1. Configurando o ambiente de desenvolvimento

A IDE Eclipse IDE for Enterprise Java and Web Developers Ferramentas para desenvolvedores que trabalham com aplicativos Java e Web, incluindo um IDE Java, ferramentas para JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA e Data Tools, Maven e Gradle, Git e muito mais. Este pacote inclui:

• Plataforma de ferramentas de dados

• Integração Git para Eclipse

• Ferramentas de Desenvolvimento Java Eclipse

• Ferramentas do desenvolvedor Eclipse Java EE

• Integração Maven para Eclipse

• Lista de Tarefas Mylyn

• Ambiente de Desenvolvimento de Plug-in Eclipse

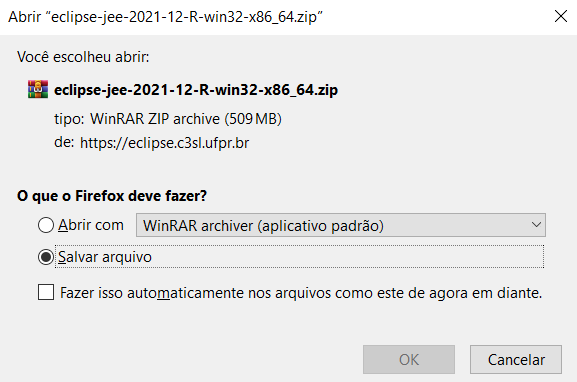
* 1. Procedimento para Configurar o ambiente de desenvolvimento

1. link: <https://www.eclipse.org/downloads/packages/release/2021-12/r/eclipse-ide-enterprise-java-and-web-developers>
2. Fazer o download do ***Eclipse IDE for Enterprise Java and Web Developers*** para Windows:

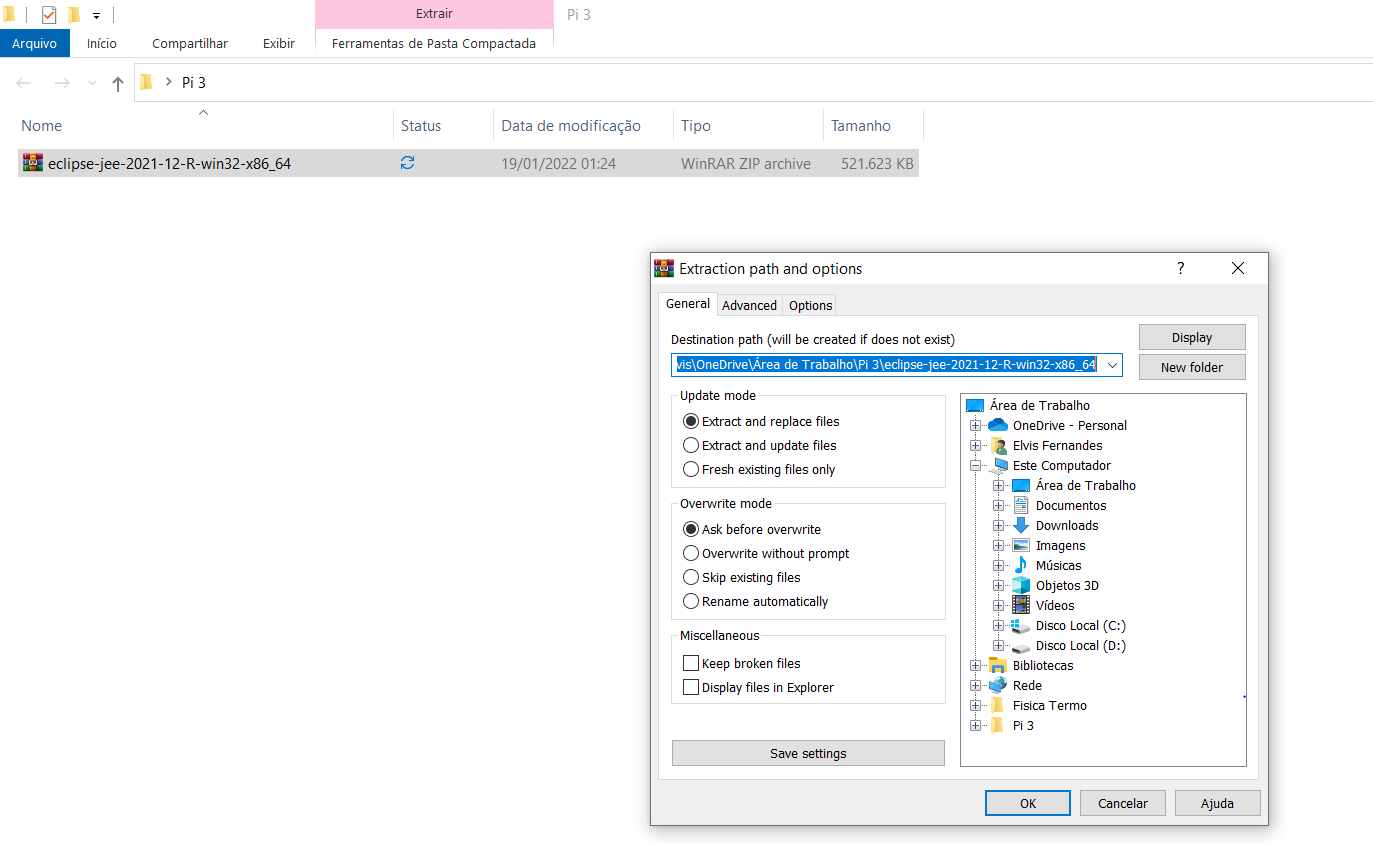


1. Clicar no Botão Download

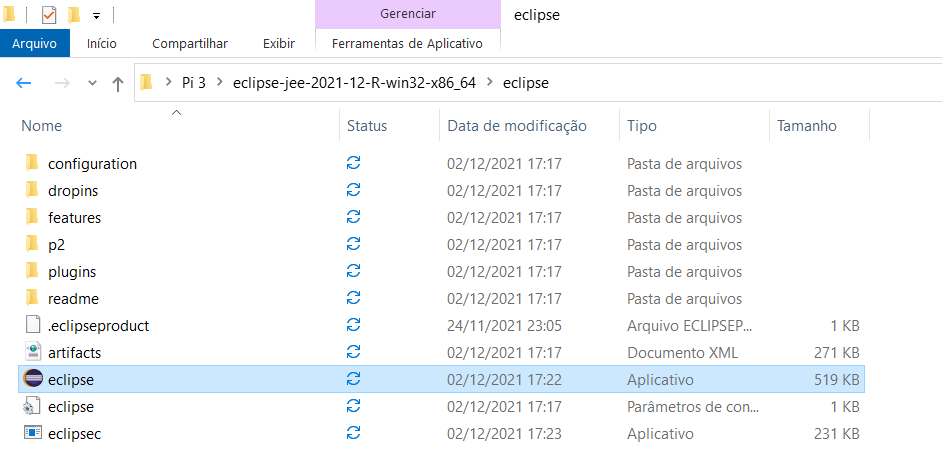




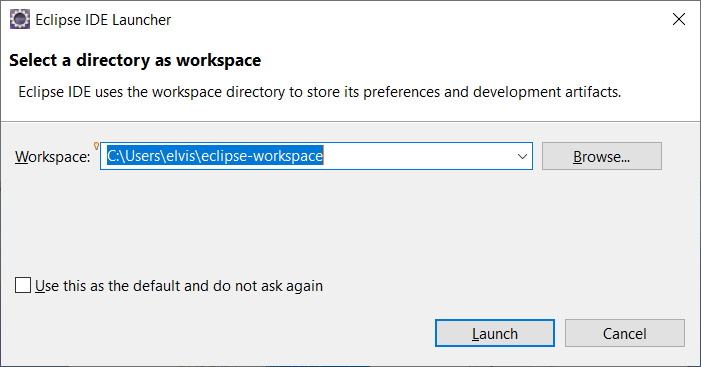
1. Instalar o Software
   1. Extrair os arquivos zipados



