**Ficha Técnica do Código**

**1. Descrição do Projeto**

**Objetivo**: Desenvolver uma aplicação web para gerenciar dados de alunos, com funcionalidades para adição, atualização, exclusão, exportação de dados para CSV, e geração de relatórios em PDF. A aplicação utiliza o framework Flask para a interface web, SQLite para armazenamento temporário dos dados, e ThingBoard para integração IoT.

**2. Tecnologias Utilizadas**

* **Flask**: Framework web em Python utilizado para criar e gerenciar a aplicação web.
* **SQLite**: Banco de dados relacional leve usado para armazenamento temporário dos dados dos alunos.
* **Pandas**: Biblioteca de manipulação de dados em Python para exportar dados em formato CSV.
* **ReportLab**: Biblioteca Python para gerar documentos PDF a partir dos dados dos alunos.
* **ThingBoard**: Plataforma IoT para gerenciar e visualizar dados, integrada à aplicação para enviar e receber dados.
* **HTML/CSS**: Linguagens de marcação e estilo usadas para criar e estilizar a interface web da aplicação.

**3. Funcionalidades da Aplicação**

* **Adição de Aluno**:
  + Permite o cadastro de novos alunos com informações como nome, idade, horário das aulas e datas das aulas.
  + As informações são armazenadas no banco de dados SQLite e enviadas para ThingBoard.
* **Atualização de Aluno**:
  + Permite a atualização dos dados dos alunos existentes na base de dados.
* **Exclusão de Aluno**:
  + Permite a remoção de registros de alunos do banco de dados com uma confirmação para evitar exclusões acidentais.
* **Exportação para CSV**:
  + Exporta todos os dados dos alunos para um arquivo CSV para análise posterior. Utiliza a biblioteca Pandas para criar e salvar o arquivo CSV.
* **Geração de Relatório PDF**:
  + Gera um relatório PDF contendo informações dos alunos e detalhes do projeto. Utiliza a biblioteca ReportLab para criar o documento PDF formatado.
* **Integração com ThingBoard**:
  + Envia dados dos alunos para ThingBoard para monitoramento e visualização em tempo real. Utiliza a API do ThingBoard para enviar dados via requisições HTTP.

**4. Estrutura do Código**

* **Back-end (Flask)**:
  + **Rotas**:
    - /: Página principal que exibe a lista de alunos.
    - /add: Rota para adicionar novos alunos.
    - /delete/<int:id>: Rota para excluir um aluno pelo ID.
    - /export\_csv: Rota para exportar os dados dos alunos para um arquivo CSV.
    - /generate\_pdf: Rota para gerar um relatório PDF com os dados dos alunos.
* **Banco de Dados (SQLite)**:
  + **Tabela alunos**:
    - **Colunas**:
      * id: Identificador único do aluno (autoincremento).
      * nome: Nome do aluno.
      * idade: Idade do aluno.
      * horario: Horário das aulas (ex.: "8:00 a 9:00" ou "19:00 a 20:00").
      * data\_aulas: Dia(s) da semana das aulas (ex.: "Segunda", "Quarta", "Sexta").
      * professor: Nome do professor associado ao horário (automaticamente atribuído com base no horário).
* **Front-end (HTML/CSS)**:
  + **Página HTML**:
    - **Formulários**:
      * Formulário para adicionar novos alunos com campos para nome, idade, horário e data das aulas.
      * Botões para exportar os dados para CSV e gerar o relatório PDF.
    - **Tabela**:
      * Exibe a lista de alunos com informações e opções para excluir registros.
* **Integração com ThingBoard**:
  + **Endpoint**:
    - http://<thingboard-ip>:<port>/api/v1/<access-token>/telemetry: URL para enviar dados para ThingBoard. Substitua <thingboard-ip>, <port>, e <access-token> pelos valores específicos do seu servidor ThingBoard.

**5. Configurações e Dependências**

* **Instalação de Pacotes**:
  + Flask: pip install flask
  + SQLite: Incluído no Python padrão.
  + Pandas: pip install pandas
  + ReportLab: pip install reportlab
  + Requests: pip install requests
* **Configuração do ThingBoard**:
  + **URL e Token de Acesso**: Configure o URL e o token de acesso para a API do ThingBoard de acordo com seu setup.
* **Banco de Dados SQLite**:
  + O banco de dados é criado automaticamente na primeira execução da aplicação. Para persistência dos dados, remova a reinicialização do banco de dados no código.

**6. Considerações Finais**

* **Segurança**: Proteja o endpoint do ThingBoard e valide as entradas do usuário para evitar problemas de segurança.
* **Interface**: A interface HTML pode ser melhorada utilizando frameworks de CSS como Bootstrap para um design mais profissional.