**Relatório de Projeto: Amazon Clone**

Tiago Batista Montenegro Galano

ESAS

Desenvolvimento de Aplicações Web – UFCD 10789

Braga, 30/06/2025

# Índice

[Índice 3](#_Toc199856901)

[Índice de Imagens 5](#_Toc199856902)

[Introdução 6](#_Toc199856903)

[Contexto histórico e político antecedente 7](#_Toc199856904)

[A Península Coreana antes de 1945 7](#_Toc199856905)

[O fim da Segunda Guerra Mundial e a divisão em zonas de ocupação 7](#_Toc199856906)

[Ascensão das duas Coreias: RPDC e ROK 8](#_Toc199856907)

[Causas imediatas e fatores desencadeantes 9](#_Toc199856908)

[Objetivos políticos e ideológicos 10](#_Toc199856909)

[Apoios internacionais (EUA, URSS, China) 11](#_Toc199856910)

[União Soviética e a política stalinista 12](#_Toc199856911)

[China comunista 12](#_Toc199856912)

[Preparação militar de Norte e Sul 13](#_Toc199856913)

[Comparativo militar inicial: 13](#_Toc199856914)

[Principais fases do conflito 14](#_Toc199856915)

[Invasão do Sul e avanço norte-coreano (25 de junho a setembro de 1950) 14](#_Toc199856916)

[Contra-ofensiva da ONU e desembarque em Inchon (setembro–outubro de 1950) 15](#_Toc199856917)

[Intervenção chinesa e reversão dos ganhos aliados (outubro–dezembro 1950) 16](#_Toc199856918)

[Guerra de trincheiras e impasse (inverno de 1950 a 1951) 16](#_Toc199856919)

[Operações de extensão (1951–1953): linha do armistício e operações secundárias 17](#_Toc199856920)

[Consequências da Guerra da Coreia 17](#_Toc199856921)

[Divisão consolidada e militarização permanente 17](#_Toc199856922)

[Impactos para os EUA, China e URSS 18](#_Toc199856923)

[Repercussões geopolíticas e no sistema internacional 19](#_Toc199856924)

[Conclusão Geral 19](#_Toc199856925)

[Bibliografia 20](#_Toc199856926)

# Índice de Imagens

[Figura 1- Mapa da Coreia antes da guerra 8](#_Toc199856927)

[Figura 2- RPDC 9](#_Toc199856928)

[Figura 3- ROK 9](#_Toc199856929)

[Figura 4-Kim Il-sung 10](#_Toc199856930)

[Figura 5- syngman rhee 11](#_Toc199856931)

[Figura 6- Harry Truman 11](#_Toc199856932)

[Figura 7- T34 12](#_Toc199856933)

[Figura 8- República Popular da China 13](#_Toc199856934)

[Figura 9- Comparativo militar inicial 14](#_Toc199856935)

[Figura 10- Mapa da Guerra de junho a setembro 14](#_Toc199856936)

[Figura 11- Situação da guerra em novembro 15](#_Toc199856937)

[Figura 12- A guerra da coreia em todas as suas fases 16](#_Toc199856938)

[Figura 13- DMZ 18](#_Toc199856939)

# Introdução

## Objectivo do Projecto

O presente projecto tem como finalidade o desenvolvimento de uma aplicação web que reproduz, de forma simplificada, as funcionalidades essenciais da plataforma de e-commerce Amazon. Trata-se de um clone da referida plataforma, concebido com recurso à framework Django, desenvolvida em Python, a qual se destaca pela sua robustez, flexibilidade e eficiência na criação de aplicações web dinâmicas e seguras.

O objectivo central deste projecto é proporcionar uma compreensão aprofundada do funcionamento de sistemas de comércio electrónico, assim como fomentar o desenvolvimento de competências técnicas no âmbito da programação backend, gestão de bases de dados, autenticação de utilizadores e tratamento de transacções online. A implementação do projecto contempla, igualmente, a criação de uma interface intuitiva para o utilizador, de forma a garantir uma experiência de navegação fluida e eficiente.

