

# Projeto: Portal de controle de estudos.

---

Universidade Federal do Rio Grande do Norte

Disciplina: Desenvolvimento de Sistemas WEB I

Professor: Jair C. Leite

Aluno: André Augusto Fernandes

## 1. Introdução

---

Este projeto foi desenvolvido para a disciplina de Desenvolvimento de Sistemas WEB I, como meio de avaliar os conhecimentos adquiridos pelo aluno ao longo da disciplina.

## 2. Objetivo

---

Os requisitos da avaliação eram desenvolver um sistema WEB composto de Frontend, com a utilização das tecnologias Html, CSS e JavaScript, além de um servidor local, Backend, utilizando Node e Javascript para processar as requisições do cliente.

## 3. Proposta

---

Da experiência pessoal, veio a ideia para o projeto: a falta de uma tecnologia que permitisse organizar e controlar de forma ágil e eficiente o estudo para concursos. Foi aí que surgiu o **Editally**, um portal que se propõe a oferecer um ambiente de controle de estudos para processos seletivos, em que se faz necessária a verticalização das principais informações e conteúdo programático objetos do edital. O candidato poderá cadastrar seu edital de forma rápida e eficiente, e acompanhar sua evolução por meio de funcionalidades pensadas para facilitar seu estudo.

## 4. Instruções de uso

---

### 4.1. Instalar o Node.js

Para rodar o servidor local, será necessário instalar o Node.js na sua máquina. Prossiga à instalação através do seguinte link: [Node JS](#).

## 4.2. Baixar arquivos

Após a instalação do Node.js, baixe a pasta `/SERVIDOR` deste repositório para o local de sua preferência em seu computador.

## 4.3. Inicializar Servidor Local

Abra o Prompt de Comando dentro da pasta `/SERVIDOR` e execute o seguinte comando:  
`node meuservidor.js`

Caso não haja problemas, seu servidor local estará ativo.

## 4.4. Acessar Portal

Com o servidor rodando, abra o browser e acesse o seguinte endereço:  
`http://localhost:3000/`

*Voilà!!!* Caso todas as etapas acima tiverem sido processadas corretamente, você terá acesso ao Portal e poderá iniciar sua jornada de estudos.

A seguir, descreverei as principais funcionalidades do site e falarei sobre sua implementação. Vamos lá!?

# 5. /

---

## 5.1. `/login.html`

Ao acessar o endereço `http://localhost:3000/`, caso não haja uma sessão ativa, o usuário será direcionado à página de `/login.html`, na qual poderá entrar com suas credenciais e acessar a página principal (`/index.html`), ou acessar o link `Criar Conta` para se cadastrar.



### 5.1.1. Implementação

Ao clicar em `Entrar`, a página enviará as credenciais do usuário para conferência no servidor. Caso haja identidade entre a base de dados e os dados passados pelo usuário, o servidor devolverá um Token ao cliente, que será armazenado localmente. Em seguida, a página `/index.html` será chamada. Para gerar o token, foi utilizado módulo `JsonWebToken (JWT)`.

## 5.2. `/cadastro.html`

Ao acessar a página de Cadastro, o usuário poderá se cadastrar e tornar-se membro do portal, passando a ter acesso ao sistema de estudos.

