

[Título do Projeto]

*Imagem ou personalização gráfica do Relatório*

[Identificação dos formandos] [Turma]

[Identificação da(s) UFCD] [Formador(es)]

[Data]

# Índice

[Índice 1](#_Toc190309870)

[1 Introdução 3](#_Toc190309871)

[2 Planeamento 4](#_Toc190309872)

[2.1 Metodologia de Desenvolvimento 4](#_Toc190309873)

[2.2 Cronograma 5](#_Toc190309874)

[**1.** **Requisitos** 12](#_Toc190309884)

[1.1. Requisitos Funcionais 12](#_Toc190309885)

[1.2. Requisitos não Funcionais 12](#_Toc190309886)

[1.3. Priorização de Requisitos 13](#_Toc190309887)

[**2.** **Desenho do Sistema e Arquitetura** 13](#_Toc190309888)

[2.1. Desenho do Sistema 13](#_Toc190309889)

[2.2. Arquitetura e Tecnologia 13](#_Toc190309890)

[2.3. Diagramas e Modelos 13](#_Toc190309891)

[**3.** **Implementação** 13](#_Toc190309892)

[3.1. Ambiente de Desenvolvimento 14](#_Toc190309893)

[3.2. Desenvolvimento 14](#_Toc190309894)

[3.3. Controle de Versão 14](#_Toc190309895)

[**4.** **Testes** 14](#_Toc190309896)

[**5.** **Evolução** 14](#_Toc190309897)

[**6.** **Conclusões** 14](#_Toc190309898)

[**7.** **Fontes** 14](#_Toc190309899)

[**Anexos** 15](#_Toc190309900)

# Introdução

Na introdução devem constar os objetivos do trabalho e a sua motivação. A introdução deverá descrever, sucintamente, o modo como o relatório está organizado nas suas diversas partes e a articulação entre elas segundo uma ideia condutora. Descreva, de forma sucinta, o problema em estudo e enuncie os principais métodos que são utilizados no relatório. Seja especialmente cuidadoso na utilização de termos técnicos ou referência a palavras-chave relativas à temática abordada.

# Planeamento

O planeamento é uma das etapas mais importantes no desenvolvimento de um projeto de software, pois define a estrutura base e orienta todas as fases subsequentes. Um bom planeamento permite antecipar riscos, otimizar recursos e garantir que os prazos sejam cumpridos, assegurando um desenvolvimento eficiente e alinhado com os objetivos do projeto.

Neste projeto, o planeamento envolveu uma abordagem abrangente, estruturada com base na definição da base de dados, na arquitetura do *backend* e *frontend*, na validação de dados e segurança e na integração com sistemas externos.

Para garantir uma implementação robusta e eficiente das funcionalidades, realizámos um planeamento detalhado que contemplou:

* Definição e estruturação da base de dados para suportar todas as operações essenciais da aplicação, assegurando integridade e eficiência no armazenamento e recuperação de dados.
* Planeamento do *backend*, estabelecendo a forma como os dados seriam processados, armazenados e disponibilizados através de *APIs*.
* Desenvolvimento do *frontend*, onde definimos a melhor abordagem para estruturar a interface e proporcionar uma experiência de utilização fluida e intuitiva.
* Validação de dados em dois níveis:
  + *Frontend* – Implementação de mecanismos de validação de formulários para garantir que os dados inseridos pelos utilizadores cumpriam os requisitos definidos.
  + *Backend* – Reforço da segurança com validações adicionais no servidor, garantindo que os dados recebidos estavam corretos e protegidos contra tentativas de inserção indevida.
* Definição de prioridades e fluxo lógico de desenvolvimento, estruturando o trabalho por fases e garantindo que cada funcionalidade fosse desenvolvida de forma interdependente e otimizada.
* Planeamento da interligação com *APIs* externas e tecnologias complementares, permitindo enriquecer a aplicação com funcionalidades adicionais e assegurar uma integração fluida com outros serviços.

Este planeamento rigoroso permitiu-nos antecipar desafios, distribuir tarefas de forma eficiente e garantir que cada componente do sistema funcionasse de forma coesa, resultando numa aplicação bem estruturada, segura e escalável.

