📦 Documentação Técnica - Bloco 4: Painel do Usuário com Sessão (Zenith IA)

# 🎯 Objetivo

Implementar o painel do usuário com autenticação por sessão, inspirado na experiência do Jotform, utilizando o pacote Flask-Login. O objetivo é garantir que apenas usuários autenticados acessem o painel, exibindo nome, e-mail e plano atual.

# ✅ Etapas realizadas

## 1. Criação do arquivo models.py

Criado arquivo app/models.py com a classe Usuario, herdando de UserMixin, contendo os atributos:  
- id, nome, email, plano  
  
Essa classe representa o usuário autenticado na sessão Flask-Login.

## 2. Configuração do Flask-Login no \_\_init\_\_.py

LoginManager instanciado, init\_app() aplicado e login\_view configurado.  
Definida função load\_user para carregar o usuário do banco e retornar objeto Usuario.

## 3. Atualização da rota /login-web com login\_user()

Adicionado uso de login\_user() para autenticar e iniciar sessão do usuário após validação da senha.  
Redirecionamento para /painel em caso de sucesso.

## 4. Criação da rota protegida /painel com @login\_required

Rota criada com proteção via @login\_required.  
Usa current\_user para acessar nome, email e plano do usuário.

## 5. Criação do template painel.html

HTML com Bootstrap exibindo informações básicas do usuário autenticado.  
Inclui botão de logout vinculado à rota /logout.

## 6. Criação da rota /logout

Função logout criada usando logout\_user() e redirecionando para /login após encerrar sessão.

# ✅ Resultado Final

• Usuário acessa /login e realiza autenticação.  
• Após o login, é redirecionado para /painel com sessão ativa.  
• Os dados do usuário são exibidos.  
• Ao clicar em “Sair”, a sessão é encerrada e ele volta para /login.  
• Tentativas de acessar /painel sem login são bloqueadas.

# 🔐 Tecnologias utilizadas

- Flask  
- Flask-Login  
- werkzeug.security  
- MySQL  
- Bootstrap 5