



**SPRING**  
Framework

## Framework Spring

Maior framework para desenvolvimento de aplicações Java. Composto de diversas bibliotecas e módulos voltados para criação de aplicações Java, principalmente para Web.

Dentro do **Spring Framework**, temos 2 principais tecnologias para desenvolvimento de aplicações web:

- **Spring MVC**

Conjunto de bibliotecas do Spring para criação de projetos web através do padrão **Model, View e Controller**.

- **Spring Boot**

Conjunto de bibliotecas do Spring para criação de projetos web baseados em arquitetura de microserviços, ou seja, projetos que funcionam como **APIs REST**.

Além destas tecnologias para desenvolvimento web, o **Spring Framework** também possui módulos para:

- Envio de emails
- Acesso a banco de dados, através da JPA e Hibernate
- Etc.

---

Escopo do projeto:

### Sistema de controle de finanças

- É preciso desenvolver uma aplicação que permita a um usuário gerenciar suas contas e movimentações financeiras. O sistema deverá permitir:
  - O usuário deverá criar uma conta de acesso no sistema
  - O usuário, através da sua conta, poderá fazer um login na aplicação
  - O sistema deverá permitir ao usuário cadastrar suas contas, consultá-las, editá-las e excluí-las
  - As contas poderão ser a pagar ou a receber e deverão ter:

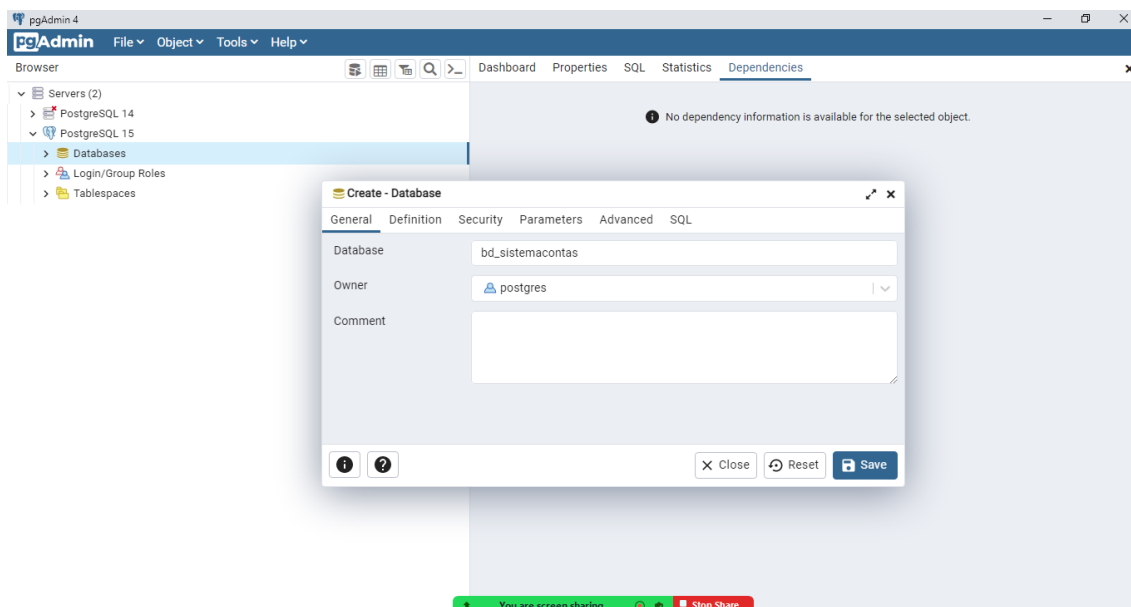
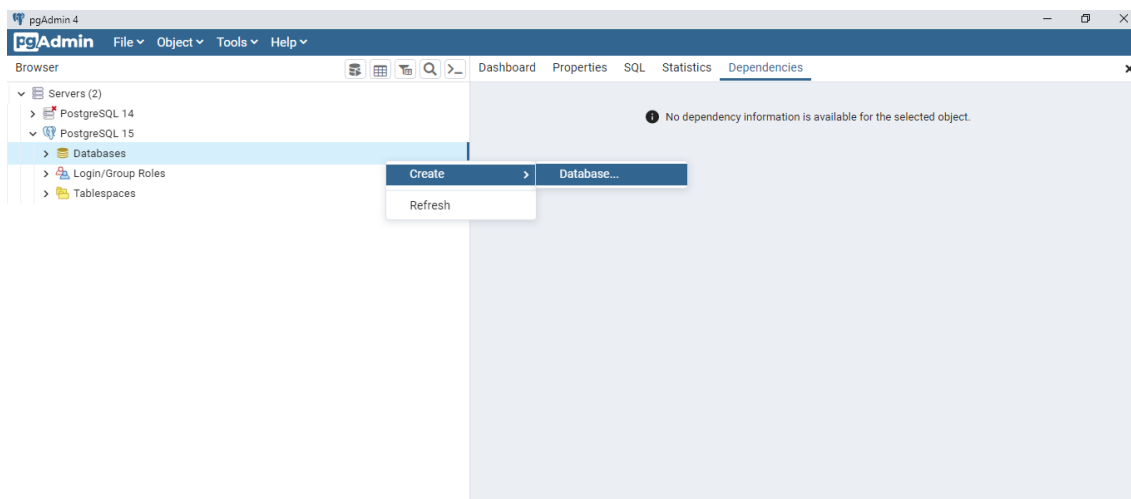
- Nome, Data, Valor, Tipo (Pagar ou Receber), Observações e Categoria (que poderá ser cadastrado pelo usuário)
- O sistema deverá gerar relatórios e dashboards para o usuário consolidando dados de suas contas
- O sistema também deverá permitir ao usuário alterar sua senha de acesso.

## Parte 01: Preparando o ambiente

- Criando o banco de dados do projeto
- Criar um repositório no GITHUB para publicar o código fonte do projeto

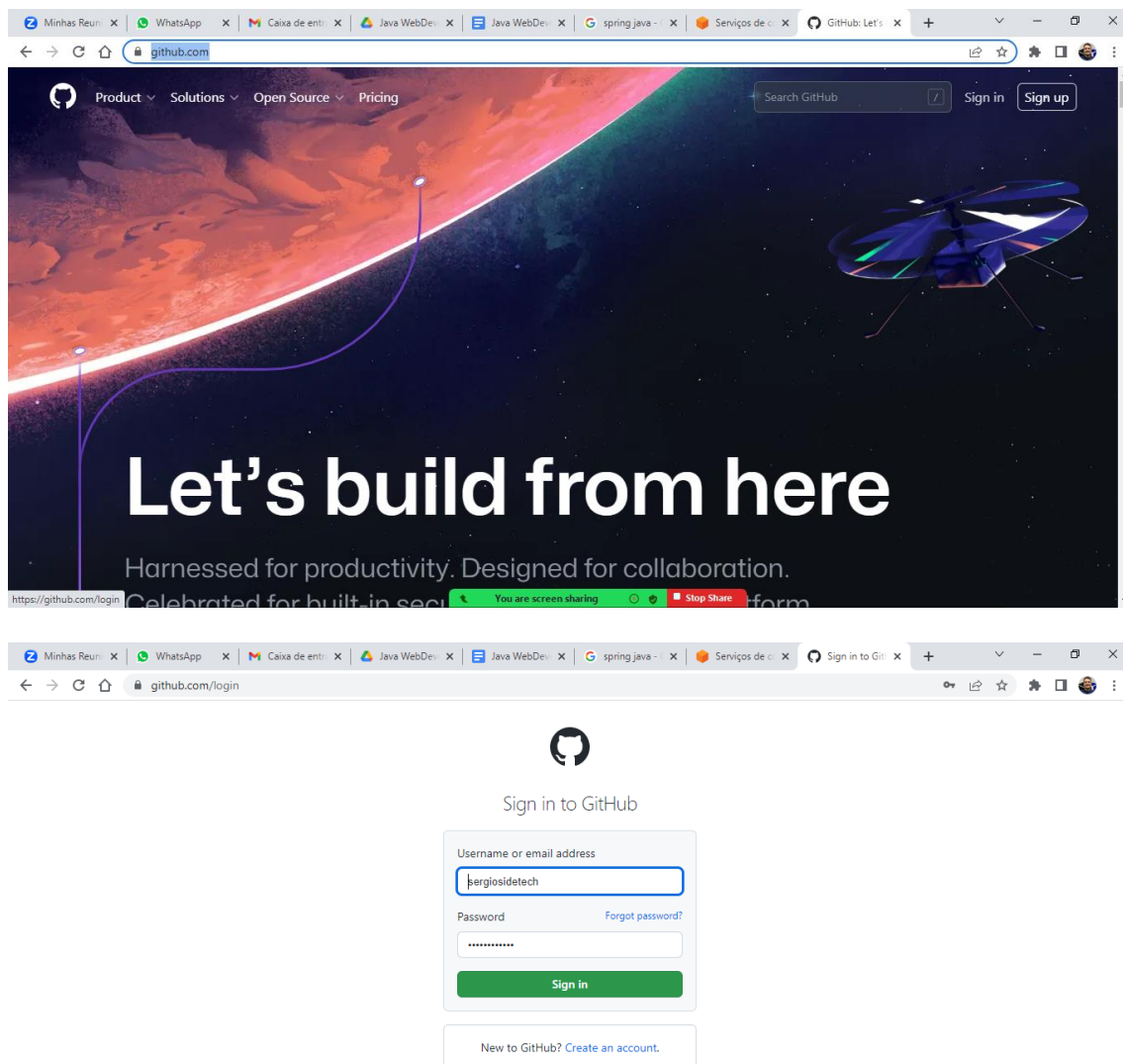
### PgAdmin

Nome do banco de dados: **bd\_sistemacontas**



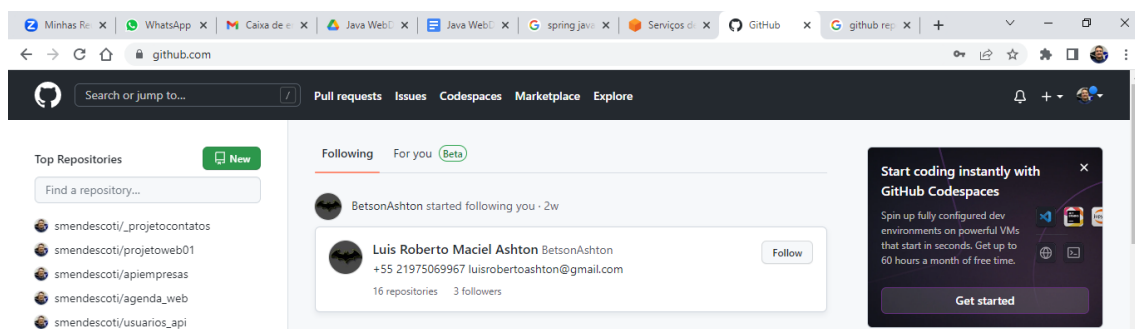
## Criando um repositório no GITHUB para publicar o código fonte do projeto:

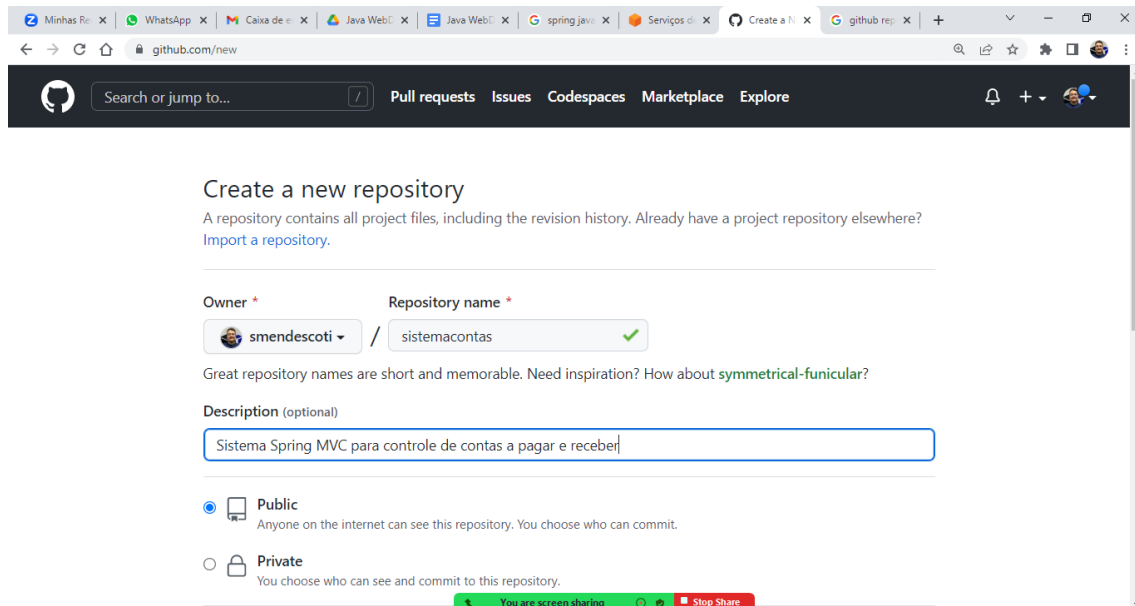
<https://github.com/>



## Repositórios no GitHub

Recurso através do qual podemos publicar o código fonte de um projeto. Para cada projeto que iremos salvar no GITHUB precisamos criar um repositório.






Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner <sup>\*</sup> Repository name <sup>\*</sup>

 smendescoti / sistemacontas ✓

Great repository names are short and memorable. Need inspiration? How about [symmetrical-funicular?](#)

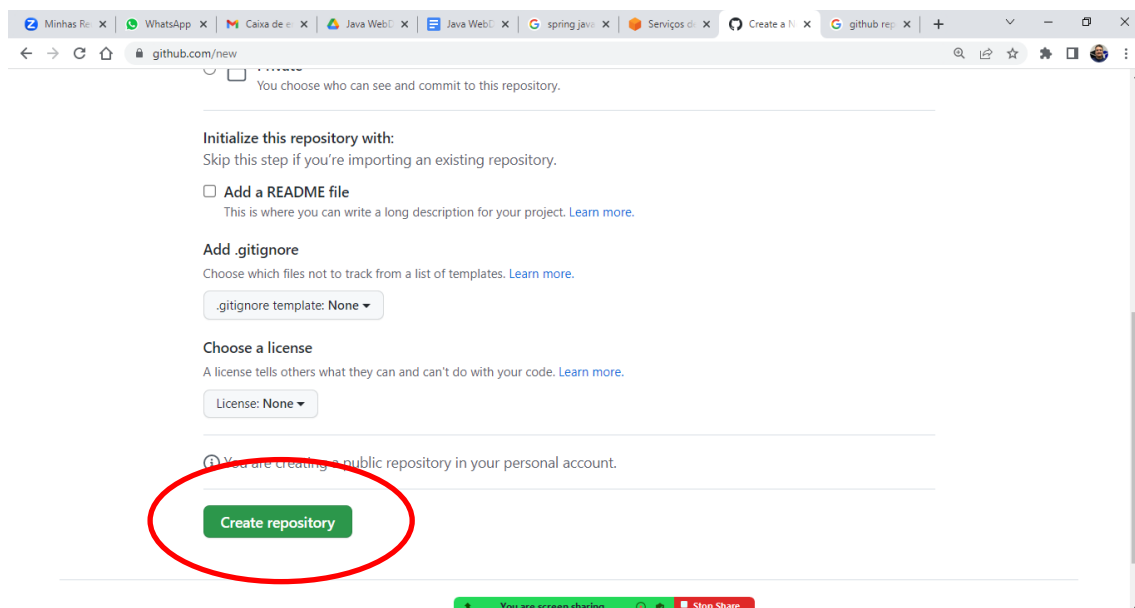
Description (optional)

Sistema Spring MVC para controle de contas a pagar e receber

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

You are screen sharing Stop Share



You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

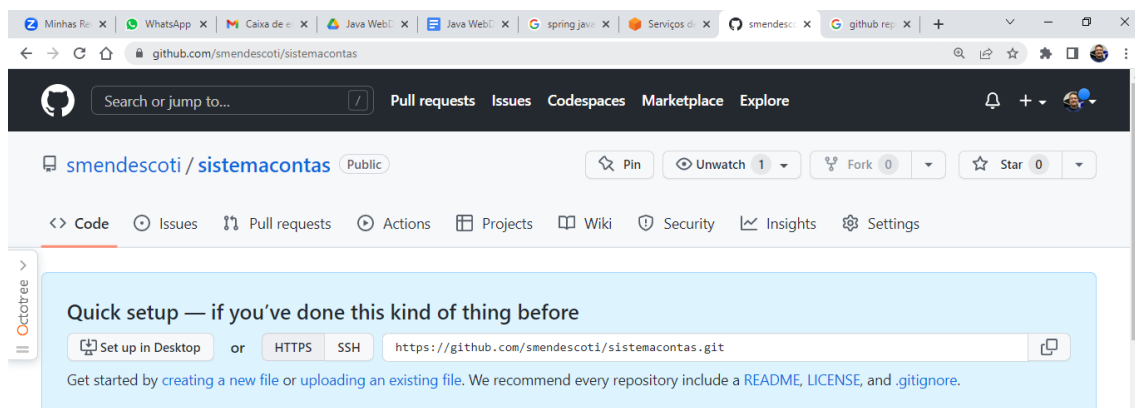
License: None

☒ You are creating a public repository in your personal account.

