

VcRiquinho

O projeto pode ser executado no docker, a utilização do docker permite uma imensa facilidade para executar o projeto, permitindo criar um ambiente pronto para executar o projeto em questão de segundos. Abaixo temos o passo a passo para executar o projeto nesse formato:

Dependências:

- [Docker](#)
- [Apache Maven](#)

Entre no diretório raiz do projeto, abra um terminal e execute o comando:
docker-compose ps

Tendo esse possível resultado:

```
PS D:\PWE\vcRiquinho> docker-compose ps
NAME          IMAGE          COMMAND          SERVICE    CREATED   STATUS    PORTS
PS D:\PWE\vcRiquinho> |
```

Com isso temos certeza que estamos na pasta correta e com o docker funcionando.

Para executar o projeto basta usar os comandos:

mvn clean package

Tendo como resultado:

```
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 155, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- war:3.3.1:war (default-war) @ vcRiquinho ---
[INFO] Packaging webapp
[INFO] Assembling webapp [vcRiquinho] in [C:\Users\flavi\OneDrive\Documentos\Trabalho\wolke-estudos\PWE\vcRiquinho\target\vcRiquinho-1.0-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\flavi\OneDrive\Documentos\Trabalho\wolke-estudos\PWE\vcRiquinho\src\main\webapp]
[INFO] Building war: C:\Users\flavi\OneDrive\Documentos\Trabalho\wolke-estudos\PWE\vcRiquinho\target\vcRiquinho-1.0-SNAPSHOT.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.581 s
[INFO] Finished at: 2024-06-10T22:37:54-03:00
[INFO] -----
```

E depois:

docker-compose up -d

Tendo como resultado:

```
PS D:\PWE\vcRiquinho> docker-compose up -d
[+] Running 5/5
✓Network vcricinho_banco      Created           0.0s
✓Network vcricinho_default    Created           0.0s
✓Volume "vcricinho_dados"     Created           0.0s
✓Container vcricinho-db-1     Started           0.1s
✓Container vcricinho-web-1    Started           0.1s
PS D:\PWE\vcRiquinho> |
```

Pronto, o projeto foi executado.

Para verificar se tudo está conforme o planejado execute os seguintes passos com resultados que deveriam ocorrer:

1 - execute:

`docker-compose ps`

```
PS C:\Users\flavi\OneDrive\Documentos\Trabalho\wolke-estudos\PWE\vcRiquinho> docker-compose ps
NAME                IMAGE                COMMAND              SERVICE  CREATED      STATUS      PORTS
vcricinho-db-1      postgres:16          "docker-entrypoint.s..." db        14 seconds ago Up 13 seconds 0.0.0.0:5433->5432/tcp
vcricinho-web-1     vcricinho-web        "catalina.sh run"    web       14 seconds ago Up 12 seconds 0.0.0.0:8080->8080/tcp
```

2- execute:

`docker-compose exec db psql -U postgres -f /scripts/check.sql`

Obs 1: Isso é apenas uma parte do resultado

Obs 2: Dependendo do sistema operacional/terminal utilizado será necessário apertar "q" para sair da visão do resultado.

Obs 3: Em caso de falha do banco de dados, será necessário limpar ou remover o volume `vcricinho_dados`

```
PS C:\Users\flavi\OneDrive\Documentos\Trabalho\wolke-estudos\PWE\vcRiquinho> docker-compose exec db psql -U postgres -f /scripts/check.sql
List of databases

```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	ICU Locale	ICU Rules	Access privileges
dbtest_vcricinho	postgres	UTF8	libc	en_US.utf8	en_US.utf8			
postgres	postgres	UTF8	libc	en_US.utf8	en_US.utf8			
template0	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=c/postgres +
template1	postgres	UTF8	libc	en_US.utf8	en_US.utf8			postgres=CTc/postgres +
vcricinho	postgres	UTF8	libc	en_US.utf8	en_US.utf8			=c/postgres +
								postgres=CTc/postgres

(5 rows)

You are now connected to database "vcricinho" as user "postgres".

```
Table "public.conta"

