

Objetivo Principal		Objetivos Específicos		Requisitos Funcionais	
Requisitos Não Funcionais		Modelo de Dados		Funcionalidades	
Interface	Tecnologias	Considerações Futuras		Modelo Físico	Modelo
	ER	RafaelaZulu25 u{5l8WZ_[4h`			

# Documentação do Sistema de Registros de Filmes Tarantino

## 1. Objetivo Principal do Sistema

O objetivo principal do sistema é fornecer uma plataforma para usuários registrarem e gerenciarem informações sobre filmes, com foco específico nos filmes de Quentin Tarantino.

## 2. Objetivos Específicos

### 2.1. Gerenciamento de usuários

- Permitir que novos usuários se registrem no sistema
- Autenticar usuários existentes através de um processo de login
- Gerenciar sessões de usuários, incluindo logout

### 2.2. Cadastro e Gerenciamento de Filmes

- Permitir que usuários autenticados adicionem novos filmes ao sistema
- Possibilidade de edição de informações de filmes existentes
- Permitir a exclusão de filmes do sistema

### 2.3. Visualização de Filmes

- Exibir uma lista de filmes cadastrados pelo usuário logado
- Ver detalhes de cada filme, como nome, ano de lançamento, diretor e sinopse

### 2.4. Personalização de Cadastros

- Associar filmes a usuários específicos através do modelo Cadastro
- Permitir que os usuários adicionem descrições específicas aos filmes cadastrados

### 2.5. Feedback ao Usuário

- Fornecer mensagens de feedback para ações como login, registro, adição, edição e exclusão de filmes

## 2.6. Segurança

- Implementar controle de acesso, garantindo que apenas usuários autenticados possam realizar operações no sistema

## 2.7. Interface Amigável

- Oferecer uma interface intuitiva e responsiva para interação com o sistema

## 2.8. Organização de Dados

- Manter uma estrutura organizada de dados, relacionando usuários, filmes e cadastros

# 3. Requisitos Funcionais (RF)

- RF01 - Registro de Usuário
- RF02 - Login
- RF03 - Logout
- RF04 - Adicionar Filme
- RF05 - Visualizar Filmes
- RF06 - Editar Filme
- RF07 - Excluir Filme
- RF08 - Mensagens de Feedback

# 4. Requisitos Não Funcionais (RNF)

- RNF01 - Desempenho
- RNF02 - Segurança
- RNF03 - Usabilidade
- RNF04 - Compatibilidade
- RNF05 - Manutenibilidade

# 5. Modelo de Dados

## 5.1. Entidades Principais

- Usuário (modelo de usuário integrado do Django)

- Filme
- Cadastro

## 5.2. Relacionamentos

- Usuário tem muitos Cadastros (um-para-muitos)
- Filme tem muitos Cadastros (um-para-muitos)
- Cadastro pertence a um usuário e a um filme (muitos-para-um para ambos)

## 6. Principais Funcionalidades Implementadas

- 6.1. Autenticação de Usuários
- 6.2. Registro de Novos Usuários
- 6.3. Adição de Filmes
- 6.4. Edição de Filmes
- 6.5. Exclusão de Filmes
- 6.6. Visualização de Filmes do Usuário

## 7. Interface do Usuário

### 7.1. Telas Principais

- Tela de Login
- Tela de Registro
- Página Inicial (Home)
- Tela de Adição de Filme
- Tela de Edição de Filme

### 7.2. Elementos de Design

- Uso de cores escuras com destaque em amarelo para elementos importantes
- Imagens de fundo relacionadas a filmes de Tarantino
- Layout responsivo para diferentes tamanhos de tela

## 8. Tecnologias Utilizadas

- Backend: Django (Python)
- Frontend: HTML, CSS, JavaScript
- Banco de Dados: SQLite (desenvolvimento) / PostgreSQL (produção)