Create repository

You are screen sharing Stop Share

<https://github.com/smendescoti/sistemacontas>



Minhas Re: x WhatsApp x Caixa de e: x Java Web: x Java Web: x spring jav: x Serviços d: x smendesc: x github rej: x

github.com/smendescoti/sistemacontas

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

smendescoti / sistemacontas Public

Pin Unwatch 1 Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/smendescoti/sistemacontas.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## Parte 02: Criando o projeto

Eclipse IDE



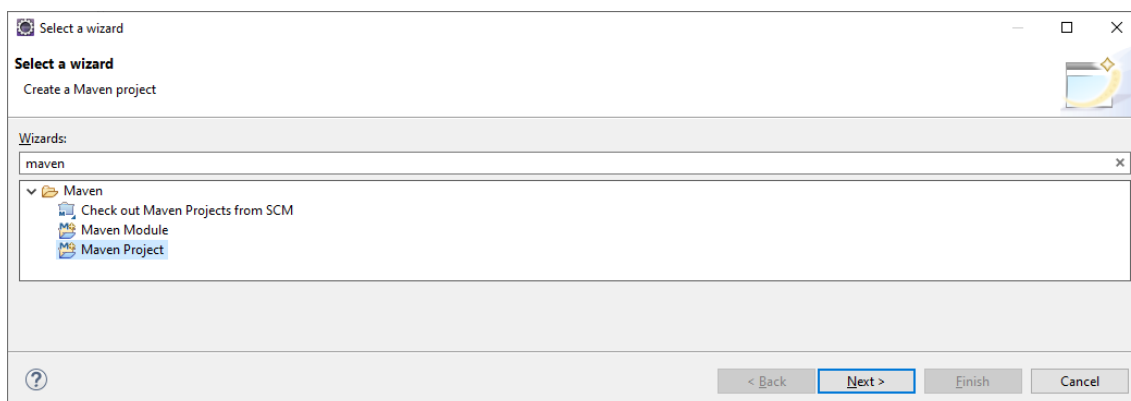
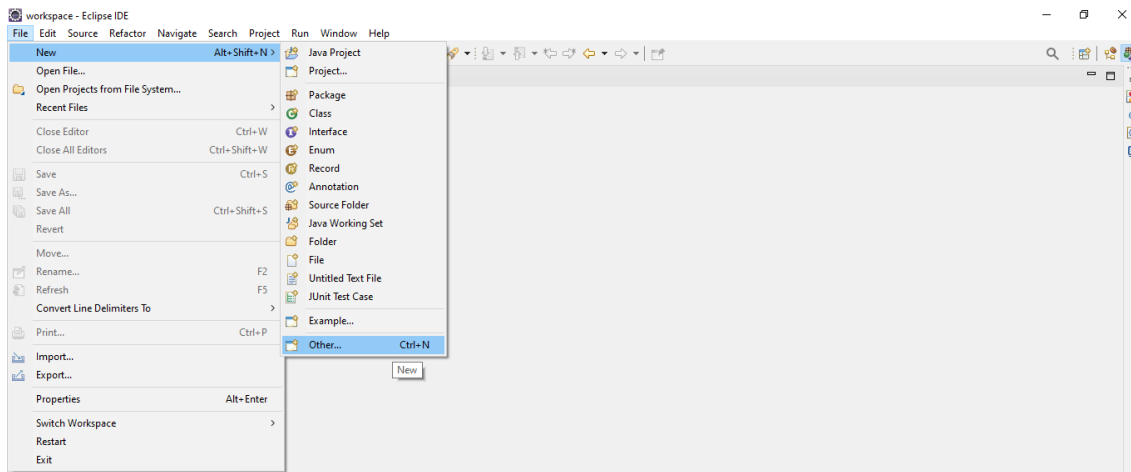
### Apache Maven

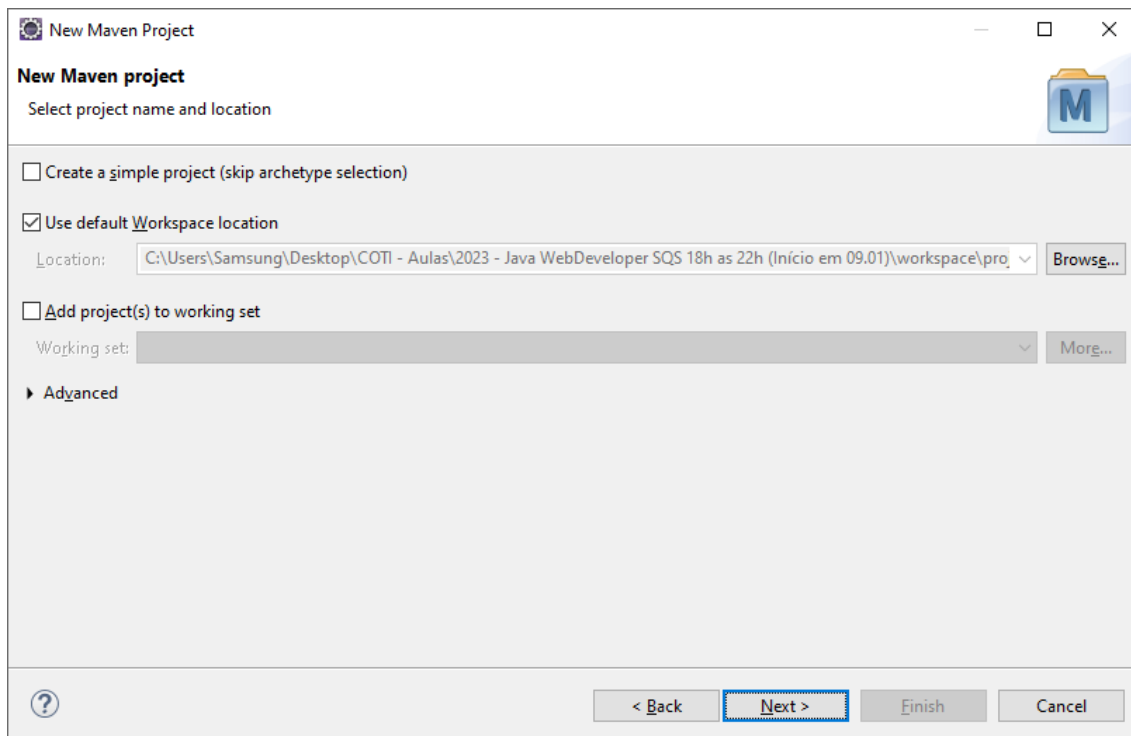
Framework Java utilizado para gerenciar os pacotes e bibliotecas utilizados em um projeto. Utiliza um arquivo de configuração no projeto para que possamos mapear quais bibliotecas serão instaladas na aplicação.

Este arquivo chama-se **/pom.xml**

Inclusive, o Maven já possui um recurso para criar projetos Java com o mínimo de configuração necessária (Hello World)

Sendo assim, vamos utilizar o Maven para criar um projeto Hello World baseado em Spring MVC.





New Maven Project

Select project name and location

☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location: C:\Users\Samsung\Desktop\COTI - Aulas\2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)\workspace\proj... Browse...

☐ Add project(s) to working set

Working set: More...

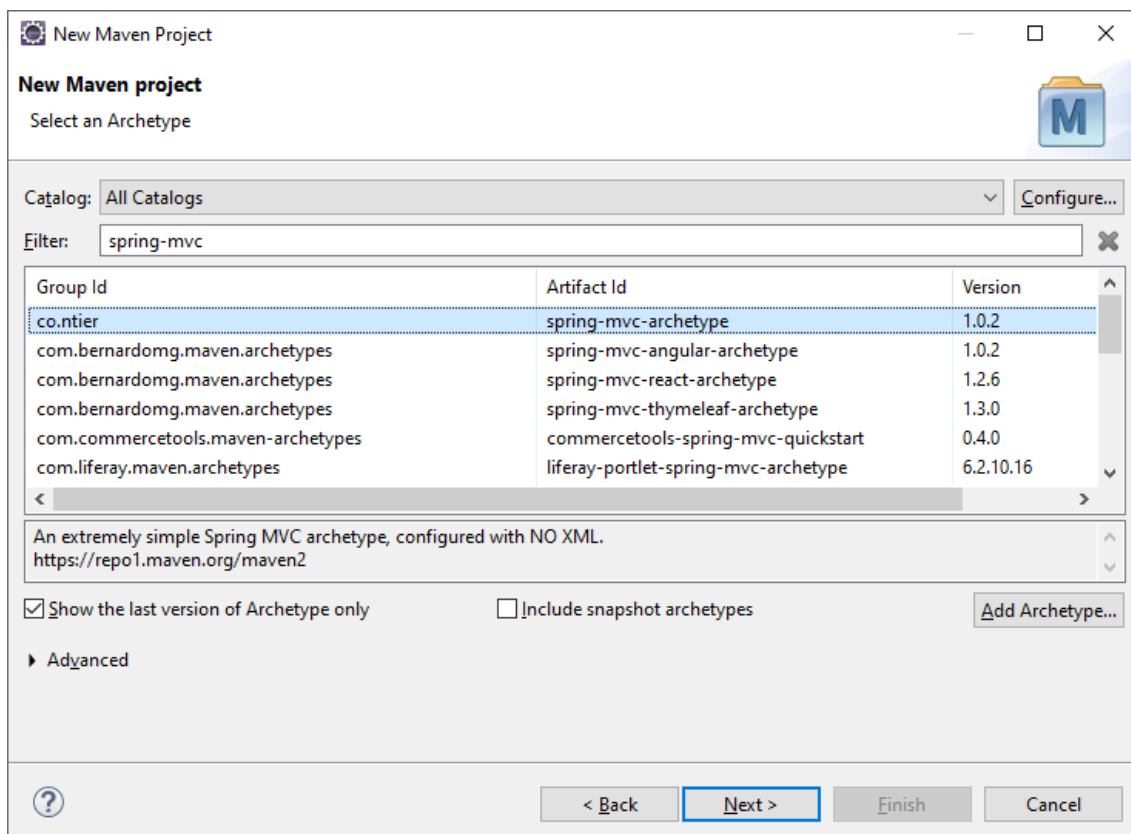
Advanced

Next > Finish Cancel

### Selecione um Archetype

Aqui, iremos escolher o tipo de projeto que o Maven irá criar.

Selecione o tipo: **spring-mvc-archetype**



New Maven Project

Select an Archetype

Catalog: All Catalogs Configure...

Filter: spring-mvc

Group Id	Artifact Id	Version
co.ntier	spring-mvc-archetype	1.0.2
com.bernardomg.maven.archetypes	spring-mvc-angular-archetype	1.0.2
com.bernardomg.maven.archetypes	spring-mvc-react-archetype	1.2.6
com.bernardomg.maven.archetypes	spring-mvc-thymeleaf-archetype	1.3.0
com.commercetools.maven.archetypes	commercetools-spring-mvc-quickstart	0.4.0
com.liferay.maven.archetypes	liferay-portlet-spring-mvc-archetype	6.2.10.16

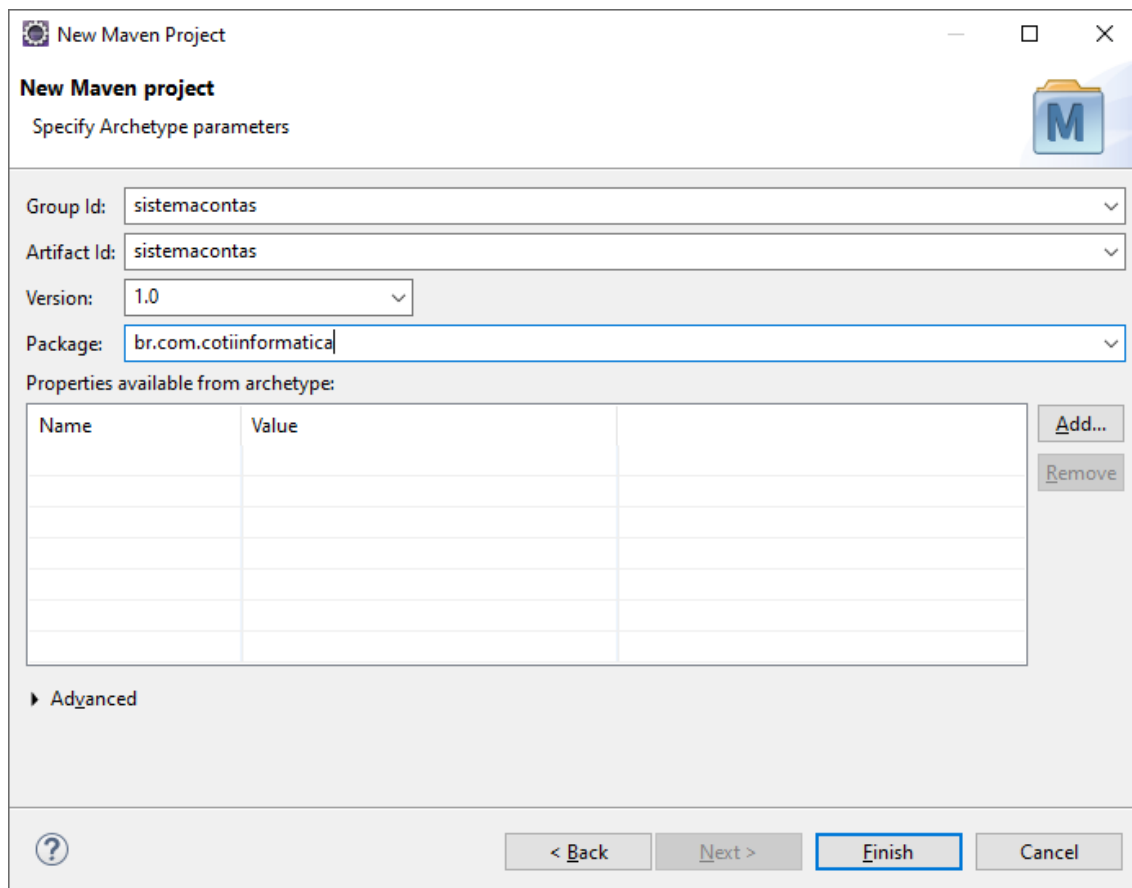
An extremely simple Spring MVC archetype, configured with NO XML.  
<https://repo1.maven.org/maven2>

☒ Show the last version of Archetype only ☐ Include snapshot archetypes Add Archetype...

Advanced

Next > Finish Cancel

Preenchendo as informações do projeto:



**New Maven Project**

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

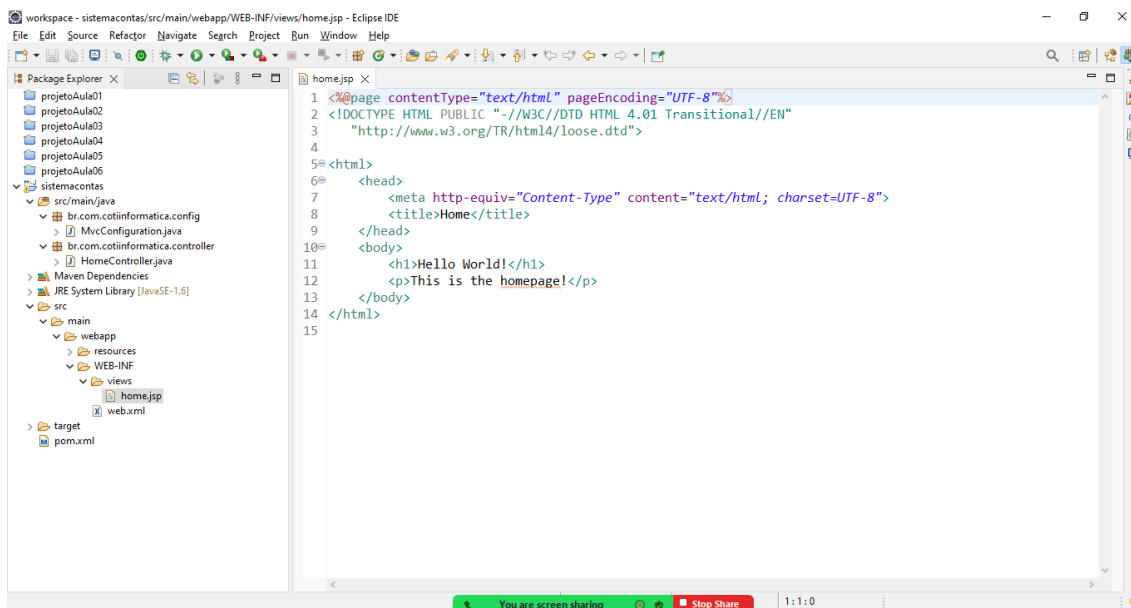
Properties available from archetype:

Name	Value

► Advanced

```

sistemacontas
├── src/main/java
│   ├── br.com.cotiinformatica.config
│   │   └── MvcConfiguration.java
│   ├── br.com.cotiinformatica.controller
│   │   └── HomeController.java
│   └── Maven Dependencies
├── JRE System Library [JavaSE-1.6]
├── src
│   └── main
│       ├── webapp
│       │   ├── resources
│       │   └── WEB-INF
│       │       └── views
│       │           ├── home.jsp
│       │           └── web.xml
│       └── target
│           └── pom.xml
    
```



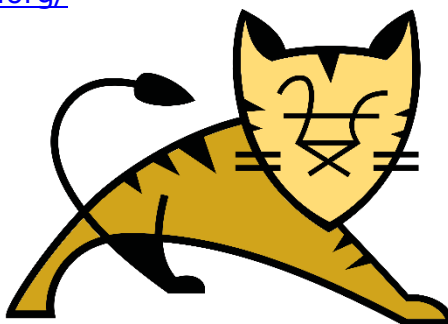
```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
3   "http://www.w3.org/TR/html4/loose.dtd">
4
5 <html>
6 <head>
7   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8   <title>Home</title>
9 </head>
10 <body>
11   <h1>Hello World!</h1>
12   <p>This is the homepage!</p>
13 </body>
14 </html>
15
```

Para que possamos executarmos o projeto, precisamos de um servidor local que possa "rodar" a aplicação. Este servidor será o **Apache Tomcat**.

