

Criando as views

As views são as telas de interface com o usuário.

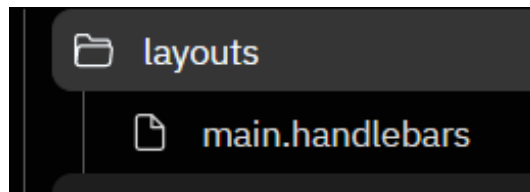
Na última aula, criamos os arquivos das views handlebars.

É importante ressaltar novamente o que é o handlebars.

O handlebars, é um módulo que utilizamos para passar parâmetros de uma forma bem mais fácil dentro das views no express. Daqui a pouco vamos ver isso na prática.

Ao utilizar o handlebars, é importante criarmos uma pasta dentro da views para armazenar a configuração básica das interfaces.

Dentro da pasta views, crie a pasta layouts, logo após crie o arquivo **main.handlebars**



Main.handlebars

```
<html lang="pt-br">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  {{! BOOSTRAP 5 }}
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN
9+nJOZ" crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1Hlcje39Wm4jDKdf19U8gl4ddQ3GYNS7NTKfAdVQ
SZe" crossorigin="anonymous"></script>
  {{!-- ARQUIVOS LOCAL --}}
  <link rel="stylesheet" href="css/styles.css" />
  <title>Sistema de Cadastro de Videos</title>
```

```
</head>

<body id="body-mobile">
  <nav class="navbar text-bg-dark">
    <div class="container">
      <span class="navbar-brand text-white">Sistema de Videos</span>
      <ul class="navbar nav">
        <li class="nav-item">
          </li>
        <li class="nav-item">
          <a class="nav-link text-white" href="/">Listar Videos</a>
        </li>
        <li class="nav-item">
          <a class="nav-link text-white" href="/Cadastrar">Cadastrar</a>
        </li>
      </ul>
    </div>
  </nav>
  <div class="container">
    {{{body}}}
  </div>
</body>
</html>
```

O arquivo é um HTML feito em bootstrap, porém com alguns detalhes de utilização de parâmetros do handlebars.

O arquivo será um cabeçalho de todas as nossas páginas.

Dentro do arquivo temos uma linha de código que é mostrada abaixo.

```
<div class="container">
  {{{body}}}
</div>
```

O `{{body}}` é um parâmetro que receberá o corpo das nossas views.

Agora vamos ver a página **Cadastrar.handlebars**

```
<h1 class="text-center my-3">Cadastro de videos</h1>
<form action="/Cadastrar" method="post">
  <div class="row w-50 d-block m-auto g-3">
    <div class="col-12">
      <label class="form-label" for="autor">Autor:</label>
      <input
        class="form-control"
        type="text"
        name="autor"
        id="autor"
        placeholder="Digite o nome do autor do video"
      />
    </div>
    <div class="col-12">
      <label class="form-label" for="titulo">Titulo:</label>
      <input
        class="form-control"
        type="text"
        name="titulo"
        id="titulo"
        placeholder="Digite o título do video."
      />
    </div>
    <div class="col-12">
      <label class="form-label" for="assunto">Assunto:</label>
      <input
        class="form-control"
        type="text"
        name="assunto"
        id="assunto"
        placeholder="Digite o assunto do video"
      />
    </div>
    <div class="col-12">
      <label class="form-label" for="descricao">Descrição:</label>
      <textarea
        class="form-control"
        name="descricao"
        id="descricao"
        placeholder="Digite a descrição do video"
      />
    </div>
  </div>
</form>
```

```

    ></textarea>
  </div>
  <div class="col-12">
    <label class="form-label" for="link">Link:</label>
    <input
      class="form-control"
      name="link"
      id="link"
      placeholder="Link do video"
    />
  </div>
  <div>
    <input class="btn btn-primary my-3" type="submit" value="Cadastrar video"
  />
  </div>
</div>
</form>

```

O arquivo cadastrar é um formulário que enviará um action com method=post.

A próxima view é a **listar.handlebars**

O código HTML é mostrado abaixo.

```

<h1 class="text-center my-3">Lista de Todos os videos Preferidos</h1>
<table class="table">
  <div class="row">
    <div class="col-12">
      <thead>
        <th>Código do video</th>
        <th>Autor</th>
        <th>Título</th>
        <th>Assunto</th>
        <th>Descrição</th>
        <th>link para visualizar</th>
      </thead>
      <tbody>
        {{#each video}}
          <tr>
            <th>{{this.id_video}}</th>
            <th>{{this.autor}}</th>
            <th>{{this.titulo}}</th>
            <th>{{this.assunto}}</th>
            <th>{{this.descricao}}</th>
            <th><div class="embed-responsive embed-responsive-16by9">

```

```

<iframe width="150" height="110" class="embed-responsive-item"
src="{{this.link}}" allowfullscreen></iframe>
</div></th>
    <th><a href="/update/{{this.id_video}}">Editar<i class="bi bi-pencil-square"></i></a></th>

    <th><form class="borde-0 m-0" action="/remove" method="POST">
    <input type="hidden" name="id_video" value="{{this.id_video}}>
    <input type="submit" value="Excluir" name="Excluir"/>

    </form></th>

</tr>
{{/each}}
</tbody>
</div>
</div>
</table>

```

O código marcado em vermelho é um **each** do nosso array que é retornado do controller com todos os objetos do banco.

