



UNIVERSIDADE  
LUSÓFONA

# Web App Serralharias

## Trabalho Final de curso

Relatório Final

Nome do Trabalho: Web App Serralharias

Nome do Aluno: João Rodrigues

Nome do Orientador: Luís Gomes

Trabalho Final de Curso | LEI | Data 23/04/2023

[www.ulusofona.pt](http://www.ulusofona.pt)

## Direitos de cópia

*(Web App Serralharias), Copyright de João Rodrigues, ULHT.*

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

---

---

## Índice

1.	Identificação do Problema.....	6
2.	Viabilidade e Pertinência .....	7
3.	Benchmarking.....	8
4.	Engenharia.....	11
4.1	Levantamento e análise de requisitos .....	11
4.2	Diagramas de Casos de Uso .....	15
4.3	Modelos relevantes .....	18
4.4	Estrutura em Árvore .....	22
5.	Solução Proposta.....	26
5.1	Introdução .....	26
5.2	Arquitetura .....	26
5.3	Tecnologias e ferramentas utilizadas .....	27
5.4	Implementação .....	29
5.5	Abrangência .....	31
6.	Método e Planeamento .....	33
7.	Resultados .....	35
8.	Conclusão e trabalhos futuros .....	41
	Anexos e ficheiros .....	44
	Bibliografia .....	45
	Glossário.....	46

---



## Índice de Figuras

<b>Figura 1</b> – Primeiro esboço.....	11
<b>Figura 2</b> - Lista de Componentes e Processos.....	12
<b>Figura 3</b> – Método de avaliação.....	12
<b>Figura 4</b> – Lista de requisitos a ser desenvolvidos.....	13
<b>Figura 5</b> – Lista de requisitos implementados /não implementados.....	14
<b>Figura 6</b> – Login/Registo Use_Case.....	15
<b>Figura 7</b> - Catálogo Use_Case.....	16
<b>Figura 8</b> - WishList Use_Case.....	17
<b>Figura 9</b> - User Management Use_Case.....	17
<b>Figura 10</b> - Modelo E-R.....	18
<b>Figura 11</b> - Modelo E-R Final.....	19
<b>Figura 12</b> - Modelo de classes.....	20
<b>Figura 13</b> – User Home Page.....	22
<b>Figura 14</b> – Category Page.....	23
<b>Figura 15</b> – User Product Page.....	24
<b>Figura 16</b> – Admin Product Page.....	24
<b>Figura 17</b> – User Profile.....	25
<b>Figura 18</b> – User Wish List.....	25
<b>Figura 19</b> – User Budget.....	25
<b>Figura 20</b> – Admin Details.....	25
<b>Figura 21</b> – Admin User List.....	25
<b>Figura 22</b> – Arquitetura.....	26
<b>Figura 23</b> – Servidor.....	30
<b>Figura 24</b> – Registered users.....	35
<b>Figura 25</b> – ImageBD.....	35
<b>Figura 26</b> – ImageToProductAssociation.....	35
<b>Figura 27</b> – ProductBD.....	36
<b>Figura 28</b> – WishListBD.....	36
<b>Figura 29</b> – CategoryBD.....	36
<b>Figura 30</b> – Products on User WishList.....	37
<b>Figura 31</b> – Category page.....	37
<b>Figura 32</b> – Password Hashes.....	38
<b>Figura 33</b> – Erros/Falhas de segurança ao fazer deploy.....	38
<b>Figura 34</b> – Teste de Compatibilidade no Microsoft Edge.....	39
<b>Figura 35</b> – Teste de Compatibilidade no FireFox.....	40

## Resumo

Este relatório descreve o desenvolvimento de uma aplicação web destinada à compra e venda de produtos e serviços relacionados à serralharia. O projeto teve início com a identificação de desafios operacionais enfrentados pela empresa, que incluíam processos desatualizados no que toca à interação com clientes e na gestão dos negócios. Para abordar essas adversidades, foi proposta uma solução tecnológica baseada em Django, um framework de desenvolvimento web em Python, escolhido devido à sua robustez, rápido desenvolvimento, adaptabilidade e segurança. A aplicação segue uma arquitetura de três camadas, com o frontend em HTML, CSS e JavaScript, conectado a um backend Django. O backend contém a lógica de negócios e a interação com a base de dados MySQL, que é hospedada em um container Docker, gerenciada por meio da plataforma DBeaver. A criação de modelos de dados, configuração de settings e programação de funcionalidades foram passos fundamentais no processo de desenvolvimento. A aplicação resultante permite aos utilizadores explorar produtos e serviços, personalizar produtos, solicitar orçamentos especializados sobre os produtos personalizados e simplificar a gestão de clientes, produtos e pedidos. Com esta solução tecnológica, espera-se otimizar a eficiência operacional, impulsionar o crescimento e fornecer uma experiência melhorada aos clientes e uma gestão facilitada por parte da administração, enquanto se posiciona de forma competitiva no mercado digital em constante evolução.

## Abstract

This report describes the development of a web application designed for buying and selling products and services related to metalwork. The project began with the identification of operational challenges faced by the company, including outdated processes in customer interaction and business management. To address these challenges, a technological solution based on Django, a Python web development framework, was proposed. Django was chosen for its robustness, rapid development, adaptability, and security. The application follows a three-tier architecture, with the frontend developed in HTML, CSS, and JavaScript, connected to a Django backend. The backend contains the business logic and interaction with the MySQL database, which is hosted in a Docker container and managed through the DBeaver platform. Creating data models, configuring settings, and programming functionalities were fundamental steps in the development process. The resulting application allows users to explore products and services, customize products, request specialized quotes for customized products, and simplify customer, product, and order management. With this technological solution, it is expected to optimize operational efficiency, boost growth, provide an enhanced customer experience, and facilitate administration, all while positioning the company competitively in the ever-evolving digital market.

**Link para repositório no git:** <https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI323-SerralhariasWebApp>

**Link para vídeo demonstrativo:** [https://youtu.be/u6VEu\\_gzrlU](https://youtu.be/u6VEu_gzrlU)

## **1. Identificação do Problema**

Serralharia é uma indústria com origens muito antigas, que pode ou não estar dentro do ramo da construção civil, que abrange muitas vertentes, desde a criação de pequenas peças de metal para decoração de imóveis, grades, portões ou até grandes infraestruturas metálicas.

Visto ser um ofício já muito antigo, desde sempre idealizado pelas nossas cabeças e projetado pelas nossas mãos, os atuais profissionais, cada vez mais velha dada a pouca aderência dos mais novos à indústria, raramente procuram o apoio que o desenvolvimento tecnológico pode vir a oferecer (à exceção de grandes empresas) no seu negócio.

Isto cria dois problemas muito relevantes. Primeiro, ao não se digitalizarem, todo o seu marketing é baseado na palavra de boca, que apesar de resultar há algumas décadas, nos dias de hoje não tem a mesma eficácia, visto estarmos numa era digital em que tudo se faz no conforto dos nossos sofás. Isto impacta diretamente o nosso volume de negócios, porque estamos todos os dias a perder potenciais clientes para as grandes empresas como IKEA, Leroy Merlin, que disponibilizam estes serviços online.

Segundo, e baseando-me na minha experiência, esta metodologia de trabalho antiquada baseada em papel e caneta, traz adversidades e limitações a níveis internos. Existe uma má gestão de stock de materiais, a relação com o cliente é pouco eficiente, a relação entre o cliente e o seu produto é quase inexistente, existe uma execução repetitiva de tarefas recorrentes, como o preenchimento de fichas de faturas e guias de transporte, e por fim, orçamentos que podiam ser mais automatizados. Todas estas implicações causam um menor fluxo de trabalho e uma menor satisfação por parte do cliente.

É preciso entender que sempre existiu grande oportunidade de mercado em produtos personalizados, dada a sua versatilidade. No entanto, a procura para este serviço nas pequenas empresas diminuiu, devido ao desenvolvimento tecnológico em conjunto com a não digitalização e a falta de divulgação. As grandes superfícies oferecem produtos padrão produzidos em bruto e vendidos em massa, que muitas vezes não correspondem 100% às necessidades das pessoas, mas por falta de conhecimento das mesmas, a facilidade de acesso e preço reduzido acabam por vencer.

Dito isto, como venho de uma família de serralheiros e trabalho na indústria à cerca de 4 anos, achei por bem usar aos meus conhecimentos de ambas as áreas (Informática e Serralharia) de forma a tentar criar uma solução que venha a resolver este problema.

## 2. Viabilidade e Pertinência

Criando uma solução tecnológica para este problema, conseguimos resolver muitos das adversidades identificadas, como por exemplo:

Com a utilização de um catálogo online bem como uma ferramenta de modelação parcial de peças, o cliente consegue sem grande esforço, desde o primeiro contacto, ver o tipo de trabalho realizado pela empresa bem como os seus serviços, e se decidir ser cliente, pode imediatamente criar a sua conta, modelar o seu produto e pedir um orçamento especializado à peça que personalizou, tudo sem precisar de contactar ninguém, enviar fotos de peças idênticas à desejada ou até mesmo desenhá-la. Com esta implementação é possível iniciar um trabalho em menos de um dia, processo este que atualmente pode demorar dias ou até mesmo uma semana até se chegar a um acordo, depois de várias chamadas por telefone ou até mesmo reuniões. Esta implementação não só ajuda na agilidade inicial em começar a produção, mas também aumenta a ligação que o cliente tem com o seu produto e o contacto que tem com a empresa.

Ao integrar um sistema de gestão através de uma ferramenta online, a empresa pode começar a ser gerida 100% digitalmente, guardando todos os dados dos clientes e faturas associadas numa BD. É possível também atualizar ativamente o seu inventário, fazer uma melhor organização dos orçamentos completos ou pendentes e começar a utilizar templates para a realização de guias, faturas e orçamentos. Após a realização deste projeto académico, esta solução visa a ser utilizada praticamente e ativamente pela empresa. É esperado que a mesma comece a funcionar de forma mais eficiente, não desperdiçando horas de trabalho em tarefas redundantes que poderiam ser aproveitadas na produção e na resposta a pedidos de orçamentos. Isto vai resultar num aumento significativo do fluxo de trabalho e volume de negócio, em comparação ao que a empresa apresenta na atualidade e ao mesmo tempo lança-la no mundo digital onde existem mais potenciais clientes e um marketing mais facilitado.



### 3. Benchmarking

Nos dias que correm, temos muitas ferramentas ao nosso dispor para utilização com subscrição gratuita, que isoladamente, permitem-nos contornar alguns dos problemas mencionados sem exigir qualquer tipo de desenvolvimento ou contratações. Como por exemplo o “Toconline”, Instagram, PHC Go ou Jotform, no entanto talvez não sejam as soluções mais indicadas, vamos aprofundar:

**Toconline** - O toconline é um software dedicado à criação de faturas e guias de transportes e à emissão das mesmas à AT, neste caso e só por si, não resolve o nosso problema uma vez ter uma finalidade única não atendendo a nenhum dos outros requisitos apresentados na nossa solução.

**Instagram** - O Instagram é uma rede social usada para diversas finalidades por todo o tipo de empresas/particulares.

Vantagens:

- Ótima ferramenta/plataforma de Marketing gratuito, praticamente inigualável em eficácia
- Plataforma ideal para primeiras interações com o cliente
- Boa solução para a criação de uma galeria/catálogo online
- Acesso a números estratégicos como: número de visitas na página, número de visualização por post, dashboard com gráficos de crescimento

Desvantagens:

- Menor credibilidade, pouco profissional
- Estamos limitados às funcionalidades da app
- Não possui CRM, o que dificulta a gestão dos clientes
- Não responde à maior parte das necessidades propostas na solução

**PHC GO** – É um software dedicado a empresas que permitia uma fácil solução de algumas áreas-chave do nosso negócio, porém, neste contexto não se pode considerar como concorrência direta.

Vantagens:

- Adaptação à empresa
- Gestão de serviços, inventário, faturas, guias, utilizadores
- CRM completo
- Serviço de Faturação e Guias de transporte, com emissão automática
- Permite fazer migração de dados
- Atualização automática em preços nos produtos
- Ferramenta de desenvolvimento com módulos pré-programados (Comercial, Segurança, Administrativa) prontos a utilizar

Desvantagens:

- Serviço pago
- Necessidade de nos sujeitarmos à linguagem da ferramenta
- Não somos titulares dos direitos de autor
- Open Source, partilhado com a comunidade, qualquer produto desenvolvido iria ficar disponível a outros possíveis empregadores
- Não existe possibilidade de criação de catálogo

Dito isto, apesar de ser uma ferramenta brilhante, não tem o potencial de criar resposta aos nossos requisitos, poderia ser apenas, uma ferramenta de auxílio no desenvolvimento de alguns dos seus componentes.

**Jotform** – Ferramenta de desenvolvimento para catálogos online

Vantagens:

- Gratuito
- Fácil de usar e rapidamente ter um catálogo operacional bastante completo
- Possui várias integrações bem estabelecidas e com renome
- Análise de resultados e relatórios de crescimento automatizados
- Sistema de CRM com templates de formulários já definidos
- Marketing via e-mail
- Integração com redes sociais
- Processamento de pagamento com várias opções disponíveis

Desvantagens:

- Leque de templates pré-definidos
- Sistema de personalização de produtos inexistente
- Utilizador não pode fazer pedidos de orçamento
- Não possui ferramenta de gestão de faturas, guias e orçamentos
- Comunicação com clientes é feita exclusivamente através da integração com as redes sociais
- Impossível fazer uma gestão cativa de inventário
- Não somos titulares dos direitos de autor

Resumindo e depois de avaliarmos outras ferramentas/softwarewares que poderiam potencialmente vir a solucionar o nosso problema, chegámos a conclusão de que nenhum deles, individualmente, preenche todos os requisitos necessários que identificámos nos capítulos anteriores.

