

Sistemas Distribuídos

Desenvolvimento Web

emerson@paduan.dev.br

WEB

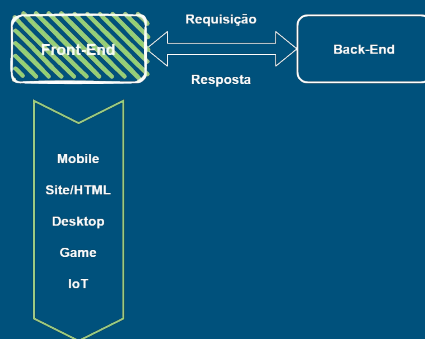
Desenvolvimento na WEB com independência de plataforma de consumo.



emerson@paduan.dev.br

WEB

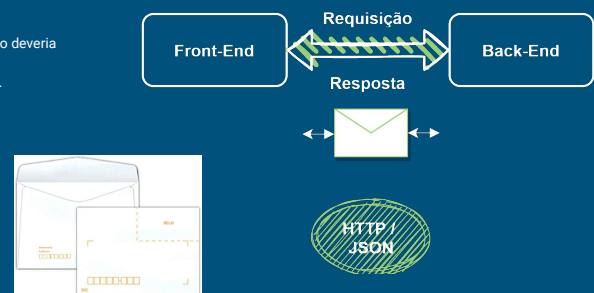
FrontEnd pode ser diversas tecnologias diferentes.



emerson@paduan.dev.br

WEB

A maneira de comunicação deveria atender a maior gama de possibilidades de clientes.



emerson@paduan.dev.br

Algumas tecnologias

emerson@paduan.dev.br

REST (Representational State Transfer)

Estilo de arquitetura de software que define um conjunto de regras para construção de serviços Web.

Quando um serviço atende a estas regras dizemos que ele é RESTful.

emerson@paduan.dev.br

Regras

1. Identificação dos recursos: cada um dos recursos da aplicação deve possuir uma identificação única utilizando o conceito de URI:

`http://localhost:8080/users;`

emerson@paduan.dev.br

Regras

2. Utilização dos métodos HTTP: além de ter a identificação do recurso, a aplicação deve prover o método HTTP a ser utilizado para consumir o recurso.

GET	Obter os dados de um recurso
POST	Criar um novo recurso
PUT	Atualizar os dados de um recurso
DELETE	Excluir um recurso

emerson@paduan.dev.br

Regras

3. Comunicação cliente / servidor sem estado (Stateless):

O servidor não deve guardar informações do cliente (consumidor) em memória. O cliente deve passar em cada requisição toda a informação necessária para consumir um recurso.

emerson@paduan.dev.br

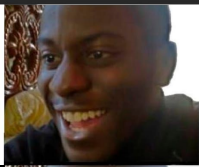
códigos HTTP

Classe	Descrição
2XX	Respostas de sucesso
4XX	Erros na requisição do cliente
5XX	Erros no servidor

emerson@paduan.dev.br

Use corretamente os códigos

Response code: 200



**Response body:
{error: "....."}**



emerson@paduan.dev.br

códigos HTTP

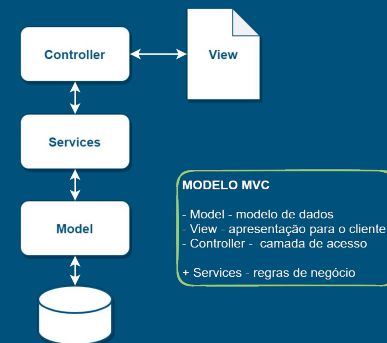
Exemplo	Descrição
200	OK
201	CREATED
400	BAD REQUEST
401	UNAUTHORIZED / unauthenticated
403	FORBIDDEN
500	Internal Server Error

emerson@paduan.dev.br

Arquitetura

emerson@paduan.dev.br

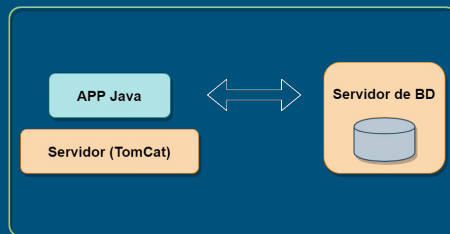
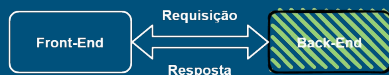
Desenvolvimento em camadas



emerson@paduan.dev.br

WEB

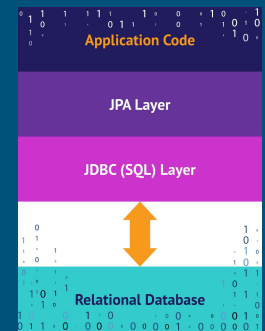
Teremos dois servidores rodando no Back-End.



emerson@paduan.dev.br

Java Persistence API (JPA)

Java Persistence API (ou simplesmente JPA) é uma API padrão da linguagem Java que descreve uma interface comum para frameworks de persistência de dados. A JPA define um meio de mapeamento objeto-relacional (ORM - Object-Relational Mapping) para objetos Java simples e comuns (POJOs)



emerson@paduan.dev.br


```
public class Usuario {
    private int id;
    private String nome;
    private String email;

    public Usuario( int id, String nome, String email) { /*... */ }

    public void setName(String nome){
        this.nome = nome;
    }

    public String getName(){
        return this.nome;
    }

    // ...Outros getters and setters...
}
```

emerson@paduan.dev.br

SEM JPA

Observe a construção da instrução de gravação dos dados no BD.

// Error handling e connection omitidos para abreviar

```
Usuario usuario = new Usuario ( 0, "Emerson Paduan", "emerson@paduan.pro.br" );
```

```
String query = " insert into usuario (id, nome, email) values (?, ?, ?)";
```

```
PreparedStatement preparedStmt = conn.prepareStatement( query );
preparedStmt.setInt (1, usuario.getId() );
preparedStmt.setString (2, usuario.getNome() );
preparedStmt.setString (3, usuario.getEmail() );
```

```
preparedStmt.execute();
```

emerson@paduan.dev.br

COM JPA

Observe a construção da instrução de gravação dos dados no BD

```
Usuario usuario = new Usuario ( 0, "Emerson Paduan", "emerson@paduan.pro.br" );  
usuarioRepository.save( usuario );
```

emerson@paduan.dev.br

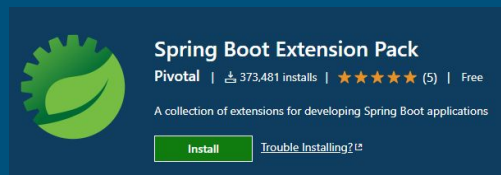
Spring & Spring Boot



emerson@paduan.dev.br

Preparando o ambiente

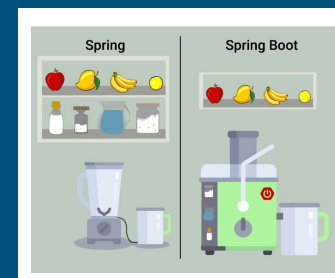
Instalar a extensão



Overview : <https://code.visualstudio.com/docs/java/java-spring-boot>

emerson@paduan.dev.br

Spring & Spring Boot



emerson@paduan.dev.br