



Interprêt

Uma plataforma para
o direito de defesa

○ Problema



As Estatísticas

+42
mi



Passageiros em
GRU

2019

6,3
mi



Turistas no Brasil

2019

9,7
mi



Comunidade surda
brasileira.

Censo 2010

896
mil



Indígenas
autodeclarados

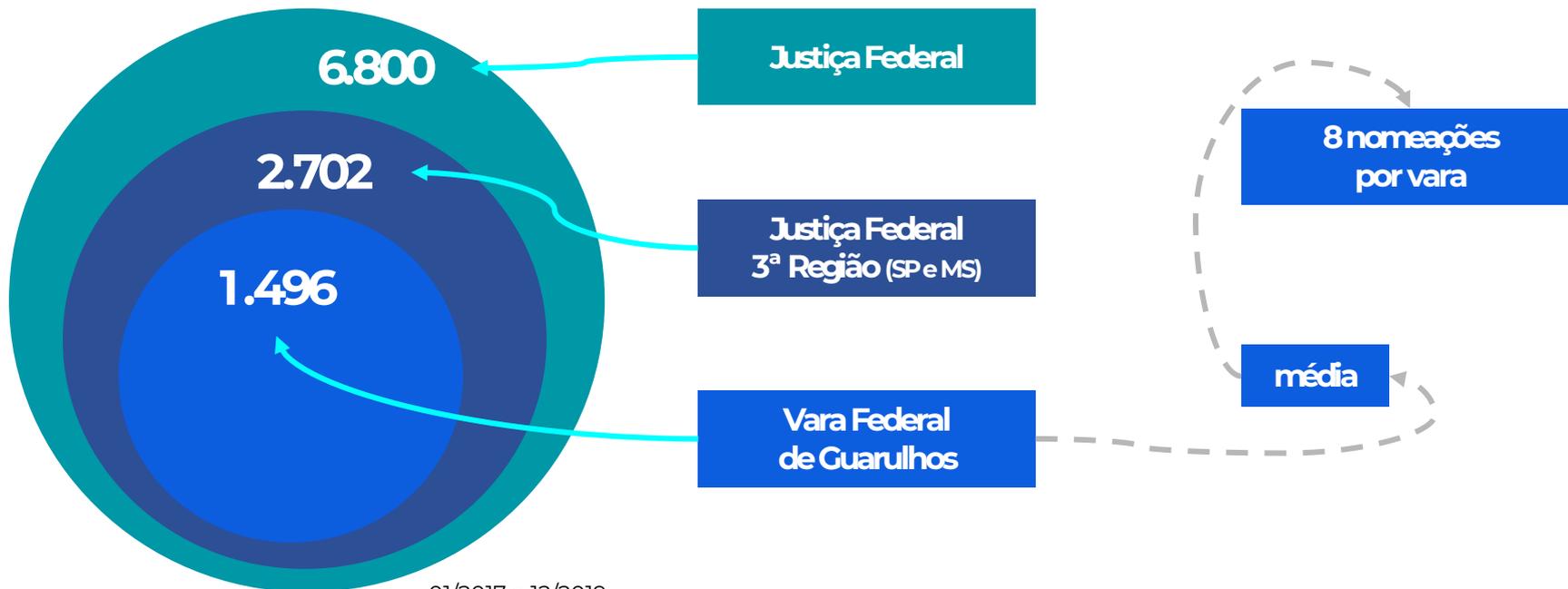
Censo 2010

Intérprete Forense

Ajuda a remover **barreiras linguísticas** e **culturais** no ambiente judicial

Intérprete Forense

Nomeações por região



01/2017 a 12/2019.

Problema



“ *Faltam intérpretes forenses qualificados.*

(Almeida e Nordin, 2017)



Problema

Justiça Federal da 3ª Região promove curso de capacitação para intérpretes forenses

Cerca de 50 profissionais foram treinados em línguas estrangeiras, indígenas e de sinais



Problema

Como podemos ajudar a
Justiça a **encontrar** e
selecionar intérpretes
qualificados?