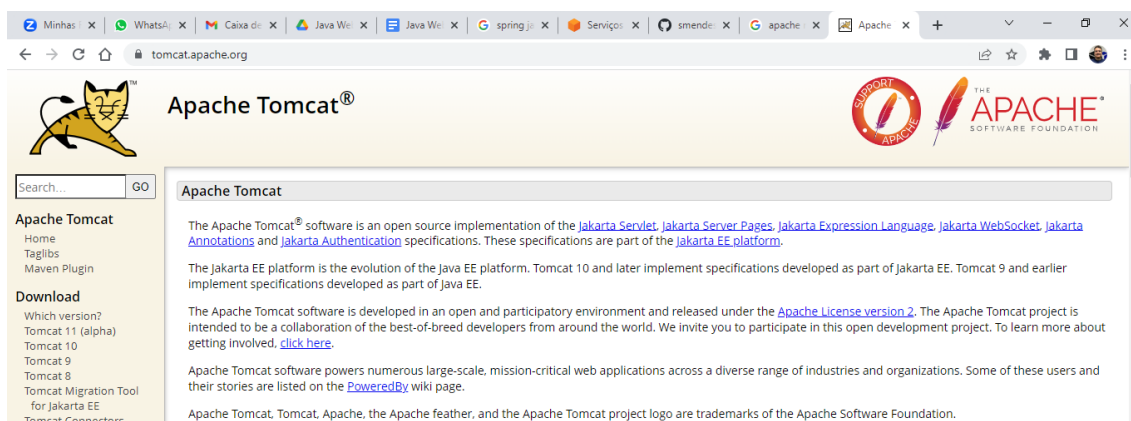
## Apache Tomcat

Servidor utilizado para executar projetos Java Web.

<https://tomcat.apache.org/>

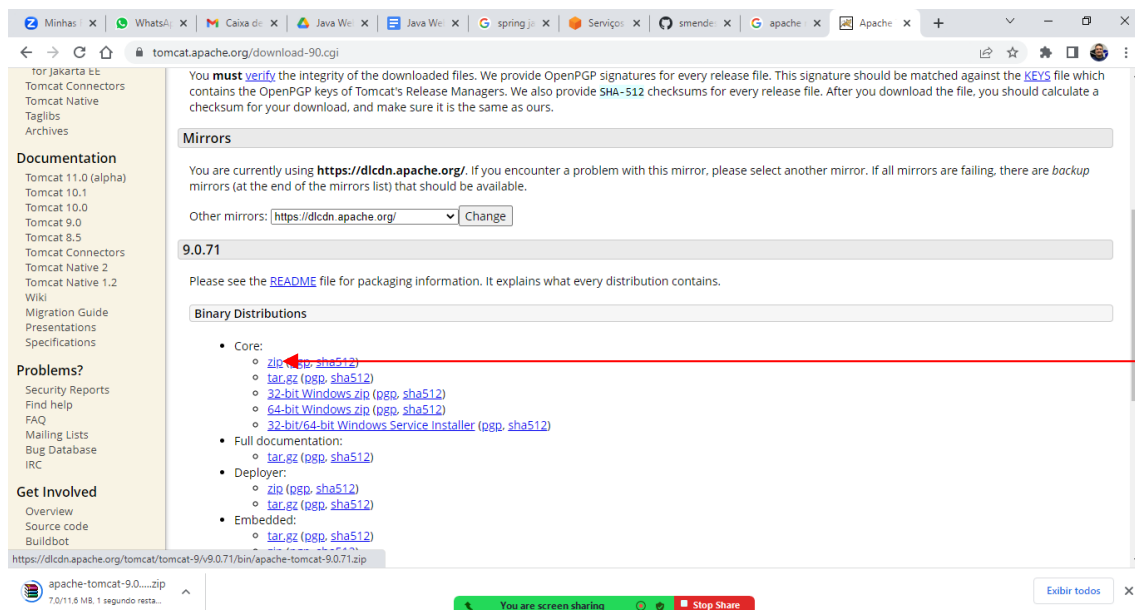


# Apache Tomcat

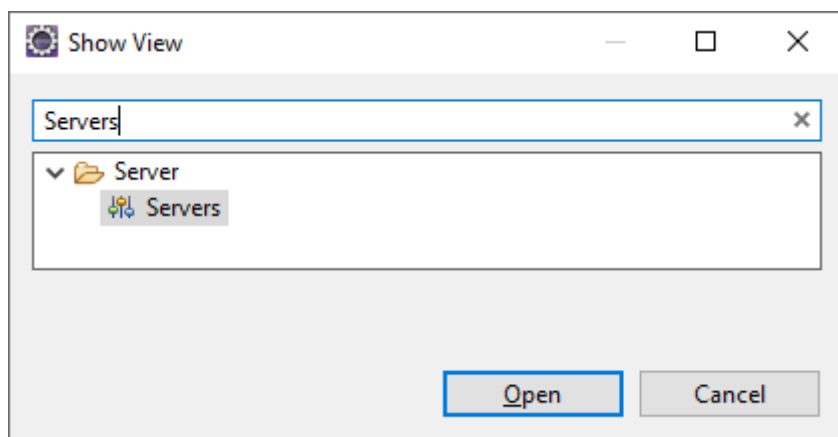
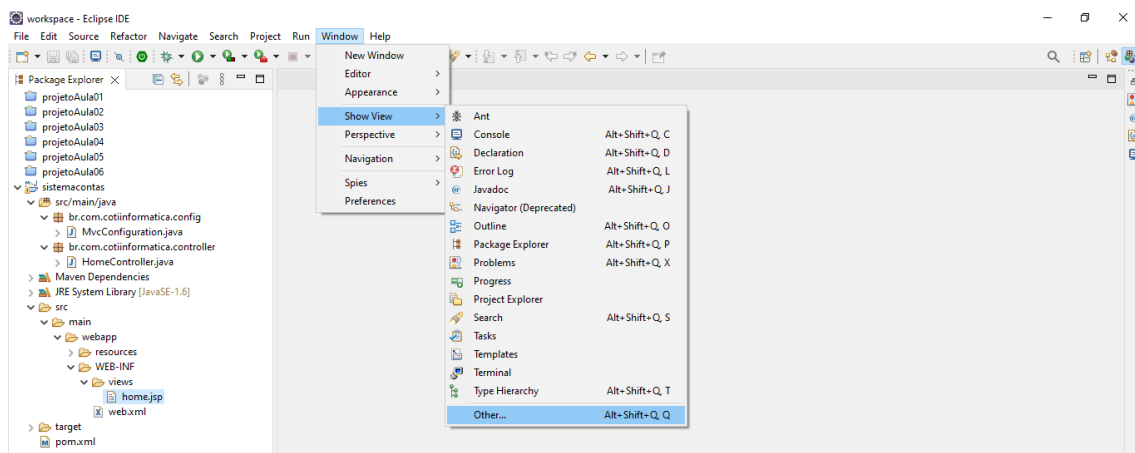




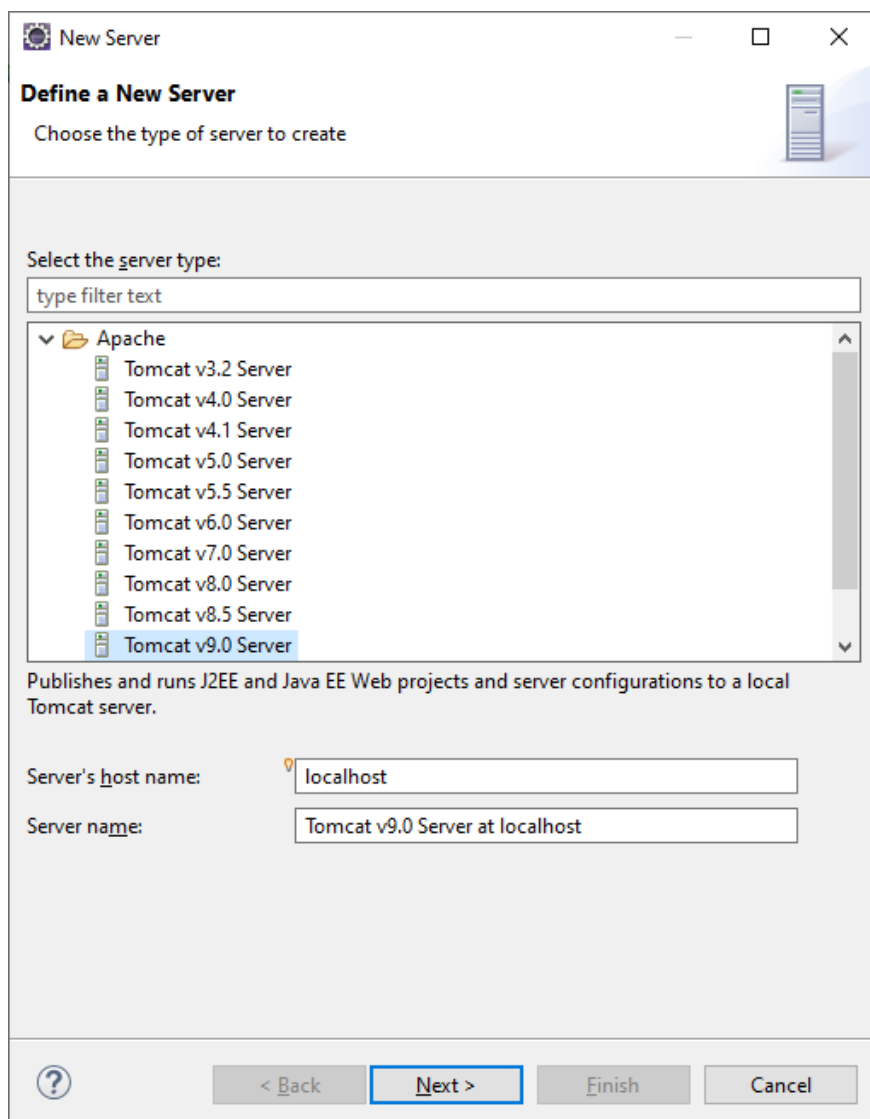
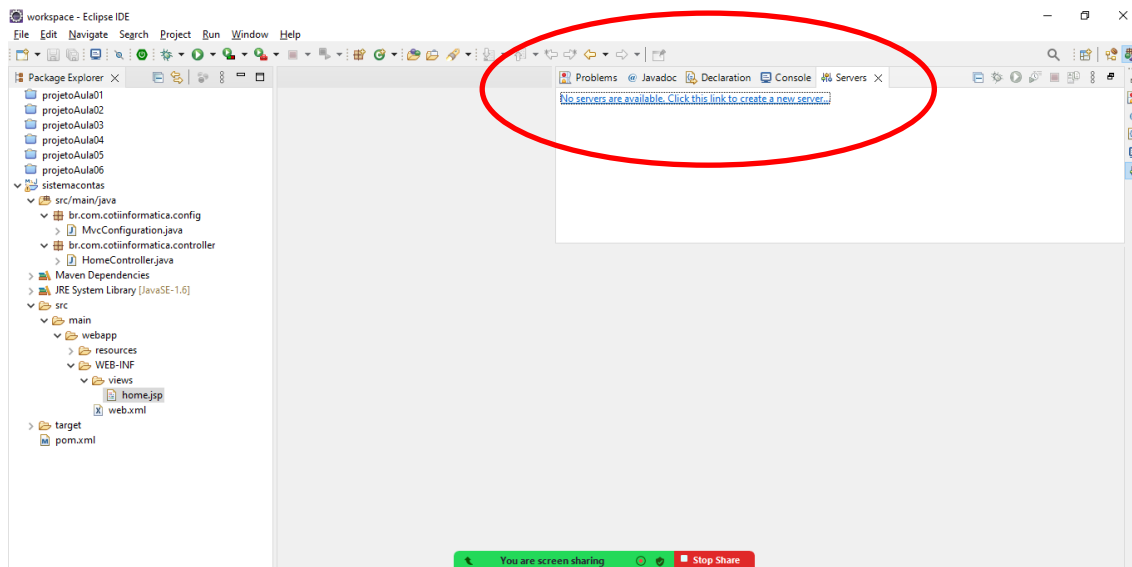
<https://tomcat.apache.org/download-90.cgi>



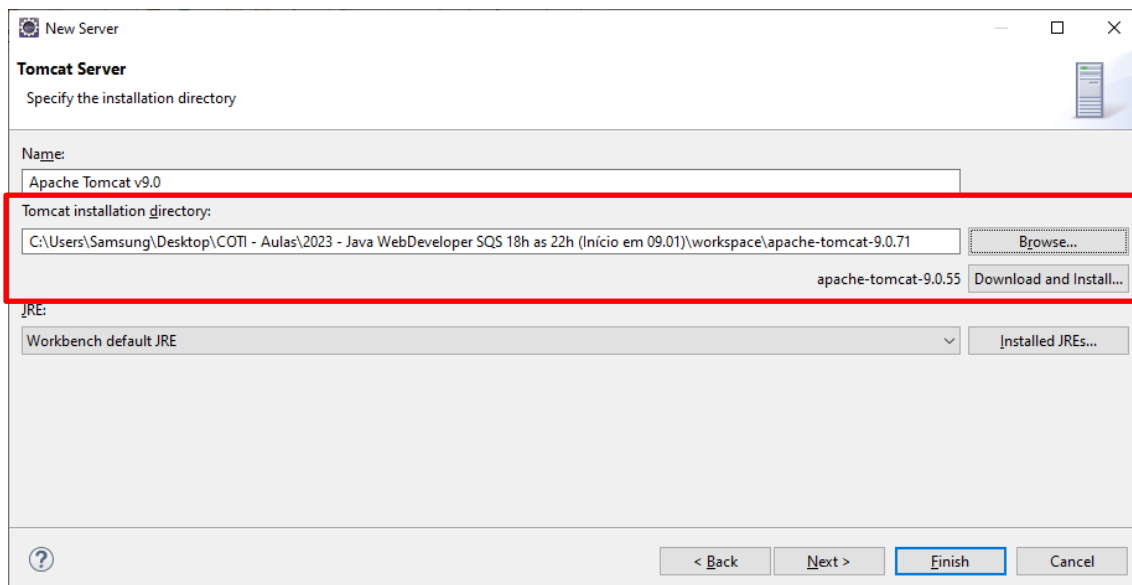
Exibindo a janela de configuração de servidores do eclipse:  
**Windows / Show View / Other**



