

# ***ENTREGA DE PRODUTO FINAL***

## ***Tema : Escola Conectada***

### ***GRUPO 9***

NOME	MATRÍCULA
Pedro Silva Seabra de Oliveira	2021200197
João Pedro Oliveira de Melo Mota Gonçalves	2021200217
Andrei Nunes	2021200485
Davi Daniel da Veiga Marques	2021200431
Érico Gaudencio da Silva	2020101292
Wellington Alves da Silva	2020101708

### ***Introdução***

O projeto **Escola Conectada** foi desenvolvido com o objetivo de criar uma plataforma intuitiva e eficiente para facilitar a comunicação e gestão em escolas públicas locais. O sistema permite que alunos, professores, pais e a administração escolar tenham acesso rápido a informações relevantes, promovendo uma gestão mais transparente e uma maior integração da comunidade escolar. Utilizando React no frontend e Flask no backend, a aplicação realiza uma integração robusta através de uma API, garantindo uma experiência de usuário fluida e dinâmica.

---

# 1. Funcionalidades do Sistema

## 1.1. Tela de Login


O backend implementa a lógica para autenticar os usuários com base no e-mail e senha. Se as credenciais forem válidas, ele retorna o papel do usuário (student ou teacher) e um status 200. Caso contrário, retorna 401 com uma mensagem de erro.

A página de login permite que os usuários façam autenticação no sistema. Existem dois papéis (roles):

- **Aluno:** Apenas visualiza disciplinas e o calendário.
- **Professor:** Pode criar, editar e excluir disciplinas.

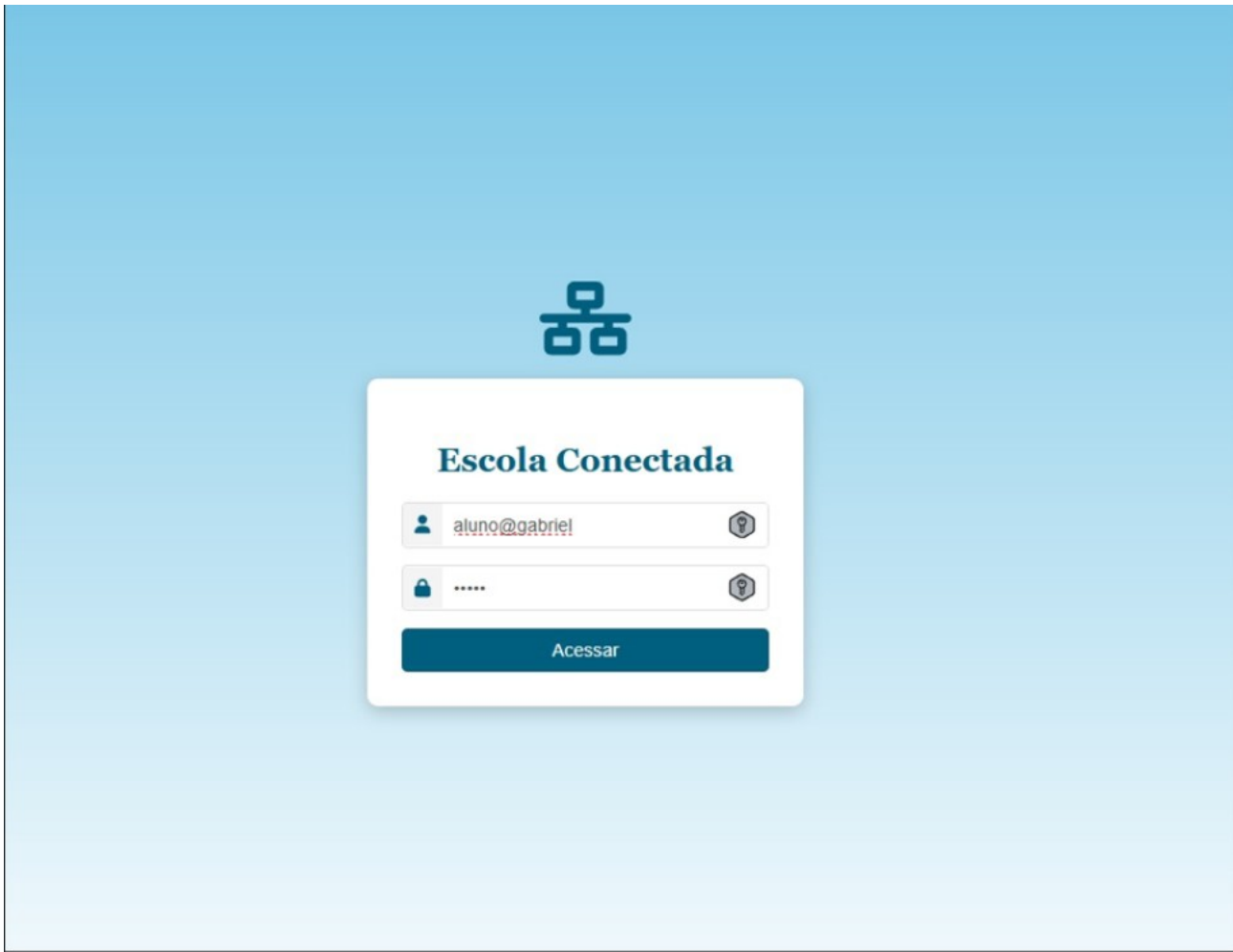
*Após o login, o sistema redireciona o usuário para a página apropriada, dependendo do seu papel.*