O listar também coloca um link caso o usuário queira atualizar ou excluir aquele objeto do banco.

É muito importante esse recurso para agilizar processos.

É um comando de repetição que fará a inclusão de todos os objetos mostrando seus atributos pelo comando **this.atributo**.

Já estamos com as views cadastrar e listar quase prontas, agora vamos codificar a `update.handlebars`

```

<h1 class="text-center my-3">Atualizar Video: {{video.titulo}}</h1>
<form action="/update" method="POST">
  <input type="hidden" name="id_video" value="{{video.id_video}}" />
  <div class="row w-50 d-block m-auto g-3">
    <div class="col-12">
      <label class="form-label" for="titulo">Titulo:</label>
      <input
        class="form-control"
        type="text"
        name="titulo"
        id="titulo"
        value="{{video.titulo}}"
      />
    </div>
  </div>

```

```
</div>
<div class="col-12">
  <label class="form-label" for="autor">Autor:</label>
  <input
    class="form-control"
    type="text"
    name="autor"
    id="autor"
    value="{{video.autor}}"
  />
</div>

<div class="col-12">
  <label class="form-label" for="assunto">Assunto:</label>
  <input
    class="form-control"
    type="text"
    name="assunto"
    id="assunto"
    value="{{video.assunto}}"
  />
</div>
<div class="col-12">
  <label class="form-label" for="descricao">Descrição:</label>
  <textarea
    class="form-control"
    name="descricao"
    id="descricao"
  >{{video.descricao}}</textarea>
</div>

<div class="col-12">
  <label class="form-label" for="link">Link do video:</label>
  <input
    class="form-control"
    type="text"
    name="link"
    id="link"
    value="{{video.link}}"
  />
</div>
<div>
  <input class="btn btn-primary my-3" type="submit" value="Atualizar video"
/>
```

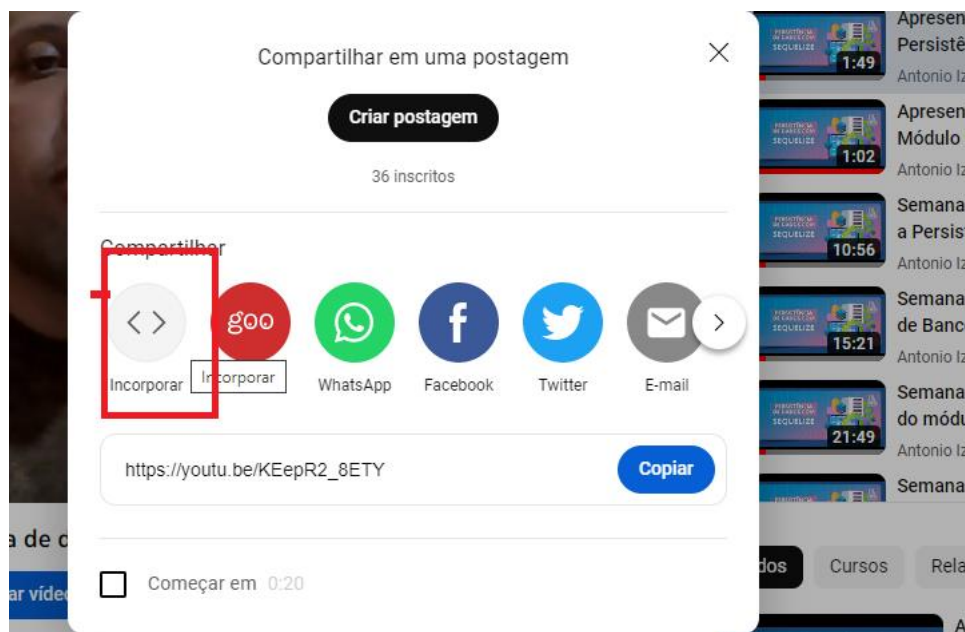
```
</div>
</div>
</form>
```

A views atualizar recebe do banco o objeto selecionado na views listar quando o usuário clica na opção editar.

Após o usuário finalizar as alterações o formulário é enviado e atualizado no banco.

Agora vamos testar nosso aplicativo.

Lembrando que o cadastro do link do vídeo deve ser feito com a opção incorporar do youtube e somente o link deve ser copiado.



```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
Qpx3a87F0Uc" title="YouTube video
player" frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture; web-
share" allowfullscreen></iframe>
```

☐ Começar em 0:05

Vamos testar nossa aplicação.


Lista de Todos os videos Preferidos

Código do video	Autor	Título	Assunto	Descrição	link para visualizar
-----------------	-------	--------	---------	-----------	----------------------

Cadastrar

Listar

Lista de Todos os videos Preferidos

Código do video	Autor	Título	Assunto	Descrição	link para visualizar	
6	Antonio Izo Júnior	Persistência de dados Apresentação	Programação	Módulo III do Reprograme-se		Editar Excluir

Teste as outras funcionalidades e veja como ficou bacana.

Pessoal, aqui finalizamos nosso módulo III - persistência de dados com sequelize.

Espero que tenha construído muita aprendizagem.

Até breve...