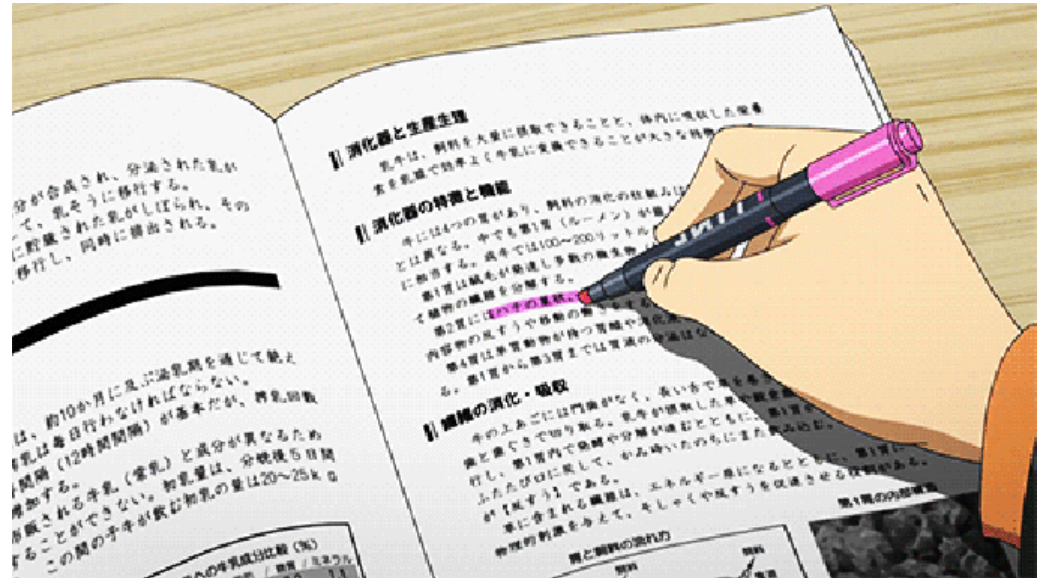




Auto estudo...

Até 10h





DAILY

A FAZER



DESENVOLVIMENTO - 5



TESTE - 3



CONCLUÍDO



Até 10h15

INTRODUÇÃO À COMPUTAÇÃO

Back - end II —

Endpoints de leitura e escrita com documentação própria





Agenda do dia



Explicação Teórica



Mão na massa





QUEM SOU EU...