A Proposta



Projeto Interpretê

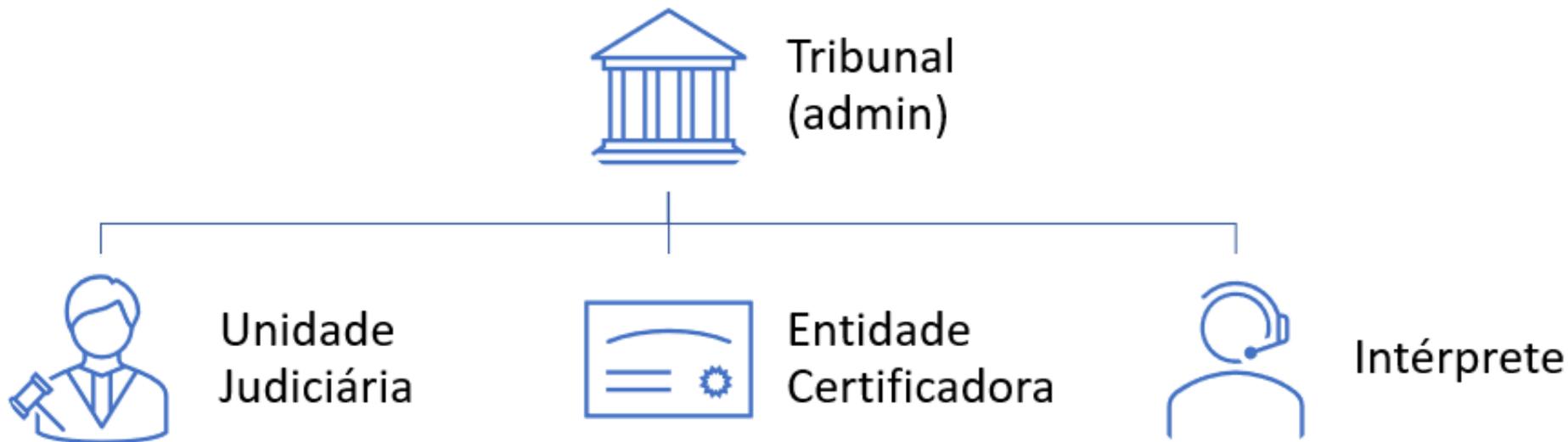


❖ Logotipo (nome da plataforma)



❖ Logotipo Simples (parte visual)

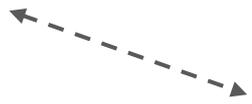
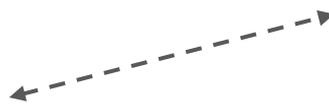
Projeto Interpretê



Plataforma



-  Entidade certificadora
-  Cadastro de Intérpretes
-  Cadastro de Audiências

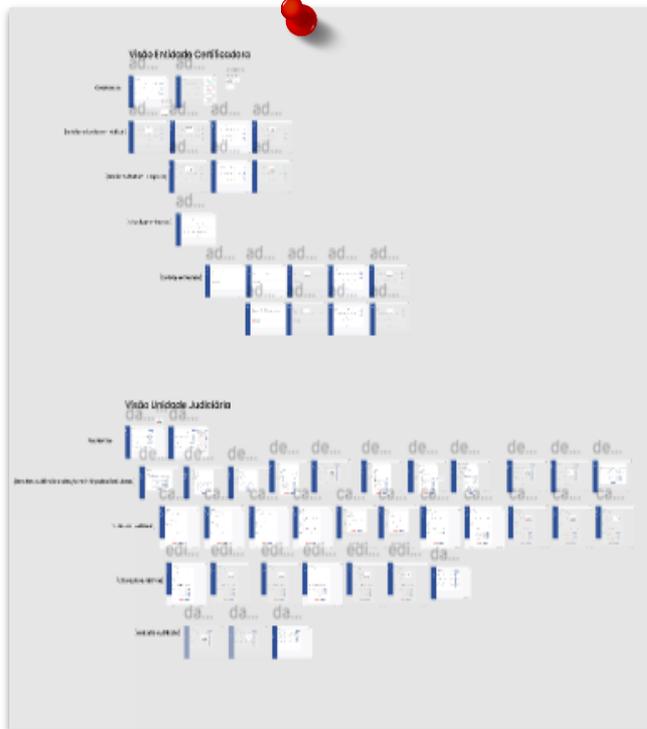


Protótipos





Design e Usabilidade



Selection colors

■	495057	100%
□	FFFFFF	100%
■	EFF2F7	100%
■	8E9397	100%
■	2D4F96	100%
■	000000	100%
■	5171A5	100%
■	74788D	100%
■	CBE8F6	100%
■	9BEEC2	100%
■	34C38F	100%
■	34C38F	10%
■	CED4DA	100%
■	FDB807	100%