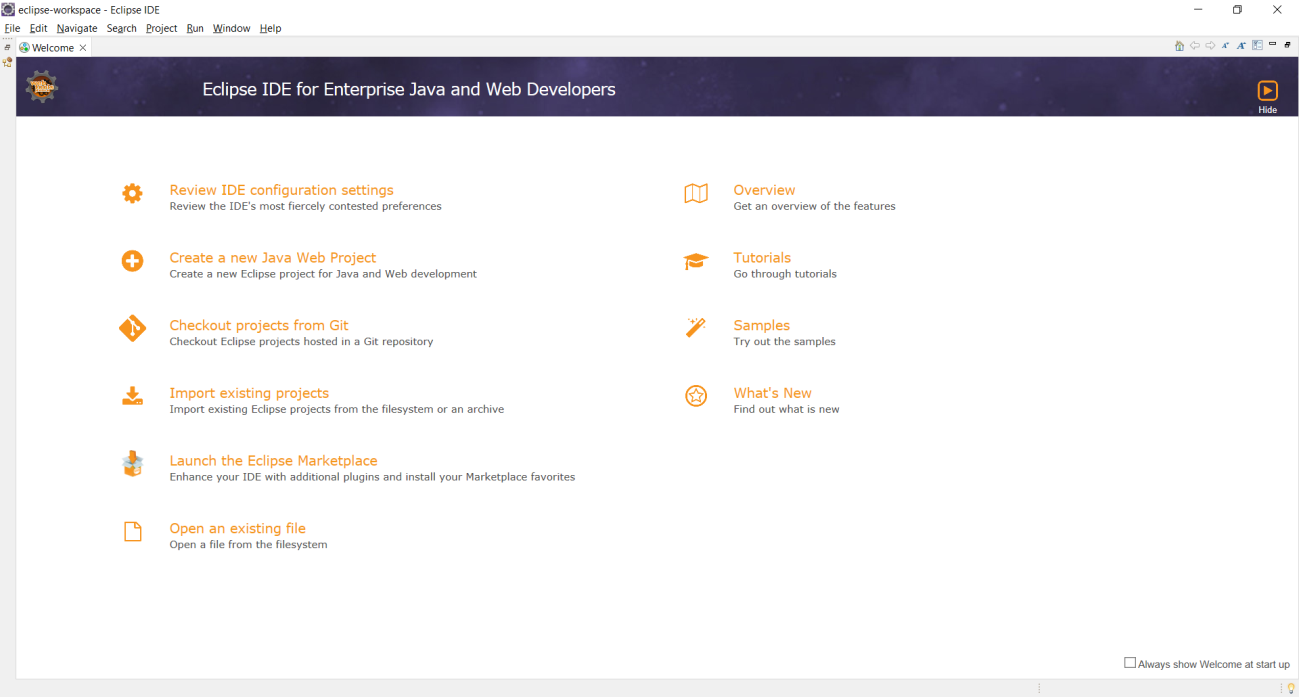
* 1. Clicar no aplicativo Eclipse



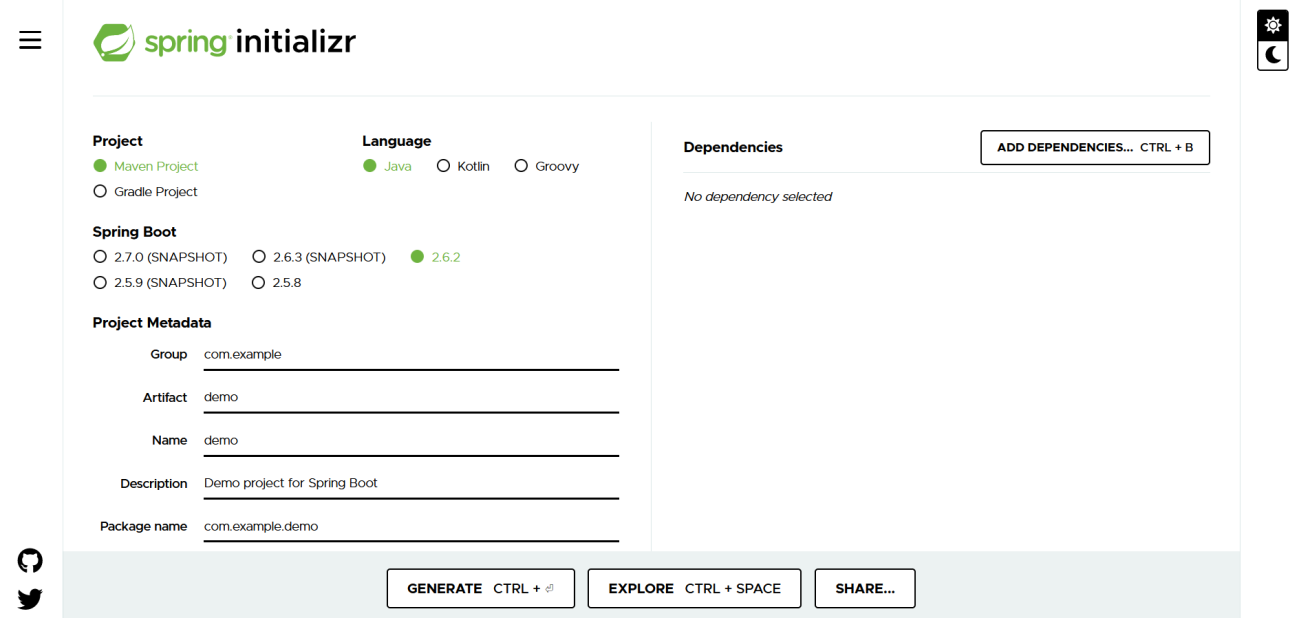
* 1. Configurar ***Workspace*** (Clicar no botão ***Launch***)



* 1. Validação da Configuração do ambiente de desenvolvimento



1. Criação do Projeto
   1. Acessar o link: [**https://start.spring.io/**](https://start.spring.io/)

****

* 1. Requisitos de Projeto:
* **Padrão: Maven Project**
* **Linguagem: Java**
* **Spring Boot: 2.62**
* **Project Metada**

