Sistema de Recursos Humanos TJ-GO

**31 de Março de 2025**

# Visão geral

Estou atualmente alocada na Diretoria de Inteligência Artificial , Ciência de Dados e Estatística, e fico com a parte de analisar dados e criar dashboards em Power BI. Para esse trabalho tive a intenção de melhorar meus conhecimentos envolvendo a integração entre formulários e páginas web com um dashboard feito em plotly que é uma biblioteca que nunca usei antes. Então essa aplicação foi criada usando dados fictícios simulando os dados de servidores e magistrados do Tribunal de Justiça do Estado de Goiás e teve a intenção de atingir os objetivos abaixo citados visando uma demanda futura que terei na minha área que atuo no TJGO.

# Como rodar a aplicação

Para garantir o funcionamento correto das bibliotecas , foi criado o arquivo requirements.txt onde possui todas as versões de bibliotecas e python utilizadas por quem desenvolveu, eu hehe. Então sugiro para um bom funcionamento a criação de um ambiente virtual (.venv):

python -m venv .venv

.venv\Scripts\activate

pip install -r requirements.txt

Depois rodar o app.py e clicar no link que aparecer no terminal. Aí você será redirecionado para a página web.

Para um bom funcionamento das dashboards é necessário ter as tabelas criadas (servidores e magistrados) . Já deixarei criadas na pasta zip e no github.

# Objetivos

1. **Formulários:** Desenvolver formulários para a captação de dados de servidores e magistrados e inseri-los em um banco de dados levando em conta a situação profissional atual. Toda vez que o servidor se encontra em uma nova situação profissional, é necessário preencher o formulário.
2. **Importar CSV:** Para casos onde já existe uma planilha com as respectivas colunas de cada tabela preenchida em um csv, há a opção de importar esse arquivo e inseri-lo no banco de dados.
3. **Dashboards**: Os dashboards possuem as apresentações de alguns desses dados em formato de tabela e de gráfico.

# Público alvo

Servidores responsáveis por preencher o banco com as situações de cada servidor/magistrado e analistas que precisam visualizar esses dados para alguma finalidade.

# Descrição do banco de dados/aplicação

1. São usadas duas tabelas : Servidores e Magistrados. Servidores possui as colunas : nome,cpf,email,cargo,status,telefone,sexo,naturalidade,raca\_cor,identidade\_genero,aprovado\_cotas,deficiencia,data\_nascimento,data\_posse,orgao\_lotacao,situacao\_profissional,data\_inicio\_situacao,data\_saida\_situacao. Magistrado possui as mesmas colunas e mais 2 : promocao, data\_promocao. A descrição desses dados ficaria muito comprida no documento mas existe uma tabela excel dentro da pasta “Tabelas”.
2. Aplicativo App.py : esse arquivo python implementa um sistema Flask para gerenciar dados de servidores e magistrados, permitindo o cadastro, importação de dados via CSV e geração de gráficos interativos. Utiliza SQLite para armazenar os dados e Plotly para criar visualizações sobre cargos, sexo, raça/cor, situação profissional e outros atributos dos servidores e magistrados. Além disso, inclui formulários para cadastro e funcionalidades de upload de CSVs para importar dado.
3. Arquivo forms.py: Esse código define dois formulários em Flask-WTF para o cadastro de servidores e magistrados, com diversos campos obrigatórios como nome, CPF, e-mail, cargo, status, e dados de identidade, além de validações e opções de escolha (select e múltiplas escolhas). Ambos os formulários são usados para coletar informações detalhadas sobre servidores e magistrados, com validações específicas para cada campo.
4. Arquivo Sistema\_RH\_TJGO.py : O código define classes em Python para registrar dados de servidores e magistrados em um banco SQLite, com atributos como nome, CPF e cargo, e métodos para salvar os dados nas tabelas correspondentes. A classe Magistrados herda da classe Servidores, adicionando atributos específicos, como informações de promoção. A classe AnalisadorDados carrega os dados do banco e utiliza pandas para facilitar a análise, mas contém um erro de digitação nos construtores das classes que precisa ser corrigido.
5. Pasta Templates: possui todos os html’s utilizados na aplicação
   1. Base\_Magistrado e Base\_Servidor: possui a estrutura da página dos formulários.
   2. Base: possui cabeçalho e rodapé do site.
   3. Dashboard\_Magistrado e Dashboard: possui a estrutura da página dos dashboards com as tabelas e gráficos.
   4. Formulario\_enviado: página que aparece quando o formulário foi adicionado no banco corretamente.
   5. Home: Estrutura da página principal do site.
   6. Importar\_CSV\_magistrados e Importar\_CSV : arquivos com as estruturas de importar csv, tanto dos servidores quanto dos magistrados.

# Link github:

# <https://github.com/laysfreitas/Projeto-Final-de-Curso---IP-Resid-ncia.git>