

UNIVERSIDADE DO MINHO
LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

Projeto - Bases Jurídicas

Diogo Aires (a91685)

Fernanda Dias (a73169)

Jorge da Silva (a80931)

30 de maio de 2023



Resumo

Este relatório retrata todo o processo de desenvolvimento do projecto e as diferentes decisões tomadas ao longo do mesmo. O projeto tem como objetivo criar um *modelo normalizado* para as *bases jurídicas* disponíveis e uma pequena *aplicação web* para consulta e pesquisa dessa informação. No momento, existem 14 bases jurídicas envolvidas no projeto. Para atingir esse objetivo, foram utilizadas as tecnologias *Json Server* ¹ e *Flask* ². O *modelo normalizado* criado permitirá a *consulta* e *pesquisa* mais eficiente dos dados jurídicos disponíveis, enquanto a aplicação web fornecerá uma interface amigável para o usuário final. Em resumo, o projeto visa melhorar a acessibilidade e a *eficiência da pesquisa* jurídica por meio de um *modelo normalizado* e uma aplicação web amigável.

Palavras-Chave: *Modelo normalizado, bases jurídicas, Json Server, Flask, aplicação web, consulta de informação, pesquisa jurídica, normalização de dados, integridade de dados, decomposição em tabelas, eficiência de pesquisa.*

¹<https://www.npmjs.com/package/json-server>

²<https://flask.palletsprojects.com/en/2.3.x/>

Índice

1	Introdução	1
2	Análise de dados	2
2.1	Desenvolvimento do programa em Python para extração e análise dos dados	2
2.2	Utilização de requests em Python para obtenção das informações necessárias	3
3	Desenvolvimento WEB	4
3.1	Aplicação	4
3.1.1	Interface	4
	<i>Homepage</i>	5
	<i>Acórdãos</i>	5
	<i>Informações</i>	6
3.1.2	Construção da Aplicação <i>WEB</i>	7
	Criação da <i>Homepage</i>	7
	Criação da <i>Acórdãos</i>	8
	Criação da <i>Informações</i>	10
4	Conclusão	11

Capítulo 1

Introdução

O objetivo deste projeto foi criar uma aplicação web para consulta e pesquisa de informações jurídicas. Para isso, foi necessário analisar diversas bases jurídicas e identificar os campos em comum. Utilizamos o Python para desenvolver programas que nos ajudaram nessa tarefa e aprendemos a fazer requisições HTTP para acessar essas bases. Com isso, criamos um único dataset utilizando o JSON Server, o que simplificou a manipulação dos dados.

Em seguida, utilizamos o Flask para implementar a parte que lidava com o processamento e manipulação dos dados do servidor no back-end e desenvolvemos a interface do usuário com tecnologias web, como HTML, CSS e JavaScript, para criar o front-end. A aplicação final permitiu aos usuários pesquisar e consultar informações jurídicas de forma rápida e fácil, utilizando uma interface amigável e intuitiva.

Em resumo, o projeto envolveu tanto o desenvolvimento do back-end quanto do front-end, e permitiu a criação de uma solução completa para consulta e pesquisa de informações jurídicas.

Capítulo 2

Análise de dados

2.1 Desenvolvimento do programa em Python para extração e análise dos dados

Na fase inicial do projeto, foi necessário fazer uma análise detalhada dos dados disponíveis em cada uma das bases jurídicas utilizadas. Para garantir a integridade dos dados e facilitar a sua manipulação, foram desenvolvidos dois programas em Python.

O primeiro programa extraiu os campos de cada dataset em formato JSON e listou-os de forma organizada, permitindo uma melhor interpretação dos dados contidos em cada dataset. Já o segundo programa comparou os campos dos diferentes datasets e identificou quais eram comuns a todos eles.

