

INTENSIVÃO START INFINITE

DIA 01
Resumo Conceitual



AGENDA

Aula Teórica

- Arquitetura de Sistemas Distribuídos
- Backend: API, REST e RESTful
- Frontend: HTML, CSS e Javascript
- Transferência de dados: JSON
- Banco de Dados: MER
- Pilhas de Desenvolvimento de Software
- CRUD Applications

Arquitetura de Sistemas Distribuídos

Sistemas Distribuídos são aqueles disponibilizados como serviço para máquinas clientes através de um ou mais máquinas servidoras. Em geral, temos três elementos principais que compõem estes sistemas.

APLICAÇÃO BACKEND

O código-fonte hospedado nos computadores **servidores** que irá realizar o **processamento** dos dados para transformá-los em informação, comunicando-se diretamente com o Banco de Dados para receber a entrada e devolver as saídas.

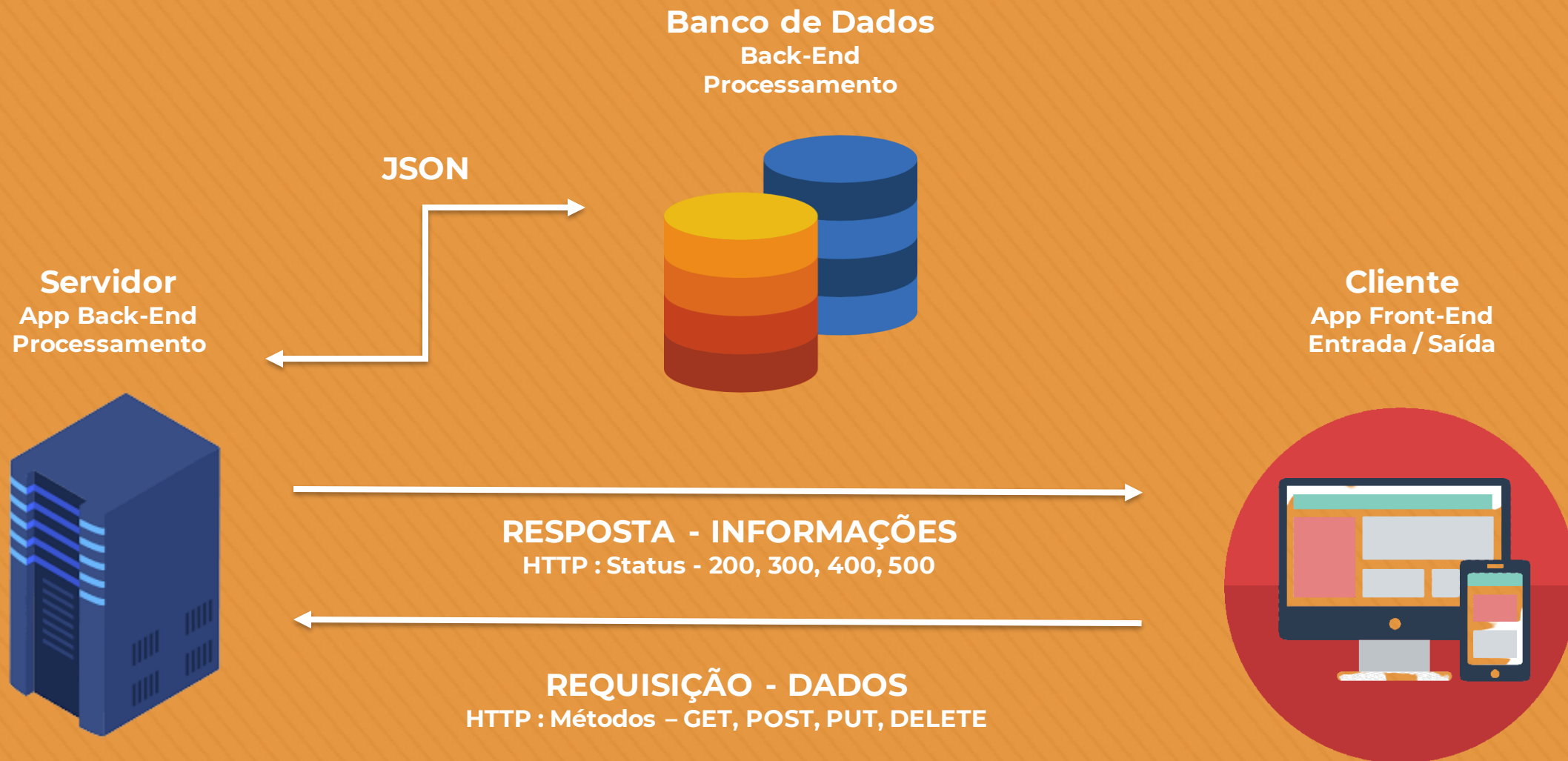
APLICAÇÃO FRONTEND

Páginas renderizadas nos dispositivos **clientes** que capturam os dados de **entrada** exibem as informações de **saída** ao usuário final. O frontend precisa passar pelo backend para se comunicar com o Banco de Dados.