data-table-audiencias

▶	bt-detalhes
▶	status
T	nome
T	data
T	325-250-1106
T	data
▶	icon-edit-delete
▶	Group 25
▶	Group 24
▶	bt-detalhes

Properties

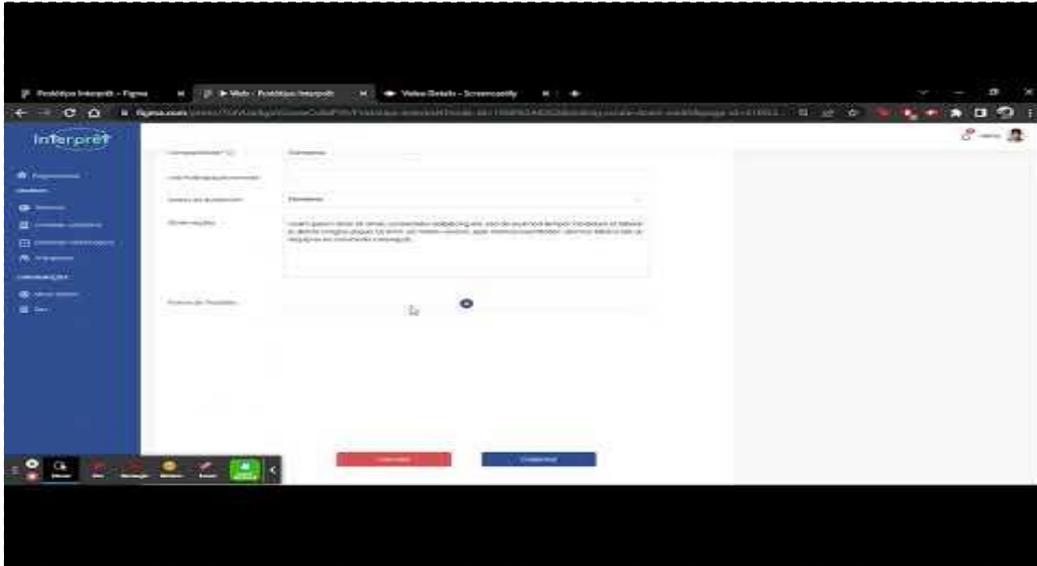
Width	1620px
Height	798px
Top	143px
Left	276px
Blend	Pass through

Code

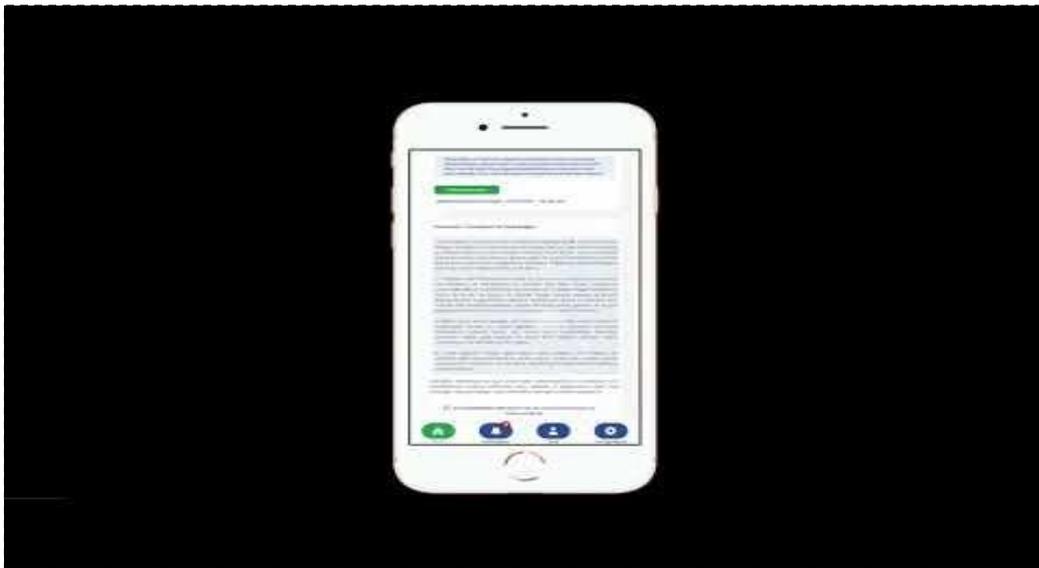
CSS <> ≡

```
/* table */  
position: absolute;  
width: 1620px;  
height: 798px;  
left: 276px;  
top: 143px;
```