## Metodologia de Desenvolvimento

Para o desenvolvimento da aplicação, foi adotada a metodologia *Scrum*, uma *framework* ágil que permite a entrega contínua de funcionalidades através de ciclos iterativos chamados sprints.

Esta abordagem foi escolhida devido à sua flexibilidade e capacidade de adaptação a eventuais mudanças nos requisitos do projeto, garantindo um desenvolvimento estruturado e eficiente.

Enquanto grupo, seguimos uma abordagem colaborativa, onde cada membro assumiu responsabilidades específicas dentro de cada sprint, assegurando um fluxo de trabalho contínuo e bem organizado.

O desenvolvimento da aplicação foi dividido nas seguintes fases, garantindo a integração harmoniosa entre *frontend* e *backend*.

* **Planeamento e Estruturação da Base de Dados**

Para suportar toda a lógica da aplicação, iniciámos o desenvolvimento com a definição do modelo relacional, assegurando que todas as tabelas e relações estivessem corretamente estruturadas.

Utilizámos *migrations* para a criação da base de dados, permitindo um controlo eficaz da sua evolução ao longo do desenvolvimento.

* **Desenvolvimento do Backend**

O *backend* foi desenvolvido com Laravel, garantindo uma arquitetura robusta e modular. Durante esta fase, implementámos:

* + Controladores e modelos, permitindo a interação entre a base de dados e o frontend através de endpoints bem definidos.
  + Migrations, seeders e factories, assegurando a criação estruturada de tabelas e a inserção de dados essenciais para os testes e validações.
  + Camada de segurança e validação, garantindo que os dados recebidos fossem verificados corretamente antes de serem processados.
* **Desenvolvimento do Frontend**

A interface da aplicação foi construída com React + Inertia.js, utilizando Vite para otimizar a compilação e garantir uma experiência fluida e responsiva. O desenvolvimento do *frontend* seguiu as seguintes diretrizes:

* + Criação de um layout em componentes, garantindo a reutilização eficiente de elementos da interface.
  + Gestão de estados e interação com o backend, assegurando que os dados fossem apresentados e atualizados de forma dinâmica e eficiente.
  + Validação de formulários no frontend, reduzindo a possibilidade de erros antes de submeter os dados ao backend.
* **Implementação das Funcionalidades Principais**

Seguindo a metodologia *Scrum*, planeámos e implementámos cada funcionalidade de forma progressiva, garantindo que todas as partes do sistema estivessem interligadas corretamente. Durante este processo, realizámos revisões regulares para validar a usabilidade e eficiência da aplicação.

* **Testes e Otimizações**

Para garantir um sistema estável e funcional, adotámos uma abordagem iterativa de testes contínuos durante cada sprint. Foram realizadas:

* + Testes no backend, validando a integridade dos dados e a segurança das APIs.
  + Testes no frontend, assegurando que a experiência do utilizador fosse intuitiva e fluida.
  + Correção de erros e otimizações, refinando a aplicação antes da entrega final.

Através desta metodologia, conseguimos desenvolver um produto robusto e bem estruturado, garantindo que cada fase do desenvolvimento estivesse alinhada com os objetivos do projeto e as necessidades dos utilizadores.

## Cronograma

O cronograma do projeto foi estruturado em fases com prazos bem delineados, garantindo um fluxo contínuo de desenvolvimento e permitindo a entrega de um produto funcional e otimizado no prazo estipulado.

O grupo decidiu iniciar o planeamento e organização do desenvolvimento com antecedência, devido à elevada complexidade e ambição do projeto.

Essa abordagem permitiu uma melhor distribuição das tarefas, minimizando riscos e garantindo um desenvolvimento mais controlado e eficiente. Assim, o trabalho foi distribuído ao longo de 12 semanas, organizando os sprints da seguinte forma:

### Sprint 1 - Autenticação - Semana 1-2

A autenticação foi implementada utilizando a autenticação intrínseca do Laravel, garantindo um sistema seguro e eficiente.

O registo de utilizadores exige uma verificação por email, utilizando Laravel Mail para o envio de um *link* de confirmação.

Para garantir a persistência da sessão, utilizámos os mecanismos nativos do Laravel, assegurando que o utilizador permanece autenticado mesmo após um *refresh* da página.

Além disso, foi incorporada a funcionalidade de recuperação de password, permitindo aos utilizadores redefinirem as suas credenciais através de um email de redefinição.

