Ministério da Educação
Universidade Federal de Itajubá – *Itajubá*
XDES03 – Programação Web
Exercícios —Node com Express
Prof.: Phyllipe Lima Francisco

Instruções Gerais:
- Atividade poderá ser feita em dupla. Informar na entrega nome e matrícula da dupla.
- Entregar a solução completa em formato ".zip" pela tarefa do SIGAA.
- Horário de Entrega definido na Tarefa do SIGAA
- Formato de Entrega: Coloque os arquivos referentes ao *front e back* em um único diretório. Apague a pasta node_modules em ambos "front" e "back". Compacte esses diretórios em um único diretório com formato ".zip" e faça a entrega no SIGAA.
- Entrega em formato diferente de ".zip" não será aceita.
- O front já foi fornecido.
- A estrutura do back também foi fornecida. Sua solução será implementada no arquivo "index.js".

Objetivos: Familiarizar com Node e Express:

- Criar rotas com express
- Tratar requisições e retornar respostas
- Acessar e manipular arquivos com Node simulando base de dados.

Tarefa Única: Sistema de Cadastro de Usuários para visualizar as disciplinas dos cursos de computação da Unifei.

Nessa tarefa você deverá criar o "back" do sistema de cadastro, implementando as rotas necessárias. A Figura 1 contém a página de login. A Figura 2 a página de cadastro. O cadastro pode ser acessado clicando em "Cadastro"

