

```
from flask import render_template, request, redirect, url_for, flash
from app import app, db
from app.models import Acao, Responsavel, Origem, Superintendencia, Usuario
from datetime import date, datetime
from werkzeug.security import generate_password_hash
```

```
@app.route('/login', methods=['GET', 'POST'])
```

```
def login():
```

```
    if request.method == 'POST':
        return redirect(url_for('index'))
    return render_template('login.html')
```

```
@app.route('/')

```

```
def index():
```

```
    hoje = date.today()
    acoes = Acao.query.all()
    for acao in acoes:
        if acao.status != "Concluída" and acao.prazo < hoje and acao.status != "Atrasada":
            acao.status = "Atrasada"
            db.session.commit()
```

```
    filtro_responsavel = request.args.get('responsavel')
```

```
    filtro_superintendencia = request.args.get('superintendencia')
```

```
    filtro_origem = request.args.get('origem')
```

```
    filtro_status = request.args.get('status')
```

```
    data_inicio = request.args.get('data_inicio')
```

```
    data_fim = request.args.get('data_fim')
```

```
    query = Acao.query
```

```
    if filtro_responsavel:
```

```

        query = query.filter_by(responsavel_id=filtro_responsavel)
    if filtro_superintendencia:
        query = query.join(Responsavel).filter(Responsavel.superintendencia_id ==
        filtro_superintendencia)
    if filtro_origem:
        query = query.filter_by(origem_id=filtro_origem)
    if filtro_status:
        query = query.filter_by(status=filtro_status)
    if data_inicio:
        query = query.filter(Acao.prazo >= datetime.strptime(data_inicio, '%Y-%m-%d').date())
    if data_fim:
        query = query.filter(Acao.prazo <= datetime.strptime(data_fim, '%Y-%m-%d').date())

    acoes_filtradas = query.all()

    responsaveis = Responsavel.query.filter_by(ativo=True).all()
    superintencias = Superintendencia.query.filter_by(ativo=True).all()
    origens = Origem.query.filter_by(ativo=True).all()
    status_opcoes = ["Não iniciada", "Em andamento", "Atrasada", "Concluída"]

    responsaveis_info = [
        {"id": r.id, "nome": r.nome, "superintendencia_id": r.superintendencia_id}
        for r in responsaveis
    ]

    return render_template(
        'index.html',
        acoes=acoes_filtradas,
        responsaveis=responsaveis,
        superintencias=superintencias,
        origens=origens,

```

```
        status_opcoes=status_opcoes,  
        responsaveis_info=responsaveis_info  
    )
```

```
@app.route('/cadastrar', methods=['GET', 'POST'])
```

```
def cadastrar():
```

```
    if request.method == 'POST':  
        origem_id = request.form['origem']  
        descricao = request.form['descricao']  
        responsavel_id = request.form['responsavel']  
        prazo_str = request.form['prazo']  
        status = request.form['status']  
        prazo = datetime.strptime(prazo_str, '%Y-%m-%d').date()
```

```
        nova_acao = Acao(  
            origem_id=origem_id,  
            descricao=descricao,  
            responsavel_id=responsavel_id,  
            prazo=prazo,  
            status=status  
        )
```

```
        db.session.add(nova_acao)  
        db.session.commit()  
        return redirect(url_for('index'))
```

```
    responsaveis = Responsavel.query.filter_by(ativo=True).all()  
    origens = Origem.query.filter_by(ativo=True).all()  
    return render_template('cadastrar.html', responsaveis=responsaveis, origens=origens)
```

```
@app.route('/editar/<int:id>', methods=['GET', 'POST'])
```

```

def editar(id):

    acao = Acao.query.get_or_404(id)

    if request.method == 'POST':

        acao.origem_id = request.form['origem']

        acao.descricao = request.form['descricao']

        acao.responsavel_id = request.form['responsavel']

        acao.prazo = datetime.strptime(request.form['prazo'], '%Y-%m-%d').date()

        acao.status = request.form['status']

        if acao.status == 'Atrasada' and acao.prazo >= date.today():

            acao.status = 'Em andamento'

        db.session.commit()

        return redirect(url_for('index'))

    responsaveis = Responsavel.query.all()

    origens = Origem.query.all()

    return render_template('editar.html', acao=acao, responsaveis=responsaveis,
origens=origens)

@app.route('/responsaveis', methods=['GET', 'POST'])
def responsaveis():

    if request.method == 'POST':

        novo = Responsavel(

            nome=request.form['nome'],

            email=request.form['email'],

            telefone=request.form['telefone'],

            funcao=request.form['funcao'],

            superintendencia_id=request.form['superintendencia_id'],

            unidade=request.form['unidade'],