Ao listar os campos, foi possível observar que havia campos que se repetiam em diferentes datasets. A identificação desses campos em comum permitiu a criação de um modelo normalizado que foi utilizado na construção da aplicação web. Dessa forma, os dados foram estruturados de maneira mais organizada e otimizada, permitindo uma melhor manipulação e consulta da informação.

```
{
  "Contencioso": "TRIBUTÁRIO",
  "Peça Processual": "PARECERES",
  "Data": "12/02/2016",
  "Processo": "2530/15.0BELRS",
  "Nº Processo/TAF": "2530/15.0BELRS",
  "Sub-Secção": "2º Juízo-2ª Secção ( Contencioso Tributário)",
  "Magistrado": "FERNANDA CARNEIRO",
  "Descritores": [
    "PRAZO",
    "IMPUGNAÇÃO JUDICIAL",
    "VICIO GERADOR DE MERA ANULABILIDADE"
  ],
  "Tema": "OUT - Outros",
  "url": "/jtcampct.nsf/a10cb5082dc606f9802565f600569da6/b158aa2de0b4124e802581d8004d6bd6?OpenDocument",
  "tribunal": "jtcampct"
},
{
  "Contencioso": "TRIBUTÁRIO",
  "Peça Processual": "PARECERES",
  "Data": "10/04/2016",
  "Processo": "9905/16",
  "Nº Processo/TAF": "00000/00/0",
  "Magistrado": "FERNANDA CARNEIRO",
  "Descritores": [
    "RESPONSÁVEL SUBSIDIÁRIO",
    "PRESCRIÇÃO DO IVA - ANOS 92 E 93"
  ],
  "Tema": "OUT - Outros",
  "url": "/jtcampct.nsf/a10cb5082dc606f9802565f600569da6/3a612986e262b3e4802581d8004ecc20?OpenDocument",
  "tribunal": "jtcampct"
},
```

Figura 2.1: Exemplo de dois acórdãos em formato JSON do tribunal jtcampct

```
[
  {
    "Dicionario 1": [
      "Contencioso",
      "Peça Processual",
      "Data",
      "Processo",
      "Nº Processo/TAF",
      "Sub-Secção",
      "Magistrado",
      "Descritores",
      "Tema",
      "url",
      "tribunal"
    ]
  },
  ...
]
```

(a) Campos do dataset jtcampct

```
[
  "url",
  "tribunal",
  "Processo",
  "Descritores"
]
```

(b) Partes comuns aos datasets

Sendo os datasets listas de acordãos em JSON, foi fácil a extração de informação dos mesmos e obtivemos os seguintes resultados (a) e (b).

2.2 Utilização de requests em Python para obtenção das informações necessárias

Para obtenção das informações necessárias, foi utilizada a biblioteca requests em Python, que permitiu fazer solicitações HTTP. A partir dessas solicitações, foram carregados dados em formato JSON, que foram armazenados num servidor local utilizando o json server.

O json server é uma ferramenta que permite a criação rápida de uma API REST completa a partir de um arquivo JSON. Com isso, foi possível disponibilizar os dados num formato que pode ser facilmente acesso pela aplicação web desenvolvida em Flask.

Essa etapa foi fundamental para o desenvolvimento do back-end da aplicação, que consistiu na criação de uma API RESTful capaz de disponibilizar os dados em formato JSON para a aplicação web. A utilização da biblioteca requests e do json server permitiu a obtenção e disponibilização dos dados de maneira rápida e eficiente, possibilitando o desenvolvimento da aplicação em um curto espaço de tempo.

Antes de usar requests, usámos o Nano ID, que é uma biblioteca JavaScript para a geração de identificadores únicos (IDs), para colocar o campo "ID" único em cada acórdão dos diferentes datasets, depois de cada acórdão estar identificado foi só fazer POST do mesmo.

Aqui está um exemplo de como fazer usar POST usando a biblioteca requests:

```
for acordao in dataset["acordaos"]:
    post = requests.post('http://localhost:3000/acordaos')

    if post.status_code <= 300:
        print(f'{post.status_code}: {acordao["id"]} {post.reason}')
    else:
        print(f'Erro {post.status_code}: {post.reason}')
```

Figura 2.3: Exemplo de dois acórdãos em formato JSON do tribunal jtcampct

Além do método POST, também outros métodos GET, PUT, DELETE e entre outros para enviar diferentes tipos de solicitações.

Capítulo 3

Desenvolvimento WEB

A introdução ao desenvolvimento web em *Flask* é essencial para compreender a estrutura e funcionalidades desta tecnologia. No projeto em questão, utilizamos o *Flask* como *framework* para criar a aplicação *WEB* que permite a consulta e pesquisa eficiente das bases jurídicas.

O *Flask* é um *framework* em *Python* que facilita a construção de aplicações web, oferecendo ferramentas para lidar com rotas, modelos e autenticação.

Com o *Flask*, podemos criar uma interface amigável e intuitiva, melhorando a acessibilidade e eficiência da pesquisa jurídica.