Poderíamos, ao tirar proveito de todas estas ferramentas em conjunto, resolver parcialmente o nosso problema sem necessitar de qualquer tipo de desenvolvimento, no entanto, para conseguir obter vantagem

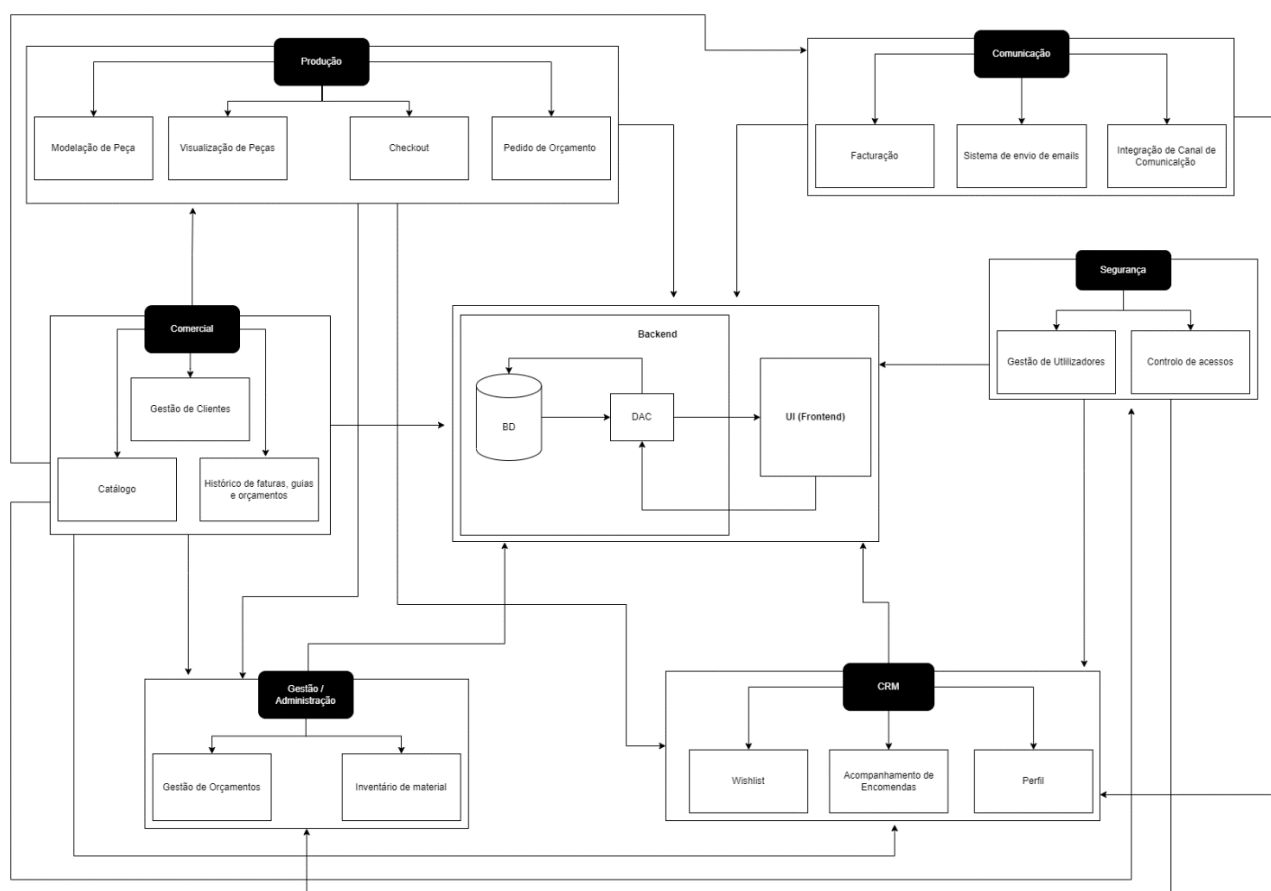
competitiva através da inovação tecnológica e transformação digital, bem como alcançar maior funcionalidade e eficácia operacional, o mais indicado será o desenvolvimento de uma solução integrada com todas as vertentes referidas nos capítulos anteriores, com o objetivo de responder a todas as necessidades da empresa.

## 4. Engenharia

A descrição deste capítulo refere o modelo de negócio e todas as oportunidades de desenvolvimento a concretizar, que devido à falta de tempo no ano letivo, não será desenvolvida na totalidade, no entanto, há uma grande possibilidade de dar continuidade à mesma após a conclusão do TFC.

### 4.1 Levantamento e análise de requisitos

Na figura 1 estamos perante o primeiro esboço da aplicação, onde foram idealizados todos os módulos existentes, acompanhados das suas componentes e as suas relações, com um detalhe de baixo-nível.



**Figura 1** – Primeiro esboço

Como follow up deste diagrama, temos a lista de processos que apresentamos a seguir na figura 2. Esta lista representa uma evolução deste diagrama em que se desenvolveu todos os seus módulos e componentes, de forma a criar os processos necessários para a sua implementação, posteriormente foi feita uma avaliação de cada processo em termos de prioridade e impacto usando a escala Likert de 5 níveis (figura 3).

Componentes	Módulos	Processos	Prioridade	Impacto
<b>1. Comercial</b>				
1.1	Catálogo	Digitalizar os produtos ou serviços através de um catálogo digital Descrição pormenorizada de cada produto/serviço Utilizador pode, no momento da compra, pedir a alteração das medidas sem alterar a base do produto e seguir para um orçamento adaptado às suas necessidades	5	5 Criar um catálogo digital com todos os nossos produtos vai ter um grande impacto económico, pois facilita o acesso do cliente ao produto
1.2	Gestão de clientes	Ficha de cliente a preencher com todos os dados do cliente e guardar na BD: - Nome - Email - Morada (Opcional) - Contacto (Opcional) - Método de Pagamento (Opcional)	4	4 Visto o atual funcionamento da empresa ser 100% tradicional com nenhum elemento digital, ter todos os dados dos clientes armazenados em BD, vai facilitar o acesso aos mesmos e melhorar a produtividade
1.3	Histórico	Poder aceder históricos de faturas, guias e orçamentos através da Administração ou CRM	1	1
<b>2. Produção</b>				
2.1	Modelação da peça	Sistema de customização/modelação de variados produtos de forma seccionada, onde o cliente escolhe o estilo de cada parte do seu produto com medidas personalizadas ao seu espaço (fator diferenciador da maioria da concorrência). Usando o exemplo de uma mesa: serão disponibilizados diferentes tipos de pé de mesa, vários tipos de tampos e cores, onde o utilizador é sujeito a uma sequência de interfaces em que escolhe o que pretende de forma a alcançar uma peça personalizada a seu gosto.	2	2 Este sistema, apesar de inovar e apelativo, não passa uma funcionalidade extra para a adicionar ao catálogo original (1.1), dito isto, não se prevê que tenha grande influência no volume de negócio
2.2	Visualização do produto	Mostrar o produto final em forma de 3 vistas: topo, perfil e angular.	2	2 Funcionalidade do processo (2.1)
2.3	Pedido de orçamento	O cliente pode, através do preenchimento de um form, fazer e enviar um pedido de orçamento de um produto personalizado	5	5 Este processo vai complementar o (1.1) e é de igual importância no processo de compra e venda
2.4	Check-out	O cliente pode prosseguir com a ordem de compra/encomenda do produto O cliente pode escolher em entrega ao domicílio ou levantamento na oficina	3 Baixa prioridade e impacto, devido à possibilidade de se finalizar o processo de compra sem o auxílio da app, através do pagamento por referência bancária	3 Baixa prioridade e impacto, devido à possibilidade de se finalizar o processo de compra sem o auxílio da app, através do pagamento por referência bancária
<b>3. Comunicação</b>				
3.1	Sistema de envio de emails	Envio automático de email ao cliente: - Confirmação do email no momento do registo da conta - Sistema de recuperação de passwords - Email discriminado com todos os dados referentes à compra de um produto por parte do cliente	4	1
3.2	Sistema de conversação	Integração do sistema de conversação com o cliente através da utilização da API do WhatsApp/Facebook - Criação de um chat de conversação a ser usado diretamente no website	2	2
3.3	Faturação	Usar as APIs disponíveis de outros softwares para: - Emissão de faturas à AT - Emissão de guias de transporte à AT - Envio automático de email descritivo em ambos os casos	1	5 Conseguir implementar este processo, significa conseguir gerir o negócio 100% através da app. É uma vantagem enorme em termos de produtividade, que vai resultar em maior rentabilidade
<b>4. Segurança</b>				
4.1	Controlo de acessos	- Conceder, remover permissões a todo o tipo de utilizadores (Admin e Cliente) - Validar acessos	4	1
4.2	Gestão de utilizadores	Operações do tipo: apagar, editar, criar, bloquear utilizadores Recolha de dados analíticos	4	1
<b>5. CRM</b>				
5.1	WishList	Permitir ao utilizador guardar produtos do seu interesse numa WishList Ativar as notificações para quando a peça estiver disponível	5	4 É do interesse do cliente poder guardar os seus produtos favoritos e ser notificado quando os mesmos ficam disponíveis. É do interesse da empresa assegurar o maior número de vendas possível
5.2	Acompanhamento de encomendas	Conseguir que o utilizador: - Veja o estado dos orçamentos que efetuou - Tenha acesso ao estado de produção do seu produto	3	3 Um cliente com mais benefícios é um cliente mais satisfeito, dar a possibilidade de acompanhar o desenvolvimento do seu produto vai inspirar mais confiança e segurança
5.3	Perfil	O cliente pode: - Ver os seus dados - Atualizar os seus dados caso seja necessário - Adicionar dados que possam estar em falta	3	2 Funcionalidade importante para o bom funcionamento da app mas sem grande impacto na vertente do negócio e cliente
<b>6. Administração</b>				
6.1	Gestão de orçamentos	Aceder aos pedidos de orçamentos recebidos e alterar o seu estado: - A orçamentar (aceite pela empresa, orçamento que se encontra em estado de realização) - Estado de espera (orçamento já concluído e enviado ao cliente, à espera de resposta por parte do cliente) - Concluído (orçamento completo e aceite por ambas as partes) - Rejeitado (orçamento rejeitado ou pelo cliente ou pela empresa)  Listar os orçamentos de acordo com o seu estado atual	2	3 Esta funcionalidade está diretamente relacionada com o processo (5.2). Apesar de não ser o que reflete maior influência no negócio, não deixa de ser importante no que toca a uma maior organização e comunicação entre empresa e cliente
6.2	Inventário	Ver o inventário de material disponível Conseguir editar para: - Remover material aquando usado - Restabelecer o stock - Adicionar novos tipos de materiais	2	5 Ao contrário do que acontece na atualidade, ter uma gestão de material em formato digital, influência diretamente o uso da matéria-prima, com uma maior rentabilidade, maior aproveitamento, mais organização e menos desperdício

Figura 2 - Lista de Componentes e Processos

<b>Prioridade</b>	Prioridade define a ordem dos processos/módulos que são pré-condições para garantir a qualidade dos requisitos mais urgentes e importantes. Avaliados usando a escala de Likert de 5 níveis.
<b>Impacto</b>	Impacto representa os módulos/processos que ao serem desenvolvidos vão mostrar mais resultados em menos tempo, representa também os processos que após digitalizados vão retornar mais rentabilidade e dar aos clientes mais controlo e funcionalidades.
<b>Processo de escolha de processos para desenvolvimento</b>	A ordem de desenvolvimento dos processos/módulos vai ser determinada com base num cálculo percentual tendo em conta as colunas de Prioridade e Impacto

Figura 3 – Método de avaliação

Após feita a qualificação de todos os processos em termos de prioridade e impacto, podemos observar que os mais relevantes seriam: o 1.1 (Catálogo), o 2.3 (Pedido de orçamento) e o 5.1 (WishList).

Contudo, apesar de o processo 2.3 (Pedido de orçamento) ter uma elevada relevância equivalente ao 1.1 (catálogo), observa-se uma dependência do mesmo (1.1), sendo assim necessário dar-lhe prioridade no desenvolvimento.

Mesmo depois de verificar que os processos da componente 4 (Segurança) tem medidas de impacto baixas (1), enquadrando com o esforço do primeiro processo a ser desenvolvido (1.1), os processos da componente 4 (Segurança) acabam por ser pré-requisitos ao desenvolvimento do processo 1.1 (Catálogo), assegurando os requisitos mínimos para a criação de perfis de user e admin com todos os dados necessários, atribuição de permissões e controlo de acessos. Tornando assim o Catálogo uma solução possível e comercialmente interessante.

Em baixo, apresento então a listagem dos requisitos referentes aos processos 1.1, 4.1, 4.2 e 5.1 que efetivamente vão ser desenvolvidos.

Área	Temática	ID	Descrição Curta do Requisito	Prioridade	Impacto
1. Catálogo	1.1 Digitalização de produtos ou serviços através de um catálogo digital	1.1.1	Criar Produto	5	3
		1.1.2	Alterar/Editar Produto (Editar descrição medidas, e número de stock, preço)	3	3
		1.1.3	Procurar Produto	4	5
		1.1.4	Apagar Produto (Produto temporariamente desativado/invisível)	2	2
		1.1.5	Listar Produto (Tornar o produto ativo/visível)	5	5
		1.1.6	Alterar a ordem dos produtos no catálogo	2	2
		1.1.7	Selecionar categoria	4	5
	1.2 Ajuste do produto	1.2.1	Identificar o Product ID	2	1
		1.2.2	Identificar o cliente	2	1
		1.2.3	Text Box dedicada a alterações desejadas pelo cliente (medidas, cor, material)	3	5
		1.2.4	Guardar pedido na BD (a ser consultado no CRM)	3	1
		1.2.5	Pedido de orçamento automático com as alterações pretendidas	3	5
2. Segurança	2.1 Controlo de Acesso	2.1.1	Adicionar permissões por tipos de user (Admin, Cliente ou Guest)	4	1
		2.1.2	Login	5	4
		2.1.3	Registo	5	4
		2.1.4	Validar credenciais de acesso/login	5	4
		2.1.5	Guardar credenciais de registo	5	4
		2.1.6	Verificar a flag de admin no momento do login e redirecionar para o URL correto (página de produto ou página de admin)	5	5
	2.2 Gestão de Utilizadores	2.2.1	Lista de todos os utilizadores	4	2
		2.2.2	Last Login por user	1	1
		2.2.3	Criar utilizador	5	3
		2.2.4	Apagar utilizador	2	1
		2.2.5	Bloquear utilizador	2	1
3. WishList		3.1	Adicionar produto	5	5
		3.2	Remover produto	2	2
		3.3	Preview do produto	4	5
		3.4	Link para a página do produto	3	4

**Figura 4** – Lista de requisitos a ser desenvolvidos

Dando seguimento ao início do desenvolvimento do projeto, fui implementando estes requisitos um a um começando pelo capítulo 1.1 de forma a ser possível a inserção de produtos no website, dei depois continuidade com a implementação do 2.1 para termos users capazes de interagir com os nossos objetos (produtos).

Uma vez concluída a fase anterior passei para a WishList (capítulo 3) seguido do pedido de orçamento (capítulo 1.2) e por fim o 2.2 para ter algum tipo de controlo sobre os utilizadores diretamente pela app sem ter de recorrer ao uso do DjangoAdministrator.

Fica então em baixo (figura5) uma versão atualizada do Exel anterior(figura4) com o status de implementação de todos os requisitos.

Legenda: A verde temos todos os requisitos que foram implementados com sucesso e a vermelho os que ficaram por implementar, neste caso só o 1.1.6.