BANCO DE DADOS

Onde ficam **armazenados** os dados que serão utilizados pelas aplicações, bem como o registro das transações realizadas no sistema. Geralmente o banco de dados é um serviço à parte e por isso mesmo possui um servidor dedicado a ele.

ARQUITETURA DE SISTEMAS DISTRIBUÍDOS



Backend: API, REST E RESTful

API – Application Programming Interface

A API é um tipo de arquitetura para desenvolvimento de aplicações backend. Essas aplicações recebem entradas de um ou mais sistemas em formato de dados e devolvem a saída também em formato de dados (geralmente em JSON) para os próprios sistemas solicitantes, ou ainda para terceiros.

REST e RESTful – Representational State Transfer

O padrão REST se baseia no protocolo HTTP. Segundo este protocolo, o dispositivo cliente pode fazer quatro tipos de requisições ao dispositivo servidor:

- GET – Obter dados
- POST – Escrever dados
- UPDATE – Atualizar dados
- DELETE – Excluir dados

As respostas que o cliente pode obter também são divididas em quatro status:

- 200 – Sucesso + Dados
- 300 – Sucesso + Mensagem
- 400 – Erro no Cliente
- 500 – Erro no Servidor

Assim, uma API RESTful é aquela que implementa as funções do protocolo HTTP.

Frontend: HTML, CSS e JS

HTML: Esqueleto

As páginas da Web são criadas utilizando a linguagem HTML para construir documentos que são renderizados pelos navegadores (Chrome, Edge, Mozilla, Safari, etc). Toda a estrutura do documento segue uma árvore de etiquetas (tags) para montar a visualização de acordo com determinada hierarquia. Mas nem só de HTML vive o Front-End: ainda temos o CSS e o JS.

CSS: Estilo

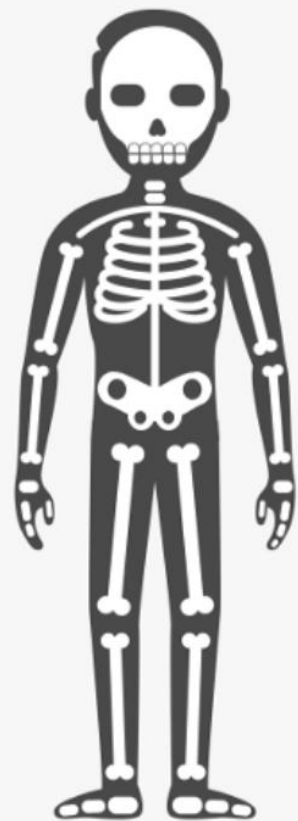
Através do CSS nós podemos referenciar tags ou elementos do HTML e modificar sua aparência, alterando por exemplo a fonte, a cor do fundo, adicionando bordas ou efeitos de movimento.

JS: Interação

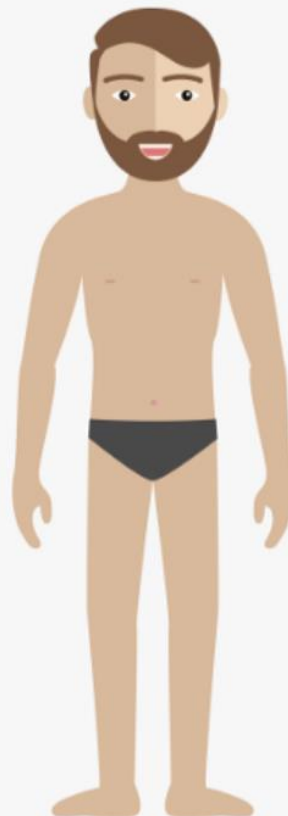
Embora o HTML seja uma estrutura estática, ele permite ao desenvolvedor criar scripts JS (Javascripts, daí o nome da linguagem). Eles são pequenos trechos de código que manipulam as tags e elementos do HTML respondendo a comandos do usuário.

Tanto as folhas de estilo CSS quanto os scripts JS podem ser importados pelo arquivo HTML, ou serem escritos diretamente no próprio arquivo.