Dr. CRISTIANO
BENITES



*Deficiente auditivo *Grau profundo*

***CID H91.8**

Especialista em leitura labial



Combinados







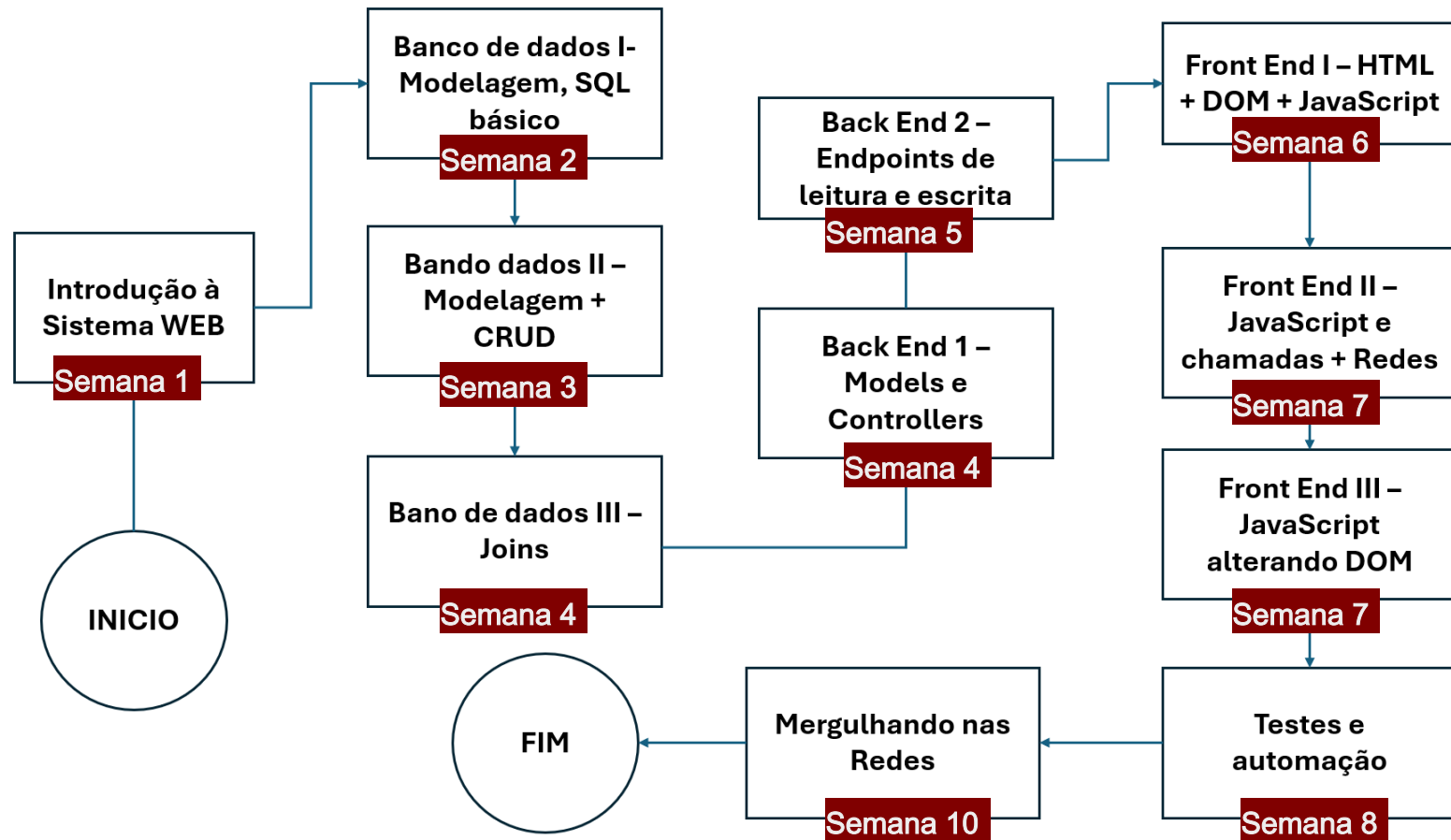
-  Você é responsável pela seu Check in na Adalove
-  Presença na aula das 10h até 12h. “Ausência acarretará em falta”
-  Atividades entregues no prazo
-  Estou sempre disponível no SLACK ou em sala de aula
-  Se planejem sempre de forma antecipada
-  Avaliem nosso encontro na Adalove. Vamos praticar feedback o tempo todo






Diagrama Módulo






Conteúdo do dia



 Back-end II - Endpoints de leitura e escrita com documentação própria

Semana 05

 Entrada e saída de dados com JSON

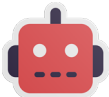
Semana 05

 HTML básico

Semana 05

 Criando Endpoints, Controllers e Models na nossa arquitetura

Semana 05

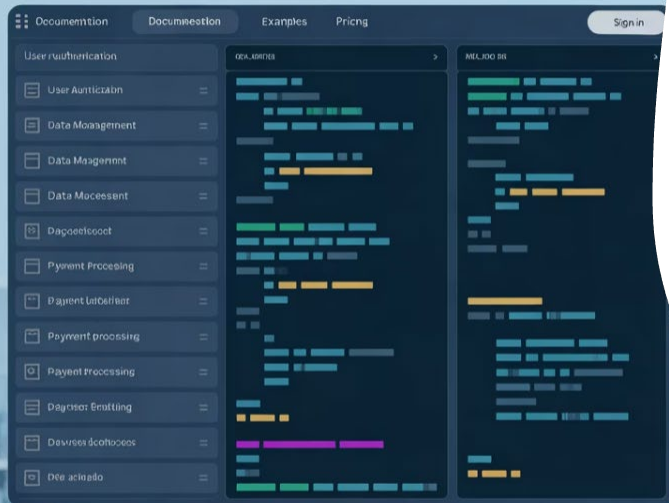


Back-end II:

Endpoints de Leitura e Escrita

Foco em APIs RESTful com documentação própria

Manipulação de dados usando JSON



Visão geral do dia



Endpoints CRUD

GET, POST, PUT e DELETE para operações completas



HTML Básico

Integração do front-end com nossa API



Arquitetura MVC

Controllers e Models bem estruturados



Fundamentos de JSON

O que é

Formato leve de intercâmbio de dados

JavaScript Object Notation

Estrutura

Pares de chave-valor

Compatível com diversas linguagens

Exemplo

```
{"nome": "Maria", "idade": 25, "ativo": true}
```

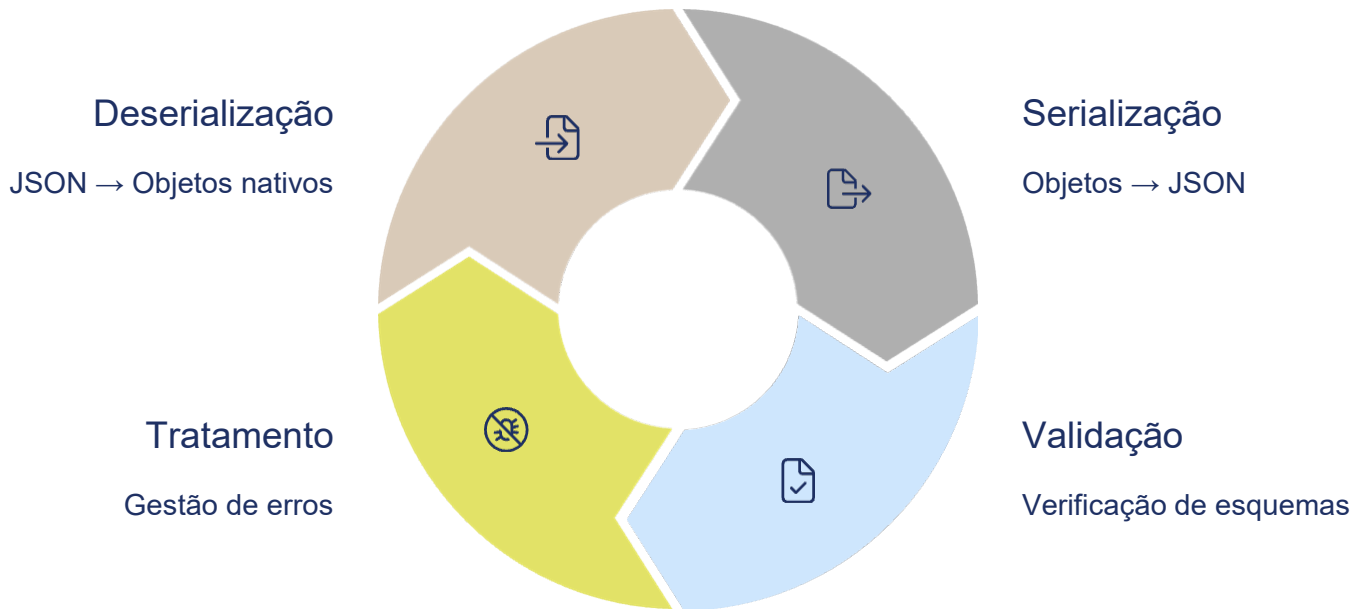
Fácil integração com várias linguagens

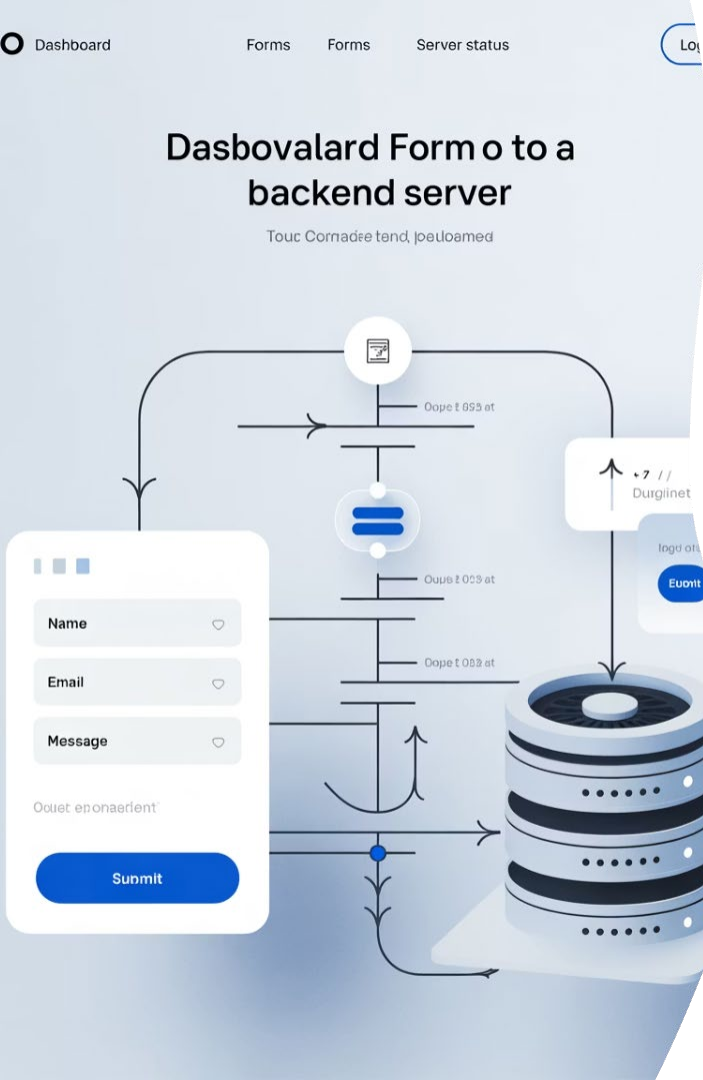
Ideal para APIs modernas