Área	Temática	ID	Descrição Curta do Requisito	Prioridade	Impacto
1. Catálogo	1.1 Digitalização de produtos ou serviços através de um catálogo digital	1.1.1	Criar Produto	5	4
		1.1.2	Alterar/Editar Produto (Editar descrição medidas, e número de stock, preço)	3	3
		1.1.3	Procurar Produto	4	5
		1.1.4	Apagar Produto (Produto temporariamente desativado/invisível)	2	2
		1.1.5	Listar Produto (Tornar o produto activo/visível)	5	5
		1.1.6	Alterar a ordem dos produtos no catálogo	2	2
		1.1.7	Selecionar categoria	5	5
	1.2 Ajuste do produto	1.2.1	Identificar o product ID	2	1
		1.2.2	Identificar o cliente	2	1
		1.2.3	Text Box dedicada a alterações desejadas pelo cliente (medidas, cor, material)	3	5
		1.2.4	Guardar pedido na BD	3	1
		1.2.5	Pedido de orçamento listado no UserProfile	3	5
2. Segurança	2.1 Controlo de Acesso	2.1.1	Adicionar permissões por tipos de user (Admin, Cliente ou Guest)	4	1
		2.1.2	Login	5	4
		2.1.3	Registo	5	4
		2.1.4	Validar credenciais de acesso/login	5	4
		2.1.5	Guardar credenciais de registo	5	4
		2.1.6	Verificar a flag de admin no momento do login mostrar os elementos correctos de acordo com a sessão	5	5
	2.2 Gestão de Utilizadores	2.2.1	Lista de todos os utilizadores	4	2
		2.2.2	Last Login por user	1	1
		2.2.3	Criar utilizador	5	3
		2.2.4	Apagar utilizador	2	1
		2.2.5	Bloquear utilizador	2	1
3. Wishlist		3.1	Adicionar produto	5	5
		3.2	Remover produto	5	5
		3.3	Preview do produto	5	5
		3.4	Link para a página do produto	5	5

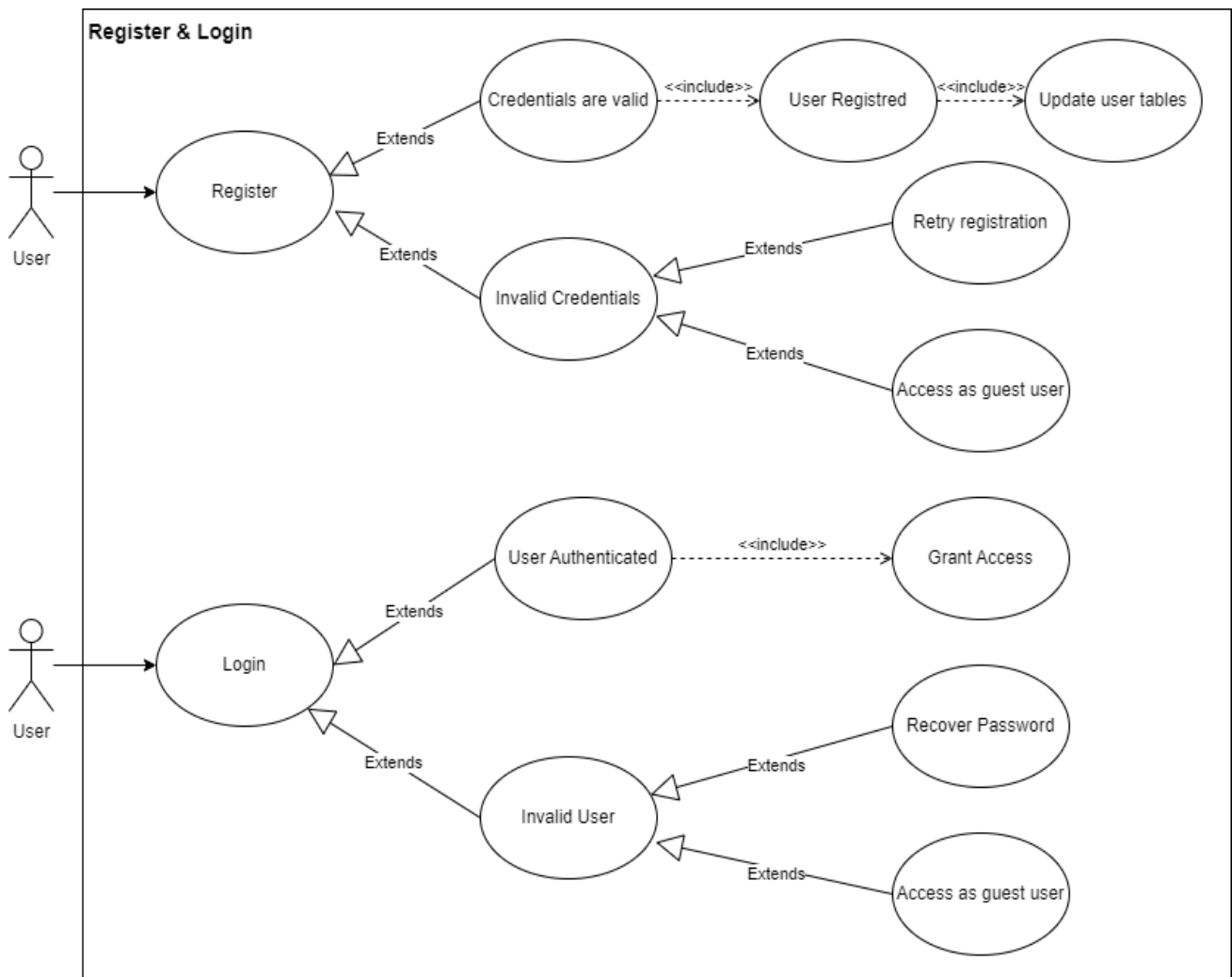
**Figura 5** – Lista de requisitos implementados /não implementados

## 4.2 Diagramas de Casos de Uso

Aqui apresentam-se alguns diagramas de use case, referentes aos processos a serem desenvolvidos mencionados no ponto anterior, nestes use case estão inseridos a grande maioria dos requisitos descritos na

Login:

- Caso de uso representativo das sequências de ações que o utilizador executa ao fazer login e no momento de criação da sua conta, bem como as suas implicações.

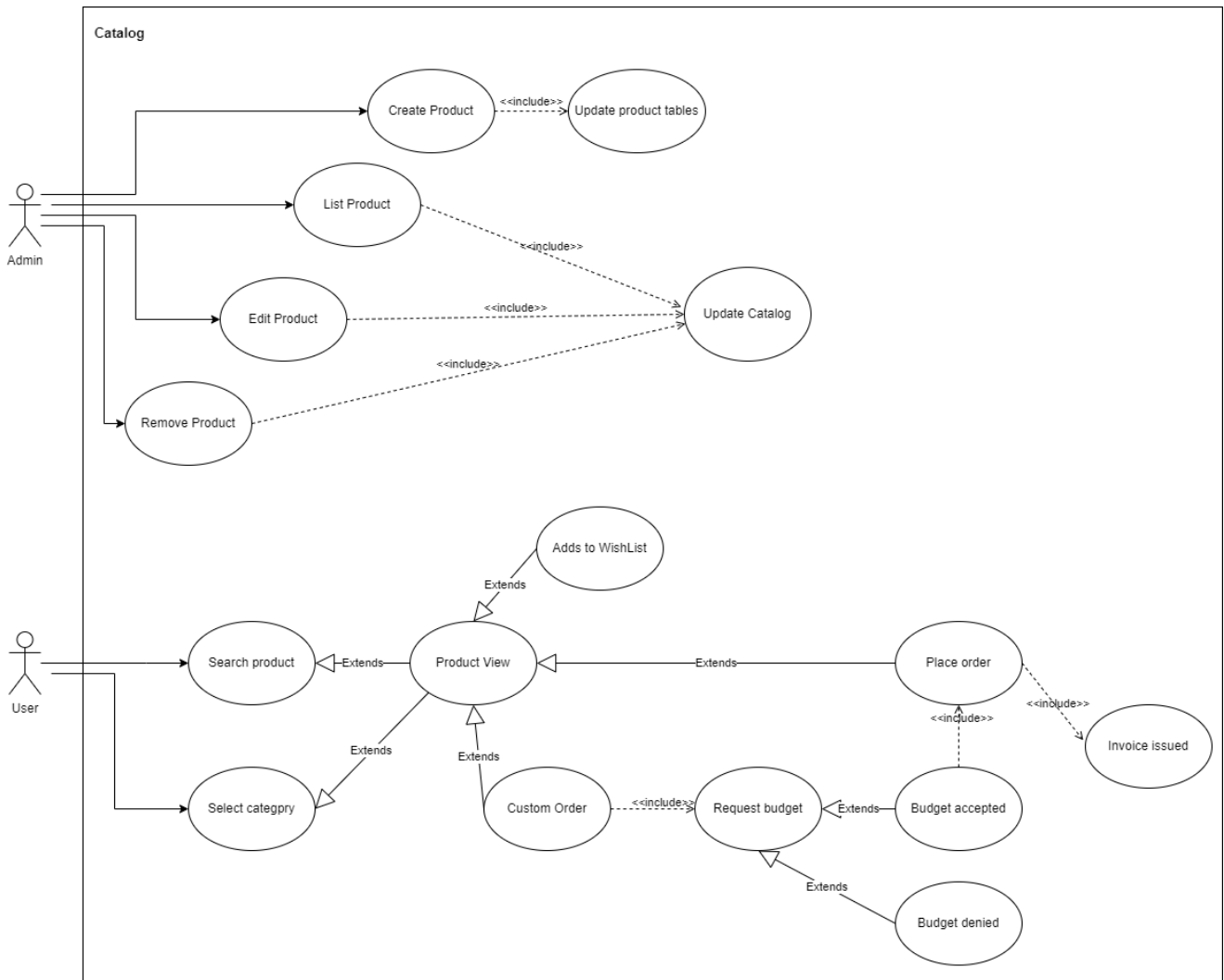


**Figura 6** – Login/Registo Use\_Case



### Catálogo:

- Sequência de ações de manipulação de catálogo disponíveis a um user do tipo admin.
- Sequência de ações que derivam da interação do user com o catálogo, desde a página inicial até à possível encomenda de um produto.



**Figura 7 – Catálogo Use\_Case**

WishList:

- Sequência de ações ou funcionalidades disponíveis a uso por parte do user para utilização e manutenção da sua WishList.

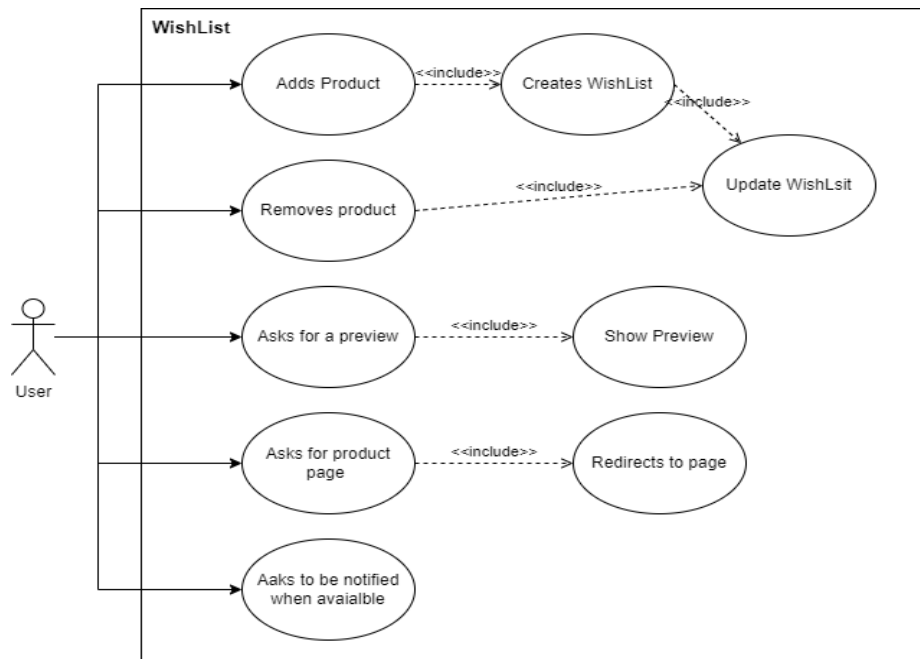


Figura 8 - WishList Use\_Case

User Management:

- Ações/Funcionalidades que um user do tipo admin pode executar para gestão de utilizadores.

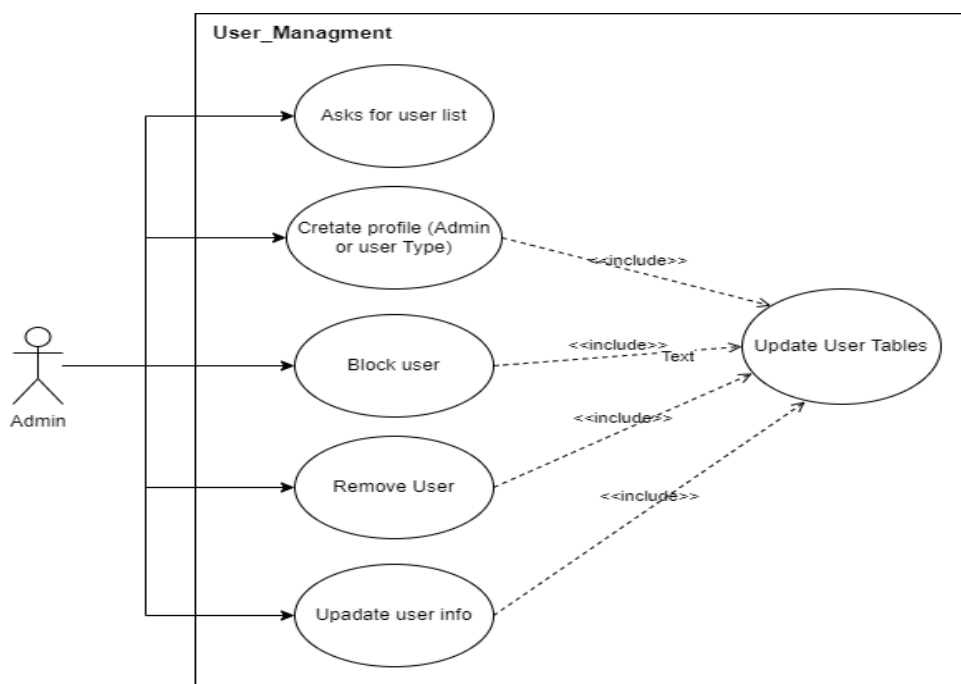
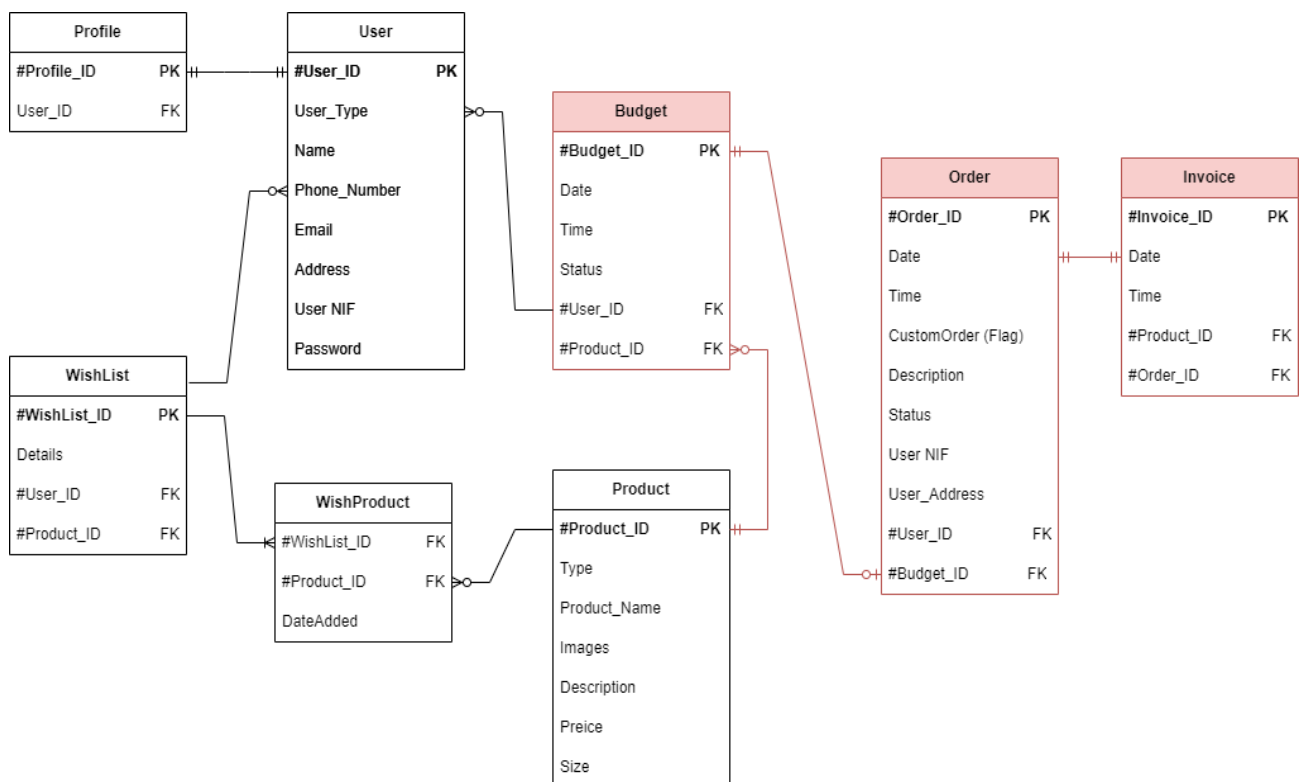


Figura 9 - User Management Use\_Case

### 4.3 Modelos relevantes

#### Modelo E-R:

- Em baixo temos o modelo E-R da estrutura de dados da aplicação.
- Foram colocadas 3 tabelas (Budget, Order, Invoice) em highlight (cor-de-rosa), estas tabelas, apesar de estarem presentes na lista de componentes e processos (figura 2), não constam na lista de requisitos (figura 3), dito isto, não foram seleccionadas como funcionalidades prioritárias e não irão chegar a fase de desenvolvimento no TFC, foram inseridas neste modelo e no modelo de classes por motivos de dar mais corpo e contextualização. Não implica, no entanto, que não venham a ser desenvolvidas pós-TFC.
- O mesmo se aplica ao modelo de classes que se segue (figura 10).



