

1. Introdução

Este trabalho apresenta o desenvolvimento de um sistema de **loja de tênis** utilizando o framework **Django** e o banco de dados **SQLite**, com o objetivo de aplicar, na prática, os conceitos de **modelagem de banco de dados** e **operações CRUD (Create, Read, Update, Delete)**.

A partir de um problema do mundo real – o cadastro e gerenciamento de produtos (tênis) e anúncios de venda – foram realizadas as etapas de **levantamento de requisitos**, **modelagem conceitual**, **modelagem lógica**, **projeto físico** e **implementação da aplicação**, conforme as orientações da disciplina de Banco de Dados.

O sistema desenvolvido permite que usuários autenticados cadastrem tênis, gerenciem um catálogo de produtos e criem anúncios de venda vinculados a esses tênis, garantindo controle de acesso e persistência dos dados em um banco relacional. Ao longo deste documento, são apresentados o minimundo do problema, os requisitos identificados, o modelo conceitual, o modelo lógico relacional, a implementação física em SQL, a modelagem orientada a objetos e a descrição da aplicação CRUD construída em Django.

2. Levantamento e Análise de Requisitos

2.1. Minimundo

Uma loja voltada para a venda de tênis de diferentes marcas, modelos, tamanhos e estilos decidiu desenvolver um sistema informatizado para organizar melhor seu catálogo de produtos e facilitar a criação de anúncios de venda, acessado apenas por usuários cadastrados.

No sistema, cada usuário possui um login e uma senha para acessar a aplicação. Após autenticado, o usuário pode cadastrar diferentes modelos de tênis disponíveis para venda. Para cada tênis cadastrado, são armazenadas informações como **marca**, **modelo**, **tamanho**, **cor**, **tipo** (por exemplo, casual, esportivo, corrida, skate) e, opcionalmente, uma **foto** ilustrativa. Esses dados são utilizados tanto para controle interno quanto para a divulgação dos produtos.

Além do cadastro de tênis, o sistema permite que o usuário crie **anúncios** associados a um tênis específico. Cada anúncio contém um **título**, uma **descrição** detalhada do produto e o **preço** de venda, além do registro automático da **data de criação** e da **última atualização**. Um anúncio sempre pertence a um único usuário (seu criador) e sempre está vinculado a um único tênis previamente cadastrado no sistema. Assim, um mesmo tênis pode ter mais de um anúncio ao longo do tempo, e um usuário pode manter vários anúncios ativos.

Os usuários podem consultar a lista de todos os tênis cadastrados, editar as informações de um tênis ou removê-lo quando não for mais necessário. Da mesma forma, podem visualizar a lista de seus anúncios, cadastrar novos anúncios, atualizar anúncios existentes (por exemplo, alterando preço ou descrição) e excluir anúncios que não estejam mais válidos. O sistema também pode disponibilizar uma visualização geral de todos os anúncios cadastrados, permitindo que os usuários naveguem pelas ofertas disponíveis.

Do ponto de vista de negócio, o sistema deve garantir que **apenas usuários autenticados** possam cadastrar, alterar ou excluir tênis e anúncios. Não é permitido criar um anúncio sem que exista um tênis associado, nem manter anúncios sem vínculo com um usuário. As informações cadastradas são armazenadas em um **banco de dados relacional**, que será utilizado para consultas, geração de relatórios simples e, futuramente, integração com outras funcionalidades de venda online ou outros sistemas.

2.2. Requisitos de Dados

Com base no minimundo, foram identificados os seguintes dados que precisam ser armazenados:

- **Usuário**
 - Identificador do usuário
 - Nome
 - E-mail
 - Senha (ou hash de senha)
 - Data de cadastro
- **Tênis**
 - Identificador do tênis
 - Marca
 - Modelo
 - Tamanho
 - Cor
 - Tipo (casual, esportivo, corrida, etc.)

- Foto (opcional)
 - **Anúncio**
 - Identificador do anúncio
 - Título
 - Descrição
 - Preço
 - Data de criação
 - Data de atualização
 - Referência ao usuário que criou o anúncio
 - Referência ao tênis associado ao anúncio
-

2.3. Requisitos Funcionais

Alguns dos principais requisitos funcionais do sistema são:

- **RF01** – Permitir que o usuário faça **login** no sistema.
- **RF02** – Permitir que o usuário faça **logout** do sistema.
- **RF03** – Permitir que o usuário **cadastre** novos tênis.
- **RF04** – Permitir que o usuário **liste** todos os tênis cadastrados.
- **RF05** – Permitir que o usuário **edite** os dados de um tênis.
- **RF06** – Permitir que o usuário **exclua** um tênis.
- **RF07** – Permitir que o usuário **cadastre** anúncios de venda associados a um tênis.
- **RF08** – Permitir que o usuário **liste** seus anúncios.
- **RF09** – Permitir que o usuário **liste todos os anúncios** cadastrados no sistema.
- **RF10** – Permitir que o usuário **visualize detalhes** de um anúncio (título, descrição, preço, foto do tênis, etc.).