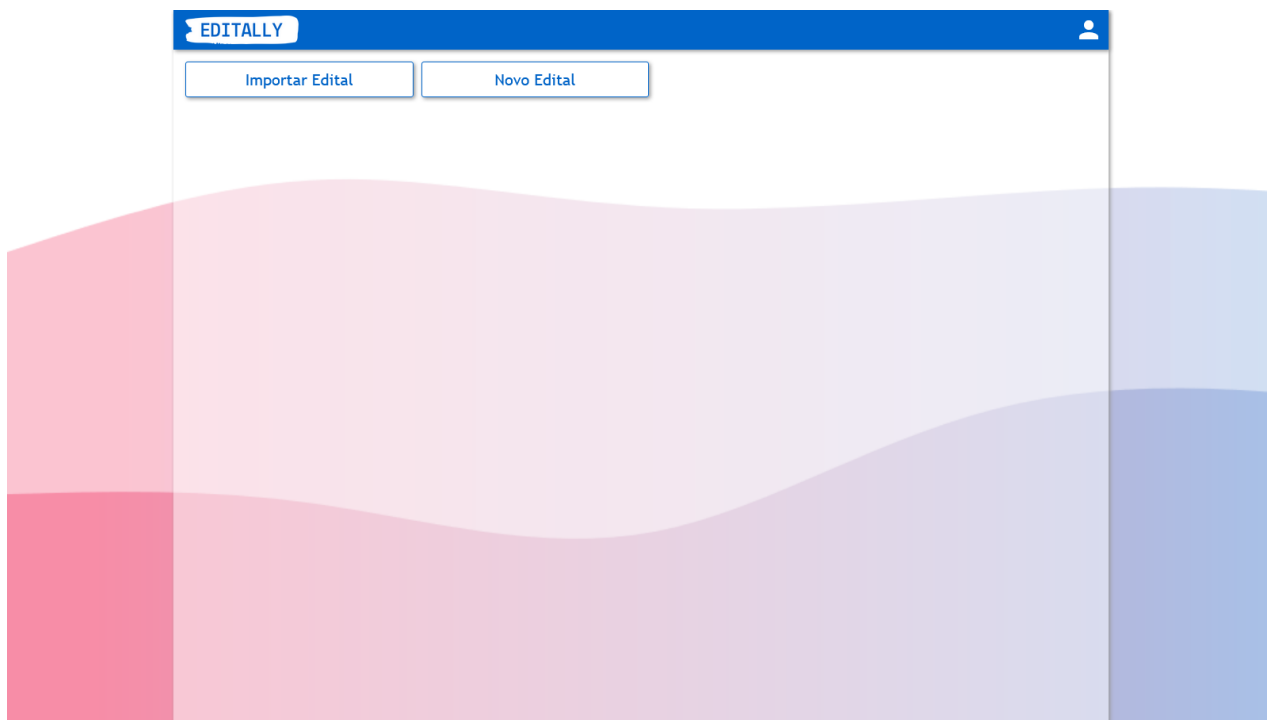


### 5.2.1. Implmentação

Ao clicar em Cadastro, as informações do usuário serão enviadas ao servidor, que irá inserir o novo membro na base de dados e devolver um Token válido para ser armazenado localmente. Em seguida, o usuário será direcionado para a página `/index.html`, na qual se dará efetivamente a utilização da página.