**Figura 10** - Modelo E-R

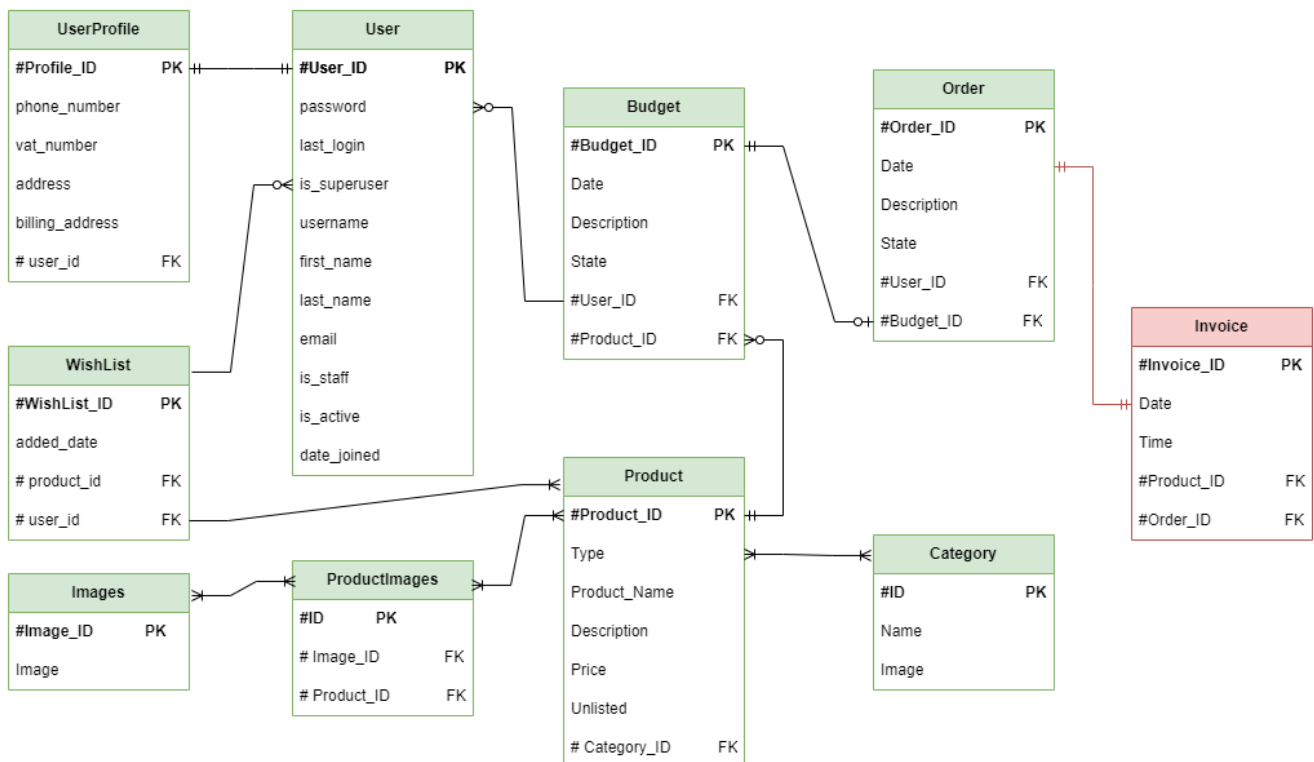
Na fase inicial do projeto, foi desenvolvido este modelo E-R (figura 10) que serviu como base para a implementação do software. No entanto, ao longo do processo de desenvolvimento, tornou-se evidente que várias modificações eram necessárias para garantir a robustez e escalabilidade do sistema.

Como podemos ver na figura 11, em comparação com a figura 10, existem algumas mudanças evidentes:

- Profile passou a User Profile que está associado a User numa relação de 1 para 1, onde serão inseridos dados adicionais referentes ao user, dados estes que não são mandatários no momento do registo.
- Alguns dos dados do User e da WishList também foram alterados por questões logística.
- A Category deixou de ser um atributo de Product e tornou-se num modelo independente para uma melhor gestão de objetos dentro da app.
- Da mesma maneira, as imagens deixaram de ser atributo do Product e foram divididas em dois modelos, o primeiro, (Images), é somente responsável por guardar todas as imagens existentes na app sem qualquer tipo de associação. O segundo, (ProductImages), é responsável pela associação das imagens aos seus devidos produtos.

Legenda/Nota (figura 11): A verde temos os modelos implementados e a vermelho os que não chegaram a ser implementados.

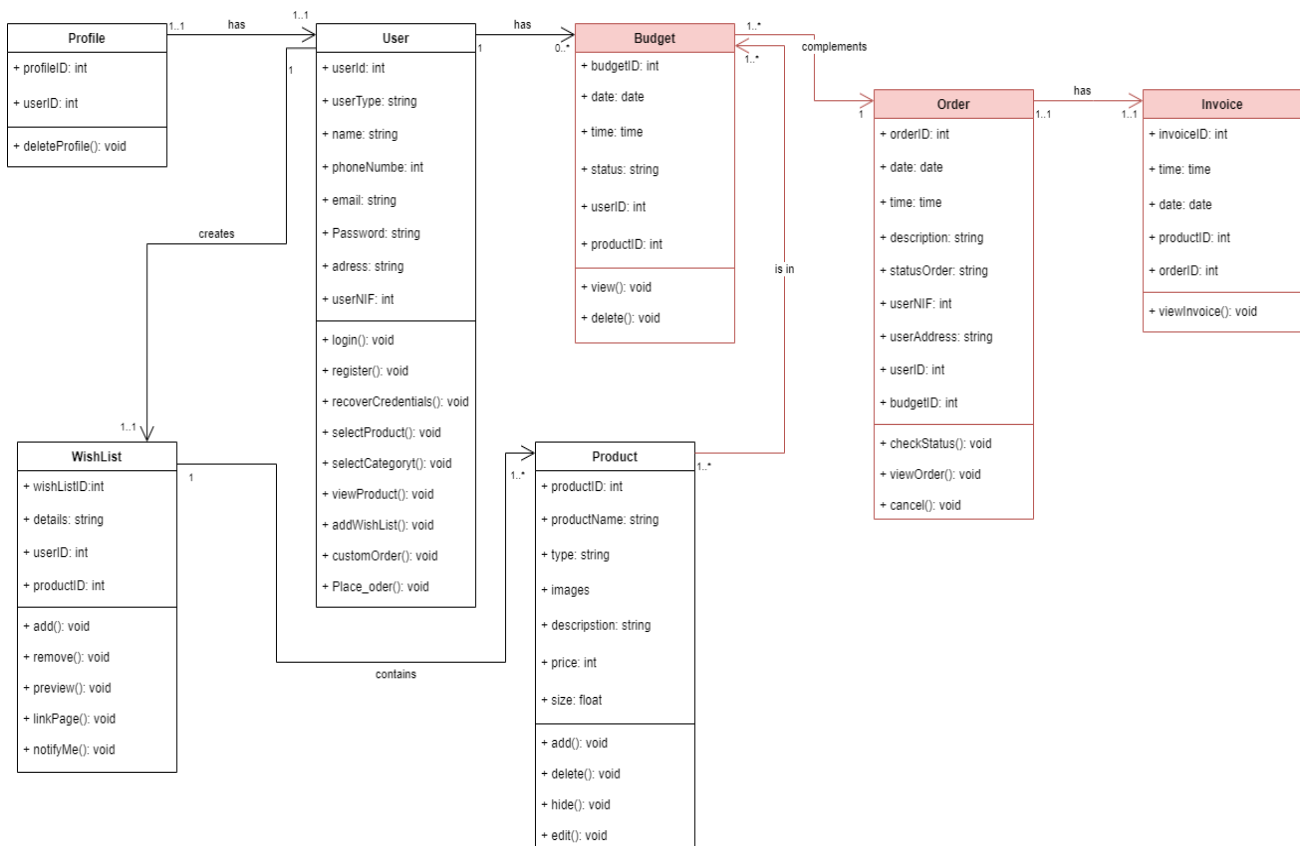
No entanto se olharmos para a figura 10, podemos verificar que os modelos Budget, Order e Invoice já estavam a vermelho antes do início do desenvolvimento definindo que a sua implementação não era esperada, no entanto, ainda foi possível a implementação do Budget e parte da Order deixando apenas o Invoice por fazer.



**Figura 11 - Modelo E-R Final**

## Modelo de Classes:

- Modelo criado para mapear de forma clara a estrutura do sistema, modelando as suas classes, atributos, operações e relações entre objetos.



**Figura 12** - Modelo de classes

A definição inicial de um modelo de classes com todas as variáveis e funções é uma prática comum no desenvolvimento de software. No entanto, quando um projeto é desenvolvido utilizando uma framework como o Django, que segue o padrão de arquitetura Model-View-Controller (MVC), algumas adaptações são necessárias, o que pode levar à invalidação do modelo de classes originalmente idealizado.

No início do projeto, optei por seguir uma abordagem de desenvolvimento orientado a objetos, criando um modelo de classes detalhado, contendo todas as variáveis e funções necessárias para representar os objetos do sistema em questão. No entanto, ao adotar a framework Django e sua arquitetura deparei-me com desafios que tornaram inviável a manutenção do modelo de classes original.

Uma das características fundamentais do Django é a separação de responsabilidades entre Model, View e Controller. O Modelo (Model) no Django é responsável pela representação dos dados e pela interação com a

BD, enquanto a lógica de negócios é manipulada principalmente pelas Views (Views.py), que lidam com a apresentação e interação com o usuário. Isto significa que as funções que inicialmente tinha definido nas classes do modelo deixaram de ser apropriadas, pois a lógica de negócios é movida para as views.

Outra característica é que o Django promove o desenvolvimento rápido ao fornecer funcionalidades pré-construídas para tarefas comuns, como autenticação de usuários, formulários de registo e login e administração da DB ser necessário a escrita de SQL, logo, manter funções nos devidos models como indicado no modelo de classes pode interferir na utilização desses recursos integrados.

Em resumo, embora o modelo de classes inicial tenha sido uma abordagem válida no início do projeto, a utilização de Django MVC exigiu uma revisão e simplificação do modelo, movendo a lógica de negócios para as views e aproveitando as vantagens oferecidas pelo framework.

## 4.4 Estrutura em Árvore

### Home/Landing Page:

Após entrar na app, é apresentada a home page, esta é idêntica no caso de ter sessão com um user do tipo guest, regular ou admin.

Em baixo é apresentada uma imagem da homepage no caso de um user regular já autenticado na sua conta. No caso de ser um admin ou guest, a única diferença é que não é apresentada a opção de adicionar um produto aos **favoritos**.

Neste ecrã temos a opção de ir para uma das **categorias**, qualquer um dos **produtos**, fazer uma **search** ou ir para a **página de perfil**, ou até mesmo adicionar um produto diretamente aos **favoritos**.

Nota: Gostaria de mencionar que o header que contém a **search bar** e o **User Profile** são persistentes em todas as páginas do projeto, por isso não iram voltar a ser mencionadas até ao fim deste capítulo.

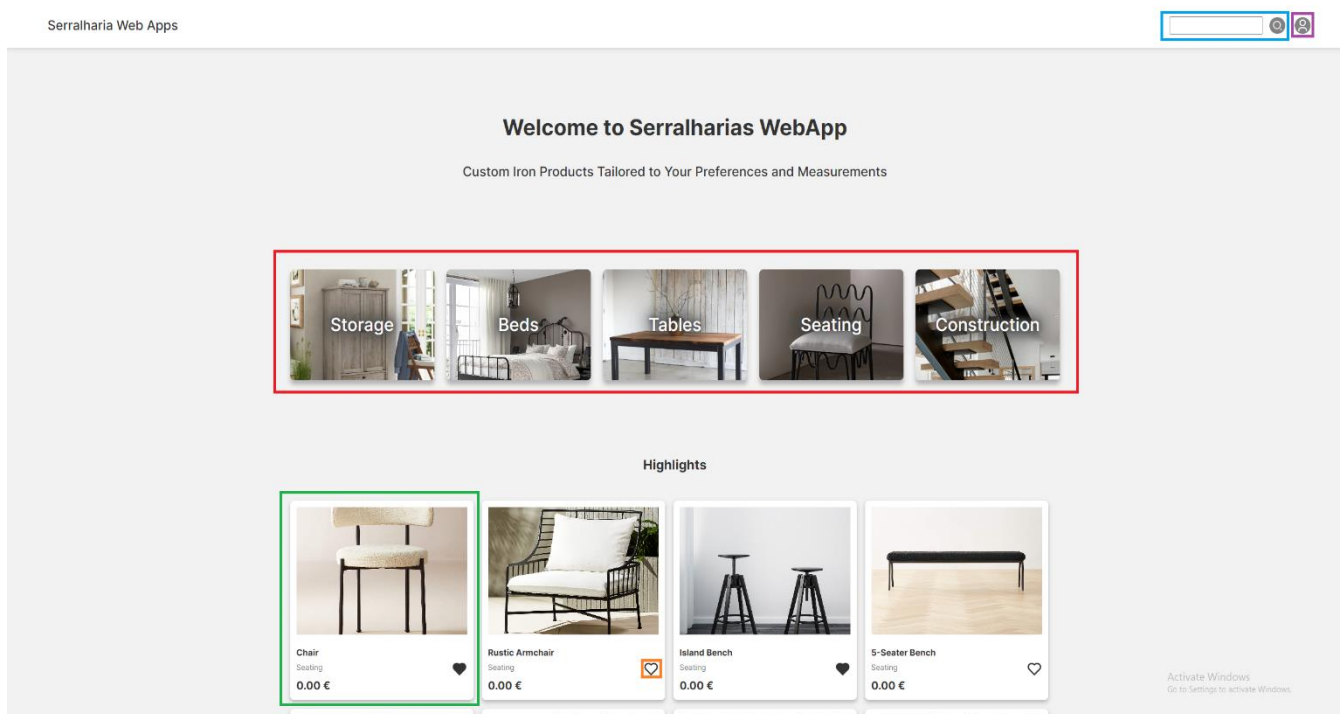
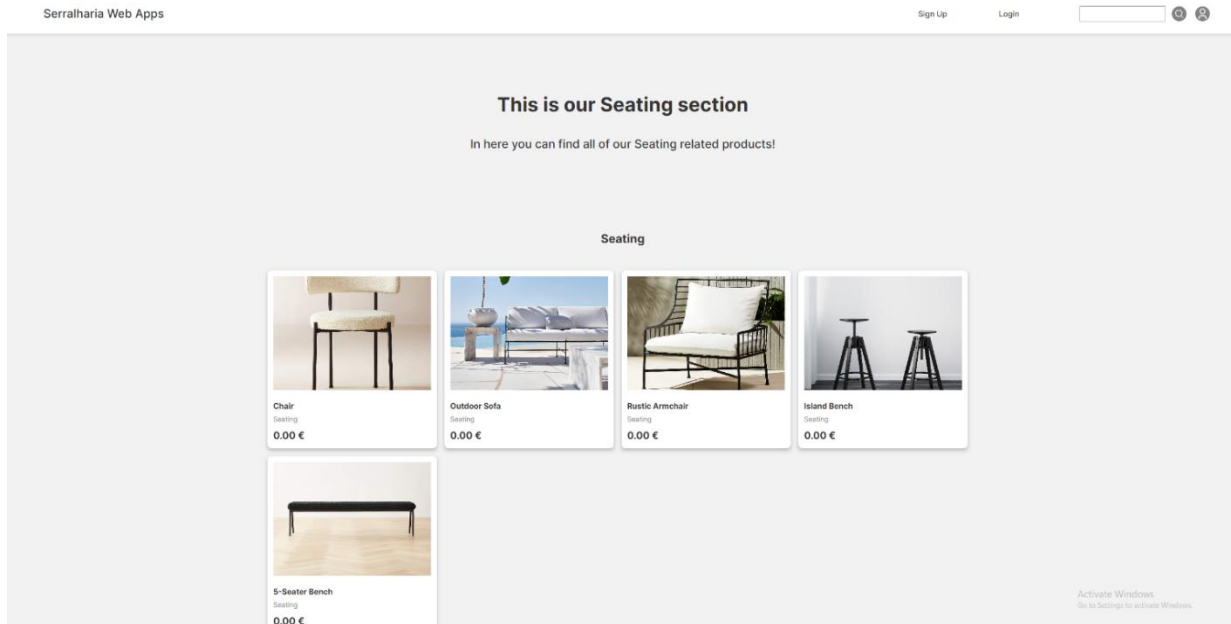


Figura 13 – User Home Page

### Category Page:

Aqui podemos observar todos os produtos relacionados por categoria, neste caso seating products, onde podemos selecionar um deles e passar para a página de produto.