A implementação do *logout* assegura que todas as sessões ativas do utilizador são encerradas de forma segura.

### Sprint 2 – Gestão de perfil do utilizador - Semana 3

Para garantir uma experiência de utilização intuitiva e eficiente, planeámos a navegação desta secção através de uma *sidebar*, permitindo ao utilizador alternar facilmente entre as diferentes áreas do seu perfil.

Dividimos as funcionalidades em categorias distintas para facilitar a navegação, proporcionando uma estrutura clara e organizada.

A abordagem ao desenvolvimento seguiu um fluxo lógico, iniciando-se com o design da interface e experiência do utilizador, assegurando que cada ação, como editar dados, visualizar encomendas ou gerir moradas, fosse acessível e intuitiva. Posteriormente, avançámos para a implementação técnica de cada funcionalidade, assegurando que o utilizador pudesse editar, visualizar e eliminar informação de forma eficiente.

Além disso, integrámos a aba de notificações, permitindo que os utilizadores recebessem alertas sobre alterações no estado das suas encomendas ou novas encomendas realizadas.

### Sprint 3 – Registo de Perfil de vendedores - Semana 4 e 5

O registo de vendedor foi estruturado em três fases distintas para facilitar o processo e garantir uma experiência fluida ao utilizador:

* **1ª fase:** Recolha dos dados pessoais do utilizador ou, caso aplicável, os dados da sua empresa, garantindo a validação necessária para prosseguir com a criação da conta de vendedor.
* **2ª fase:** Criação da primeira loja do vendedor, onde são introduzidas informações essenciais como nome, descrição, localização e detalhes de contacto.
* **3ª fase:** Possibilidade de adicionar os primeiros produtos à loja recém-criada, permitindo que o vendedor comece a operar de imediato na plataforma.

Esta abordagem estruturada garantiu que o fluxo de *onboarding* dos vendedores fosse intuitivo, evitando sobrecarga de informações e permitindo uma transição suave para a gestão da loja dentro da aplicação.

### Sprint 4 – Area de gestão do vendedor - Semana 6 até 11

A Área de Gestão do Vendedor foi planeada para fornecer uma visão completa dos negócios do utilizador na plataforma. Devido à sua complexidade e ao número de funcionalidades envolvidas, esta feature foi desenvolvida ao longo de várias semanas, sendo dividida em **sprints** para garantir um desenvolvimento organizado e eficiente. A abordagem seguiu uma lógica progressiva, assegurando que cada funcionalidade fosse implementada de forma sólida antes de avançar para a seguinte.

**Planeamento dos Sprints da Área de Gestão do Vendedor**

* **Sprint 1 (Semana 6-7): Perfil do Vendedor**

O primeiro passo no desenvolvimento desta funcionalidade foi garantir que o vendedor pudesse visualizar e editar as suas informações. Assim, começámos por implementar:

Exibição dos dados do vendedor, incluindo informações pessoais e, se aplicável, dados da empresa.

Edição das informações do vendedor, permitindo atualizar os seus dados facilmente.

* **Sprint 2 (Semana 7-8): Gestão de Lojas**

Após definir a estrutura do perfil do vendedor, avançámos para a funcionalidade de gestão de lojas, onde implementámos:

* + Listagem de lojas criadas pelo vendedor.
  + Possibilidade de criar uma loja (respeitando o limite de 3 lojas por vendedor).
  + Edição e remoção de lojas.
* **Sprint 3 (Semana 8-9): Gestão de Produtos**

Com a gestão das lojas estabelecida, passámos ao desenvolvimento das funcionalidades relacionadas com os produtos dentro de cada loja:

* + Visualização detalhada de cada loja específica.
  + Listagem dos produtos disponíveis em cada loja.
  + Adição de novos produtos com imagens, descrição, preço da unidade de produto.
  + Edição e remoção de produtos existentes.
  + Implementação de um sistema de pesquisa para facilitar a busca por produtos dentro da loja.
* **Sprint 4 (Semana 9-10): Gestão de Encomendas**

Após a implementação da gestão de produtos, o próximo foco foi a funcionalidade de encomendas. Para isso, desenvolvemos:

* Listagem de encomendas organizadas por loja ou exibição global de todas as encomendas do vendedor.
* Visualização dos detalhes das encomendas, incluindo cliente, produtos adquiridos e estado da encomenda.
* Possibilidade de atualizar o estado de uma encomenda (exemplo: "Em Processamento", "Enviado", "Concluído").
* Opção de cancelamento de encomendas diretamente pela interface do vendedor.
* **Sprint 5 (Semana 10-11): *Dashboard* Dinâmico**

Na última fase do desenvolvimento da Área de Gestão do Vendedor, implementámos um *dashboard* interativo para oferecer uma visão geral dos negócios do vendedor. As funcionalidades incluíram:

* Exibição de métricas importantes, como volume de vendas, número de encomendas e avaliações de clientes.
* Representação gráfica dos dados para facilitar a interpretação de tendências e desempenho da loja.

Esta funcionalidade foi desenvolvida **paralelamente** com as restantes áreas da aplicação, garantindo que a experiência do vendedor estivesse alinhada com o fluxo de navegação do cliente. Devido à complexidade desta feature, foi necessário dedicar um período de desenvolvimento mais extenso para garantir uma implementação robusta e escalável.

Este planeamento permitiu que a **Área de Gestão do Vendedor** fosse construída de forma progressiva, assegurando que cada componente estivesse completamente funcional antes de avançar para a seguinte, resultando numa interface intuitiva e eficiente para os utilizadores da plataforma.

### Sprint 5 - Implementação da Listagem de Produtos e Páginas Essenciais – Semana 7

Nesta fase, focámo-nos na construção das principais páginas da aplicação, assegurando que a navegação fosse intuitiva e funcional. As implementações incluíram:

* **Homepage:** Desenvolvida para apresentar produtos em destaque, lojas recomendadas e categorias populares, proporcionando uma primeira experiência envolvente ao utilizador.
* **Cards de Produtos:** Cada produto foi representado num card com imagem, nome, preço e vendedor, garantindo um design responsivo e apelativo.
* **Página de Lojas:** Criámos um layout dedicado onde cada loja apresenta a sua descrição, localização e lista de produtos disponíveis.
* **Página do Carrinho de Compras:** Desenvolvida para proporcionar um fluxo de compra claro e simplificado, permitindo que o utilizador visualize os produtos selecionados, modifique quantidades e avance para o checkout.
* **Navegação e Usabilidade:** Implementámos menus intuitivos e filtros de pesquisa para melhorar a experiência do utilizador e facilitar a descoberta de produtos e lojas.

### Sprint 6 - Desenvolvimento do Mapa Interativo – Semana 8

Para melhorar a experiência do utilizador na descoberta de lojas próximas, desenvolvemos um mapa interativo, utilizando geolocalização para fornecer informações em tempo real. As funcionalidades implementadas incluíram:

* **Geolocalização do Utilizador:** Permitiu identificar a localização do utilizador e exibir as lojas mais próximas.
* **Exibição de Lojas no Mapa:** Cada loja foi representada por um ícone interativo, permitindo que o utilizador aceda rapidamente à página da loja.
* **Integração na Página da Loja:** O mapa foi incorporado na página de cada loja, mostrando a sua localização exata e facilitando a navegação para o cliente.
* **Geocodificação de Moradas:** Garantimos que todas as moradas inseridas pelos vendedores fossem corretamente convertidas em coordenadas para serem exibidas no mapa.

### Sprint 7 - Implementação da Pesquisa – Semana 9

A pesquisa foi identificada como um elemento essencial para a navegabilidade da aplicação, permitindo que os utilizadores encontrassem rapidamente as lojas disponíveis na plataforma.

No entanto, devido a limitações de tempo e à priorização de outras funcionalidades críticas, neste sprint conseguimos apenas implementar a pesquisa por lojas.

A implementação focou-se nos seguintes aspetos:

* **Barra de Pesquisa Global**: Disponível em todas as páginas da aplicação, permitindo que o utilizador procurasse lojas pelo nome.
* **Apresentação dos Resultados**: Cada loja encontrada era apresentada num card de loja, contendo informações relevantes, como nome, localização e o card é um link direto para a página dessa loja.
* **Redirecionamento para a Página da Loja**: O utilizador podia clicar diretamente no card para ser levado à página da loja correspondente.

