

---

---

# Postgresql e Aplicação Web: instalação e uso

Pedro Losco Takecian  
Leonardo Tadashi Kamaura  
Data Group - IME/USP

---



# 1. Instalação

→ **Ubuntu 22.04**

\$ sudo apt install postgresql

→ **Acesso local**

Instalação padrão provê apenas acesso local.



## 2. Utilização

→ **Acesso via cliente psql**

```
$ sudo -i -u postgres
```

```
$ psql
```

→ **Criação de usuário do BD**

```
# CREATE USER testeusr WITH ENCRYPTED PASSWORD 'testeusr';
```

→ **Criação do BD**

```
# CREATE DATABASE teste WITH OWNER testeusr;
```

→ **Usar o banco criado**

```
# \c teste
```



### 3. Permissões

→ **/etc/postgresql/<versão>/main/pg\_hba.conf**

local	all	all	
		scram-sha-256	
host	all	all	0.0.0.0/0
	scram-sha-256		
host	all	all	::/0
		scram-sha-256	

→ **/etc/postgresql/<versão>/main/postgresql.conf**

listen\_addresses = '\*'

→ **Reinicializar o serviço do Postgresql**

\$ sudo systemctl restart postgresql



## 4. Estruturas

→ **Acesso como usuário testeusr**

```
$ psql -U testeusr teste
```

→ **Criação da tabela "pessoa"**

```
# CREATE TABLE pessoa (  
    id serial,  
    nusp character varying(10) NOT NULL,  
    nome character varying(255) NOT NULL,  
    PRIMARY KEY(id),  
    UNIQUE(nusp)  
);
```

→ **Inserção na tabela "pessoa"**

```
# INSERT INTO pessoa(nusp, nome) VALUES ('123456', 'Pessoa teste');
```



## 4. Estruturas

### → Criação da tabela "perfil"

```
# CREATE TABLE perfil (  
    id serial,  
    codigo character varying(10) NOT NULL,  
    descricao character varying(10) NOT NULL,  
    PRIMARY KEY(id),  
    UNIQUE(codigo)  
);
```

### → Inserção na tabela "perfil"

```
# INSERT INTO perfil(codigo, descricao) VALUES ('EST', 'Estudante');
```



## 4. Estruturas

### → Criação da tabela "pessoaperfil"

```
# CREATE TABLE pessoaperfil (  
    id serial,  
    pessoa_id integer NOT NULL,  
    perfil_id integer NOT NULL,  
    data_inicio timestamp with time zone NOT NULL,  
    data_fim timestamp with time zone NOT NULL,  
    PRIMARY KEY(id),  
    UNIQUE(pessoa_id, perfil_id, data_inicio, data_fim),  
    FOREIGN KEY(pessoa_id) REFERENCES pessoa(id),  
    FOREIGN KEY(perfil_id) REFERENCES perfil(id)  
);
```

### → Inserção na tabela "pessoaperfil"

```
# INSERT INTO pessoaperfil(pessoa_id, perfil_id, data_inicio, data_fim) VALUES  
(1, 1, '2023-05-01 14:00:00 +03:00', '2023-05-30 10:00:00 +03:00');
```



## 5. Functions

→ **Exemplo**

```
CREATE FUNCTION lista_pessoaperfil()  
RETURNS SETOF pessoaperfil AS $$  
    SELECT * FROM pessoaperfil  
    ORDER BY data_inicio;  
$$ LANGUAGE SQL;
```

→ **Utilização**

```
SELECT lista_pessoaperfil();
```





## 6. PostgreSQL GUI: pgAdmin

→ **Instalação**

<https://www.pgadmin.org/download/pgadmin-4-apt/>



## 7. Demo: acesso ao BD

### → Instalação

```
$ sudo apt install openjdk-17-jdk unzip  
$ wget https://services.gradle.org/distributions/gradle-8.1.1-bin.zip  
$ unzip gradle-8.1.1-bin.zip  
$ sudo mkdir -p /opt/gradle  
$ sudo mv gradle-8.1.1 /opt/gradle/  
$ cd; vim .profile
```

```
    if [ -d "/opt/gradle/gradle-8.1.1/bin" ] ; then  
        PATH="/opt/gradle/gradle-8.1.1/bin:$PATH"  
    fi
```

```
$ gradle -v
```

Obs: Se necessário, ajustar a variável de ambiente JAVA\_HOME.



## 7. Demo: acesso ao BD

### → Rodar o demo


```
$ cd  
$ wget https://data.ime.usp.br/mac0350-2023/demo.zip  
$ cd demo  
$ gradle bootRun  
$ Acessar via http://localhost:8080/pessoas
```

### → Tecnologias empregadas

```
$ Gradle  
$ Spring Boot  
$ Spring Web  
$ Thymeleaf  
$ JDBC API
```

https://start.spring.io

Meet the Spring team this August at SpringOne.

 **spring** initializr

Project

☒ Gradle - Groovy

☐ Gradle - Kotlin

☐ Maven

Language

☒ Java

☐ Kotlin

☐ Groovy

Spring Boot

☐ 3.1.1 (SNAPSHOT)

☒ 3.1.0

☐ 3.0.8 (SNAPSHOT)

☐ 3.0.7

☐ 2.7.13 (SNAPSHOT)

☐ 2.7.12

Project Metadata

Group

com.example

Artifact

demo

Name

demo

Description

Demo project for Spring Boot

Package name

com.example.demo

Packaging

☒ Jar

☐ War

Java

☐ 20

☒ 17

☐ 11

☐ 8

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Thymeleaf

TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

JDBC API

SQL

Database Connectivity API that defines how a client may connect and query a database.

Spring Boot DevTools



DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

PostgreSQL Driver

SQL

A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.



GENERATE

CTRL + G

EXPLORE

CTRL + SPACE

SHARE...