

# Aula 03 - JPA

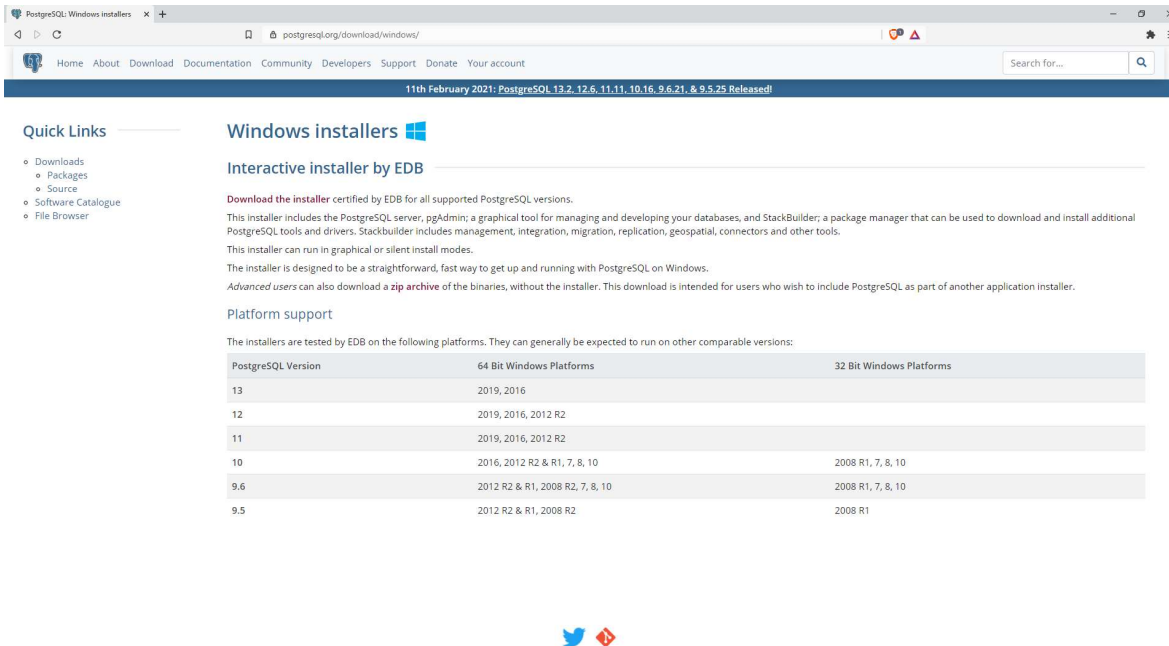
Prof. Glauco Todesco

# Instalação – PostgreSQL – 12.6

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

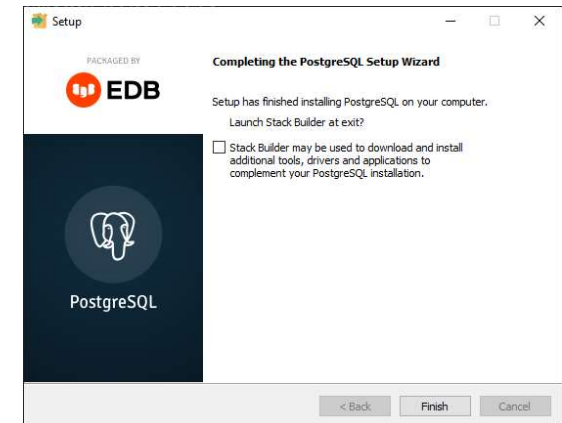
Password: 1234567

Não precisa instalar o Stack Builder



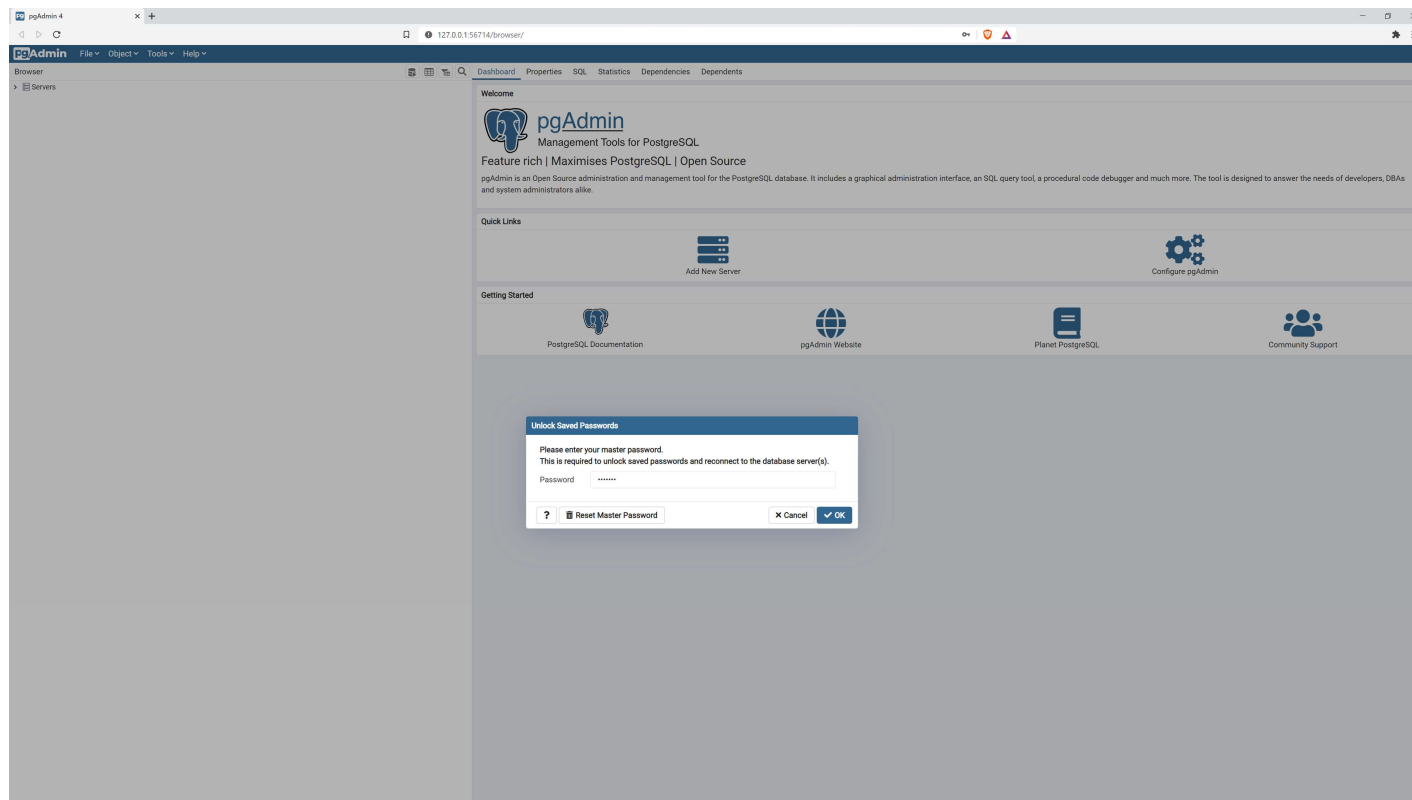
The screenshot shows the PostgreSQL Windows installers page. It features a navigation bar with links like Home, About, Download, Documentation, Community, Developers, Support, Donate, and Your account. A search bar is also present. The main content area is titled 'Windows installers' and includes a section for 'Interactive installer by EDB'. This section describes the installer, which includes the PostgreSQL server, pgAdmin, and StackBuilder. It also mentions that the installer can run in graphical or silent install modes and is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows. A 'Platform support' section follows, stating that the installers are tested by EDB on specific platforms and can generally be expected to run on other comparable versions. Below this is a table with three columns: PostgreSQL Version, 64 Bit Windows Platforms, and 32 Bit Windows Platforms.