Inicialmente, a nossa intenção era expandir a pesquisa para incluir produtos e vendedores, oferecendo resultados diferenciados com opções interativas, como adicionar produtos ao carrinho diretamente da pesquisa e visualizar perfis de vendedores. No entanto, devido a constrangimentos de tempo e à complexidade adicional envolvida, não foi possível avançar com essa implementação nesta sprint.

A pesquisa foi, portanto, desenvolvida com um escopo mais restrito, ficando o aprimoramento desta funcionalidade planeado para futuras iterações do projeto

### Sprint 8 - Implementação da Gestão de Encomendas, Métodos de Pagamento e Envio de Faturas – Semana 10

Nesta fase, garantimos que o fluxo de compra e pagamento estivesse completo e integrado com a lógica da aplicação. As implementações incluíram:

* **Gestão de Encomendas**: Criámos um sistema onde os utilizadores podiam consultar o histórico de encomendas, visualizar detalhes e acompanhar o estado da entrega.
* **Cálculo de Portes de Envio**: Desenvolvemos um algoritmo que calcula o custo do envio com base no peso da encomenda e na distância do destinatário.
* **Escolha de Método de Pagamento**: Integramos diferentes opções de pagamento, incluindo cartão de crédito/débito e pagamentos digitais.
* **Geração de Faturas**: Implementámos um sistema que permite gerar faturas automaticamente para os clientes após a conclusão da compra.
* **Escolha de Método de Pagamento:** Integramos diferentes opções de pagamento, incluindo cartão de crédito/débito ou pagamento via PayPal.
* **Envio de Faturas:** Implementámos um sistema que permite enviar faturas automaticamente para os clientes após a conclusão da compra.

### Sprint 9 - Ligação das Encomendas e Notificações na Gestão de Perfil do Utilizador – Semana 11

Nesta última fase, focámo-nos na integração das **notificações em tempo real** e na ligação com as encomendas. As funcionalidades desenvolvidas foram:

* **Notificações para Alterações no Estado das Encomendas**: Os utilizadores passaram a receber notificações sempre que o estado da sua encomenda fosse atualizado (exemplo: "Encomenda enviada", "Encomenda entregue").
* **Histórico de Notificações**: Criámos uma aba de notificações dentro do perfil do utilizador para que pudesse rever atualizações anteriores.
* **Integração Total com o Perfil do Utilizador**: As notificações e encomendas foram organizadas dentro da gestão do perfil, permitindo ao utilizador aceder rapidamente a todas as informações relevantes.

**Testes contínuos (Durante todo o projeto):** Sempre que uma nova funcionalidade foi implementada, foram realizados testes para garantir a consistência da aplicação e a integração correta entre funcionalidades.

# Requisitos

O levantamento de requisitos para a nossa aplicação foi realizado com o objetivo de garantir que a solução desenvolvida atende às necessidades dos utilizadores e dos vendedores de produtos sustentáveis do projeto.

A estruturação dos requisitos baseou-se em critérios de viabilidade técnica, usabilidade e escalabilidade do sistema, garantindo que a plataforma possa crescer de forma eficiente sem comprometer a experiência do utilizador.

A identificação das funcionalidades a implementar levou em consideração a experiência de navegação dos utilizadores, a segurança na gestão de dados e a otimização do fluxo de compras e vendas.

A organização e priorização dos requisitos permitiu um planeamento estruturado do desenvolvimento, dividindo as funcionalidades em diferentes níveis de importância.

Dessa forma, assegurou-se que os elementos essenciais fossem desenvolvidos prioritariamente, enquanto funcionalidades complementares poderiam ser adicionadas conforme a evolução do projeto.

## Requisitos Funcionais

**Autenticação e Registo**

* O utilizador pode registar-se e iniciar sessão com e-mail e palavra-passe.
* O utilizador deve verificar o seu e-mail através de um link enviado antes de concluir o registo.
* O utilizador pode recuperar a palavra-passe através de um e-mail de redefinição.
* O utilizador pode manter a sessão ativa, optando por se lembrar da sessão ao iniciar sessão.
* O sistema deve garantir segurança no armazenamento das credenciais, utilizando hashing de palavras-passe.Pesquisa e Listagem de Produtos

**Pesquisa e Listagem de Produtos**

* O utilizador pode pesquisar lojas pelo nome através da barra de pesquisa global.
* A página de resultados deve exibir todas as lojas correspondentes à pesquisa, apresentando cartões com informações básicas e um link para a página da loja.
* No futuro, a pesquisa será expandida para permitir pesquisa por produtos e vendedores, com diferentes funcionalidades.

