresentados todos os medicamentos classificados como antibióticos de forma a este os aprovar. De seguida, os medicamentos que necessitam de receita médica serão direcionados para uma janela onde será emitida a receita e só depois ficaram disponíveis para serem expedidas pelo fornecedor. Este desenvolvimento irá facilitar a gestão de encomendas de medicamentos uma vez que a lista disponível será sempre a mais atualizada, deixando de existir dificuldade de fazer chegar às explorações a listagem de artigos que deve ser tida em conta, os médicos veterinários conseguem ir recebendo automaticamente quais os produtos que tem de ser aprovados e as receitas médicas podem ir sendo emitidas mesmo que ainda existiam produtos pendentes de aprovação pelo médico veterinário.

Num olhar para o departamento de informática, identificamos uma necessidade

que seria a gestão dos equipamentos atribuídos aos funcionários, uma vez que as atribuições eram registadas através de um processo manual existindo várias falhas. Nesse sentido, decidimos que seria importante a elaboração de uma janela onde fosse possível registar equipamentos para stock e a partir daí os mesmos serem atribuídos aos diversos funcionários do grupo, sendo que o portal deve gerar a informação de faturação entre as diversas empresas. Devia ser possível também gerar documentos de atribuição de equipamentos (a serem assinados pelo responsável do mesmo) bem como etiquetas identificativas dos equipamentos para uma melhor catalogação dos mesmos. Este desenvolvimento viria melhorar a atribuição e gestão de equipamentos dos utilizadores a fim de saber o que cada um tem em sua posse e, no momento da demissão, qual o valor a ser cobrado face a equipamentos que foram atribuídos e não são devolvidos.

A nível da parte da assistência informática, sentiu-se a necessidade de elaborar uma janela para a gestão de *tickets* de assistência. Até então a gestão desses *tickets* era feita de forma direta no ERP Eticadata, contudo com o aumento da equipa de assistência e no número de *tickets* (média de 1000 *tickets* mensais) essa tarefa ficou bastante dificultada. Assim, deveria desenvolver-se uma janela simples onde seriam apresentados os *tickets* e onde poderiam ser alocados aos respetivos técnicos e onde seriam registados os tempos gastos em cada intervenção, sendo depois estes dados inseridos em ordens de reparação no ERP Eticadata. Os técnicos deveriam ter ainda um calendário onde aparecia o seu planeamento de tarefas com base na prioridade definida pelo gestor da equipa de assistência.

O portal Gestão Valgrupo tinha sido desenvolvido anteriormente no âmbito da Gestão Pecuária, contudo era necessário migrar essas funcionalidades para a nova versão utilizada. Contudo não se tratava apenas de um processo de migração, mas também de ajuste face à realidade existente, uma vez que existiam funcionalidades desenvolvidas que já não se enquadravam. Nesse sentido reunimos com o departamento de gestão e controlo onde apresentamos a nossa proposta de reestruturação e onde nos foi identificado o que deveria ser alterado face ao que existia desenvolvido. Dividimos o processo de reestruturação em várias fases, começando a reestruturar a parte da criação dos lotes de animais, uma vez que era a parte inicial da gestão dos animais. De seguida, seria necessário reestruturar os pedidos de saídas de animais das diversas explorações para outras explorações ou para matadouros. Na terceira fase de desenvolvimento foi reformulado tudo o que tinha a ver com o planeamento das saídas dos animais, a confirmação dos movimentos entre explorações e matadouros e a confirmação das cargas efetuadas. A quarta fase desta reformulação seria a parte da faturação dos animais e dos transportes, esta parte com integração direta no ERP Eticadata. A quinta fase deste projeto seria trazer para o portal Gestão Valgrupo

os diversos custos que influenciam cada lote de animal, nomeadamente custos de ração, sêmen e medicamentos bem como custos gerais da exploração que deveriam ser repartidos por todos os lotes que existiam durante um período temporal. A última fase desta reestruturação seria apresentar todos os dados numa consulta onde seriam apresentados todos os custos/rendimentos de cada lote de forma a conseguir analisar os índices de produção e quais os melhores ou piores lotes de animais.

Por fim, o último projeto que tínhamos como objetivo iniciar o seu desenvolvimento seria uma plataforma de gestão das compras do grupo desde a sua fase de encomenda à fase de pagamento. Este desenvolvimento deveria ter uma componente de aprovação de documentos por empresa, existindo a necessidade de ficar bastante moldável tendo em conta a organização do grupo. O projeto da central de compras necessitaria de ter integração com o ERP do grupo em duas vertentes diferentes.

Uma das vertentes seria utilizar o portal para descarregar informação no ERP em todos os momentos de aprovação de documentos, ou seja, após aprovação da encomenda a um fornecedor, após aprovação da respetiva fatura e confirmação de que está disponível para o departamento da tesouraria proceder ao respetivo pagamento. Uma outra vertente, será para os casos em que não existe uma requisição por detrás de uma fatura, por exemplo despesas com portagens ou combustíveis em postos externos ao grupo. Nessa situação, a fatura deverá ser lançada diretamente no ERP Eticadata e depois disso será importada para o portal Gestão Valgrupo de forma a ser validada por quem se encontrar na lista de aprovadores para aquele tipo de documento.

Com base em todos estes projetos era necessário alocar os projetos aos respetivos programadores. Nesse sentido, ficou-me atribuído a gestão dos contratos de cereal em parceria com outro programador que iria fazer toda a parte de integração entre o portal Gestão Valgrupo e o ERP Eticadata, através da Web API, utilizando a linguagem de programação C#. Assim, eu seria responsável por toda a lógica do lado do portal e enviar os dados necessários para integração. De seguida, o projeto seguinte que me deveria ser alocado seria o projeto da requisição de viaturas por parte dos colaboradores da empresa. Para a área da produção de carne ficaria como responsável pelo sistema de contagem de animais na linha de abate e pela entrada, pesagem e saída de viaturas das unidades de produção. A nível do departamento de informática iria desenvolver o projeto da gestão de tickets. Por fim, ficara ainda responsável por desenvolver o projeto de gestão dos medicamentos para as explorações do grupo.

Contudo, com a adição de novos elementos ao grupo de trabalho, foi necessário reajustar os projetos que cada um dos programadores iniciais do projeto tinham

consigo de forma a ir enquadrando os novos elementos. Assim, ficou da minha responsabilidade a gestão de contratos de cereal, contudo, devido a toda a complexidade do projeto, fui auxiliado por um outro elemento no desenvolvimento de tabelas auxiliares do projeto. A requisição de viaturas também ficou da minha responsabilidade tal como previsto e o projeto da contagem de animais. Nos projetos da entrada, pesagem e saída de viaturas a minha colaboração foi reduzida para desenvolver apenas a parte da comunicação entre o portal e as básculas e impressoras de talões, ficando o resto da lógica de entradas e saídas a cargo de outro programador. No projeto da gestão de tickets, a minha participação também foi reduzida, ficando apenas da minha responsabilidade a elaboração de *views* e *stored procedures* para comunicação entre o portal e o ERP Eticadata, uma vez que neste caso não se iria utilizar a *Web API*. Fiquei ainda responsável pelo desenvolvimento do projeto de gestão dos medicamentos para as explorações do grupo.

Além destes desenvolvimentos, seria ainda da minha responsabilidade a implementação no terreno de todos os projetos desenvolvidos por mim, nomeadamente a gestão dos contratos de cereal, a requisição de viaturas, o projeto da contagem de animais e o projeto da gestão de medicamentos para o grupo.

TRABALHO REALIZADO

3.1 METODOLOGIA DE DESENVOLVIMENTO

Durante o processo de desenvolvimento de qualquer *software* existe a necessidade de adotar uma metodologia de desenvolvimento de forma a facilitar o processo inerente ao seu desenvolvimento. Uma metodologia de desenvolvimento ou processo de *software* é um conjunto de atividades estruturadas e que são necessárias para o desenvolvimento de um *software*. Existem vários processos de *software* diferentes, mas todos baseiam-se nos mesmos pontos: especificação, desenho e implementação, validação e evolução.

As metodologias de desenvolvimento dividem-se em: metodologias de desenvolvimento com plano e metodologias de desenvolvimento ágeis [10], [11]. A aplicação de uma metodologia depende do caso em análise.

3.1.1 Metodologias de Desenvolvimento com Plano

Nas metodologias de desenvolvimento com plano, todas as atividades são planeadas no início do projeto e o processo é medido por comparação com esse plano.

Esta técnica de desenvolvimento foi muito utilizada no passado, num contexto de desenvolvimento muito diferente do atual, uma vez que tendo em conta que os acessos a computadores eram limitados e não existiam ferramentas modernas de apoio ao desenvolvimento de *software*, o custo de fazer alterações e correções era muito alto. Tendo em conta esta situação, o *software* era todo planeado e documentado antes de se iniciar o processo de implementação. Uma das metodologias de desenvolvimento no plano mais utilizada é a metodologia de desenvolvimento em cascata (*Waterfall Model*), uma vez que, existem fases separadas e distintas para a especificação, desenvolvimento e validação.

Neste modelo, existem fases separadas e bem definidas para a análise e defini-

ção de requisitos, desenho de sistemas de *software*, implementação e testes unitários, integração e testes de sistema e operação e manutenção.

O uso de uma metodologia de desenvolvimento com plano apresenta vantagens e desvantagens face à metodologia de desenvolvimento ágil.

Esta metodologia apresenta como principal vantagem existência de uma boa documentação e testes unitários, uma vez que é investido muito tempo na elaboração da documentação e análise dos requisitos bem como aos testes.

A principal desvantagem deste modelo é a dificuldade em acomodar alterações depois do processo ser iniciado, pois só quando uma fase estiver totalmente completa é que se pode avançar para a fase seguinte. Assim, havendo alterações do cliente depois de uma fase já estar concluída existe uma grande quantidade de alterações a efetuar. É adequado apenas para sistemas empresariais em que se espera que as alterações sejam mínimas, porém são poucos os sistemas que têm requisitos estáveis. Além disto, este tipo de modelo deve ser usado apenas para sistemas de grandes dimensões que são desenvolvidos em vários locais, pois este modelo ajuda na coordenação do trabalho. Na figura 2 é possível verificar o esquema desta metodologia.

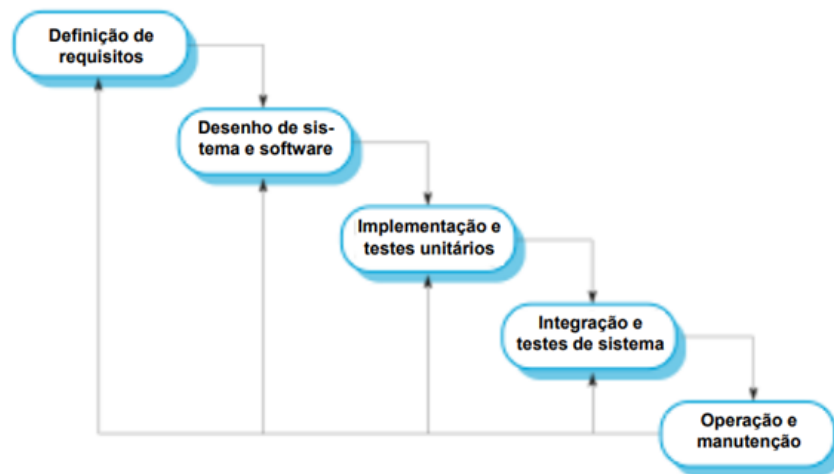


Figura 2: Metodologias de desenvolvimento com plano

3.1.2 Metodologias de Desenvolvimento Ágeis

As metodologias de desenvolvimento ágeis são metodologias onde o planeamento é feito de forma incremental o que torna mais fácil alterar o processo para refletir alterações dos requisitos do cliente ao longo do desenvolvimento do projeto.

Neste modelo, o desenvolvimento é feito de forma incremental, sendo que a especificação, desenvolvimento e validação são feitas de forma intercalada. (Figura 3)



Figura 3: Metodologias de desenvolvimento ágeis

O uso de uma metodologia de desenvolvimento ágil apresenta vantagens e desvantagens face à metodologia de desenvolvimento com plano.

