



# Design Patterns no Node.js

# O que são Design Patterns?

- Patterns são regras de **organização** do código;
- Constante evolução;
- Otimização prematura;
- Patterns comuns:
  - Singleton (a classe só pode ter uma instância);
  - Repository (abstração da conexão com o banco);
  - Service (abstração de lógica);
  - Observer (comunicação por eventos);

# Qual vamos utilizar?

- Singleton Pattern
  - O Node.js transforma toda exportação de objeto em um singleton;
  - Novos requires/imports importam a mesma instância;
- Service Pattern
  - Abstrair lógica do controller;
  - Quando? (code smell)
    - Código muito denso;
    - Condicionais dentro do controller;
    - Regra de negócio redundante;

# Por que não usar mais patterns?

- Características específicas do Node.js;
- Patterns são úteis em projetos conforme crescem;
- Quando uma aplicação Node.js cresce muito você tem outras alternativas:
  - Micro-serviços;
  - Serverless;
- Vai construir uma aplicação monolítica? Melhor utilizar o framework como o AdonisJS que já tem os patterns aplicados;

# BORA CODAR!