3.1 Aplicação

A aplicação desenvolvida como parte deste projeto tem como objetivo principal oferecer aos usuários uma plataforma interativa para pesquisar e consultar acórdãos jurídicos de diferentes tribunais. Por meio dessa aplicação, os usuários podem acessar informações relevantes e atualizadas sobre decisões judiciais.

A funcionalidade principal da aplicação é fornecer aos usuários a capacidade de realizar pesquisas personalizadas com base em critérios específicos, como número do processo, data, tribunal, entre outros. Os usuários podem inserir os parâmetros de pesquisa desejados e obter uma lista de acórdãos que correspondam aos seus critérios.

Além da pesquisa, a aplicação também oferece recursos adicionais, como a visualização detalhada de cada acórdão selecionado, que inclui informações sobre as partes envolvidas, relatores, votos, ementas e demais informações relevantes para análise jurídica.

A interação entre a aplicação e o usuário é realizada por meio de uma interface intuitiva e amigável, que permite uma experiência de uso agradável. Os usuários podem navegar facilmente pelos recursos da aplicação, realizar pesquisas de forma eficiente e acessar os acórdãos desejados com facilidade.

Esta aplicação *WEB* é uma ferramenta útil para profissionais do meio jurídico, como advogados, juízes, estudantes de direito e pesquisadores, que necessitam de um acesso rápido e prático aos acórdãos jurídicos para fins de análise e pesquisa.

3.1.1 Interface

A interface da nossa aplicação web, intitulada com "Bases Jurídicas", foi projetada cuidadosamente para fornecer aos utilizadores uma experiência intuitiva e eficiente.

Através de uma combinação de elementos visuais atrativos e funcionalidades bem organizadas, a nossa aplicação tem como objetivo facilitar a navegação e o acesso às informações jurídicas relevantes.

A página inicial, conhecida como "*Homepage*", serve como ponto de partida, apresentando aos usuários a proposta de valor da aplicação e oferecendo um resumo do seu conteúdo. Além disso, as seções "*Acórdãos*" e "*Informações*" permitem que os utilizadores encontrem e explorem acórdãos específicos, além de obter detalhes sobre o projeto e os seus criadores.

Homepage

A *Homepage* da nossa aplicação web, intitulada com *Bases Jurídicas*, desempenha um papel crucial ao fornecer uma visão geral e direcionar os usuários para as principais funcionalidades e recursos disponíveis.

Esta serve como o ponto de partida para os visitantes, oferecendo uma apresentação concisa do propósito e dos recursos da nossa aplicação.

A principal função da *Homepage* é apresentar aos usuários a proposta de valor da nossa aplicação, ou seja, destacar o que torna nosso serviço único e valioso para os profissionais e estudantes de direito. Ela oferece um breve resumo do que podem esperar ao explorar o conteúdo disponível.

Além disso, a *Homepage* atua como um *hub* central, fornecendo acesso rápido a recursos importantes da aplicação. Por exemplo, os usuários podem encontrar dois botões principais na página inicial: **Acórdãos** e **Informações**. Ao clicar nestes botões, os utilizadores serão redirecionados para secções específicas da aplicação onde poderão encontrar uma ampla variedade de acórdãos, bem como informações relacionadas ao campo jurídico e até mesmo sobre os criadores.



Figura 3.1: Interface - Homepage

Acórdãos

A página de acórdãos da nossa aplicação *WEB* desempenha um papel fundamental ao fornecer uma lista de acórdãos disponíveis para consulta e pesquisa.

Esta página tem como objetivo permitir que os usuários encontrem e acessem informações específicas sobre acórdãos jurídicos relevantes.

A página de acórdãos apresenta uma tabela com informações essenciais sobre cada acórdão, como ID,

data, processo, descritores e tribunal. Esses dados são exibidos em colunas correspondentes na tabela, facilitando a visualização e a navegação pelos acórdãos.

Além disso, a página oferece a funcionalidade de pesquisa, onde os usuários podem digitar palavras-chave no campo de pesquisa para filtrar os acórdãos exibidos na tabela. À medida que os usuários digitam, os acórdãos são filtrados com base nas correspondências encontradas nos campos da tabela. Isso permite que os usuários encontrem rapidamente acórdãos relevantes com base em suas consultas de pesquisa.

Voltar

Acórdãos

Search...

