

# Resumo Completo e Integrado do Sistema de Impressões Acadêmicas

O Sistema de Impressões Acadêmicas é uma aplicação web desenvolvida com Django para gerenciar pedidos de impressão feitos por professores e administrados por usuários do tipo administrador. Ele possui controle de autenticação, permissões, upload de arquivos, detalhamento de pedidos, fluxo de aprovação e um painel específico para cada tipo de usuário.

## Funcionamento Geral

O login é realizado pela tela inicial, onde o usuário informa nome e senha cadastrados. Tanto professores quanto administradores utilizam o mesmo formulário, porém são redirecionados a painéis diferentes após a verificação do tipo de usuário. Para cadastrar um novo usuário, basta acessar a opção “Registrar” ou “Cadastro”, preencher os dados e definir o tipo (professor ou admin, conforme permitido).

Professores enviam pedidos de impressão por meio do botão “Nova Impressão” ou “Criar Pedido”, onde preenchem os campos exigidos e fazem upload dos arquivos. Esses arquivos são salvos na pasta `uploads/` e gerenciados pelo modelo `ArquivoPedido`. Professores também podem editar ou excluir pedidos, desde que estejam com status “pendente”. Eles visualizam exclusivamente os pedidos criados por eles, graças ao controle de permissões baseado no campo `tipo` do usuário.

Administradores possuem permissão total: podem ver todos os pedidos, detalhar, aprovar, rejeitar e excluir qualquer solicitação. Ao detalhar um pedido como administrador, são exibidos botões de ação; já o professor acessa apenas a visualização, sem opções administrativas.

## Controle de Permissões e Segurança

O sistema garante que cada usuário veja apenas o que lhe corresponde, usando verificações nas views e nos templates. Caso alguém tente acessar uma URL que pertence a outro tipo de usuário, o sistema redireciona automaticamente e exibe uma mensagem de acesso restrito. Os painéis (professor e admin) mostram estatísticas personalizadas — como total de pedidos, pendentes, rejeitados e entregues — de acordo com o tipo de usuário.

Mensagens de feedback são exibidas usando o sistema de mensagens do Django e incluídas nos templates através do arquivo `messages.html`. O layout é responsivo, permitindo uso em computador, tablet ou celular. O logout é realizado pelo botão “Sair” no menu superior.

A busca e a filtragem dos pedidos estão disponíveis tanto para professores quanto para administradores. A paginação é feita com o [Paginator](#) do Django, organizada para facilitar a navegação.

## Modelagem dos Dados

A base do sistema é formada por dois modelos principais:

- [PedidoDeImpressao](#): armazena todos os dados do pedido, incluindo quantidade, tipo, observações, status (pendente, aprovado, rejeitado, entregue) e o usuário que fez o pedido.
- [ArquivoPedido](#): relaciona os arquivos enviados a cada pedido, armazenando-os automaticamente na pasta [uploads/](#).

Cada usuário é relacionado aos pedidos por meio de uma ForeignKey, e o tipo do usuário (definido no modelo de usuário customizado) determina as permissões no sistema.

## Fluxo de Autenticação

Após o login, o sistema identifica o tipo do usuário e o redireciona para o painel correspondente. O logout retorna para a tela de login. Esse fluxo garante organização, segurança e separação clara de funcionalidades entre os perfis.

## Estrutura do Projeto

O projeto possui a seguinte organização:

### Raiz do Projeto

- **manage.py**: arquivo padrão para executar comandos (migrations, servidor, criação de superusuário).
- **db.sqlite3**: banco de dados principal.
- **requirements.txt**: lista de dependências.
- **Dockerfile / docker-compose.yml**: arquivos de containerização para padronizar ambiente e facilitar deploy.
- **uploads/**: pasta onde ficam os arquivos enviados.
- **prints/**: imagens utilizadas na documentação.

## Pasta config/

Contém configurações globais:

- **settings.py**: configurações do Django, apps instalados, banco de dados e templates.
- **urls.py**: rotas principais, incluindo as dos apps **usuarios** e **impressoes**.
- **asgi.py / wsgi.py**: arquivos para servidores ASGI/WSGI.
- **init.py**: inicializador do pacote.

## Pasta usuarios/

Gerencia autenticação e tipos de usuário:

- **models.py**: modelo de usuário customizado com campo **tipo**.
- **forms.py**: formulários de login e registro.
- **views.py**: lógica de login, logout, registro e redirecionamento por tipo de usuário.
- **urls.py**: rotas das páginas de usuários.
- **admin.py**: configuração do modelo de usuário no Django Admin.
- **templates/usuarios/**: telas de login, registro e painel admin.

