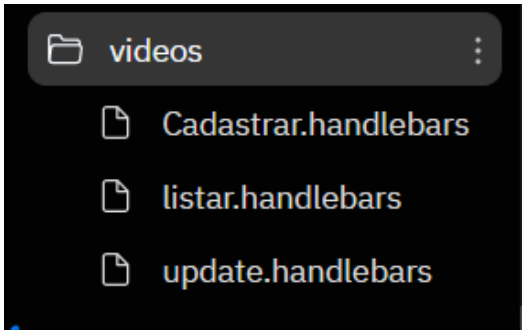
Criando as rotas

Para configurar nossas rotas, vamos deixar os arquivos das views criados.

Já criamos na aula passada a view Cadastrar.handlebars.

Agora crie as views listar.handlebars e update.handlebars.



Vamos para nosso arquivo de rotas routesVideo.js na pasta routes.

Vamos começar criando as contantes para os módulos que utilizaremos.

```
const express = require("express");
const ControllerVideo = require("../controllers/ControllerVideo");
const router = express.Router();
```

Agora vamos criar nossa router **/Cadastrar**, toda vez que for chamada /Cadastrar com o method informado, chamaremos nossa função de cadastro que criamos no arquivo controller.

```
router.get("/Cadastrar", ControllerVideo.cadastrarVideo);

router.post("/Cadastrar", ControllerVideo.VideoCreate);
```

A Cadastrar com router.get chama a view Cadastrar que mostrará o formulário para o usuário.

A Cadastrar com router.post será o retorno do formulário para cadastro no banco.

```
router.get("/", ControllerVideo.listarVideos);
```

A router.get / mostra todos os vídeos que temos no banco de dados.

```
router.get("/update/:id_video", ControllerVideo.UpdateVideo);
router.post("/update", ControllerVideo.VideoUpdate);
```

O router.get /update/:id_video retorna para a view update os valores do objeto com o id_video selecionado para alteração.

O router.post /update envia para o banco os dados para atualização.

Finalizando nossas rotas, temos o delete.

```
router.post("/remover", ControllerVideo.removerVideo);
```

O router.post /remover chama a função para remover o objeto do banco.

```
module.exports = router;
```

Agora colocamos nosso a exportação do módulo.

Nosso arquivo completo é mostrado abaixo.

```
const express = require("express");
const ControllerVideo = require("../controllers/ControllerVideo");
const router = express.Router();

router.get("/Cadastrar", ControllerVideo.cadastrarVideo);

router.post("/Cadastrar", ControllerVideo.VideoCreate);

router.get("/", ControllerVideo.listarVideos);

router.get("/update/:id_video", ControllerVideo.UpdateVideo);

router.post("/update", ControllerVideo.VideoUpdate);

router.post("/remover", ControllerVideo.removerVideo);

module.exports = router;
```

Agora vamos organizar nosso **index.js**

Até agora nosso index está assim:

```
//BIBLIOTECAS/MODULOS UTILIZADOS
const database = require("./db/db");
const express = require("express");
const app = express();

//SINCRONISMO COM O BANCO DE DADOS
```

```
try {
    database.sync().then(() => {
    app.listen(9443,() => { console.log('Servidor rodando') });
    })
}
catch(erro) {
    console.log("Houve uma falha ao sincronizar com o banco de dados. ", erro);
};
```

Vamos colocar os módulos que instalamos e ainda não configuramos.Coloque o código acima do sincronismo com o banco de dados.

```
const hand = require("express-handlebars");
//MODELS
const Video = require("./models/Video");
const VideoRoutes = require("./routes/routesVideo");
//CONTROLLERS
const VideosControllers = require("./controllers/ControllerVideo");
```

Agora vamos configurar a utilização do handlebars e informar que iremos utilizar as views.

Além de configurar o nosso servidor.

```
//UTILIZAÇÃO DO HANDLEBARS
app.engine("handlebars", hand.engine());
app.set("view engine", "handlebars");
app.use(express.urlencoded({extended: true,}));

app.use(express.json());
app.use(express.static("public"));

//ROTAS
app.use("/", VideoRoutes);
```

Nosso arquivo index já está finalizado.

Temos que configurar nossas views.

Até a próxima aula...