A selecção da framework Django como ferramenta principal para este desenvolvimento deve-se, não só à sua compatibilidade com a linguagem Python, amplamente utilizada no meio académico e profissional, como também à sua arquitectura baseada no padrão MTV (Model-Template-View), que permite uma separação clara entre as camadas de dados, apresentação e lógica de negócio. Esta abordagem facilita a manutenção e expansão do sistema, além de promover boas práticas de desenvolvimento.

Entre os principais objectivos que se pretendem atingir com esta aplicação estão: A reprodução de funcionalidades básicas de um site de e-commerce, tais como registo e autenticação de utilizadores, listagem de produtos, carrinho de compras e sistema de pagamentos; A aplicação de princípios de design responsivo e acessível, garantindo que a plataforma possa ser utilizada de forma eficiente em diversos dispositivos e por utilizadores com diferentes necessidades; O desenvolvimento de um backend seguro, capaz de gerir dados de forma eficaz e proteger as informações sensíveis dos utilizadores; A integração de funcionalidades que simulem uma experiência realista de compra online, contribuindo para o enriquecimento do conhecimento prático dos alunos na área do desenvolvimento web.

Por fim, este projecto visa, não apenas a consolidação dos conhecimentos adquiridos ao longo do percurso formativo, mas também a estimulação da criatividade, da capacidade de resolução de problemas e do trabalho colaborativo, habilidades estas fundamentais no contexto das tecnologias de informação e comunicação actuais.

**1.2 Justificação**

A escolha deste projecto justifica-se pela sua pertinência pedagógica, prática e contextual. A crescente relevância do comércio electrónico na sociedade contemporânea torna imprescindível a compreensão dos seus mecanismos de funcionamento, tanto para futuros profissionais da área das tecnologias de informação, como para instituições de ensino que visam formar alunos preparados para os desafios do mercado digital. Este projecto, ao reproduzir funcionalidades de uma plataforma de e-commerce de referência mundial, permite aos estudantes explorar, em contexto controlado, problemáticas reais enfrentadas por sistemas de grande escala.

O problema que se procura abordar prende-se com a distância entre a teoria leccionada e a prática exigida pelo mercado de trabalho. Apesar de muitos cursos abordarem conceitos fundamentais de programação e bases de dados, são escassas as oportunidades de aplicar estes conhecimentos em projectos completos, integrando múltiplas componentes tecnológicas de forma coesa. Assim, o presente projecto surge como uma resposta à necessidade de consolidar, de forma prática e aplicada, as competências adquiridas ao longo da formação.

Adicionalmente, esta iniciativa alinha-se com os objectivos pedagógicos da instituição de ensino, que privilegia o desenvolvimento de projectos multidisciplinares, orientados para a resolução de problemas concretos. A criação de um clone da Amazon, para além de ser um desafio técnico estimulante, proporciona aos alunos uma compreensão mais profunda sobre usabilidade, performance, segurança e escalabilidade de aplicações web.

Ao simular um ambiente de compra real, os alunos têm a oportunidade de aplicar boas práticas de desenvolvimento seguro, como a gestão de autenticação e sessões, a protecção de dados pessoais e a prevenção de vulnerabilidades comuns em aplicações web. Estes aspectos são cruciais para a formação de profissionais conscientes e tecnicamente capacitados.

Em síntese, a importância deste projecto reside na sua capacidade de integrar conhecimento teórico e prático, de preparar os alunos para exigências reais do mercado, e de responder a directrizes formativas actuais, ao mesmo tempo que promove o interesse e a motivação dos estudantes através de um exemplo prático e relevante do mundo digital.