ID	Data	Processo	Descritores	Tribunal
rB9-gaMi3N	03/10/1998	97-0847	['ACESSO AOS TRIBUNAIS.', 'PRINCÍPIO DA PROPORCIONALIDADE.', 'CUSTAS.', 'PROCESSO FISCAL.']}	atco1
r1smK3Y4d	03/10/1998	97-0215	['QUESTÃO PRÉVIA.', 'INCONSTITUCIONALIDADE SUSCITADA NO PROCESSO.', 'ACESSO AOS TRIBUNAIS.', 'PROCESSO CIVIL.', 'PRINCÍPIO DO CONTRADITÓRIO.', 'ESTADO DE DIREITO DEMOCRÁTICO.']}	atco1
5iwXiXn9ge	03/09/1998	97-0370	['IMPOSTOS.', 'TRIBUTAÇÃO DA EMPRESA.', 'PRINCÍPIO DA LEGALIDADE TRIBUTARIA.', 'RETROATIVIDADE DA LEI FISCAL.', 'INTERPRETAÇÃO DA LEI.', 'DIREITO ORDINÁRIO ANTERIOR.', 'REVISÃO DA CONSTITUIÇÃO.', 'COMPETÊNCIA DO TRIBUNAL CONSTITUCIONAL.', 'LEI FISCAL.', 'ESTADO DE DIREITO.', 'LEI INTERPRETATIVA.']}	atco1
qjkbK4Qp4V	03/09/1998	97-0272	['PRINCÍPIO DA PROPORCIONALIDADE.', 'ILÍCITO DE MERA ORDENAÇÃO SOCIAL.', 'PRINCÍPIO DA NECESSIDADE.', 'CRIMINALIZAÇÃO.', 'PRINCÍPIO DA JUSTIÇA.', 'INTERESSE PÚBLICO.', 'PRINCÍPIO DA CULPA.', 'AMBIENTE.']}	atco1
uAS155GU2	03/09/1998	94-0039	['PROCESSO CONSTITUCIONAL.', 'FISCALIZAÇÃO CONCRETA DA CONSTITUCIONALIDADE.', 'APLICAÇÃO DE NORMA ARGUIDA DE INCONSTITUCIONAL.', 'PRESSUPOSTO DO RECURSO.', 'CONHECIMENTO DO RECURSO.', 'INTERPRETAÇÃO DE NORMA.', 'INUTILIDADE SUPERVENIENTE.', 'IMPRESSA.', 'CRIME DE IMPRESSA.', 'LIBERDADE DE INFORMAÇÃO.']}	atco1
b8pG4rms7g	03/09/1998	96-0846	['PROCESSO CONSTITUCIONAL.', 'FISCALIZAÇÃO CONCRETA DA CONSTITUCIONALIDADE.', 'RECLAMAÇÃO.', 'PRESSUPOSTO DO RECURSO.', 'ADMISSIBILIDADE DO RECURSO.']}	atco1
yYGiZJOar	03/09/1998	97-0403	['PROCESSO CONSTITUCIONAL.', 'FISCALIZAÇÃO CONCRETA DA CONSTITUCIONALIDADE.', 'RECLAMAÇÃO.', 'ADMISSIBILIDADE DO RECURSO.', 'DESERÇÃO DE RECURSO.']}	atco1
-K1BpyD6MZ	03/09/1998	96-0882	['TÍTULO EXECUTIVO.', 'FUNÇÃO JURISDICIONAL.', 'DÍVIDA HOSPITALAR.', 'SERVIÇO NACIONAL DE SAÚDE.']}	atco1
-UetXJR7_	03/05/1998	94-0181	['CONSTITUIÇÃO FISCAL.', 'PRINCÍPIO DA LEGALIDADE TRIBUTARIA.', 'PRINCÍPIO DA TIPICIDADE DOS IMPOSTOS.', 'NORMA.', 'DECISÃO DE TRIBUNAL.']}	atco1
01g_OdXhIx	03/05/1998	97-0633	['DIREITO AO RECURSO.', 'DESPACHO DE PRONÚNCIA.', 'GARANTIAS DE PROCESSO CRIMINAL.', 'GARANTIAS DE DEFESA.', 'PRINCÍPIO DA IGUALDADE DE ARMAS.']}	atco1
vHnBLD5kQk	03/05/1998	93-0418	['FISCALIZAÇÃO ABSTRACTA DA CONSTITUCIONALIDADE.', 'AUTARQUIA LOCAL.', 'FINANÇAS LOCAIS.', 'CONSIGNAÇÃO DE RECEITAS.', 'TUTELA ADMINISTRATIVA.']}	atco1
flLcUf8Td	03/05/1998	93-0486	['INICIATIVA PRIVADA.', 'PROPRIEDADE PRIVADA.', 'CONTRATO COLECTIVO DE TRABALHO.', 'ACORDO COLECTIVO DE TRABALHO.', 'DIREITO DE CONTRATAÇÃO COLECTIVA.', 'PORTARIA DE EXTENSÃO.', 'LIBERDADE CONTRATUAL.', 'SEGURANÇA NO EMPREGO.']}	atco1

