





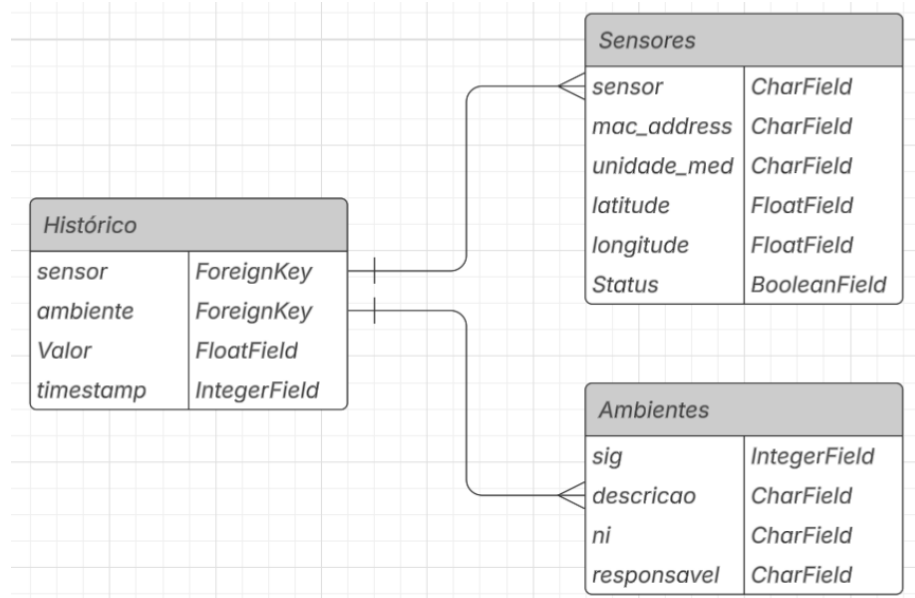


ATIVIDADE	PROJETO INTEGRADOR - SITUAÇÃO PROBLEMA
CONTEXTO e DESCRIÇÃO	
<p>A escola TecnoVille está desenvolvendo um projeto de transformação urbana baseado no conceito de Smart City. A ideia é implementar sensores em pontos estratégicos da cidade para coletar dados em tempo real sobre:</p> <ul style="list-style-type: none">•  Temperatura•  Umidade•  Luminosidade•  Contador de pessoas <p>Esses sensores serão instalados em locais como praças, corredores, pátios etc.</p> <p>A gestão escolar contratou sua equipe técnica para desenvolver somente o back end do sistema de monitoramento usando Django e Django Rest Framework. Os dados dos sensores serão enviados para a API e posteriormente visualizados por um painel web (a ser feito por outra equipe).</p> <p> Desafio proposto Você deve modelar e desenvolver o back end completo da aplicação, com base nos seguintes requisitos:</p> <p> Requisitos Funcionais</p> <ul style="list-style-type: none">• O sistema deve permitir registrar sensores, com campos como:<ul style="list-style-type: none">○ ID do sensor○ Sensor (temperatura, umidade, luminosidade, contador)○ Identificação (mac-address)○ Localização (latitude e longitude)○ Status operacional (ativo/inativo)• Deve ser possível registrar medições vindas dos sensores:<ul style="list-style-type: none">○ ID do sensor relacionado○ Ambiente (relacionado com a tabela de ambientes)○ Valor da medição○ Data e hora da leitura (timestamp)• Deve se registrar os ambientes da escola:<ul style="list-style-type: none">○ Sig (código do ambiente)○ Descrição○ Ni (número de identificação do responsável pelo ambiente)○ Responsável• A API deve fornecer endpoints para:<ul style="list-style-type: none">○ Cadastrar, editar e listar sensores○ Cadastrar e listar medições por sensor○ Listar as medições mais recentes (últimas 24h, por exemplo) <p>Requisitos do Projeto:</p> <ul style="list-style-type: none">• Back-End (Django Rest Framework):<ul style="list-style-type: none">○ 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:<ul style="list-style-type: none">▪ Temperatura (°C)▪ Luminosidade (lux)▪ Umidade (%)▪ Contador(num)○ Os dados devem ser armazenados em um banco de dados dbsqlite.○ Implementar autenticação utilizando JSON Web Tokens (JWT) para proteger os endpoints.	



- **Login**
 - Criar um super usuário para o nosso api_smart.
 - username = lin
 - password = 123
- **Relacionamento entre tabelas**
 - Os relacionamentos deverão ser aplicados nas tabelas conforme diagrama já mencionado acima.
- **Gerenciamento dos Sensores:**
 - Nas páginas Sensores e Ambientes os elementos deverão ser listados com 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).
- **Dados:**
 - Criar método para capturar dados de sensores e ambientes que estão nas planilhas disponibilizadas.
 - Os dados poderão ser exportados no formato de planilhas.
- **Integração entre Front End e Back End:**
 - Utilizar **Axios** no React para consumir a API Django.
 - Criar uma interface intuitiva para cadastro e acompanhamento das OS.
 - Inicie com uma página de login com a opção de cadastro de usuário.
 - Ao logar direcione para a página home em que teremos todas as opções, ou seja, como para cada tabela será criada uma página então deve-se colocar todos os links para todas as páginas.
 - A página de Ordem de Serviço deverá possuir apenas o Create, já as outras deverá possuir o CRUD completo.