## 9. Considerações Futuras

- Implementação de um sistema de busca de filmes
- Adição de uma funcionalidade de avaliação de filmes
- Integração com APIs externas para obter informações adicionais sobre os filmes

## 10. Modelo Físico do Banco de Dados

### 10.1 Modelo Físico do Banco de Dados (SQL Lite)

```
-- Tabela de Usuários (built-in do Django)
CREATE TABLE auth_user (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    password VARCHAR(128) NOT NULL,
    last_login DATETIME,
    is_superuser BOOLEAN NOT NULL,
    username VARCHAR(150) NOT NULL UNIQUE,
    first_name VARCHAR(30) NOT NULL,
    last_name VARCHAR(150) NOT NULL,
    email VARCHAR(254) NOT NULL,
    is_staff BOOLEAN NOT NULL,
    is_active BOOLEAN NOT NULL,
    date_joined DATETIME NOT NULL
);

-- Tabela de Filmes
CREATE TABLE filme (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nome VARCHAR(255) NOT NULL,
    ano_lancamento INTEGER NOT NULL,
    diretor VARCHAR(255) NOT NULL,
    sinopse TEXT
);

-- Tabela de Cadastro (relação entre Usuário e Filme)
CREATE TABLE cadastro (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL,
    filme_id INTEGER NOT NULL,
    descricao TEXT,
    FOREIGN KEY (user_id) REFERENCES auth_user (id),
    FOREIGN KEY (filme_id) REFERENCES filme (id)
);
```

### 10.2 Diagrama de Relacionamento

User		Cadastro		Filme	
id (PK)		id (PK)		id (PK)	
username		user_id (FK)		nome	
email		filme_id (FK)		ano_lancamento	
password		descricao		diretor	
first_name				sinopse	
last_name					
is_staff					
is_active					
date_joined					
last_login					

### 10.3 Geração de Diagramas

Para gerar diagramas do modelo de dados, você pode usar as seguintes ferramentas e comandos:

```
pip install django-extensions
pip install graphviz
pip install pygraphviz
python manage.py graph_models -a -o myapp_models.png
pip install django-schema-graph
python manage.py runserver
```

Após instalar o django-schema-graph, você pode acessar o diagrama do esquema em /schema/ na sua aplicação em execução.

## 11. Modelo de Entidade-Relacionamento (ER)

### 11.1 Descrição do Diagrama ER

O sistema de registro de filmes Tarantino possui as seguintes entidades e relações:

**Entidades:**

- **Usuário (auth\_user):** id (PK), username, email, password, first\_name, last\_name, is\_staff, is\_active, date\_joined, last\_login
- **Filme:** id (PK), nome, ano\_lancamento, diretor, sinopse
- **Cadastro:** id (PK), user\_id (FK para Usuário), filme\_id (FK para Filme), descrição

Relações:

- Um usuário pode cadastrar vários filmes (relação um-para-muitos entre Usuário e Cadastro).
- Um filme pode ser cadastrado por vários usuários (relação um-para-muitos entre Filme e Cadastro).

11.2 Diagrama ER em Texto

+-----+		+-----+		+-----+
User		Cadastro		Filme
+-----+		+-----+		+-----+
id (PK)	<-----	id (PK)	----->	id (PK)
username		user_id (FK)		nome
email		filme_id (FK)		ano_lancamento
password		descricao		diretor
first_name		+-----+		sinopse
last_name				+-----+
is_staff				
is_active				
date_joined				
last_login				
+-----+				