- **RF11** – Permitir que o usuário **edite** um anúncio (título, descrição, preço).
 - **RF12** – Permitir que o usuário **exclua** um anúncio.
 - **RF13** – Garantir que apenas usuários autenticados possam acessar as funcionalidades de cadastro, edição e exclusão.
 - **RF14** – Registrar automaticamente as **datas de criação e atualização** dos anúncios.
-

3. Análise Funcional

Nesta seção são descritas as principais transações (funcionalidades) do sistema, incluindo entradas, saídas e regras de negócio associadas.

3.1. Transações Principais

T1 – Realizar Login

- **Descrição:** Autenticar o usuário no sistema.
 - **Entradas:** E-mail (ou nome de usuário), senha.
 - **Saídas:** Acesso às funcionalidades internas do sistema ou mensagem de erro.
 - **Regras de negócio:**
 - O usuário deve existir na base de dados.
 - A combinação de e-mail (ou username) e senha deve ser válida.
 - Em caso de falha, deve ser exibida mensagem informando que as credenciais são inválidas.
-

T2 – Cadastrar Tênis

- **Descrição:** Registrar um novo tênis no sistema.
- **Entradas:** Marca, modelo, tamanho, cor, tipo, foto (opcional).

- **Saídas:** Registro do tênis armazenado no banco de dados e exibição na listagem de tênis.
 - **Regras de negócio:**
 - Todos os campos obrigatórios (marca, modelo, tamanho, cor, tipo) devem ser preenchidos.
 - O usuário deve estar autenticado.
-

T3 – Listar Tênis

- **Descrição:** Exibir todos os tênis cadastrados.
 - **Entradas:** (Opcional) filtros de pesquisa (por marca, modelo, tamanho etc.).
 - **Saídas:** Lista de tênis com suas principais informações.
 - **Regras de negócio:**
 - Apenas usuários autenticados podem acessar a lista.
 - Podem ser aplicados filtros de busca para facilitar a localização dos produtos.
-

T4 – Editar Tênis

- **Descrição:** Alterar as informações de um tênis existente.
 - **Entradas:** Identificador do tênis, novos dados (marca, modelo, tamanho, cor, tipo, foto).
 - **Saídas:** Dados atualizados no banco e refletidos na listagem.
 - **Regras de negócio:**
 - O tênis deve existir no banco de dados.
 - O usuário deve estar autenticado.
-

T5 – Excluir Tênis

- **Descrição:** Remover um tênis do sistema.
 - **Entradas:** Identificador do tênis.
 - **Saídas:** Tênis removido da base e da listagem.
 - **Regras de negócio:**
 - O tênis deve existir.
 - O sistema deve solicitar confirmação antes da exclusão (por meio de modal ou tela de confirmação).
 - O usuário deve estar autenticado.
 - Dependendo das regras, pode haver tratamento específico se existirem anúncios vinculados àquele tênis.
-

T6 – Cadastrar Anúncio

- **Descrição:** Criar um anúncio de venda para um tênis já cadastrado.
 - **Entradas:** Seleção de um tênis, título, descrição, preço.
 - **Saídas:** Anúncio armazenado e exibido na listagem de anúncios.
 - **Regras de negócio:**
 - O usuário deve estar autenticado.
 - É obrigatório selecionar um tênis previamente cadastrado.
 - Título, descrição e preço são campos obrigatórios.
 - A data de criação deve ser preenchida automaticamente.
-

T7 – Listar Meus Anúncios

- **Descrição:** Exibir todos os anúncios cadastrados por um determinado usuário.
- **Entradas:** Identificação do usuário logado (obtida pela sessão).

- **Saídas:** Lista de anúncios daquele usuário.
 - **Regras de negócio:**
 - Apenas o próprio usuário pode visualizar a lista de “Meus anúncios”.
 - O usuário deve estar autenticado.
-

T8 – Listar Todos os Anúncios

- **Descrição:** Exibir todos os anúncios cadastrados no sistema.
 - **Entradas:** (Opcional) filtros de busca.
 - **Saídas:** Lista geral de anúncios com informações principais (título, tênis, preço).
 - **Regras de negócio:**
 - O usuário deve estar autenticado.
 - Podem ser aplicados filtros de busca para facilitar a navegação pelos anúncios.
-

T9 – Visualizar Detalhes do Anúncio

- **Descrição:** Mostrar as informações completas de um anúncio específico.
 - **Entradas:** Identificador do anúncio.
 - **Saídas:** Página com título, descrição, preço, dados do tênis (marca, modelo, tamanho, foto) e usuário responsável.
 - **Regras de negócio:**
 - O anúncio deve existir.
 - O usuário deve estar autenticado.
-