Figura 3.2: Interface - Acórdãos

Informações

A página de informações desempenha um papel crucial ao fornecer detalhes sobre o projeto 'Bases Jurídicas', incluindo a descrição do projeto, a estrutura dos *datasets*, a lista de tribunais disponíveis e um link para informações sobre os autores. Essa página tem como objetivo oferecer uma visão abrangente e informativa sobre o projeto, permitindo que os usuários entendam melhor o contexto e a natureza das informações jurídicas disponíveis na aplicação web.

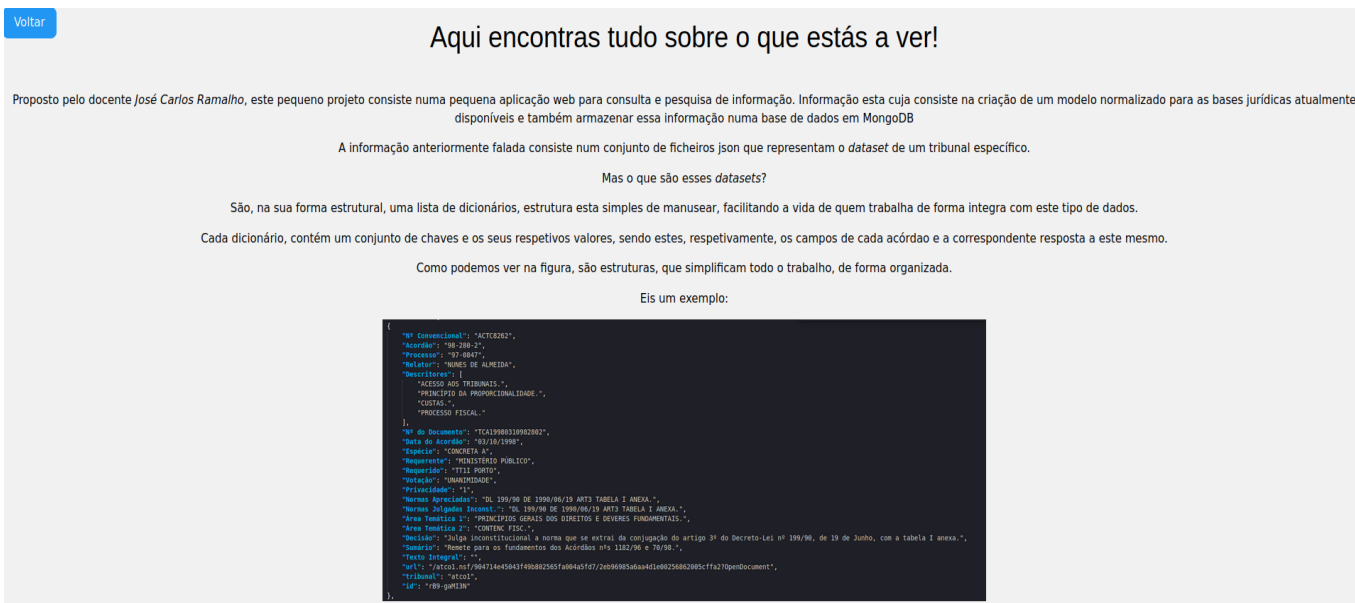


Figura 3.3: Interface - Informações

3.1.2 Construção da Aplicação WEB

A criação de aplicações WEB é uma tarefa essencial na era digital em que vivemos. O *Flask*, um *framework* web leve e poderoso baseado em *Python*, oferece uma abordagem flexível para o desenvolvimento dessas aplicações. No cerne de uma aplicação *Flask* está o arquivo `app.py`, que funciona como o ponto de entrada principal. Neste arquivo, definimos as rotas, os controladores e outras configurações importantes para o funcionamento da aplicação.