11.3 Código PlantUML para Gerar o Diagrama ER

```
@startuml
entity "User" as user {
    + id : int <<PK>>
    --
    username : varchar
    email : varchar
    password : varchar
    first_name : varchar
    last_name : varchar
    is_staff : boolean
    is_active : boolean
    date_joined : datetime
    last_login : datetime
}

entity "Filme" as filme {
    + id : int <<PK>>
    --
    nome : varchar
    ano_lancamento : int
}
```

```

    diretor : varchar
    sinopse : text
}

entity "Cadastro" as cadastro {
    + id : int <<PK>>
    --
    user_id : int <<FK>>
    filme_id : int <<FK>>
    descricao : text
}

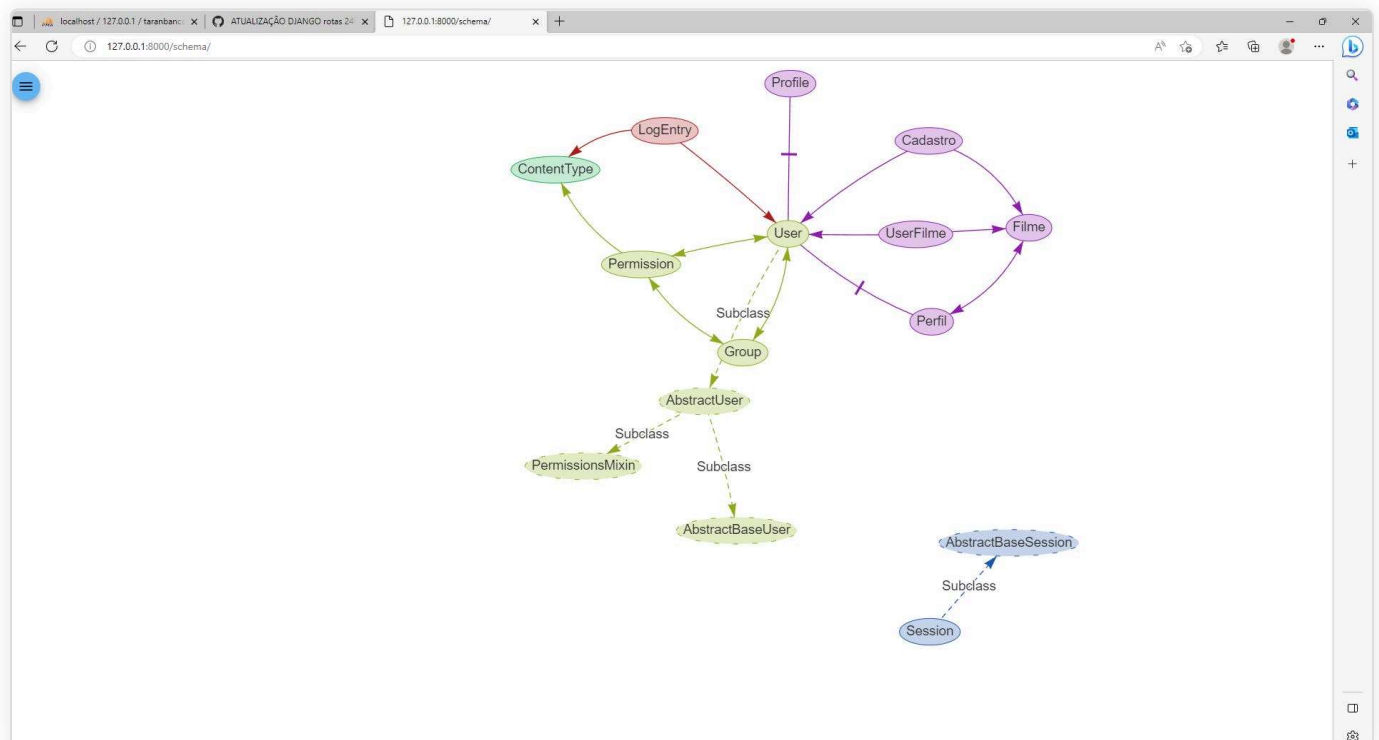
user ||--o{ cadastro : "cadastra"
filme ||--o{ cadastro : "foi cadastrado"
@enduml

```

Para gerar a imagem do diagrama ER, você pode copiar o código acima e colá-lo em um editor PlantUML online.

## 12. Diagramas Visuais

### Diagrama de Relacionamento



### Diagrama ER