**Group: com.temperatura**

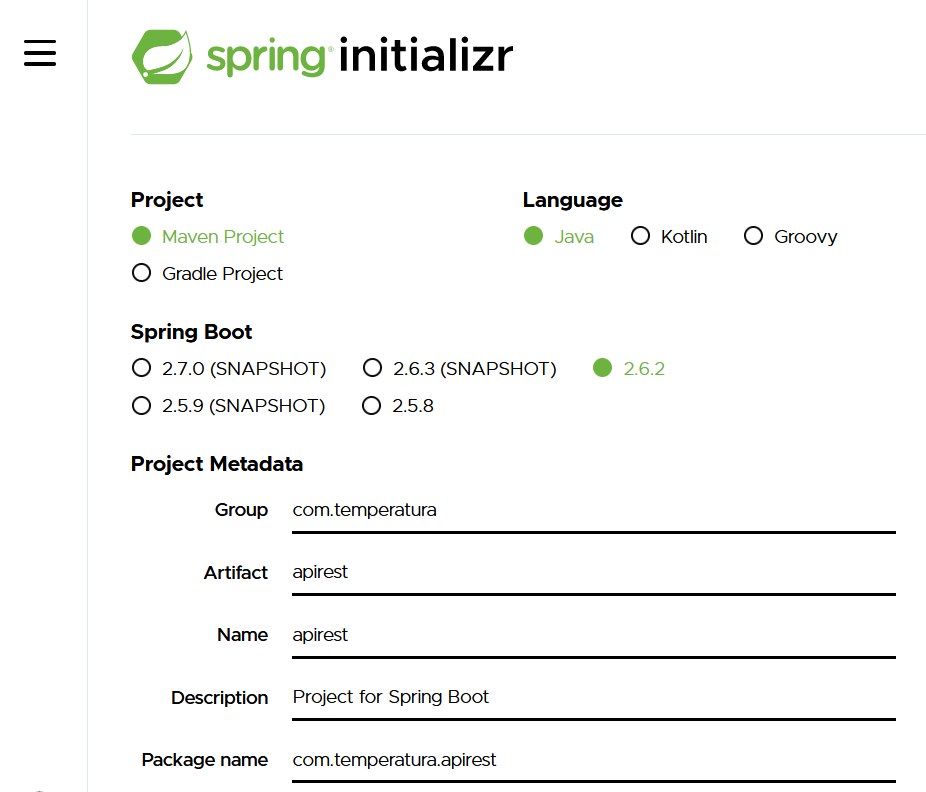
**Artifact: apirest**

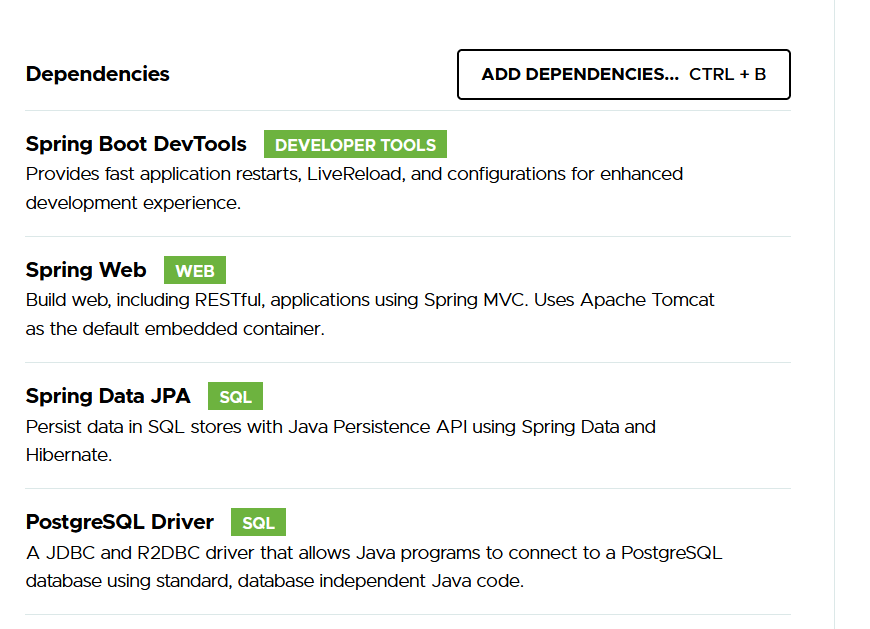
**Name: apirest**

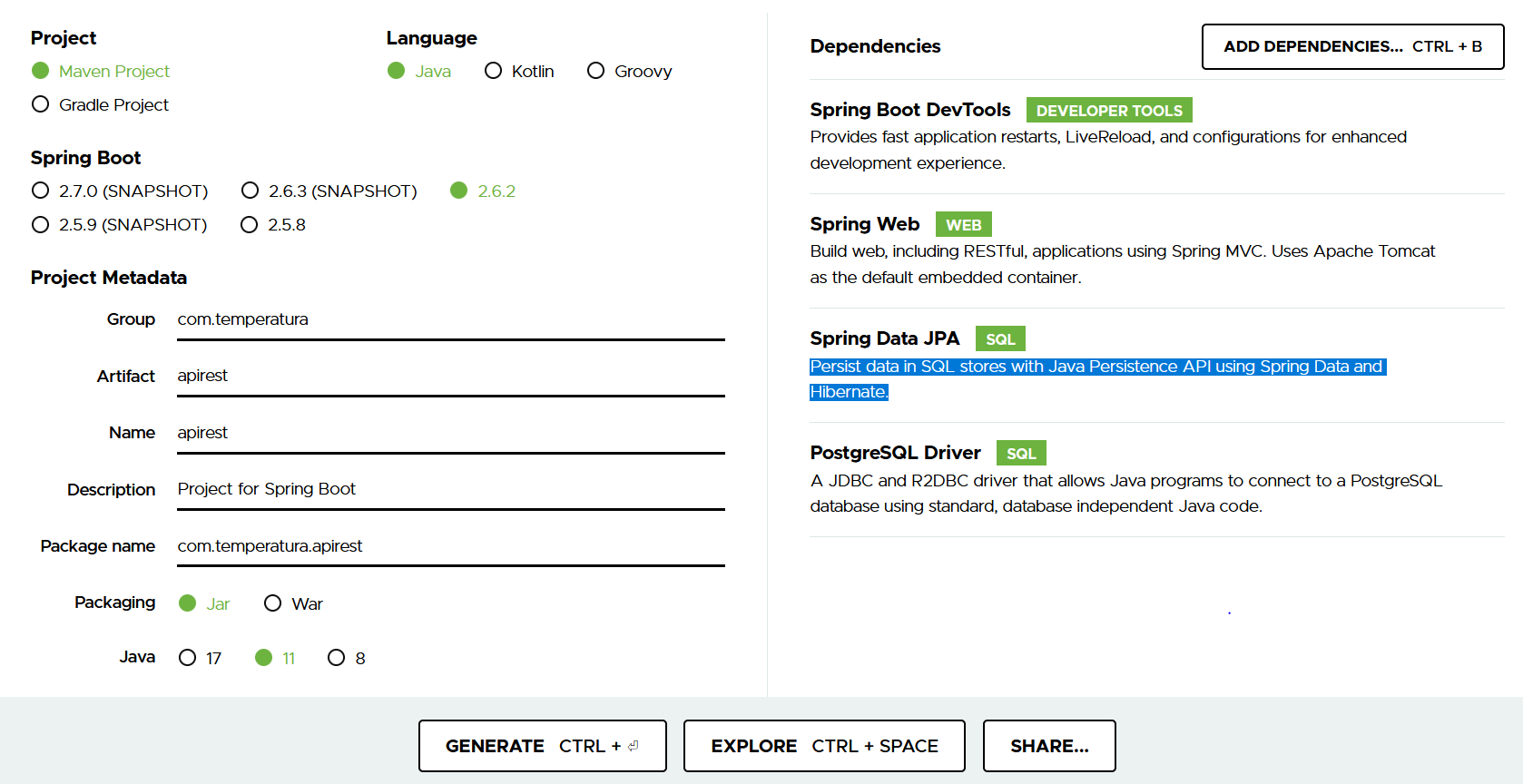
**Description: Project for Spring Boot**

**Package name: com.temperatura.apirest**

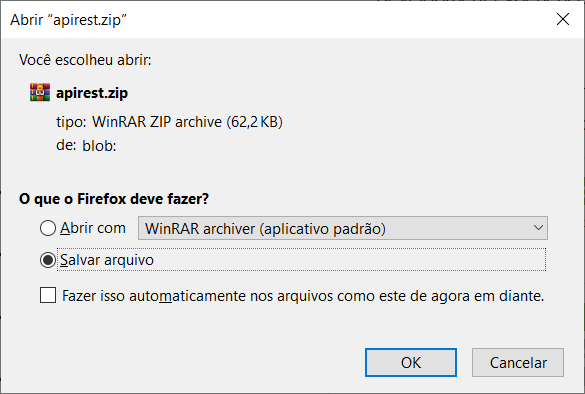
* **Packging: Jar**
* **Java: 11**
  1. Dependências:
* ***Spring Boot DevTools***: Fornece reinicializações rápidas de aplicativos, LiveReload e configurações para uma experiência de desenvolvimento aprimorada.
* ***Spring Web WebBuild web:*** Criar aplicativos web, incluindo RESTful, usando Spring MVC. Usa o Apache Tomcat como o contêiner incorporado padrão.
* ***Spring Data JPA***: Persistir dados em armazenamentos SQL com Java Persistence API usando Spring Data e Hibernate.
* ***PostgreSQL Driver***: Um driver JDBC e R2DBC que permite que programas Java se conectem a um banco de dados PostgreSQL usando código Java padrão independente de banco de dados..

****

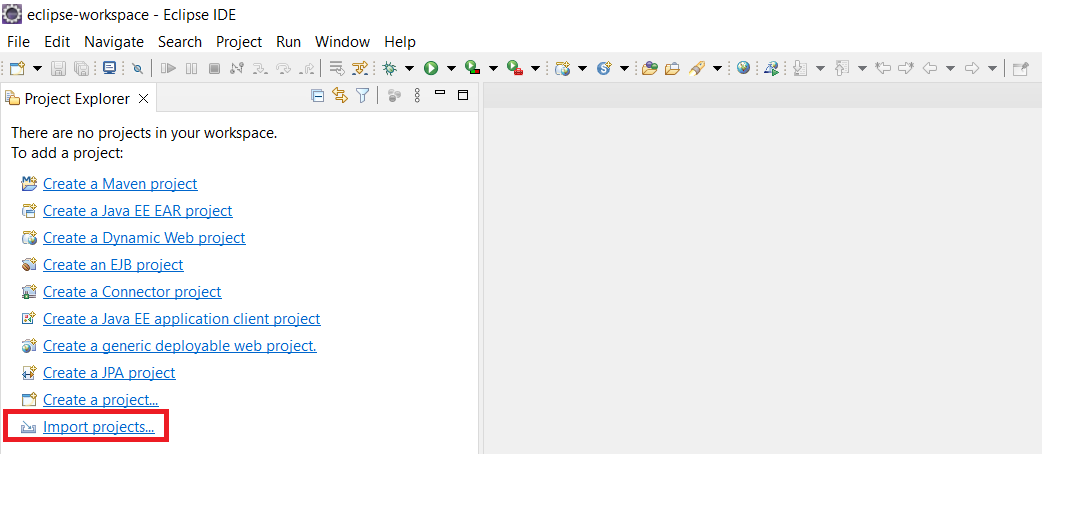
****



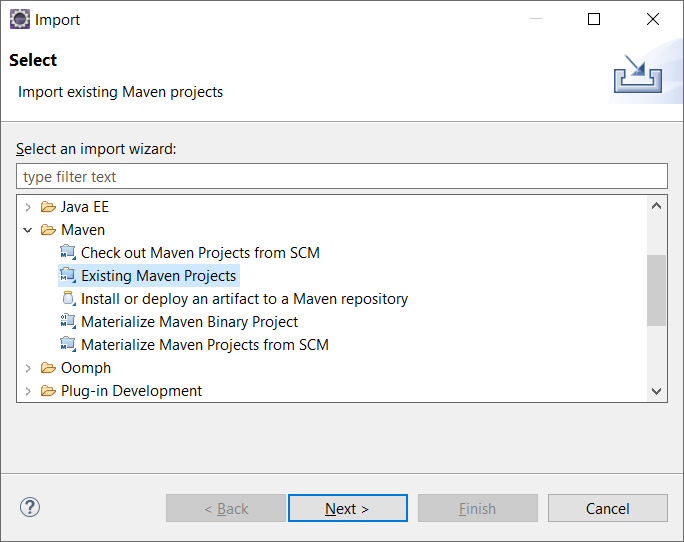
* 1. Gerar o arquivo ***Maven*** (Clicar no botão ***GENERATE***)



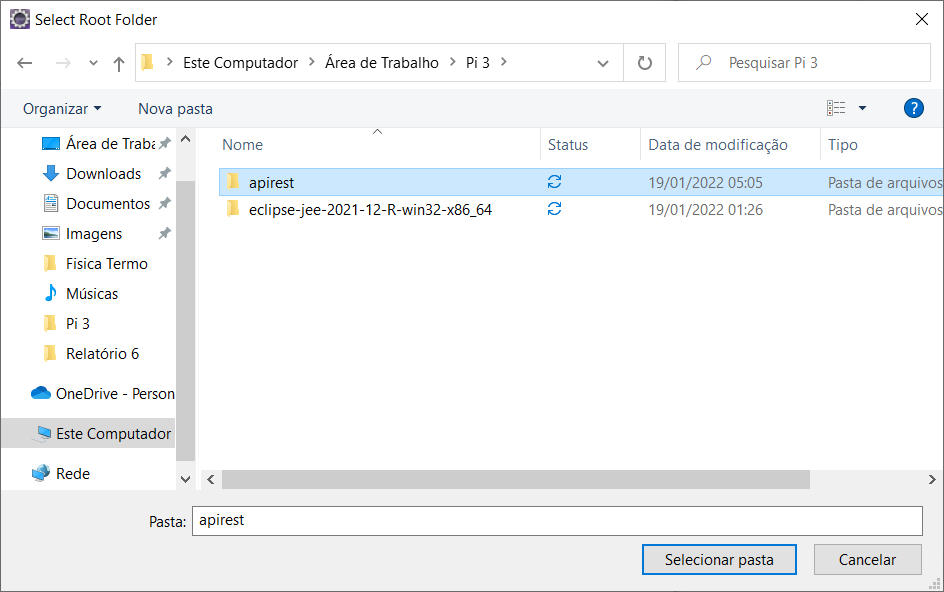
* 1. Importar o arquivo ***Maven*** gerado (apirest)
* Clicar em ***Import projects***



