

## **1. Visão Geral do Projeto**

O **Projeto MapaUESC** é uma aplicação web destinada a fornecer um mapa aprimorado e interativo do campus da Universidade Estadual de Santa Cruz (UESC).

O objetivo é criar uma ferramenta de navegação que auxilie visitantes, alunos e funcionários a localizar pontos de interesse, salas, auditórios e obter informações detalhadas sobre os prédios do campus.

## **2. Contexto do Projeto**

Este projeto está sendo desenvolvido como parte da disciplina **PROGRAMAÇÃO PARA APLICAÇÕES WEB-2025.2.**

**Professor:** Jorge Lima de Oliveira Filho

**Equipe de Desenvolvimento (Time 3):**

- Maria Eduarda
- João Fidelis
- Lucas Ceu
- Henrique Daniel
- Sara Raquel
- Ryan

O projeto é estruturado em Sprints com entregas parciais ao longo do semestre, culminando em uma apresentação final.

## **2. Funcionalidades Planejadas**

O escopo do mapa aprimorado, conforme detalhado nos arquivos de planejamento (service-details.html e starter-page.html), inclui:

- **Mapa Interativo:** Utilização de uma imagem estática do mapa da UESC como base para interações.
- **Localização Detalhada:** Informações sobre auditórios (ex: Paulo Souto), detalhes por andar dos prédios (torres e pavilhões) e numeração de salas.
- **Interface Intuitiva:** Uma barra lateral com nomes de locais e modais de informação que aparecem ao clicar ou passar o mouse.
- **Navegação e Rotas:**
- Exibição da localização atual e em tempo real do usuário.
- Cálculo de rotas de origem-destino (com ponto de partida padrão na guarita).
- **Feedback:** Uma área de contato para sugestões e atualizações do mapa.

## **4. Pilha de Tecnologia (Stack)**

Foi identificada uma discrepância entre a arquitetura planejada (descrita nos materiais do curso) e a implementação atual (o código-fonte fornecido).

### **Implementação Atual (Código Fornecido)**

O código-fonte atual está estruturado como uma **aplicação monolítica em Django**.

- **Backend: Django**
- **Database: SQLite** (configuração padrão)
- **Frontend: Templates Django (DTL)**, utilizando HTML, CSS e JavaScript.
- **Tema/Base:** O layout é baseado no template "iPortfolio" da BootstrapMade.

## **Arquitetura Planejada (Sprints do Curso)**

O planejamento das Sprints, documentado nos materiais da disciplina e no arquivo service-details.html, descreve uma arquitetura de microserviços ou SPA (Single Page Application):

- **Frontend (Sprints 1 e 2): Angular**
- **Backend (Sprint 3): Spring Boot**

**Nota:** A aplicação Django existente parece servir como base inicial ou protótipo, enquanto o plano de curso prevê o desenvolvimento de uma nova aplicação usando Angular e Spring Boot.

## 5. Estrutura do Projeto (Django)

A estrutura do código Django fornecido segue o padrão de projeto do framework:

```
mapauesc-main/
|   └── mapauesc/      # Configuração do Projeto Django
|       |   ├── settings.py    # Configurações (Apps, DB, Static)
|       |   ├── urls.py     # URLs principais do projeto
|       |   ├── wsgi.py
|       |   └── asgi.py
|
|   └── core/          # App Django principal
|       |   ├── models.py    # Define o modelo 'Portfolio'
|       |   ├── views.py     # Funções de view (index, contact, etc.)
|       |   ├── forms.py     # Define o 'ContactForm'
|       |   ├── admin.py     # Registro do 'Portfolio' no Admin
|       |   ├── migrations/   # Migrações do banco
|       |       |   └── 0001_initial.py
|       |   ├── templates/core/ # Templates HTML
|       |       |   ├── index.html
|       |       |   ├── service-details.html
|       |       |   ├── contact.html
|       |       |   ...
|       |   └── static/       # Arquivos Estáticos (CSS, JS)
|           |   ├── css/
|           |   ├── js/
|           |   ├── img/
|           |   └── vendor/
|
|   └── .gitignore
|
└── manage.py        # Utilitário de gerenciamento
└── README.md        # Documentação original
```

## 6. Instalação e Execução (Versão Django)

Para executar a versão Django do projeto, siga os passos abaixo (baseado no README.md original):

### 1. Clone o repositório:

```
git clone https://github.com/jmfsantos-sys/mapauesc.git  
cd mapauesc
```

### 2. Crie e ative um ambiente virtual:

- Windows

```
python -m venv venv  
venv\Scripts\activate
```

- Linux/macOS

```
python -m venv venv  
source venv/bin/activate
```

### 3. Instale as dependências:

```
pip install -r requirements.txt
```

### 4. Rode as migrações do banco de dados:

```
python manage.py migrate
```

### 5. Crie um superusuário (opcional, para /admin):

```
python manage.py createsuperuser
```

### 6. Execute o servidor de desenvolvimento:

```
python manage.py runserver
```

### 7. Acesse <http://127.0.0.1:8000/> no seu navegador.



## 8. Análise do Código e Pontos de Atenção

- **Configurações de Produção:** O arquivo settings.py está em modo de depuração (DEBUG = True) e contém uma SECRET\_KEY de placeholder. Isso não é seguro para produção.
- **Configuração de Email:** O backend de e-mail está configurado para o GMail, mas usa credenciais de placeholder (seuemail@gmail.com, sua\_senha). Além disso, a view contact está programada para enviar e-mails para seuemail@dominio.com. Isso precisa ser configurado corretamente para que o formulário de contato funcione.
- **Conflito de Views/URLs:** Existe um conflito de lógica nos arquivos mapauesc/urls.py e core/views.py.
  - Ambos os arquivos definem uma função/rota portfolio\_details duas vezes.
  - Uma rota é path('portfolio/<int:id>', ...) e espera um ID.
  - A outra rota é path('portfolio-details/', ...) e não passa um ID.
  - No entanto, ambas as rotas apontam para a *mesma* função views.portfolio\_details, que (na sua segunda definição) espera um id. Isso significa que acessar a URL /portfolio-details/ provavelmente resultará em um erro.