

Professores

Celso Rodrigo Giusti
Daniel Manoel Filho
Isadora Dias Afonso
Marlon Palata Fanger
Rodrigues

Persistência de Dados



Objetivos

Nesta aula, buscaremos alcançar as seguintes competências técnicas:

- **Compreender** o conceito de persistência de dados em aplicações Back-end.
- **Configurar** a conexão entre o ambiente Node.js e o servidor MySQL.
- **Executar** operações básicas de banco de dados (Leitura e Escrita) via código.

O que é Persistência de Dados?

Persistência de dados é a capacidade de **armazenar informações de forma permanente**, mesmo após o encerramento da aplicação.

Sem persistência:

```
let usuarios = [];
```

Se o servidor reiniciar → dados perdidos.

Com persistência:

Dados armazenados em banco.

Reiniciou o servidor → dados continuam lá.

Por que precisamos persistir dados?

O que acontece com um sistema de escola se os dados dos alunos forem apagados toda vez que o servidor reinicia?

Sistemas reais precisam guardar:

- Usuários
- Senhas
- Produtos
- Vendas
- Logs
- Configurações

Tipos de Banco de Dados

Banco Relacional (SQL)

Exemplos:

- MySQL
- PostgreSQL
- Microsoft SQL Server

Características:

- Tabelas
- Linhas e colunas
- Relacionamentos
- Linguagem SQL

Banco Não Relacional (NoSQL)

Exemplos:

- MongoDB
- Redis

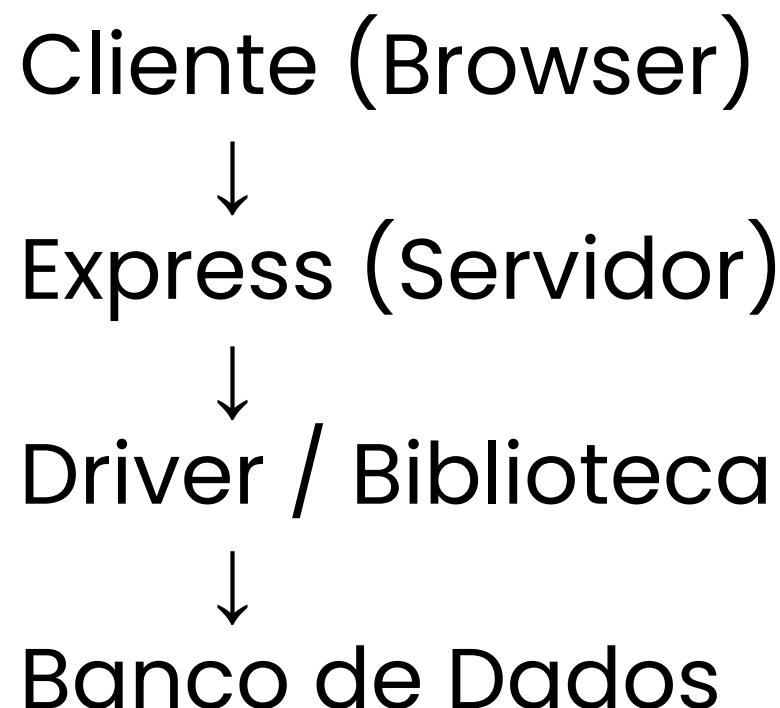
Características:

- Documentos
- JSON
- Flexibilidade de estrutura
- Escalabilidade horizontal

O Ecossistema Node.js e MySQL

Para que o Node.js "converse" com o MySQL, precisamos de um driver de conexão. O pacote mais comum para essa tarefa é o **mysql2**, que permite enviar comandos SQL diretamente do código JavaScript.

Fluxo geral



O que é um Driver?

É a biblioteca que permite que o Node converse com o banco.

Exemplos:

- mysql2
- pg
- mongoose

Estruturação Profissional do Projeto

Uma aplicação organizada facilita manutenção, testes e escalabilidade. Por isso, separaremos a aplicação em dois arquivos principais:

```
cinema-api/
  └── src/
    ├── config/
    │   └── database.js
    ├── app.js
    └── server.js

  └── .env
  └── package.json
```

Variáveis de Ambiente

Nunca é recomendado inserir credenciais diretamente no código-fonte. Isso representa um risco de segurança.

Utilizamos um arquivo `.env` para armazenar informações sensíveis como:

- Host
- Usuário
- Senha
- Nome do banco
- Porta

```
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=1234
DB_NAME=cinema
DB_PORT=3306
PORT=3000
```

Vamos praticar?





Escola SENAI “Italo Bologna”

Av. Goiás, 139 – Itu/SP

Telefone

(11) 2396-1999

Instagram

@senai.itu

Facebook

/senai.itu

Site

<https://sp.senai.br/unidade/itu/>