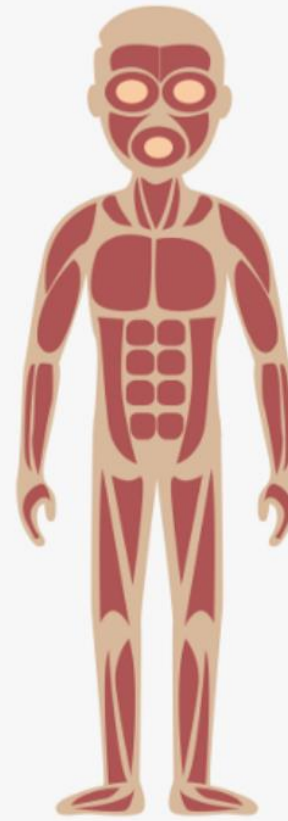
FRONTEND



HTML



CSS



Javascript

Transferência de dados: JSON

Para que cada componente do sistema distribuído possa enviar e receber dados de outros componentes, eles precisam padronizar a sua comunicação. Desta forma, os dados que trafegam entre o Banco de Dados, o Servidor Backend e o Cliente Frontend precisam atender ao mesmo formato. Nos últimos anos, o formato de dados que tem tomado conta da Web é o JSON.

O formato JSON é utilizado para estruturar dados em formato de texto e permitir a troca de dados entre aplicações de forma simples, leve e rápida. Os dados contidos em um arquivo no formato JSON devem ser estruturados por meio de uma coleção de pares com nome e valor ou ser uma lista ordenada de valores.

Chave: corresponde ao identificador do conteúdo, representado através de um texto entre aspas.

Valor: representa o conteúdo correspondente ao identificador, e pode conter os seguintes tipos de dados:

- String
- Number
- Boolean
- Array
- Object
- Null

Banco de Dados: MER

A principal representação na Modelagem de Dados é o **Modelo Entidade Relacionamento**, também conhecido como MER. Nele são feitas representações em alto nível dos elementos e interações que ocorrem no mundo real e que devem ser armazenados no sistema. A partir do MER são gerados posteriormente as Tabelas do Banco de Dados e as Classes da Aplicação Backend, pois há uma correspondência direta entre as três representações.

No banco de dados, a linguagem padrão é o SQL: Linguagem Estruturada de Consulta, criada pela IBM nos anos 1980 e amplamente utilizada até os dias atuais.

REPRESENTAÇÃO	MER	BANCO DE DADOS	APLICAÇÃO
Modelo de algo existente no mundo real	Entidade	Tabela	Classe
Representante de um modelo	Instância	Registro	Objeto
Características do modelo	Atributo	Campo	Atributo
Relacionamento entre modelos	Relacionamento	Chaves	Método

Pilhas de Desenvolvimento

No contexto do desenvolvimento de software, stack (pilha em inglês) é o conjunto de tecnologias utilizadas para criar aplicações. Fazem parte da pilha de desenvolvimento:

- Linguagem de Programação Backend
- IDE – Ambiente de Desenvolvimento Integrado (editor de código + debugger + console + linha de comando)
- Framework para Frontend ou Backend
- Sistema Gerenciador de Banco de Dados
- Ferramenta para testes
- Ferramenta para controle de versão
- Ferramenta para publicação
- Servidor de autenticação (opcional)

EXEMPLO

- Linguagem de programação Backend: **Javascript/Typescript**
- Framework para Backend: **Node, Express e TypeORM**
- Sistema Gerenciador de Banco de Dados: **PostgreSQL**
- Ferramenta para testes: **Jest**
- Ferramenta para controle de versão: **Git e Github**
- Ferramenta para publicação: **Heroku ou Digital Ocean**
- Servidor de autenticação: **Firebase**
- IDE: **Visual Studio Code**