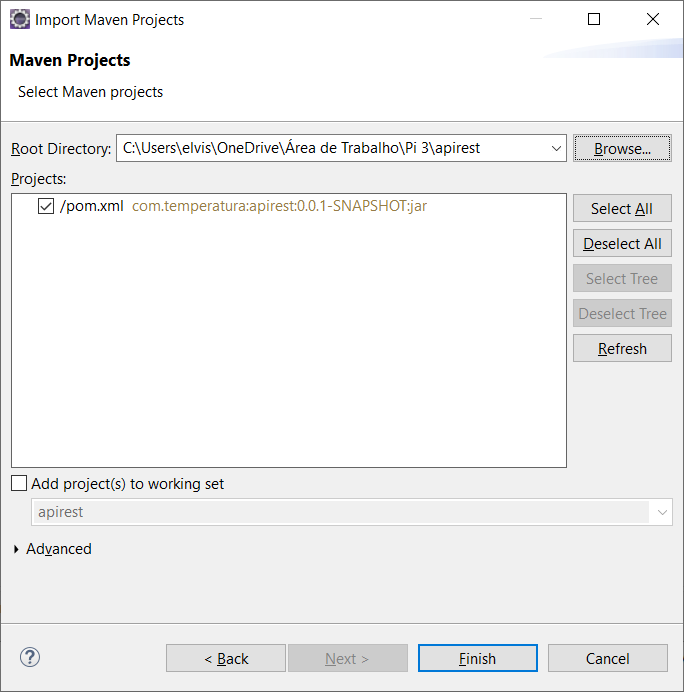
* Selecionar ***Maven > Existing Maven Projects***



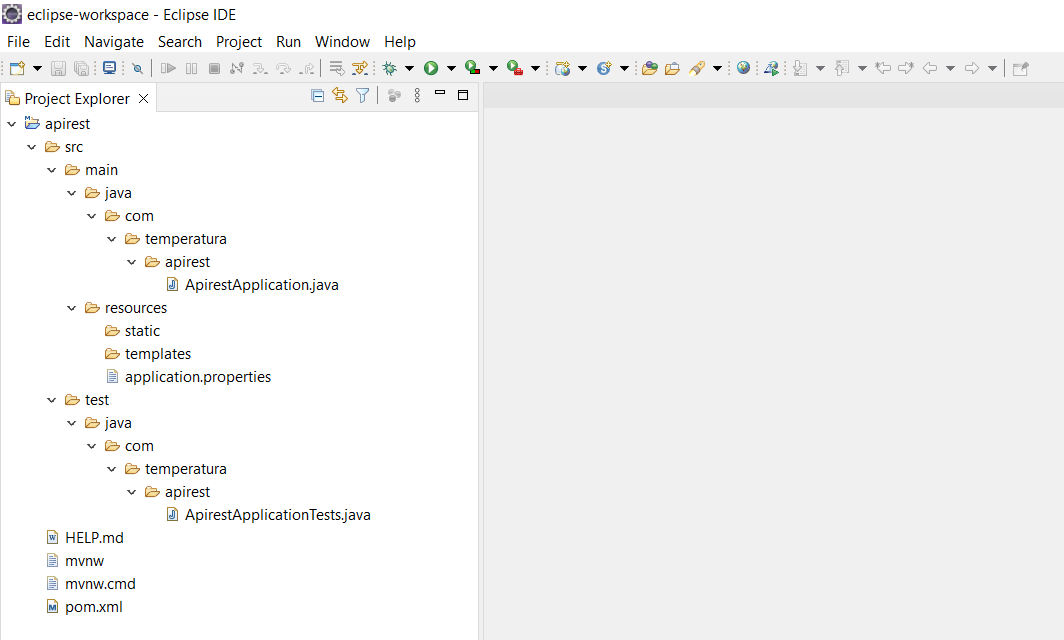
* Clicar em ***Next***
* Selecionar a pasta do o arquivo ***Maven*** (gerado e extraído)

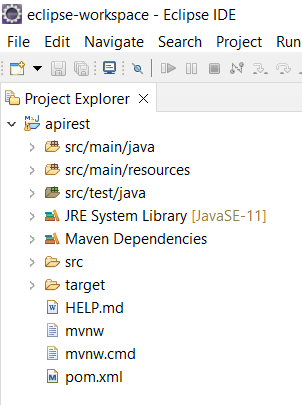


* Clicar em ***Finish***



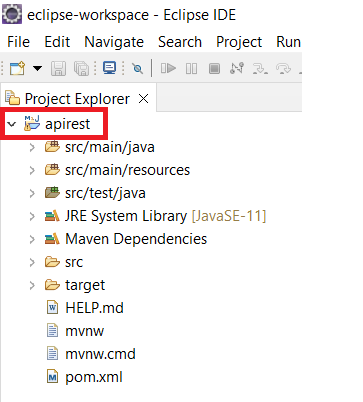
2.6 Estrutura do arquivo ***Maven*** Base



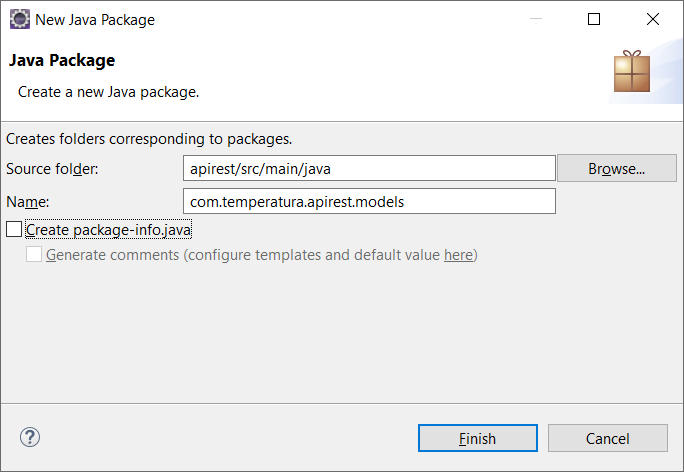


1. Desenvolvimento do Projeto
   1. Criação do Pacote Models

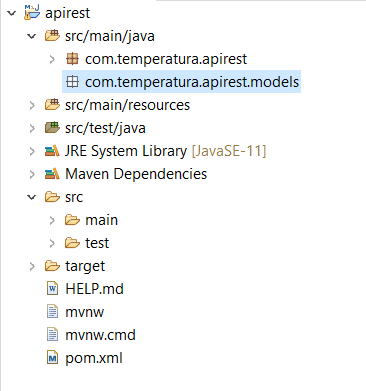
* Clicar com o botão direito do mouse em ***apirest > New > Package***



* Source folder: **apirest/src/main/Java**
* Name: **com.temperatura.apirest.models**
* Clicar em ***Finish***

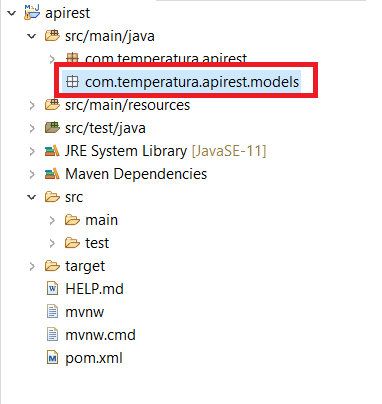


* Validação da criação do Models

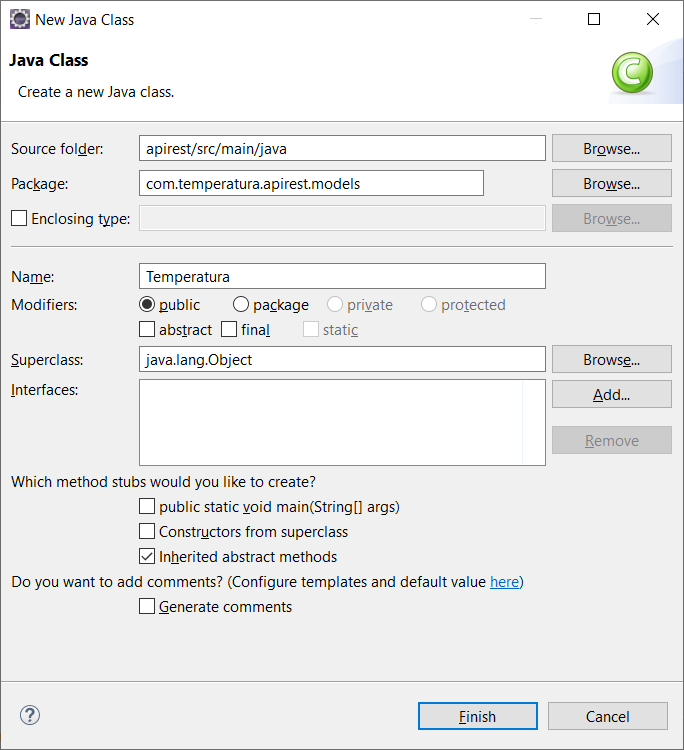


* 1. Criação da classe Temperatura.java dentro do pacote.models
* Clicar com o botão direito do mouse em:

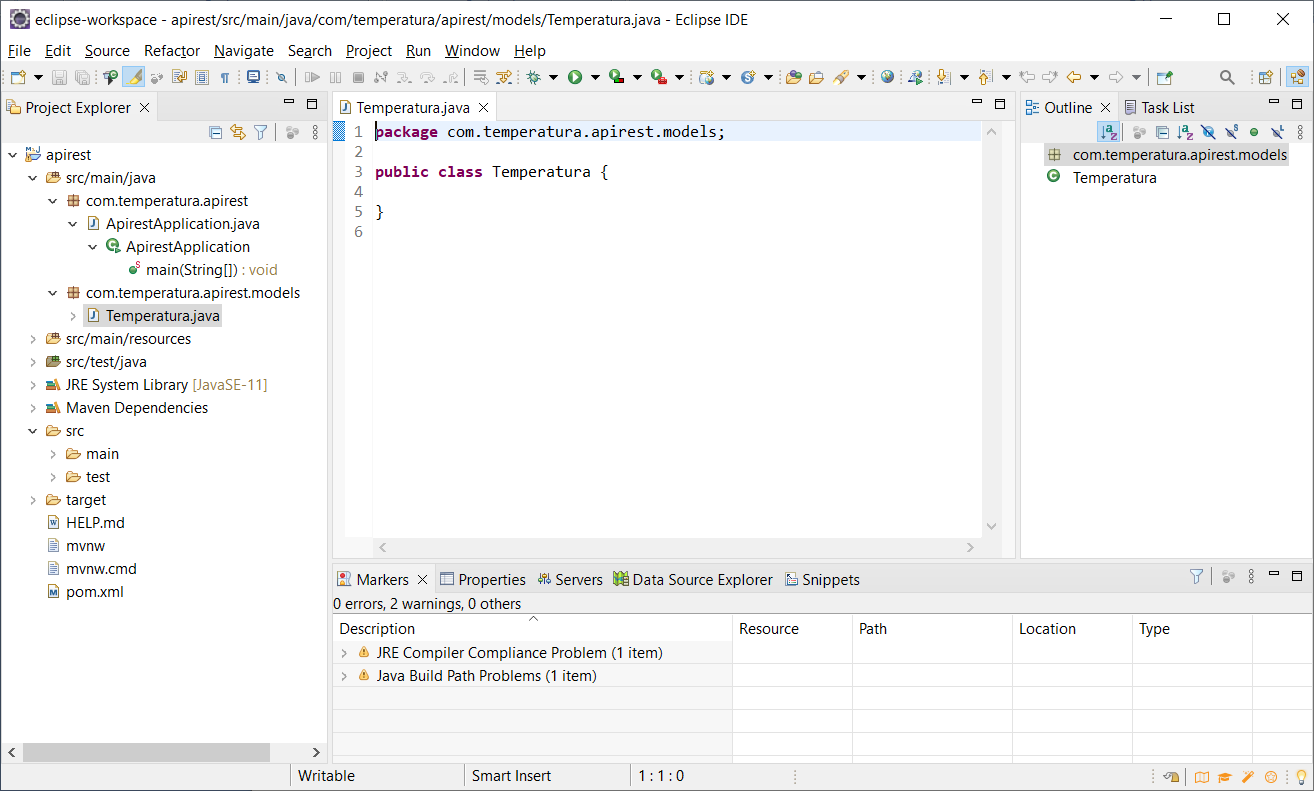
***com.temperatura.apirest.models > New > Class***



* Name: **Temperatura**
* Clicar em ***Finish***



* Validação da criação do pacote Models

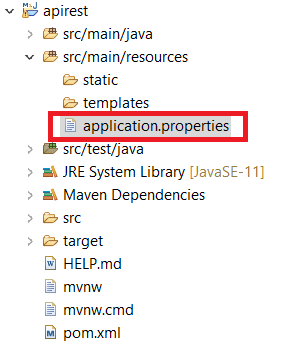


* 1. Desenvolvimento do Código da Classe Temperatura

|  |
| --- |
| **package** com.temperatura.apirest.models;  **import** java.io.Serializable;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.Table;  @Entity  @Table(name = "TB\_TEMPERATURA")  **public** **class** Temperatura **implements** Serializable {  /\*\*  \*  \*/  **private** **static** **final** **long** ***serialVersionUID*** = 1L;  @Id  @GeneratedValue(strategy = GenerationType.***AUTO***)  **private** **long** idDispositivo;  **private** String valorSensor;  **public** **long** getIdDispositivo() {  **return** idDispositivo;  }  **public** **void** setIdDispositivo(**long** idDispositivo) {  **this**.idDispositivo = idDispositivo;  }  **public** String getValorSensor() {  **return** valorSensor;  }  **public** **void** setValorSensor(String valorSensor) {  **this**.valorSensor = valorSensor;  }  } |

4) **Configurar o banco de dados no arquivo *application.properties***

* Abrir o arquivo ***application.properties***

****

* 1. Desenvolvimento do Código do Arquivo de Banco de Dados

|  |
| --- |
| spring.jpa.properties.hibernate.jdbc.lob.non\_contextual\_creation=true  #Banco local - Elvis  spring.datasource.url= jdbc:postgresql://localhost:5432/temperatura\_apirest  spring.datasource.username=postgres  spring.datasource.password=banco123  spring.jpa.hibernate.ddl-auto=update |

Em que:

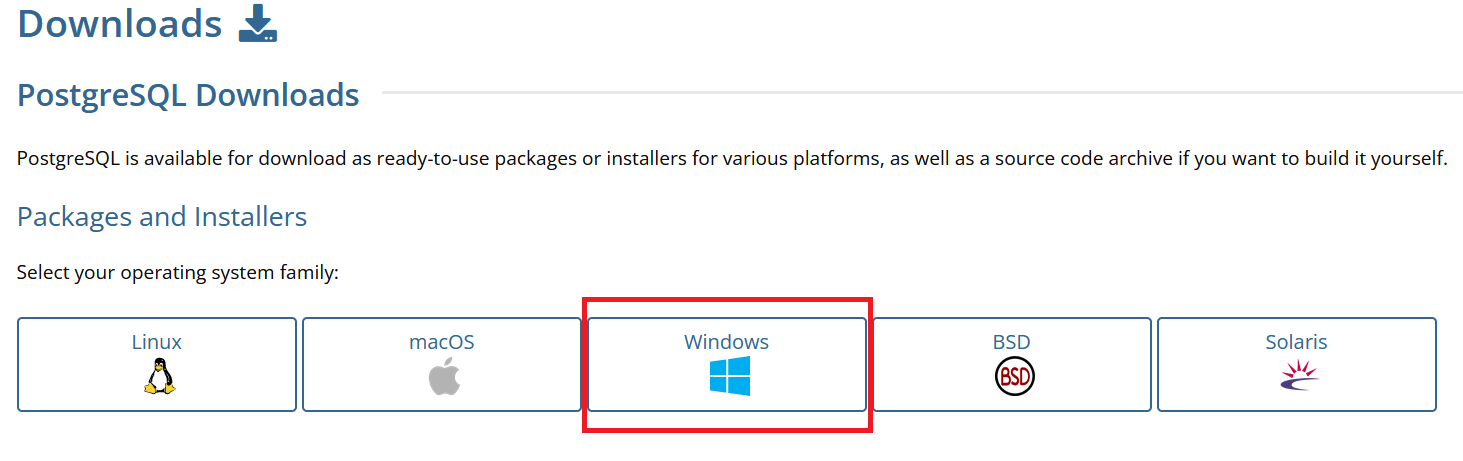
* Nome do banco de dados: ***temperatura\_apirest***
* Usuário do banco de dados: ***postgres***
* Senha do banco de dados: ***banco123***

**5) Instalação do Banco de Dados PostgresSQL**

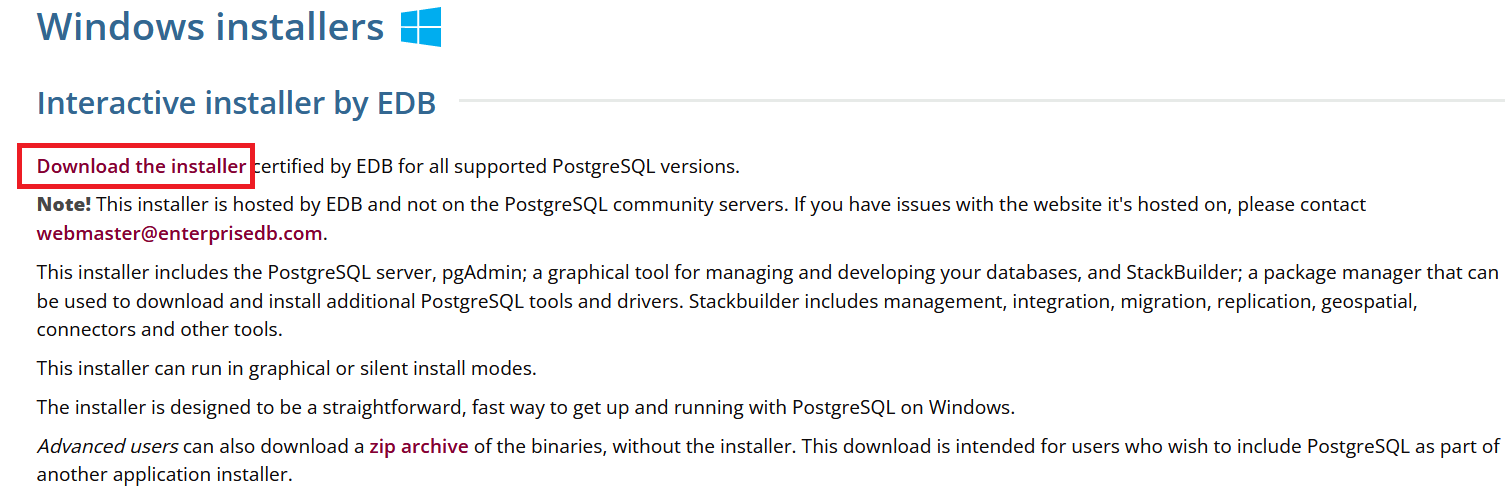
* **Acessar o link:** [**https://www.postgresql.org/**](https://www.postgresql.org/)
* **Clicar em Download**