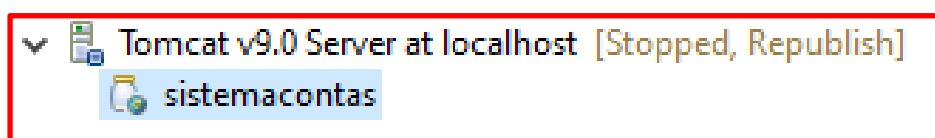
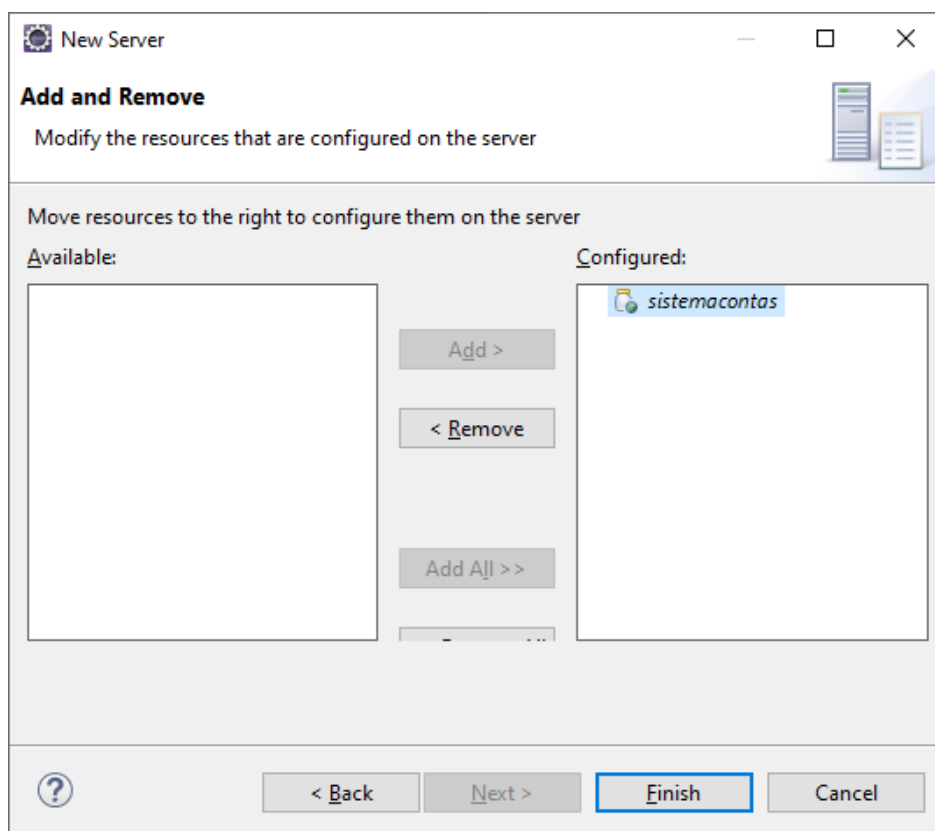
Configurando o servidor:



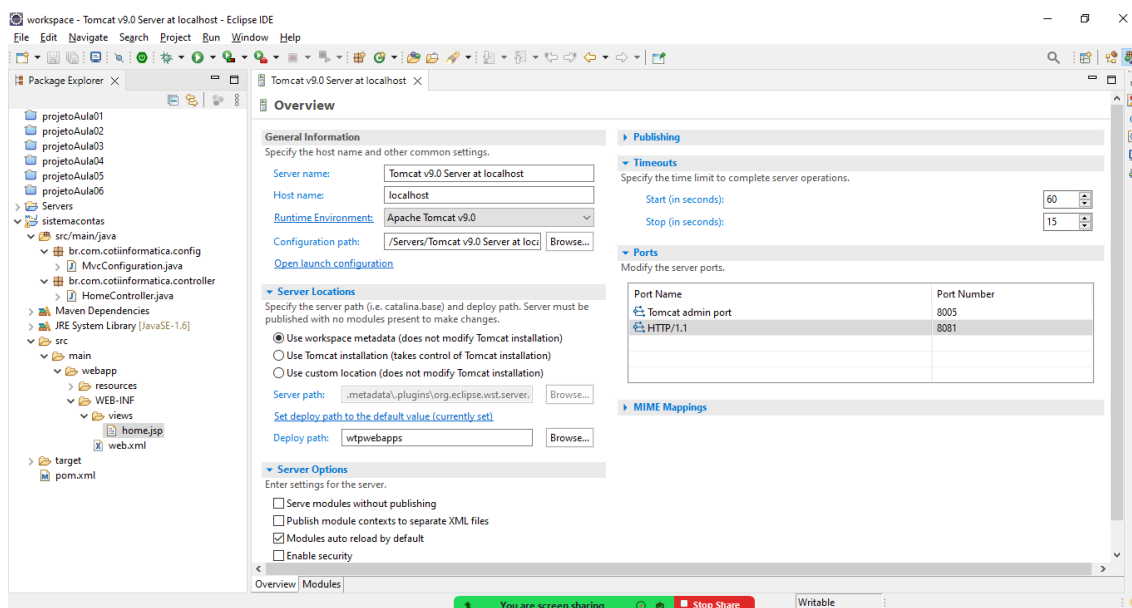
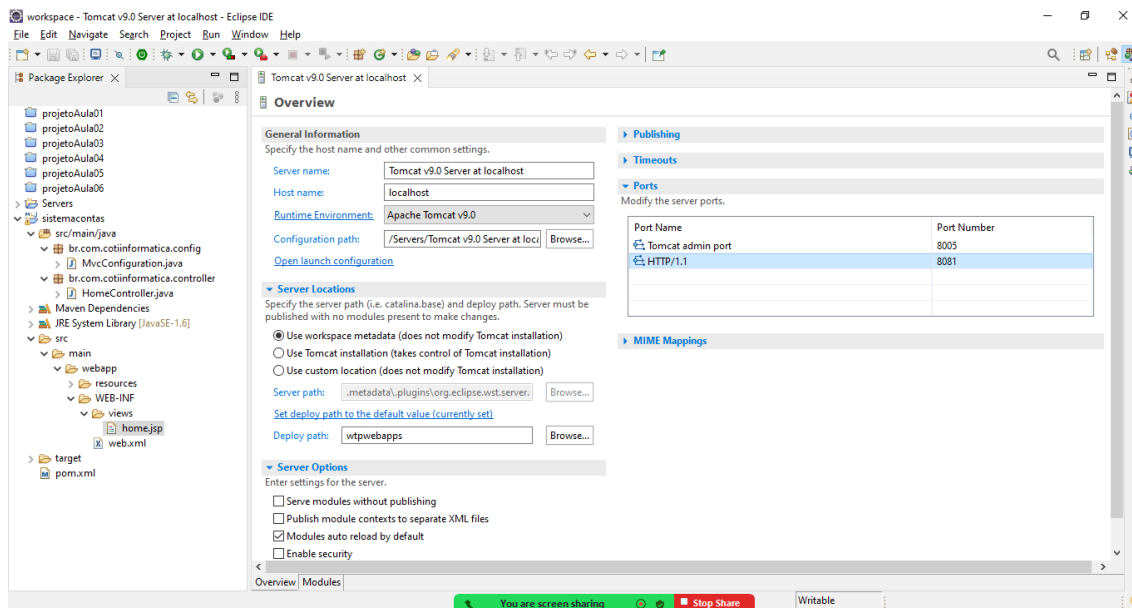
Selecione a pasta do servidor:



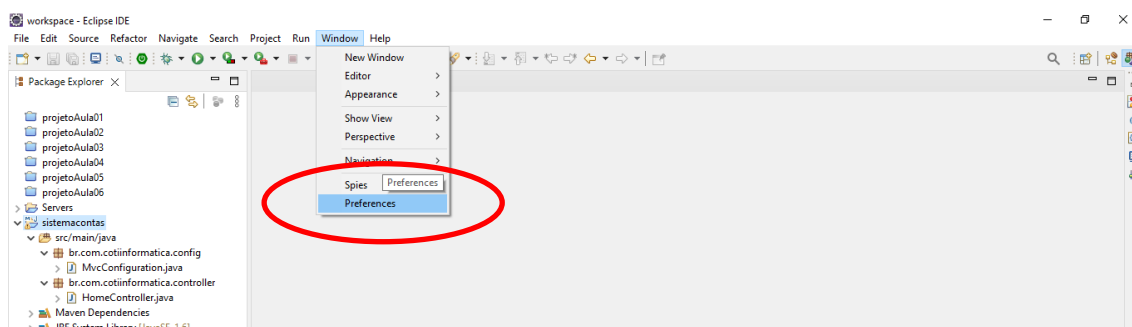
Adicionando o projeto no servidor:

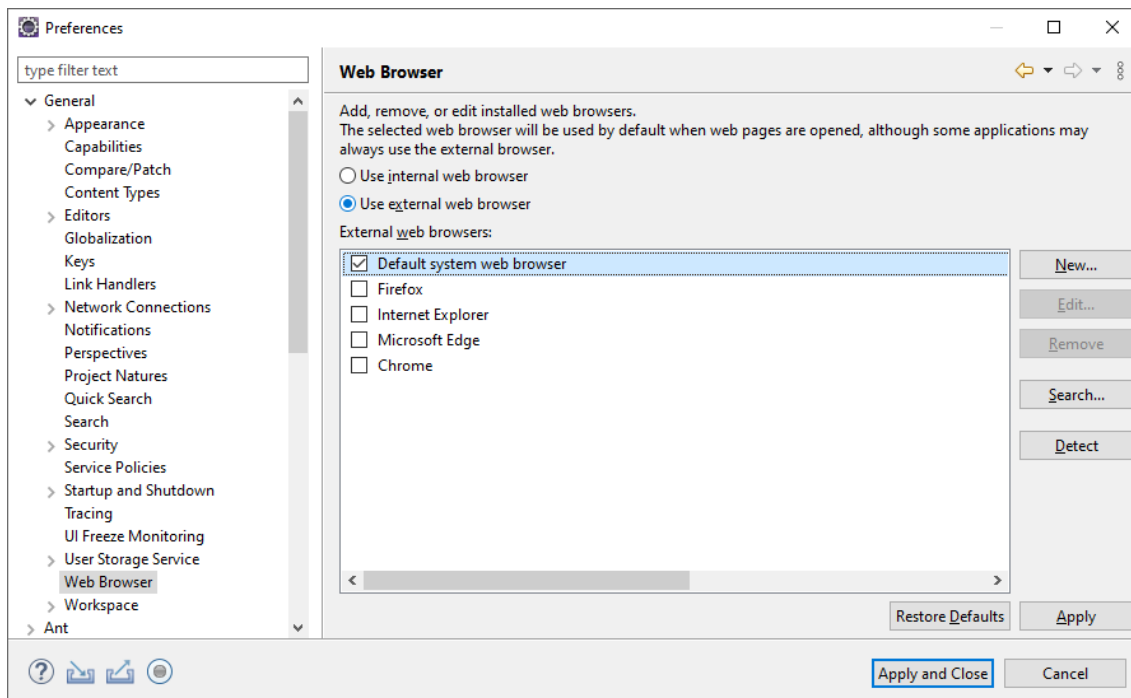


## Configurando a porta para execução do projeto:

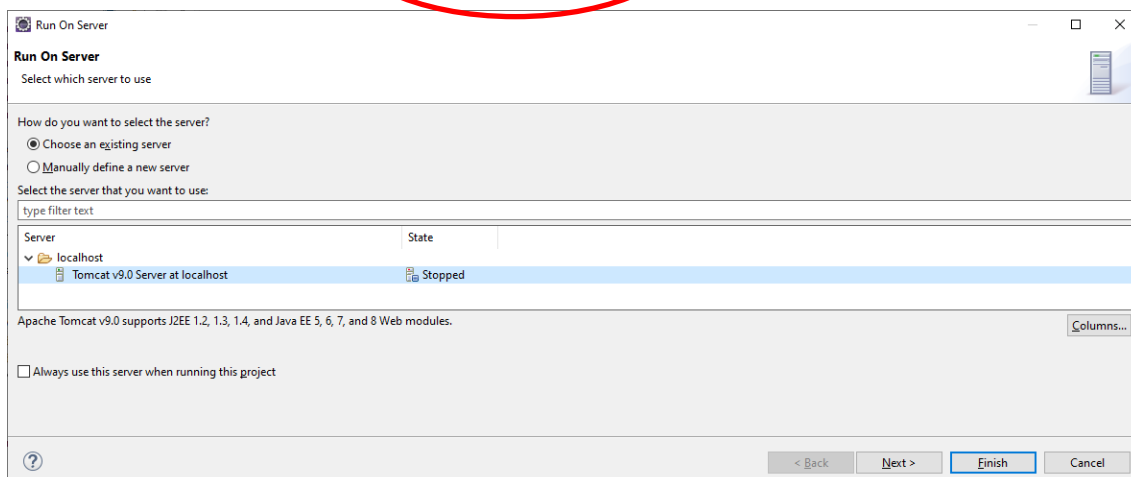
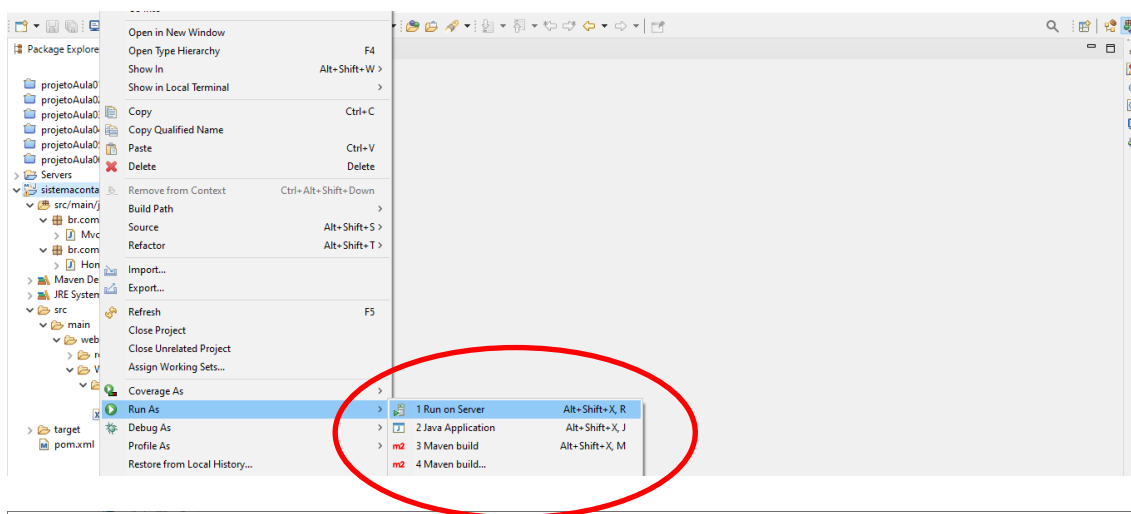


Antes de executar o projeto, vamos configurar em qual navegador ele será executado:

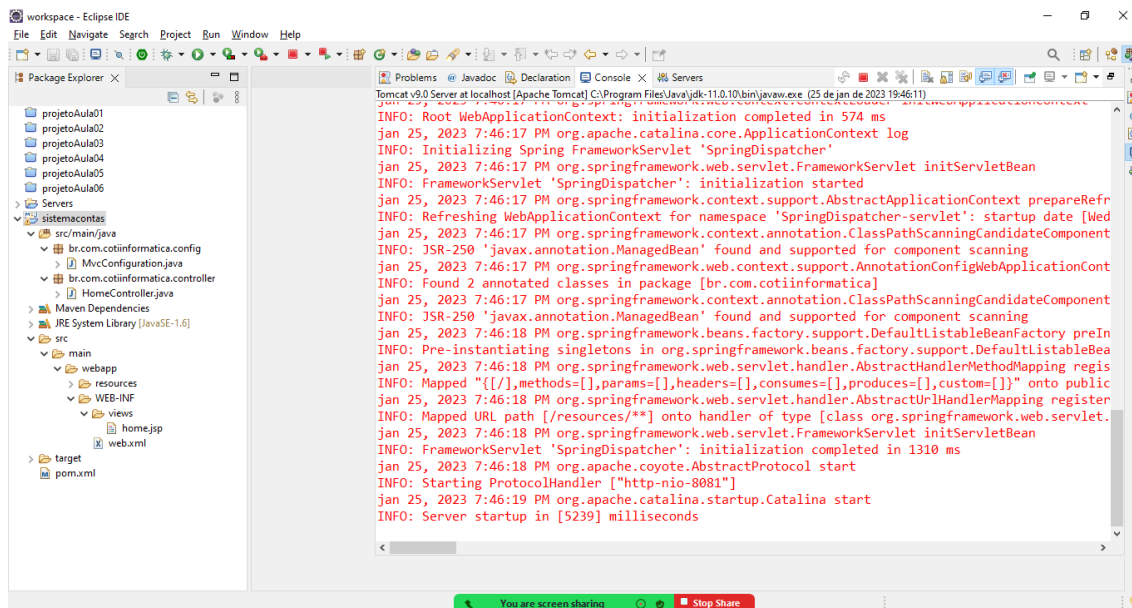




Executando o projeto:



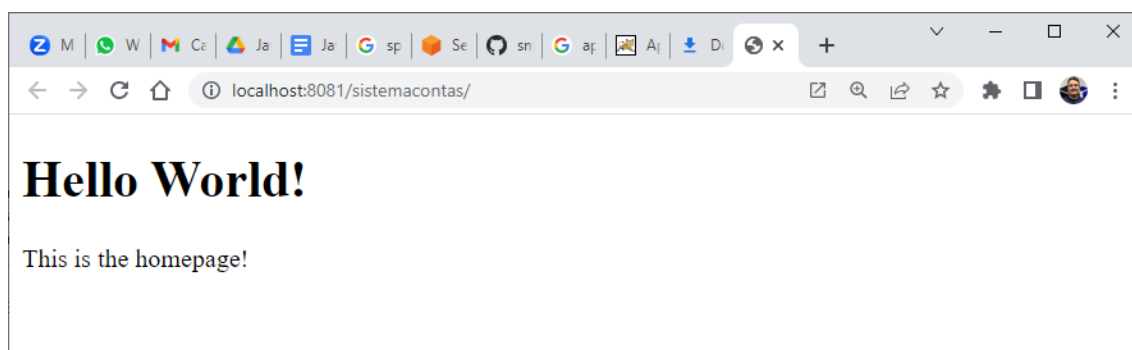
### Log de execução do tomcat:



```

Tomcat v9.0 Server at localhost [Apache Tomcat/9.0.10]
INFO: Root WebApplicationContext: initialization completed in 574 ms
jan 25, 2023 7:46:17 PM org.apache.catalina.core.ApplicationContext log
INFO: Initializing Spring FrameworkServlet 'SpringDispatcher'
jan 25, 2023 7:46:17 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: FrameworkServlet 'SpringDispatcher': initialization started
jan 25, 2023 7:46:17 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing WebApplicationContext for namespace 'SpringDispatcher-servlet': startup date [Wed
jan 25, 2023 7:46:17 PM org.springframework.context.annotation.ClassPathScanningCandidateComponent
INFO: JSR-250 'javax.annotation.ManagedBean' found and supported for component scanning
jan 25, 2023 7:46:17 PM org.springframework.web.context.support.AnnotationConfigWebApplicationContext
INFO: Found 2 annotated classes in package [br.com.cotiinformatica]
jan 25, 2023 7:46:17 PM org.springframework.context.annotation.ClassPathScanningCandidateComponent
INFO: JSR-250 'javax.annotation.ManagedBean' found and supported for component scanning
jan 25, 2023 7:46:17 PM org.springframework.beans.factory.support.DefaultListableBeanFactory preIn
INFO: Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBea
jan 25, 2023 7:46:17 PM org.springframework.web.servlet.handler.AbstractHandlerMethodMapping regis
INFO: Mapped "([/],methods=[],params=[],headers=[],consumes=[],produces=[],custom=[])" onto public
jan 25, 2023 7:46:17 PM org.springframework.web.servlet.handler.AbstractHandlerMethodMapping regis
INFO: Mapped URL path [/resources/*] onto handler of type [class org.springframework.web.servlet.
jan 25, 2023 7:46:17 PM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: FrameworkServlet 'SpringDispatcher': initialization completed in 1310 ms
jan 25, 2023 7:46:17 PM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8081"]
jan 25, 2023 7:46:17 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in [5239] milliseconds
  
```

<http://localhost:8081/sistemacontas/>



### /WEB-INF/views/**home.jsp**

Página inicial do projeto (Está configurada como página de inicialização do projeto). As páginas web do projeto Java possuem extensão **JSP – Java Server Pages**.

