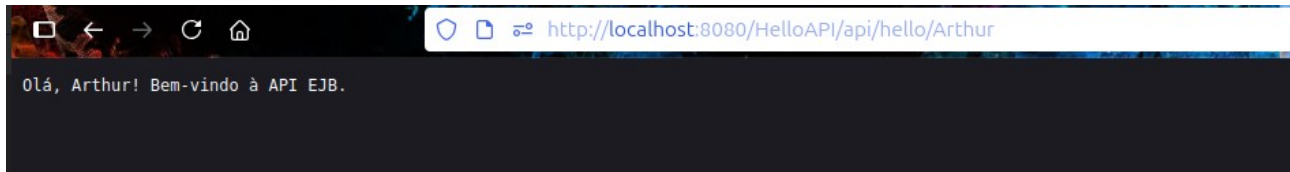


# Relatório – Laboratório com Jakarta EE, EJB e API RESTful

ALUNO: Arthur Roberto da Silva

MATRICULA: 536330



## Objetivo

Implementar uma aplicação Java corporativa utilizando **Jakarta EE 10**, criando uma **API RESTful com JAX-RS** integrada a um **EJB Stateless**. O projeto seria empacotado como um arquivo WAR e implantado em um servidor de aplicações compatível (WildFly), permitindo o consumo do serviço via requisição HTTP.

## Etapas Realizadas

### 1. Configuração do Projeto

- Projeto Maven criado com `groupId=com.exemplo` e `artifactId=HelloAPI`.
- Dependência adicionada ao `pom.xml` para Jakarta EE 10 (`jakarta.jakartaee-api`).
- Plugin `maven-compiler-plugin` configurado para usar Java 11.

### 2. Implementação da API

- `HelloService.java`: classe EJB com método `sayHello(String name)` retornando mensagem personalizada.
- `HelloResource.java`: classe REST com endpoint `/api/hello/{name}` que injeta o `HelloService`.
- `RestApplication.java`: define o caminho base da API como `/api`.

### 3. Compilação e Empacotamento

- Comando executado:  
**`mvn clean package`**
- Compilação finalizada com sucesso.
- Arquivo `HelloAPI.war` gerado na pasta `target/`.
- **Deploy no WildFly**
- WAR copiado para `standalone/deployments/`.

- WildFly implantou o projeto com sucesso, sem erros no log.
- **Teste da API**
- Acessado no navegador:

`http://localhost:8080/HelloAPI/api/hello/Arthur`

- Resposta obtida com sucesso:

Olá, Arthur! Bem-vindo à API EJB.

## Resultado Final

A prática foi **executada com sucesso**. Todos os passos — desde a configuração do projeto até o acesso ao endpoint REST — funcionaram corretamente. O uso de EJB com injeção via `@Inject` e o endpoint definido com JAX-RS foram devidamente registrados e respondidos.