Protótipo Web



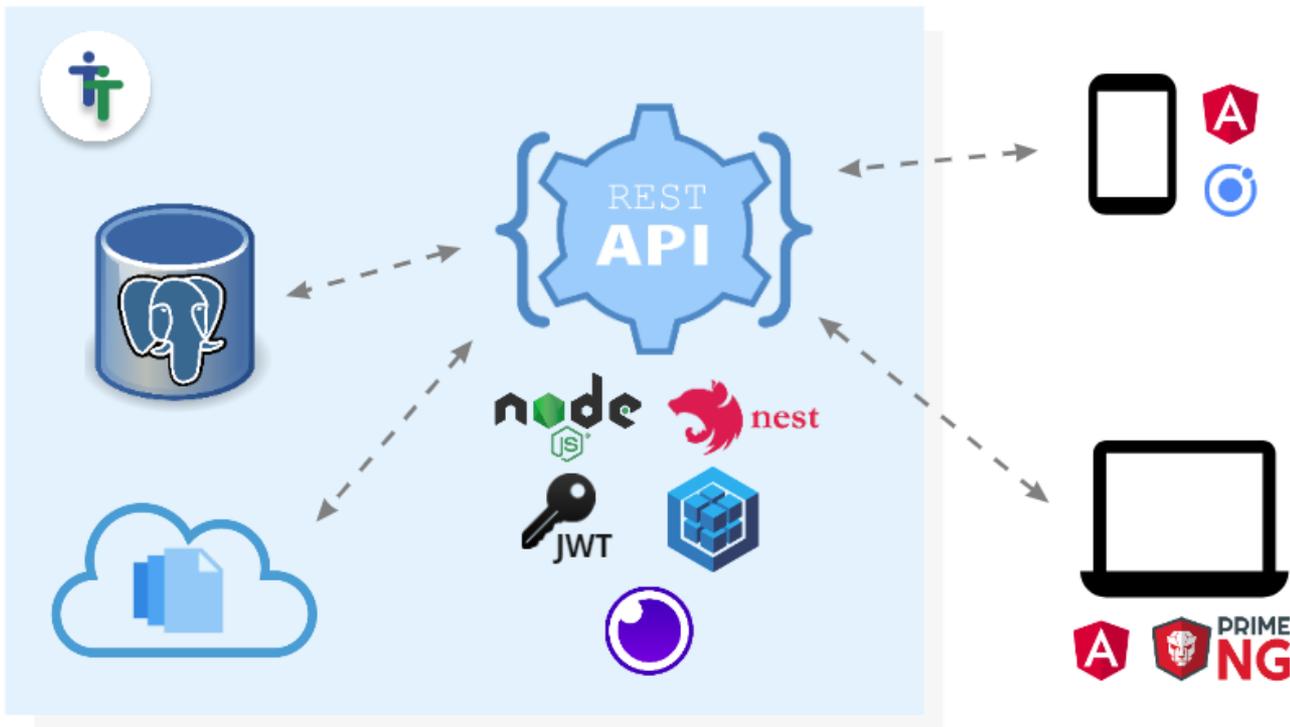
Protótipo Mobile



MVP



Arquitetura



Desenvolvimento

The image features a collage of development-related elements. At the top, the word "Desenvolvimento" (Development) is written in large white letters against a blue background filled with various "welcome" words in different languages (e.g., "Bienvenido", "Tervetuloa", "Bel Bonjou", "Belong").