Esta metodologia apresenta como principal vantagem a redução do custo face a alterações de requisitos por parte do cliente, uma vez que a quantidade de alterações que devem ser feitas é bastante inferior às alterações necessárias em modelos com plano.

Com recurso a esta metodologia, é também mais fácil obter a opinião do cliente face ao trabalho desenvolvido, uma vez que estes podem comentar eventuais demonstrações de *software* e ver partes que já se encontram implementadas. Com esta metodologia é, também, possível acelerar o prazo de entrega do *software* ao cliente.

Como desvantagens, esta metodologia apresenta o facto de existir pouca documentação, uma vez que não é eficiente estar a desenvolver documentação para todas as versões do sistema. Uma outra desvantagem é que com a adição de novos requisitos os sistemas tendem a degradar-se a menos que seja gasto tempo e dinheiro a melhorar o código.

3.2 METODOLOGIA UTILIZADA

A equipa de desenvolvimento do Valgrupo, considerou várias hipóteses de metodologias de desenvolvimento que permitissem gerir todas as fases inerentes à evolução do projeto. Tendo em conta a natureza deste projeto, escolheu-se as metodologias ágeis. Após analisar várias alternativas, optou-se por utilizar fundamentos das metodologias *Kanban* e *Scrum* [12].

De notar, que esta decisão de escolher as metodologias de desenvolvimento foi elaborada no momento que a equipa de desenvolvimento assumiu o projeto, apesar de já estar em andamento. Neste momento ainda não tinha iniciado o estágio curricular, mas estive envolvido nessa decisão.

3.2.1 *Scrum*

O *scrum* não é uma metodologia, mas sim, uma *framework*, pelo que é composta pela equipa de *Scrum*, por eventos e artefactos.

O *Scrum* é assim uma estrutura onde o desenvolvimento é feito de forma iterativa, através da realização de *sprints*, que neste caso têm a duração de quatro semanas, onde se cria um incremento do produto (*Product Increment*). No planeamento de cada *sprint* realiza-se o *Sprint Planning* onde o objetivo é planear o trabalho a ser realizado durante o próximo *sprint*. Este evento, conta com a presença de todos os envolvidos diretamente no projeto: os desenvolvedores e o *product owner*, que no contexto do projeto é o responsável do departamento de informática que faz reuniões com os diversos departamentos que pedem desenvolvimentos de forma a definir tarefas para toda a equipa de desenvolvimento. Uma vez que se estão a desenvolver projetos longos e demorados, é nesta reunião que é feita a divisão de novas subtarefas para as próximas 4 semanas. No final de cada *sprint* é realizada a *Sprint Review*, onde é apresentado o incremento desenvolvido de modo a obter *feedback*. Na reunião de *Sprint Review* é feito também o evento de *Sprint Retrospective*, onde cada desenvolvedor atribuiu uma nota a si próprio face à sua classificação no *sprint* que terminou. Diariamente, ao início do dia de trabalho é realizada a *Daily Scrum*, onde é feito um ponto de situação face ao que foi realizado no dia anterior, eventuais dificuldades e qual o plano de tarefas para o dia atual.

Ao longo do processo de desenvolvimento do projeto, a equipa teve acesso ao *Product Backlog*, que é composto por todas as funcionalidades que foram recolhidas na fase inicial do levantamento de requisitos. Estas funcionalidades estão escritas

na forma de *User Stories*, que são uma descrição de um requisito do utilizador do produto sob o ponto de vista do mesmo. Outro conceito usado foi o *Sprint Backlog*, que contém todas as *User Stories* a implementar durante um *sprint*. O *Product Increment* é a soma de todas as *User Stories* realizadas durante um *sprint*. A figura 4 apresenta em modo de esquema os pontos referidos.

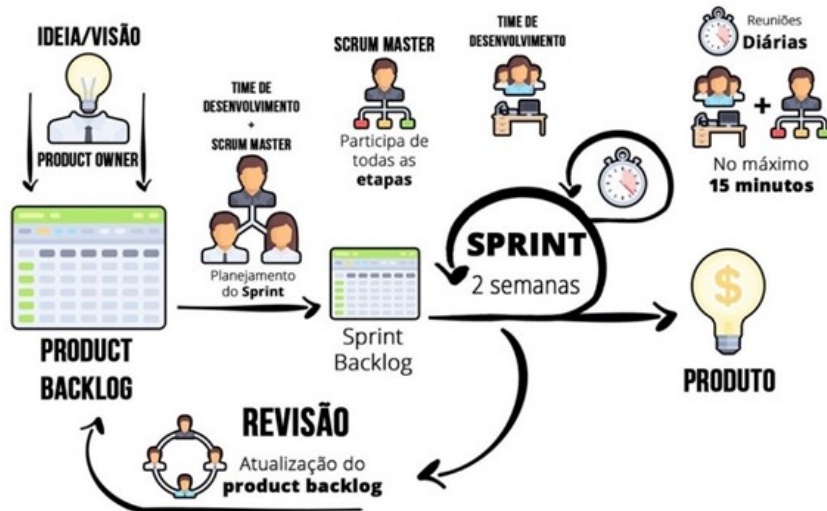


Figura 4: Scrum

3.2.2 Kanban

O *Kanban* é uma metodologia ágil para controlar a produção e gerir tarefas. Esta metodologia tem uma vertente visual, uma vez que funciona através de um esquema de colunas e cartões de forma a facilitar a visualização do que precisa de ser feito por todos os elementos da equipa de desenvolvimento. O *Kanban* é um método que tem como objetivo aumentar a produtividade e otimizar a realização de tarefas.

3.3 FERRAMENTAS UTILIZADAS

Ao longo do próximo capítulo serão apresentadas várias das ferramentas utilizadas durante o processo de estágio no Valgrupo de forma a gerir a equipa de desenvolvimento, tarefas e código.

3.3.1 Jira

Para todo o processo de gestão de tarefas a desenvolver recorreu-se à ferramenta *Jira*. Nesta ferramenta, distribui-se as *Users Stories* em 5 pontos evolutivos de desenvolvimento de forma a melhorar a visualização do processo evolutivo: a coluna “*To Do*”, onde se colocam todas *Users Stories* para se implementar no sprint decorrente, a coluna “*In Progress*” com as tarefas que se encontram a decorrer no momento atual, a coluna “*Blocked*” que representa as *Users Stories* que se encontram bloqueadas por alguma razão, a coluna “*Code Review*”, que representa as *Users Stories* já desenvolvidas mas que se encontram a ser analisadas por parte da equipa de desenvolvimento de forma a que possam ser dadas sugestões de melhoria ao código implementado e a coluna “*Done*” que representa todas as *Users Stories* que já passaram por este processo e estão prontas para entregar ao cliente final. Existe ainda um conjunto de tarefas que devem ser desenvolvidas nos próximos *sprints*, e a isso chamamos “*Backlog*”. A figura 5 apresenta o quadro do *Jira* da equipa de desenvolvimento do Valgrupo.

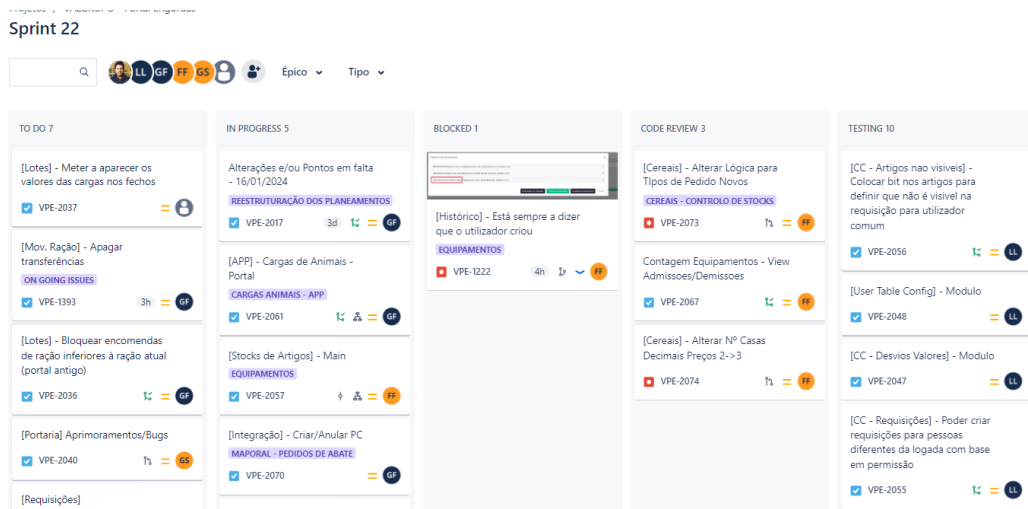


Figura 5: Jira

3.3.2 Bitbucket

Para a gestão do código do projeto, recorreu-se à ferramenta *Bitbucket*. Esta ferramenta possuiu uma boa interação com a ferramenta *Jira* e é nela que são feitos os vários *pull requests* que representam a fase em que os diversos elementos da equipa de desenvolvimento analisam o código dos restantes de forma a dar sugestões de melhoria face ao código apresentado. É através desta ferramenta que o código

se encontra dividido em vários *branches* onde cada programador trabalha no seu *branch*. A figura 6 apresenta a janela dos *pull requests* do projeto Gestão Valgrupo.






































Summary	Created	Activity	Reviewers
 [Inventários de Ração] - Store e Update -  inventario_racao_rest_3 → inventario_racao_rest <small>Gabriel Frazão - #2660, updated 2 days ago</small>	3 days ago	 6	 +2
 [Reformulação show] - Continuação  → master <small>Rodrigo Conceição Ventura - #2677, updated 2 days ago</small>	2 days ago	No activity	 +3
 Alterado secção para faturas no tipo de pedidos de misturas  → master <small>Miguel Alexandre Borges da Silva - #2676, updated 2 days ago</small>	2 days ago	No activity	 +2
 [Items] - Socampestre, Valcivil, backend campos comuns -  VPE-1916-dev-items → VPE-1916-main-items <small>Guilherme Silva - #2661, updated 2 days ago</small>	3 days ago	 26	 +2
 Frontend Changes  → artigos-main-copy <small>Fábio Faria - #2674, updated 2 days ago</small>	2 days ago	 8	 +2
 Backend Changes  → artigos-main-copy <small>Fábio Faria - #2675, updated 2 days ago</small>	2 days ago	 1	 +2
 VPE-2086 Agendamento e Farmacos  → master <small>Fábio Faria - #2671, updated 2 days ago</small>	2 days ago	No activity	 +2
 Animal billing mid  → animal-billing-main <small>Pedro Rainho - #2647, updated 3 days ago</small>	8 days ago	 99	 +2
 [Planeamentos] - Calcular os animais máximos no carro com base nos m2 (por espécie) -  validar_max_capacidade_veiculos → master <small>Gabriel Frazão - #2613, updated 2024-03-06</small>	17 days ago	No activity	 +2
 Alterações Portaria -  no merge até ser testado  → master <small>Guilherme Silva - #2619, updated 2024-03-05</small>	16 days ago	 1	 +3

Figura 6: Bitbucket

3.4 IMPLEMENTAÇÃO

Ao longo da presente secção serão apresentadas as tecnologias utilizadas durante o processo de estágio no Valgrupo, apresentando as tecnologias utilizadas, as linguagens de programação, a API, a base de dados e as frameworks utilizadas.

Neste capítulo serão também enumerados um conjunto de projetos, apresentando o seu contexto e o modelo lógico dos mesmos.

3.4.1 Tecnologias

Em todo o desenvolvimento de um projeto existe um conjunto de dados fundamentais ao seu funcionamento. A escolha da linguagem de programação, dos servidores que irão suportar todos os acessos, do motor de base de dados que irá guardar todos os dados. Contudo, como este era um projeto que já se encontrava em andamento vários pontos importantes à estrutura base do projeto já se encontravam decididos.

