

Documentação do Sistema E-commerce

1. Modelo Entidade-Relacionamento (MER)

1.1 Entidades e Atributos

1. **Usuário**
 - id (PK)
 - nome
 - email
 - senha
 - data_cadastro
2. **Anúncio**
 - id (PK)
 - titulo
 - descricao
 - preco
 - data_publicacao
 - usuario_id (FK)
 - categoria_id (FK)
3. **Categoria**
 - id (PK)
 - nome
 - descricao
4. **Pergunta**
 - id (PK)
 - texto
 - data_pergunta
 - usuario_id (FK)
 - anuncio_id (FK)
5. **Resposta**
 - id (PK)
 - texto
 - data_resposta
 - pergunta_id (FK)
6. **Compra**
 - id (PK)
 - data_compra
 - valor_total
 - usuario_id (FK)
 - anuncio_id (FK)
7. **Favorito**
 - id (PK)
 - data_adicao
 - usuario_id (FK)
 - anuncio_id (FK)

1.2 Relacionamentos

- Usuário **possui** vários Anúncios (1:N)
- Usuário **faz** várias Perguntas (1:N)
- Usuário **realiza** várias Compras (1:N)
- Usuário **adiciona** vários Favoritos (1:N)
- Anúncio **pertence** a uma Categoria (N:1)
- Anúncio **recebe** várias Perguntas (1:N)
- Pergunta **tem** uma Resposta (1:1)
- Anúncio **é vendido em** várias Compras (1:N)
- Anúncio **é favoritado em** vários Favoritos (1:N)

2. Diagrama de Navegação

2.1 Páginas Públicas

- **Início** (/)
 - Visualização de produtos em destaque
 - Acesso ao login/cadastro
 - Busca de produtos
- **Produtos** (/produtos)
 - Listagem de produtos
 - Filtro por categoria
 - Detalhes do produto
- **Login** (/entrar)
 - Formulário de login
 - Link para cadastro
- **Cadastro** (/cadastro)
 - Formulário de cadastro
 - Link para login

2.2 Área do Usuário (Requer Autenticação)

- **Meus Anúncios** (/usuario/anuncios)
 - Listar anúncios
 - Criar novo anúncio
 - Editar anúncio
 - Excluir anúncio
 - Ver perguntas
- **Minhas Compras** (/usuario/compras)
 - Histórico de compras
 - Detalhes das compras
- **Minhas Vendas** (/usuario/vendas)
 - Relatório de vendas
 - Status das vendas
- **Favoritos** (/usuario/favoritos)
 - Lista de produtos favoritos

– Remover dos favoritos

3. Rotas Implementadas

```
@app.route('/')
def home():
    return render_template('inicio.html')

@app.route('/entrar', methods=['GET', 'POST'])
def login():
    return render_template('login.html')

@app.route('/cadastro', methods=['GET', 'POST'])
def register():
    return render_template('cadastro.html')

@app.route('/produtos')
def products():
    return render_template('produtos.html')

@app.route('/usuario/anuncios')
@login_required
def user_ads():
    return render_template('usuario_anuncios.html')

@app.route('/usuario/compras')
@login_required
def user_purchases():
    return render_template('usuario_compras.html')

@app.route('/usuario/vendas')
@login_required
def user_sales():
    return render_template('usuario_vendas.html')

@app.route('/usuario/favoritos')
@login_required
def user_favorites():
    return render_template('usuario_favoritos.html')

@app.route('/sair')
@login_required
def logout():
    logout_user()
    return redirect(url_for('home'))
```

4. Instruções para Publicação no GitHub

1. Criar um novo repositório no GitHub

- Acesse github.com
- Clique em “New repository”
- Nome: trabalho_frameworks
- Descrição: Sistema de e-commerce desenvolvido com Flask
- Deixe o repositório público
- Não inicialize com README

2. Publicar o código

```
# Na pasta do projeto
git init
git add .
git commit -m "Versão inicial do projeto"
git branch -M main
git remote add origin https://github.com/seu-usuario/trabalho_frameworks.git
git push -u origin main
```

5. Justificativa das Escolhas Técnicas

1. Framework Flask

- Simplicidade e flexibilidade
- Ideal para projetos de médio porte
- Fácil integração com extensões

2. SQLAlchemy como ORM

- Abstração do banco de dados
- Mapeamento objeto-relacional robusto
- Facilidade na manutenção

3. Flask-Login para Autenticação

- Gerenciamento de sessões seguro
- Decoradores para proteção de rotas
- Fácil implementação

4. Bootstrap 5 para Frontend

- Framework CSS responsivo
- Componentes prontos para uso
- Boa experiência do usuário