**Gestão do Carrinho de Compras**

* O utilizador pode adicionar produtos ao carrinho e definir a quantidade desejada.
* O utilizador pode remover produtos do carrinho a qualquer momento.
* O carrinho deve atualizar automaticamente o total da compra, considerando os produtos adicionados.
* O utilizador pode avançar para o checkout, onde escolhe o método de pagamento e a morada de entrega.

**Gestão de Encomendas**

* O utilizador pode criar uma encomenda, selecionando produtos e definindo a morada de entrega.
* O vendedor pode visualizar e gerir as encomendas recebidas, alterando o estado da encomenda (exemplo: "Em processamento", "Enviado", "Concluído").
* O administrador pode visualizar todas as encomendas, gerir disputas e cancelar pedidos se necessário.
* O utilizador recebe notificações sobre o estado da encomenda, garantindo acompanhamento em tempo real.

**Gestão de Lojas e Produtos (Vendedores)**

* O vendedor pode criar, editar e apagar lojas virtuais, com um limite máximo de três lojas.
* O vendedor pode adicionar, editar e remover produtos da sua loja.
* O vendedor pode gerir o stock de produtos, atualizando as quantidades disponíveis.
* O vendedor pode visualizar todas as encomendas associadas às suas lojas e modificar o estado das mesmas.
* O vendedor pode analisar métricas de desempenho no painel de controlo, como volume de vendas e avaliações.

**Mapa Interativo**

* O utilizador pode visualizar lojas próximas com base na sua localização, utilizando geolocalização.
* O utilizador pode clicar no ícone da loja no mapa para aceder diretamente à página da loja.
* O sistema armazena coordenadas das lojas através da geocodificação das moradas inseridas pelos vendedores.

**Avaliações e Comentários**

* O comprador pode avaliar uma loja do vendedor após uma compra, atribuindo uma pontuação de 1 a 5 estrelas e adicionar um comentário.
* Cada vendedor possui um rating global, que corresponde à média das avaliações de todas as suas lojas.

## Requisitos não Funcionais

**Desempenho**

* A aplicação deve responder instantaneamente às interações do utilizador, sem necessidade de recarregar páginas, aproveitando a *renderização* dinâmica do *React*.
* As atualizações da interface devem ser refletidas em tempo real através de sincronização eficiente com a base de dados.

**Usabilidade**

* A interface deve ser intuitiva e responsiva, garantindo uma experiência fluida em qualquer dispositivo.
* Todos os componentes *Material UI* devem ser utilizados de forma consistente para garantir uma experiência homogénea e moderna.
* A experiência do utilizador deve ser otimizada para evitar passos desnecessários no fluxo de compra e gestão de lojas.
* O design deve permitir ações rápidas, evitando cliques redundantes e mantendo uma navegação fluida.

**Compatibilidade**

* A aplicação deve ser compatível com todos os navegadores modernos (Google Chrome, Mozilla Firefox, Microsoft Edge e Safari) na última versão estável.
* Deve suportar diferentes resoluções de ecrã, garantindo que a experiência seja otimizada para:
  + Desktop (>1200px)
  + Tablets (768px - 1200px)
  + Dispositivos móveis (<768px)
* A aplicação deve ser otimizada para toques e gestos em dispositivos móveis, garantindo uma navegação eficiente sem depender de cliques de rato.
* O sistema deve manter suporte a futuras atualizações de bibliotecas como React e Material UI sem comprometer a estabilidade da aplicação.

## Priorização de Requisitos

A priorização dos requisitos é fundamental para garantir um desenvolvimento eficiente, focando primeiro nas funcionalidades essenciais para o funcionamento da nossa aplicação.

Para isso, utilizamos a **Matriz MoSCoW**, que classifica os requisitos em quatro categorias:

**Must Have (Obrigatórios)**

São requisitos críticos para o funcionamento da aplicação. Sem eles, o sistema não pode ser considerado funcional.