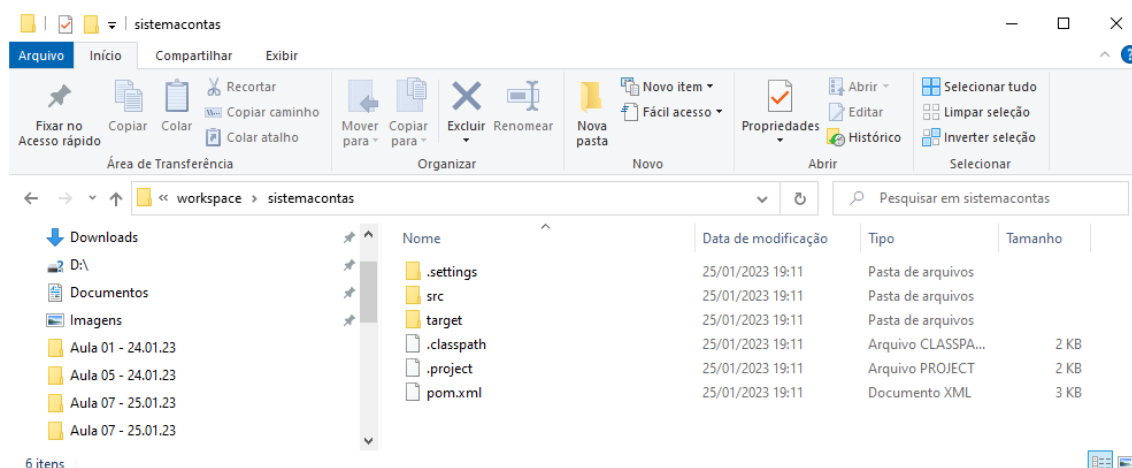
```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Home</title>
</head>
<body>
    <h1>Hello World!</h1>
    <p>This is the homepage!</p>
</body>
</html>
  
```

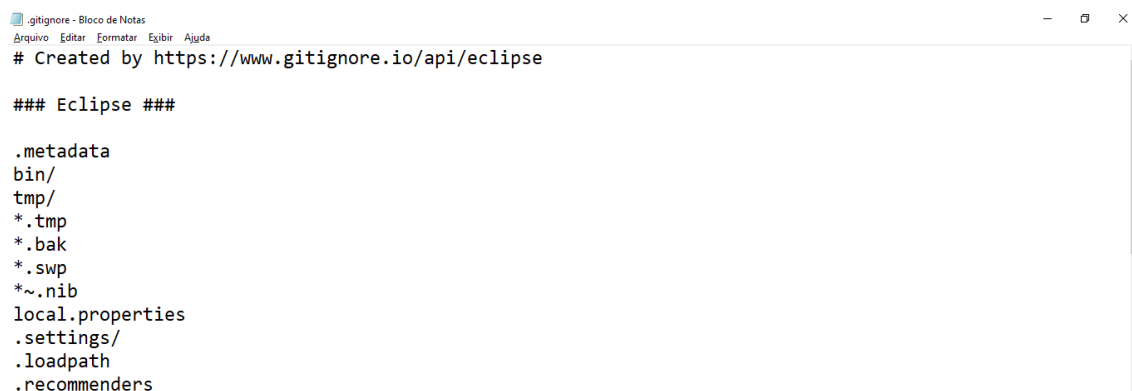
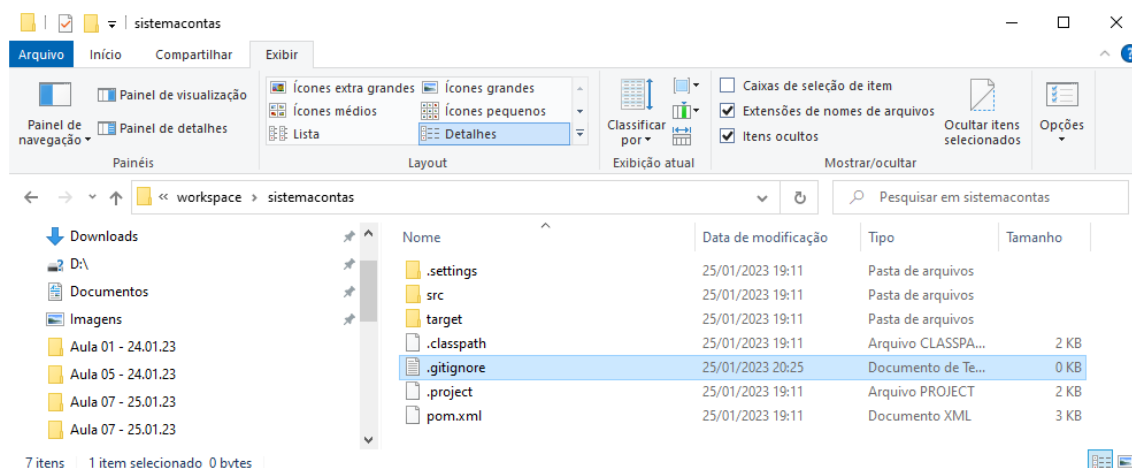
## Podemos agora já publicar o código fonte do projeto no GITHUB:

\*\* Para que possamos publicar o projeto no GITHUB, precisamos criar alguns arquivos de configuração no projeto:



## .gitignore

Arquivo de configuração onde vamos definir quais pastas e arquivos do projeto não serão enviados para o GITHUB.



```
# Created by https://www.gitignore.io/api/eclipse

### Eclipse ###

.metadata
bin/
tmp/
*.tmp
*.bak
*.swp
*~.nib
local.properties
.settings/
.loadpath
.recommenders

# External tool builders
.externalToolBuilders/

# Locally stored "Eclipse launch configurations"
*.launch

# PyDev specific (Python IDE for Eclipse)
*.pydevproject

# CDT-specific (C/C++ Development Tooling)
.cproject

# Java annotation processor (APT)
.factorypath

# PDT-specific (PHP Development Tools)
.buildpath

# sbteclipse plugin
.target

# Tern plugin
.tern-project

# TeXlipse plugin
.texlipse

# STS (Spring Tool Suite)
.springBeans

# Code Recommenders
.recommenders/

# Scala IDE specific (Scala & Java development for Eclipse)
.cache-main
.scala_dependencies
.worksheet

### Eclipse Patch ###
# Eclipse Core
.project

# JDT-specific (Eclipse Java Development Tools)
.classpath

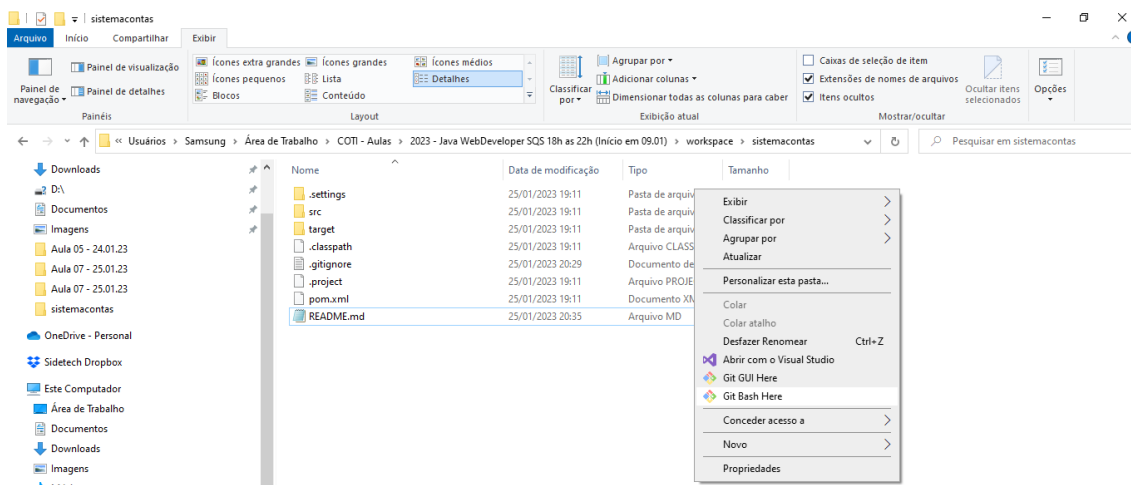
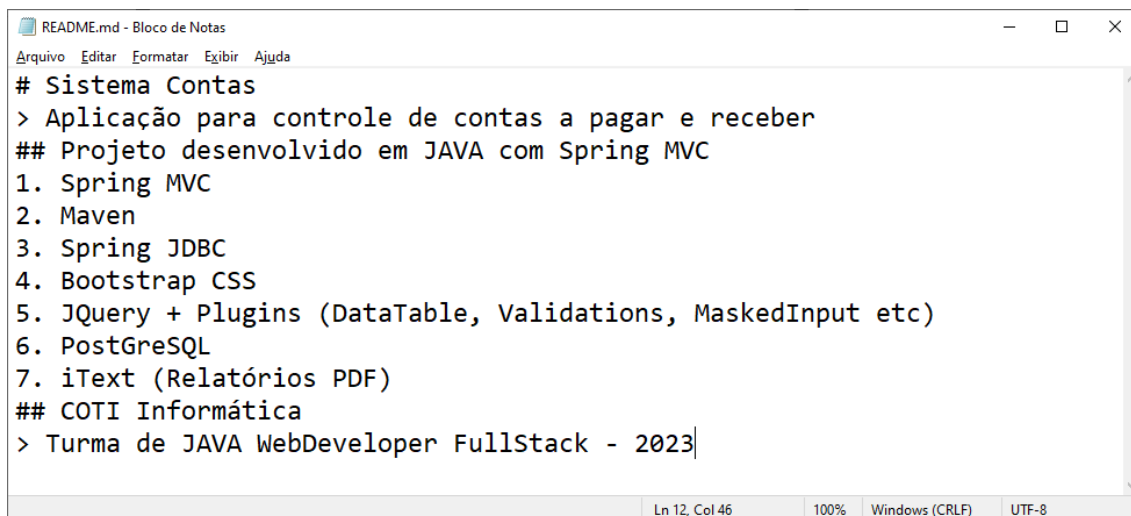
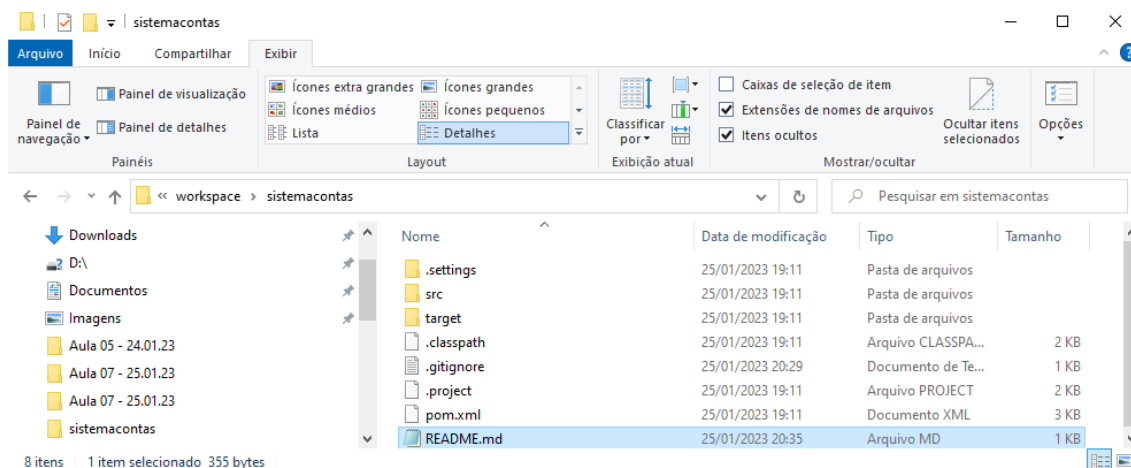
# End of https://www.gitignore.io/api/eclipse
```



## README.md

(Arquivo de instruções do projeto)

Arquivo utilizado pelo GITHUB para criarmos uma página de apresentação do projeto, falando sobre o sistema e explicando o que é necessário para executar a aplicação.

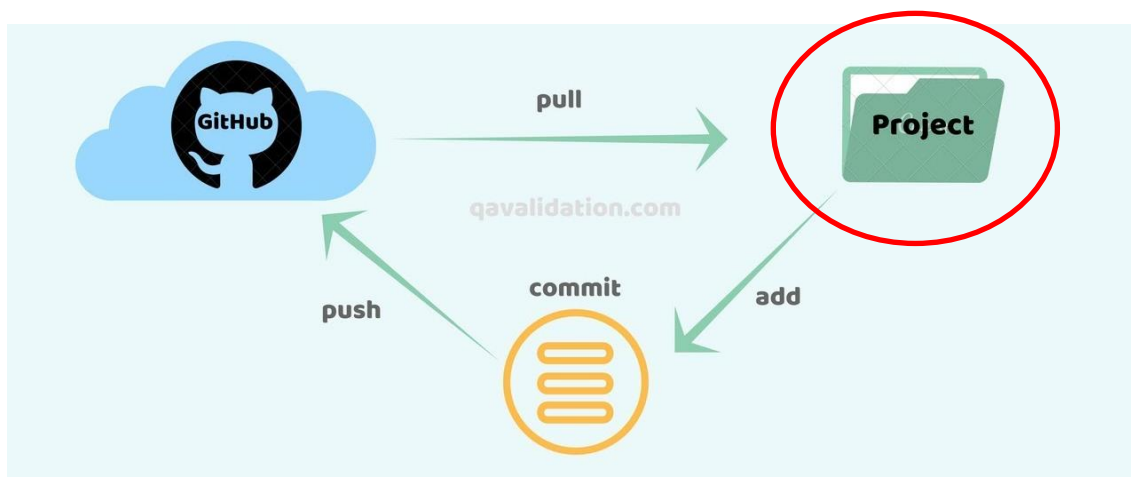


## Registrando o usuário do GITHUB:

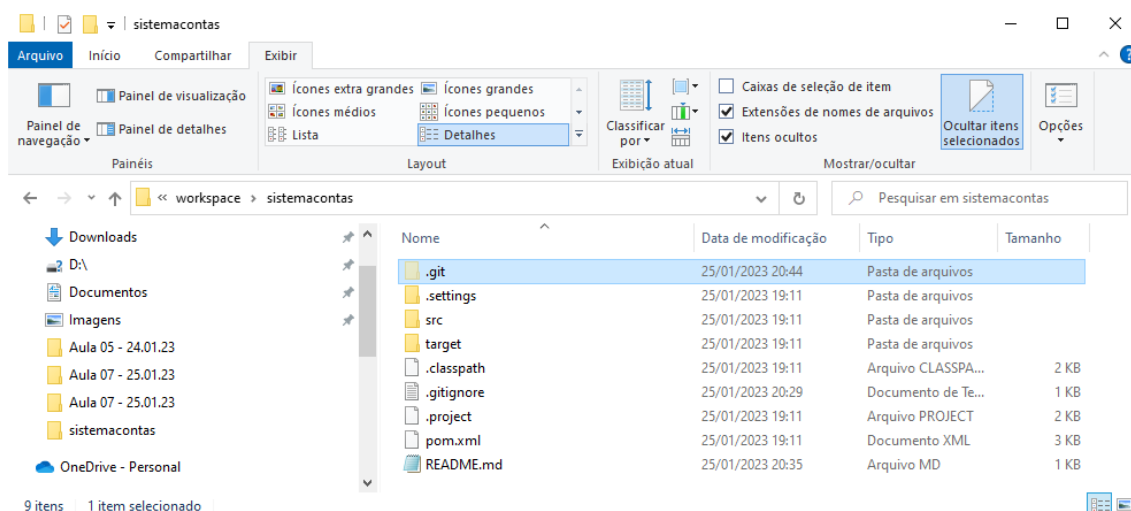
```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/sistemacontas
$ git config --global user.name 'smendescoti'
```