### 5.3. `/index.html`

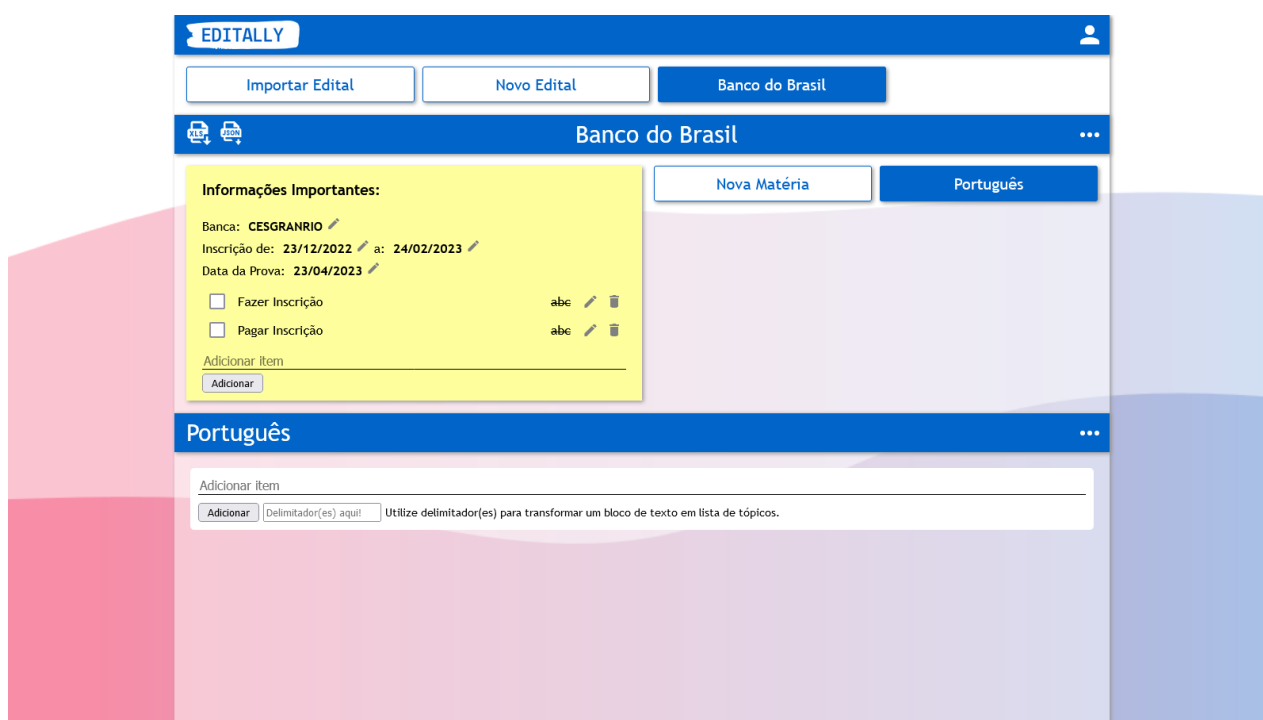
Uma vez logado no sistema, o usuário dará início à sua jornada de estudos, cadastrando um novo edital ou importando um criado anteriormente.



Uma vez cadastrado um novo edital, o usuário poderá proceder a preencher as principais informações sobre o certame e cadastrar as matérias do conteúdo programático.



Cada matéria terá uma lista que poderá ser preenchida com os tópicos cobrados no edital.



O candidato poderá inserir os tópicos um a um, ou deixar o sistema fazer o trabalho "sujo" por ele. Neste sentido, ele poderá simplesmente copiar, a partir do edital, o bloco inteiro de tópicos da matéria e, através do uso de um ou mais delimitadores, transformar o texto em uma lista de tópicos.



Uma vez completa a criação do edital, ou a qualquer momento, o estudante poderá exportá-lo para um arquivo **JSON** ou **Excel**, através dos ícones à esquerda na barra do

editais.