On the left, a screenshot of a GitHub repository for "Interpret-APP / tcc_cco" is shown. The page includes a navigation bar with "Code", "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", and "Insights". The main content area displays the "Home" page, which states: "O repositório consiste em um monorepo, contendo três projetos:" followed by a list of projects: "API back-end", "Mobile", and "Front-web". Below this, there are links to tutorials for setting up the project, instructions on workflow, and first steps. A "Pages" sidebar on the right lists "Home", "Interprêr", "Endpoints", "Migração Database", "Primeiros Passos", "Setup do Projeto", and "Workflow".

On the right, a screenshot of a Trello board titled "PROJETO INTÉRPRETES" is displayed. The board is organized into three columns: "Backlog Front-end", "Em andamento" (In Progress), and "Concluído" (Completed). Each column contains a list of tasks with progress indicators and assignees. The "Em andamento" column includes tasks like "IBL-012 - Login/logout", "IBL-013 - Cadastro intérprete (mobile)", "IBL-016 - Agendamento de Audiências", "IBL-002 - Escrever artigo científico", "IBL-004 - Levantamento de requisitos", "IBL-005 - Pesquisa sobre interpretação forense", "IBL-010 - Estudar Postgres", "IBL-017 - Questionário Online", and "IBL-019 - Verificar ambiente online para deploy (BD e aplicação)". The "Concluído" column includes tasks like "IBL-001 - Aprovação pela entidade patrocinadora (JF)", "IBL-003 - Criar modelo relacional", "IBL-005 - Verificar requisitos LGPD", "IBL-018 - Entrevistas", "IBL-007 - Criar identidade visual", "IBL-008 - Layout web", and "IBL-009 - Layout mobile".

Servidor Web - API REST

A screenshot of the Visual Studio Code editor interface. The top status bar shows "Sun Jun 5 17:08:48". The Explorer sidebar on the left shows a project structure with folders like "BACK-END" and files like "postos.controller.ts". The main editor window displays the code for "PostosController" in "postos.controller.ts". The code includes imports for "@Controller", "@Post", "@Put", "@Body", "@Param", and "@UseGuards". It defines a "PostosController" class with two methods: "createPosto" and "updatePosto". The "createPosto" method is an asynchronous function that takes a "Posto" object and returns a "Posto" object. The "updatePosto" method is an asynchronous function that takes a "postId" string and a "Posto" object, and returns a "Posto" object. The code is written in TypeScript. A video call window is visible in the top right corner of the editor, showing two participants.