```

```

        tipo_acesso=request.form['tipo_acesso']
    )
    db.session.add(novo)
    db.session.commit()
    return redirect(url_for('responsaveis'))

    todos = Responsavel.query.all()
    superintendencias = Superintendencia.query.filter_by(ativo=True).all()

    return render_template('responsaveis.html', responsaveis=todos,
superintendencias=superintendencias)

@app.route('/editar_responsavel/<int:id>', methods=['GET', 'POST'])
def editar_responsavel(id):
    responsavel = Responsavel.query.get_or_404(id)
    superintendencias = Superintendencia.query.filter_by(ativo=True).all()

    if request.method == 'POST':
        responsavel.nome = request.form['nome']
        responsavel.email = request.form['email']
        responsavel.telefone = request.form['telefone']
        responsavel.funcao = request.form['funcao']
        responsavel.superintendencia_id = request.form['superintendencia_id']
        responsavel.unidade = request.form['unidade']
        responsavel.tipo_acesso = request.form['tipo_acesso']

        db.session.commit()
        return redirect(url_for('responsaveis'))

    return render_template('editar_responsavel.html', responsavel=responsavel,
superintendencias=superintendencias)

@app.route('/origens', methods=['GET', 'POST'])

```

```

def origens():
    if request.method == 'POST':
        nova = Origem(nome=request.form['nome'])
        db.session.add(nova)
        db.session.commit()
        return redirect(url_for('origens'))

    todas = Origem.query.all()
    return render_template('origens.html', origens=todas)

@app.route('/superintendencias', methods=['GET', 'POST'])
def superintendencias():
    if request.method == 'POST':
        nova = Superintendencia(nome=request.form['nome'])
        db.session.add(nova)
        db.session.commit()
        return redirect(url_for('superintendencias'))

    todas = Superintendencia.query.all()
    return render_template('superintendencias.html', superintendencias=todas)

@app.route('/responsavel/desativar/<int:id>', methods=['POST'])
def desativar_responsavel(id):
    responsavel = Responsavel.query.get_or_404(id)
    responsavel.ativo = False
    db.session.commit()
    return redirect(url_for('responsaveis'))

@app.route('/responsaveis/ativar/<int:id>', methods=['POST'])
def ativar_responsavel(id):
    responsavel = Responsavel.query.get_or_404(id)