* Autenticação de utilizadores (Login/Registo via email).
* Pesquisa e listagem de produtos nome de loja.
* Gestão de carrinho de compras (adicionar/remover produtos, visualizar total).
* Processamento de encomendas (seleção de morada, checkout e confirmação).
* Gestão de lojas e produtos para vendedores.
* Mapa interativo para localização de lojas próximas.
* Segurança na autenticação.
* Compatibilidade com dispositivos móveis e desktop.

**Should Have (Importantes)**

Funcionalidades desejáveis, que melhoram a experiência do utilizador e a gestão da plataforma, mas que podem ser implementadas numa segunda fase.

* Histórico de encomendas para utilizadores e vendedores.
* Sistema de avaliações e comentários para vendedores e produtos.
* Gestão avançada de permissões (diferenciar administradores, vendedores e clientes).
* Sistema de caching para otimização do tempo de resposta.
* Melhoria na acessibilidade (suporte para leitores de ecrã e navegação por teclado).

**Could Have (Opcionais)**

Funcionalidades que agregam valor à aplicação, mas não são essenciais no lançamento inicial.

* Chat ao vivo entre compradores e vendedores.
* Estatísticas de vendas para vendedores.
* Recomendações personalizadas de produtos baseadas no histórico do utilizador.
* Implementação de múltiplos idiomas para expansão internacional.

**Won’t Have (Para Evolução Futura)**

Funcionalidades que não serão desenvolvidas na fase inicial, mas podem ser adicionadas em futuras atualizações.

* Integração com múltiplos métodos de pagamento.
* Sistema de fidelização e promoções para clientes frequentes.
* Marketplace com anúncios patrocinados para vendedores destacados.
* Suporte para integração com redes sociais para partilha de produtos.

1. **Desenho do Sistema e Arquitetura**

O capítulo de Desenho e Arquitetura é uma etapa crítica no desenvolvimento de software, onde as especificações e requisitos definidos, anteriormente, são transformados em representações gráficas e técnicas que servirão como guias para a implementação do sistema. Este capítulo assegura que o software está estruturado e organizado de forma eficiente e eficaz.

# Desenho do Sistema

Nesta seção, o desenho geral do sistema é abordado, envolvendo a criação de diagramas, layouts e outros elementos visuais que ilustram a estrutura e o funcionamento do software.

# Arquitetura e Tecnologia

Os modelos de arquitetura descrevem a estrutura de alto nível do sistema. Eles incluem a tecnologia a utilizar, a organização dos componentes do software, bem como as suas relações (por exemplo no caso de um sistema distribuídos).

# Diagramas e Modelos

Neste subcapítulo, são criados e apresentados diversos diagramas, como os de classes, sequência, componentes, entre outros (por exemplo diagramas relacionais das BD), para representar visualmente aspetos específicos do sistema. Estes diagramas ajudam na compreensão, comunicação e documentação da estrutura e dinâmica do Sistema.

1. **Implementação**

O capítulo de implementação é onde a conceção do software começa a tomar forma física. Nesta fase, o design e a arquitetura são transformados em código, seguindo os requisitos estabelecidos. A implementação é uma etapa prática e crítica, envolvendo a codificação, a revisão do código e a preparação para os testes e a implantação.

# Ambiente de Desenvolvimento

Esta seção descreve a configuração do ambiente de desenvolvimento. Inclui a seleção e a configuração das ferramentas de desenvolvimento (IDEs, entre outros), sistemas de controle de versão, *frameworks* e bibliotecas necessárias para desenvolver o software.

# Desenvolvimento

Neste subcapítulo, os programadores devem descrever em linhas gerais como o código fonte do software é coerente com desenho e arquitetura pretendida, eventualmente explicando opções técnicas tomadas durante o desenvolvimento. Também se deve detalhar as linguagens de programação utilizadas (e.g. C, JAVA, C#, HTML, CSS, JS, PHP, SQL, etc.), os padrões adotados (e.g estruturada, objetos, MVC, APIs, etc.).

# Controle de Versão

O controle de versão é essencial para rastrear e gerir as mudanças no código fonte. Esta seção explica o sistema de controle de versão utilizado (e.g. CVS, GIT, etc), os protocolos (estabelecidos dentro do grupo) para *commits*, revisões, *merges* e outros aspetos considerados críticos para garantir a integridade e a segurança do código.