T10 – Editar Anúncio

- **Descrição:** Alterar as informações de um anúncio existente.
 - **Entradas:** Identificador do anúncio, novos dados (título, descrição, preço, tênis vinculado).
 - **Saídas:** Dados do anúncio atualizados no banco de dados.
 - **Regras de negócio:**
 - O anúncio deve existir.
 - Apenas o usuário que criou o anúncio pode editá-lo.
 - A data de atualização deve ser ajustada automaticamente.
-

T11 – Excluir Anúncio

- **Descrição:** Remover um anúncio do sistema.
 - **Entradas:** Identificador do anúncio.
 - **Saídas:** Anúncio removido da base e das listagens.
 - **Regras de negócio:**
 - O anúncio deve existir.
 - Apenas o usuário proprietário do anúncio pode excluí-lo.
 - O sistema deve solicitar confirmação antes da exclusão.
-

4. Modelo Conceitual (MER)

Com base no minimundo e nos requisitos levantados, foram identificadas as seguintes entidades principais:

4.1. Entidade USUÁRIO

- **Descrição:** Representa os usuários que acessam o sistema e podem cadastrar tênis e anúncios.

- **Atributos:**

- `id_usuario` (chave primária)
- `nome`
- `email`
- `senha`
- `data_cadastro`

4.2. Entidade TÊNIS

- **Descrição:** Representa os produtos (tênis) cadastrados no sistema.

- **Atributos:**

- `id_tenis` (chave primária)
- `marca`
- `modelo`
- `tamanho`
- `cor`
- `tipo`
- `foto` (opcional)

4.3. Entidade ANÚNCIO

- **Descrição:** Representa os anúncios de venda criados para um tênis específico.

- **Atributos:**

- `id_anuncio` (chave primária)
- `titulo`

- `descricao`
 - `preco`
 - `data_criacao`
 - `data_atualizacao`
 - `id_usuario` (chave estrangeira → USUÁRIO)
 - `id_tenis` (chave estrangeira → TÊNIS)
-

4.4. Relacionamentos

- **USUÁRIO – ANÚNCIO**
 - Um usuário pode criar vários anúncios.
 - Cada anúncio é criado por um único usuário.
 - Cardinalidade: **USUÁRIO (1) — (N) ANÚNCIO**
- **TÊNIS – ANÚNCIO**
 - Um tênis pode aparecer em vários anúncios.
 - Cada anúncio está vinculado a um único tênis.
 - Cardinalidade: **TÊNIS (1) — (N) ANÚNCIO**

Conceitualmente, existe uma relação **N:M entre USUÁRIO e TÊNIS**, mediada pela entidade **ANÚNCIO**, pois:

- um usuário pode anunciar vários tênis;
- um tênis pode ser anunciado por diferentes usuários (em momentos distintos).

4.5. Diagrama Entidade-Relacionamento (MER)

Aqui inserir a imagem `mer.png` como Figura 1

Legenda: *Figura 1 – Diagrama Entidade-Relacionamento do sistema de loja de tênis.*

5. Modelo Lógico Relacional

A partir do modelo conceitual, obteve-se o seguinte modelo lógico relacional:

5.1. Tabela USUARIO

USUARIO(

- `id_usuario` **INT** – chave primária
- `nome` **VARCHAR(100)** – não nulo
- `email` **VARCHAR(100)** – não nulo, único
- `senha` **VARCHAR(255)** – não nulo
- `data_cadastro` **DATETIME** – pode ter valor padrão de data/hora atual

)

5.2. Tabela TENIS

TENIS(

- `id_tenis` **INT** – chave primária
- `marca` **VARCHAR(50)** – não nulo
- `modelo` **VARCHAR(100)** – não nulo
- `tamanho` **INT** – não nulo
- `cor` **VARCHAR(50)** – não nulo
- `tipo` **VARCHAR(50)** – não nulo
- `foto` **VARCHAR(255)** – opcional

)

5.3. Tabela ANUNCIO

ANUNCIO(

- `id_anuncio` **INT** – chave primária
- `titulo` **VARCHAR(100)** – não nulo
- `descricao` **TEXT** – não nulo
- `preco` **DECIMAL(10,2)** – não nulo
- `data_criacao` **DATETIME** – valor padrão: data/hora atual
- `data_atualizacao` **DATETIME** – atualizado a cada alteração
- `id_usuario` **INT** – chave estrangeira → `USUARIO(id_usuario)`
- `id_tenis` **INT** – chave estrangeira → `TENIS(id_tenis)`

)

5.4. Normalização

As tabelas foram construídas de forma a atender às principais **formas normais**:

- **Primeira Forma Normal (1FN):**
 - Todos os atributos são atômicos.
 - Não existem grupos repetidos ou atributos multivalorados.
- **Segunda Forma Normal (2FN):**
 - Não há chaves primárias compostas nas tabelas propostas.
 - Logo, não existem dependências parciais de parte da chave.
- **Terceira Forma Normal (3FN):**
 - Todos os atributos não-chave dependem diretamente da chave primária da tabela.
 - Não há dependências transitivas entre atributos não-chave (por exemplo, em `ANUNCIO`, o preço não depende de outro atributo que não a própria chave

do anúncio).