### ***LOGIN DO PROFESSOR***



The image shows a login form for a professor. At the top, there is a blue icon representing a school or network. Below it, the text "Escola Conectada" is displayed in a bold, dark blue font. The form consists of two input fields: the first for the email address, which contains "professor@fabiohenriquesilva", and the second for the password, which is masked with dots. Both fields have a small blue icon on the left and a key icon on the right. Below the input fields is a large blue button labeled "Acessar".

## *LOGIN DO ALUNO*

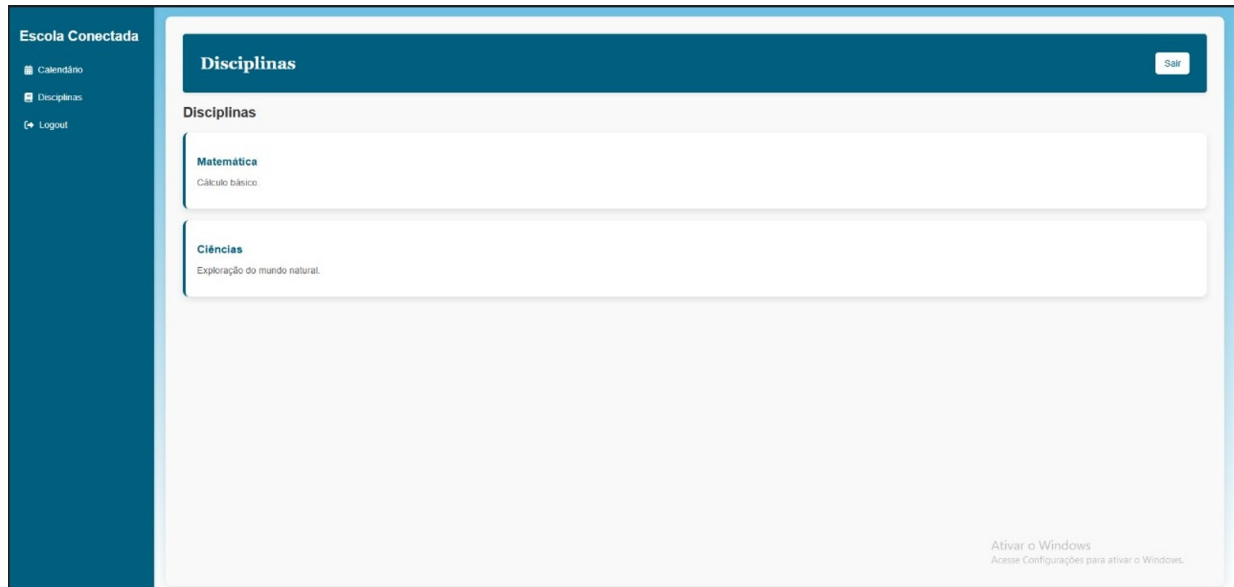


### **1.2. Página de Disciplinas**

A página de disciplinas exibe uma lista de disciplinas disponíveis.