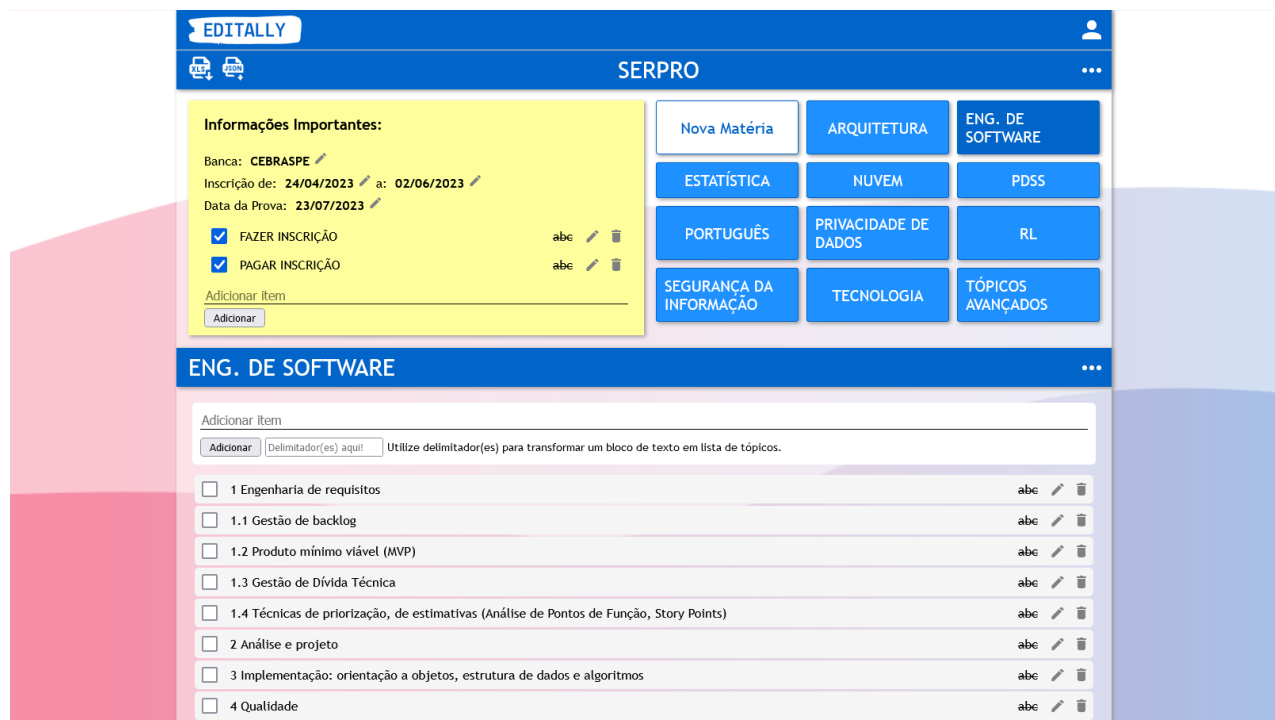
The screenshot displays the EDITALLY web application interface. At the top, there is a blue header with the 'EDITALLY' logo and a user profile icon. Below the header, a navigation bar contains three buttons: 'Importar Edital', 'Novo Edital', and 'Banco do Brasil'. The main content area is titled 'Banco do Brasil' and features a yellow box on the left with 'Informações Importantes:' containing details about the 'CESGRANRIO' bank exam, including the registration date (23/12/2022), exam date (24/02/2023), and exam date (23/04/2023). It also has checkboxes for 'Fazer Inscrição' and 'Pagar Inscrição', both of which are checked. To the right of this box are several blue buttons for subject selection: 'Nova Matéria', 'Atualidades do Mercado Financeiro', 'Conhecimentos Bancários', 'Estatística', 'Inglês', 'Matemática', 'Português', and 'T.I.'. Below these buttons is a section titled 'Atualidades do Mercado Financeiro' which includes a text input field for adding items and a list of six topics with checkboxes and edit/delete icons. The topics are: 1 - Os bancos na Era Digital: Atualidade, tendências e desafios; 2 - Internet banking; 3 - Mobile banking; 4 - Open banking; 5 - Novos modelos de negócios; and 6 - Fintechs, startups e big techs.

A importação e exportação de editais permite ao usuário administrar seu ambiente de estudos de forma ágil e eficiente.