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:

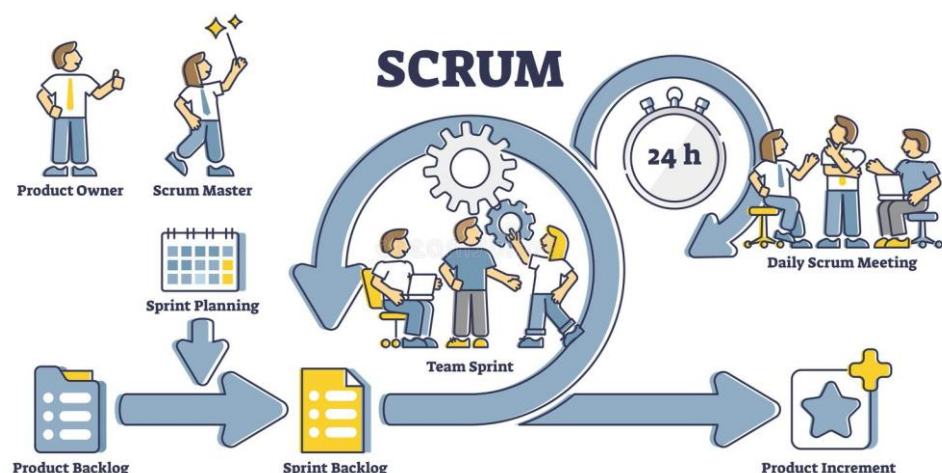
- **Product Owner:** Responsável por definir os requisitos e prioridades do produto. Para este projeto, o papel será desempenhado pelo instrutor.
- **Scrum Master:** Responsável por garantir que a equipe siga as práticas do Scrum. Pode ser um aluno designado ou o próprio instrutor.
- **Equipe de Desenvolvimento:** 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 de 15 minutos 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:

- **Desenvolvimento do Back-End:**

- **Histórias de Usuário:**

- Como administrador, eu quero criar um endpoint para registrar dados de sensores, para que eu possa armazenar os dados de temperatura, luminosidade e umidade.
 - Como administrador, eu quero criar um endpoint para visualizar os dados dos sensores, para que eu possa monitorar as condições ambientais.
 - 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.

- **Testes e Simulação:**

- Implementar scripts para gerar dados simulados de sensores.
 - Testar a API com ferramentas como Postman ou Insomnia.
 - Garantir que a aplicação móvel exiba corretamente os dados simulados.

Cronograma do Projeto:

- **Sprint 1 (1 semana):**

- Planejamento da Sprint.
 - Configuração do ambiente de desenvolvimento (Django e React).
 - Implementação inicial do back-end (endpoints básicos e autenticação JWT).
 - Implementação inicial do front-end (telas de login e dashboard).
 - Daily Scrums e reunião de revisão e retrospectiva ao final da Sprint.

- **Sprint 2 (1 semana):**

- Planejamento da Sprint.
 - Finalização dos endpoints e integração com banco de dados.
 - Desenvolvimento das funcionalidades de visualização de dados no front-end.
 - Testes e validação dos dados simulados.
 - Daily Scrums e reunião de revisão e retrospectiva ao final da Sprint.

Critérios de Avaliação – Back End (Django)

Nº	Critérios	Descrição	Peso (%)
1.	Autenticação e Permissões	Signin com JWT	5
2.		signup	5
3.	Modelagem de Dados (Django)	Modelagem de todas as tabelas: Históricos, Sensores e Ambientes.	10
4.		Relações apropriadas (ForeignKeys) e validações.	5
5.	API Rest	Implementação dos endpoints CRUD para todas as páginas que possua dados.	15
6.	Filtros	Localizar por ID de sensor.	2
7.		Localizar por tipo de sensor, exemplo “temperatura”.	2
8.		Localizar por data de sensor.	2
9.		Localizar por código “sig” do ambiente.	2
10.		Localizar por ID do Histórico.	2
11.		Filtro duplo por data e por sensor	3
12.		Filtro triplo por data e por sensor e hora	4
13.	Funcionalidades	Implementação de exportação de relatórios em Excel (XLSX ou CSV).	10
14.	Popular banco de dados	Popular o banco a partir da planilha temperatura.	2
15.		Popular o banco a partir da planilha umidade.	2
16.		Popular o banco a partir da planilha luminosidade.	2
17.		Popular o banco a partir da planilha contador.	2
18.	Organização do Código e Boas Práticas	Estrutura do código, modularidade e organização do código Django. Código limpo.	5
19.	Metodologias Ágeis	De acordo com o prazo proposto, divida-o em sprints e faça um documento explicando o que seria entregue em cada uma.	5
20.		Crie um documento e acrescente dados artificiais de: Product Owner, Scrum Master, Sprints(datas), Sprint Planning e Sprint Retrospective.	5

Observações:

- O desenvolvimento será **individual**.
- Se o aluno precisar do professor caso não conseguir prosseguir por ser uma sequência, será auxiliado, mas perderá o requisito.
- **Entrega:** O aluno deverá criar repositório no Github **privado** e dar acesso ao professor (lindomarbatistao@gmail.com), **enviando um e-mail com endereço do repositório**.
- A correção será feita apenas no Insomnia, não será utilizado front-end.