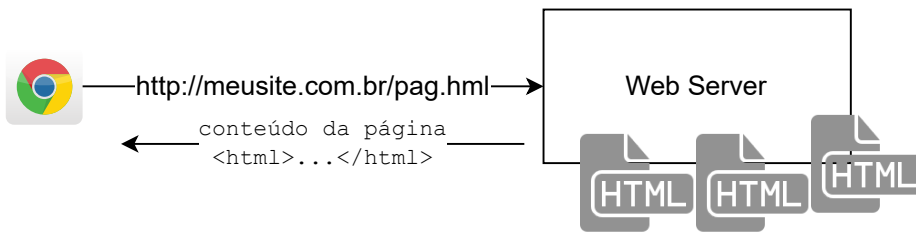


Web Server x Application Server



Componentes da URL

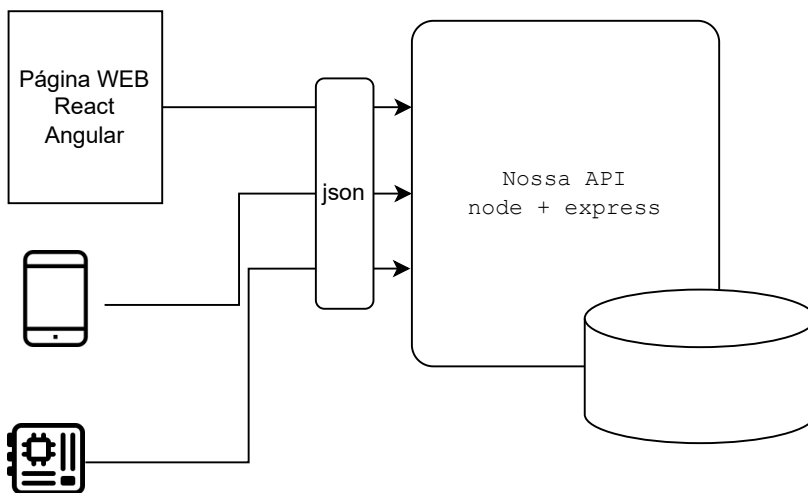
protocolo: HTTP
endereço da máquina: IP ou nome
porta: 80 (pode ser omitida)
recurso: pag.html (sempre está organizado de forma hierárquica)

App Server

/recurso é mapeado para uma function javascript
ao acessar o recurso, eu executo a função e a mensagem de retorno é o resultado da execução da função

Formato JSON

```
{
  "atributo": "valor texto",
  "atributo": 123456,
  "atributo": {
    "atributo": "valor"
  }
}
```



Lib x API x Fw

Biblioteca = 1 ou + arquivos com funções escritas na mesma linguagem que eu uso pra fazer meus sistemas, de modo que ela automatizam algumas tarefas (uso mais geral possível)

API - biblioteca para eu integrar o meu sistema com outros sistemas

Framework - é um conjunto de bibliotecas que possuem uma estrutura pré-fabricada (e por conta disso você precisa seguir suas especificações e seus padrões)

Endpoints

www.meusite.cmo.br:3000/produtos
GET --> recupera
POST --> insere
PUT --> altera
DELETE --> apaga

Modelo de Maturidade de Richardson

Level 0 - nenhuma organização

www.meusite.com.br/cadastrarProduto (POST)

www.meusite.com.br/listarProdutos (GET)

Level 1 - começo a ter alguma hierarquia na recuperação da informação

GET

/produtos --> todos os produtos

/produtos/1 --> detalhes do produto numero 1

Level 2 - hierarquia porém diferenciando os métodos HTTP

CRUD -> Create/Read/Update/Delete (BD)

mapear as operações do CRUD com os métodos HTTP

C (create) --> POST /produtos POST --> incluir

R (read) --> GET /produtos GET --> listar

U (update) --> PUT /produtos PUT --> aterar

D (delete) --> DELETE /Produtos DELE --> apagar

Arquitetura da API

tudo fica na "pasta raiz"

/src

-- app.js --> unica responsabilidade -> iniciar a API

/src/routes

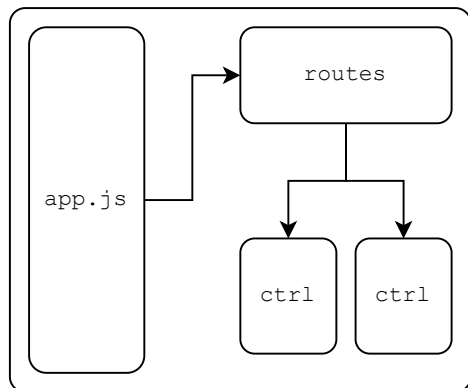
index.js --> este é o arquivo que conterà as rotas definidas (todos os endpoints da minha aplicação)

/src/controllers

produtoController.js --> cada controller possui a lógica para tratar as requisições

clienteController.js

pedidoController.js



rotas
associa
url
a métodos
(functions)
dos
controllers

controllers
possuir métodos para
realizar o CRUD usando
os models

para cada Model, tenho
um controller

Model (tabela) produto
produtoController

Model (tabela) depto
deptoController

Model(tabela) cliente
clienteController

Tabelas no banco ==> estruturas no Javascript (MODEL)