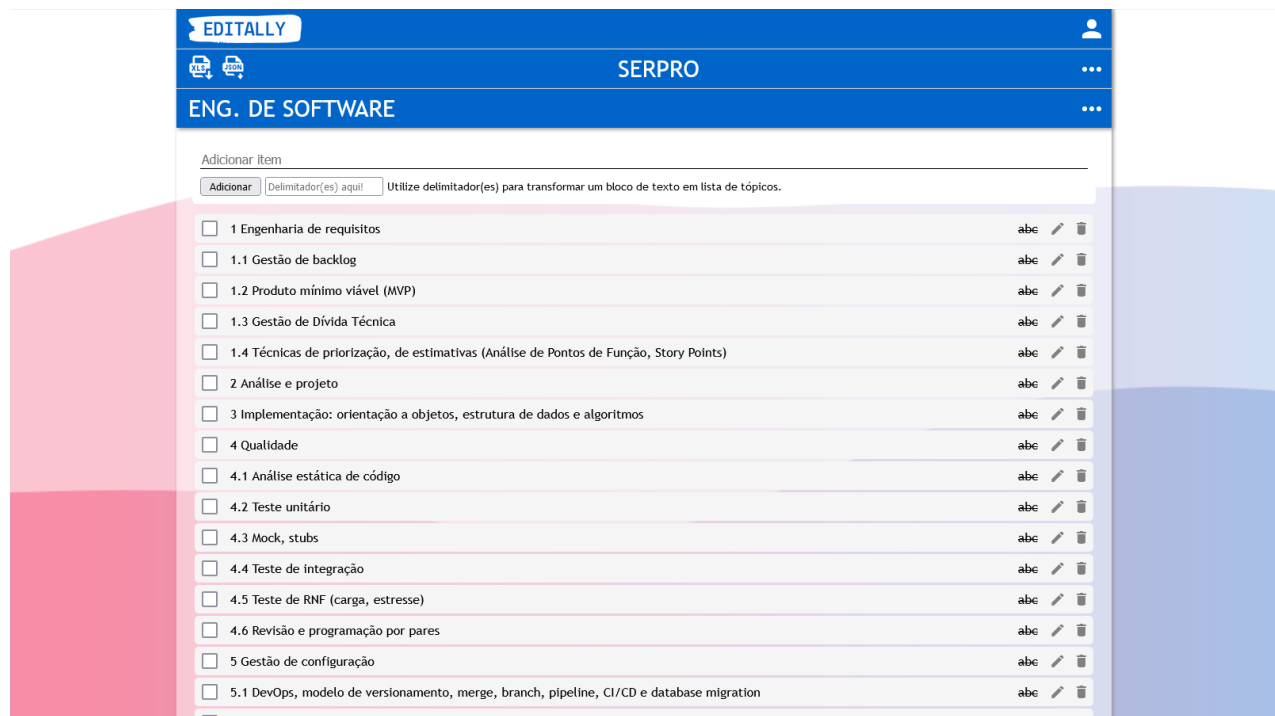
This screenshot shows the same EDITALLY web application interface as the previous one, but with additional tabs visible in the navigation bar: 'SERPRO' and 'TJ-RN'. The 'Banco do Brasil' tab is still selected. The layout and content of the main area are identical to the first screenshot, showing the exam information, subject selection buttons, and the list of topics for 'Atualidades do Mercado Financeiro'.

Ao rolar a página para baixo, algumas informações serão ocultadas para facilitar a exibição do conteúdo, enquanto outras de maior relevância, como os cabeçalhos de cada

container, serão empilhados e mantidos sempre visíveis.



Na imagem abaixo, podemos observar o comportamento ao rolar a página para visualizar os tópicos mais inferiores.