```

```
responsavel.ativo = True

db.session.commit()

return redirect(url_for('responsaveis'))
```

```
@app.route('/origens/editar/<int:id>', methods=['GET', 'POST'])
```

```
def editar_origem(id):
```

```
    origem = Origem.query.get_or_404(id)
```

```
    if request.method == 'POST':
```

```
        origem.nome = request.form['nome']
```

```
        db.session.commit()
```

```
        return redirect(url_for('origens'))
```

```
    return render_template('editar_origem.html', origem=origem)
```

```
@app.route('/origens/desativar/<int:id>', methods=['POST'])
```

```
def desativar_origem(id):
```

```
    origem = Origem.query.get_or_404(id)
```

```
    origem.ativo = False
```

```
    db.session.commit()
```

```
    return redirect(url_for('origens'))
```

```
@app.route('/origens/ativar/<int:id>', methods=['POST'])
```

```
def ativar_origem(id):
```

```
    origem = Origem.query.get_or_404(id)
```

```
    origem.ativo = True
```

```
    db.session.commit()
```

```
    return redirect(url_for('origens'))
```

```
@app.route('/editar_superintendencia/<int:id>', methods=['GET', 'POST'])
```

```
def editar_superintendencia(id):
```

```

superintendencia = Superintendencia.query.get_or_404(id)

if request.method == 'POST':
    superintendencia.nome = request.form['nome']
    db.session.commit()
    return redirect(url_for('superintendencias'))

    return render_template('editar_superintendencia.html',
superintendencia=superintendencia)

@app.route('/desativar_superintendencia/<int:id>', methods=['POST'])
def desativar_superintendencia(id):
    superintendencia = Superintendencia.query.get_or_404(id)
    superintendencia.ativo = False
    db.session.commit()
    return redirect(url_for('superintendencias'))

@app.route('/ativar_superintendencia/<int:id>', methods=['POST'])
def ativar_superintendencia(id):
    superintendencia = Superintendencia.query.get_or_404(id)
    superintendencia.ativo = True
    db.session.commit()
    return redirect(url_for('superintendencias'))

@app.route('/esqueci_senha', methods=['GET', 'POST'])
def esqueci_senha():
    if request.method == 'POST':
        email = request.form['email']

        # Aqui você pode adicionar lógica para envio de e-mail de recuperação, se desejar

        return render_template('esqueci_senha.html', mensagem="Se o e-mail estiver cadastrado,
enviaremos as instruções.")

    return render_template('esqueci_senha.html')

```



```

@app.route('/usuarios', methods=['GET', 'POST'])
def usuarios():
    if request.method == 'POST':
        email = request.form['email']
        if Usuario.query.filter_by(email=email).first():
            flash('Este e-mail já está cadastrado. Escolha outro.', 'danger')
            usuarios = Usuario.query.all()
            return render_template('usuarios.html', usuarios=usuarios)

        novo = Usuario(
            nome=request.form['nome'],
            email=email,
            acesso_responsaveis='acesso_responsaveis' in request.form,
            editar_responsaveis='editar_responsaveis' in request.form,
            acesso_origens='acesso_origens' in request.form,
            editar_origens='editar_origens' in request.form,
            acesso_superintendencias='acesso_superintendencias' in request.form,
            editar_superintendencias='editar_superintendencias' in request.form,
            acesso_acoes='acesso_acoes' in request.form,
            editar_acoes='editar_acoes' in request.form,
        )
        novo.set_senha(request.form['senha'])
        db.session.add(novo)
        db.session.commit()
        flash('Usuário cadastrado com sucesso!', 'success')
        return redirect(url_for('usuarios'))

    usuarios = Usuario.query.all()
    return render_template('usuarios.html', usuarios=usuarios)

```

```

from app.models import Usuario # certifique-se de ter importado

```

```
from flask import flash
from werkzeug.security import generate_password_hash

@app.route('/usuarios/editar/<int:id>', methods=['GET', 'POST'])
def editar_usuario(id):
    usuario = Usuario.query.get_or_404(id)

    if request.method == 'POST':
        usuario.nome = request.form['nome']
        usuario.email = request.form['email']
        usuario.acesso_responsaveis = 'acesso_responsaveis' in request.form
        usuario.editar_responsaveis = 'editar_responsaveis' in request.form
        usuario.acesso_origens = 'acesso_origens' in request.form
        usuario.editar_origens = 'editar_origens' in request.form
        usuario.acesso_superintendencias = 'acesso_superintendencias' in request.form
        usuario.editar_superintendencias = 'editar_superintendencias' in request.form
        usuario.acesso_acoes = 'acesso_acoes' in request.form
        usuario.editar_acoes = 'editar_acoes' in request.form
        usuario.ativo = 'ativo' in request.form

        nova_senha = request.form.get('nova_senha')
        if nova_senha:
            usuario.senha_hash = generate_password_hash(nova_senha)

    db.session.commit()

    flash('Usuário atualizado com sucesso!', 'success')

    return redirect(url_for('usuarios'))

return render_template('editar_usuario.html', usuario=usuario)
```