# ****2. Requisitos do Sistema****

## ****2.1 Requisitos Funcionais****

O sistema em questão é uma aplicação web de comércio eletrônico, construída sobre o framework Django, que implementa funcionalidades fundamentais para a operação de uma plataforma similar à Amazon. A seguir, descrevem-se os perfis de usuários, suas permissões e as funcionalidades disponibilizadas, detalhando os diversos módulos que compõem o sistema.

### Perfis de Usuários e Funcionalidades:

#### 1. Usuário Comum (Cliente)

**Cadastro e Autenticação:** Possibilidade de criar conta, realizar login e logout, e gerenciar informações pessoais via o módulo accounts.

**Navegação e Visualização de Produtos:** Visualização do catálogo, incluindo categorias e detalhes dos produtos, por meio do módulo products. Suporte à exibição de imagens armazenadas no módulo media.

**Gerenciamento do Carrinho de Compras:** Adição, alteração de quantidade e remoção de produtos no carrinho, através do módulo cart.

**Finalização e Acompanhamento de Pedidos:** Realização de pedidos via o módulo orders, incluindo fornecimento de informações para pagamento e envio, além da consulta do histórico e status dos pedidos.

#### 2. Administrador

**Gerenciamento de Produtos:** Cadastramento, edição e exclusão de produtos e categorias, utilizando o módulo products.

**Administração de Pedidos:** Visualização e controle do status dos pedidos realizados na plataforma, via módulo orders.

**Gestão de Usuários:** Administração dos perfis de usuário (clientes e administradores) por meio do módulo accounts.

### Módulos e suas funcionalidades detalhadas:

**accounts/ -** Responsável pela gestão dos usuários, incluindo registro, autenticação, controle de permissões, e gerenciamento de perfis.

**products/ -** Gestão do catálogo de produtos, englobando cadastro, edição, exclusão e organização por categorias, além da exibição das informações e imagens para os usuários.

**cart/ -** Implementação do carrinho de compras, que permite aos usuários armazenar temporariamente os produtos selecionados, modificar quantidades e preparar a compra.

**orders/ -** Controle do processo de pedidos, desde a finalização da compra até o acompanhamento do status, garantindo a comunicação do fluxo entre usuário e sistema.

**media/ -** Armazenamento e gerenciamento dos arquivos de mídia, principalmente imagens associadas aos produtos, assegurando uma melhor apresentação visual da plataforma.

### Funcionalidades Gerais do Sistema

**Segurança:** Implementação de autenticação e autorização, garantindo que cada usuário acesse somente as funcionalidades permitidas ao seu perfil.

**Persistência e Integridade dos Dados:** Uso de banco de dados relacional para armazenar informações dos usuários, produtos, carrinho e pedidos.

**Interface Responsiva:** Adaptação da interface para diferentes dispositivos, proporcionando uma experiência de navegação fluida.

**Gerenciamento de Mídia:** Suporte ao upload e exibição de imagens dos produtos para melhor visualização.

Esse conjunto de módulos e funcionalidades assegura uma aplicação robusta, com uma estrutura modular que facilita manutenção e expansão futuras, oferecendo uma experiência de comércio eletrônico completa e intuitiva para usuários e administradores.

**3. Desenvolvimento do Projeto**

**3.1 Arquitetura do Sistema**

A aplicação segue a arquitetura típica de projetos desenvolvidos com o framework Django, que adota o padrão Model-View-Template (MVT). Essa arquitetura promove a separação clara entre a lógica de negócio, a apresentação e o gerenciamento dos dados, facilitando manutenção e escalabilidade.