Além disso, para uma experiência do usuário mais rica, é comum utilizar os diretórios *'template'* e *'static'*. O diretório *'template'* é onde armazenamos os arquivos *HTML* que definem a estrutura e o *layout* das páginas da aplicação, enquanto o diretório *'static'* é utilizado para armazenar arquivos estáticos, como folhas de estilo *CSS*, *scripts JavaScript* e *imagens*, que são utilizados para aprimorar a aparência e a funcionalidade da aplicação.

Criação da Homepage

Inicialmente, desenvolvemos uma página inicial para a nossa aplicação.

A estrutura básica da página é definida usando a marcação *HTML*. Esta inclui uma secção de cabeçalho, onde definimos o título da página como *"Homepage"*, adicionando uma folha de estilo *CSS* externa para aplicar estilos à página. A página utiliza a biblioteca *W3.CSS* para fornecer um design moderno e responsivo.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Homepage</title>
    <link rel="stylesheet" href="/static/w3.css"/>
```

Figura 3.4: HTML : Homepage Tittle

Na seção do corpo da página, adicionamos um logotipo centralizado e uma breve descrição sobre o propósito do site. O logotipo é exibido usando a tag ``, com algumas propriedades *CSS* aplicadas para definir sua largura e centralizá-lo na página. A descrição é exibida em um parágrafo (`<p>`) e também está centralizada verticalmente na página.

```

```

Figura 3.5: HTML : tag ``

Além disso, a página possui dois botões que levam a diferentes seções do site. Esses botões são criados usando a tag `<button>` e estão envoltos em tags `<a>` para fornecer links para outras páginas do site. Cada botão tem uma classe *CSS* para aplicar o estilo, por exemplo, `"w3-blue"` da biblioteca *W3.CSS*.

```
<a href="{{ url_for('mostrar_acordaos') }}"><button class="w3-button w3-blue">Acórdãos</button></a>
<a href="{{ url_for('mostrar_informacoes') }}"><button class="w3-button w3-blue">Informações</button></a>
```

Figura 3.6: HTML : tag `<button >`

Criação da *Acórdãos*

A página *Acórdãos*, foco principal do projeto, é um dos caminhos a partir da página inicial (*Homepage*). Esta é também definida usando a marcação *HTML*, assim como, para o estilo da página, usamos da mesma forma a biblioteca *W3.CSS*. A página inclui uma seção de cabeçalho, onde definimos o título da página como *"Acórdãos"*, adicionando uma folha de estilo *CSS* externa para aplicar estilos à página.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Acórdãos</title>
    <link rel="stylesheet" href="/static/w3.css"/>
```

Figura 3.7: HTML : Acórdãos Tittle

De seguida, encontramos uma barra de pesquisa onde os utilizadores podem digitar palavras-chave para filtrar os acórdãos exibidos na tabela. A barra de pesquisa é representada por um elemento de entrada de texto (`<input>`) com um atributo `type` definido como `"text"`. O elemento possui um identificador único `"searchBox"` e uma classe CSS `"w3-input w3-left w3-round-large w3-small w3-blue"` para aplicar o estilo adequado. Adicionamos um ouvinte de eventos (registrar uma função para ser executada quando um determinado evento ocorrer num elemento específico) ao elemento para capturar a digitação do usuário e realizar o filtro na tabela de acórdãos.

```
<input type="text" name="search_query" placeholder="Search..." id="searchBox" class="w3-input w3-left w3-round-large w3-small w3-blue">
```

Figura 3.8: HTML : tag `<input>`

A tabela de acórdãos é representada por um elemento `<table>`. Esta possui uma classe CSS `"w3-table-all w3-striped w3-border w3-hoverable"`, aplicando um estilo da tabela da biblioteca *W3.CSS*. A tabela contém um cabeçalho (`<thead>`) e um corpo (`<tbody>`). O cabeçalho contém uma linha (`<tr>`) com os títulos das colunas: *ID*, *Data*, *Processo*, *Descritores* e *Tribunal*. As células de cada linha contêm os dados correspondentes a cada acórdão.

```
<table class="w3-table-all w3-striped w3-border w3-hoverable">
  <thead>
    <tr class="w3-blue">
      <th>ID</th>
      <th>Data</th>
      <th>Processo</th>
      <th>Descritores</th>
      <th>Tribunal</th>
    </tr>
  </thead>
  <tbody id="tableBody">
```