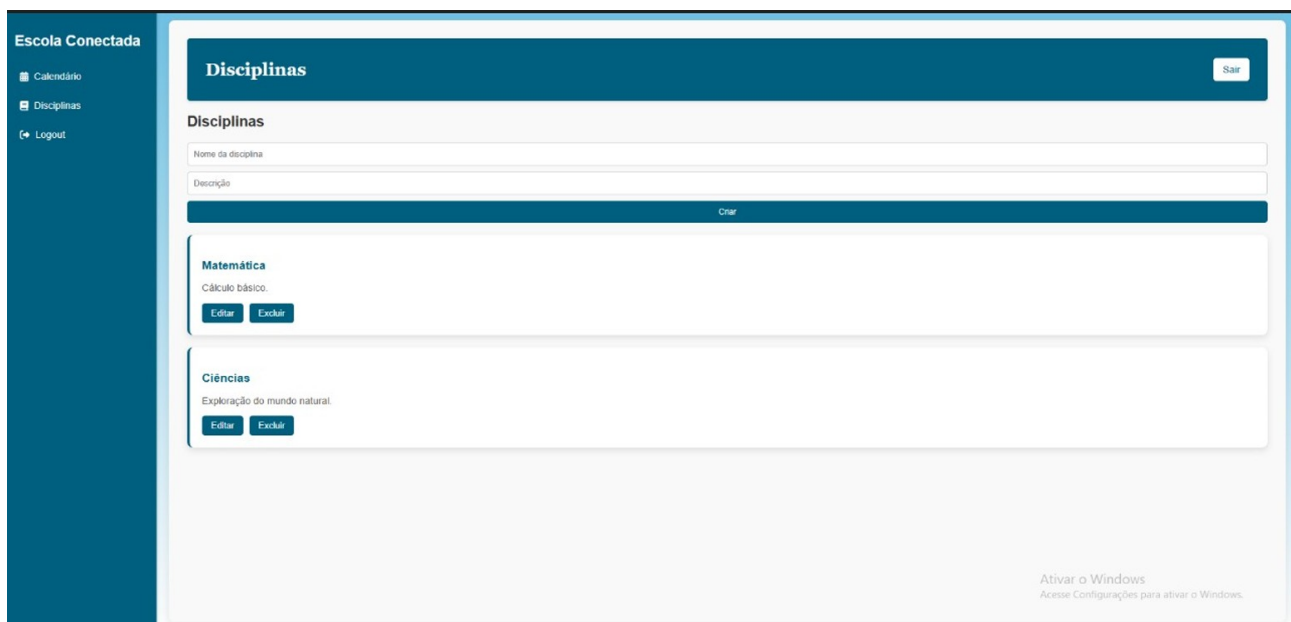
- **Professores** podem:
  - Criar novas disciplinas.
  - Leitura e visualização das disciplinas.
  - Editar disciplinas existentes.
  - Excluir disciplinas.
- **Alunos** podem:
  - Visualizar a lista apenas.