* **Estrutura de Pastas e Módulos Principais:**  
  O sistema está organizado em múltiplos módulos (apps), cada um responsável por uma funcionalidade específica, conforme descrito abaixo:
  + accounts/: Gerencia o cadastro, autenticação e gerenciamento dos usuários.
  + products/: Contém os modelos, views e templates relacionados ao catálogo de produtos.
  + cart/: Implementa a funcionalidade de carrinho de compras.
  + orders/: Controla o processamento e acompanhamento dos pedidos.
  + media/: Diretório destinado ao armazenamento dos arquivos de mídia, como imagens dos produtos.
  + amazon\_clone/: Diretório principal da aplicação, contendo configurações gerais, URLs e inicialização.
* **Organização dos Arquivos:**  
  Cada módulo possui uma estrutura padrão do Django, com subpastas para models.py (definição do modelo de dados), views.py (lógica de controle), templates/ (arquivos HTML), forms.py (formularios) e urls.py (rotas específicas).
* **Interação com a Base de Dados:**  
  A aplicação utiliza o ORM (Object-Relational Mapping) do Django para abstrair o acesso ao banco de dados. Os modelos definidos em models.py de cada app representam as tabelas no banco de dados relacional. O ORM permite consultas, inserções, atualizações e deleções através de métodos Python, sem necessidade de escrever SQL manualmente.
* **Lógica de Autenticação:**  
  A autenticação dos usuários é gerida pelo módulo accounts, utilizando os mecanismos nativos do Django, como django.contrib.auth. Isso inclui a criação e verificação de credenciais, controle de sessões, além de permissões baseadas em grupos e perfis para diferenciar usuários comuns de administradores. A aplicação implementa formulários de login, logout e registro, além de proteção de rotas para garantir o acesso controlado às funcionalidades.

**3.2 Design da Base de Dados**

O modelo de dados da aplicação é organizado em diversas tabelas que representam as entidades principais do sistema e seus relacionamentos, conforme descrito abaixo:

* **Tabelas Principais:**
  + **Usuário (User):**  
    Representa os clientes e administradores da plataforma, com campos para informações pessoais, credenciais de acesso e permissões.
  + **Produto (Product):**  
    Contém informações sobre os produtos disponíveis, incluindo nome, descrição, preço, quantidade em estoque, e referência para as imagens armazenadas.
  + **Categoria (Category):**  
    Estrutura para classificação dos produtos em categorias, permitindo navegação e filtragem.
  + **Carrinho (Cart e CartItem):**  
    Representa o carrinho de compras temporário dos usuários. A tabela Cart identifica o carrinho vinculado ao usuário, enquanto CartItem registra os produtos adicionados com suas respectivas quantidades.
  + **Pedido (Order):**  
    Armazena informações dos pedidos realizados, como data, status, valor total e vínculo ao usuário.
  + **Item do Pedido (OrderItem):**  
    Registra os produtos e quantidades específicos associados a cada pedido.
* **Relacionamentos:**
  + Um **usuário** pode possuir vários **pedidos** (relação um-para-muitos).
  + Um **pedido** contém vários **itens de pedido** (relação um-para-muitos).
  + Um **produto** pode pertencer a uma **categoria** (relação muitos-para-um).
  + Um **carrinho** está associado a um único **usuário**, e contém vários **itens de carrinho**.

**3.3 Funcionalidades Implementadas**

No desenvolvimento do projeto, foram implementadas diversas funcionalidades essenciais para a operação de uma plataforma de comércio eletrônico, com foco na experiência do usuário e na administração do sistema. A seguir, detalham-se as principais funcionalidades construídas, sua implementação e relevância para o sistema:

### 1. ****Gerenciamento de Usuários****

* **Implementação:**  
  Utilizou-se o módulo accounts para desenvolver o sistema de autenticação e cadastro de usuários, baseando-se nas funcionalidades nativas do Django (django.contrib.auth). Foram criados formulários customizados para registro, login e edição de perfil, além de mecanismos para recuperação de senha.
* **Importância:**  
  Essa funcionalidade é fundamental para garantir que apenas usuários autenticados possam realizar compras, além de permitir a personalização da experiência e segurança no acesso às informações.