```

Column	Type	Collation	Nullable	Default
id	integer		not null	nextval('conta_id_seq'::regclass)
documento_titular	character varying(30)		not null	
montante_financieiro	numeric(10,2)		not null	
id_produto	integer		not null	
cdi	numeric(10,2)		not null	
tipo_conta	tipo_conta		not null	
ativo	boolean			true
data_criacao	timestamp without time zone			CURRENT_TIMESTAMP
data_modificacao	timestamp without time zone			CURRENT_TIMESTAMP

Indexes:

```
"conta_pkey" PRIMARY KEY, btree (id)
"sem_duplicidade_conta_documento" UNIQUE CONSTRAINT, btree (documento_titular, tipo_conta)
```

Foreign-key constraints:

```
"conta_documento_titular_fkey" FOREIGN KEY (documento_titular) REFERENCES pessoa(documento_titular)
"conta_id_produto_fkey" FOREIGN KEY (id_produto) REFERENCES produto(id_produto)
```

Triggers:

```
set_data_modificacao BEFORE UPDATE ON conta FOR EACH ROW EXECUTE FUNCTION atualiza_data_modificacao()
```

Usando o projeto:

Após a execução bem sucedida dos containers, entre em:

<http://localhost:8080/vcRiquinho/home>

Ao entrar nesta tela você terá opções de login e cadastrar-se



Ao apertar será direcionado para a página correspondente.

Login: Você poderá se logar em uma conta criada por você ou uma conta já cadastrada.

Usuário:

email: empresas@email.com

senha: password2

Administrador:

email: mariasantos@email.com

senha: password1

As senhas são criptografadas no banco, não permitindo a visualização das mesmas

Ao se logar como usuário

Você terá as opções:



Simulação: Direciona para uma página onde o usuário pode simular seus ganhos, baseando-se nas suas contas cadastradas

Profile: Acessa o perfil do usuário, podendo deletar o perfil permanentemente, deletar ou inserir novas contas

Logout: É desconectado da conta atual

Ao se conectar como administrador você tem acesso a duas novas opções, “simulação geral” e “painel de produtos”:



Simulação geral: Permite a simulação de ganhos de todas as contas cadastradas no banco de uma vez

Painel de produtos: Painel de controle sobre os produtos, onde pode se adicionar, remover e modificar os produtos já cadastrados.

Baixando dependências:

Docker:

Baixe o docker no site oficial:

<https://www.docker.com/products/docker-desktop/>

Apache tomcat:

Baixe o Apache tomcat no site oficial:

<https://tomcat.apache.org/download-90.cgi>

9.0.89

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)


Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

Postgres:

Baixe o postgresSQL no site oficial:

<https://www.postgresql.org/download/>


Downloads 


PostgreSQL Downloads


PostgreSQL is available for download as ready-to-use packages or installers for various platforms, as well as a source code archive if you want to build it yourself.


Packages and Installers


Select your operating system family:

Linux


macOS


Windows


BSD


Solaris


Source code

The source code can be found in the main [file browser](#) or you can access the source control repository directly at git.postgresql.org. Instructions for building from source can be found in the [building from source](#) page.

Beta/RC Releases and development snapshots (unstable)

There are source code and binary [packages](#) of beta and release candidates, and of the current development code available for testing and evaluation of new features. Note that these are for **testing purposes only**, and not for production systems.

3rd party distributions

Caso tenha dúvidas consulte:

Windows:

<https://www.youtube.com/watch?v=UbX-2Xud1JA>

Linux:

<https://www.hostinger.com.br/tutoriais/instalar-postgresql-ubuntu-20-04>

https://aprendendoprogramar.com.br/tutoriais/postgresql/02_instalacao/01_baixando_e_instalando/#instalando-o-postgresql-no-centos

Apache Maven

Windows:

Acesse a página de download oficial do maven:

<https://maven.apache.org/download.cgi>

Na parte files:

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.9.6-bin.tar.gz	apache-maven-3.9.6-bin.tar.gz.sha512	apache-maven-3.9.6-bin.tar.gz.asc
Binary zip archive	apache-maven-3.9.6-bin.zip	apache-maven-3.9.6-bin.zip.sha512	apache-maven-3.9.6-bin.zip.asc
Source tar.gz archive	apache-maven-3.9.6-src.tar.gz	apache-maven-3.9.6-src.tar.gz.sha512	apache-maven-3.9.6-src.tar.gz.asc
Source zip archive	apache-maven-3.9.6-src.zip	apache-maven-3.9.6-src.zip.sha512	apache-maven-3.9.6-src.zip.asc

Aperta na opção “binary zip archive”:

Binary tar.gz archive	apache-maven-3.9.6-bin.tar.gz
Binary zip archive	apache-maven-3.9.6-bin.zip

Você será perguntado onde gostaria de armazenar o arquivo baixado.