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/sistemacontas
$ git config --global user.email 'sergio.coti@gmail.com'
```

Em seguida, vamos iniciar um repositório do GIT na pasta do projeto:



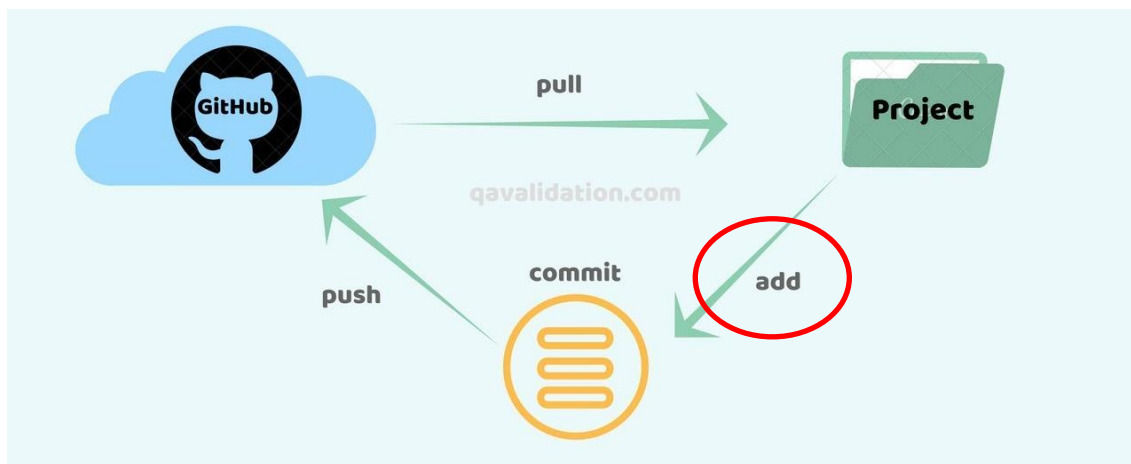
```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/sistemacontas
$ git init
```



## COMMIT

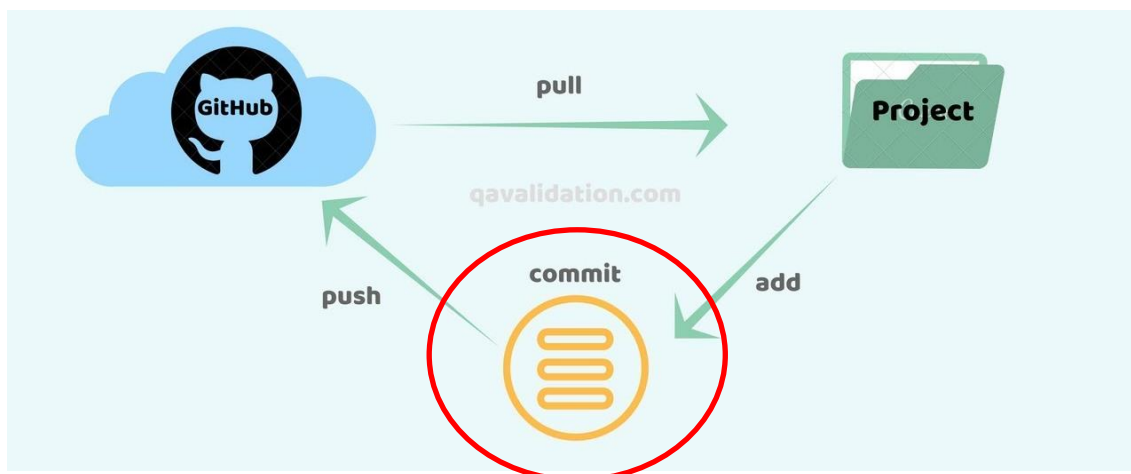
Nome dado ao processo de enviar arquivos do projeto local para o repositório do GITHUB. Em um commit podemos enviar desde muitos arquivos e pastas até somente alterações pequenas feitas no projeto.

- Primeiro, vamos adicionar todos os arquivos do projeto no COMMIT.



```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/sistemacontas (master)
$ git add .
```

- Em seguida, vamos dar um nome para o COMMIT.



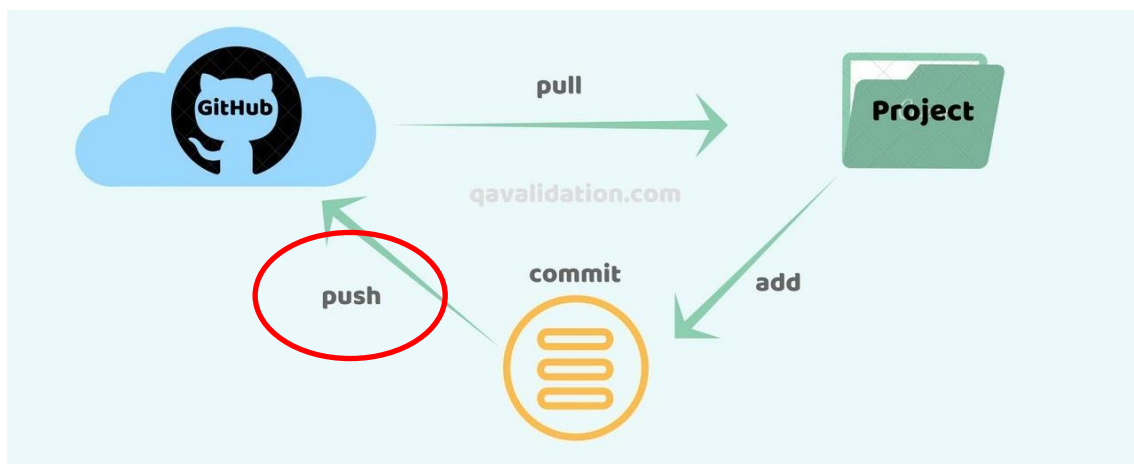
```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 -
Java WebDeveloper SQS 18h as 22h (Início em
09.01)/workspace/sistemacontas (master)
$ git commit -m 'first commit'
```

```

MINGW64/C:/Users/Samsung/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas
$ git init
Initialized empty Git repository in C:/Users/Samsung/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas/.git/
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (master)
$ git add .
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (master)
$ git commit -m 'first commit'
[master (root-commit) 8d4f71c] first commit
13 files changed, 376 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 pom.xml
create mode 100644 src/main/java/br/com/cotiinformatica/config/MvcConfiguration.java
create mode 100644 src/main/java/br/com/cotiinformatica/controller/HomeController.java
create mode 100644 src/main/webapp/WEB-INF/views/home.jsp
create mode 100644 src/main/webapp/WEB-INF/web.xml
create mode 100644 src/main/webapp/resources/style.css
create mode 100644 target/classes/br/com/cotiinformatica/config/MvcConfiguration.class
create mode 100644 target/classes/br/com/cotiinformatica/controller/HomeController.class
create mode 100644 target/m2e-wtp/web-resources/META-INF/MANIFEST.MF
create mode 100644 target/m2e-wtp/web-resources/META-INF/maven/sistemacontas/sistemacontas/pom.properties
create mode 100644 target/m2e-wtp/web-resources/META-INF/maven/sistemacontas/sistemacontas/pom.xml
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (master)
$
  
```

## PUSH

Comando para fazer o upload do conteúdo de um COMMIT para o repositório do GITHUB:



```

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (master)
$ git branch -M main
  
```

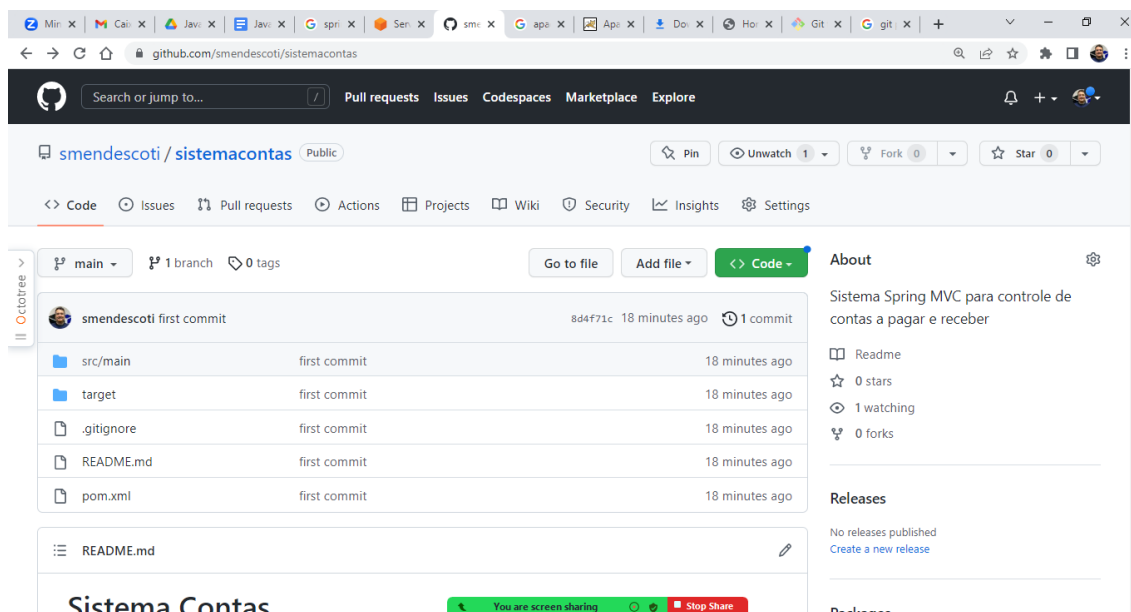
```

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git remote add origin
https://github.com/smendescoti/sistemacontas.git
  
```

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)  
\$ git push -u origin main

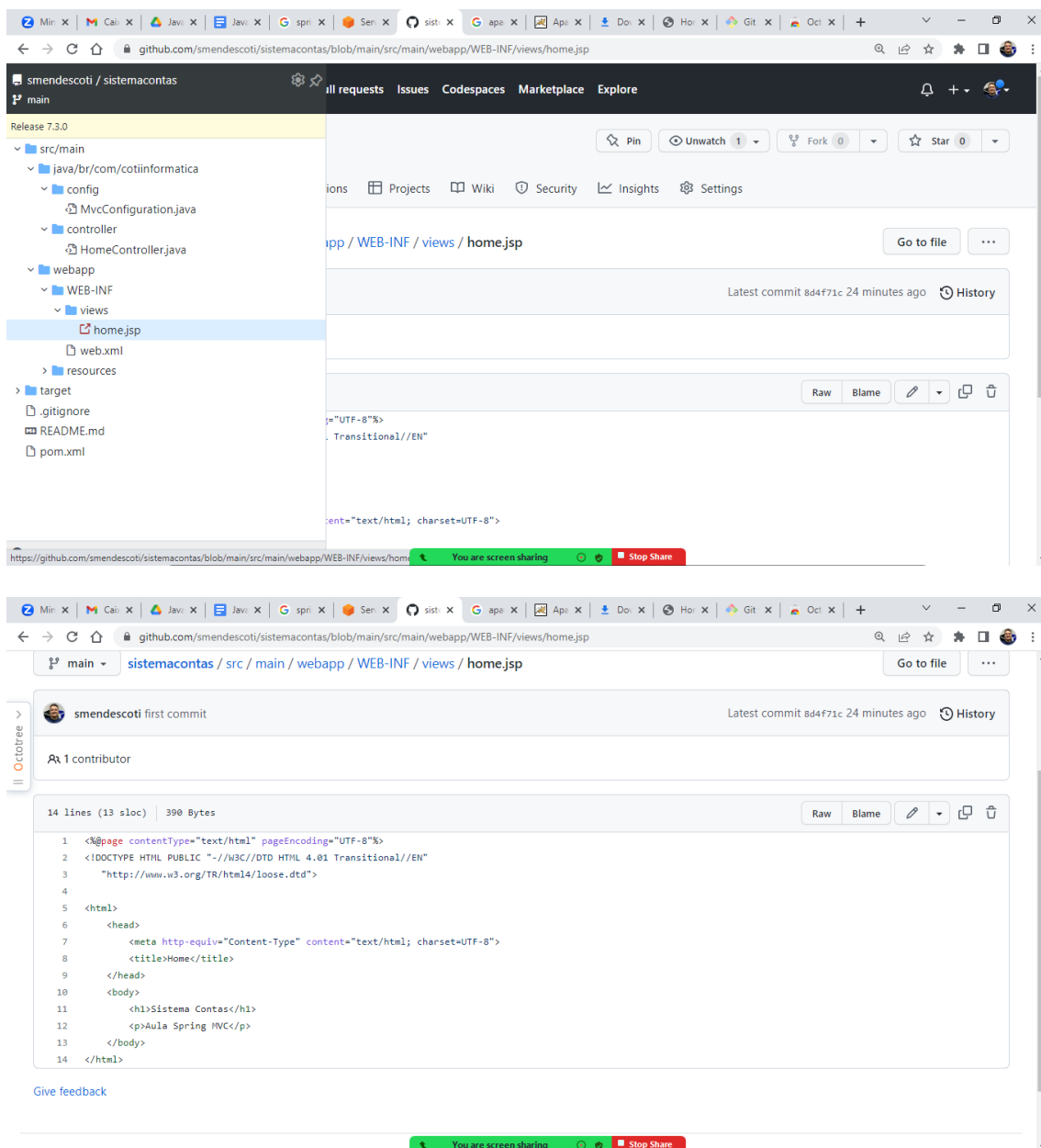
```
MINGW64~/Users/Samsung/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (master)
$ git branch -M main
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git remote add origin https://github.com/smendescoti/sistemacontas.git
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git push -u origin main
Enumerating objects: 39, done.
Counting objects: 100% (39/39), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (39/39), 6.10 KiB | 568.00 KiB/s, done.
Total 39 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/smendescoti/sistemacontas.git
 * [new branch] main -> main
branch 'main' set up to track 'origin/main'.
```

<https://github.com/smendescoti/sistemacontas>



<https://chrome.google.com/webstore/detail/octotree-github-code-tree/bkhaagjahfmjljalopjnoealnfndnagc>





The screenshot shows a GitHub repository for 'smendescom/sistemacontas'. The file browser on the left shows the path: src/main/java/br/com/cotinformatica/webapp/WEB-INF/views/home.jsp. The main area displays the content of home.jsp, which is an HTML file. The code is as follows:

```
<?page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Home</title>
</head>
<body>
<h1>Sistema Contas</h1>
<p>Aula Spring MVC</p>
</body>
</html>
```

## Voltando ao projeto:

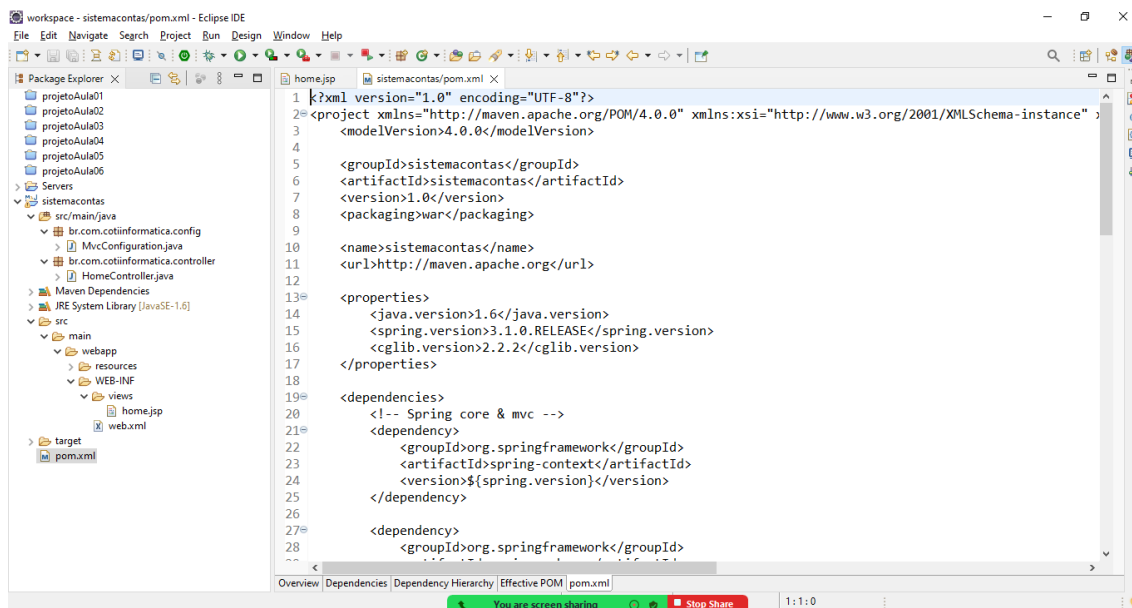
Instalando bibliotecas que precisamos para desenvolver o projeto:

- Driver JDBC para conexão com o PostgreSQL
- Lombok

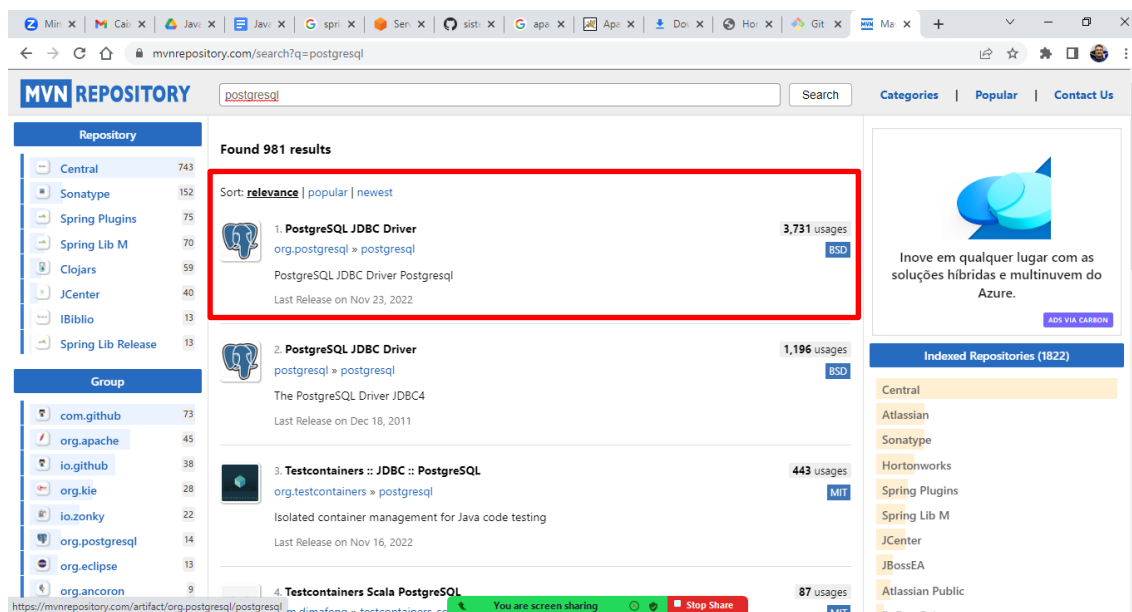
## /pom.xml

Principal arquivo de configuração do framework MAVEN. É utilizado para que possamos configurar quais bibliotecas serão instaladas no nosso projeto.

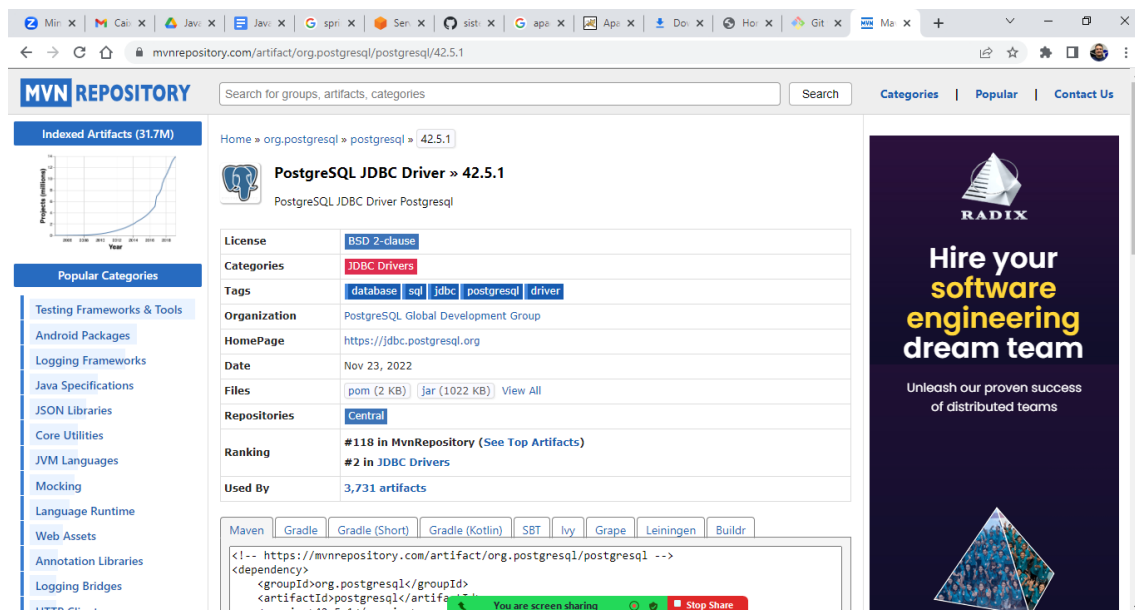




## Adicionando a biblioteca do POSTGRESQL:

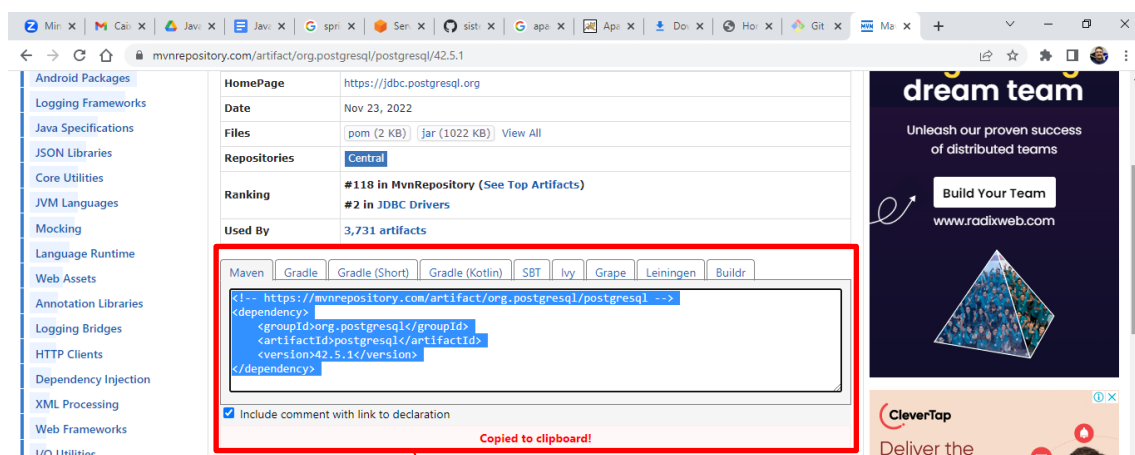


<https://mvnrepository.com/artifact/org.postgresql/postgresql/42.5.1>



The screenshot shows the Maven Repository page for the PostgreSQL JDBC Driver version 42.5.1. The page includes a search bar, a sidebar with popular categories, and a main content area with details about the artifact. The details section shows the license (BSD 2-clause), categories (JDBC Drivers), tags (database, sql, jdbc, postgresql, driver), organization (PostgreSQL Global Development Group), homepage (https://jdbc.postgresql.org), date (Nov 23, 2022), files (pom (2 KB), jar (1022 KB)), repositories (Central), ranking (#118 in Maven Repository, #2 in JDBC Drivers), and used by (3,731 artifacts). A code snippet for the dependency is also shown.

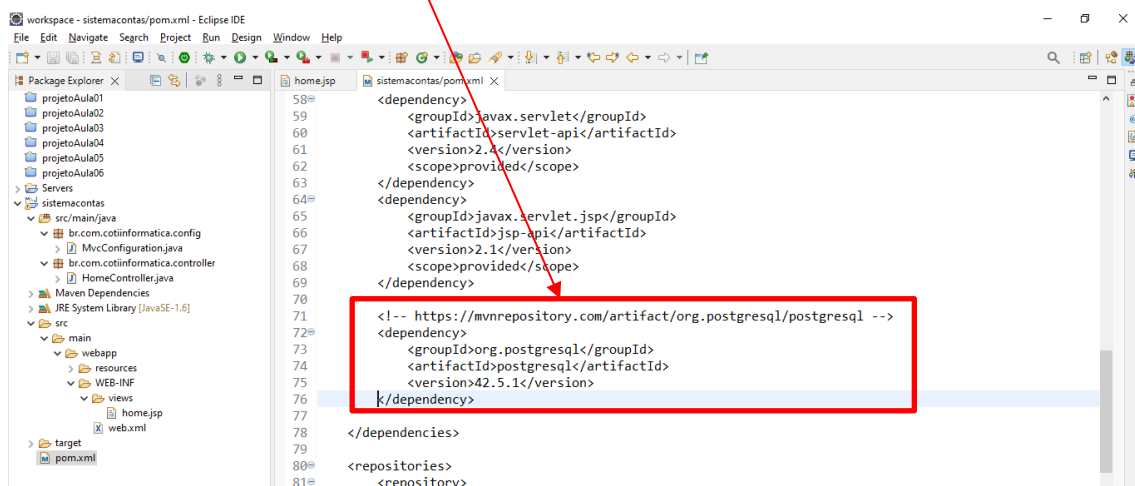
Copie o XML para incluir a referência da biblioteca no /pom.xml



This screenshot shows the same Maven Repository page, but with the XML dependency code highlighted in a red box. The code is as follows:

```
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.5.1</version>
</dependency>
```

A red arrow points from this code to the next screenshot, indicating that it should be copied to the clipboard.

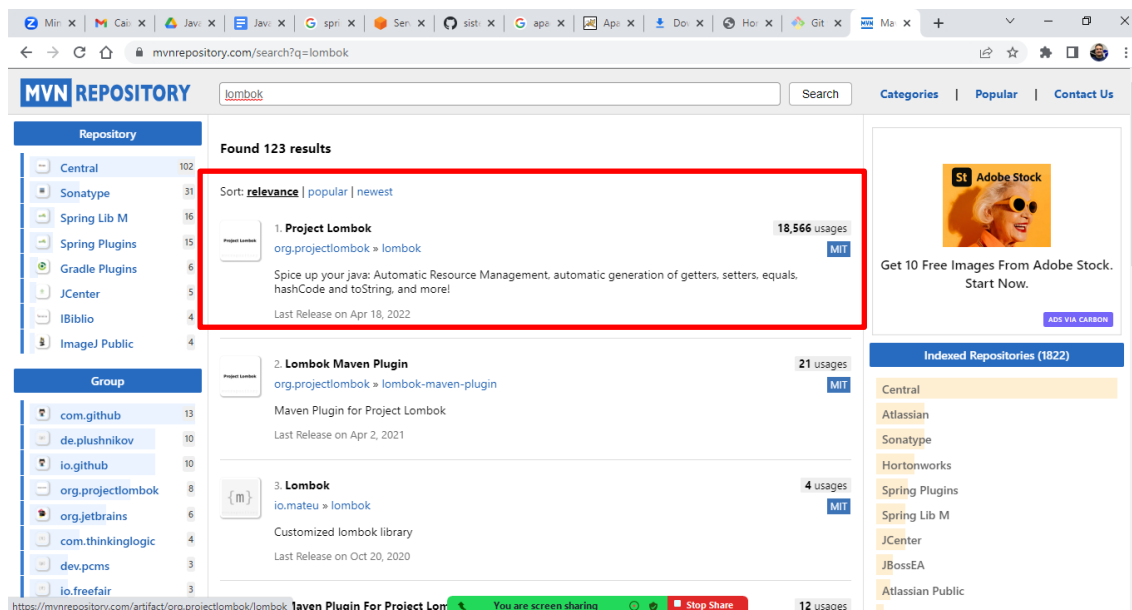


This screenshot shows the Eclipse IDE with the 'pom.xml' file open. The copied XML dependency code is pasted into the file, and a red box highlights the new entry. The code is as follows:

```
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.5.1</version>
</dependency>
```

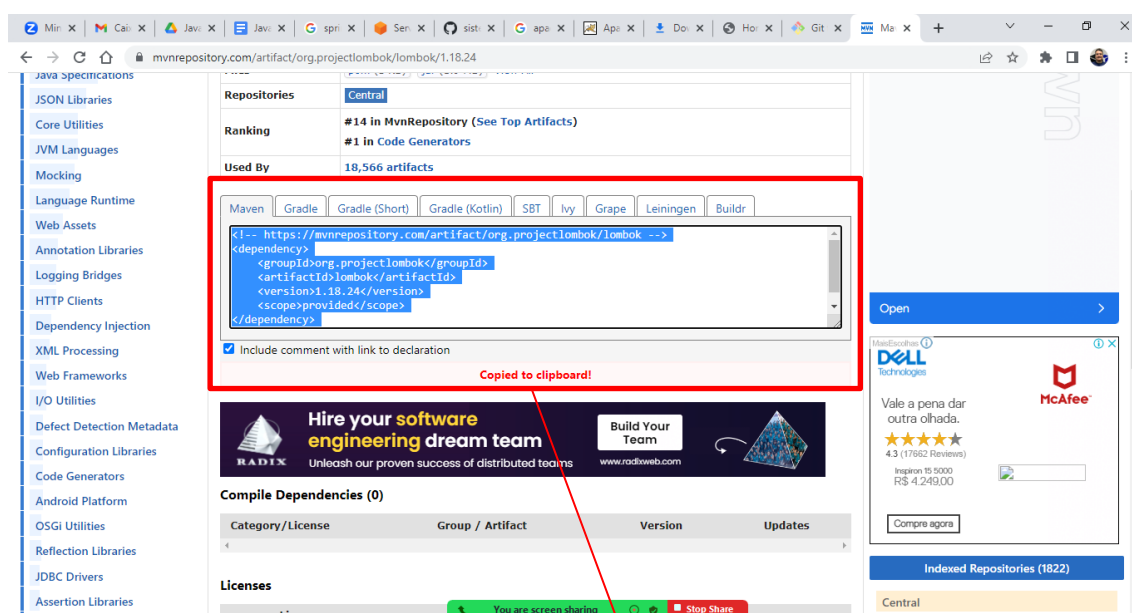


### Adicionando a biblioteca do LOMBOK:



The screenshot shows the Maven Repository search results for 'lombok'. The search bar contains 'lombok'. The results are sorted by relevance. The first result is 'Project Lombok' with 18,566 usages. The second result is 'Lombok Maven Plugin' with 21 usages. The third result is 'Lombok' with 4 usages. The left sidebar shows the repository list with 'Central' at the top. The right sidebar shows 'Indexed Repositories (1822)'.

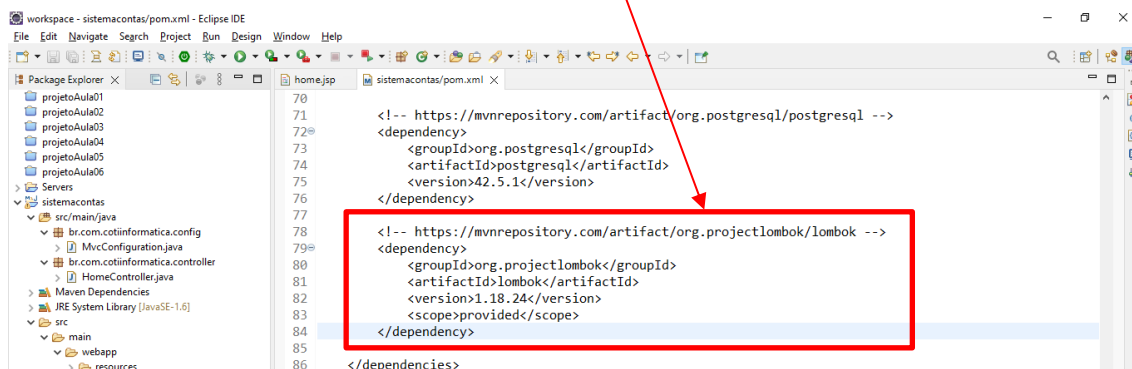
<https://mvnrepository.com/artifact/org.projectlombok/lombok/1.18.24>



The screenshot shows the Maven Repository artifact page for 'lombok/1.18.24'. The page displays the artifact's details, including its ranking (#14 in MvnRepository) and usage (18,566 artifacts). The 'Used By' section shows the artifact is used by various build tools like Maven, Gradle, and SBT. The 'Compile Dependencies' section is empty. The 'Licenses' section shows the artifact is licensed under the MIT license. A red box highlights the 'Maven' tab, which contains the following XML snippet:

```
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.24</version>
  <scope>provided</scope>
</dependency>
```

A red arrow points from this snippet to the 'pom.xml' file in the Eclipse IDE screenshot below.



The screenshot shows the Eclipse IDE with the 'pom.xml' file open. The file contains the following XML snippet, which is highlighted by a red box:

```
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.5.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.24</version>
  <scope>provided</scope>
</dependency>
```

A red arrow points from the 'pom.xml' file in the Eclipse IDE to the 'pom.xml' file in the Maven Repository screenshot above.

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>sistemacontas</groupId>
  <artifactId>sistemacontas</artifactId>
  <version>1.0</version>
  <packaging>war</packaging>

  <name>sistemacontas</name>
  <url>http://maven.apache.org</url>

  <properties>
    <java.version>1.6</java.version>
    <spring.version>3.1.0.RELEASE</spring.version>
    <cglib.version>2.2.2</cglib.version>
  </properties>

  <dependencies>
    <!-- Spring core & mvc -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>${spring.version}</version>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>${spring.version}</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-orm</artifactId>
      <version>${spring.version}</version>
      <type>jar</type>
      <scope>compile</scope>
    </dependency>

    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-test</artifactId>
      <version>${spring.version}</version>
      <type>jar</type>
      <scope>test</scope>
    </dependency>

    <!-- CGLib for @Configuration -->
    <dependency>
```

```
<groupId>cglib</groupId>
<artifactId>cglib-nodep</artifactId>
<version>${cglib.version}</version>
<scope>runtime</scope>
</dependency>

<!-- Servlet Spec -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.4</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.1</version>
  <scope>provided</scope>
</dependency>

<!-- https://mvnrepository.com/artifact
      /org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.5.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact
      /org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.24</version>
  <scope>provided</scope>
</dependency>

</dependencies>

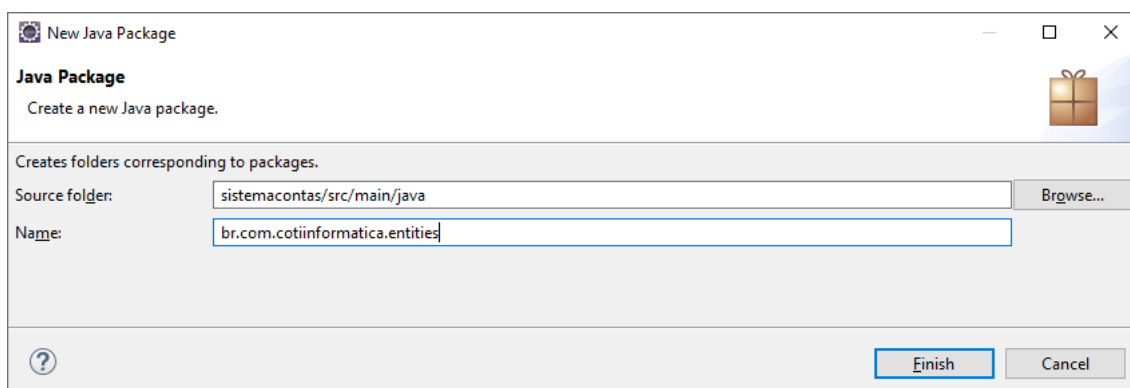
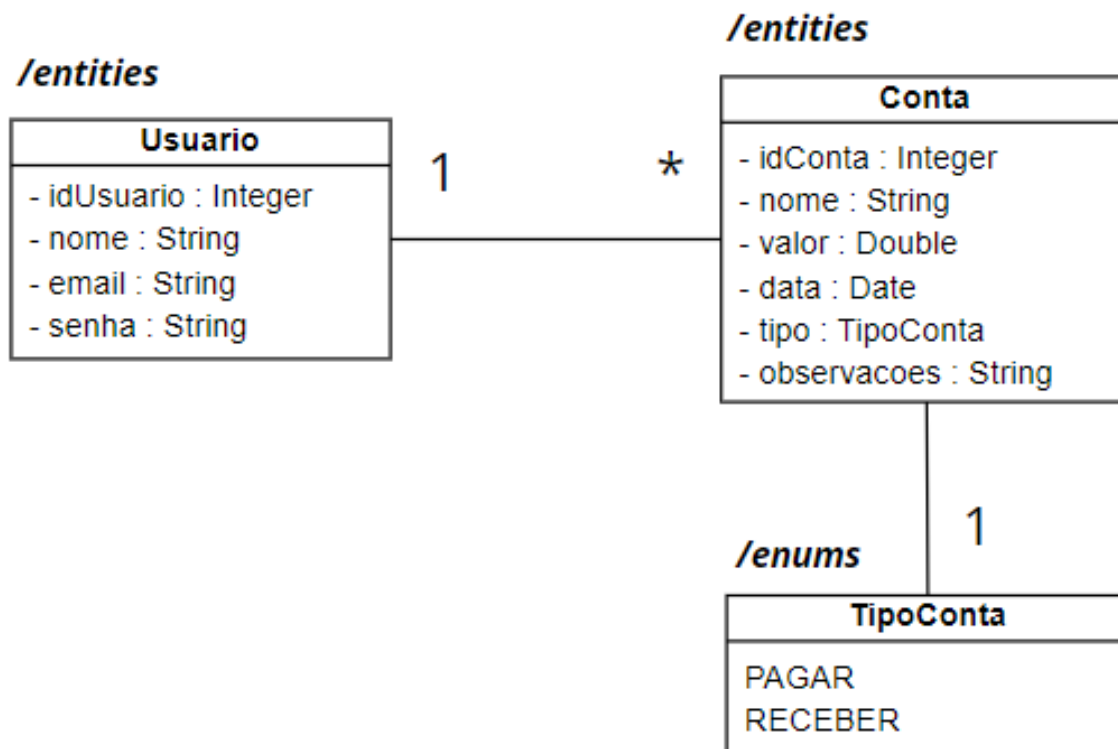
<repositories>
  <repository>
    <id>springsource-milestones</id>
    <name>SpringSource Milestones Proxy</name>
    <url>https://oss.sonatype.org/content
      /repositories/springsource-milestones</url>
  </repository>
</repositories>

<build>
  <finalName>sistemacontas</finalName>
  <plugins>
    <plugin>
```

```
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-compiler-plugin</artifactId>
<version>2.0.2</version>
<configuration>
    <source>${java.version}</source>
    <target>${java.version}</target>
</configuration>
</plugin>
</plugins>
</build>
</project>
```

## Parte 03: Modelando as entidades do projeto

Criando as classes JavaBean para definir a primeira versão do modelo de entidades do sistema:



```
package br.com.cotiinformatica.enums;
```

```
public enum TipoConta {  
    PAGAR,  
    RECEBER  
}
```

## JavaBeans

Padrão Java para construção de classes que tem como objetivo modelar entidades de um sistema (modelo de dados). Essas classes são criadas em Java seguindo o padrão:

- Atributos privados
- Construtor sem entrada de argumentos
- Construtor com entrada de argumentos (sobrecarga de métodos)
- Métodos de encapsulamento
  - Setters
  - Getters
- Sobrescrita de métodos da classe Object

```
package br.com.cotiinformatica.entities;
```

```
import java.util.Date;
```

```
import br.com.cotiinformatica.enums.TipoConta;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Getter;
```

```
import lombok.NoArgsConstructor;
```

```
import lombok.Setter;
```

```
import lombok.ToString;
```

```
@Setter
```

```
@Getter
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@ToString
```

```
public class Conta {
```

```
    private Integer idConta;
```

```
    private String nome;
```

```
    private Double valor;
```

```
    private Date data;
```

```
    private TipoConta tipo;
```

```
    private String observacoes;
```

```
    private Usuario usuario;
```

```
}
```

```
package br.com.cotiinformatica.entities;
```

```
import java.util.List;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Getter;
```

```
import lombok.NoArgsConstructor;
```

```
import lombok.Setter;
```

```
import lombok.ToString;
```

```
@Setter
```

```
@Getter
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@ToString
```

```
public class Usuario {
```

```
    private Integer idUsuario;
```

```
    private String nome;
```

```
    private String email;
```

```
    private String senha;
```

```
    private List<Conta> contas;
```

```
}
```

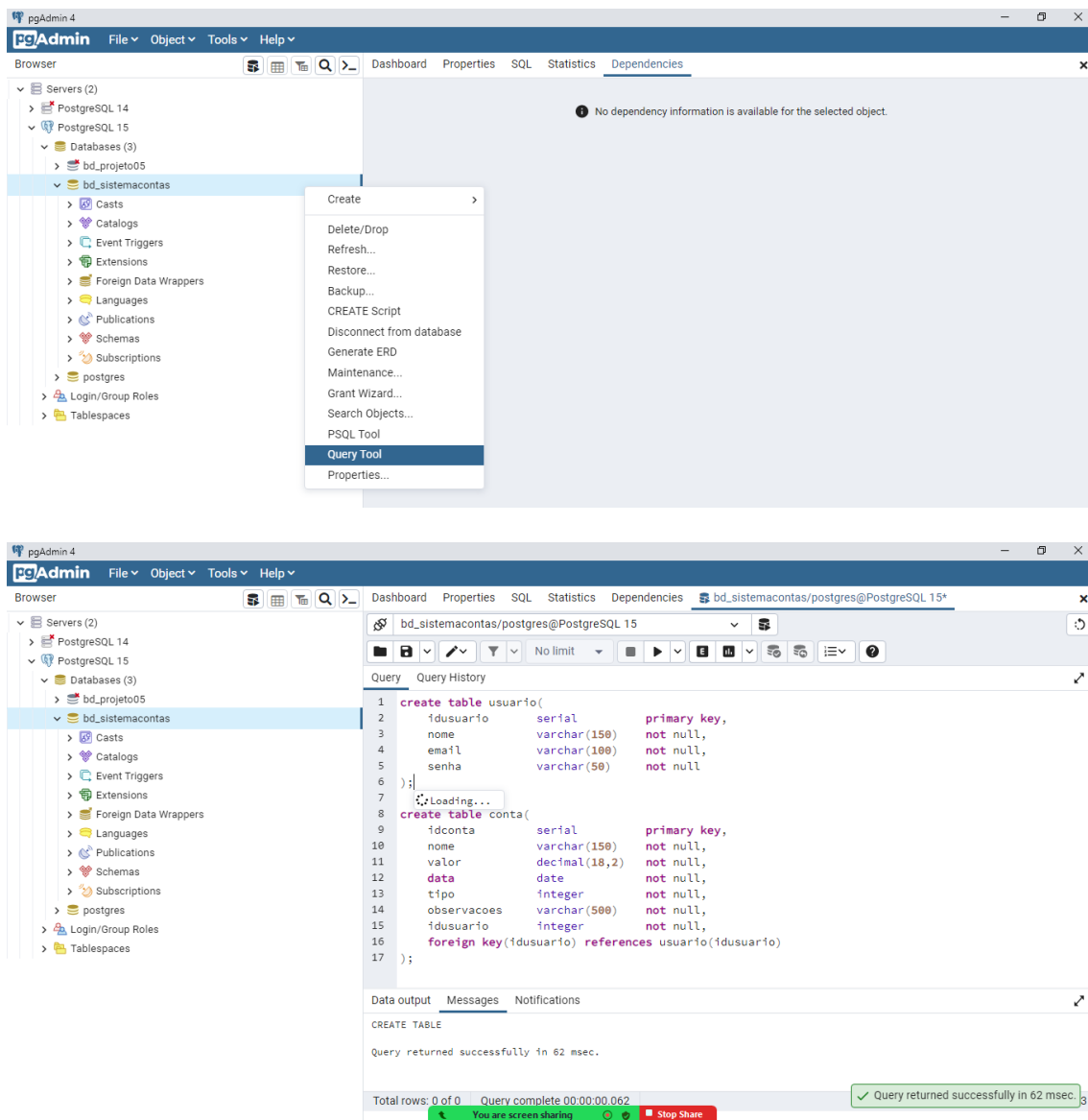
---

### Script para criação das tabelas no banco de dados:

```
create table usuario(  
    idusuario          serial          primary key,  
    nome               varchar(150)    not null,  
    email              varchar(100)    not null,  
    senha              varchar(50)     not null  
);
```

```
create table conta(  
    idconta            serial          primary key,  
    nome               varchar(150)    not null,  
    valor              decimal(18,2)   not null,  
    data               date            not null,  
    tipo               integer         not null,  
    observacoes        varchar(500)    not null,  
    idusuario          integer         not null,  
    foreign key(idusuario) references usuario(idusuario)  
);
```

### Executando no PostgreSQL:



The top screenshot shows the pgAdmin 4 interface with the 'Query Tool' menu option selected. The bottom screenshot shows the SQL query execution results for creating two tables: 'usuario' and 'conta'.

```

1 create table usuario(
2   idusuario serial primary key,
3   nome varchar(150) not null,
4   email varchar(100) not null,
5   senha varchar(50) not null
6 );
7
8 create table conta(
9   idconta serial primary key,
10  nome varchar(150) not null,
11  valor decimal(18,2) not null,
12  data date not null,
13  tipo integer not null,
14  observacoes varchar(500) not null,
15  idusuario integer not null,
16  foreign key(idusuario) references usuario(idusuario)
17 );
  
```

Query returned successfully in 62 msec.

### Fazendo o commit do trabalho para o GITHUB:

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)

\$ git add .

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)

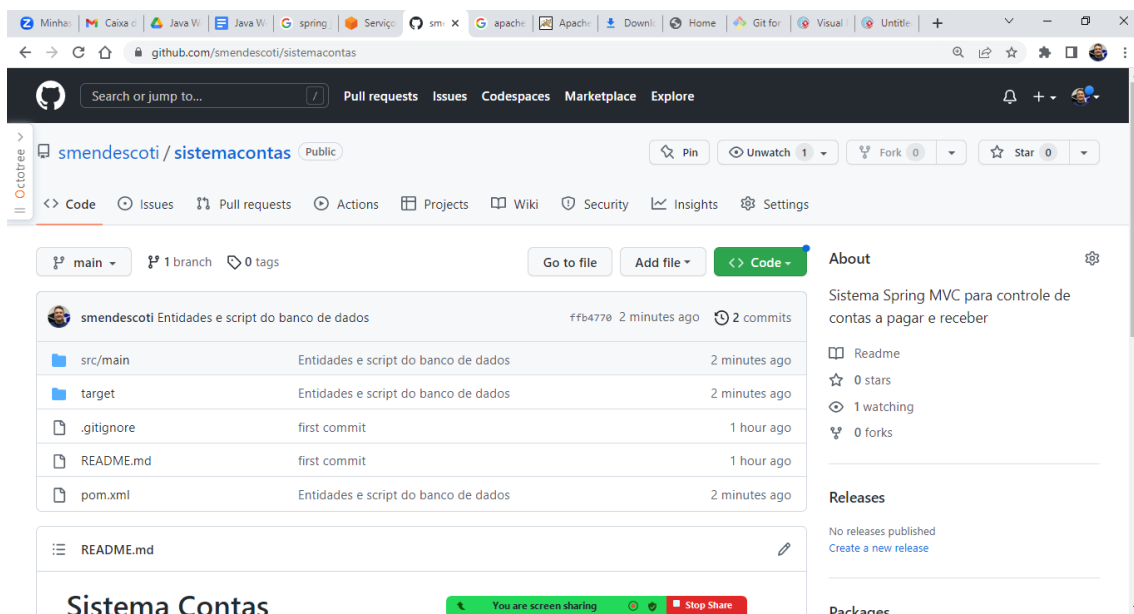
\$ git commit -m 'Entidades e script do banco de dados'

### Fazendo o PUSH:

```
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git push -u origin main
```

```
MINGW64/c/Users/Samsung/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git add .
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git commit -m 'Entidades e script do banco de dados'
[main ffb4770] Entidades e script do banco de dados
11 files changed, 123 insertions(+), 1 deletion(-)
create mode 100644 src/main/java/br/com/cotiinformatica/entities/Conta.java
create mode 100644 src/main/java/br/com/cotiinformatica/entities/Usuario.java
create mode 100644 src/main/java/br/com/cotiinformatica/enums/TipoConta.java
create mode 100644 src/main/java/br/com/cotiinformatica/sql/script.sql
create mode 100644 target/classes/br/com/cotiinformatica/entities/Conta.class
create mode 100644 target/classes/br/com/cotiinformatica/entities/Usuario.class
create mode 100644 target/classes/br/com/cotiinformatica/enums/TipoConta.class
create mode 100644 target/classes/br/com/cotiinformatica/sql/script.sql
Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java webDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)
$ git push -u origin main
Enumerating objects: 52, done.
Counting objects: 100% (52/52), done.
Delta compression using up to 8 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (33/33), 5.14 KiB | 752.00 KiB/s, done.
Total 33 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/smendescoti/sistemacontas.git
 8d4f71c..ffb4770 main -> main
branch 'main' set up to track 'origin/main'.
```

<https://github.com/smendescoti/sistemacontas>



The screenshot shows the GitHub repository page for `smendescoti/sistemacontas`. The repository is public and has 1 branch (main) and 0 tags. The commit history shows a recent commit by `smendescoti` titled "Entidades e script do banco de dados" (ffb4770) 2 minutes ago, with 2 commits in total. The file list includes `src/main`, `target`, `.gitignore`, `README.md`, and `pom.xml`. The repository description is "Sistema Spring MVC para controle de contas a pagar e receber". The repository has 0 stars, 1 watching, and 0 forks. The README section is partially visible, showing the title "Sistema Contas".