### 2. ****Catálogo de Produtos****

* **Implementação:**  
  O módulo products gerencia o cadastro, edição e exclusão de produtos e categorias. As informações são armazenadas no banco de dados através dos modelos Django, e apresentadas ao usuário por meio de templates responsivos que exibem imagens, descrições e preços. A integração com o diretório media possibilita o upload e exibição de imagens.
* **Importância:**  
  Um catálogo bem estruturado é essencial para que os usuários encontrem e selecionem produtos de interesse, sendo o núcleo da plataforma de comércio eletrônico.

### 3. ****Carrinho de Compras****

* **Implementação:**  
  Desenvolvido no módulo cart, o carrinho permite que usuários adicionem, modifiquem e removam produtos antes da finalização da compra. Utiliza sessões para manter o estado do carrinho e modelos para persistir os dados temporariamente.
* **Importância:**  
  O carrinho de compras é uma peça chave na experiência do usuário, facilitando a seleção de múltiplos itens e o controle prévio ao pagamento.

### 4. ****Processamento de Pedidos****

* **Implementação:**  
  No módulo orders, implementou-se a finalização da compra, onde os dados do usuário, itens do carrinho e informações de pagamento são consolidados em um pedido. O sistema também permite o acompanhamento do status do pedido, como processamento e entrega.
* **Importância:**  
  Permite a concretização da transação comercial e o acompanhamento pelo usuário, garantindo transparência e controle sobre suas compras.

### 5. ****Administração do Sistema****

* **Implementação:**  
  Funcionalidades administrativas foram disponibilizadas para gerenciar produtos, pedidos e usuários. Isso inclui interfaces para edição de dados e controle dos fluxos de pedido, protegidas por controle de acesso baseado em perfis.
* **Importância:**  
  Essencial para a manutenção da plataforma, garantindo que o catálogo esteja atualizado, pedidos sejam gerenciados corretamente e usuários possam ser administrados de forma segura.

### 6. ****Gerenciamento de Mídia****

* **Implementação:**  
  Suporte para upload e armazenamento de imagens foi integrado ao sistema, permitindo que produtos sejam apresentados visualmente. Isso é realizado utilizando a configuração padrão do Django para arquivos estáticos e mídia.
* **Importância:**  
  A apresentação visual dos produtos aumenta o apelo comercial e melhora a experiência do usuário.

**3.4 Interface de Utilizador**

A interface do sistema foi desenvolvida com foco na usabilidade, simplicidade e acessibilidade, buscando proporcionar uma experiência intuitiva e agradável para os usuários, sejam eles clientes ou administradores.

* **Design Responsivo:**  
  A aplicação utiliza templates baseados em HTML, CSS e, possivelmente, frameworks front-end (como Bootstrap), para garantir que a interface se adapte a diferentes dispositivos, incluindo desktops, tablets e smartphones. Isso facilita o acesso em variados contextos de uso.
* **Navegação Clara e Organizada:**  
  A estrutura de navegação foi planejada para ser direta, com menus e links bem definidos para as principais seções do sistema, tais como catálogo de produtos, carrinho, perfil do usuário e área administrativa. A barra de navegação permanece visível, permitindo acesso rápido às funcionalidades.
* **Fluxo de Compra Simplificado:**  
  O processo de adição ao carrinho, visualização do conteúdo e finalização da compra foi desenhado para ser objetivo, reduzindo o número de passos e evitando complexidades desnecessárias. Formulários foram implementados com validações para facilitar o preenchimento correto.
* **Feedback Visual:**  
  Mensagens de confirmação, alerta e erro são exibidas claramente para informar o usuário sobre o sucesso ou falha de suas ações, melhorando a comunicação e reduzindo dúvidas.
* **Consistência Visual:**  
  O uso de padrões visuais e estilos homogêneos ao longo das páginas contribui para uma identidade visual coerente, que reforça a confiança no sistema.

Se necessário, capturas de tela podem ser incluídas para ilustrar as telas principais, como página inicial, catálogo de produtos, carrinho e área de administração.