Obs: Recomendo que seja armazenado na raiz “C:/”, como na imagem a seguir

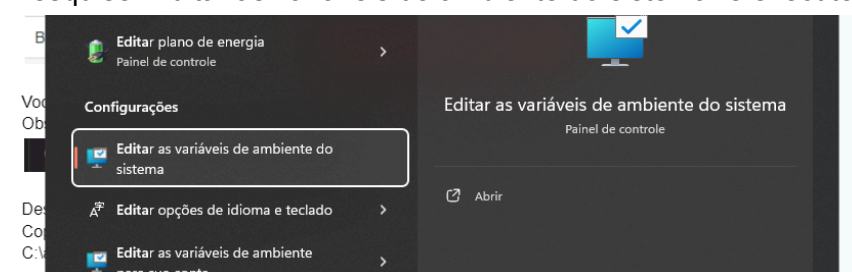
C:\apache-maven-3.9.6-bin.zip

Descompacte o arquivo baixado

Copie o caminho do arquivo descompactado, que possivelmente vai ser dessa forma:

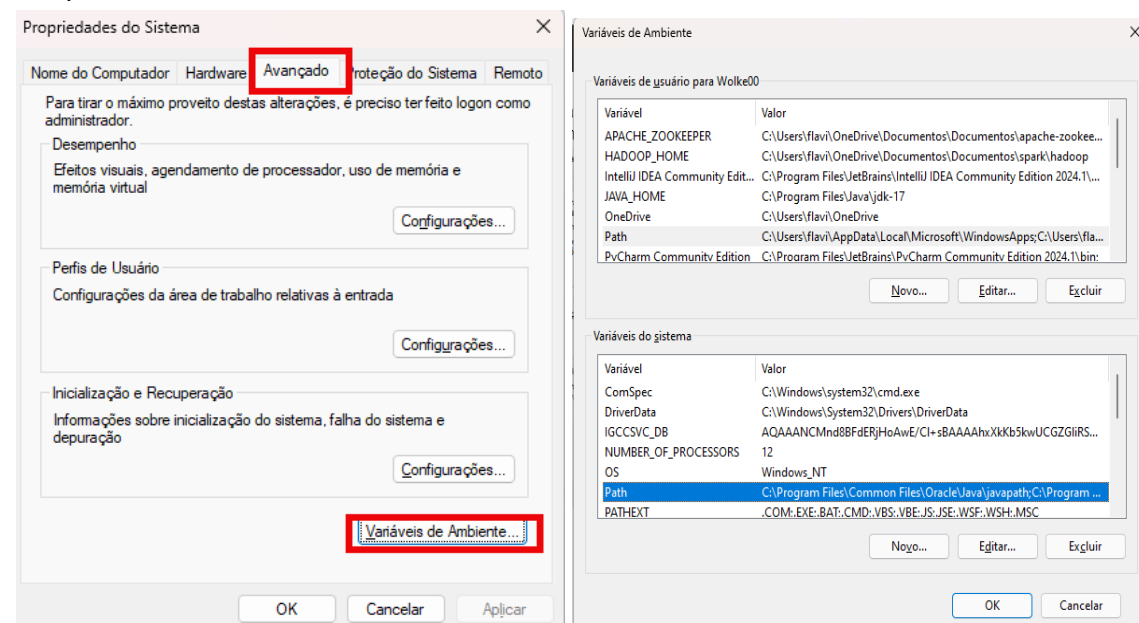
C:\apache-maven-3.9.6

Pesquise “Editar as variáveis de ambiente do sistema” e execute



Em “avançado” aperte “variáveis de ambiente”, na tela que abrir (imagem abaixo) procure na em “Variáveis de sistema” a opção Path, aperte editar, depois adicione o caminho:

C:\apache-maven-3.9.6\bin



C:\Program Files\DOCKER\DOCKER\resources\

C:\apache-maven-3.9.6\bin

Linux:
Se vira