6. Projeto Físico / Implementação em SQL (DDL)

A implementação física foi realizada no SGBD **SQLite**, utilizando o framework Django para geração das tabelas via migrations. A seguir, são apresentados comandos SQL que representam a criação das tabelas principais:

```
CREATE TABLE USUARIO (  
    id_usuario      INTEGER PRIMARY KEY AUTOINCREMENT,  
    nome            VARCHAR(100) NOT NULL,  
    email           VARCHAR(100) NOT NULL UNIQUE,  
    senha           VARCHAR(255) NOT NULL,  
    data_cadastro   DATETIME DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE TENIS (  
    id_tenis        INTEGER PRIMARY KEY AUTOINCREMENT,  
    marca           VARCHAR(50) NOT NULL,  
    modelo          VARCHAR(100) NOT NULL,  
    tamanho         INTEGER NOT NULL,  
    cor             VARCHAR(50) NOT NULL,  
    tipo            VARCHAR(50) NOT NULL,  
    foto            VARCHAR(255)  
);  
  
CREATE TABLE ANUNCIO (  
    id_anuncio      INTEGER PRIMARY KEY AUTOINCREMENT,  
    titulo           VARCHAR(100) NOT NULL,  
    descricao        TEXT NOT NULL,  
    preco            DECIMAL(10,2) NOT NULL,  
    data_criacao     DATETIME DEFAULT CURRENT_TIMESTAMP,  
    data_atualizacao DATETIME DEFAULT CURRENT_TIMESTAMP,  
    id_usuario       INTEGER NOT NULL,  
    id_tenis         INTEGER NOT NULL,  
    FOREIGN KEY (id_usuario) REFERENCES USUARIO(id_usuario),  
    FOREIGN KEY (id_tenis)   REFERENCES TENIS(id_tenis)  
);
```

Observação: no projeto real, as tabelas são criadas e gerenciadas automaticamente a partir das classes definidas em `models.py` (Django), por meio de comandos de migração (`python manage.py makemigrations` e `python manage.py migrate`), que geram o arquivo `db.sqlite3`.

7. Modelagem Orientada a Objetos (Diagrama de Classes)

A modelagem orientada a objetos foi realizada com base nas entidades identificadas no modelo de dados, resultando nas seguintes classes conceituais:

- **Classe Usuario**
 - Atributos: `id`, `nome`, `email`, `senha`, `data_cadastro`
 - Métodos (conceituais): `autenticar()`, `cadastrar()`, `atualizarDados()`
- **Classe Tennis**
 - Atributos: `id`, `marca`, `modelo`, `tamanho`, `cor`, `tipo`, `foto`
 - Métodos: `cadastrar()`, `atualizar()`, `remover()`
- **Classe Anuncio**
 - Atributos: `id`, `titulo`, `descricao`, `preco`, `data_criacao`, `data_atualizacao`, `usuario`, `tenis`
 - Métodos: `criar()`, `editar()`, `excluir()`

Os relacionamentos entre as classes seguem o mesmo padrão do modelo de dados:

- Um **Usuario** possui vários **Anuncio** (associação 1:N).
- Um **Tennis** possui vários **Anuncio** (associação 1:N).
- Cada **Anuncio** está associado a exatamente um **Usuario** e um **Tennis**.

8. Aplicação CRUD (Django + SQLite)

A aplicação foi desenvolvida utilizando o framework **Django**, com o banco de dados padrão **SQLite**. O sistema está organizado em módulos principais: autenticação, gestão de tênis e gestão de anúncios.

8.1. Funcionalidades Implementadas

- **Autenticação**
 - Tela de login para acesso ao sistema.
 - Logout para encerramento da sessão.
 - Controle de acesso às funcionalidades internas (somente usuários autenticados).
- **Gestão de Tênis**
 - Cadastro de novos tênis.
 - Listagem de todos os tênis cadastrados.
 - Edição dos dados de um tênis.
 - Exclusão de tênis, com confirmação.
- **Gestão de Anúncios**
 - Cadastro de anúncios vinculados a um tênis já existente.
 - Listagem dos anúncios do usuário logado.
 - Listagem de todos os anúncios cadastrados no sistema.
 - Visualização de detalhes de um anúncio.
 - Edição dos dados do anúncio (título, descrição, preço).
 - Exclusão de anúncios, com confirmação.