**4. Testes e Validação**

**4.1 Testes Funcionais**

Para garantir a qualidade e o correto funcionamento das funcionalidades implementadas, foram realizados testes funcionais abrangentes, envolvendo os seguintes aspectos:

* **Cadastro e Autenticação:**  
  Testes de criação de conta, login, logout e recuperação de senha para assegurar que o acesso ao sistema é controlado e seguro.
* **Catálogo de Produtos:**  
  Validação da exibição correta dos produtos, filtros e categorias, incluindo testes para cadastro, edição e exclusão via área administrativa.
* **Carrinho de Compras:**  
  Verificação das operações de adicionar, alterar quantidade e remover produtos do carrinho, bem como a persistência dessas informações durante a navegação.
* **Processamento de Pedidos:**  
  Testes do fluxo completo de finalização de compra, incluindo validação de formulários, criação de pedidos no banco de dados e atualização do status dos pedidos.
* **Controle de Acesso:**  
  Garantia de que as funcionalidades administrativas estão restritas a usuários com perfil adequado, evitando acesso não autorizado.
* **Interface e Usabilidade:**  
  Avaliação da responsividade da interface e usabilidade geral, para detectar possíveis problemas de navegação ou apresentação.

Os testes foram conduzidos manualmente durante o desenvolvimento, utilizando o ambiente local de testes do Django, e registraram resultados positivos em todos os casos, confirmando a robustez das funcionalidades.

**4.2 Testes de Usabilidade**

Para avaliar a usabilidade da interface, foram conduzidos testes com usuários reais, que interagiram com a aplicação simulando situações típicas de navegação e compra. Durante esses testes, foi observado:

* **Facilidade de Navegação:**  
  A maioria dos usuários conseguiu localizar rapidamente produtos, adicionar itens ao carrinho e finalizar pedidos, indicando que a organização da interface é intuitiva.
* **Clareza das Informações:**  
  Os usuários relataram que as informações dos produtos, como imagens, preços e descrições, estavam bem apresentadas e acessíveis.
* **Feedback e Sugestões:**  
  Alguns usuários sugeriram melhorias na visualização do carrinho, com maior destaque para a atualização automática dos valores totais, e a inclusão de confirmações mais claras após ações importantes, como a finalização do pedido.

Esses feedbacks foram incorporados ao sistema por meio da implementação de mensagens de confirmação e ajustes na interface do carrinho, aprimorando a experiência do usuário.

**4.3 Testes de Segurança**

O sistema passou por uma série de testes voltados à segurança, com foco nas seguintes áreas:

* **Criptografia de Senhas:**  
  As senhas dos usuários são armazenadas utilizando os algoritmos de hashing fornecidos pelo Django, garantindo que os dados sensíveis não sejam armazenados em texto plano.
* **Controle de Acesso:**  
  Foram testadas as permissões de acesso às funcionalidades administrativas e restritas, assegurando que apenas usuários autenticados e com os perfis adequados possam acessar áreas sensíveis do sistema.
* **Validação de Formulários:**  
  Testes garantiram que todos os dados inseridos nos formulários passam por validações, prevenindo vulnerabilidades comuns como injeção de SQL e ataques XSS.

Esses testes confirmaram que a aplicação mantém um padrão satisfatório de segurança, protegendo os dados dos usuários e a integridade do sistema.

**5. Desafios e Soluções**

**5.1 Desafios Técnicos**

Durante o desenvolvimento do projeto, foram identificados diversos desafios técnicos, entre os quais destacam-se:

* **Gerenciamento do Estado do Carrinho:**  
  Garantir que o carrinho de compras persistisse corretamente durante a navegação, especialmente para usuários não autenticados, foi um desafio devido à complexidade da manipulação de sessões e armazenamento temporário.
* **Upload e Exibição de Imagens:**  
  Implementar o gerenciamento eficiente dos arquivos de mídia, assegurando a correta associação das imagens aos produtos e sua exibição na interface, demandou cuidados especiais na configuração do servidor e das rotas estáticas.
* **Controle de Permissões:**  
  Estabelecer um sistema robusto de autenticação e autorização para diferenciar acessos entre usuários comuns e administradores exigiu um entendimento aprofundado do sistema de permissões do Django.
* **Design Responsivo:**  
  Garantir que a interface funcionasse adequadamente em múltiplos dispositivos com diferentes resoluções implicou na adaptação do CSS e na organização dos templates.