Na altura em que iniciei o estágio foi uma altura em que se decidiu que, pela dimensão do projeto, alterar a tecnologia de *frontend* do projeto. Até aí, não existia

a divisão entre o *backend* e o *frontend* e com o avançar do projeto decidimos que seria a melhor altura para começar a fazer essa divisão e, gradualmente, reformular as páginas existentes para esta nova tecnologia.

3.4.2 Linguagens de Programação

Linguagem de programação [9] é uma linguagem formal em que, através de uma série de instruções, permite que seja desenvolvido um conjunto de ordens e ações consecutivas de forma a criar programas que controlem o comportamento físico e lógico de uma máquina.

Assim, o programador e a máquina comunicam através dessa linguagem de forma a especificar a forma como o *software* deve operar, como os dados devem ser transmitidos e armazenados e quais as ações que o *software* deve executar de acordo com cada circunstância.

Ou seja, a linguagem de programação é um sistema de comunicação estruturado, composto por conjunto de símbolos, palavras chave, regras semânticas e sintáticas que permitem o entendimento entre um programador e uma máquina.

Num contexto de linguagem de programação é muito comum ouvir os termos interpretado e compilado [6]. As linguagens compiladas, das quais são exemplo a linguagem *C* e *C++*, são convertidas diretamente na máquina num código de máquina para que o processador possa executar, o que faz com tendam a ser mais rápidas e eficientes face a linguagens interpretadas, contudo com o desenvolvimento da compilação *just-in-time*, essa distância vem diminuindo. As linguagens interpretadas, das quais são exemplo o *PHP* e *JavaScript*, passam por um programa linha por linha e executam cada comando. A figura 7, apresenta uma comparação face a vantagens e desvantagens destes dois tipos de linguagens.

	Linguagens Compiladas	Linguagens Interpretadas
Vantagens	Maior velocidade	Maior flexibilidade
		Interpretores executam o código
Desvantagens	Tempo de compilação	Menor velocidade
	Dependência do código binário	

Figura 7: Linguagens Compiladas vs Linguagens Interpretadas

3.4.2.1 PHP

A linguagem PHP [8] (acrônimo de *Hypertext Preprocessor* e, originalmente, *Personal Home Page*) é uma linguagem de programação voltada para o desenvolvimento de aplicações *web*, favorecendo a conexão entre os servidores e a interface do utilizador.

A linguagem PHP é uma linguagem de *script* criada para comunicações do lado do servidor e que é capaz de lidar com várias funções como recolha de formulários, gestão de ficheiros e alterações à base de dados. Esta linguagem é muito utilizada para desenvolvimento *web* uma vez que pode ser integrado num simples ficheiro de HTML ou apresenta uma vantagem em sites *WordPress*, uma vez que este é criado através desta linguagem.

Esta linguagem apresenta várias vantagens para justificar a sua utilização amplamente, uma vez que é de fácil aprendizagem e tem uma ótima documentação com funções e exemplos de utilização; é uma linguagem amplamente utilizada em sites, existindo estatísticas que indicam que setenta e nove por cento dos sites utilizam o PHP; é uma linguagem open-source, podendo ser utilizada gratuitamente; integração com vários motores de bases de dados.

3.4.2.2 JavaScript

A linguagem *JavaScript* [7] é uma linguagem de programação que permite que sejam implementados itens complexos em páginas *web*, sendo a terceira camada das tecnologias básicas para o desenvolvimento *web* sendo a segunda e primeira camada as linguagens *CSS* e *HTML*, linguagens também utilizadas neste projeto.

Assim, a linguagem *HTML* é uma linguagem de marcação utilizada para estruturar e dar significado ao conteúdo, através da definição de parágrafos, cabeçalhos, tabelas ou até mesmo vídeos e imagens. A linguagem *CSS* é uma linguagem de regras de estilo utilizada para aplicar estilo ao conteúdo *HTML* através da definição de cores e fontes ou os mais variados ajustes visuais à página. A linguagem *JavaScript* é uma linguagem de programação que permite fazer com que o conteúdo se torne dinâmico. A utilização dos três em conjunto torna estas linguagens bastante poderosas.

3.4.3 API

API [5] significa *Application Programming Interface* e no contexto de *API*, a palavra aplicação refere-se a qualquer *software* com uma função distinta. A interface pode ser pensada como um contrato de serviço entre duas aplicações, comunicando através de solicitações e respostas.

Neste sentido as *API* são mecanismos que permitem que dois componentes comuniquem utilizando um conjunto de definições e protocolos. Nesta arquitetura, a aplicação que envia a solicitação é chamada de cliente e a que envia a resposta de servidor.

Existem quatro formas diferentes pelas quais as APIs podem funcionar:

1. *APIs SOAP*: Essas *APIs* usam o *Simple Object Access Protocol* onde o cliente e o servidor trocam mensagens usando *XML*.
2. *APIs RPC*: Essas *APIs* são conhecidas como *Remote Procedure Calls* onde o cliente conclui uma função no servidor e o servidor envia a saída de volta ao cliente.
3. *APIs WebSocket*: As *APIs WebSocket* oferecem um suporte à comunicação bidirecional entre aplicações cliente e servidor.
4. *APIs Rest*: As *APIs Rest* são as mais populares e flexíveis encontradas na *Web* atualmente. Esta *API* define um conjunto de funções como *GET*, *PUT*, *DELETE*, *POST* onde os clientes podem utilizar para obter dados do servidor.

3.4.4 Base de dados

Uma base de dados é um conjunto de informações organizadas, normalmente armazenadas num servidor local ou online. É geralmente controlado por um sistema de gestão de base de dados (*SGBD*). Atualmente existem vários tipos de bases de dados, sendo elas base de dados relacionais, orientadas a objetos, não relacionais, de código aberto, na *cloud*, multimodelo, de documentos, entre outras, sendo as mais utilizadas base de dados relacionais e não relacionais.

Uma base de dados relacional é uma base de dados em que os dados são organizados em tabelas com diversas colunas e linhas. Esta fornece a maneira mais eficiente e flexível de aceder às informações de forma estruturada. Algumas bases de dados relacionais são o *MySQL*, *Amazon Aurora*, *Microsoft SQL Server* e *MariaDB*. Já numa base de dados não relacional (ou *NoSQL*) é permitido que dados não estrutu-

rados e semiestruturados sejam armazenados e alterados (ao contrário de uma base de dados relacional que define como os dados inseridos devem ser compostos). Estes tipos de base de dados tornam-se comuns em aplicações da *web* mais complexas. Algumas bases de dados não relacionais são *MongoDB*, *Redis*, *Memcached* e *Hbase*.

3.4.4.1 *MySQL*

MySQL é o software de base de dados relacional de código aberto mais popular de todos. É desenvolvido, distribuído e suportado pela *Oracle Corporation*. Este *software* é de código aberto, ou seja, qualquer pessoa pode utilizar e modificar o seu código. A linguagem utilizada para aceder a esta base de dados é o *SQL (Structured Query Language)*. Dependendo do ambiente de programação, o programador pode inserir o código *SQL* de forma a gerar, por exemplo relatórios, adicionar, remover e editar dados. Esta linguagem pode ser incorporada em outras linguagens como, por exemplo, a utilização de código *SQL* juntamente com *Visual Basic* ou usar uma linguagem *API* específica que oculta o *SQL*.

O código *SQL* é definido pelo padrão *ANSI / ISO SQL*. Este código tem vindo a evoluir ao longo dos anos, tendo a sua última versão sido lançada em 2016. O *MySQL Server* é muito rápido, confiável e fácil de utilizar. Como não necessita de muitas especificações de hardware, este pode ser executado em qualquer tipo de máquina, juntamente com outras aplicações e servidores web. Se este servidor for alojado numa máquina apenas destinada a ele, o programador pode aproveitar toda a memória RAM, potência do *CPU* e a capacidade de E/S (Entrada/Saída) disponível. O *MySQL Server* também pode ser instalado num cluster de máquinas em rede.

Em constante desenvolvimento, o servidor *MySQL* oferece, atualmente, um conjunto de funções úteis tais como a sua velocidade, conectividade e segurança que o tornam altamente adequado para aceder a bases de dados. O *software* de base de dados *MySQL* é um sistema cliente/servidor, que consiste num servidor *SQL Multithread* que suporta vários *backends*, programas e bibliotecas de clientes, ferramentas administrativas e várias interfaces de programação de aplicações (*APIs*).

3.4.5 *Frameworks*

Uma *framework* é uma estrutura na qual é possível criar um *software*. São, normalmente, associados a uma linguagem de programação específica e são adequados para diferentes tipos de tarefas. Ao utilizar uma *framework*, esta serve como base e,

assim, o programador não começa a desenvolver completamente do zero.

A utilização de uma *framework* economiza tempo e reduz o risco de erros e, além disto, apresentam outras vantagens, tais como apresentação de um código mais seguro, teste e descoberta de *bugs* mais simples, evita a duplicação de código, sendo ele limpo e mais adaptável, e poder vir a ser estendido no futuro.

Existem diferentes tipos de *frameworks*, sendo algumas delas de aplicações da *web* (*AngularJS*, *Django*, *Rails*, *Express*, *Node.js*, *Laravel*, por exemplo) e de desenvolvimento mobile (*Flutter*, *Xamarin*, *React Native* e *Ionic*, por exemplo).

3.4.5.1 *Laravel*

Laravel é uma *framework* de desenvolvimento *web* com sintaxe elegante e expressiva. É uma *framework PHP* gratuita de código aberto que fornece um grande conjunto de recursos que ajudam na construção de aplicações *PHP*. Este fornece um ponto de partida para a criação de uma aplicação *web* ou *website*, permitindo que o programador desenvolva um bom trabalho sem se preocupar muito com os detalhes.

Esta *framework* é uma ferramenta poderosa, sendo aconselhável tanto a pessoas experientes em *PHP* e desenvolvimento *web* como em pessoas menos experientes nestas áreas. O *Laravel* apresenta algumas vantagens interessantes que fazem com que seja a preferida por parte de alguns desenvolvedores *web*, tais como:

1. Ser progressivo: esta *framework* apresenta várias documentações online, tanto em PDF como em vídeo e, assim, consegue ajudar um pequeno aprendiz de desenvolvimento *web*. Quanto ao apoio a desenvolvedores profissionais, o *Laravel* oferece ferramentas robustas, sendo ajustado para desenvolver aplicações *web* e *websites* profissionais e que proporcionam o trabalho em equipa.
2. Ser escalonável: devido à natureza de escalonamento do *PHP* e o suporte presente no *Laravel* para sistemas com cache rápida como, por exemplo o *Redis*, o escalonamento é muito rápido, sendo as aplicações *web* e *websites* desenvolvidos em *Laravel* dimensionados para lidar com muitas solicitações.
3. Pertencer à comunidade: muitos desenvolvedores e programadores *web* contribuem para a criação de um sistema mais robusto e amigável.

Além destas vantagens, o *Laravel* também apresenta uma vasta gama de características como, por exemplo:

1. Modularidade: o *Laravel* fornece cerca de 20 bibliotecas e módulos integrados que ajudam a melhorar a aplicação.
2. Testabilidade: inclui recursos que ajudam no teste de *websites* e aplicações *web*. Esses recursos ajudam a manter o código de acordo com os requisitos.
3. Encaminhamento: fornece uma abordagem flexível para o utilizador definir rotas na aplicação *web* ou *website*. Ao criar rotas, ajuda a aplicação a aumentar o seu desempenho.
4. Gestão de configurações: uma aplicação *web* ou *website* criado em *Laravel* funciona em diferentes ambientes, o que significa que haverá mudanças constantes na sua configuração. Esta *framework* fornece uma abordagem consistente para lidar com essas configurações.
5. Construtor de consultas e *OPM*: incorpora um construtor de consultas que ajuda a verificar informações de base de dados. Este fornece *OPM* (*Object Relational Mapper*) e a implementação do *ActiveRecord* ao qual é dado o nome de *Eloquent*.
6. *Schema Builder*: o *Schema Builder* guarda as definições da base de dados e o seu esquema no código *PHP*. Também mantém um registo de mudanças nas migrações da base de dados.
7. Mecanismo de *Template*: utiliza o motor *Blade Template*, uma linguagem de *templates* usada para projetar *layouts* com blocos predefinidos que incluem conteúdo dinâmico.
8. Email: inclui uma classe de email que ajuda no envio de emails com conteúdo e anexos das aplicações *web*.
9. Autenticação: facilita os métodos de autenticação e *login* das aplicações *web* pois inclui recursos importantes como recursos de registo, esquecimento da *password* e envio de lembretes da mesma.
10. *Redis*: utiliza o *Redis* para conseguir ligar-se a uma sessão existente e à *cache* de uso geral. Este reage diretamente com a sessão.
11. *Queues*: inclui serviços de fila, tais como o envio de um grande número de emails aos utilizadores. Estas filas ajudam a concluir estas tarefas, de forma que não seja necessário esperar pela finalização da tarefa anterior.

