



Tutorial de instalação do ecommerce web – MáximaTech

Requisitos mínimos

1. JDK 11
2. Postgre SQL 12
3. Apache Tomcat 8.5

Passo 1 – Configuração do Postgres:

1º Crie um usuário no banco de dados com o login: **postgres** e a senha: **cafe**

2º Crie um banco de dados chamado **ecommerce**.

Passo 2 Configuração do Tomcat

1º Acessar o Tomcat pelo endereço: <http://localhost:8080/manager/html> e entrar com as credenciais do seu tomcat.

Ao acessar, aparecerá a seguinte página:

The screenshot displays the Apache Tomcat Manager web interface. At the top, there's a navigation bar with links like 'List Applications', 'HTML Manager Help', 'Manager Help', and 'Server Status'. Below this is the 'Applications' table, which lists deployed applications with columns for Path, Version, Display Name, Running status, Sessions, and Commands. The table shows three applications: '/', '/docs', and '/manager'. Below the table is the 'Deploy' section, which includes fields for 'Context Path (required)', 'XML Configuration file path', and 'WAR or Directory path', along with a 'Deploy' button. There's also a 'WAR file to deploy' section with a 'Choose File' button and a 'Deploy' button. The 'Configuration' section includes a 'Re-read TLS configuration files' button and a 'TLS host name (optional)' field. At the bottom, there's a 'Diagnostics' section with a 'Find leaks' button and a warning message about triggering a full garbage collection.

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy
Deploy directory or WAR file located on server

Context Path (required):
XML Configuration file path:
WAR or Directory path:

WAR file to deploy

Select WAR file to upload No file chosen

Configuration
Re-read TLS configuration files

TLS host name (optional)

Diagnostics
Check to see if a web application has caused a memory leak on stop, reload or undeploy

This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

TLS connector configuration diagnostics

2º - Na parte **Deploy** do apache, clique no botão “Choose File” e selecione o arquivo “**ecommerce-api.war**” contido na pasta de instalação, esse arquivo será a API principal da aplicação web.

3º Logo após selecionar o arquivo, clique no botão “**Deploy**”, assim o apache realizará o deploy da API no servidor de aplicação.

Obs: O contexto da aplicação será definido com o próprio nome do arquivo: “**/ecommerce-api**”.

4º - Faça o mesmo procedimento com o arquivo: “**calculo-frete-api.war**”, também encontrado na pasta de instalação, para adicionar o microserviço de frete no servidor de aplicação.

Obs: O contexto da aplicação será definido com o próprio nome do arquivo: “**/calculo-frete-api**”.

Veja na imagem abaixo o local especificado para o deploy dos dois arquivos:

The screenshot shows the Apache Tomcat Manager web interface. The 'Deploy' section is active, and the 'WAR file to deploy' subsection is highlighted with a red box. This section contains a 'Select WAR file to upload' button, a 'Choose File' button, and a 'No file chosen' button. Below these buttons is a 'Deploy' button. The 'Configuration' section below it shows a 'Re-read TLS configuration files' button and a 'TLS host name (optional)' input field. The 'Diagnostics' section at the bottom shows a 'Find leaks' button and a 'TLS connector configuration diagnostics' button.

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file path:

WAR or Directory path:

WAR file to deploy

Select WAR file to upload No file chosen

Configuration

Re-read TLS configuration files

TLS host name (optional)

Diagnostics

Check to see if a web application has caused a memory leak on stop, reload or undeploy

This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

4º - Para a instalação é o do microserviço web “**ecommerce-front-end**”, encontrado na pasta de instalação. Selecione os arquivos da pasta “**ecommerce-front-end.zip**”, que já estão compilados e coloque em algum servidor que esteja na mesma máquina que os demais servidores para começar a utilizar. Pode ser utilizado como servidor, por exemplo, o ISS (Gerenciador de Serviços de Informações da Internet) do Windows para hospedar o micro serviço web.

Observações a serem feitas

1ª - Os arquivos foram compilados para executar em localhost.

2ª - As APIs são executadas na porta 8080 do apache tomcat.

Exemplo da aplicação principal:

<http://localhost:8080/ecommerce-api/>

Exemplo do microserviço de frete:

<http://localhost:8080/calculo-frete-api/>

Não foram adicionadas as variáveis de ambiente (enviroment e application.properties) de produção nos projetos.

Tecnologias utilizadas no desenvolvimento.

BackEnd

- Java 11
- Spring Boot 2.2
- Spring Data JPA
- Hibernate
- Maven
- PostgreSQL
- Apache Tomcat

FrontEnd

- Angular 8
- Angular Material
- Flex Layout