* Clicar em ***Windows***



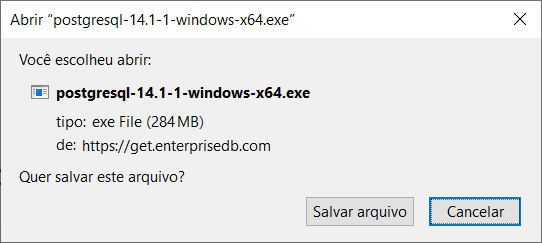
* Clicar em ***Download the installer***

****

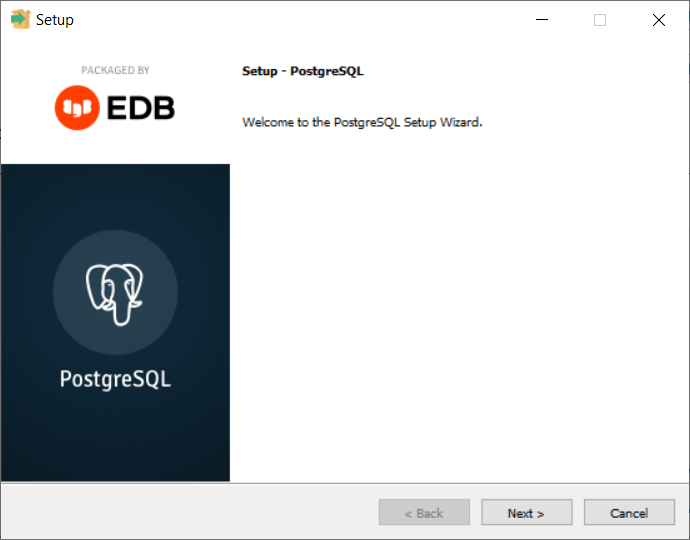
* Selecionar a versão ***14.1*** para ***Windows 64 bits***

****

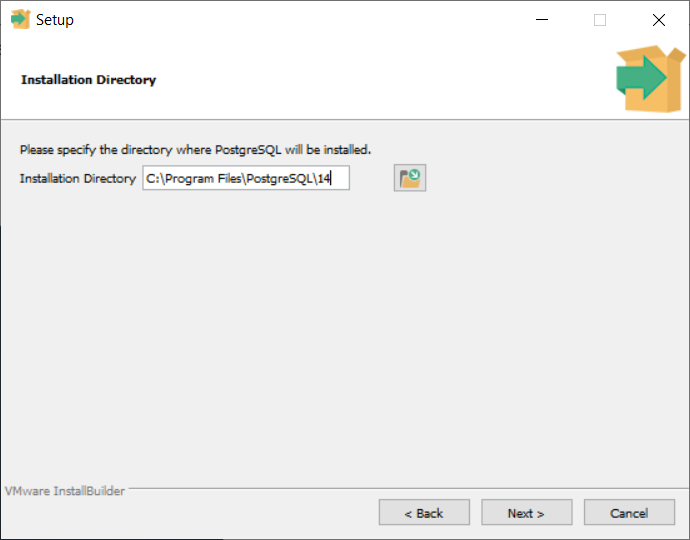
* Clicar em ***Salvar arquivo***



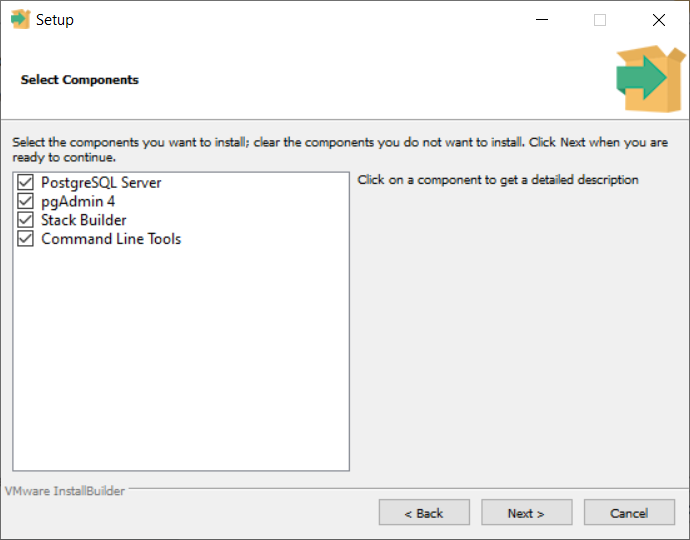
* Abrir o aplicativo **postgresql-14.1-1-windows-x64**
* **Clicar em Next**



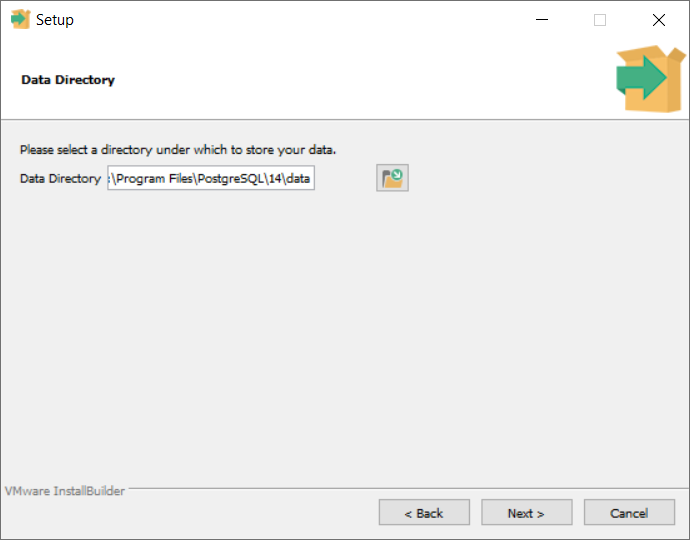
* Escolher o diretório de Instalação
* Clicar em ***Next***



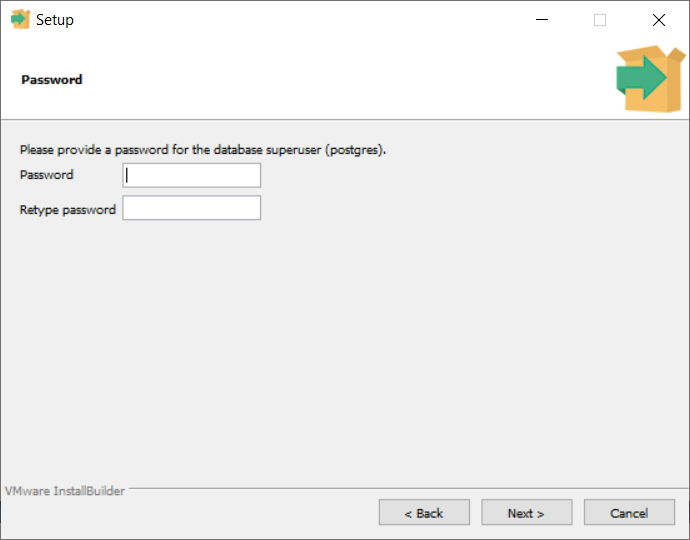
* Selecionar os componentes
* Clicar em *Next*



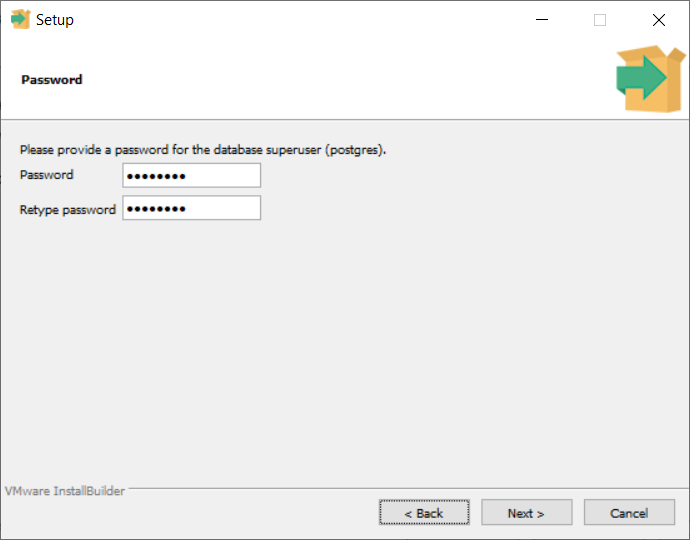
* Escolher o diretório de dados
* Clicar em ***Next***