- ***VISÃO DOS ALUNOS AO SELECIONAREM “DISCIPLINAS”***



---

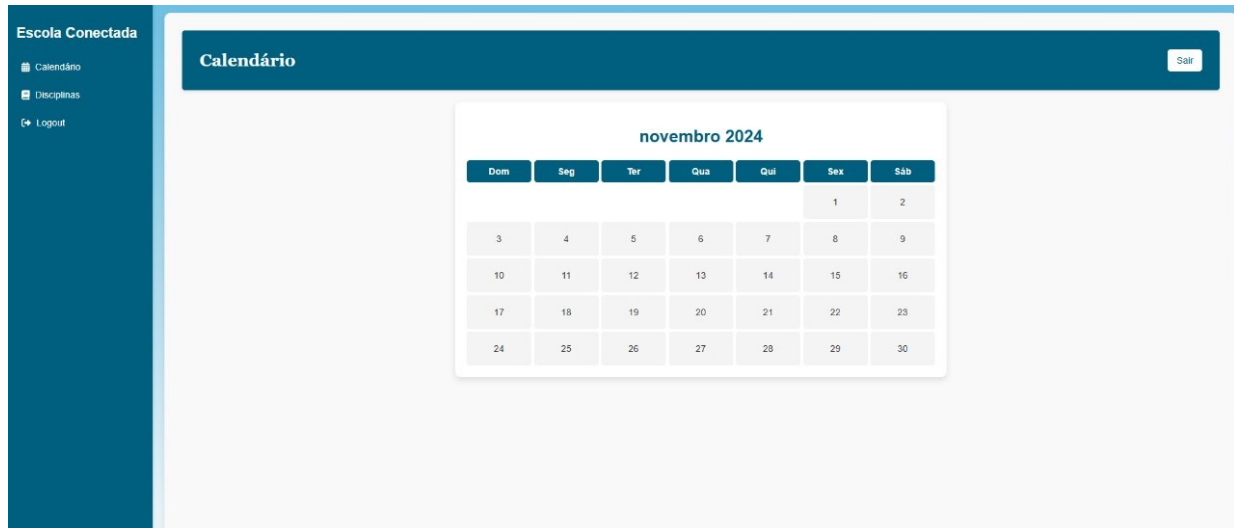
***VISÃO DOS PROFESSORES AOS SELECIONAREM “DISCIPLINAS”***



## 1.3. Página de Calendário

A página de calendário exibe um calendário visual para os usuários. É responsiva e possui um efeito de hover ao passar o mouse pelos dias. Por enquanto, não há funcionalidade de clique.

### *VISÃO DE TODOS USUÁRIOS AO ACESSAREM O CALENDÁRIO*



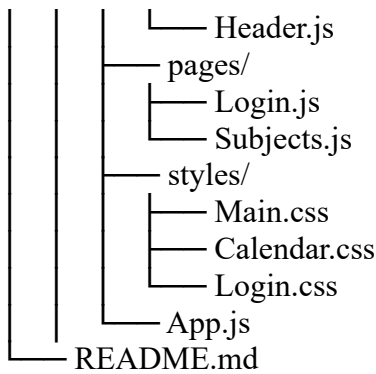
## 2. Tecnologias Utilizadas

- **Frontend:** React, CSS.
- **Backend:** Flask, Flask-CORS.
- **Comunicação:** API REST utilizando Axios.
- **Estilização:** CSS puro com foco em responsividade e design moderno.

## 3. Estrutura do Projeto

A estrutura de diretórios do projeto foi organizada da seguinte forma:

```
css
Copiar código
projeto-escola-conectada/
├── backend/
│   ├── app.py
│   └── requirements.txt
├── frontend/
│   ├── public/
│   ├── src/
│   │   ├── components/
│   │   ├── Calendar.js
│   │   └── Sidebar.js
```



---

## 4. Backend

O backend foi implementado em Flask, com uma API REST para autenticação de usuários e gerenciamento de disciplinas. Ele inclui as seguintes rotas:

### 4.1. Rota de Login

Rota responsável por autenticar o usuário com base no email e senha.

- Método: POST
- URL: /login

#### *Código:*

```
@app.route('/login', methods=['POST'])
def login():
    data = request.get_json()
    email = data.get("email")
    password = data.get("password")

    user = next((u for u in users if u["email"] == email and u["password"] == password), None)
    if user:
        return jsonify({"message": "Login successful", "role": user["role"], "email": email}), 200

    return jsonify({"message": "Invalid credentials"}), 401
```

### 4.2. Rotas para Gerenciamento de Disciplinas

Essas rotas são responsáveis por listar, criar, editar e excluir disciplinas.

