

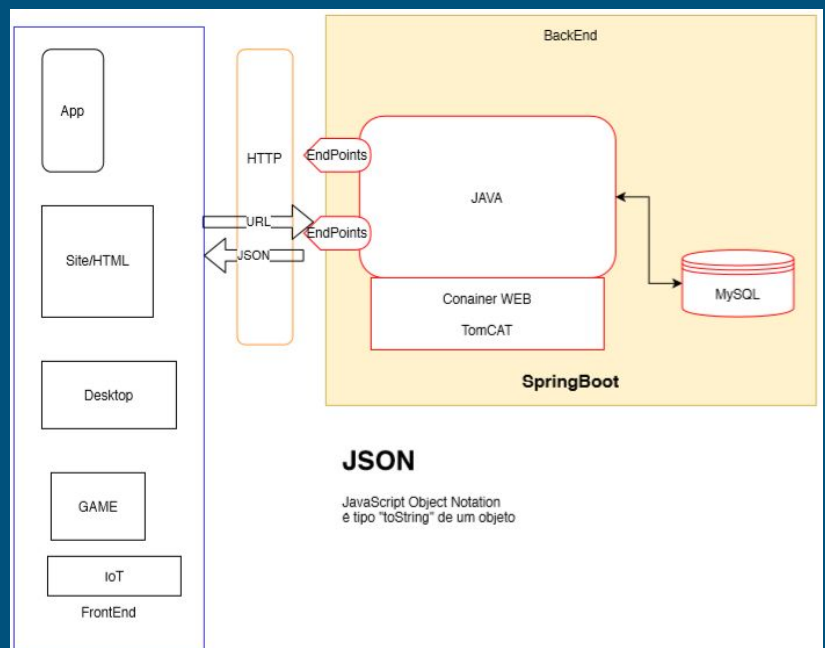
# Aula 08

## Desenvolvimento Web

emerson@paduan.pro.br

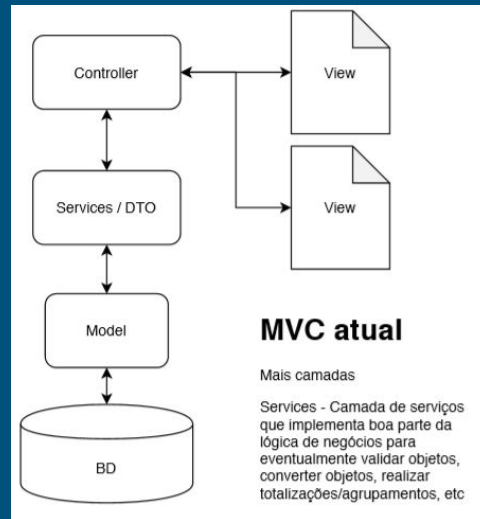
### WEB

Desenvolvimento na WEB com independência de plataforma de consumo.



emerson@paduan.pro.br

# Desenvolvimento em camadas



[emerson@paduan.pro.br](mailto:emerson@paduan.pro.br)

# Algumas tecnologias

[emerson@paduan.pro.br](mailto:emerson@paduan.pro.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.pro.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.pro.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.

<b>GET</b>	Obter os dados de um recurso
<b>POST</b>	Criar um novo recurso
<b>PUT</b>	Atualizar os dados de um recurso
<b>DELETE</b>	Excluir um recurso

emerson@paduan.pro.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.pro.br

# códigos HTTP

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

[emerson@paduan.pro.br](mailto:emerson@paduan.pro.br)

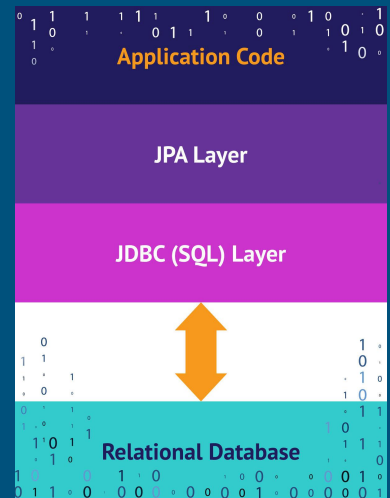
# códigos HTTP

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

[emerson@paduan.pro.br](mailto:emerson@paduan.pro.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.pro.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.pro.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.pro.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" );