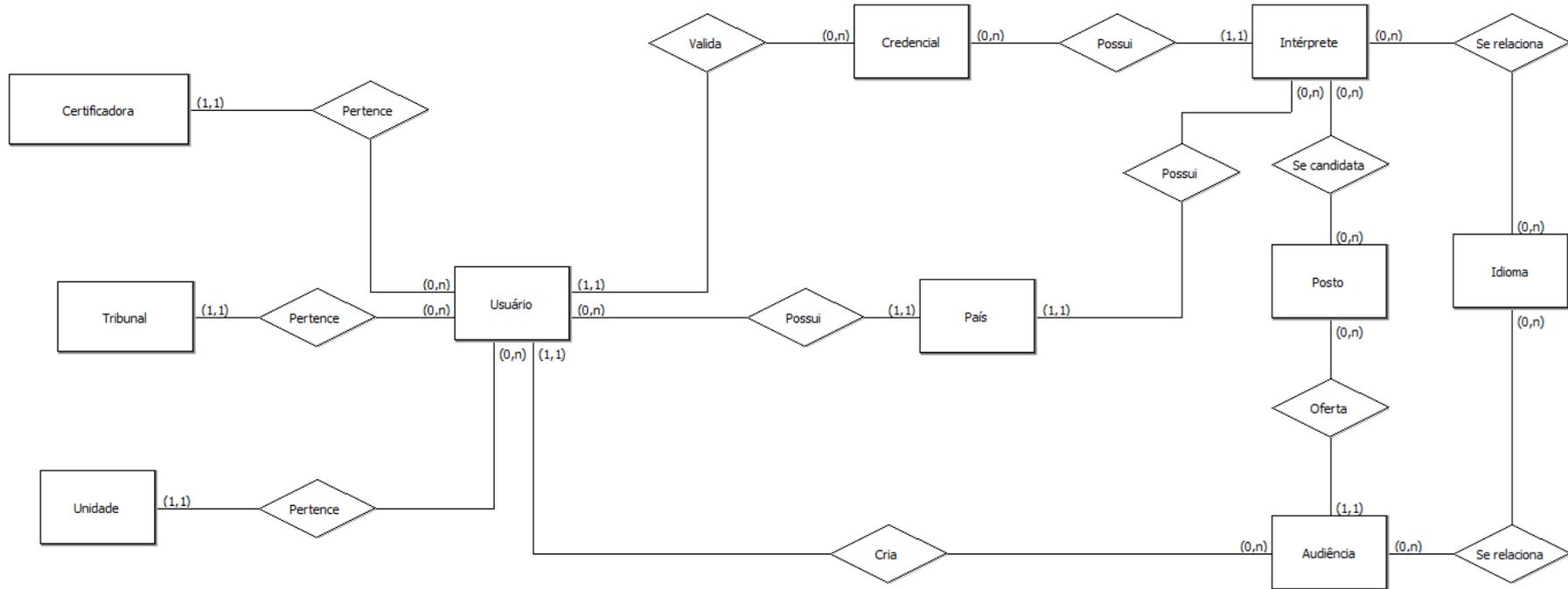
```
import { Controller, Post, Put, Body, Param, UseGuards } from '@nestjs/common';
import { PostosService } from './postos.service';
import { UsuariosJwtAuthGuard } from './guards/usuarios-jwt-auth.guard';

@Controller('postos')
export class PostosController {
  constructor(private readonly postosService: PostosService) {}

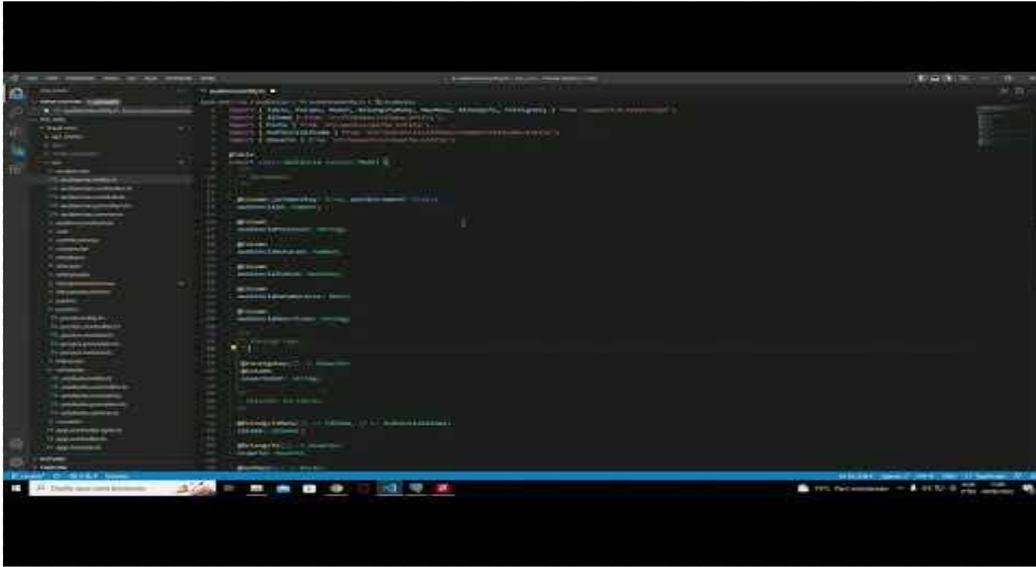
  @Post()
  @UseGuards(UsuariosJwtAuthGuard)
  async createPosto(@Res() response, @Body() posto: Posto) {
    const newPosto = await this.postosService.createPosto(posto);
    return response.status(HttpStatus.CREATED).json(newPosto);
  }

  @Put('/:postId')
  @UseGuards(UsuariosJwtAuthGuard)
  async updatePosto(@Res() response, @Body() posto: Posto, @Param('postId') postId: string) {
    await this.postosService.updatePosto(postoId, posto);
  }
}
```