PostgreSQL Version	64 Bit Windows Platforms	32 Bit Windows Platforms
13	2019, 2016	
12	2019, 2016, 2012 R2	
11	2019, 2016, 2012 R2	
10	2016, 2012 R2 & R1, 7, 8, 10	2008 R1, 7, 8, 10
9.6	2012 R2 & R1, 2008 R2, 7, 8, 10	2008 R1, 7, 8, 10
9.5	2012 R2 & R1, 2008 R2	2008 R1



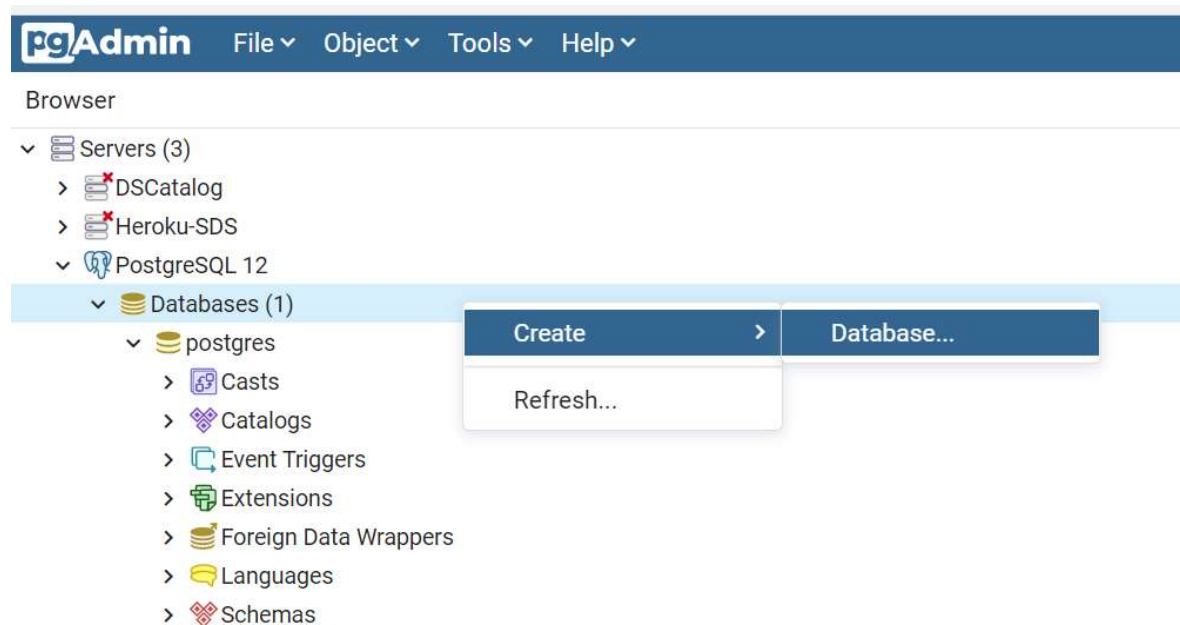
# PostgreSQL 12.6

- Execute o pgAdmin4 (password: 1234567)



# Crie um database

- Botão Direito em DataBase -> Create: jpa-samples



# Criar um projeto Maven

- Ctrl + Shif + P
- Maven Project
- QuickStart JDK 8
  - Groupid: br.facens.jp
  - ArtifactId: exemplos
  - Enter e Yes.
- Abra o projeto:
  - cd exemplos
  - code .