#### *Código:*

```
@app.route('/subjects', methods=['GET', 'POST'])
def handle_subjects():
    if request.method == 'GET':
        return jsonify(subjects), 200
```

```

if request.method == 'POST':
    data = request.get_json()
    new_subject = {
        "id": len(subjects) + 1,
        "name": data.get("name"),
        "description": data.get("description"),
    }
    subjects.append(new_subject)
    return jsonify(new_subject), 201

@app.route('/subjects/<int:subject_id>', methods=['PUT', 'DELETE'])
def modify_subject(subject_id):
    subject = next((s for s in subjects if s["id"] == subject_id), None)
    if not subject:
        return jsonify({"message": "Subject not found"}), 404

    if request.method == 'PUT':
        data = request.get_json()
        subject["name"] = data.get("name", subject["name"])
        subject["description"] = data.get("description", subject["description"])
        return jsonify(subject), 200

    if request.method == 'DELETE':
        subjects.remove(subject)
        return jsonify({"message": "Subject deleted"}), 200

```

---

## 5. Frontend

### 5.1. Página de Login

A página de login foi implementada em React e utiliza a biblioteca Axios para enviar as credenciais ao backend. O design foi estilizado com CSS, incluindo o ícone de "Escola Conectada".

#### *Código:*

```

const handleLogin = async (e) => {
    e.preventDefault();
    try {
        const response = await axios.post('http://127.0.0.1:5000/login', { email, password });
        const { role, email } = response.data;
        localStorage.setItem('role', role);
        localStorage.setItem('email', email);
        navigate(role === 'teacher' ? '/subjects' : '/calendar');
    } catch (error) {
        setError('Credenciais inválidas');
    }
};

```

---

## 5.2. Página de Disciplinas

A página de disciplinas exibe a lista de disciplinas, com diferentes permissões para alunos e professores:

- **Professor:** Botões de "Criar", "Editar" e "Excluir" são exibidos.
  - **Aluno:** Apenas a lista de disciplinas é exibida.
- 

## 5.3. Página de Calendário

O calendário foi implementado em React com um layout grid para exibir os dias do mês atual. É responsivo e possui um efeito de hover ao passar o mouse.

### *Código:*

```
<div className="calendar-grid">
  <div className="day-name">Dom</div>
  <div className="day-name">Seg</div>
  <div className="day-name">Ter</div>
  <div className="day-name">Qua</div>
  <div className="day-name">Qui</div>
  <div className="day-name">Sex</div>
  <div className="day-name">Sáb</div>
  {generateCalendar()}
</div>
```

---

## 6. Estilização (CSS)

O design do sistema foi implementado com **CSS puro**, com foco em:

- **Responsividade:** Ajusta o layout para dispositivos menores.
- **Hover:** Adiciona feedback visual ao passar o mouse.

### *CSS para o Calendário:*

```
.day:hover {
  background: #6ec1e4;
  color: white;
  transform: scale(1.1);
}
```

---



## 7. Testes

### 7.1. Testes de Funcionalidade

- Testes realizados para confirmar que apenas professores têm acesso às funcionalidades de criação, edição e exclusão.
- Testes de login para verificar as credenciais de alunos e professores.

### 7.2. Testes Visuais

- Responsividade foi validada em diferentes tamanhos de tela.
- Efeitos de hover verificados no calendário.

---

## Conclusão

O projeto "Escola Conectada" proporciona uma interface intuitiva e funcional, segmentando eficientemente as permissões entre alunos e professores, o que garante uma navegação personalizada e segura para cada tipo de usuário. Com a integração sólida entre backend e frontend, o sistema estabelece uma base robusta para futuras expansões, como a inclusão de funcionalidades adicionais no calendário, melhorias no gerenciamento de notificações e até a implementação de um banco de dados completo para armazenamento de dados em tempo real. Dessa forma, a plataforma tem potencial para evoluir e atender cada vez mais às necessidades da comunidade escolar, promovendo um ambiente educacional mais organizado, acessível e interativo.

***LINK DO REPOSITÓRIO NO GITHUB PARA VALIDAÇÃO:***

<https://github.com/seabra27/escola-conectada.git>