

# A LINGUAGEM UNIVERSAL DOS DADOS

Dominando Objetos e JSON

MSC. SANDIR RODRIGUES CAMPOS



ESTRUTURA  
DE DADOS

# **Objetivos de Aprendizagem**

# Agenda da Aula

1

## A Ideia por Trás dos Objetos

Foco: Conceitual

30 min

2

## JSON - A Sintaxe Universal

Foco: Prático (Sintaxe)

40 min

3

## JSON em Ação: APIs e Serialização

Foco: Aplicação Real

40 min

4

## Desafio Final: Modelagem Avançada

Foco: Consolidação

10 min

# Do Mundo Real à Programação

Tudo ao nosso redor pode ser descrito por suas características.

Em programação, não lidamos com o objeto físico, mas com uma lista organizada de seus atributos.

## OBJETO: CARRO

Marca	"Ford"
Modelo	"Mustang"
Ano	1969
Cor	"Vermelho"
Motor	"V8"

# A Estrutura Chave-Valor

Para o computador, um objeto é apenas uma lista organizada de propriedades.

## **CHAVE (KEY)**

O nome da propriedade (ex: "Cor"). É o rótulo.

## **VALOR (VALUE)**

O dado real (ex: "Vermelho"). É o conteúdo.

**Carro:**

**marca:** "Ford"

**modelo:** "Mustang"

**ano:** 1969

**cor:** "Vermelho"

## EXERCÍCIO 01

# Descrivendo o Mundo

Em duplas, escolham **2 itens** da lista ao lado e descrevam suas propriedades usando a estrutura chave-valor.

FORMATO ESPERADO (TEXTO SIMPLES)

Objeto:

Chave: Valor

Chave: Valor

Chave: Valor

## OPÇÕES DE MODELAGEM

### 1. Smartphone

marca, modelo, cor,  
armazenamento...

### 2. Produto de Mercado

nome, preço, validade, categoria...

### 3. Aluno

nome, matrícula, curso, email...

# JSON

## JavaScript Object Notation

### **LEVE**

Estrutura minimalista com poucos caracteres, ideal para transferência rápida de dados pela rede.

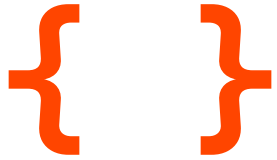
### **LEGÍVEL**

Formato de texto simples que é fácil para humanos lerem, escreverem e entenderem.

### **UNIVERSAL**

Independente de linguagem. É nativamente suportado por Python, Java, C#, PHP e muitas outras.

# As Regras do Jogo



## ESTRUTURA

Todo objeto JSON começa e termina com **chaves**.



## CHAVES (KEYS)

Nomes de propriedades entre **aspas duplas**.



## ATRIBUIÇÃO

Use **dois-pontos** para ligar chave e valor.



## SEPARAÇÃO

Use **vírgula** entre pares. **Atenção:** o último não leva vírgula.

# Tipos de Dados em JSON

## PRIMITIVOS

### String

Texto entre aspas duplas.

```
"Olá Mundo"
```

### Number

Inteiros ou decimais (ponto).

```
42, 3.14, -10
```

### Boolean

Verdadeiro ou Falso.

```
true, false
```

### Null

Ausência intencional de valor.

```
null
```

## ESTRUTURAIS

### Object

Coleção de pares chave-valor.

```
{ "id": 1, "nome":  
  "Ana" }
```

### Array

Lista ordenada de valores.

```
[ "HTML", "CSS",  
  "JS" ]
```

# Exercício 2: Conserte o JSON

Identifique o erro de sintaxe que impede a validação em cada caso.