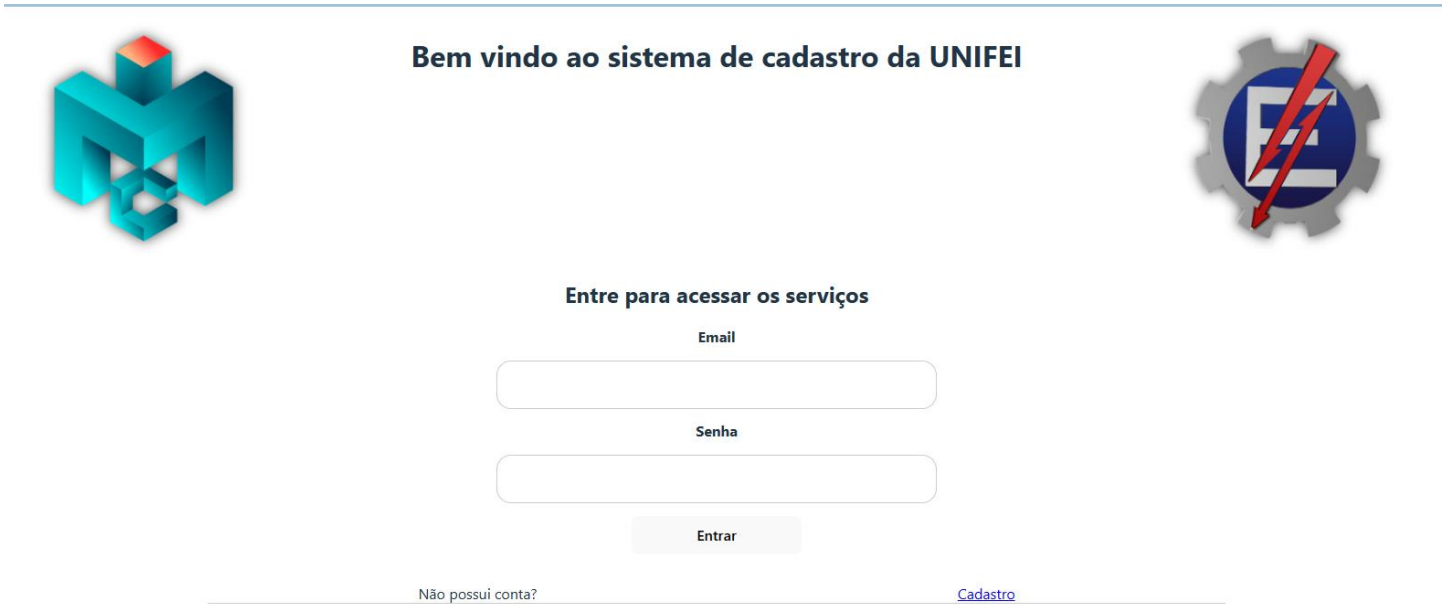

Figura 1

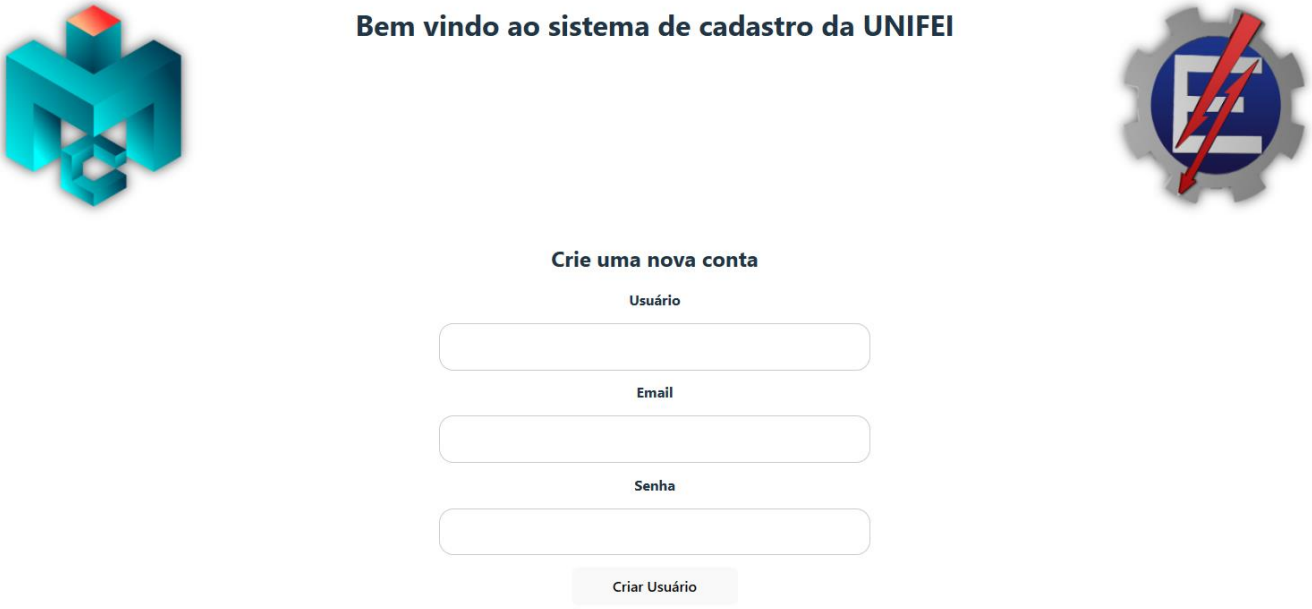

Figura 2

Funcionamento:

1 – Login.

Após feito o login, o sistema irá abrir uma página que irá listar as disciplinas como mostrado na Figura 3. O sistema já possui um usuário cadastrado. (note que se encontra no arquivo "banco-dados-usuario.json", mas a rota não está implementada.)
Email: jaum@teste.com.br
Senha: 123