Com o *Laravel* é possível criar *seeds*, ou seja, preencher as bases de dados com valores de teste através de código. Estes valores ajudam, por exemplo, a verificar como a aplicação *web* se comporta com tabelas cheias de dados e gráficos com muitos valores. Através do comando *artisan*, os desenvolvedores podem iniciar os seus projetos e, controladores e outros componentes da aplicação *web* e, assim, acelerar o desenvolvimento da mesma.

3.4.5.2 *Vue JS*

Vue.JS é uma *framework* progressiva de *JavaScript*. É utilizada para construir interfaces de utilizador e aplicações de página única. Ao contrário de outras *frameworks* monolíticas, o *Vue.JS* foi projetado para ser utilizado de forma incremental. Juntamente com HTML e CSS é possível começar a desenvolver aplicações *web* e *websites* em *Vue.JS*.

A sua biblioteca central concentra-se na camada de visualização, sendo fácil selecionar e integrar com outras bibliotecas ou projetos existentes, mas, o *Vue.JS*, quando utilizado juntamente com ferramentas modernas e bibliotecas de suporte, tem a possibilidade de desenvolver aplicações bastante sofisticadas.

O *Vue.js* apresenta muita documentação que facilita na sua aprendizagem. A sua curva de aprendizagem é menor do que noutras *frameworks*, tais como *ReactJS* e *Angular*. Esta *framework* apresenta um recurso muitas vezes subestimado que é o desenvolvimento móvel cruzado. Juntamente com *Alibaba*, *Native Script* e *Ionic*, o *Vue* consegue criar aplicações móveis.

O *Vue.JS* apresenta uma comunidade de desenvolvedores *web* muito ativa, o que faz com que esta *framework* evolua e cresça com o passar do tempo. Esta comunidade tem várias ferramentas e bibliotecas importantes que um desenvolvedor ou programador necessita. Por exemplo, *Vue Router* é utilizado para qualquer tipo de *routing*.

Em suma a *framework Vue.JS* é uma estrutura fácil de crescimento rápido e adaptável para implementar no desenvolvimento de aplicações *web* e *websites*. Apresenta uma baixa curva de aprendizagem, o que facilita a programação e o desenvolvimento *web* por parte de pessoas menos experientes e é apoiado por grandes empresas como *Alibaba*, *Xiaomi* e *Laracast*.

3.4.6 *Datacenter e Hardware*

O *Datacenter* e o *Hardware* é uma parte importante em qualquer projeto, uma vez que ter boas estruturas físicas e bons componentes instalados pode ser fundamental para ajudar a garantir a boa funcionalidade do sistema.

Nesse sentido, aquando do início do estágio todos os cenários existentes encontravam-se alojados nos servidores de *cloud* da *Google Cloud*. Contudo, o objetivo desde o

início foi muito claro: o projeto deveria ficar alojado nas instalações da sede do grupo de forma a que todas as configurações fossem feitas pela equipa de IT da Valsabor, além de que, era necessário diminuir o tempo de comunicação entre o servidor *Web* e o servidor de integração para os ERP's do grupo.

3.4.6.1 *Migração para Cenário Dedicado*

De forma a ser possível migrar o projeto para um cenário dedicado, tal como pretendido, seria necessário um conjunto de alterações em diversos níveis na sede do grupo, as instalações da Valsabor, S.A.

Aliadas às obras de remodelação que se encontram em curso nas instalações da Valsabor, no valor de vários milhões de euros, foi projetado a criação de um novo e mais moderno *datacenter* com o objetivo de suportar os vários serviços do grupo, que ao longo dos últimos anos cresceu rapidamente. Com essa visão em mente, o servidor *Web* seria um dos servidores que iriam ficar alojados.

Em paralelo à criação do novo *datacenter* era necessário garantir uma linha de backup à rede, uma vez que apenas existia uma linha de fibra vinda do exterior. Solicitou-se à operadora que já possuía a cobertura de rede nas instalações, a Altice Empresas, a passagem de mais uma linha de fibra, contudo vinda de uma direção oposta à linha existente. Esta decisão deveu-se a não existir outras operadoras com uma cobertura tão eficiente do local como tem a Altice Empresas. A juntar à linha de backup de fibra, foi adicionada um backup 4G no router da operadora, para caso nenhuma das soluções de fibra esteja ativa, a rede fique suportada através de um backup de rede móvel, que garantirá que a empresa não irá parar.

Assim, com a criação de um *datacenter* moderno, a passagem de uma nova linha de fibra e um backup de rede móvel, começavam a ser reunidas as condições físicas de forma a suportar os cenários de pré-produção e produção que estavam alocados ao projeto.

3.4.6.2 *Cenário Pré-Produção*

Num universo de empresas como o Valgrupo possuiu, onde o horário de funcionamento das diversas unidades é bastante alargado, é fundamental a existência de um cenário de pré-produção, uma vez que os *deploys* devem ser cuidadosos e feitos apenas quando necessário e o menor número de vezes possível para não empactar com os serviços.

Tendo em conta estas contrapartidas, o projeto onde me encontrei inserido durante o estágio na Valsabor, apresentava sete cenários de pré-produção, dedicados a testes e *feedback* por parte dos utilizadores para que, só depois, de ser testado por um conjunto de operadores, fossem colocados os novos desenvolvimentos no cenário de produção.

Neste cenário, os testes são realizados em 3 fases distintas. Numa primeira fase, o programador prepara todo o projeto numa das instâncias do cenário de pré-produção. Após o projeto estar levantado, o mesmo faz alguns testes básicos de forma a garantir que a maior parte do código desenvolvido não apresenta problemas e tem o comportamento esperado. Na segunda fase de testes, o elemento do departamento de *IT* alocado aos testes realiza alguns testes a nível do funcionamento correto do projeto. Após ter passado os dois níveis de testes sem qualquer problema identificado, a solução é passada ao departamento responsável pelo desenvolvimento, que é responsável por fazer o teste final ao projeto e só após a confirmação em que tudo está conforme pretendido é feito o *deploy* do código desenvolvido. Quando existe algum pedido de alterações durante os 3 níveis de teste, todo o processo de teste começa desde o início.

3.4.6.3 *Cenário Produção*

Após todos os testes em todos os desenvolvimentos estarem preparados é agendado a passagem para o cenário de produção. Tendo em conta a imensidade de projetos que estão alocados ao portal Gestão Valgrupo, qualquer paragem deste irá afetar unidades de produção de carne, unidades de produção de ração e unidades de produção de animais, sendo que essa razão necessário agendar a colocação online de novos desenvolvimentos. Nesse sentido, estes *deploys* são sempre efetuados fora do horário laboral, de forma a tentar afetar o menor volume de áreas.

Com todas as dificuldades acima descritas, é fundamental que neste cenário os desenvolvimentos estejam todos o mais estáveis possíveis, pois erros crasos podem causar paragens em várias unidades.

3.4.7 *Projetos*

A nível de projetos onde estive inserido durante todo o processo de estágio no Valgrupo, destaco três projetos daqueles em que estive inserido: todo o processo de troca dos cenários de produção e pré-produção dos servidores da *Google Cloud* para

as instalações da Valsabor, o projeto de gestão de contratos de matérias primas e o projeto de entrada e saída de viaturas das unidades produtivas.

3.4.7.1 Migração Cenários

A migração de cenários de produção e pré-produção entre os servidores da *Google Cloud* e os servidores presentes nas instalações da Valsabor foi uma tarefa que obrigou a fazer um conjunto de testes de forma a garantir que tudo estava a funcionar corretamente no momento da troca.

Após a construção do novo *datacenter* e a colocação de redundância de fibra por parte do fornecedor de redes estávamos em condições de avançar para a migração dos cenários.

Nesse sentido existiram um conjunto de testes que foram feitos por parte da equipa de IT de forma a garantir que a redundância estava a funcionar de forma a garantir que, caso existisse algum problema, toda a comunicação continuava a funcionar. Após esses testes, foi preparado um servidor dedicado para alojar os dois cenários. Nesse servidores, foram configuradas duas máquinas virtuais com a distribuição do *Linux Ubuntu 22 LTS*. Nessas máquinas, foram configuradas instâncias *docker* onde cada instância ficaria dedicado ao cenário, alojando assim o servidor *web* e a base de dados. Como no cenário de pré-produção, existem várias instâncias do mesmo, foram configurados várias instâncias *docker* de forma a trabalharem isoladas umas das outras. No cenário de produção, foi configurado apenas uma única instância do projeto.

Para garantir que tudo correria da melhor forma possível, em primeiro lugar foi feita a migração do cenário de pré-produção para testar todas as potenciais falhas aquando da migração do cenário de produção. Após a migração, testou-se em diversos sites do grupo o tempo de resposta em vários separadores no servidor da *Google Cloud* em comparação com o servidor interno.

Tendo em conta que tudo correu bem na mudança do cenário de pré-produção, estávamos em condições de avançar para a troca do cenário de produção com uma segurança de que tudo iria decorrer conforme previsto sem qualquer problema, algo que efetivamente aconteceu.

3.4.7.2 Gestão Contratos Matérias Primas

O projeto de Gestão de Contratos de Matérias Primas por parte da área de negócio da Valnutri foi um projeto bastante complexo, tendo em conta as várias características do negócio e o objetivo final do projeto: gerir os saldos de contratos ao dia e integrar diretamente no ERP os documentos contabilísticos da compra.

Nesse sentido, foram feitas várias reuniões com o departamento responsável por estas compras e por pessoas da área da produção de forma a conseguirmos desenhar a melhor solução face aos requisitos do negócio e todas as suas vertentes.

Com base em todas as reuniões que foram efetuadas e após entender correr-

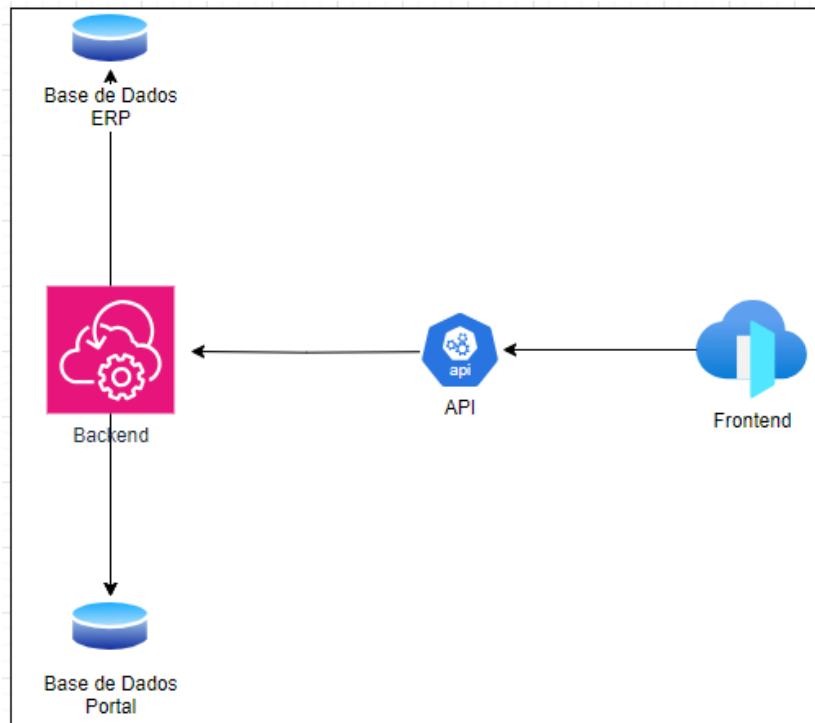


Figura 8: Modelo Lógico Gestão Contratos Matérias Primas

tamente as várias necessidades, chegou-se ao seguinte modelo lógico, figura 8, onde existe um *frontend* desenvolvido em VueJS que faz pedidos via a API ao *backend* do projeto, em *Laravel*. Consoante um conjunto de regras, o *backend* direciona os pedidos para a parte do portal, onde os dados são armazenados na base de dados *MySQL*, ou, ao tratar-se de integração de dados no ERP, para a base de dados deste.

