

## ATIVIDADE

## PROJETO INTEGRADOR - SITUAÇÃO PROBLEMA

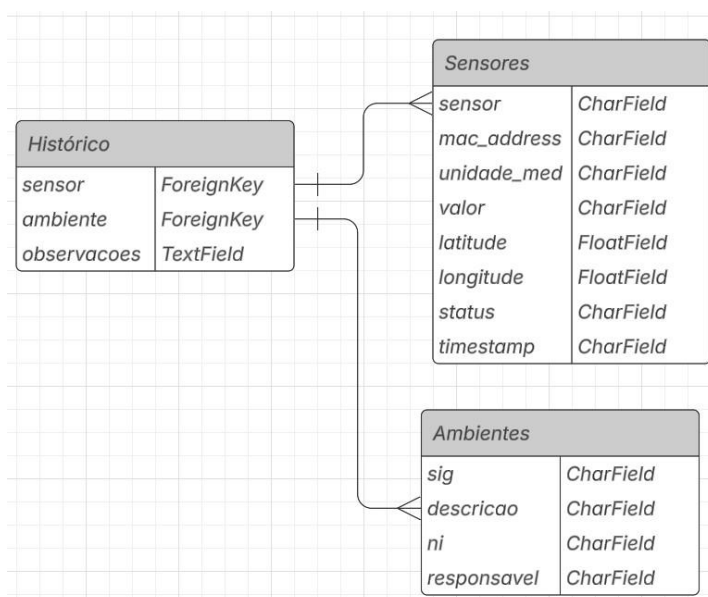
### CONTEXTO e DESCRIÇÃO

No entanto, como os sensores ainda estão em fase de aquisição e instalação, os dados fornecidos para testes serão simulados. O objetivo do projeto é desenvolver um back-end utilizando Django Rest Framework para gerenciar esses dados e disponibilizá-los para um front-end desenvolvido em React, que será usado para monitorar as condições em tempo real. A autenticação será realizada através de JSON Web Tokens (JWT).

A partir de agora criaremos um projeto chamado “smart\_city”. Este projeto vai comportar uma aplicação com a finalidade principal de coletar e expor os dados dos sensores da cidade inteligente. Para isso construiremos algumas APIs juntamente com o Banco de Dados da aplicação.

### Requisitos do Projeto:

1. **Back-End (Django Rest Framework):**
  - Criação de uma API RESTful para gerenciar dados de sensores.
  - A API deve ter endpoints para criar, ler, atualizar e deletar (CRUD) dados dos sensores e ambientes.
  - Os dados dos sensores devem incluir:
    - ✦ Temperatura (°C)
    - ✦ Luminosidade (lux)
    - ✦ Umidade (%)
    - ✦ Contador(num)
  - Os dados devem ser armazenados em um banco de dados db.sqlite.
  - Implementar autenticação utilizando JSON Web Tokens (JWT) para proteger os endpoints.



### 2. Login

Criar um super usuário para o nosso api\_smart.

- username = seu primeiro nome (exatamente) sem acentuação.
- password = seu número de matrícula no senai (está no portal)

**3. Relacionamento entre tabelas**

- Os relacionamentos deverão ser aplicados nas tabelas conforme diagrama já mencionado acima.

**4. Gerenciamento dos Sensores:**

- Crie as opções de CRUD para cada registro.
- Desenvolva opções de localização de dados, principalmente por sensor, data e status.
- Atualizar o status do sensor (ativo, inativo).

**5. Dados:**

- Criar método para capturar dados de sensores e ambientes que estão nas planilhas disponibilizadas.
- Os dados poderão ser exportados no formato de planilhas.

**Metodologia Scrum:**

A equipe utilizará a metodologia Scrum para organizar e gerenciar o desenvolvimento do projeto. O Scrum é uma estrutura ágil que promove o desenvolvimento iterativo e incremental, permitindo a adaptação rápida às mudanças e foco na entrega de valor.