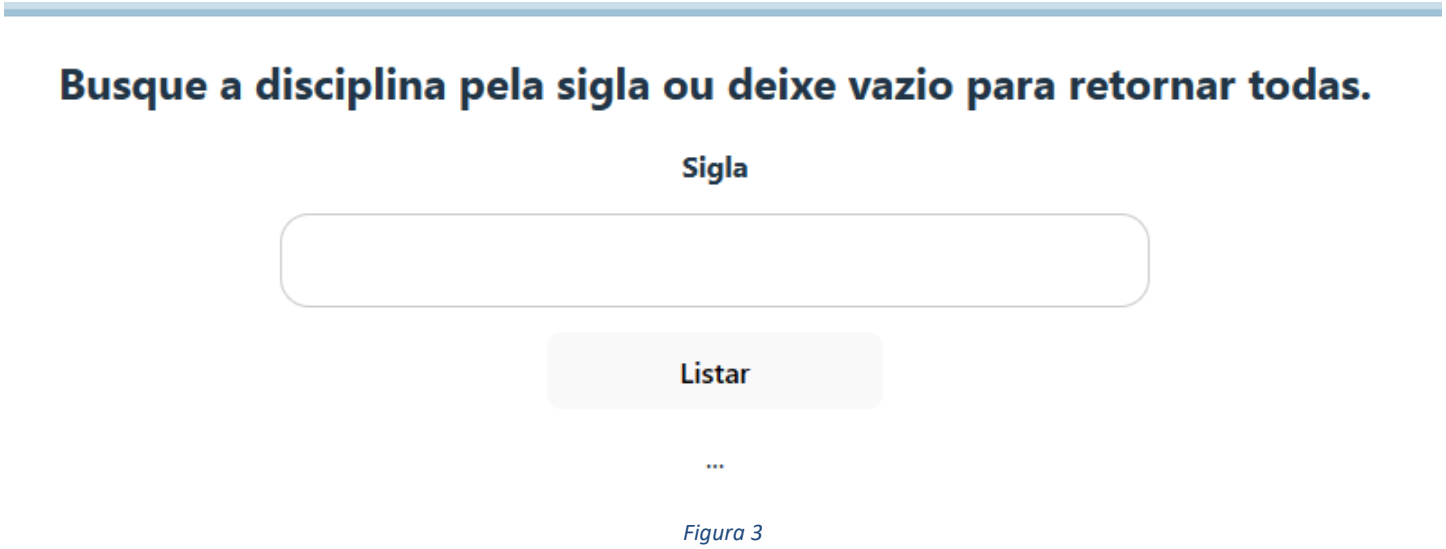

Figura 3

2 – Ao clicar no botão “Listar” deixando o campo “Sigla” vazio, todas as disciplinas serão retornadas. Essas disciplinas se encontram no arquivo “disciplinas.json”. A Figura 4 apresenta o retorno esperado (após o servidor implementado corretamente).

Figura 4

3 – Ao clicar no botão “Listar” colocando uma sigla válida, os dados dessa disciplina serão retornados como mostrado na Figura 5.

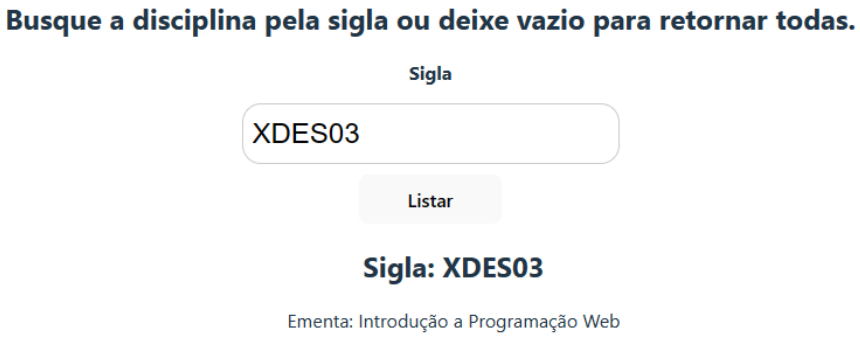

Figura 5

Ambiente:

1 - Executando o front

Execute o comando “npm i”, na pasta “front” para instalar todas as dependências necessárias. Para executar o “front”, digite o comando “npm run dev”. Esse comando deve ser executado na raiz do “front”, onde se encontra o arquivo “package.json”. Nesse momento o sistema não irá funcionar adequdamente pois não existe um servidor implementado (back).

2 – Executando o back

Execute o comando “npm i” na pasta “back” para instalar todas as dependências necessárias. Em seguida execute o comando “nodemon index.js”. Se o comando não funcionar, instale o “nodemon” de forma global com o comando “npm i -g nodemon”.

Requisito:

Sua tarefa consiste em codificar o arquivo “index.js” que se encontra no diretório “back” e implementar todos os requisitos.