## CASO A

```
{ nome: "João", idade:  
25 }
```

## CASO B

```
{ 'cidade': 'São  
Paulo', 'uf': 'SP' }
```

## CASO C

```
{ "tags": [ "tech",  
"mobile", ] }
```

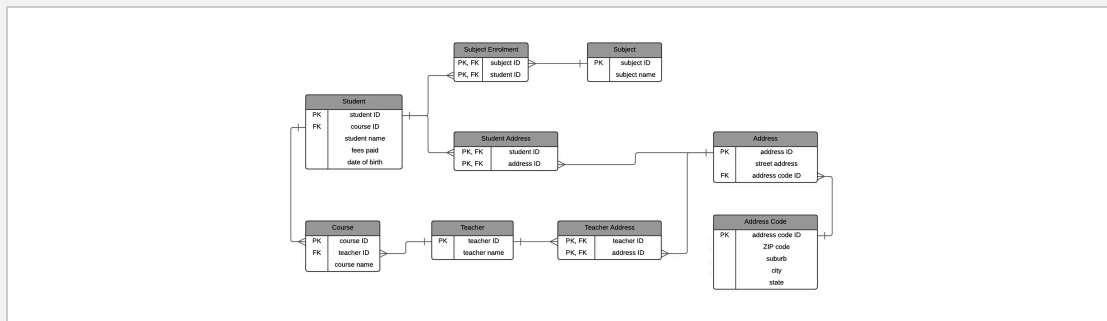
# Aninhamento e Arrays

## ANINHAMENTO (OBJECTS)

Um valor pode ser outro objeto completo. Ideal para agrupar dados relacionados (ex: Endereço).

## ARRAYS (LISTAS)

Um valor pode ser uma lista de itens entre colchetes [ ]. Ideal para múltiplos registros.



```
{  
  "nome": "Ana Silva",  
  "vip": true,  
  
  "endereco": {  
    "rua": "Av. Paulista",  
    "numero": 1000,  
    "cidade": "SP"  
  },  
  
  "telefones": [  
    "(11) 99999-1111",  
    "(11) 98888-2222"  
  ]  
}
```

## EXERCÍCIO 03

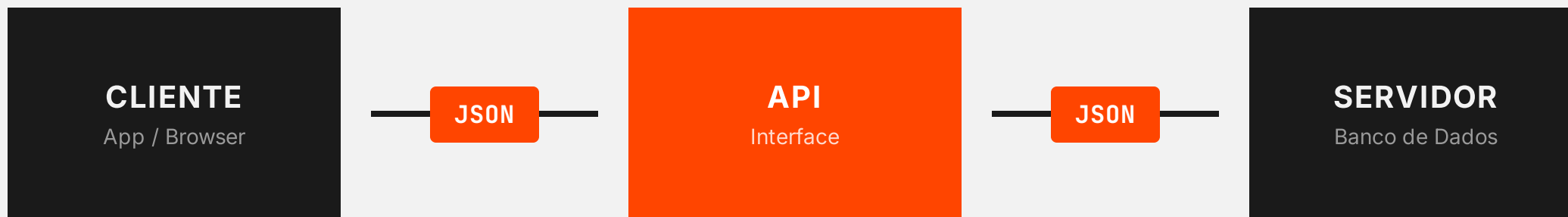
# Pedido Online

- ID do Pedido `number`
- Data da Compra `string`
- Valor Total `number`
- Dados do Cliente `object`
- Lista de Produtos `array of objects`

## ESTRUTURA SUGERIDA

```
pedido { }  
  id  
  data  
  total  
  
  cliente { }  
    nome  
    email  
  
  itens [ ]  
    produto 1 { }  
    produto 2 { }
```

# APIs: A Ponte Entre Sistemas



---

## O QUE É API?

Application Programming Interface. É um contrato que define como dois sistemas diferentes podem conversar e trocar informações de forma segura.

---

## O PAPEL DO JSON

É o idioma universal dessa conversa. A API usa JSON para empacotar os dados solicitados pelo cliente e enviá-los de volta.

# Requisição e Resposta

## O PEDIDO (REQUEST)

GET Data

```
GET /api/produtos/42
```



## A RESPOSTA (RESPONSE)

```
HTTP 200 OK
```

```
{ "id": 42, "nome": "Notebook Pro",  
  "preco": 4500.00, "estoque": true }
```