3.4.7.3 Entrada e saída de viaturas

O projeto da gestão de entradas e saídas de viaturas nas principais unidades de produção do grupo, nomeadamente fábricas de rações e matadouros foi requisitado por parte do departamento de produção destas unidades, uma vez que atualmente cada unidade tinha o seu *software* e não existia uniformidade entre unidades.

Este projeto apresentou um grau de complexidade uma vez que tínhamos como objetivo desenvolver uma aplicação que pudesse ser uniforme para todas as diferentes áreas de produção, num sentido de impor a uniformidade entre unidades. Com este objetivo, a forma como foi estruturada a aplicação tornou-se mais complexa.

Além da complexidade acima descrita, o projeto de gestão de entradas e saídas de viaturas, comunica com as básculas de pesagens de camiões através de pedidos *HTTPS* para um serviço desenvolvido por uma empresa externa que devolve os pesos registados por estas. Além da comunicação com as básculas existe ainda a comunicação com impressoras IP de talões, onde são impressos os talões de pesagens que acompanham as cargas.

Com as reuniões que foram efetuadas junto dos responsáveis pela elaboração das pesagens nos diferentes *softwares* existentes, de forma a entender quais as particularidades de cada área, bem como reuniões junto dos departamentos de produção de forma a entender as necessidades de cada um. Assim chegou-se ao seguinte modelo lógico, figura 9.

Neste modelo lógico, podemos verificar que o *frontend* comunica com o *backend* do projeto através da API. O *backend* é responsável por devolver os pesos registados pelas balanças, neste caso através de um serviço desenvolvido através de *JAVA* por um fornecedor do grupo. Após a obtenção dos pesos, os registos são armazenados na base de dados *MySQL* do portal.

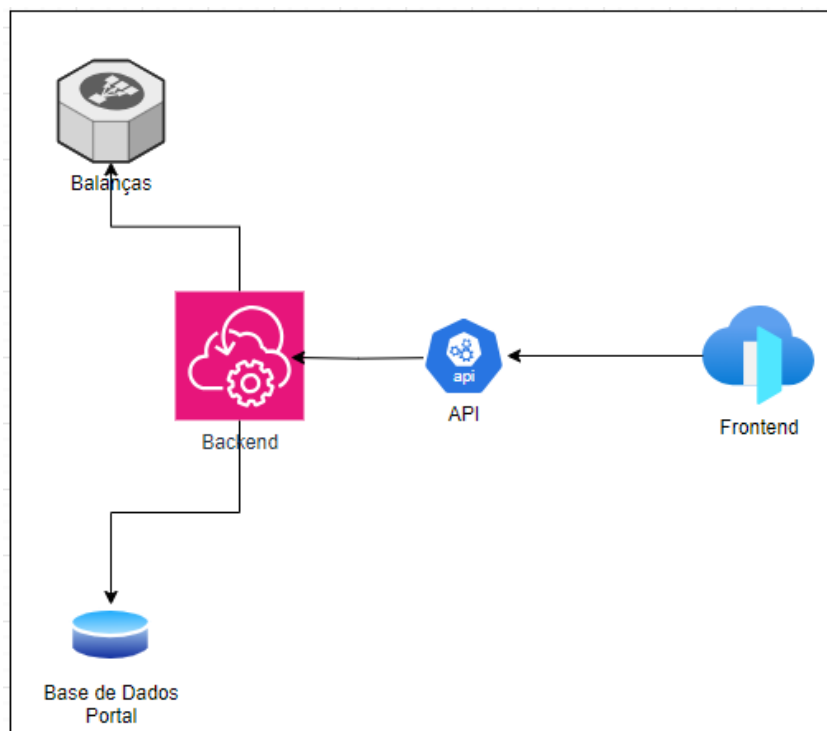


Figura 9: Modelo Lógico Entrada e Saída de Viaturas

IMPLEMENTAÇÃO E TESTES REALIZADOS

Ao longo do presente capítulo serão apresentadas as metodologias, formulários e testes realizados para aplicar as tarefas apresentadas no capítulo transato.

4.1 METODOLOGIAS E FERRAMENTAS UTILIZADAS NOS TESTES

Ao longo desta secção serão apresentadas as várias dificuldades encontradas durante os desenvolvimentos bem como os testes e tarefas necessárias para conseguir finalizar os projetos.

4.1.1 *Hardware*

A nível de *hardware*, e tal como apresentado nos capítulos anteriores, existiu uma mudança a nível do alojamento do portal Gestão Valgrupo durante o período de estágio. Para tal, foi necessário fazer um conjunto de alterações a níveis físicos nas instalações da Valsabor para ser possível suportar estas mudanças sem influenciar o bom funcionamento do sistema nas várias unidades.

No projeto de remodelação das instalações da Valsabor, estava contemplado a criação de um novo e moderno *datacenter* de forma a ficar com os principais serviços do grupo, uma vez que o que existia até então não tinha os padrões de segurança necessários.

4.1.1.1 *Caraterísticas do Datacenter*

Nas instalações da Valsabor foi então criado este novo *datacenter* com características que foram pensadas para que seja o principal local de dados no grupo. Assim, tem que ser uma estrutura segura uma vez que tem um elevado nível de importância.

Nesse sentido este *datacenter* foi elaborado tendo em mente as seguintes características:

- Ambiente isolado de interferências externas;

- Corrente elétrica estabilizada com *UPS Online* e gerador *backup* dedicado
- Climatização controlada com proteção contra formação de humidade e partículas
- Controlo de acessos
- Facilidade de controlo de *hosts* físicos *on-site* com um *KVM* por bastidor com ligação a estes
- Cablagem adequada através de fibra e cabos *ethernet shielded* de alta categoria
- Proteção contra inundação com bastidores assentes em pisos flutuantes com escoamento de água adequado

A nível de servidores, existe uma rede dedicada de 10Gbps a interligar todos os servidores. Esta rede é utilizada para replicação entre servidores e respetivos sistemas de backups. Estes, encontram-se equipados com pelo menos 8TB de *SSD* em *raid* 10, sistema operativo *Windows Server 2022* e a plataforma de virtualização *hyper-v*, que foi escolhida pela sua facilidade de uso. Nestes equipamentos, os *hosts* físicos apenas tem instalado o software de virtualização e nas máquinas virtuais encontra-se o sistema operativo mais apropriado consoante as necessidades da aplicação.

A nível de sistema de *backups* estes encontram-se assentes na plataforma *Veeam Software*, que segue a filosofia par-a-par, uma vez que existem servidores de backups instalados em sites geograficamente distantes um do outro.

A nível de redundância foi implementada redundância par-a-par, que foi idealizada tendo em mente que, um servidor físico a falhar, iria colocar uma série de serviços *offline* por tempo indeterminado. Assim, o sistema implementado consiste em dois *hosts* físicos que partilham entre si todas as máquinas virtuais que possuem.

Na figura 10 é possível verificar que existem dois servidores semelhantes. A verde, encontram-se as máquinas virtuais *online* e a cinzento as que estão *offline*. Assim, num momento de falha, as máquinas virtuais que se encontravam *offline*, assumem o lugar daquelas que estavam *online*, no servidor redundante. Esta replicação é semi-instântanea, sendo praticamente nula a chance de perda de dados.

A nível de rede existiu uma reformulação necessária, uma vez que o grupo não estava uniforme, devido à informática ser recente e à quantidade de empresas que haviam sido adquiridas ao longo dos últimos anos. Nesse sentido, pode-se dividir a rede em duas áreas específicas: rede *WAN* e rede *LAN*.

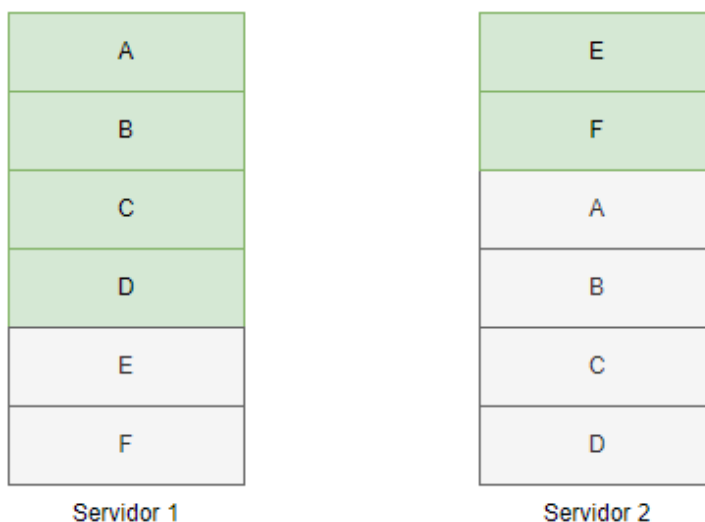


Figura 10: Replicação entre servidores

4.1.1.2 Rede WAN

A nível da rede *WAN* em cada um dos sites do grupo existem duas ligações, uma através de fibra e outra através de 4G. Dependendo do site e da sua importância no contexto do grupo existem velocidades diferentes acordadas com o *ISP*. Nas principais unidades do grupo, onde se encontram localizados um maior número de colaboradores, existe também uma linha de fibra de redundância. Entre todos os sites, existe uma *VPN* centralizada que é fornecida pelo *ISP*. Nesta rede, existe ainda um sistema de *VOIP* baseado em *Global Connect* da Altime.

4.1.1.3 Rede LAN

Com todas as mudanças previstas e de forma a garantir uma maior segurança ao grupo, toda a rede *LAN* foi reestruturada de forma a que, cada um dos sites, tenha dois *softwares* de *firewall*: *OPNSense* para *routing* e *vpns client-to-site*; *Arista* para controlo de perímetro, de forma a existir proteção contra *malware* e vírus bem como controlo de acesso à internet e filtro categórico baseado em utilizadores com integração *LDAP/Active Directory*.

Nas instalações da Valsabor, existe uma linha de backup dentro do recinto, alinhada em anel, onde existem *switchs* equipados com *SFP* de 10Gbps e que se encontram configurados com o protocolo *MSTP*, *loop-detection-prevention* e respetivas *Vlans*, onde estas separam a rede corporativa, produção, gestão de rede, *VOIP* e rede *guest*. Na figura 11, é possível verificar o esquema de rede implementado no grupo.

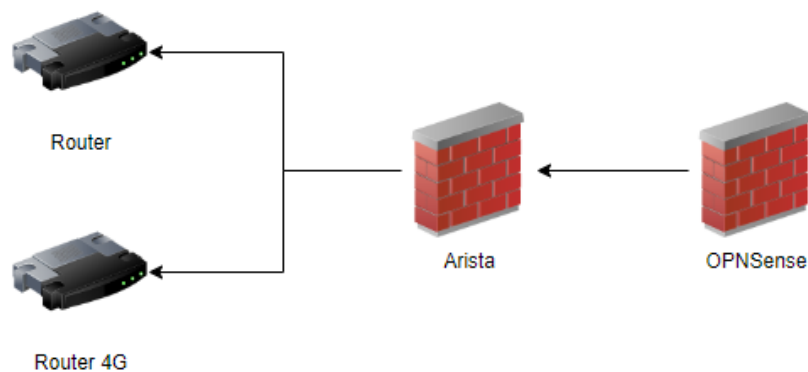


Figura 11: Esquema de rede

4.1.2 Desenvolvimento de aplicações Web

No capítulo anterior foi apresentado os vários cenários de produção existentes no Valgrupo: o cenário de pré-produção e o cenário de produção. Ambos estes cenários tem uma grande importância no conjunto de empresas que compõem o grupo, uma vez que existe uma grande quantidade de utilizadores a utilizar diariamente, como ferramenta de trabalho, o portal Gestão Valgrupo. Nesse sentido, é fundamental a utilização de um cenário de pré-produção, de forma a garantir que os novos desenvolvimentos, ao serem disponibilizados no cenário de produção não irão causar incompatibilidades ou erros.