## Pasta impressoes/

Gerencia todo o fluxo de pedidos:

- **models.py**: modelos **PedidoDeImpressao** e **ArquivoPedido**.
- **forms.py**: formulários para criação/edição de pedidos.
- **views.py**: lógica de criação, edição, exclusão, aprovação e listagem dos pedidos, com controle de permissões.
- **urls.py**: rotas específicas do app.
- **templates/impressoes/**:
  - **base.html**: template base com header e mensagens.

- **painel\_admin.html**: painel de administração com todas as ações.
- **painel\_professor.html**: painel restrito ao professor.
- **criar\_pedido.html, editar\_pedido.html**: formulários.
- **detalhar\_pedido\_admin.html e detalhar\_pedido\_professor.html**: visualização dos detalhes.
- **includes/header.html**: menu superior com opções dinâmicas.
- **includes/messages.html**: mensagens de feedback.
- **login.html e index.html**: telas gerais.

## Docker e Deploy

O projeto conta com:

- **Dockerfile** para gerar a imagem do sistema.
- **docker-compose.yml** para subir o ambiente facilmente.
- **workflow do GitHub Actions** que gera builds automáticos da imagem.

## Resumo Funcional Final

O sistema permite que professores criem e acompanhem pedidos de impressão; administradores analisam, aprovam, rejeitam e gerenciam o fluxo completo. Há segurança por tipo de usuário, organização visual clara, controle de arquivos, busca, filtros, responsividade, mensagens de feedback e uma estrutura bem dividida entre autenticação e gerenciamento de pedidos.

### 1. Como faço login no sistema?

O usuário acessa a tela de login e informa seu nome e senha cadastrados. Professores e administradores utilizam o mesmo formulário, mas são direcionados para painéis diferentes de acordo com o tipo de usuário.

### 2. Como faço para cadastrar um novo usuário?

Na tela inicial, basta clicar em “Registrar” ou “Cadastro”, preencher os dados solicitados e selecionar o tipo de usuário (professor ou administrador).

### 3. Como envio um pedido de impressão?

No painel do professor, clique em “Nova Impressão” ou “Criar Pedido”, preencha os campos obrigatórios, faça upload dos arquivos e envie o pedido.

**4. Posso editar ou excluir um pedido já enviado?**

Sim. Desde que o pedido esteja com status “pendente”, o professor pode editar ou excluir usando os botões correspondentes ao lado do pedido.

**5. Como aprovo ou rejeito pedidos?**

Administradores acessam o painel admin, clicam em “Detalhar” no pedido e utilizam os botões “Aprovar” ou “Rejeitar”.

**6. O sistema garante que cada usuário só vê o que deve?**

Sim. Professores veem apenas seus próprios pedidos. Administradores veem todos. Isso é controlado pelo campo `tipo` e pela lógica das views.

**7. Como funciona o controle de permissões entre professor e administrador?**

O campo `tipo` diferencia o perfil. As views verificam esse campo e entregam apenas o conteúdo adequado. O admin tem controle total; o professor controla apenas seus pedidos.

**8. O que acontece se um usuário tentar acessar uma URL de outro perfil?**

O sistema identifica o tipo do usuário e redireciona automaticamente para o painel correto, exibindo uma mensagem de acesso restrito.

**9. Como funciona o upload e armazenamento dos arquivos?**

Os arquivos enviados nos pedidos são geridos pelo modelo `ArquivoPedido` e salvos na pasta `uploads/`. O FileField do Django cuida dos caminhos e gerenciamento.

**10. Como é feita a paginação e busca dos pedidos?**

Com o `Paginator` do Django, que divide os pedidos por páginas. Há também campos de busca e filtros por status para facilitar a navegação.

**11. Como são exibidas as mensagens de feedback?**

Usando o sistema de mensagens do Django, incluídas no template com `messages.html`.

**12. O sistema funciona em celular?**

Sim. O layout é completamente responsivo.

**13. Como faço logout?**

Basta clicar em “Sair” no menu superior.

**14. Como funciona o detalhamento de pedidos para cada tipo de usuário?**

O administrador vê os detalhes com botões de ação (aprovar, rejeitar, excluir). O professor vê apenas os detalhes, sem ações administrativas.

**15. Como foi feita a modelagem dos dados?**

Com dois modelos principais: `PedidoDeImpressao` (dados do pedido) e `ArquivoPedido` (arquivos do pedido). O usuário é relacionado ao pedido via ForeignKey.

**16. Como funciona o fluxo de autenticação e redirecionamento?**

Após login, o sistema verifica o tipo e redireciona para painel admin ou professor. Logout volta para a tela de login.

**17. O sistema permite busca e filtros?**

Sim, em ambos os painéis: professor e administrador.

**18. Como vejo estatísticas dos pedidos?**

Nos painéis há contadores exibindo totais, pendentes, entregues e rejeitados — personalizados conforme o tipo de usuário.