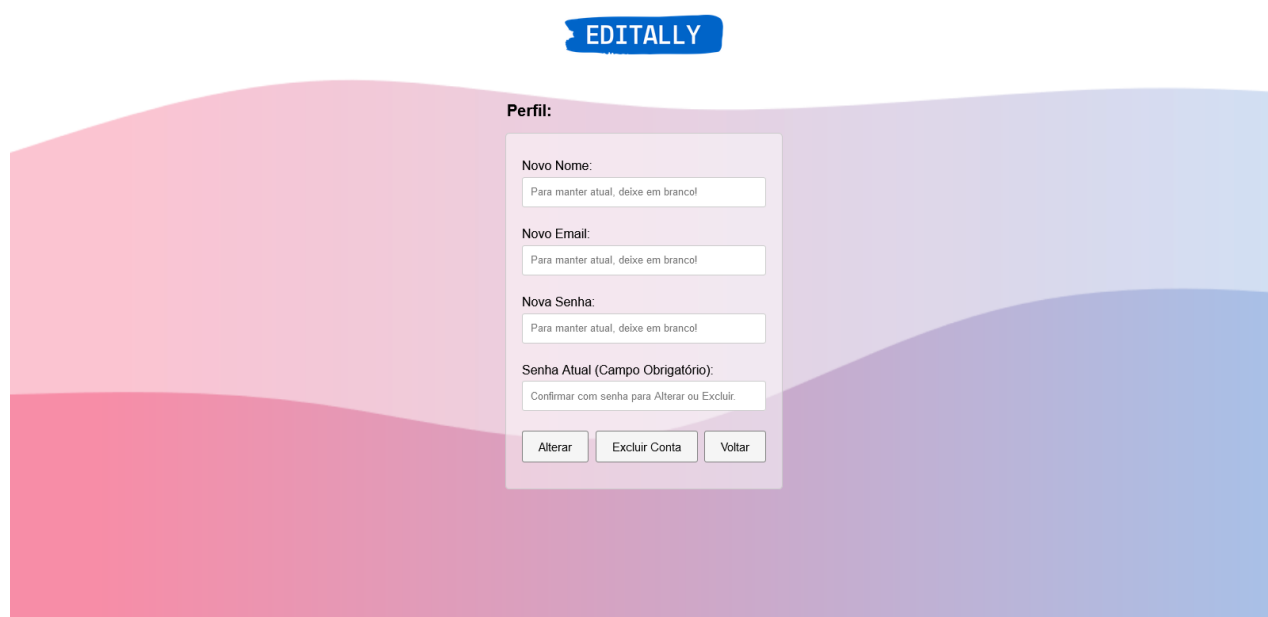
### 5.3.1. Implementação



A página `/index.html` opera da seguinte maneira: Ao ser inicialmente invocada, verifica a existência de um Token no armazenamento local. Caso haja um token, o mesmo será enviado ao servidor para validação. Uma vez válido, o usuário terá acesso à página e receberá as informações do usuário contidas no servidor. Recebidos os dados, o aluno poderá visualizá-los sem que novas requisições sejam feitas ao servidor. No entanto, cada vez que o estudante realizar alterações aos dados, tais mudanças serão enviadas ao servidor para gravação e posterior recuperação.

## 5.4. `/perfil.html`

A página `/perfil.html` poderá ser acessada através do menu  (usuário) em `/index.html`, ou invocada manualmente, caso haja um token válido no cliente. Aqui, o usuário poderá alterar suas informações de cadastro ou excluir sua conta.



A interface de perfil do usuário, intitulada "EDITALLY", apresenta um formulário para a edição de dados. O formulário, sob o título "Perfil:", contém os seguintes campos:

- Novo Nome:** Campo de texto com o placeholder "Para manter atual, deixe em branco".
- Novo Email:** Campo de texto com o placeholder "Para manter atual, deixe em branco".
- Nova Senha:** Campo de texto com o placeholder "Para manter atual, deixe em branco".
- Senha Atual (Campo Obrigatório):** Campo de texto com o placeholder "Confirmar com senha para Alterar ou Excluir".

Na base do formulário, há três botões de ação: "Alterar", "Excluir Conta" e "Voltar".