De forma a combater possíveis erros que possam impactar com o trabalho diário dos operadores, no contexto do Valgrupo, é dada uma grande importância ao processo de testes. Nesse sentido, os testes serão realizados em 3 fases distintas e por operadores diferenciados. Assim, existem várias instâncias de cada cenário de pré-produção onde, cada programador tem a responsabilidade de montar todo o seu código de forma a simular um *deploy* para o cenário de produção e, caso exista algum erro na preparação da instância, este será resolvido sem afetar o cenário principal. De seguida, é da responsabilidade do programador, testar as diferentes funcionalidades de cada projeto, desde as operações *CRUD* à aplicação de filtros de pesquisa e navegação entre diferentes páginas. Assim, ainda antes de ser visto por alguém externo à equipa de desenvolvimento, o risco de existirem erros acaba por ser mais reduzido, uma vez que as principais funcionalidades se encontram testadas.

Após os testes por parte do programador estarem concluídos, o processo de teste avança para um novo nível. Desta feita, os testes são realizados por parte de um elemento do departamento de IT que está alocado aos testes do projeto e na implementação do projeto no terreno juntamente com o programador. Nesta fase, o *tester*, realiza novamente testes de performance, porém desta feita tentando provocar o

erro do sistema. É este utilizador o responsável pelos primeiros testes de usabilidade de forma a sugerir algumas melhorias que possam facilitar a uma melhor utilização por parte do utilizador final.

Após os testes por parte do departamento de desenvolvimento e do departamento de IT estarem concluídos com sucesso, o projeto será disponibilizado ao departamento responsável pelo pedido do desenvolvimento. Aqui, serão simulados vários casos de uso reais por parte utilizadores, que, irão ficar a utilizar o projeto após concluído. Nesta fase também são sugeridas melhorias ou ajustes a nível da usabilidade do projeto. Estas melhorias de usabilidade serão analisadas por parte dos departamento de desenvolvimento e, caso não se tratem de alterações fundamentais ao bom funcionamento do projeto, o mesmo irá avançar para o cenário de produção, ficando essas alterações para uma segunda versão do projeto.

Durante os vários processos de testes, sempre que é identificado um erro em qualquer dos níveis, é recomeçado de início toda a testagem, ou seja, não existe qualquer e mínima alteração ao sistema que seja colocado no cenário de produção sem passar por todo este conjunto de testes. O fluxograma seguinte, figura 12, apresenta todo o processo de testes de desenvolvimento no Valgrupo.

Ao longo do estágio o processo de testes foi evoluindo e sendo ajustado de forma a corrigir algumas situações que impactavam com este processo. Inicialmente o processo de testes não envolvia os programadores, sendo os responsáveis pelos mesmos o *tester* e a pessoa do departamento que requereu o desenvolvimento. Contudo, quando o *tester* tentava montar o projeto existiam vários erros bem como funcionalidades simples apresentavam problemas de usabilidade. Nesse sentido, foi decidido, que o programador seria parte integrante de todo este processo de forma a que, caso existam erros iniciais, sejam rapidamente corrigidos por este e só após um filtro inicial será atribuído a responsabilidade de testar ao *tester*. Com a alteração deste processo conseguiu-se acelerar o processo de testes com vista à colocação no cenário de produção de novas funcionalidades.

Existe um conjunto de tarefas que cada um dos elementos responsáveis pelos vários testes deve apontar todas as notas acerca dos testes realizados. A cada reinício dos testes a versão de teste deve ser incrementada, conforme representa a figura 13.

Todos estes processos acabam por ser dificultados pela própria estrutura da empresa, que obriga a um grande rigor no processo de testes, uma vez que a amplitude horária de funcionamento das diversas empresas do grupo é bastante alargada e existe

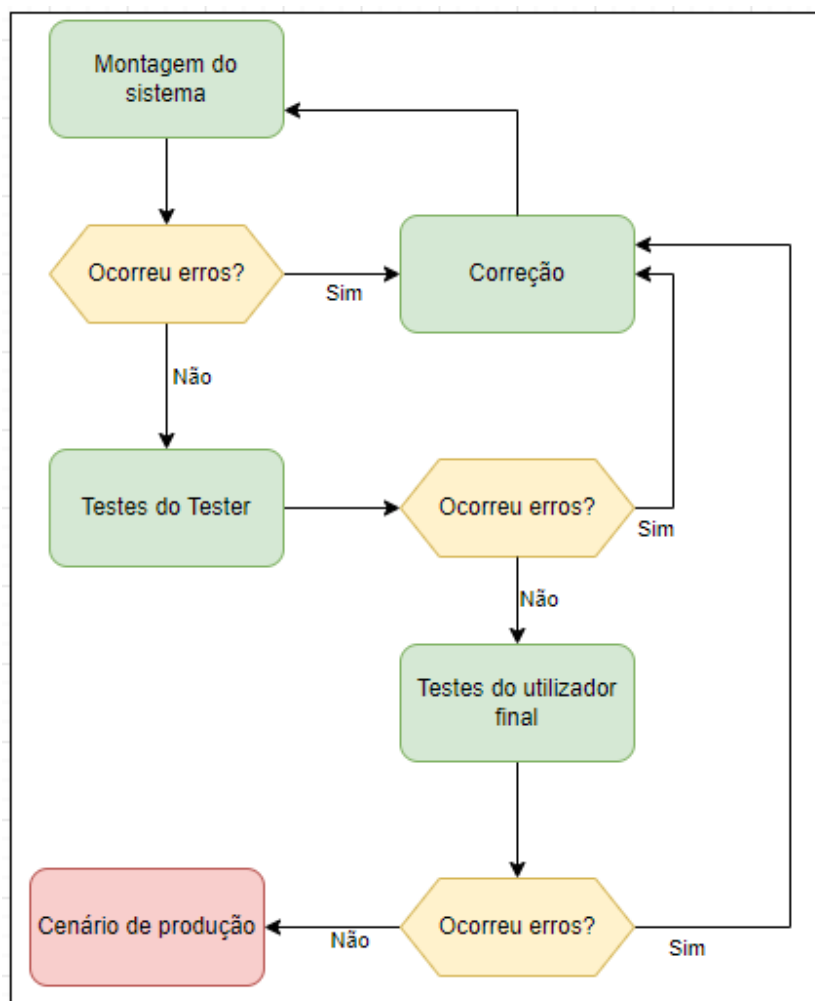


Figura 12: Fluxograma Testes de Desenvolvimento

um período temporal curto onde é possível colocar *online* novos desenvolvimentos. Caso alguma coisa corra mal durante esse processo, todas as alterações tem que ser revertidas e agendado um novo *deploy*, uma vez que não existe tempo útil para corrigir o problema no momento.

4.1.2.1 Gestão Contratos Matérias Primas

O projeto Gestão de Contratos de Matérias Primas foi um projeto amplamente complexo tendo em conta todas as complexidades da empresa. Neste contexto, o projeto teria que ser desenvolvido de forma a garantir uma correta gestão dos dados, porém com bastantes complexidades.

Uma das complexidades deste projeto é poder existir compras a fornecedores externos por uma empresa de forma a consumir essa matéria prima, pode ser uma empresa a comprar ao fornecedor externo e depois vender internamente a outras empresas do grupo, uma vez que os contratos de matérias primas são elaborados em nome de certas empresas do grupo e depois essas vão fazendo a respetiva gestão

Nível de teste	Tarefa	Versão:	
		Impeditivo de avançar	Descrição
Nível 1	Montagem do sistema		
	Testes a operações CRUD		
	Testes de filtros		
	Testes de exportação para Excel		
	Testes de navegação entre páginas		
Nível 2	Testes a operações CRUD		
	Testes de filtros		
	Testes de exportação para Excel		
	Testes de navegação entre páginas		
	Testes de usabilidade e recomendações		
Nível 3	Testes a operações CRUD		
	Testes de filtros		
	Testes de exportação para Excel		
	Testes de navegação entre páginas		
	Testes de usabilidade e recomendações		

Figura 13: Níveis de testes

entre elas. Todos estes pontos aumentaram a complexidade da aplicação que se pretendia elaborar. Um outro ponto que demonstra a importância de todo o sistema funcionar corretamente é que, desde o momento em que foi ativa esta solução, caso a encomenda a fornecedor não fosse integrada corretamente no ERP, a fábrica não conseguia dar entrada da matéria prima, logo não conseguia produzir o seu produto final.

Durante a realização deste projeto e mesmo após a conclusão, foram feitos questionários aos utilizadores que iriam ser os responsáveis por trabalhar com a plataforma. Os formulários realizados não têm cariz quantificável uma vez que são elaborados de forma a obter recomendações de melhoria e satisfação das funcionalidades. Nesse sentido, seguem-se algumas das propostas de melhorias indicadas pelos utilizadores:

1. Da parte do departamento de compra das matérias primas foi solicitado a inserção de um campo para a indicação do lote do produto por parte do fornecedor, conforme presente na figura 14. Assim, a cada carga de matéria prima, é confirmado qual o lote do fornecedor, ficando assim com todo o processo de rastreabilidade do lado da aplicação. Quando este lote não existe criado no ERP, o sistema de integração cria-o de forma automática.

Figura 14: Adição do campo do lote

2. Pelo departamento de produção das fábricas surgiu a questão de que seria útil uma janela onde fossem apresentadas as diversas cargas programadas para um determinado dia para uma fábrica. Nessa janela, deveria ser possível alterar os dados referentes à matrícula do veículo, uma vez que quando se trata de fornecedores externos, a viatura que vem nem sempre coincide com a viatura planeada, sendo que a figura 15 mostra a janela desenvolvida. Assim, ao trocar a matrícula deveria atualizar a encomenda de fornecedor no ERP para os dados corretos.

Data de Entrega	Nº Pedido	Artigo	Fornecedor	Entidade Requirante	Local de Carga	Quantidade	Quantidade Aprovada	Transportadora	Meio de Expedição	Motorista	Volta	Ações
30/04/2024	3706	1 PX103 - Oxido Zinco 72%	QUIMALMEL - QUIMICOS E MINERAIS, LDA	Valmex Vendas Novas	Albergaria-a-Velha	8 000.000	8 000.000	A definir	---	---	---	
27/03/2024	3751	1 PX323 - Luctarom 3416Z Advance	Lucta, SA	Valmex Vendas Novas	Espanha	6 000.000	6 000.000	A definir	---	---	---	
27/03/2024	3751	1 PX325 - Luctarom 4972Z (Eucalipto)	Lucta, SA	Valmex Vendas Novas	Espanha	100.000	100.000	A definir	---	---	---	

Figura 15: Janela de inserção de dados pelas fábricas

3. O departamento de compra das matérias primas solicitou, também, a inserção de um quadro resumo no momento da integração da fatura, tanto no caso da compra como na venda, onde seria possível confirmar os dados que realmente foram integrados, nomeadamente valor total do documento, valor do IVA e o número do documento na contabilidade, uma vez que é necessário para identificar a fatura junto do departamento de contabilidade. Assim, com esse desenvolvimento, deixaram de ter a necessidade de consultar o ERP para retirarem informação acerca dos documentos inseridos.

4.1.2.2 *Entrada e saída de viaturas*

O projeto Entrada e saída de viaturas foi um projeto que, de um ponto de vista inicial parecia ser um projeto com pouca complexidade, porém ao tentar juntar todas as várias opções existentes nas várias unidades de produção acabou por ficar bastante complexo.