Suporte nativo em linguagens de programação



Manipulando JSON no Back-end





HTML Básico para Desenvolvedores Back-end

Estrutura Básica

Documento HTML com tags essenciais

DOCTYPE, html, head, body

Formulários

Tags form, input e button

Métodos GET e POST

Atributos

action, method, name, value

Direcionamento de dados

Renderização

Exibição de dados do back-end

Templates dinâmicos

Criando Controllers Eficientes



Estrutura Padrão

Classe com métodos específicos



Métodos CRUD

Um para cada operação



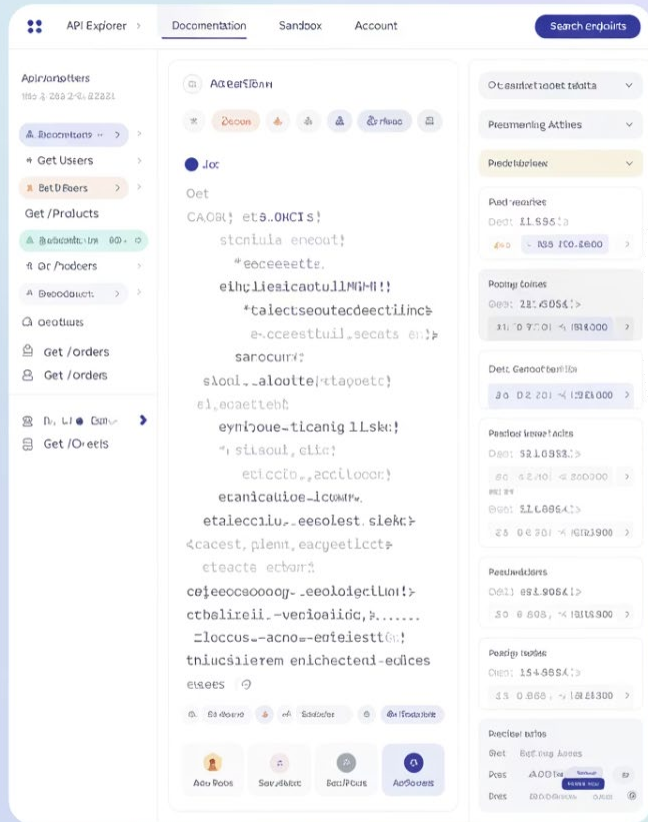
Validação

Verificar dados de entrada



Resposta

Status e dados apropriados



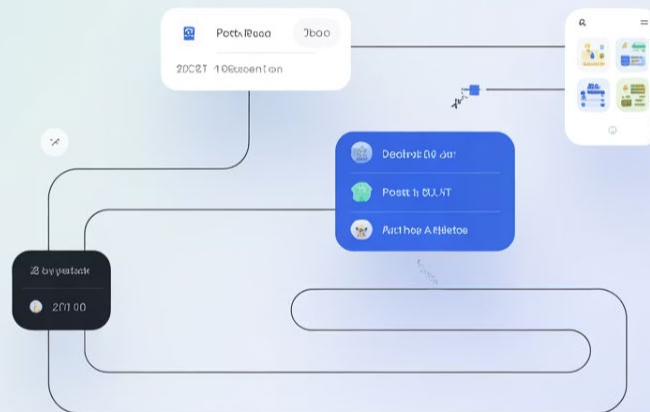
Endpoints de Leitura (GET)