****

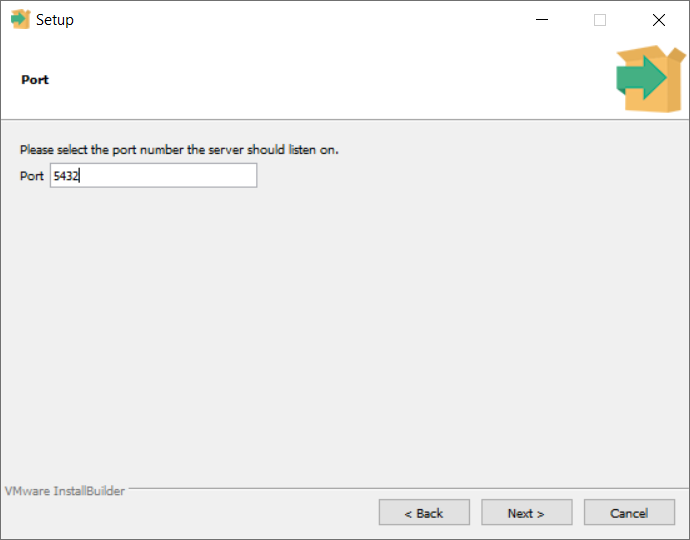
* **Digitar uma senha para o superusuário postgres**

****

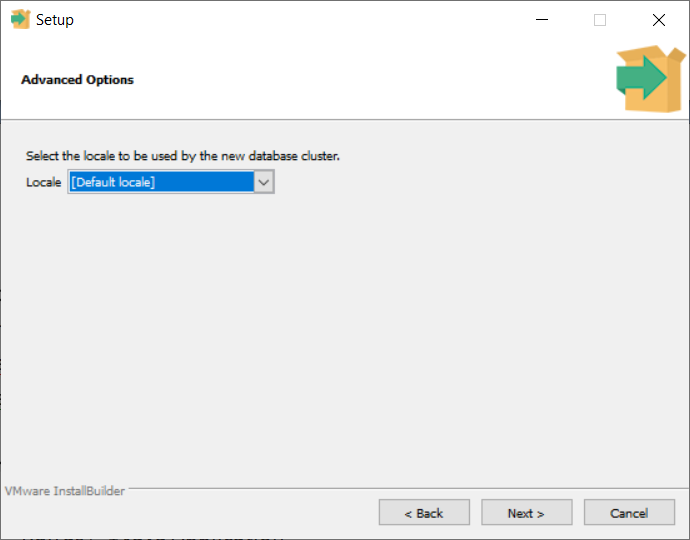
* **Senha escolhida: banco123**
* Clicar no botão ***Next***

****

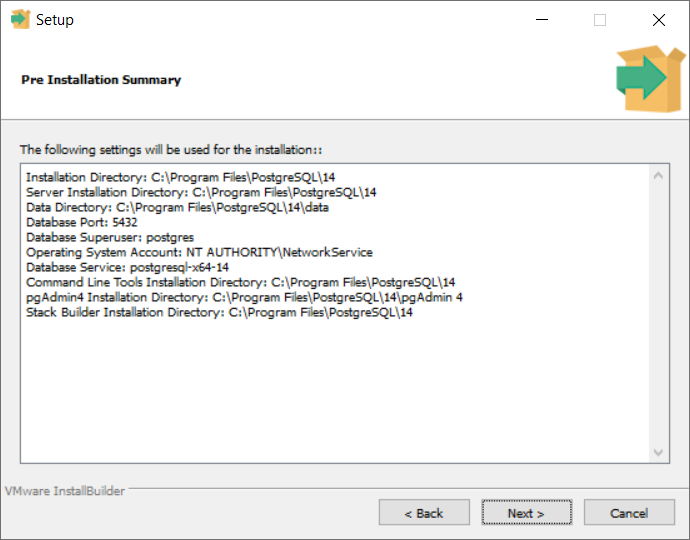
* Selecionar o número da Porta
* Clicar em ***Next***

****

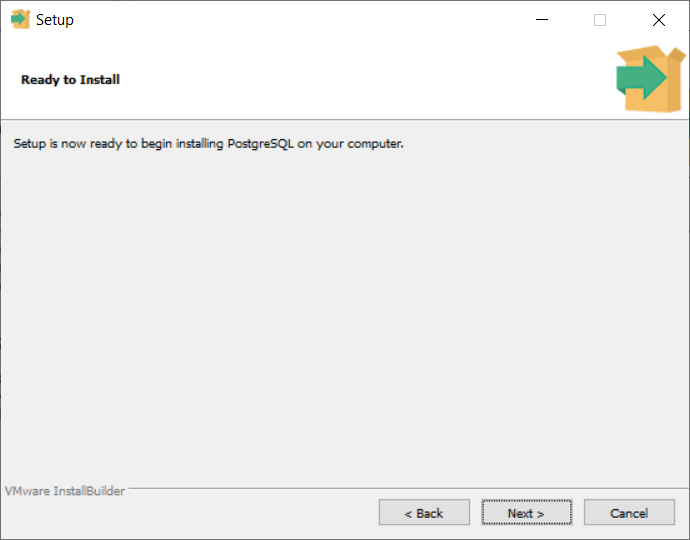
* Selecionar o local do banco de dados (**cluster**)
* ***Default Locale***
* Clicar em ***Next***

****

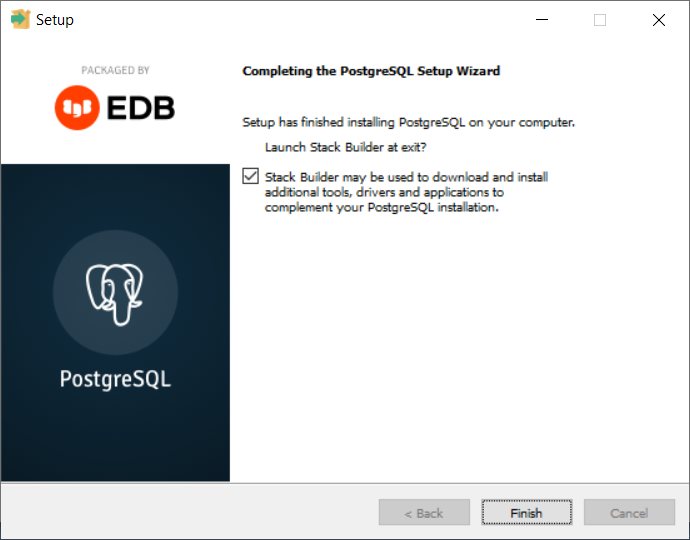
* Clicar em ***Next***

****

* Clicar em ***Next***

****

* Clicar em ***Finish***

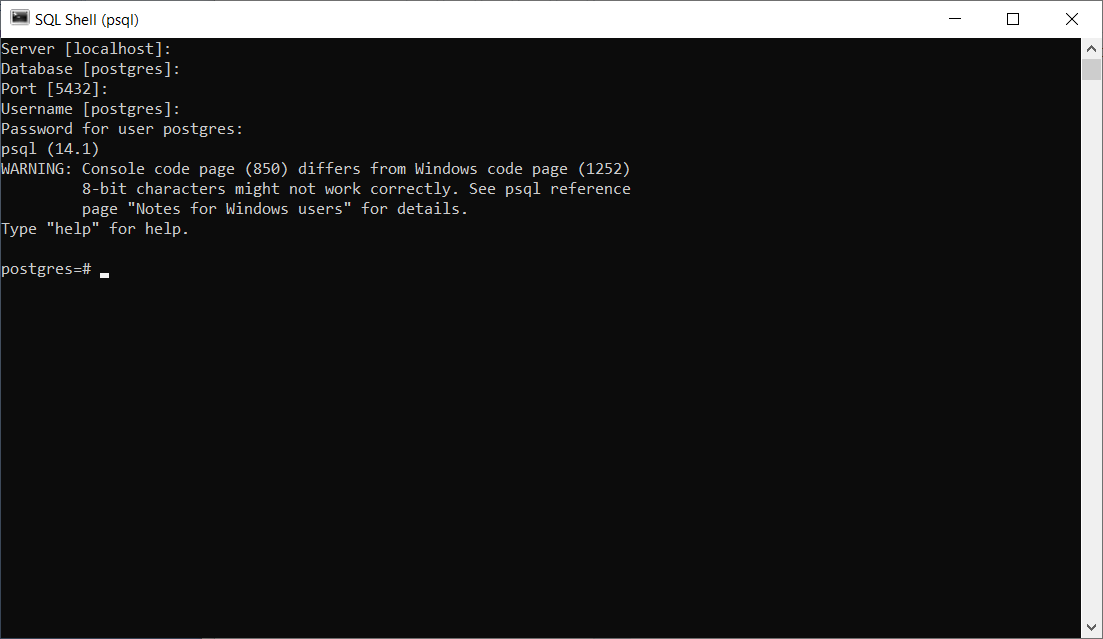


* 1. **Criação do Banco de Dados Local**
* Abrir o SQL Shell (psql)