# Serialização e Desserialização

## MEMÓRIA (OBJETO)

```
user = {  
  id: 1,  
  name: "Ana"  
}
```

SERIALIZAÇÃO



## TEXTO (JSON)

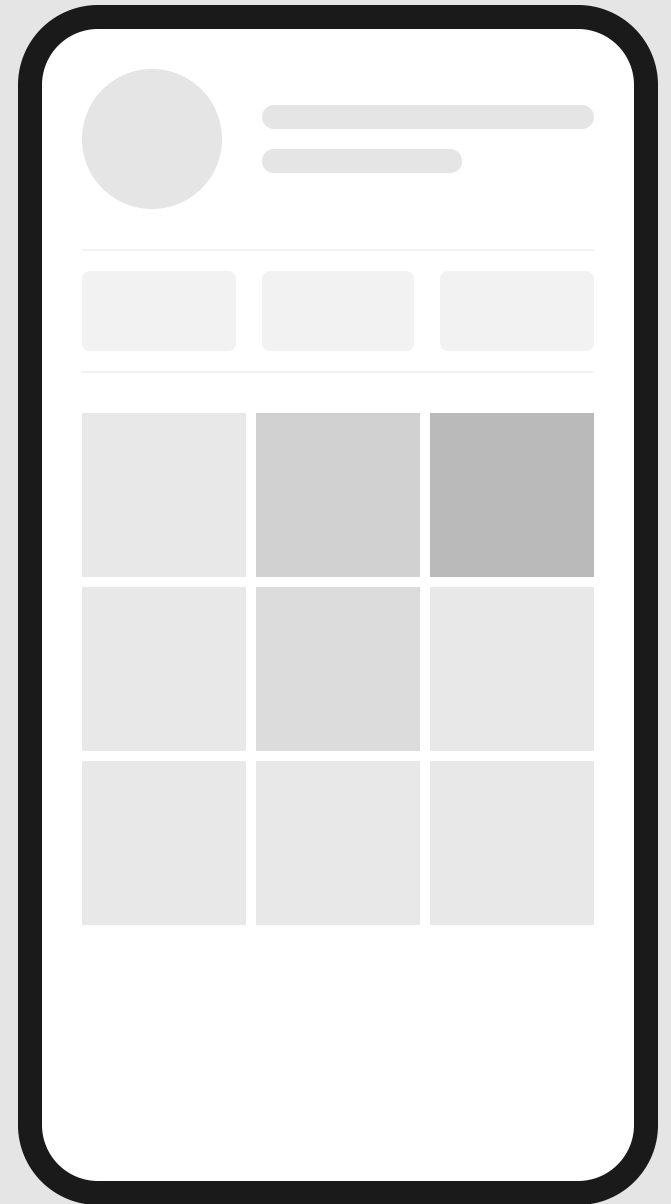
```
'{"id":1,"name":"Ana"}'
```

## EXERCÍCIO 04

# Engenharia Reversa

Imagine que você é o Backend Developer. Esboce a estrutura JSON necessária para alimentar a tela ao lado.

- ✓ `username (string)`
- ✓ `is_verified (boolean)`
- ✓ `stats (object: followers...)`
- ✓ `posts (array of objects)`



# DESAFIO FINAL

## Catálogo de Produtos

Você é o arquiteto de dados de um novo e-commerce. Sua missão é projetar a estrutura JSON que será usada para cadastrar milhares de produtos eletrônicos.

## REQUISITOS TÉCNICOS

---

### 01 ESTRUTURA COMPLEXA

O modelo deve conter objetos aninhados (ex: especificações técnicas).

### 02 ARRAYS DE OBJETOS

Deve incluir listas de objetos (ex: variações de preço por moeda).

### 03 TIPAGEM RIGOROSA

Números devem ser `number`, textos devem ser `string`.

# Estrutura Obrigatória

## CAMPOS BÁSICOS

---

<code>id</code>	Number
<code>nome</code>	String
<code>marca</code>	String
<code>disponivel</code>	Boolean

## CAMPOS COMPLEXOS

---

`caracteristicas`

Object

```
{
  processador: String,
  memoria_ram: String
}
```

`precos`

Array of Objects

```
[
  { moeda: String, valor: Number }
]
```

`tags`

Array of Strings

# O que Aprendemos Hoje

# Continuando a Jornada

## PRÓXIMOS NÍVEIS

---

### Consumo de APIs

Aprender a usar `fetch()` no JavaScript para buscar dados reais de servidores.

### Bancos NoSQL

Explorar o MongoDB, um banco de dados que armazena informações nativamente em formato JSON.

### Validação de Dados

Garantir a integridade do JSON usando ferramentas como Zod ou JSON Schema.

## FERRAMENTAS

---

### JSONPlaceholder

API gratuita para testes e prototipagem.

### JSONLint

Validador e formatador de JSON online.

### Postman

Plataforma para testar requisições de API.

# MISSÃO CUMPRIDA

---

Você agora fala a **língua da web**.  
Pratique, explore e construa coisas incríveis.

**DÚVIDAS?**

**PROFESSOR@EMAIL.COM**