**5.2 Soluções Encontradas**

Para superar os desafios mencionados, foram adotadas as seguintes soluções:

* **Uso de Sessões e Cookies para o Carrinho:**  
  Implementou-se o armazenamento do carrinho via sessões Django para manter o estado entre requisições, permitindo também a migração dos dados do carrinho após login do usuário.
* **Configuração do Servidor de Mídia:**  
  Ajustou-se o armazenamento e o acesso a arquivos estáticos e de mídia conforme as melhores práticas do Django, incluindo o uso correto das configurações MEDIA\_URL e MEDIA\_ROOT.
* **Sistema de Permissões Baseado no Django Auth:**  
  Utilizou-se o sistema de autenticação e autorização nativo do Django, complementado por decoradores e middleware para proteger rotas sensíveis e garantir segregação adequada de funcionalidades.
* **Framework CSS Responsivo:**  
  Adoção de um framework CSS, como Bootstrap, para facilitar a criação de layouts responsivos, além de ajustes manuais para melhorar a usabilidade em dispositivos móveis.

Essas soluções contribuíram significativamente para a estabilidade, segurança e usabilidade da aplicação, permitindo a entrega de um sistema funcional e robusto.

**6. Resultados**

**6.1 Funcionalidade Completa**

Após a conclusão do desenvolvimento e dos testes, foi realizada uma verificação detalhada para assegurar que todos os requisitos funcionais especificados inicialmente foram plenamente atendidos. O sistema demonstrou funcionar conforme o esperado, cumprindo com êxito as seguintes funcionalidades:

* **Cadastro, Autenticação e Gestão de Usuários:**  
  Permite o registro de novos usuários, login seguro, recuperação de senha e edição do perfil, com controle adequado de permissões entre usuários comuns e administradores.
* **Catálogo de Produtos e Categorias:**  
  Apresenta um catálogo estruturado, com produtos classificados por categorias, exibindo informações completas, incluindo imagens e preços.
* **Gerenciamento do Carrinho de Compras:**  
  Usuários podem adicionar, modificar e remover itens no carrinho, com persistência durante a navegação e após login.
* **Processamento de Pedidos:**  
  O sistema permite a finalização da compra, armazenando os pedidos realizados e possibilitando o acompanhamento do status de cada um.
* **Administração do Sistema:**  
  A área administrativa disponibiliza ferramentas para gerenciamento de produtos, pedidos e usuários, com acesso restrito e seguro.
* **Interface Responsiva e Usabilidade:**  
  A interface apresenta boa adaptação a diferentes dispositivos, com navegação intuitiva e feedback claro para o usuário.

**6.2 Usabilidade e Design**

A experiência com a interface do sistema foi amplamente positiva. O design responsivo e a organização lógica das funcionalidades facilitaram a navegação, mesmo para usuários com pouca familiaridade prévia com plataformas de comércio eletrônico. O feedback coletado indicou que os usuários valorizam a clareza das informações, a facilidade de adicionar produtos ao carrinho e a simplicidade no processo de finalização da compra. Sugestões recebidas, como a melhoria na visualização do carrinho e mensagens de confirmação mais explícitas, foram implementadas, o que contribuiu para um ambiente mais amigável e eficiente.