**Figura 14** – Category Page

### Product Page:

No caso de um user regular que se encontre na product page, o mesmo tem algumas opções, pode ver as restantes fotos relacionadas com o produto, pode fazer uma custom order que irá abrir um formulário (ver no vídeo), pode adicionar ou remover o produto da WishList que será automaticamente á sua WishList dentro do seu perfil e pode navegar para outro produto relacionado dentro da mesma categoria.



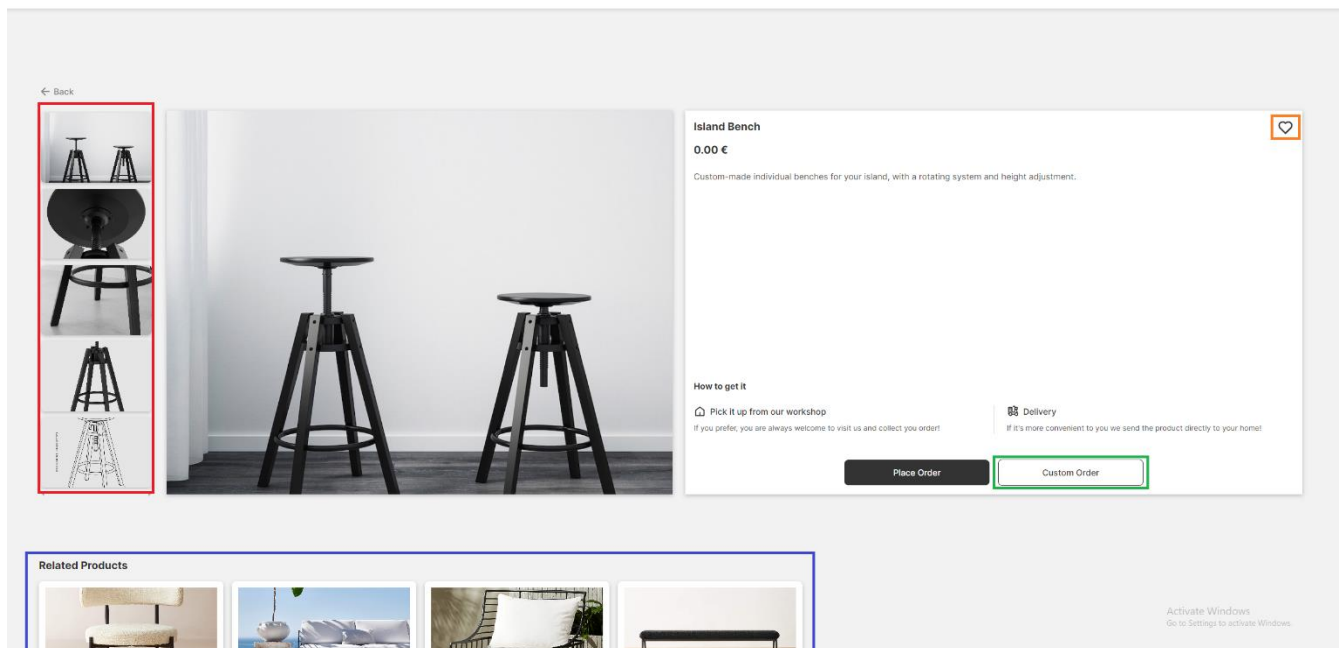


Figura 15 – User Product Page

No caso de um user do tipo admin, este tem funcionalidades um pouco diferentes, perde a capacidade de adicionar á WishList, mas pode **editar** o produto que irá abrir um formulário com especificações a preencher (ver no vídeo), tem também a possibilidade de tornar um **produto invisível** ou até mesmo **apagá-lo** permanentemente.

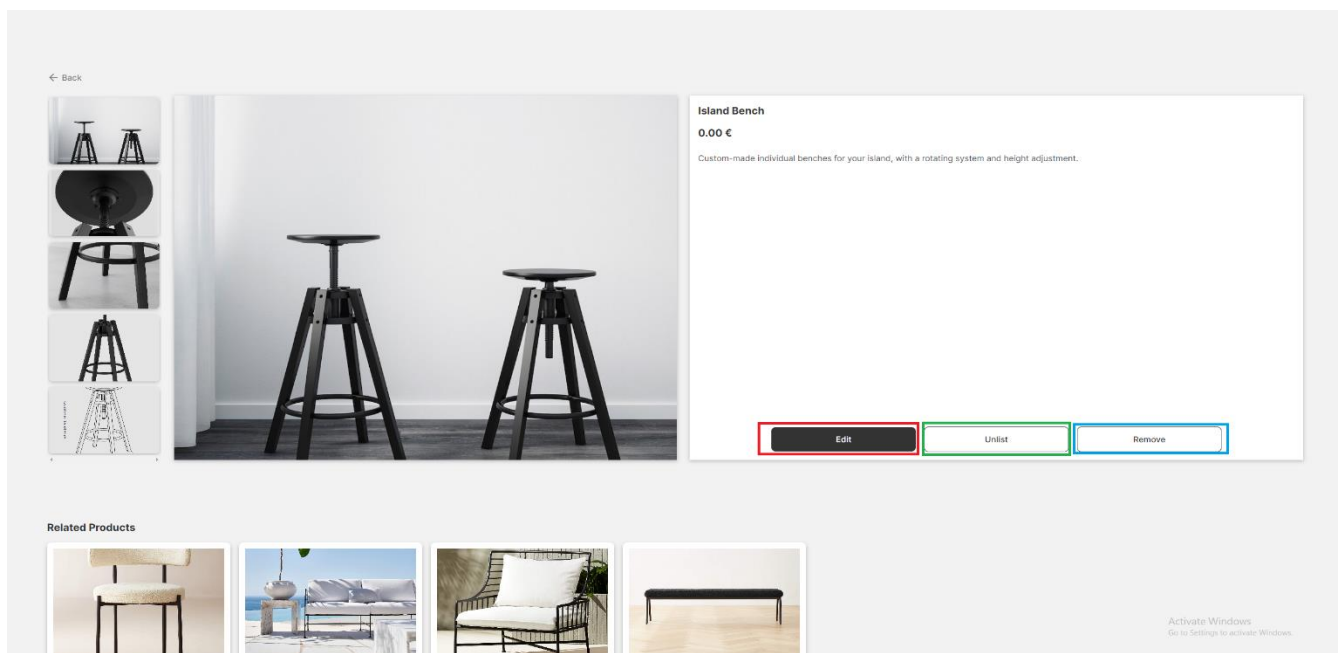


Figura 16 – Admin Product Page

## User Profile:

Com uma sessão de user regular, em qualquer ecrã da app é possível clicar no icon de User Profile. Aqui temos um **sistema de tabs baseado em JavaScript** que nos permite navegar entre diferentes tabs para mostrar informação relevante ao user:

- Profile: Todos os dados guardados no momento do registo bem como um **formulário para introduzir dados adicionais** e um botão de **logout**.
- WishList: Lista de todos os **produtos adicionados**, clicáveis que redirecionam para a página do produto em questão, com botão de **favoritos**.
- Budget: Lista de pedidos personalizados (Custom Oder) com o seu estado atual.

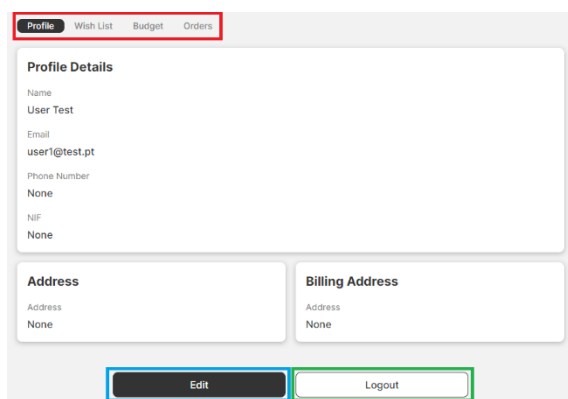


Figura 17 – User Profile

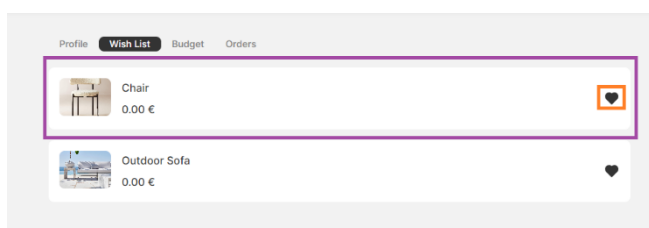


Figura 18 – User Wish List

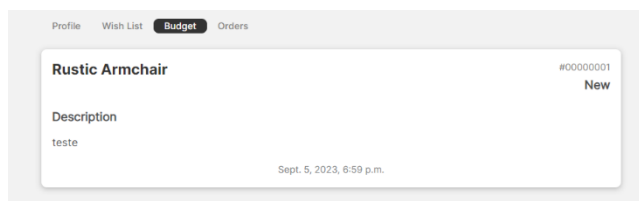


Figura 19 – User Budget

Com uma sessão de admin, a tab de Profile mantém-se, no entanto em vez de vermos uma WishList, vemos **uma lista de todos os users** com algumas informações (como por exemplo **date\_joined** e **last\_login**) e duas operações, **bloquear user** no qual o mesmo fica incapaz de executar o login até que o seu estado seja alterado, e por fim **apagar user**.

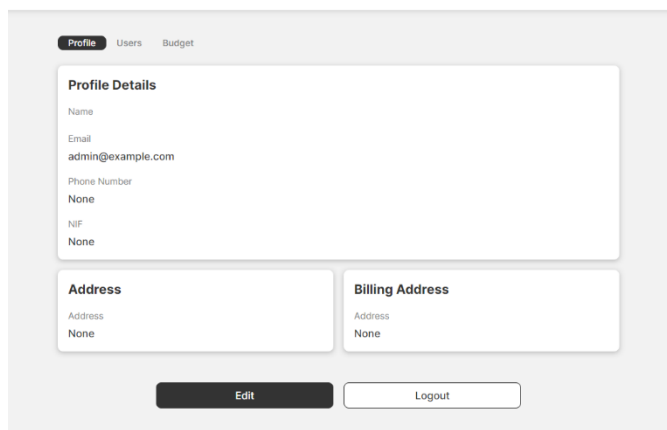


Figura 20 – Admin Details

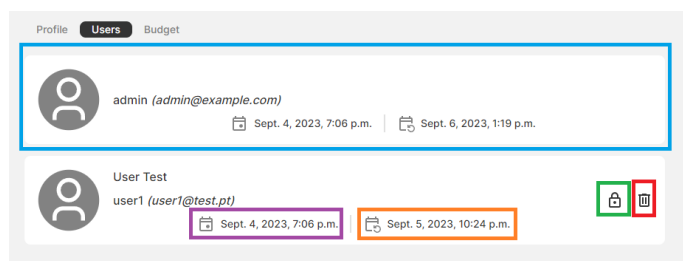


Figura 21 – Admin User List

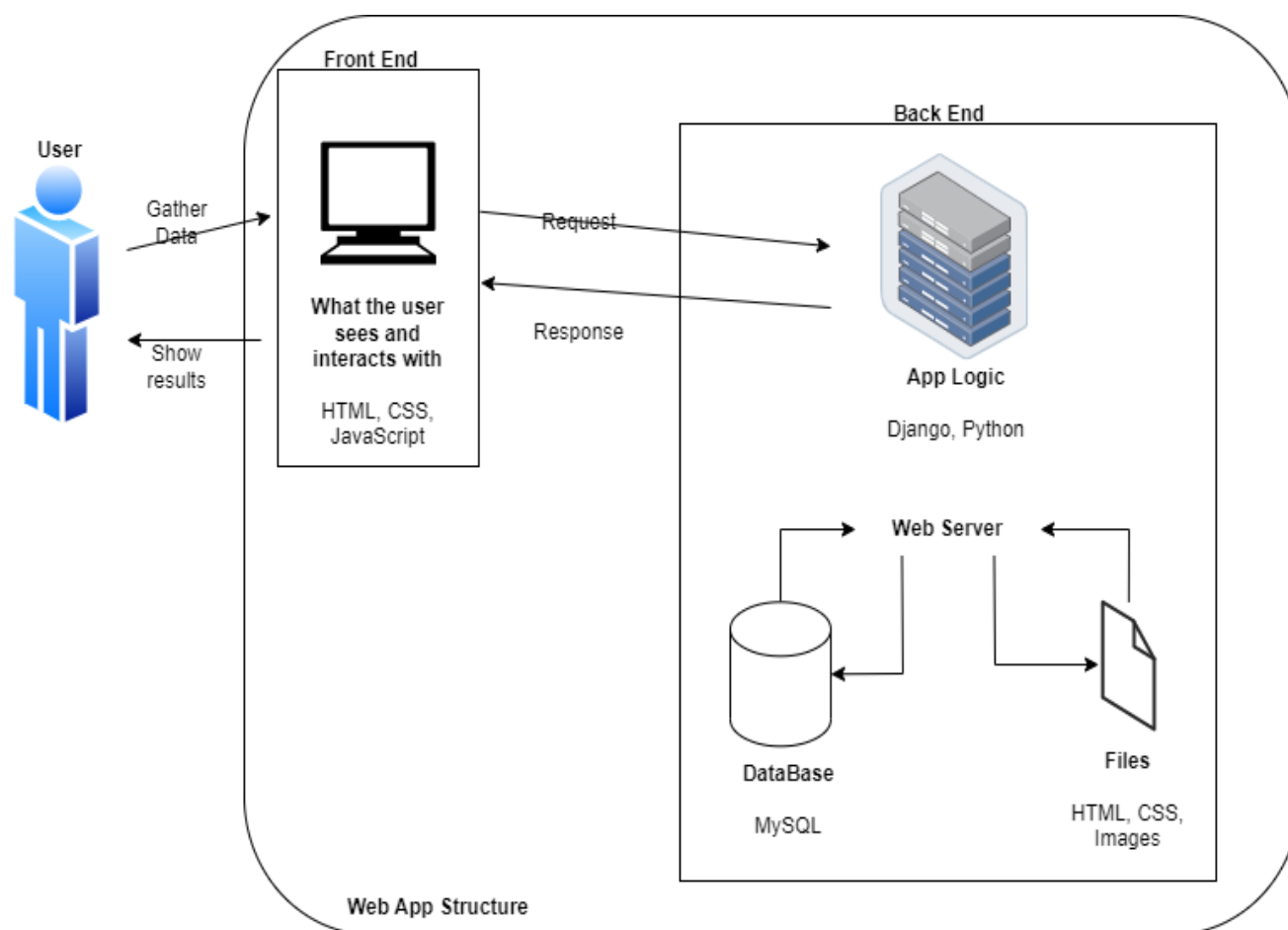
## 5. Solução Proposta

### 5.1 Introdução

A solução proposta no que toca a requisitos e funcionalidades, passa por tudo o que já se falou e demonstrou no capítulo 4, trata-se do desenvolvimento de uma aplicação web que consiga resolver o problema apresentado no capítulo 1.