Banco de Dados



Banco de Dados



Insomnia

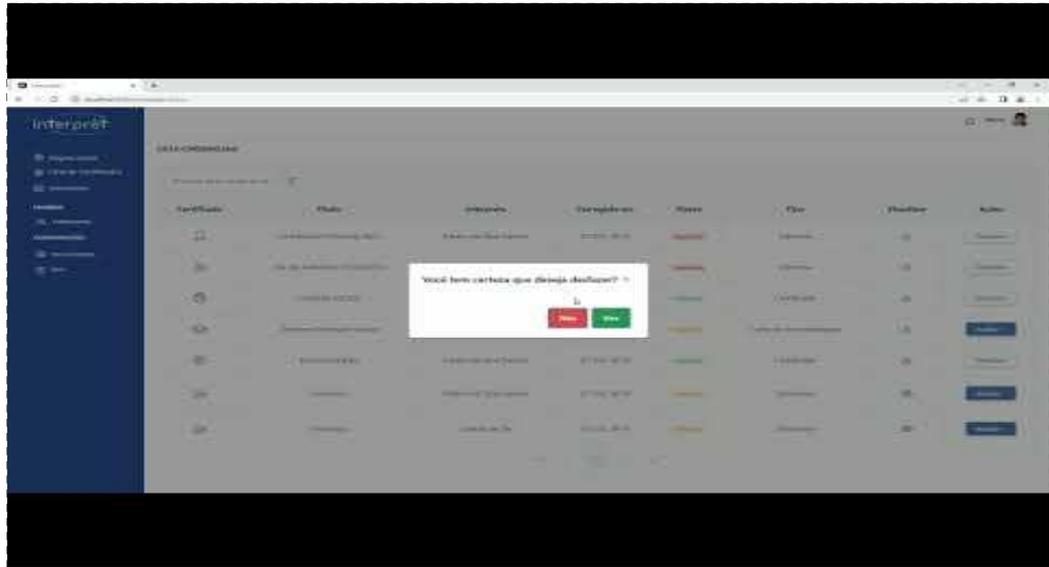


Sequelize

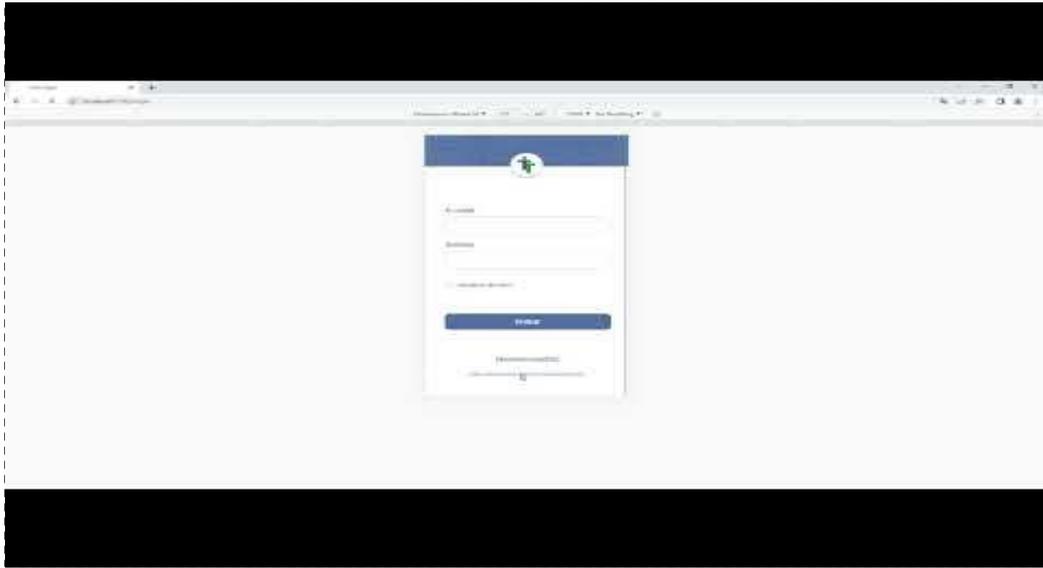


PostgreSQL

Cliente Web



Cliente Mobile



Obrigado!