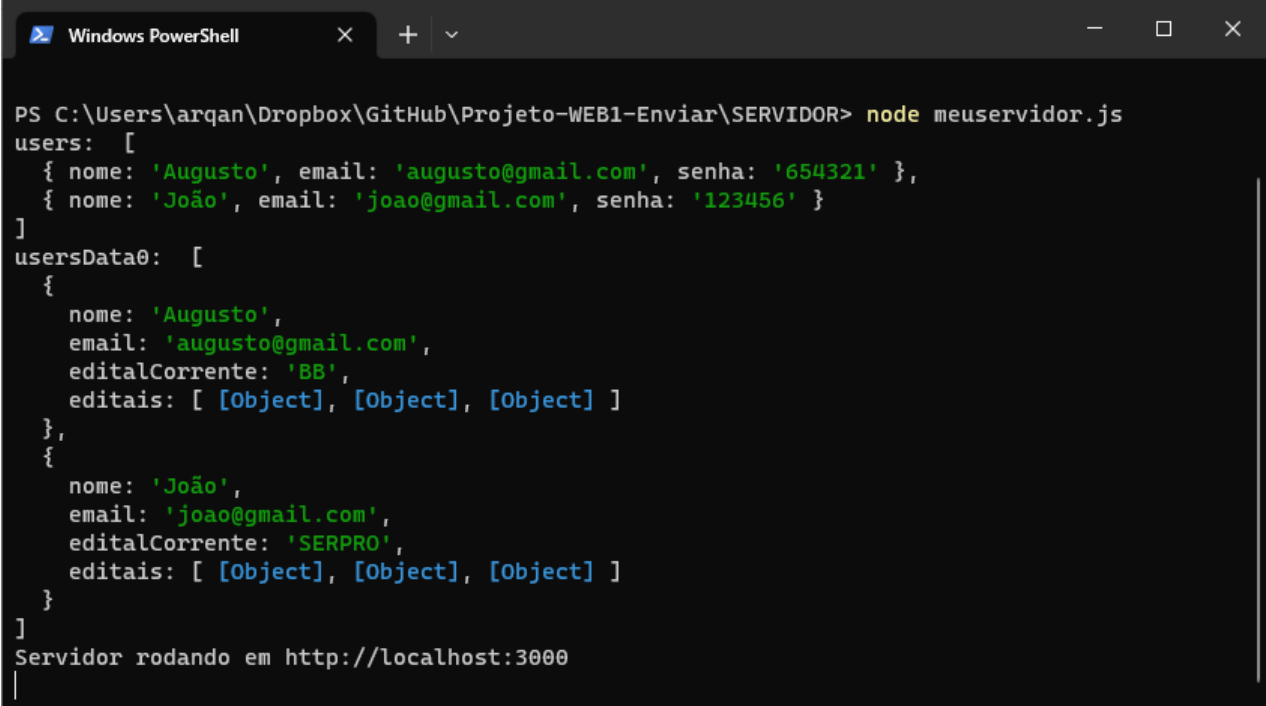
### 5.4.1. Implementação

Assim como nas implementações anteriores, aqui, ao processar a operação solicitada, será feita a validação do token enviado pela página ao servidor e, em seguida, o envio das informações para serem registradas.

## 6. Servidor

Para fazer a interação Frontend <-> Backend, foi criado um servidor local utilizando Node.js e Javascript.

O código para criar o servidor está implementado no arquivo `/SERVIDOR/meuservidor.js` e pode ser executado dentro da própria pasta com o seguinte comando: `node meuservidor.js`



```
PS C:\Users\arqan\Dropbox\GitHub\Projeto-WEB1-Enviar\SERVIDOR> node meuservidor.js
users: [
  { nome: 'Augusto', email: 'augusto@gmail.com', senha: '654321' },
  { nome: 'João', email: 'joao@gmail.com', senha: '123456' }
]
usersData0: [
  {
    nome: 'Augusto',
    email: 'augusto@gmail.com',
    editalCorrente: 'BB',
    editais: [ [Object], [Object], [Object] ]
  },
  {
    nome: 'João',
    email: 'joao@gmail.com',
    editalCorrente: 'SERPRO',
    editais: [ [Object], [Object], [Object] ]
  }
]
Servidor rodando em http://localhost:3000
```

O endereço do servidor local foi configurado como `http://localhost:3000`, que deverá ser utilizado no browser para acessar a página.

## 7. Tecnologias utilizadas

Para desenvolver o projeto, foram utilizados Html, CSS, Javascript e Node.js.

### 7.1. Frameworks

Não foram utilizados frameworks no projeto. Contudo, para continuar desenvolvendo o site, provavelmente será utilizado algum framework para implementar funcionalidades mais sofisticadas.

### 7.2. Bibliotecas/Módulos

Para implementar o sistema, foram utilizados os seguintes módulos pelo lado do servidor:



```
1  const http = require('http');  
2  const fs = require('fs');  
3  const path = require('path');  
4  const url = require('url');  
5  const jwt = require('jsonwebtoken');
```

Já pelo lado do cliente foram utilizadas algumas bibliotecas de ícones e módulo de conversão de arquivos.

## 8. Em desenvolvimento 🚧

Seguem em desenvolvimento as seguintes funcionalidades para versões futuras:

- Manipulação de listas;
- Medidor de progresso;
- Melhorar Responsividade;
- Recuperação de senha.

## 9. Conclusão

O projeto se provou de grande utilidade para solidificar os conhecimentos adquiridos durante o curso, proporcionando uma boa base de conhecimento e capacitação em sistemas WEB.