Terá um desenvolvimento de back office que realiza os processos de gestão interna, de forma a garantir que todas as operações executadas diariamente são executadas sem percalços, e um front office responsável por todas as interações com os clientes e também pelos processos de venda, pós-venda e marketing.

### 5.2 Arquitetura



**Figura 22** - Arquitetura

Componentes são parte essencial da arquitetura de uma WebApp. Determinam as funcionalidades e os recursos necessários para a aplicação e a qualidade da experiência proporcionada ao utilizador, podem ser separadas em algumas categorias:

### UI/UX:

- Responsáveis pelo esboço e teste de todos os ecrãs, bem como a sua sequência e relações. Garante o futuro desenvolvimento do projeto de forma a providenciar uma boa experiência ao cliente.
- Todos os mockups foram desenvolvidos através do Figma.

### Front End:

Desenvolvimento técnico do UI e de todas as suas funcionalidades e processos, incluindo animações, navegação e integração com o back end.

Como ilustrado na figura, o desenvolvimento de front end conta com programação em HTML para estrutura da app, CSS para a parte estética e JavaScript para criação de conteúdo dinâmico e interativo.

### Back End:

- Web Server:
  - O browser que permite ao cliente interagir com a aplicação, fornece conteúdo estático e.g.: páginas HTML, ficheiros, imagens etc...
- Application server:
  - Responsável por todo o tipo de código e lógica da aplicação que gera conteúdo dinâmico.
  - Desenvolvido usando Python e JavaScript.
- DataBase server:
  - A DB server armazena, recupera e disponibiliza informações/dados que são geridos pelo servidor para o funcionamento da aplicação.
  - Desenvolvido em MySQL.

## **5.3 Tecnologias e ferramentas utilizadas**

### Lista de ferramentas usadas até ao momento:

- Django: O projeto vai ser desenvolvido com o auxílio do Django, uma framework de alto nível que usa Python como linguagem de programação. Optou-se pela utilização do django dado o conhecimento obtido na UC Programação Web.
- Figma: Usado para mockups e UI's.
- DrawIO: Ferramenta bastante versátil na realização de diagramas, foi usada para fazer Casos de Uso, Modelo E-R e modelo de classes e arquitetura.
- Chrome: Browser preferenciado no desenvolvimento do projeto.
- Office: Microsoft Word para a realização do relatório, Excel para o desenvolvimento das listas de requisitos e processos (figura 2; figura 4) e PowerPoint para a apresentação do projeto.

- DBeaver: Ferramenta de software usada na criação e gestão da base de dados.
- Docker Desktop: Software utilizado na criação do nosso container para ligação com a Base de Dados.
- VSCode: IDE escolhido para o desenvolvimento do projeto.
- ChatGPT: Ferramenta única e imprescindível na realização do projeto, como no esclarecimento de dúvidas, definição de conceitos, bem como auxílio ao desenvolvimento

## 5.4 Implementação

### Recursos necessários ao desenvolvimento:

Aqui indicam-se alguns requisitos computacionais e não computacionais para tornar possível a realização do projeto.

Computador com capacidade para correr:

### Django & VSCode:

- Sistema Operacional Windows 10
- CPU: i5
- GPU: 6GB ou mais GRAPHICS
- RAM: 8GB
- Armazenamento: 3GB
- DirectX: 11

### MySQL:

- Versão mais recente (5.0.x) do [Java™ 2 SDK](#)
- Qualquer sistema operacional Windows de 32 bits suportado pela Microsoft
- Suporte a protocolo TCP/IP
- WinZip ou outra ferramenta Windows que possa ler arquivos .ZIP, para descompactar o arquivo de distribuição.
- Espaço suficiente na unidade de disco para desempacotar, instalar, e criar os bancos de dados de acordo com seus requisitos.

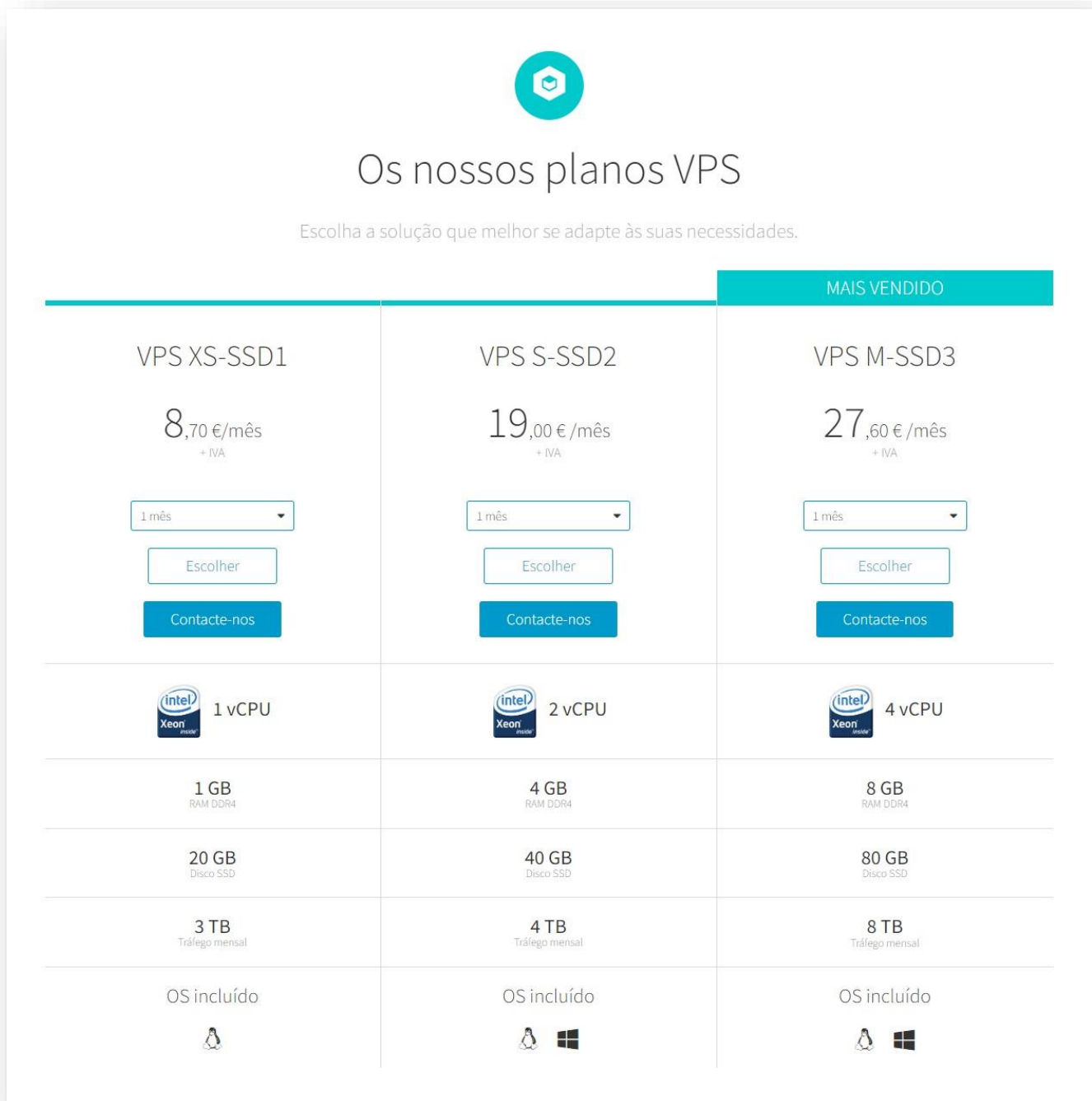
### Infraestrutura:

- Domínio.
- DNS (record que aponta o domínio para o IP do servidor)
- Servidor Linux com IP público
- Instalar Python no servidor (através de SSH)
- Enviar o site para o servidor (através de SCP)
- Colocar o site a correr com o comando: Python exemplo.py
- Instalar o MySQL no server

- Criar tabelas BD que depois são metidas no programa Python. Como a BD e o programa Python estão no mesmo server, a ligação é feita ao endereço localhost

### Servidor:

Usando os serviços da amen ([www.amen.pt](http://www.amen.pt)), temos algumas opções com os seus requisitos mínimos:



Os nossos planos VPS

Escolha a solução que melhor se adapte às suas necessidades.









		MAIS VENDIDO
VPS XS-SSD1	VPS S-SSD2	VPS M-SSD3
8,70 €/mês + IVA	19,00 €/mês + IVA	27,60 €/mês + IVA
1 mês	1 mês	1 mês
Escolher	Escolher	Escolher
Contacte-nos	Contacte-nos	Contacte-nos
 1 vCPU	 2 vCPU	 4 vCPU
1 GB RAM DDR4	4 GB RAM DDR4	8 GB RAM DDR4
20 GB Disco SSD	40 GB Disco SSD	80 GB Disco SSD
3 TB Tráfego mensal	4 TB Tráfego mensal	8 TB Tráfego mensal
OS incluído 	OS incluído  	OS incluído  

Figura 23 - Servidor

## 5.5 Abrangência

Para o desenvolvimento da solução, foram e vão ser aplicados conhecimentos de áreas científicas variadas que foram estudados nas UC's do curso de LEI, passo a indicar a sua aplicação.

### Engenharia e requisitos de testes:

Nesta UC trabalhamos com gestão de sistemas de informação, engenharia de requisitos, teste de software e aprendemos os conceitos para a gestão de um projeto com a utilização direta de várias ferramentas indicadas a cada um destes processos. Disto isto, é essencial na realização deste projeto, desde o seu planeamento até à sua conclusão.

Lista de conceitos/ferramentas usadas:

- Fichas de levantamento de requisitos
- Fichas de detalhe de requisito
- Diagramas de use case
- Todo o tipo de planeamento como identificação de funcionalidades, diferenciação de módulos e processos e escolha de requisitos prioritários, planeamento do flow de trabalho.

### Programação-Web:

Nesta UC aprendemos o funcionamento da web, suas tecnologias, paradigmas e padrões de desenho e implementação na cloud. Trabalhamos com a framework Django em conjunto com a utilização de HTML, CSS, JavaScript e Python.

Gostaria de mencionar que a escolha do Django para desenvolvimento deste projeto foi fortemente influenciada pela sua utilização nesta UC.

Lista de conceitos/ferramentas usadas:

- Django
- HTML
- CSS
- JavaScript
- Python
- Cloud

É preciso deixar claro que a UC Inteligência Artificial foi importante na escolha da linguagem, o Python.



### Base de Dados:

Nesta UC ficamos a entender o que é e como funciona uma base de dados e a sua importância em qualquer tipo de empresa. Aprendemos as bases de SQL – Structured Query Language, a estruturar uma base de dados e a desenvolver e usar modelos de Entidade-Relação.

Lista de conceitos/ferramentas usadas:

- Base de dados
- Modelos E-R
- MySQL

### Engenharia de software:

Nesta cadeira trabalhamos o desenho e modelação de software bem como o seu ciclo de vida, planeamento e gestão do desenvolvimento de projetos de software, verificação e validação de testes e manutenção do sistema de software.

Lista de conceitos/ferramentas usadas:

- Diagramas de arquitetura e descrição dos seus componentes
- Modelo de classes

O conhecimento obtido na UC Linguagens de Programação II onde se trabalha com programação orientada a objetos em Java também ajudou na realização dos modelos de classes, seus atributos e funções.

### Interação-Humano-Máquina:

Nesta UC aprendemos a relação das pessoas com os computadores, o que é a usabilidade e como tirar partido dela, casos de uso, modelos de navegação, desenho para web e mobile centrado às necessidades do utilizadores entre outros.

Lista de conceitos/ferramentas usadas:

- UI/UX
- Mockups
- Diagramas de use case

### Arquitetura de computadores, Sistemas Operativos

Estas UC auxiliaram a realização do capítulo 5.4.

Lista de conceitos/ferramentas usadas:

- Componentes computacionais, Armazenamento e Servidores

## 6. Método e Planeamento

### Metodologia de Desenvolvimento

Neste capítulo, descrevo a metodologia e o planeamento que segui no desenvolvimento do projeto, construída utilizando a framework Django. Além disso, realizo uma análise crítica do cumprimento do calendário proposto nos relatórios anteriores, baseando-me nas avaliações de progresso efetuadas ao longo do ano. Esta secção oferece uma visão geral do processo de trabalho e ajuda a avaliar a eficácia do planeamento inicial.

Para a implementação da aplicação web, optei por adotar uma abordagem de desenvolvimento baseada em agrupamento de tarefas, onde, em cada entrega, avaliava o trabalho pretendido e dividir os temas, tópicos e tarefas em grupos atribuindo-lhes um deadline. De início, nas primeiras fases do projeto, foi uma ótima estratégia pois as tarefas eram bem definidas com temas bem caracterizados (Identificação do Problema, Benchmarking, proposta de solução, realização de diagramas, identificação de requisitos, entre outros,).

Após o início efetivo do desenvolvimento, a abordagem de agrupamento de tarefas começou a apresentar desafios significativos. Isto deveu-se principalmente à mudança na natureza das tarefas à medida que avancei para as fases de implementação. Enquanto nas fases iniciais do projeto as tarefas eram bem definidas e os temas eram claramente caracterizados, a implementação revelou-se mais complexa do que o previsto.

### Dificuldades na Integração de Requisitos

Outro desafio que enfrentei ao tentar implementar a abordagem incremental foi a integração de requisitos. À medida que os diferentes grupos de tarefas eram concluídos, surgia a necessidade de integrar outros componentes de forma eficaz. Isso provou ser mais complicado do que o esperado, uma vez que algumas partes do sistema dependiam de outras para funcionar corretamente.

Por exemplo, a implementação da funcionalidade WishList dependia dos models Product e User estarem bem implementados e preparados para a associação com a WishList, o que não se verificou porque previamente tinha dado prioridade à implementação do Registo e do Login sem ter em consideração a futura implementação da WishList. Este contratempo obrigou-me a recuar no trabalho e reescrever todos os models envolvidos neste processo. Essas dependências tornaram a implementação de novas funcionalidades um processo desafiante, uma vez que alterações num componente afetavam diretamente outros componentes.