usuarioManager.save( usuario );
```

emerson@paduan.pro.br

# Maven

O Apache Maven é uma ferramenta de automação e gerenciamento de projetos Java. Ele gerencia dependências e automatiza tarefas de compilação e execução.



[emerson@paduan.pro.br](mailto:emerson@paduan.pro.br)

# Spring & Spring Boot

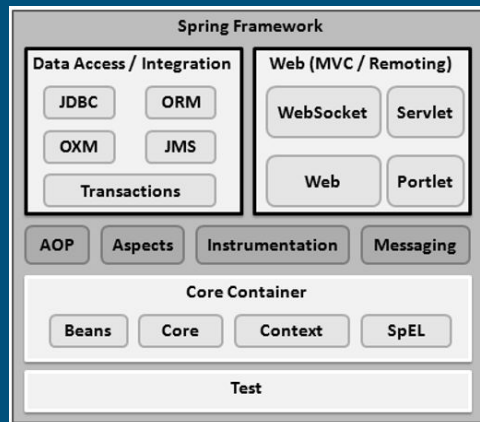


[emerson@paduan.pro.br](mailto:emerson@paduan.pro.br)



# Spring

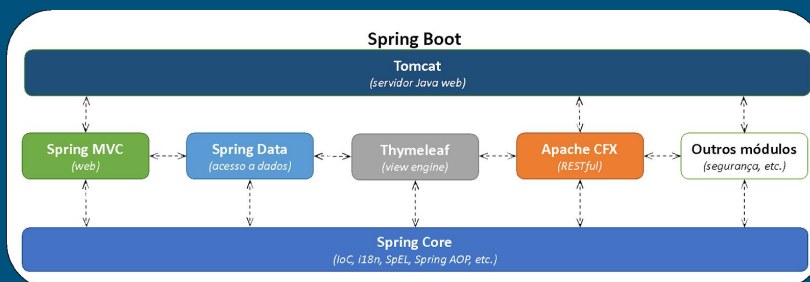
O Spring é um framework Java criado com o objetivo de facilitar o desenvolvimento de aplicações.



emerson@paduan.pro.br

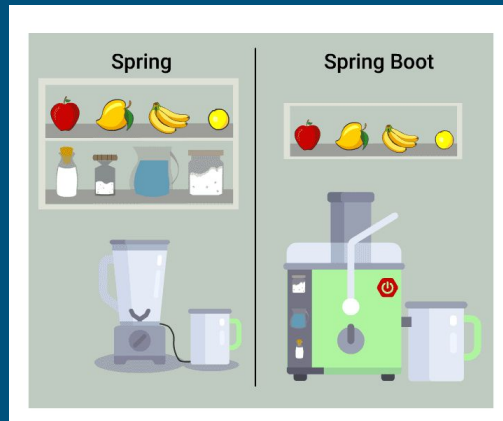
# Spring Boot

O Spring Boot fornece a maioria dos componentes baseados no Spring necessários em aplicações em geral de maneira pré-configurada, tornando possível termos uma aplicação rodando em produção rapidamente, com um menor esforço de configuração e implantação.



emerson@paduan.pro.br

# Spring & Spring Boot



emerson@paduan.pro.br

## Spring Initializer


<https://start.spring.io/>

The screenshot shows the Spring Initializer web form. It has a dark theme with the 'spring initializr' logo at the top left. The form is divided into several sections: 'Project' with radio buttons for 'Maven Project' (selected) and 'Gradle Project'; 'Language' with radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'; 'Spring Boot' with radio buttons for various versions, where '2.3.1' is selected; and 'Project Metadata' with input fields for 'Group' (com.example), 'Artifact' (demo), and 'Name' (demo). A 'Dependencies' section on the right shows 'No dependency selected'.

emerson@paduan.pro.br

# Preparando....

Instalar a extensão



**Spring Boot Extension Pack**  
Pivotal | 373,481 installs | ★★★★★ (5) | Free

A collection of extensions for developing Spring Boot applications

[Install](#) [Trouble Installing?](#)

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

emerson@paduan.pro.br

# Let's Code!

```
$this->rule_exists( $resource_details['id'], $role_details['id'] ) {  
    if ( $access == false ) {  
        // Remove the rule as there is currently no need for it  
        $details['access'] = !$access;  
        $this->_sql->delete( 'acl_rules', $details );  
    } else {  
        // Update the rule with the new access value  
        $this->_sql->update( 'acl_rules', array( 'access' => $access )  
    }  
}  
foreach( $this->rules as $key=>$rule ) {  
    if ( $details['role_id'] == $rule['role_id'] && $details['resource_id'] == $rule['resource_id'] ) {  
        if ( $access == false ) {  
            unset( $this->rules[ $key ] );  
        } else {  
            $this->rules[ $key ]['access'] = $access;  
        }  
    }  
}
```

emerson@paduan.pro.br