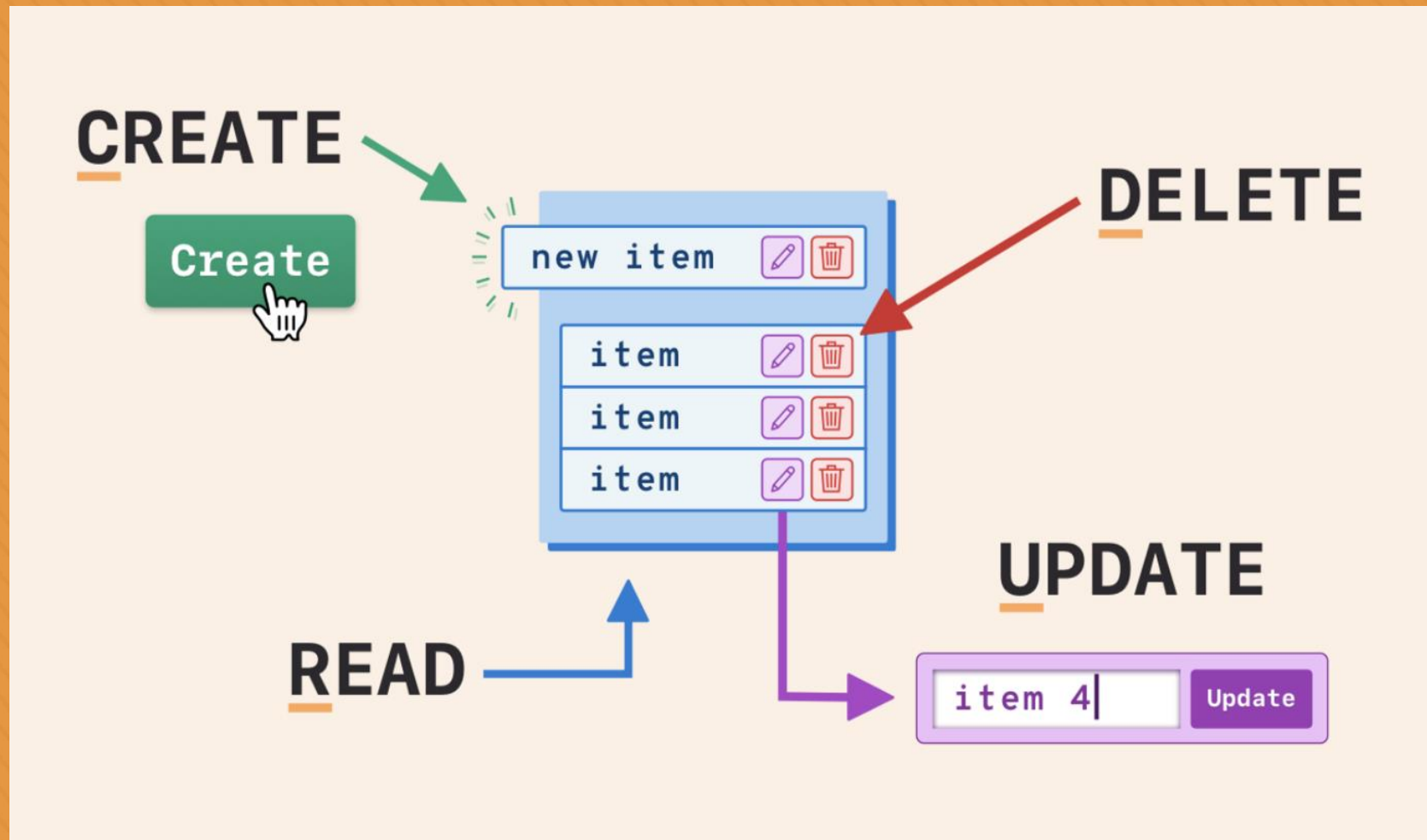
CRUD Applications

CRUD é um acrônimo para as 4 funções básicas que um software deve executar: Create (criar um item), Read (visualizar um item), Update (atualizar um item) e Delete (excluir um item). Em cada ambiente do Sistema Distribuído, essas operações são executadas de maneira distinta.

Em uma API RESTful, são implementadas as funções CRUD para cada uma das entidades do sistema. Entidade, como sabemos, é a representação de algo que existe no mundo real. Assim, um sistema de cadastro de notas de estudantes terá como entidades, por exemplo: Alunos, Professores, Cursos, Turmas, Aulas e Atividades. E para cada uma delas serão implementadas as funções CRUD.

AMBIENTE	CREATE	READ ALL	READ ONE	UPDATE	DELETE
BANCO DE DADOS	Insert	Select(*)	Select WHERE id	Update WHERE id	Delete WHERE id
BACKEND	Create()	ReadAll()	ReadOne(id)	Update(id)	Delete(id)
HTTP	Post	Get	Get(id)	Put(id)	Delete(id)
FRONTEND	Adicionar	Listar	Detalhes	Atualizar ou Editar	Excluir

CRUD APPLICATIONS



danielle.carvalho@go.senac.br

Serviço Nacional de
Aprendizagem
Comercial - Goiás

go.senac.br
senacgoias
senacgoias