### Ajustes na Metodologia

Perante estes desafios, tive a necessidade de ajustar metodologia de desenvolvimento. Em vez de me sujeitar à abordagem de agrupamento de tarefas, adotei uma abordagem mais adaptável. Incorporei a ideia de agrupar menos requisitos em cada grupo, mas que estivessem interligados e dependentes um dos outros, mesmo que

não estivessem presentes no mesmo grupo (fig4), depois organizei estes grupos de forma sequencial tendo em conta a sua prioridade. Esta mudança permitiu-me lidar melhor com a complexidade crescente do projeto e facilitou a integração contínua de novas funcionalidades.

### **Revisões Periódicas**

Em conjunto com o Orientador de projeto, foram realizadas reuniões de revisão quase todas as semanas, mas sempre mais recorrentes perto das datas de entrega, nas quais analisávamos o progresso até o momento, discutíamos os desafios encontrados e planeávamos as próximas etapas. Estas reuniões foram fundamentais para manter o projeto alinhado com os nossos objetivos e garantir que estávamos no caminho certo.

### **Análise Crítica**

Ao longo do ano, fizemos avaliações periódicas do progresso do projeto. Notou-se que, em alguns casos, houve desvios em relação ao calendário proposto. Os principais fatores que contribuíram para esses desvios foram:

- Dificuldade de gestão / atribuição de tempo entre o TFC e projetos de outras unidades curriculares
- Complexidade não prevista na integração de alguns sistemas, como a WishList, inserção de imagens na BD
- Algumas dificuldades ao longo do desenvolvimento, uma vez que foi necessário visitar e reescrever partes do código à medida que avançava para a implementação de novas funcionalidades. Devido à natureza iterativa do projeto e dependências entre funcionalidades.

## 7.Resultados

### Plano de testes e validação:

#### 1.Testes de integração:

- Verificar se o sistema é capaz de realizar o registo e o login de users corretamente.
  - ✓ Teste executado com sucesso em LocalHost, valida os requisitos da temática 2.1(fig5)
  - ✓ Podemos verificar na imagem em baixo dois users do tipo admin e regular instanciados na nossa BD

id password	last_login	is_superuser username first_name last_name email	is_staff is_active date_joined
1 pbkdf2_sha256\$39000\$0\$P52u3D0M0b3Fakl1mNj56dP3z71xJm2room3IzDjbJFQvRVuedbtN3Gola00=	2023-09-05 21:11:03.146066	1 admin User Test admin@example.com	1 1 2023-09-04 19:06:40.567665
2 pbkdf2_sha256\$39000\$0\$Te3TRXVL0ZXAgr6J64HbeSQq9F9TQcU11z1q1FKAU/HpHNDsZ3VCykpvtzVozb10=	2023-09-05 22:24:45.875628	0 user1 User Test user1@test.pt	0 1 2023-09-04 19:06:40.758061

Figura 24 – Registered users

- Verificar se o sistema está a armazenar as informações dos models na BD corretamente.
  - ✓ Teste executado com sucesso em LocalHost, valida os requisitos 1.1.1, 1.2.4, 2.1.5 e 3 (fig5)

Grid	123 id	ABC image
1	1	products/aparador1.png
2	2	products/aparador2.png
3	3	products/aparador3.png
4	4	products/aparador4.png
5	5	products/banco1.png
6	6	products/banco2.png
7	7	products/banco3.png
8	8	products/banco4.png
9	9	products/bancoilha1.png
10	10	products/bancoilha2.png
11	11	products/bancoilha3.png
12	12	products/bancoilha4.png
13	13	products/bancoilha5.png
14	14	products/beliche1.png
15	15	products/beliche2.png
16	16	products/beliche3.png
17	17	products/berco1.png
18	18	products/berco2.png
19	19	products/berco3.png

Figura 25 – ImageBD

Grid	123 id	123 product_id	123 image_id
1	25	1	25
2	26	1	26
3	27	1	27
4	28	1	28
5	29	1	29
6	67	2	67
7	68	2	68
8	69	2	69
9	70	2	70
10	71	2	71
11	30	3	30
12	31	3	31
13	32	3	32
14	33	3	33
15	34	3	34
16	9	4	9
17	10	4	10
18	11	4	11
19	12	4	12
20	13	4	13

Figura 26 – ImageToProductAssociation

app_product   Enter a SQL expression to filter results (use Ctrl+Space)						
	id	name	description	price	category_id	unlisted
1	1	Chair	Black lacquered iron chair with a wooden seat.	0	4	0
2	2	Outdoor Sofa	Iron sofa specially treated for outdoor use. Painted in green	0	4	1
3	3	Rustic Armchair	Multipurpose rustic-style armchair painted in brown	0	4	0
4	4	Island Bench	Custom-made individual benches for your island, with a rotating system and height adjustment.	0	4	0
5	5	5-Seater Bench	Simple bench for indoor or outdoor use with 5 seats, outdoor treatment, and painted in the color of your choice	0	4	0
6	6	Sideboard	Simple sideboard with two iron doors and a wooden top measuring 1200x600x200mm	0	1	0
7	7	Wall Shelf	Large wall shelf with various levels at different heights and space for a television in the center. All dimensions tailored to your needs.	0	1	0
8	8	Hanging Shelves	Hanging shelves with an iron base and wooden shelves. Custom-made and painted to your liking	[NULL]	1	0
9	9	5-Level Sideboard	Sideboard as shown in the photo, 5 levels all made of iron and sheet metal.	0	1	0
10	10	Dining Table	Dining table with an iron base, wooden or iron top, and painted in the color of your choice.	[NULL]	3	0
11	11	TV Stand	Small iron TV stand with a wooden top for your television. 1000x40x40	0	3	0
12	12	Nightstand	100% iron nightstand measuring 600x300x250mm	0	3	0
13	13	Garden Table	Outdoor table with the necessary treatments, painted in dark green, measuring 2000x1100x800mm	0	3	0
14	14	Suite Bed	Double bed for a suite with dimensions of 2100x2000x60mm	0	2	0
15	15	Double Bed	Standard double bed measuring 1800x1600x60mm	0	2	0
16	16	Bunk Bed	Bunk bed with a ladder measuring 1100x1800x1900mm	0	2	0
17	17	Single Bed	Iron single bed measuring 1100x1800x600mm	0	2	0
18	18	Crib	Crib with safety guards measuring 900x1000x1200mm	0	2	0
19	19	Railings	Exterior railings with treatment, design, dimensions, and color of your choice. The photos are reference examples already made.	[NULL]	5	0
20	20	Exterior Gates	Exterior gate with a motor, treatment, design, dimensions, and color of your choice. The photos are reference examples already made.	[NULL]	5	0
21	21	Man Door	Exterior man door with treatment, design, dimensions, and color of your choice. The photos are reference examples already made.	[NULL]	5	0
22	22	Staircases	Interior staircase made of iron beam and sheet metal with design, dimensions, and color of your choice. The photos are reference examples already made.	[NULL]	5	0
23	23	Staircase Guards/Railings	Guards/railings for interior staircases or balconies with design, dimensions, and color of your choice. The photos are reference examples already made.	[NULL]	5	0
24	24	Coffee Table	Small iron coffee table for the center of the living room	0	3	0
25	25	Outdoor Cabinet	Outdoor cabinet with treatment	0	1	0
26	26	Gas Cylinder Door	Protection door for gas cylinder	0	5	0

Figura 27 –ProductBD

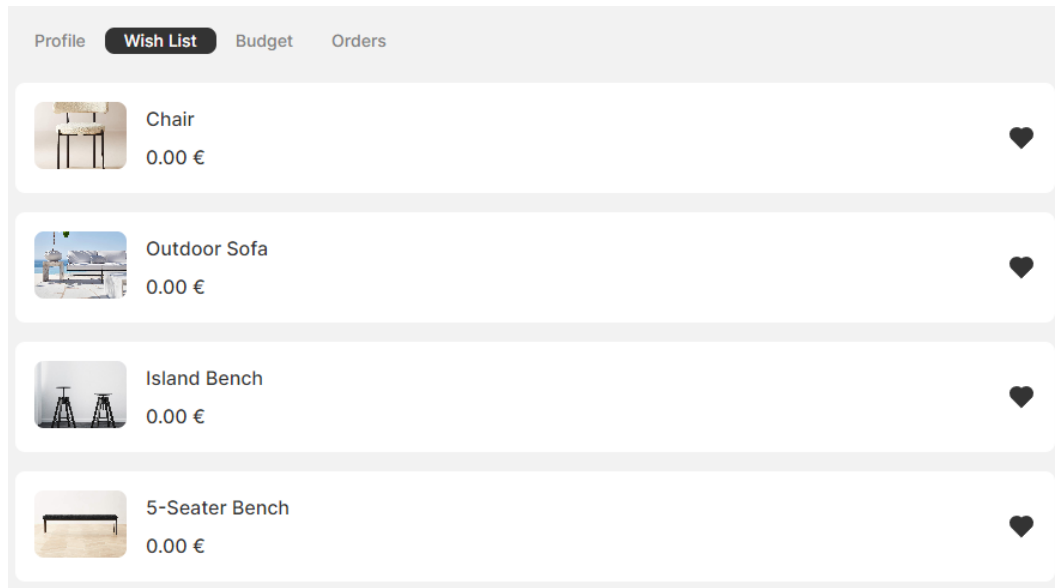
app_wish   Enter a SQL expression to filter results (use Ctrl+Space)				
	id	added_date	product_id	user_id
1	1	2023-09-05 21:18:29.929826	1	2
2	2	2023-09-05 21:18:30.932090	2	2
3	3	2023-09-06 13:29:44.315650	4	2

Figura 28 – WishListBD

app_category   Enter a SQL expression to filter results (use Ctrl+Space)			
	id	name	image
1	1	Storage	categories/storage.png
2	2	Beds	categories/beds.png
3	3	Tables	categories/tables.png
4	4	Seating	categories/seating.png
5	5	Construction	categories/construction.png

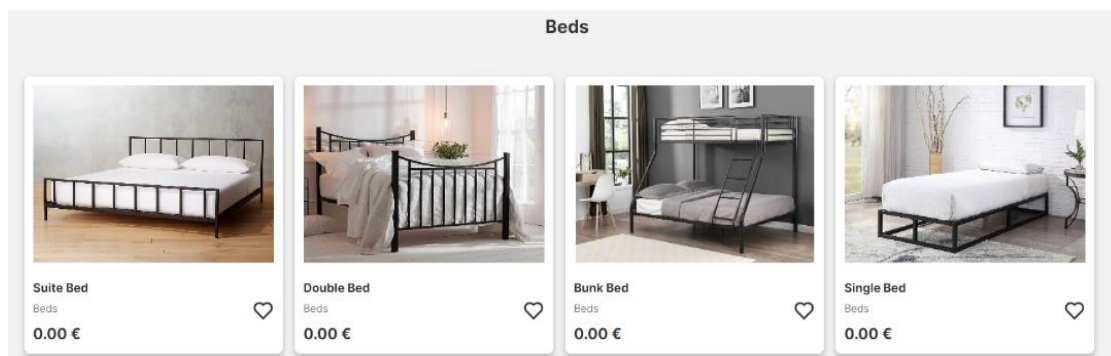
Figura 29– CategoryBD

- Verificar se a app permite ao user adicionar produtos aos favoritos.
  - ✓ Teste executado com sucesso em LocalHost, valida os requisitos da temática 3 (fig5)



**Figura 30**– Products on User WishList

- Verificar se a app está a mostrar bem as listas de categorias e produtos nos sítios esperados.
  - ✓ Teste executado com sucesso em LocalHost, valida os requisitos 1.1.1 (fig5)



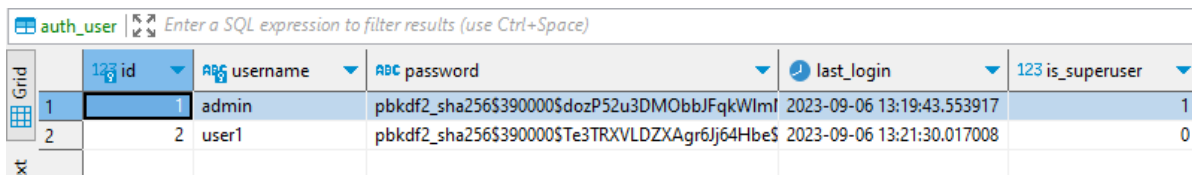
**Figura 31** – Category page

## 2. Testes de desempenho:

- Executar testes de carga e tráfego para ver a capacidade de resposta.
  - × Teste não executado

## 3. Testes de segurança:

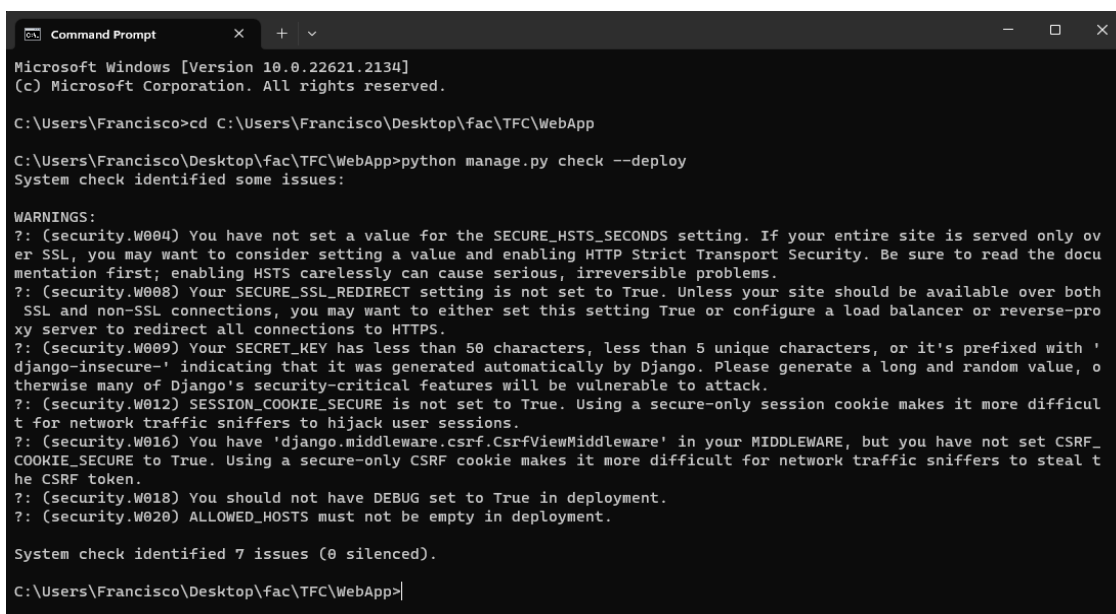
- Verificar se o sistema está a armazenar as passwords dos user na BD com hashes.
  - ✓ Teste executado com sucesso em LocalHost, valida os requisitos 2.1.5 (fig5)



Grid	123 id	ABG username	ABC password	last_login	123 is_superuser
1	1	admin	pbkdf2_sha256\$390000\$dozP52u3DMObbJFqkWImI	2023-09-06 13:19:43.553917	1
2	2	user1	pbkdf2_sha256\$390000\$Te3TRXVLDZXAgr6Jj64Hbe\$	2023-09-06 13:21:30.017008	0