1 – Rota “/login”. Codifique a rota “/login” que deverá tratar o verbo POST.
A rota deverá realizar as seguintes tarefas.

1.1 – Acessar o arquivo “banco-dados-usuario.json” e verificar se o usuário e senha estão corretos. Se estiver incorreto, deverá retornar a mensagem “`Usuario ou senhas incorretas.`”

1.2 – Acessar o arquivo “banco-dados-usuario.json” e verificar se o usuário existe. Se não existir, deverá retornar a mensagem “`Usuario com email ${email} não existe. Considere criar uma conta!.`”

1.3 – Acessar o arquivo “banco-dados-usuario.json” e verificar se usuário existe. Se existir e a senha estiver correta, deverá retornar a mensagem “`Autenticado com Sucesso`”

1.4 – Caso tenha sucesso, o front já está preparado para trocar para a página que irá listar as disciplinas. Por isso certifique que a mensagem segue o padrão especificado nos itens 1.1-1.3

2 – Rota "/create". Codifique a rota "/create" que deverá tratar o verbo POST.
A rota deverá realizar as seguintes tarefas.

1.1 – Acessar o arquivo "banco-dados-usuario.json" e verificar se o email já está cadastrado. Se estiver, retornar a mensagem "`Usuario com email ${email} já existe.`"

1.2 – Caso o email não esteja cadastrado, o usuário será criado com sucesso e inserido no banco de dados com um "id" incremental. Em seguida deverá retornar a mensagem "`Tudo certo usuario criado com sucesso.`"

1.3 – Caso tenha sucesso, o front já está preparado para mostrar o botão "Fazer Login" logo abaixo do cadastro. Por isso certifique que a mensagem segue o padrão especificado nos itens 1.1-1.2. A Figura 6 apresenta esse botão. Nesse exemplo o usuário com email jaum@teste2.com.br foi criado.

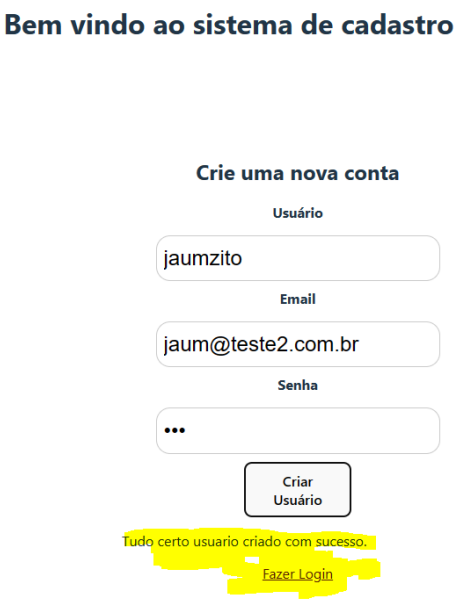

Figura 6

3 – Rota "/disciplinas". Codifique a rota de "/disciplinas" que deverá tratar o verbo GET.
A rota deverá realizar as seguintes tarefas.

1.1 – Acessar o arquivo "disciplinas.json" e o retornar como resposta da requisição todas as disciplinas. Exemplo: `return res.json(disciplinas);`

4 – Rota "/disciplinas/:sigla". Codifique a rota de "/disciplinas/:sigla" que deverá tratar o verbo GET e possui um "path parameter". O front já está preparado para enviar o parâmetro.
A rota deverá realizar as seguintes tarefas.

1.1 – Acessar o arquivo "disciplinas.json" e o retornar como resposta da requisição apenas a disciplina referente a sigla digitada no "forms". Para isso será

necessário varrer o arquivo “disciplinas.json” e retornar apenas o JSON referente a disciplina.

1.2 – Caso nenhuma sigla seja encontrada, basta devolver a mensagem: “`Sigla Não Encontrada!`”

Dica: Quando retornar algo relacionado a erro, tente especificar um código (status code) iniciando com “4”.
Exemplo de uma mensagem de erro:

```
res.status(403).send(`Usuário não encontrado`);
```