Figura 3.9: HTML : Table Head

Os acórdãos são preenchidos dinamicamente usando uma estrutura de repetição em *Python*, e cada acórdão é representado por uma linha (`<tr>`) no corpo da tabela. As células (`<td>`) dentro de cada linha exibem os atributos do acórdão, como *ID*, *Data*, *Processo*, *Descritores* e *Tribunal*. Dependendo dos dados disponíveis, como *Data do Acórdão* ou *Data da Decisão*, a célula correspondente é preenchida com a informação apropriada. Os *IDs* dos acórdãos são exibidos como botões, permitindo aos utilizadores abrir cada acórdão em uma nova guia.

```
<tbody id="tableBody">
  {% for acordao in acordaos['acordaos'] %}
  <tr class="w3-bordered">
    <td><a href="/acordaos/{{ acordao['id'] }}" target="_blank"><button class="w3-button w3-blue w3-round-large">{{ acordao['id'] }}</button></a></td>
    {% if 'Data do Acórdão' in acordao %}
    <td>{{ acordao['Data do Acórdão'] }}</td>
    {% elif 'Data da Decisão' in acordao %}
    <td>{{ acordao['Data da Decisão'] }}</td>
    {% else %}
    <td>{{ acordao['Data'] }}</td>
    {% endif %}
    <td>{{ acordao['Processo'] }}</td>
    <td>{{ acordao['Descritores'] }}</td>
    <td>{{ acordao['tribunal'] }}</td>
  </tr>
  {% endfor %}
</tbody>
```

Figura 3.10: HTML : Table Body

Por fim, implementamos um *script JavaScript* para habilitar a funcionalidade de filtragem de acórdãos com base na barra de pesquisa. O *script* associa o elemento de entrada de texto e a tabela de acórdãos a variáveis e adiciona um ouvinte de eventos (neste caso: *keyup*) para capturar as alterações na barra de pesquisa. A cada digitação do utilizador, o *script* percorre as células da tabela e oculta as linhas que não correspondem ao filtro.

```

<script>
const searchBox = document.getElementById('searchBox');
const tableBody = document.getElementById('tableBody');
const rows = tableBody.getElementsByTagName('tr');

searchBox.addEventListener('keyup', function () {
    const filter = searchBox.value.toLowerCase();

    for (let i = 0; i < rows.length; i++) {
        const row = rows[i];
        const cells = row.getElementsByTagName('td');
        let visible = false;

        for (let j = 0; j < cells.length; j++) {
            const cell = cells[j];

            if (cell.innerHTML.toLowerCase().indexOf(filter) > -1) {
                visible = true;
                break;
            }
        }

        row.style.display = visible ? '' : 'none';
    }
});
</script>

```

Figura 3.11: HTML : Script

Criação da *Informações*

Além do caminho para a pagina *Acórdãos*, encontramos (na *Homepage*) agora um caminho para a página *Informações*. Semelhante à construção das outras páginas, aqui encontramos algumas informações sobre o propósito da construção da aplicação *WEB* referida até aqui, bem como alguns detalhes importantes como a origem das informações jurídicas apresentas. Além disso, é possível encontrar um caminho para uma página com algumas informações sobre os criadores do conteúdo apresentado.

Capítulo 4

Conclusão

Com a conclusão deste projeto é possível afirmar que o grupo concretizou o desafio esperado: implementar uma aplicação web para consulta e pesquisa de informações jurídicas. A realização deste projeto contribuiu para que os elementos do grupo entendessem como utilizar as funcionalidades da biblioteca *requests* em Python, como funciona o *Json Server*, o *Flask* e ainda conhecimentos básicos de *HTML*, *CSS* e *Javascript* com auxílio do site *w3schools*¹.

Os desafios que fomos confrontando ao longo do desenvolvimento do projeto contribuíram para a consolidação e a progressão do conhecimento.

Em suma, todos os elementos estão satisfeitos com o trabalho que foi feito ao longo do semestre visto que o resultado é uma aplicação web prática e funcional, tal como nos foi proposto.

No futuro, se for necessário introduzir mais bases de dados jurídicas poderíamos tentar tornar a aplicação mais intuitiva de forma a que fosse ainda mais simples aceder a todos os dados pretendidos.

¹<https://www.w3schools.com/>