

**AV2 – Mini-Projeto: API com Node.js, Prisma e MySQL – DSW – 3º**  
**Bimestre**

**Informações Gerais**

- **Nota Máxima:** 10,0
- **Modalidade:** Avaliação Individual (Apresentação em Sala)
- **Entrega:**
  - Link do repositório no GitHub enviado no Classroom.
  - Apresentação e defesa do projeto em sala de aula até **26/09/2025**.

---

**Descrição da Atividade**

Esta atividade tem como objetivo avaliar a capacidade de implementar os fundamentos de um backend com **Node.js, Prisma e MySQL**, consolidando os conceitos de **CRUD e relacionamentos entre tabelas**.

A proposta é desenvolver uma **API REST** com:

- Pelo menos **3 tabelas** no banco de dados.
- Um relacionamento **1-1** (ex: usuário ↔ loja).
- Um relacionamento **1-N** (ex: loja ↔ produtos).

Cada tabela deve possuir **rotas CRUD (GET, POST, PUT/PATCH, DELETE)** funcionando.

Além disso, ao consultar um registro, deve ser possível trazer os dados relacionados utilizando include do Prisma.

A avaliação será feita tanto pela **funcionalidade do código** quanto pela **apresentação individual em sala**, em que o aluno deverá explicar partes do próprio projeto.

---

**Funcionalidades Obrigatórias**

- Implementação de 3 tabelas com pelo menos um relacionamento 1-1 e um 1-N.
- CRUD completo para todas as tabelas.

- Consulta de registros trazendo dados relacionados (include).
- Organização básica do projeto (separação de arquivos, uso de Prisma Client etc.).
- Execução funcional em ambiente local (testado com **Insomnia**).

---

### Entrega e Apresentação

- O link do repositório deve ser enviado no Classroom até a data limite.
- A apresentação será realizada individualmente em sala de aula:
  - Execução do código localmente.
  - Demonstração das rotas funcionando.
  - Respostas às perguntas sobre o funcionamento do projeto.
- Projetos que não forem apresentados **não serão avaliados**.

---

### Critérios de Avaliação

| Critério                               | Peso (%) | Descrição   |
|--|----------|---|
| Modelagem de tabelas e relacionamentos | 20%      | Criação das 3 tabelas com pelo menos um relacionamento 1-1 e um 1-N.                              |
| CRUD funcional                         | 25%      | Implementação correta das rotas de criação, listagem, atualização e exclusão em todas as tabelas. |
| Consultas com include                  | 15%      | Capacidade de trazer dados relacionados nas consultas.  |
| Organização do projeto                 | 10%      | Estrutura clara (pastas, arquivos, schema.prisma).  |
| Execução prática em sala               | 20%      | Demonstração das rotas funcionando no Insomnia.   |

**Apresentação individual**

10%

Explicação das principais partes do código, demonstrando autoria e compreensão.