Uma das principais complexidades é, que pelo facto de o grupo possuir uma empresa de transportes de mercadorias, os motoristas não terem um carro atribuído a si, ou seja, o mesmo motorista pode estar envolvido em diferentes processos de entrada e saída de viaturas. Uma das regras da empresa é que qualquer veículo de transporte de mercadorias tem que ser pesado à entrada e à saída das instalações bem como no processo de carga: antes e depois de carregar. Assim, uma viatura chega às instalações com um motorista, o processo de tara antes da carga pode ser efetuado por outro, a obtenção do peso bruto pode ser um motorista diferente a realizar e a saída das instalações pode ser um outro a fazê-lo. A juntar a isto, estes processos

podem decorrer em dias diferentes, contudo tem que ficar tudo associado ao mesmo registo.

Existiu um conjunto de questões que foram feitas aos utilizadores que utilizavam esta funcionalidade de forma a verificar potenciais melhorias ao projeto de forma a que ficasse mais amigável do utilizador e até mesmo ajustado a algumas realidades. Os formulários realizados não têm caráter quantificável uma vez que são elaborados de forma a obter recomendações de melhoria e satisfação das funcionalidades. Assim conseguiu-se um conjunto de opiniões, que foram analisadas pelo departamento e decidiu-se colocar para desenvolvimento as seguintes alterações propostas:

1. Da parte dos utilizadores que fazem o registo das entradas e saídas das unidades de produção bem como a obtenção dos pesos das básculas, foi sugerido a adição de câmaras de reconhecimento de matrículas de forma a inserir o registo da entrada ou saída, ficando da parte do utilizador apenas o término do preenchimento de alguns dos dados. Nesse sentido, avançou-se com o desenvolvimento dessas propostas, contudo não foi possível adicionar todas as unidades de produção a trabalhar desta forma, uma vez que algumas questões físicas impossibilitavam um bom reconhecimento por parte da câmara. A unidade da Valsabor foi a unidade onde não foi possível adicionar este novo método de registo, porém, aquando do término das obras de remodelação está previsto ser possível ativar este mecanismo.
2. O departamento da expedição de carne, da Valsabor, pediu a alteração a uma funcionalidade onde, inicialmente era a portaria a indicar o destino, cliente e produto de cada veículo de transporte. Contudo, este procedimento originou muitos problemas na análise dos dados, uma vez que nem sempre eram preenchidos de forma mais correta. Dessa forma, foi pedido para que esses campos fossem deslocados para uma janela onde o departamento da expedição já iria preencher alguns campos específicos do transporte. Assim, garantia-se o correto preenchimento dos dados para posterior análises.
3. Do departamento de expedição de carne veio também a sugestão de obrigar a dar uma justificação sempre que haja uma diferença significativa entre o peso da guia de transporte e o peso que o veículo obtém através da balança. Nesse caso, terá que existir uma justificação para o sistema sinalizar que o veículo pode sair das instalações. Caso contrário, terá que ser totalmente descarregado e reiniciado todo o processo de carga novamente.

4.2 TESTES DE SEGURANÇA

Durante todo o processo de estágio, a segurança foi sempre um fator a ter em mente tanto no departamento de informática com junto da administração do grupo. Nesse sentido, foi dedicada uma parte do tempo de desenvolvimento aos testes de segurança de forma a garantir que o se estava a produzir cumpria os requisitos necessários.

Nesse sentido existiram alguns conjuntos de testes de *frontend* que era realizados, através da distribuição *Linux Kali*, sempre que se preparava a colocação no cenário de produção de um desenvolvimento:

1. Análise à configuração dos *virtual hosts*
2. Análise à Estrutura Física vs Estrutura Indexável
3. Testes de código
4. Análise ao *software* desenvolvido

Relativamente aos testes à análise de configuração dos *virtual hosts*, recorri à ferramenta *NIKTO* [4]. Esta é uma ferramenta *open source* amplamente utilizada para testar a segurança de servidores *web*. Esta ferramenta realiza uma série de verificações automatizadas em busca de vulnerabilidades conhecidas e configurações inadequadas que possam representar um risco de segurança. Assim, o *NIKTO* permite identificar o servidor de forma a verificar se o mesmo está desatualizado e vulnerável a *exploits* conhecidos, verifica falhas de vulnerabilidades comparando a versão do *software* instalado com uma lista de vulnerabilidades conhecidas para determinar se o servidor está em risco. O *NIKTO* pode verificar se o servidor está configurado de forma segura, analisando as diretorias sensíveis e se existem ficheiros de configuração expostos. Além disto o *NIKTO* verifica a configuração do *SSL/TLS* do servidor, incluindo a força dos algoritmos de criptografia e a presença de certificados auto-assinados ou expirados bem como outras configurações relacionados com o transporte de dados. Assim, esta ferramenta é poderosa para avaliação de servidores *web*, ajudando os profissionais de segurança e administradores de sistemas a identificar e corrigir vulnerabilidades antes que sejam exploradas por atacantes.

No contexto do estágio, foram elaborados testes com a ferramenta *NIKTO* tanto através da rede interna, figura 16, como da rede externa, figura 17, de forma a analisar o nível de segurança para ataques exteriores ou para ataques interiores. As figuras seguintes apresentam um resultado da ferramenta a ser utilizada fora da rede interna da empresa e fora da rede interna.

Nas figuras 16 e 17 é possível verificar que o teste feito através da rede interna

```

(kali@kali)~$ niktto -c all -host https://gestao.valgrupo.pt -o scan.txt
- Niktto v2.5.0
-----
+ Target IP:          192.168.0.211
+ Target Hostname:   gestao.valgrupo.pt
+ Target Port:       443
-----
+ SSL Info:          Subject: /CN=portal.valgrupo.pt
                   Ciphers: TLS_AES_256_GCM_SHA384
                   Issuer: /C=US/O=Let's Encrypt/CN=R3
+ Start Time:       2024-03-29 08:32:36 (GMT-4)
-----
+ Server: nginx
+ /: Retrieved x-powered-by header: PHP/7.4.33.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the original-content-type-header/
+ /: Cookie XSRF-TOKEN created without the secure flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Cookie XSRF-TOKEN created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Cookie gestao_valgrupo_session created without the secure flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ Root page / redirects to: https://gestao.valgrupo.pt/login
+ Hostname 'gestao.valgrupo.pt' does not match certificate's names: portal.valgrupo.pt. See: https://cwe.mitre.org/data/definitions/297.html
+ /login/: This might be interesting.
+ /register/: This might be interesting.
+ 26585 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:         2024-03-29 11:34:51 (GMT-4) (10935 seconds)
-----
+ 1 host(s) tested

```

Figura 16: Resultado da ferramenta *NIKTO* através da rede interna

```

(kali@kali)~$ niktto -c all -host https://gestao.valgrupo.pt -o scan.txt
- Niktto v2.5.0
-----
+ Target IP:          67.28.160.104
+ Target Hostname:   gestao.valgrupo.pt
+ Target Port:       443
-----
+ SSL Info:          Subject: /CN=portal.valgrupo.pt
                   Ciphers: TLS_AES_256_GCM_SHA384
                   Issuer: /C=US/O=Let's Encrypt/CN=R3
+ Start Time:       2024-03-29 11:48:43 (GMT-4)
-----
+ Server: nginx
+ /: Retrieved x-powered-by header: PHP/7.4.33.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsploit.com/articles/x-content-type-options-header/
+ /: Cookie XSRF-TOKEN created without the secure flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Cookie XSRF-TOKEN created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Cookie gestao_valgrupo_session created without the secure flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ Root page / redirects to: https://gestao.valgrupo.pt/login
+ Hostname 'gestao.valgrupo.pt' does not match certificate's names: portal.valgrupo.pt. See: https://cwe.mitre.org/data/definitions/297.html
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect: Connect failed: ; Connection timed out at /var/lib/niktto/plugins/LW2.pm line 5254.
+ Connection timed out
+ Scan terminated: 20 error(s) and 8 item(s) reported on remote host
+ End Time:         2024-03-29 11:50:37 (GMT-4) (594 seconds)
-----
+ 1 host(s) tested

```

Figura 17: Resultado da ferramenta *NIKTO* através da rede externa

apresentou um maior nível de detalhe, apresentando informações acerca das rotas */login* e */register*, apresentando estas como potencialmente interessantes. Através da rede externa, o mesmo exemplo apresentou um erro e a conexão caiu devido a *time out*.

Relativamente aos testes à análise da estrutura física vs estrutura indexável, utilizei a ferramenta *DIRB* [2], que é uma abreviação para *Directory Buster* e é uma ferramenta *open source* utilizada para descobrir diretorias e ficheiros ocultos num servidor web, explorando a estrutura de ficheiros do alvo. Esta ferramenta opera a enviar solicitações HTTP para o servidor *web* alvo, tentando várias combinações de diretorias e nome de ficheiros comuns com o objetivo de descobrir de encontrar ficheiros que não possam estar vinculados diretamente a partir de páginas visíveis mas que ainda possam ser acedidos e representar vulnerabilidades de segurança.

No contexto do estágio, foram elaborados testes com a ferramenta *DIRB* tanto através da rede interna como da rede externa, figuras 18 e 19 respetivamente, onde podemos analisar que através da rede externa o teste foi bastante menos eficaz não conseguindo devolver os mesmos dados que através da rede interna. Pode-se também analisar, através das figuras seguintes, que não existem diretorias privadas

que estejam a retornar dados sem ser o código *HTTP 302*.

Relativamente aos testes de código recorreu-se aos *softwares SQLMap* [3] e *ZaProxy* [1]. O *SQLMap* é uma ferramenta *open source* que automatiza o processo de deteção e exploração de vulnerabilidades de injeção de SQL em aplicações *web*. Esta ferramenta foi projetada para ajudar os *testers* de segurança a identificar falhas nas aplicações *web* uma vez que a injeção de SQL consiste numa técnica na qual o invasor utiliza código SQL malicioso em campos de formulários ou parâmetros URL e se a aplicação não validar e sanitizar os dados, essa técnica pode ser utilizada para executar comandos SQL não autorizados. O *ZaProxy* (*Zed Attack Proxy*) é uma ferramenta de segurança projetada para encontrar vulnerabilidades em aplicações *web*. Este *software* é utilizado para identificar e explorar uma ampla variedade de vulnerabilidades, como *SQL Injection*, *cross-site scripting (XSS)* entre outras. Na figura 20, é possível verificar um resultado obtido através do *ZaProxy*, onde são apresentados vários alertas após ser corrido contra o projeto.

Relativamente à análise ao *software* desenvolvido foi utilizada a ferramenta *OpenVAS* (*Open Vulnerability Assessment System*) que é uma estrutura de teste de segurança utilizada para análise e gestão de vulnerabilidades em redes e sistemas. O *OpenVAS* é uma ferramenta extremamente útil para ajudar a proteger sistemas contra ameaças de segurança, permitindo que os administradores de sistemas identifiquem e corrijam vulnerabilidades antes que sejam exploradas por atacantes.

A nível de base de dados, no início do estágio existia apenas um utilizador comum a toda a plataforma. Durante a realização do mesmo, obteve-se por segregar o acesso à base de dados em vários utilizadores, um para cada tipo de área desenvolvida. Assim, é possível aplicar políticas de segurança mais granulares, reduzindo o risco de acessos não autorizados a informações sensíveis; o facto de distribuir o acesso aos dados entre vários utilizadores pode melhorar a escalabilidade do sistema, uma vez que permite aguentar com um maior volume de tráfego e processar mais solicitações ao mesmo tempo; é possível otimizar o desempenho do sistema uma vez que se um utilizador estiver mais sobrecarregado os outros não serão afetados. Contudo para que os pontos acima referidos sejam realmente vantajosos é necessário projetar cuidadosamente a arquitetura do sistema de forma a garantir que atenda às necessidades específicas dos projetos.

A nível de testes da API do *software* implementou-se esta a requerer autenticação adequada para aceder aos recursos protegidos de forma a garantir que apenas os utilizadores autorizados tem acesso aos dados. A nível de controlo de acessos existe a verificação das permissões e de *tokens* de sessão bem como chaves de API.