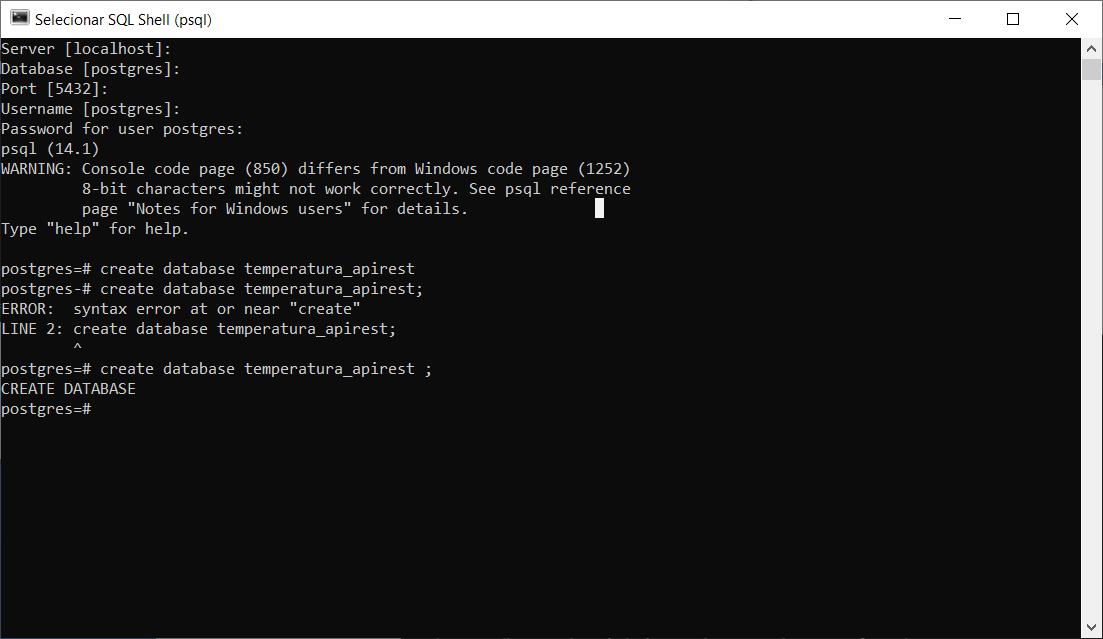
* Configurar

1. Apertar Enter (no teclado)
2. Apertar Enter (no teclado)
3. Apertar Enter (no teclado)
4. Apertar Enter (no teclado)
5. Digitar a senha banco123
6. Apertar Enter (no teclado)

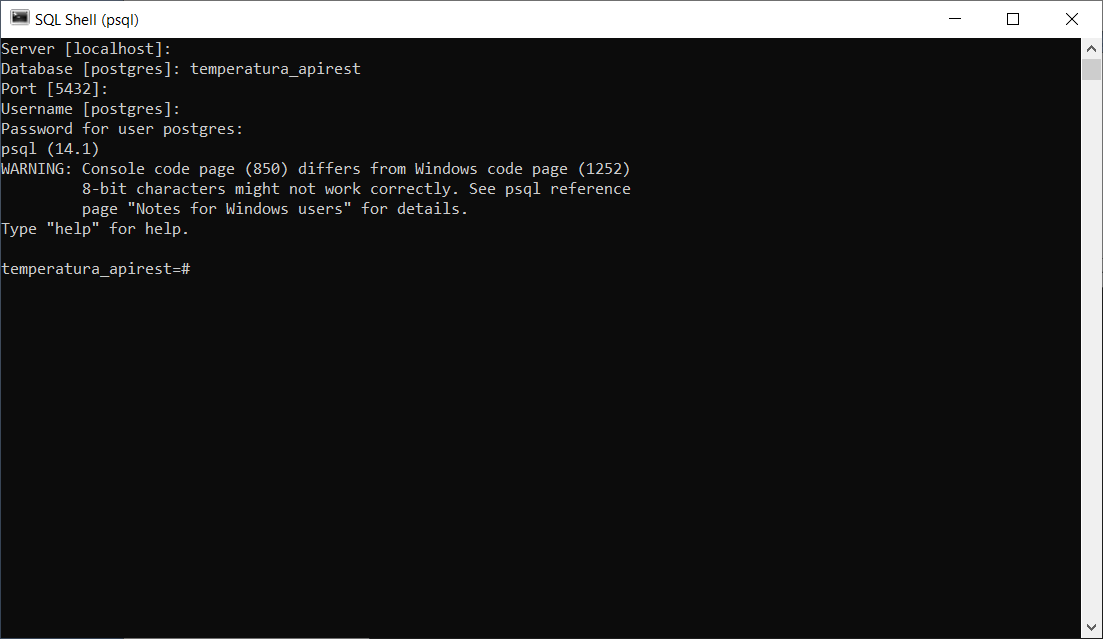


* Criar o banco de dados com o seguinte comando:

***create database temperatura\_apirest;***

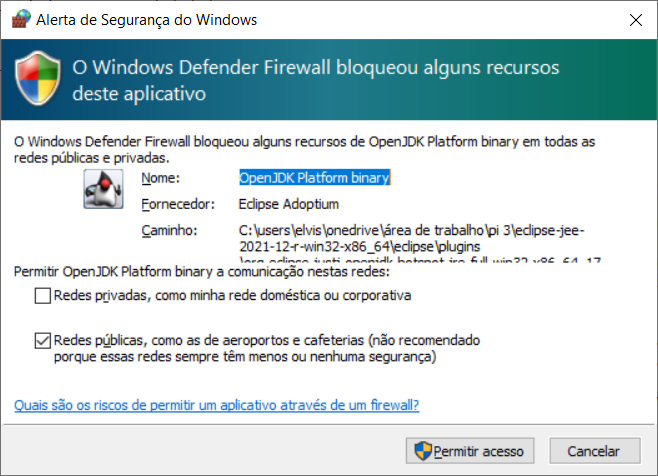
******

* Acessar o banco de dados temperatura\_apirest

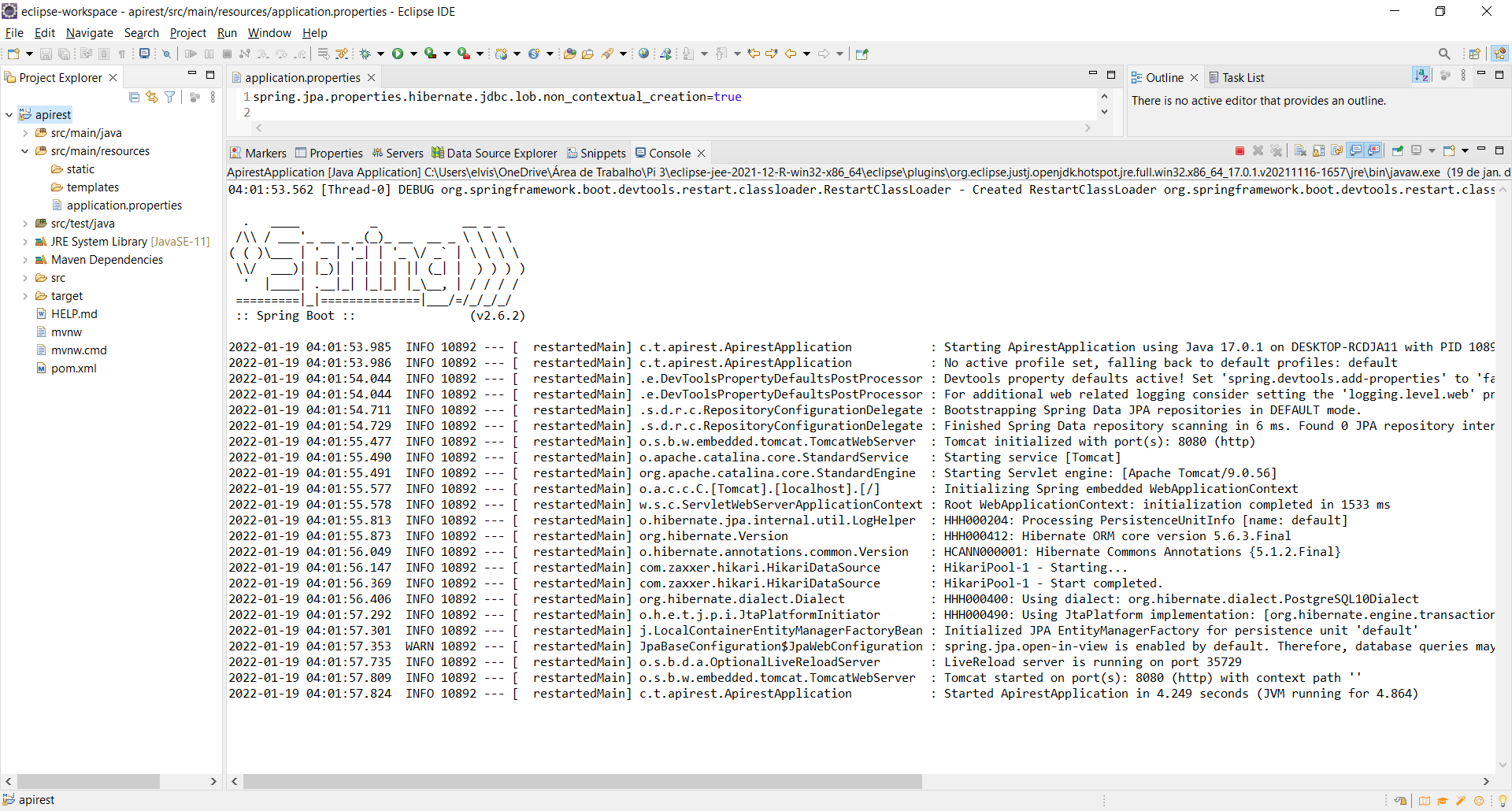
****

1. **Testando a conexão da aplicação com o banco de dados**

* Botão direito em ApiRestApplication.java
* Run as > 1 Java Application
* Clicar em **Permitir Acesso**



* Tela de Conexão Estabelecida



* Verificando a criação das tabelas geradas no banco de dados ***PostgresSQL***
* Digitar o comando: ***\dt***