# Adicionar Dependências (pom.xml)

```
<dependency>  
  <groupId>org.postgresql</groupId>  
  <artifactId>postgresql</artifactId>  
  <version>42.2.19</version>  
  <scope>runtime</scope>  
</dependency>
```

```
<dependency>  
  <groupId>org.eclipse.persistence</groupId>  
  <artifactId>org.eclipse.persistence.jpa</artifactId>  
  <version>2.7.8</version>  
</dependency>
```

# Configurar: persistence.xml

- Crie a pasta resource dentro da pasta main.
- Crie a pasta META-INF dentro da pasta resource.
- Crie dentro da pasta META-INF o arquivo **persistence.xml**

# persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
```

```
  <persistence-unit name="PU_SAMPLES" transaction-type="RESOURCE_LOCAL">
```

```
    <properties>
```

```
      <property name="javax.persistence.schema-generation.database.action" value="create"/>
```

```
      <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/jpa-samples"/>
```

```
      <property name="javax.persistence.jdbc.user" value="postgres"/>
```

```
      <property name="javax.persistence.jdbc.password" value="1234567"/>
```

```
      <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver"/>
```

```
    </properties>
```

```
  </persistence-unit>
```

```
</persistence>
```



# Crie uma classe Book

```
package br.facens.jpa.example01;
```

```
public class Book {  
    private Long id;  
    private String author;  
    private double price;  
    private String title;  
}
```

**Adicione gets, sets, toString, hashCode e equals**

# Transformando a Classe em Entidade

- Marque a classe como @Entity
- Marque o atributo id como @Id
- Implemente a interface Serializable.

# Inserindo uma linha no Banco:

- Crie a classe InsertBook com o método main:
- Dentro do main:
  1. Crie um EntityManagerFactory:

```
EntityManagerFactory factory = Persistence.createEntityManagerFactory("PU_SAMPLES");
```

2. Crie um EntityManager:

```
EntityManager em = factory.createEntityManager();
```

# Inserindo uma linha no Banco:

## 3. Crie um objeto Book com Dados:

```
Book b1 = new Book();  
b1.setId(1l);  
b1.setAuthor("Glauco Todesco");  
b1.setPrice(10.22);  
b1.setTitle("The Art of Java Programing");
```

## 4. Faça a persistência do objeto através do EntityManager:

```
em.persist(b1);
```

# Inserindo uma linha no Banco:

5. Para sincronizar os objetos com o banco de dados é necessário ter uma transação e um commit:

```
em.getTransaction().begin();  
em.getTransaction().commit();
```

6. Feche todas as conexões:

```
em.close();  
factory.close();
```

# Versão Final:

```
public class InsertBook {  
    public static void main(String[] args) {  
  
        EntityManagerFactory factory = Persistence.createEntityManagerFactory("PU_SAMPLES");  
  
        EntityManager em = factory.createEntityManager();  
  
        Book b1 = new Book();  
        b1.setId(11);  
        b1.setAuthor("Glauco Todesco");  
        b1.setPrice(10.22);  
        b1.setTitle("The Art of Java Programming");  
  
        em.persist(b1);  
  
        em.getTransaction().begin();  
        em.getTransaction().commit();  
  
        em.close();  
        factory.close();  
    }  
}
```

# Execução:

- Ao executar o arquivo teremos um erro (teste)
- É necessário adicionar a entidade no persistence.xml

```
<persistence-unit name="PU_SAMPLES" transaction-type="RESOURCE_LOCAL">
```

```
<class>br.facens.jpa.example01.Book</class>
```

```
<properties>
```

# Veja no Banco de Dados o resultado:

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, with the 'public' schema expanded and the 'book' table selected. A context menu is open over the 'book' table, with the 'View/Edit Data' option selected, which has further opened a sub-menu showing 'All Rows' as the chosen view. The main panel displays the SQL query editor with the query: `SELECT * FROM public.book ORDER BY id ASC`. Below the query editor, a table of results is shown, containing one row of data.

	price		title
	double precision		character varying (255)
1	10.22	Glauco Todesco	The Art of Java Programming



# Executando um Select com JQPL

- JPQL = Java Persistence Query Language

```
EntityManagerFactory factory = Persistence.createEntityManagerFactory("PU_SAMPLES");

EntityManager em = factory.createEntityManager();

List <Book> list = em.createQuery("SELECT o FROM Book o", Book.class).getResultList();

for(Book b : list){
    System.out.println(b);
}

em.close();
factory.close();
```