Endpoint	Função	Status
GET /produtos	Listar todos	200 OK
GET /produtos/:id	Buscar um	200 OK
GET /produtos?query	Filtrar	200 OK
GET /produtos (não encontrado)	Vazio	404 Not Found

Restic API Write R→ Operatine Operations

 Pesocoes

⇒ Ololopah



Post

Jsornen

Put

Delete

450

NNK

XMI

XMI

Endpoints de Escrita (POST, PUT, DELETE)

Método	Operação	Status Sucesso	Resposta
POST	Criar novo	201 Created	Recurso criado
PUT	Atualizar completo	200 OK	Recurso atualizado
PATCH	Atualizar parcial	200 OK	Campos modificados
DELETE	Remover	204 No Content	Sem corpo

1. Entrada e saída de dados com JSON

✓ Onde está aplicado:

- Alunos:
 - Rota `/alunos/curso/:curso_id` retorna alunos filtrados em formato JSON (arquivo `alunoController.js`, método `byCurso`)
- Isso permite ensinar como uma API RESTful responde com dados em formato estruturado para front-ends ou integrações.

📁 Referência:

- `controllers/alunoController.js`
- `routes/alunos.js`



2. HTML básico

✓ Onde está aplicado:

- Views com formulários HTML em EJS para **entrada de dados**
 - `views/alunos/index.ejs` → Cadastro e listagem de alunos + cursos
 - `views/professores/index.ejs` → Cadastro e listagem de professores
- Uso de `<form>`, `<table>`, `<input>`, `<select>`, e interação com rotas POST/GET.

Referência:

- `views/alunos/index.ejs`
- `views/professores/index.ejs`



3. Criando Endpoints, Controllers e Models na nossa arquitetura

✓ Onde está aplicado:

Endpoints (Rotas)

- Rotas configuradas para cada entidade:
 - `routes/alunos.js`
 - `routes/cursos.js`
 - `routes/professores.js`
- Todas seguem padrão REST: `GET`, `POST`, `POST edit`, `POST delete`

Controllers

- Arquivos que centralizam a lógica de negócio:
 - `controllers/alunoController.js`
 - `controllers/cursoController.js`
 - `controllers/professorController.js`

Models

- Camada responsável pelo banco de dados:
 - `models/aluno.js`
 - `models/curso.js`
 - `models/professor.js`



✓ RESUMO VISUAL

Tema da Aula

Entrada e saída com JSON

HTML básico

Endpoints com Controllers e Models

Onde Está no Projeto

`alunoController.js` → método `byCurso`

`index.ejs` (alunos e professores)

`routes/`, `controllers/`, `models/`



BORA FAZER UMA
ATIVIDADE PARA
FIXAR OS CONCEITOS!

MÃO NA
MASSA



https://github.com/cristianobenites/Materiais_Aulas/blob/main/Back_End/Modulo2/aula_6/Parte_1_alunos.md



Avalie nossa aula de hoje na Adalove!



Aluno



Professor

MUITO OBRIGADO!



Formações Acadêmicas

Graduações

Enfermagem

Ciência da Computação

Tecnologia em Redes de Computadores

Programação de Computadores

Sistema de Informação

Pós- Graduações

Mestre em Engenharia Elétrica e da Computação

Doutor em Engenharia Elétrica e da Computação

MBA Em Data Center e Computação em Nuvem

MBA em Gestão da Tecnologia da Informação

MBA Engenharia de Software

Certificações

DELL EMC PROFISSIONAL – Cloud Infrastructure and Services;

DELL PROVEN PROFISSIONAL – Information Storage and Management.

ISO 27001

MCSA Windows SERVER - 410, 411 e 412;

Linux – Lpic 1 e Lpic 2;

Microsofit Certified Profiissional;

Symantec Backup Exec;

Symantec Endpoint Protection,

Cobit;

ITIL;



Professor

Dr. Cristiano

Benites

OBRIIGADO
E LET'S BORA!