1. **Testes**

Neste capítulo deve ser explicar de que forma a aplicação foi testada, no sentido de melhorar as suas funcionalidades, bem como para verificar e corrigir eventuais erros existentes.

1. **Evolução**

Nesta secção deve demonstrar como a sua aplicação poderá escalar de forma a por exemplo acomodar um aumento de utilizadores...

1. **Conclusões**

Nas conclusões deve ser feita uma síntese do corpo do relatório, destacando todos os aspetos julgados como relevantes no trabalho efetuado. Deverá fazer-se uso do espírito crítico com vista a sugerir e a identificar possíveis evoluções no trabalho.

1. **Fontes**

[identificação] autor, "título do livro", editora, número da edição, ano da edição. [identificação] nome do site da Internet, "link para a página".

[identificação] autor, "título do artigo", acta da conferência, volume, páginas, local de apresentação do artigo, ano. Exemplo:

**para um livro**

[1] Parsons,J.D., The Mobile Radio Propagation Channel, Pentech Press, London, UK, 1992.

i.e., nome(s) do(s) autor(es), título do livro, nome da editora, local da edição, país da edição, ano da edição.

**para um artigo em revista**

[2] Fernandes,L., "Developing a System Concept and Technologies for Mobile Broadband Communications", IEEE Personal Communications Magazine, Vol. 2, No. 1, Feb. 1995, pp. 54-59.

i.e., nome(s) do(s) autor(es), título do artigo, nome da revista, volume da edição, número da edição, mês da edição (abreviado com 3 letras, excepção aos meses com 4 letras), ano da edição, páginas.

**para uma comunicação em conferência**

[3] Correia,L.M. and Francês,P.O., "A Propagation Model for the Average Received Power in an Outdoor Environment in the Millimetre Waveband", in Proc. of VTC’94 - 44 th IEEE Vehicular Technology Conference, Stockholm, Sweden, June 1994.

i.e., nome(s) do(s) autor(es), título da comunicação, nome da conferência, local da conferência, país da conferência, mês da conferência (abreviado com 3 letras, excepção aos meses com 4 letras), ano da conferência.

**Anexos**

Devem ser remetidos, para anexo, todos os documentos (datasheets, technical reports, etc.) que sendo referidos no relatório não são considerados essenciais para a compreensão do trabalho desenvolvido.

Cada anexo deverá ser identificado com numeração romana, sem título, e mencionado na parte do texto que complementa.

Um dos anexos que deverá constar é um pequeno tutorial de utilização do sistema criado.

**Formatação e outras recomendações a ter em consideração**

O relatório deve ser escrito num processador de texto, tomando-se em consideração os seguintes aspetos:

Tipo de letra: Deve ser coerente em todo o relatório (Arial ou Calibri)

Tamanho de letra

Título do capítulo: 16 bold Secção: 14 bold Subsecção: 12 bold

Texto: 11

Espaço entre linhas: 1,5

O texto deve ser justificado

Todas as imagens e tabelas devem estar numeradas e legendadas

Exemplo:

1 – TÍTULO DO CAPÍTULO

1.1– TÍTULO DA SECÇÃO

1.1.1 – TÍTULO DA SUB-SECÇÃO

Margens: Superior: 2 cm Inferior: 2 cm Esquerda: 3 cm Direita: 2 cm

**Deve-se sempre evitar:**

* A subjetividade
* As figuras de estilo
* As palavras desnecessárias
* As repetições
* O excesso de adjetivação
* As palavras ambíguas

**Clareza e objetividade:**

O texto deve ser agradável de ler e esclarecedor do leitor, mesmo quando este não é especialista na matéria.

Deve rever o texto e ter especial atenção a:

* A pontuação e ortografia (em caso de dúvida, consultar um prontuário ou um dicionário)
* A divisão do texto em parágrafos que facilitem a leitura
* A utilização de títulos, subtítulos, numerações, alíneas, etc., quando necessário
* Cuidado na apresentação
* Inclusão da bibliografia dos documentos consultados
* Não fazer um trabalho tão longo que se torne maçador
* Não tentar impressionar com a quantidade, mas sim com a qualidade do trabalho
* Não juntar em anexo publicações ou documentos desnecessários
* Não copiar integral ou parcialmente os textos consultados – identificar as citações e fazer resumos