Figura 32 – Password Hashes

- Verificar se o sistema está protegido contra-ataques de injeção SQL.
  - × Os querysets do Django são protegidos contra a injeção de SQL, pois suas queries são construídas usando a query parameterization. O código SQL de uma query é definido separadamente dos parâmetros da query.
- Verificar se o sistema está protegido contra-ataques de cross-site-scripting (XSS).
  - × O uso de templates do Django protege-nos contra a maioria dos ataques de XSS. No entanto, é importante entender as proteções que ele oferece e suas limitações.
  - × Os modelos do Django escapam caracteres específicos que são particularmente perigosos para o HTML. Embora isto proteja os users contra a maioria das entradas maliciosas, não é totalmente infalível. Por exemplo, ele não protegerá o seguinte: `<style class={{ var }}>...</style>`



```

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Francisco>cd C:\Users\Francisco\Desktop\fac\TFC\WebApp

C:\Users\Francisco\Desktop\fac\TFC\WebApp>python manage.py check --deploy
System check identified some issues:

WARNINGS:
?: (security.W004) You have not set a value for the SECURE_HSTS_SECONDS setting. If your entire site is served only over SSL, you may want to consider setting a value and enabling HTTP Strict Transport Security. Be sure to read the documentation first; enabling HSTS carelessly can cause serious, irreversible problems.
?: (security.W008) Your SECURE_SSL_REDIRECT setting is not set to True. Unless your site should be available over both SSL and non-SSL connections, you may want to either set this setting True or configure a load balancer or reverse-proxy server to redirect all connections to HTTPS.
?: (security.W009) Your SECRET_KEY has less than 50 characters, less than 5 unique characters, or it's prefixed with 'django-insecure-' indicating that it was generated automatically by Django. Please generate a long and random value, otherwise many of Django's security-critical features will be vulnerable to attack.
?: (security.W012) SESSION_COOKIE_SECURE is not set to True. Using a secure-only session cookie makes it more difficult for network traffic sniffers to hijack user sessions.
?: (security.W016) You have 'django.middleware.csrf.CsrfViewMiddleware' in your MIDDLEWARE, but you have not set CSRF_COOKIE_SECURE to True. Using a secure-only CSRF cookie makes it more difficult for network traffic sniffers to steal the CSRF token.
?: (security.W018) You should not have DEBUG set to True in deployment.
?: (security.W020) ALLOWED_HOSTS must not be empty in deployment.

System check identified 7 issues (0 silenced).

C:\Users\Francisco\Desktop\fac\TFC\WebApp>
  
```

Figura 33 – Erros/Falhas de segurança ao fazer deploy

#### 4. Testes de usabilidade:

- Realizar testes de usabilidade à app recorrendo a users reais:
  - ✓ Teste executado com sucesso em LocalHost.
  - ✓ Devido ao projeto correr em local host torna-se mais complicado a sua divulgação para execução de testes de usabilidade, no entanto os mesmos foram executados e o seu feedback tido em consideração.
- Analisar o feedback e fazer alterações consoante.
  - ✓ Foram alterados muitos componentes de UI/UX devido a feedback exterior como pro exemplo a página de perfil do user(fig17) e alguns dos formulários.

#### 5. Testes de compatibilidade:

- Testar a compatibilidade da app com outros browsers.
  - ✓ Testado no Chrome(Browser usado no desenvolvimento), FireFox e Microsoft Edge com sucesso.

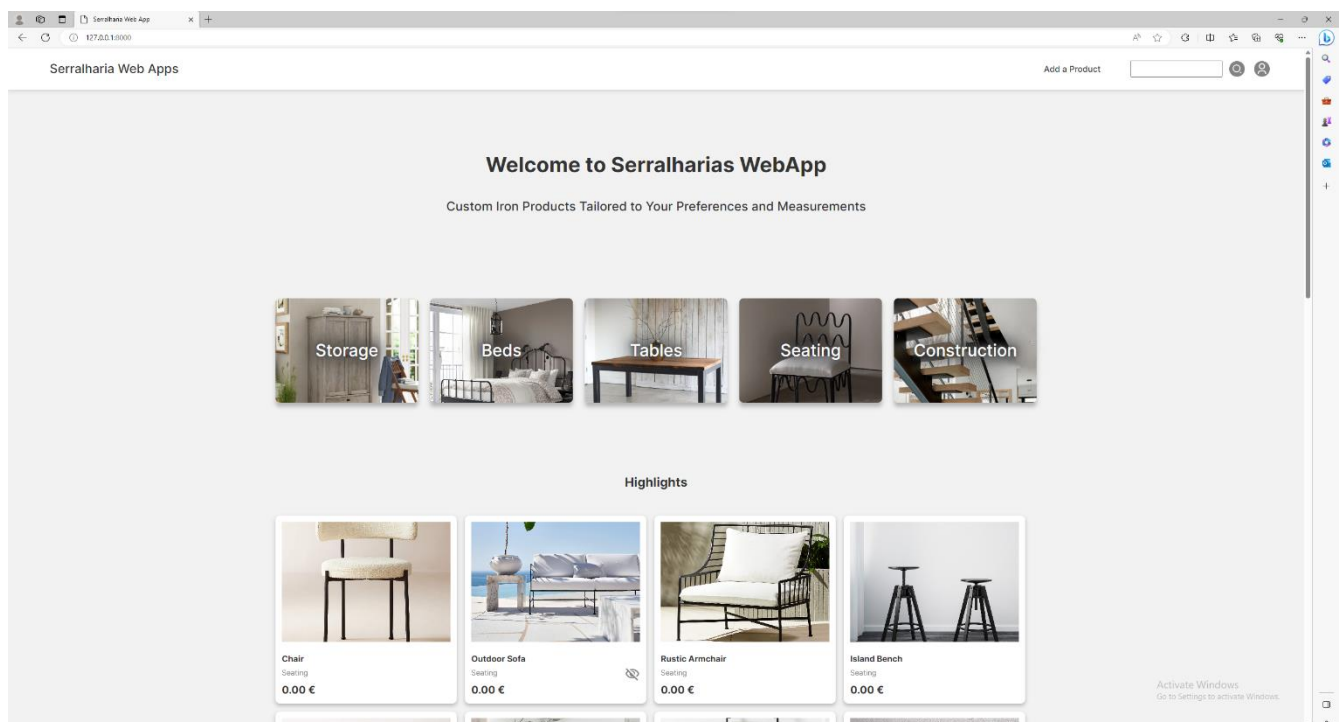
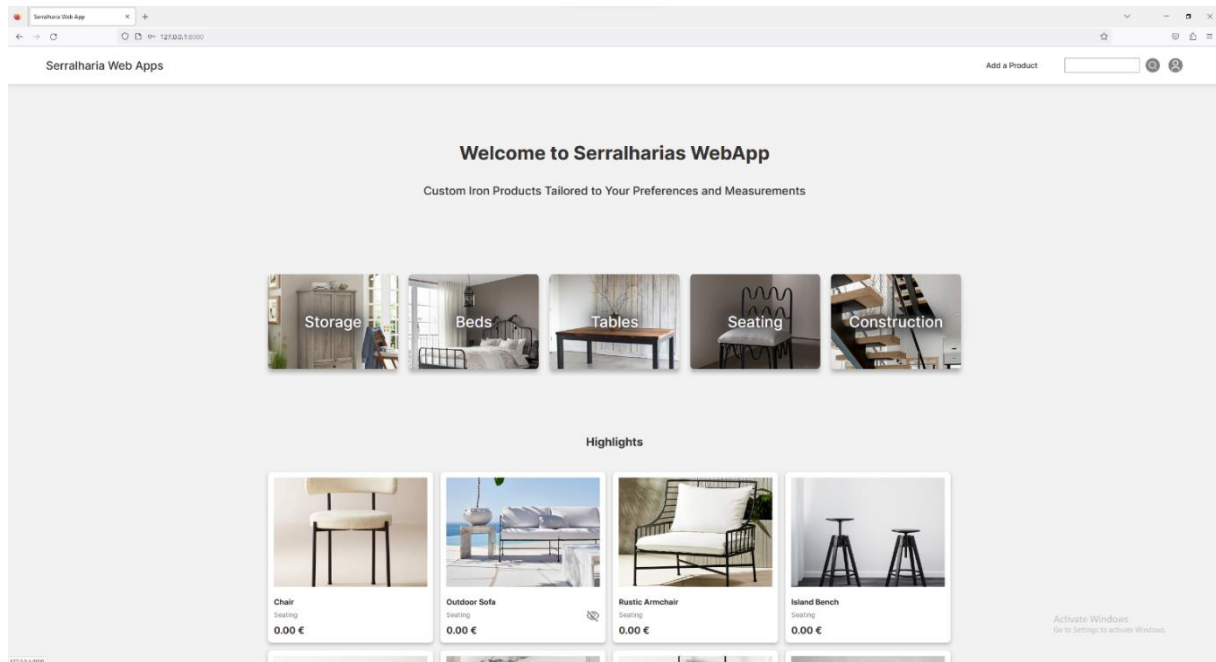


Figura 34 – Teste de Compatibilidade no Microsoft Edge





**Figura 35**– Teste de Compatibilidade no FireFox

#### 6. Testes de Regressão:

- Testar se as alterações recentes no código não causam problemas em outras áreas não relacionadas da app.
  - ✓ Devido a um projeto pouco modular em que há pouca separação entre componentes dentro das views.py, foi testado que se partirmos alguma das views presentes em views.py, o projeto fica corrompido.

#### 7. Teses de aceitação:

- Executar testes de aceitação para garantir que a app passa em todos os requisitos especificados em fases anteriores.
  - ✓ Como visto em capítulos anteriores (Cap4.1) posso afirmar que todos os requisitos foram implementados com sucesso exceto o requisito 1.1.6 (fig5).

Temos de ter atenção que o projeto está a ser desenvolvido em localhost e que o ambiente de produção pode ser diferente, dito isto tem de haver um certo cuidado ao testar a app somente em localhost, pois existem testes que nem sequer podem ser executados.

É também importante frisar que todos os dados dos nossos modelos presentes na BD foram gerados a partir dos formulários incluídos na app, ou seja, podem não ser dados 100% fictícios, mas são dados exemplo criados por mim ou pelos usuários sujeitos aos testes de usabilidade. Estes foram os dados que serviram de suporte aos testes á aplicação executados neste capítulo.

## **8. Conclusão e trabalhos futuros**

Em baixo, resumi o processo de desenvolvimento da aplicação desde as fases mais iniciais até à sua conclusão.

### **Identificação do Problema**

O primeiro passo foi identificar o problema, que consistia em processos demorados e confusos na interação com clientes e empresa. A necessidade de uma solução tecnológica que simplificasse esses processos foi clara, justificando a criação da aplicação.

### **Viabilidade e Pertinência**

Foi realizada uma análise de viabilidade para avaliar a pertinência de desenvolver a aplicação. Concluiu-se que a aplicação era viável e pertinente no âmbito em que se insere, pois melhora a eficiência operacional, otimiza a gestão de clientes e aumenta o potencial de crescimento do negócio.

### **Benchmarking**

Para garantir que a aplicação atendesse aos padrões da indústria e às necessidades dos utilizadores, foi realizado um benchmarking para avaliar soluções similares no mercado. Que permitiu identificar melhores práticas e funcionalidades-chave a serem implementados na aplicação.

### **Levantamento de Requisitos**

Em seguida, procedemos ao levantamento de requisitos detalhados, onde foram identificadas as funcionalidades essenciais da aplicação. Isso incluiu a definição de requisitos funcionais e não funcionais, como autenticação, segurança e produção.

### **Criação de Modelos e Diagramas**

Com base nos requisitos levantados, criamos modelos e diagramas para representar a estrutura da aplicação. Isto incluiu diagramas de casos de uso, diagramas de classe, diagramas de E-R entre outros, para definir a arquitetura e a organização dos nossos dados.

### **Proposta de solução**

Aqui foi apresentada a proposta da solução idealizada, mencionado a sua arquitetura, variadas tecnologias necessárias ao desenvolvimento bem como os recursos necessários á implementação, foi também falado no papel e impacto que várias unidades curriculares tiveram na realização deste projeto.

### **Criação da Base de Dados e Modelos**

Uma etapa crítica foi a criação da base de dados e a definição dos modelos de dados necessários para armazenar informações sobre produtos, clientes, pedidos e faturas. Foi essencial criar modelos que permitissem a população da base de dados com dados de exemplo, mas aplicáveis, facilitando o teste e o desenvolvimento.

### **Configuração do Django**

Para garantir que a aplicação funcionasse corretamente, configurei o Django para estabelecer a ligação com a base de dados. Isso incluiu definir as settings de acesso à base de dados entre outros.

### **Criação do UI**

A interface do utilizador (UI) desempenha um papel fundamental na usabilidade da aplicação. Portanto, tentei criar os elementos de UI de forma a garantir que a aplicação fosse intuitiva e funcional para os utilizadores. Na fase de desenvolvimento, tentei manter-me fiel aos mockups realizados anteriormente pois os mesmos tiveram um bom feedback.

### **Programação da Lógica de Negócio**

Por fim, implementamos toda a lógica de negócio da aplicação. Isto envolveu a programação de funcionalidades como criação e autenticação de users, inserção de produtos, solicitação de orçamentos, gestão de produtos e users, entre outras.

Ao seguir essas etapas, consegui desenvolver uma aplicação que aborda o problema identificado, fornecendo uma solução tecnológica abrangente para melhorar a eficiência operacional e impulsionar o crescimento do negócio.

### **Trabalho Futuro:**

Para melhorar a aplicação e torná-la mais eficiente e versátil, podemos dar continuidade ao seu desenvolvimento ao realizar as seguintes etapas:

### **Gestão de Estado dos Pedidos:**

Acabar de Implementar o sistema de gestão de estado de pedidos para que os utilizadores possam acompanhar o progresso dos seus pedidos. (A BD e os Models já estão preparados para esta adição)

#### **Criação de Invoices:**

Desenvolver a capacidade de gerar faturas automaticamente para os pedidos concluídos e enviar por e-mail.

#### **Criação de API e Separação Frontend/Backend:**

Separar a aplicação em frontend e backend, criando uma API para melhor modularização e compatibilidade com diferentes tipos de interfaces de usuário, como aplicações web, desktop ou móveis.

#### **Implementação de um CRM:**

Integrar um sistema de Gestão de Relacionamento com o Cliente (CRM) para melhorar o atendimento ao cliente e acompanhar as interações com os utilizadores.

#### **Validação de Email:**

Adicionar validação de email para permitir futuras comunicações com os utilizadores, como notificações de alterações no estado de um pedido.

#### **Testes de Usabilidade:**

Realizar mais testes de usabilidade para identificar e resolver problemas de interface do utilizador e melhorar a experiência do utilizador.

#### **Testes Unitários:**

Implementar testes unitários para os modelos e a lógica de negócios da aplicação, garantindo que ela seja robusta e livre de erros.

#### **Colocação em Produção:**

Preparar a aplicação web para produção, alugando um servidor, adquirindo um domínio e configurando todas as etapas necessárias para torná-la acessível ao público.

Esses planos futuros visam melhorar a funcionalidade, a segurança e a usabilidade da aplicação, tornando-a uma ferramenta mais completa, eficaz e funcional para os utilizadores e para a gestão de negócios da serralharia.

## **Anexos e ficheiros**

Todos os ficheiros e diagramas estão presentes neste relatório, por esta mesma razão optei por não os incluir neste capítulo.

## **Bibliografia**

- [DEISI23] DEISI, Regulamento de Trabalho Final de Curso, Set. 2023.
- [ULHT23] Universidade Lusófona de Humanidades e Tecnologia, [www.ulusofona.pt](http://www.ulusofona.pt), acedido em Out. 2023.

**Glossário**

LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
AT	Autoridade Tributária
PHC	Software de Gestão
CRM	Customer relationship management
BD	Base de Dados
DB	DataBase
DAC	Digital Analog Converter
UI	User Interface
UX	User Experiencie
E-R	Entidade-Relação
PK	Primary Key
FK	Foreign Key
MVC	Model-View-Controller
UC	Unidade Curricular
TCP	Transmission Control Protocol
IP	Internet Protocol
DNS	Domain Name System
SQL	Structured Query Language
XSS	Cross Site Scripting