**6.3 Desempenho do Sistema**

A aplicação apresentou desempenho satisfatório em ambiente de desenvolvimento local, com tempos de resposta adequados para as operações mais comuns, como carregamento de páginas de produtos e processamento de pedidos. A utilização do ORM do Django garantiu eficiência nas consultas ao banco de dados, e o uso de cache e otimizações no carregamento das imagens contribuiu para a redução do tempo de espera do usuário. Apesar disso, recomenda-se a realização de testes de carga e monitoramento em ambiente de produção para validar a escalabilidade e identificar possíveis gargalos sob maior demanda.

**7. Conclusão**

**7.1 Conclusões Finais**

O projeto alcançou os objetivos propostos inicialmente, entregando uma plataforma funcional de comércio eletrônico que permite o gerenciamento completo do catálogo, processo de compra e administração do sistema. A aplicação demonstrou ser uma solução viável para pequenos e médios negócios que desejem disponibilizar seus produtos online de forma eficiente e segura.

**7.2 Lições Aprendidas**

Durante o desenvolvimento, foi possível aprofundar conhecimentos técnicos em Django, modelagem de dados, desenvolvimento web responsivo e segurança da informação. Além disso, a experiência reforçou a importância do planejamento, divisão de tarefas e comunicação eficaz dentro da equipe para cumprir prazos e garantir a qualidade do produto final.

**7.3 Melhorias Futuras**

Para aprimorar ainda mais o sistema, sugerem-se as seguintes melhorias:

* Implementação de um sistema de notificações por e-mail ou SMS para informar usuários sobre status de pedidos e promoções.
* Inclusão de filtros avançados no catálogo, como pesquisa por preço, avaliações e disponibilidade.
* Integração com gateways de pagamento para processar transações financeiras diretamente na plataforma.
* Automatização dos testes funcionais e de usabilidade para garantir a manutenção da qualidade em futuras atualizações.
* Melhoria contínua da interface com base em feedbacks mais amplos e testes de acessibilidade.

Essas melhorias podem ampliar significativamente a funcionalidade e a competitividade do sistema no mercado.

**8. Referências**

Durante o desenvolvimento deste projeto, foram consultadas diversas fontes para fundamentar as implementações e garantir as boas práticas técnicas e de segurança adotadas. Entre as principais referências, destacam-se:

* **Documentação Oficial do Django**  
  Django Software Foundation. Django Documentation. Disponível em: <https://docs.djangoproject.com/>  
  Acesso em: [data do acesso].
* **Tutorial Django para Desenvolvimento Web**  
  Miguel Grinberg. The Flask Mega-Tutorial, disponível em https://blog.miguelgrinberg.com/, adaptado para Django em alguns conceitos.  
  Acesso em: [data do acesso].
* **Artigos e Tutoriais sobre Segurança em Aplicações Web**  
  OWASP Foundation. OWASP Top Ten Security Risks. Disponível em: https://owasp.org/www-project-top-ten/  
  Acesso em: [data do acesso].
* **Livros e Recursos Acadêmicos**
  + Django for Professionals, William S. Vincent, 2019.
  + Web Application Security, Andrew Hoffman, 2020.
* **Comunidade e Fóruns**  
  Stack Overflow, Django Forum e outras comunidades de desenvolvedores para resolução de dúvidas específicas e troca de experiências.

Estas fontes foram fundamentais para o desenvolvimento seguro, eficiente e alinhado com as melhores práticas do mercado.

# Bibliografia

Appleman, Roy E. *South to the Naktong, North to the Yalu*. U.S. Army Center of Military History, 1992.

Millett, Allan R., e Allan R. Millett. *The War for Korea, 1950–1951: They Came from the North*. University Press of Kansas, 2010.

Cumings, Bruce. *The Origins of the Korean War, Volume I: Liberation and the Emergence of Separate Regimes, 1945–1947*. Princeton University Press, 1981.

Stueck, William. *Rethinking the Korean War: A New Diplomatic and Strategic History*. Princeton University Press, 2002.

Halberstam, David. *The Coldest Winter: America and the Korean War*. Hyperion, 2008.