**Papéis no Scrum:**

- **Product Owner:** Responsável por definir os requisitos e prioridades do produto. Para este projeto, o papel será desempenhado pelo instrutor.
- **Scrum Master:** Responsável por garantir que a equipe siga as práticas do Scrum. Pode ser um aluno designado ou o próprio instrutor.
- **Equipe de Desenvolvimento:** Composta pelos alunos, que são responsáveis pela implementação dos requisitos.

**Artefatos do Scrum:**

- **Product Backlog:** Lista priorizada de todas as funcionalidades desejadas no produto. Inclui histórias de usuário detalhando os requisitos.
- **Sprint Backlog:** Conjunto de histórias de usuário selecionadas do Product Backlog para serem trabalhadas durante a Sprint.
- **Incremento:** Soma de todos os itens do Product Backlog completados durante uma Sprint e todas as Sprints anteriores.

**Eventos do Scrum:**

- **Sprint Planning:** Reunião no início de cada Sprint para definir quais histórias de usuário do Product Backlog serão trabalhadas.
- **Daily Scrum:** Reuniões diárias para sincronizar as atividades e resolver impedimentos.
- **Sprint Review:** Reunião no final da Sprint para revisar o trabalho realizado e adaptá-lo conforme necessário.
- **Sprint Retrospective:** Reunião para refletir sobre a Sprint e identificar melhorias para o próximo ciclo.

**Tarefas a Serem Realizadas:****1. Desenvolvimento do Back-End:** ○**Histórias de Usuário:**

1. Como administrador, eu quero criar um endpoint para registrar dados de sensores, para que eu possa armazenar os dados de temperatura, luminosidade e umidade.
2. Como administrador, eu quero criar um endpoint para visualizar os dados dos sensores, para que eu possa monitorar as condições ambientais.
3. Como administrador, eu quero implementar autenticação JWT, para garantir que apenas usuários autorizados acessem os dados.

○ **Tarefas:**

- ✦ Configurar projeto Django e instalar o Django Rest Framework e djangorestframework-jwt.
- ✦ Criar modelos para dados de sensores.
- ✦ Implementar serializers e views.
- ✦ Configurar URLs e autenticação JWT.

**Critérios de Avaliação – Back End (Django)**

Nº	Critério	Descrição	Peso (%)
1	<b>Autenticação e Permissões</b>	Signin com JWT	7
2		signup	7
4	<b>Modelagem de Dados (Django)</b> Criação correta dos modelos	Modelagem de todas as tabelas: Históricos, Sensores e Ambientes.	10
5		Relações apropriadas (ForeignKeys) e validações.	5
6	<b>API Rest (Django Rest Framework)</b>	Implementação dos endpoints <b>CRUD</b> para todos os modelos.	15
7		Incluir filtro para localizar por ID de sensor.	4
8		Incluir filtro para localizar por tipo de sensor, exemplo “temperatura”.	4
9		Incluir filtro para localizar por data de sensor.	4
10		Incluir filtro para localizar por código “sig” do ambiente.	4
11		Incluir filtro para localizar por ID do Histórico.	4
12	<b>Documentação</b>	Criação da documentação da API.	5
13	<b>Funcionalidades</b>	Implementação um endpoint para <b>exportação</b> de relatórios em Excel (XLSX ou CSV).	8
14	<b>Popular banco de dados</b> (Observação: o ideal é quebrar o campo <b>timestamp</b> para facilitar as pesquisas)	Desenvolvimento de código para popular o banco a partir da planilha temperatura.	2
15		Desenvolvimento de código para popular o banco a partir da planilha umidade.	2
16		Desenvolvimento de código para popular o banco a partir da planilha luminosidade.	2
17		Desenvolvimento de código para popular o banco a partir da planilha contador.	2
18	<b>Organização do Código e Boas Práticas</b>	Estrutura do código, modularidade e organização do código Django. Código limpo.	10
19	<b>Apresentação</b>	Apresentar trabalho para à classe em Canvas.	5

Observações:

- O desenvolvimento será **individual**.
- O aluno deverá criar repositório no Github **privado** e dar acesso ao professor ([dorival.jrr@gmail.com](mailto:dorival.jrr@gmail.com)), enviando um e-mail com endereço do repositório.