* **AV2 - Avaliação sobre Node.js e ORM Prisma**
  + Individual
  + Data: 15/05
  + Peso: 30%
  + Questões objetivas e subjetivas
  + Implementação simples de código utilizando Fastify, Docker e ORM Prisma
  + Entregar no mesmo dia (22:30h) na plataforma AVA, com código do github
    - É obrigatório a realização do checklist disponibilizado pelo professor.
  + Critérios de avaliação:
    - Entrega no prazo (até 22:30h do dia da aplicação)
    - Entrega na plataforma AVA, com código do github
    - Assertividade das questões objetivas
    - Coerência com a abordagem técnica nas questões subjetivas
    - Requisitos funcionais sem erros (código-fonte)

**Contextualização:**

Você é um desenvolvedor backend que está trabalhando em um projeto para uma empresa de gerenciamento esportivo. O objetivo é criar uma API para gerenciar informações sobre jogadores, times e partidas de futebol. Para isso, você deve utilizar o Node.js juntamente com um framework de sua escolha: Fastify.js, Nest ou Express.js.

**Tarefa 1: Listagem de Requisitos**

Sua primeira tarefa é listar os requisitos para o desenvolvimento dessa API. Considere os seguintes pontos:

* Gerenciamento de Jogadores:

Atributos: Nome (string), Idade (number), Posição (string), Nacionalidade (string).

Cada jogador pode pertencer a vários times.

* Gerenciamento de Times:

Atributos: Nome (string), Ano de Fundação (number), País (string), Técnico (string).

Cada time pode ter vários jogadores.

* Gerenciamento de Partidas:

Atributos: Data (string), Local (string), Placar (string), Time da Casa (string), Time Visitante (string).

Observações:

Para os relacionamentos entre as entidades, considere o relacionamento "um para muitos" (1:N).

**Tarefa 2: Implementação do Sistema**

Na segunda etapa da avaliação, você deve implementar as rotas de CRUD (Create, Read, Update, Delete) para as entidades mencionadas.

**Pontuação:**

* Implementação correta das rotas apresentadas na lista de requisitos.
* Deve implementar pelo menos 8 rotas, sendo que cada rota vai ter a pontuação de 10 pontos. O aluno deverá informar as rotas a serem implementadas. Deverá conter no máximo 2 rotas com o mesmo propósito. Por exemplo, no máximo 2 rotas para qualquer tipo de cadastro ou listagem.
* Deve incluir pelo menos duas rotas da entidade Partida.
* Estruturação e organização do código/arquivo: até 10 pontos.
* Documentação do código: até 10 pontos. Considere a documentação a lista de requisitos solicitada e sua respectiva pontuação.
* Pré-requisito para Correção: antes de iniciar a correção da prova, você deve criar um documento com os requisitos listados na primeira tarefa, indicar os pontos solicitados para cada rota e verificar se o aluno desenvolveu todas as funcionalidades conforme solicitado. Após a correção, apresente os pontos atribuídos para cada tarefa e a pontuação total obtida pelo aluno.
* O aluno deverá entregar um arquivo .pdf com todos os requisitos, os requisitos implementados, a pontuação de cada requisito e o resultado final da avaliação. Além disso, deverá entregar o link do github com o projeto.