No software foi ainda implementado *tokens* CSRF para proteção contra ataques de *Cross-Site Request Forgery* (CSRF) e *Cross-Site Scripting* (XSS). A segurança da comunicação entre o cliente e a API está garantindo através da utilização do protocolo HTTPS com certificados SSL/TLS. Foi ainda implementado, no servidor o serviço *Fail2Ban* para proteger a API contra ataques de negação de serviço (DOS) e ataques de força bruta, ficando todas as tentativas de acesso em ficheiros de *logs*. É importante notar que testar a segurança de uma API deve ser um processo contínuo e iterativo, integrado ao ciclo de vida do desenvolvimento de software, para garantir que a API permaneça protegida contra as ameaças em constante evolução do ambiente cibernético. Além disso, a colaboração entre os programadores e os profissionais de segurança é fundamental para garantir que todas as vulnerabilidades sejam identificadas e corrigidas de forma eficaz.

4.3 RESULTADOS

Com todo o processo de estágio é importante realçar que todos os projetos foram bastante enriquecedores para o meu crescimento pessoal e profissional. Contudo, destaco como projeto mais importante de todos aqueles em que participei foi o projeto de Gestão de Contratos de Cereal, uma vez que foi um projeto que me fez ter contacto com uma realidade diferente, na medida em que todas as integrações entre os diversos sistemas e toda a complexidade envolvente foi algo muito entusiasmante e desafiador. Além disso, este desenvolvimento, por toda a parte da implementação junto dos operadores, obrigou-me a trabalhar com outras pessoas do grupo, conhecer outras empresas e andar no terreno a conhecer os processos.

Com todos os desenvolvimentos existentes, e tendo em conta que o grupo estava muito órfão de tecnologias, conseguiu-se desenvolver projetos enriquecedores para os diversos operadores na medida em que se conseguiu retirar trabalho humano e passar o mesmo para as máquinas, além de que foram montados sistemas rígidos para evitar falhas em operações manuais, sendo o *feedback* final muito satisfatório.

Contudo, nem todos os pontos do estágio foram pontos totalmente positivos, na medida em que neste caminho foram encontradas diversos obstáculos. Desde à falta de estrutura informática para algumas implementações, relutância dos operadores à mudança e falta de tempo destes para a fase de início de projetos. Porém, conseguiu-se ultrapassar estes obstáculos e o projetos foram bem sucedidos.

```
└─$ dirb https://gestao.valgrupo.pt
┌───┴───┐
│ Trash │
└───┴───┘
DIRB v2.22
By The Dark Raver
┌───┴───┐
│       │
└───┴───┘
START_TIME: Fri Mar 29 08:33:46 2024
URL_BASE: https://gestao.valgrupo.pt/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
┌───┴───┐
│       │
└───┴───┘
GENERATED WORDS: 4612

--- Scanning URL: https://gestao.valgrupo.pt/ ---
+ https://gestao.valgrupo.pt/audits (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/clientes (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/companies (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/contacts (CODE:302|SIZE:374)
=> DIRECTORY: https://gestao.valgrupo.pt/css/
+ https://gestao.valgrupo.pt/departments (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/destinations (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/employees (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/export (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/favicon.ico (CODE:200|SIZE:0)
+ https://gestao.valgrupo.pt/home (CODE:302|SIZE:374)
=> DIRECTORY: https://gestao.valgrupo.pt/img/
+ https://gestao.valgrupo.pt/index.php (CODE:302|SIZE:414)
=> DIRECTORY: https://gestao.valgrupo.pt/js/
+ https://gestao.valgrupo.pt/login (CODE:200|SIZE:12317)
+ https://gestao.valgrupo.pt/logout (CODE:405|SIZE:630)
+ https://gestao.valgrupo.pt/logs (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/products (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/profile (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/register (CODE:200|SIZE:5747)
+ https://gestao.valgrupo.pt/robots.txt (CODE:200|SIZE:24)
=> DIRECTORY: https://gestao.valgrupo.pt/scripts/
=> DIRECTORY: https://gestao.valgrupo.pt/vendor/

--- Entering directory: https://gestao.valgrupo.pt/css/ ---
=> DIRECTORY: https://gestao.valgrupo.pt/css/fonts/

--- Entering directory: https://gestao.valgrupo.pt/img/ ---

--- Entering directory: https://gestao.valgrupo.pt/js/ ---

--- Entering directory: https://gestao.valgrupo.pt/scripts/ ---

--- Entering directory: https://gestao.valgrupo.pt/vendor/ ---

--- Entering directory: https://gestao.valgrupo.pt/css/fonts/ ---
```

Figura 18: Resultado da ferramenta *DIRB* através da rede interna

```
(kali㉿kali)-[~]
└─$ dirb https://gestao.valgrupo.pt

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Fri Mar 29 08:09:50 2024
URL_BASE: https://gestao.valgrupo.pt/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

----- Scanning URL: https://gestao.valgrupo.pt/ -----

+ https://gestao.valgrupo.pt/audits (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/clientes (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/companies (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/contacts (CODE:302|SIZE:374)

=> DIRECTORY: https://gestao.valgrupo.pt/css/
+ https://gestao.valgrupo.pt/departments (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/destinations (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/employees (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/export (CODE:302|SIZE:374)
+ https://gestao.valgrupo.pt/favicon.ico (CODE:200|SIZE:0)
+ https://gestao.valgrupo.pt/home (CODE:302|SIZE:374)

=> DIRECTORY: https://gestao.valgrupo.pt/img/
+ https://gestao.valgrupo.pt/index.php (CODE:302|SIZE:414)

=> DIRECTORY: https://gestao.valgrupo.pt/js/
+ https://gestao.valgrupo.pt/login (CODE:200|SIZE:12317)
+ https://gestao.valgrupo.pt/logout (CODE:405|SIZE:630)
+ https://gestao.valgrupo.pt/logs (CODE:302|SIZE:374)

(!) FATAL: Too many errors connecting to host
(Possible cause: OPERATION TIMEOUT)

-----
END_TIME: Fri Mar 29 08:15:49 2024
DOWNLOADED: 2409 - FOUND: 14
```

Figura 19: Resultado da ferramenta *DIRB* através da rede externa

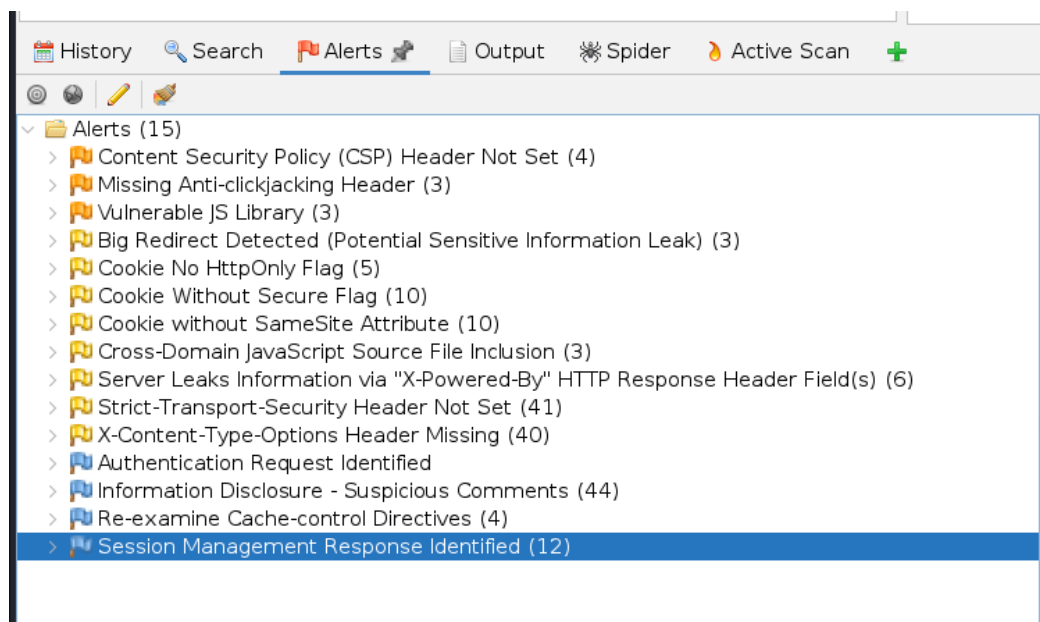


Figura 20: Resultado da ferramenta *ZaProxy*

CONCLUSÃO

Após todo o período de estágio no grupo de empresas Valgrupo, mais concretamente na empresa Valsabor, considero que foi uma experiência de vida e profissional muito enriquecedora onde os objetivos a que me propus inicialmente foram completados. No início tinha como objetivo conseguir fazer a diferença na equipa de trabalho e no projeto em que me iria inserir. Nesse sentido, a minha colaboração foi bastante útil através dos conhecimentos que trouxe para a equipa tanto adquiridos na Licenciatura em Engenharia Informática do Instituto Politécnico de Leiria bem como curso de mestrado em Cibersegurança e Informática Forense. Com os conhecimentos adquiridos neste curso, foi possível fazer testes adicionais de segurança com o objetivo de melhorar a plataforma a nível de segurança informática.

5.1 TRABALHOS FUTUROS

Com o objetivo de uma melhoria continua do projeto e de aumentar a segurança existente, existe um conjunto de tarefas que ainda não foi possível implementar por toda a complexidade do projeto, contudo são desenvolvimentos previstos de implementar num curto período de tempo:

- Implementar um certificado de domínio e um certificado para API de forma a fazer a distinção entre os dois;
- O login na plataforma deve ser efetuado através de um segundo fator de autenticação de forma a adicionar ao processo de login uma camada extra de segurança
- Validação de utilizadores através de certificado digital, de forma a garantir que o utilizador é realmente quem diz ser

BIBLIOGRAFIA

- 1 - The zap homepage, <https://www.zaproxy.org/> (accessed Mar. 29, 2024).
- 2 - Dirb | Kali Linux tools, <https://www.kali.org/tools/dirb/> (accessed Mar. 29, 2024).
- 3 - SQLMAP: Automatic SQL Injection and database takeover tool, <https://sqlmap.org/> (accessed Mar. 29, 2024).
- 4 - Nikto 2.5 | cirt.net, <https://cirt.net/Nikto2> (accessed Mar. 29, 2024).
- 5 - What is an API? - application programming interface explained - AWS, <https://aws.amazon.com/what-is/api/> (accessed Mar. 29, 2024).
- 6 - Linguagens de Programação Interpretadas X compiladas: Qual É ..., <https://www.freecodecamp.org> (accessed Mar. 29, 2024).
- 7 - O que É javascript? - Aprendendo Desenvolvimento web | MDN, <https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript> (accessed Mar. 29, 2024).
- 8 - O que É php? Guia Básico de Programação Php, <https://www.hostinger.pt/tutoriais/o-que-e-php-guia-basico> (accessed Mar. 29, 2024).
- 9 - O que É uma linguagem de programação, <https://rockcontent.com/br/blog/linguagem-de-programacao/> (accessed Mar. 29, 2024).
- 10 - Methodology for software development | blog Santander Open Academy, <https://www.santanderopenacademy.com> (accessed Mar. 29, 2024).
- 11 - Modelos de Desenvolvimento De Projetos: Qual escolher?, <https://pt.linkedin.com/pulse/modelos-de-desenvolvimento-projetos-qual-escolher-anio-ales-carin> (accessed Mar. 29, 2024).
- 12 - Scrum: Entenda a Metodologia Utilizada na Gestão de Projetos, <https://engenharia360.com> (accessed Mar. 29, 2024).

BIBLIOGRAFIA

13 - Laravel - the PHP framework for web artisans, <https://laravel.com/> (accessed Mar. 29, 2024).

14 - Início Valgrupo - Transformação e Criação Agropecuária, <https://www.valgrupo.pt/> (accessed Mar. 29, 2024).

15 - Vue.js, <https://vuejs.org/> (accessed Mar. 30, 2024).

DECLARAÇÃO

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título “*Estágio no Grupo Valgrupo*”, é original e foi realizado por Estudante Gonçalo Miguel Conceição Vicente (2210510) sob orientação de Professor Doutor Carlos Manuel Gonçalves Antunes (carlos.antunes@ipleiria.pt).

Leiria, Março de 2024

Estudante Gonçalo Miguel Conceição Vicente