

# Tecnologias de Banco de Dados

Msc. Eliezio Soares  
eliezio.soares@ifrn.edu.br



**INSTITUTO FEDERAL**  
Rio Grande do Norte

Campus  
Carrais Novos

# Preparação do ambiente

- Criar Máquina virtual
  - Debian 9.5
  - Disco  $\geq$  10GB
  - Iso disponível por pen drive ou em:
    - <https://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware/9.5.0+nonfree/amd64/iso-dvd/>

# Preparação do ambiente

- Adicionar repositório apt do PostgreSQL:

```
bash -c "echo deb http://apt.postgresql.org/pub/repos/apt/ stretch-pgdg main >>  
/etc/apt/sources.list.d/pgdg.list"
```

- Importar a chave de assinatura:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc  
apt-key add <file>
```

- Atualizar a lista de pacotes:

```
apt -y update
```

- Instalar PostgreSQL:

```
apt-get install postgresql-10 postgresql-contrib libpq-dev
```



# Configuração do Banco - Usuário Padrão

- Usuário `postgres`
  - Durante o processo e instalação do serviço, uma conta de usuário local é criada para a execução do mesmo.
    - Esse usuário postgres é criado sem senha.
    - Pode ser alterada com:

```
passwd postgres
```
  - É criada uma conta para o postgresSQL também, um usuário do SGBD.
    - Esse usuário postgres é criado sem senha.
    - Pode ser alterada com o comando **SQL**:

```
ALTER USER postgres WITH PASSWORD 'abc123...';
```

# DIRETÓRIO DE DADOS

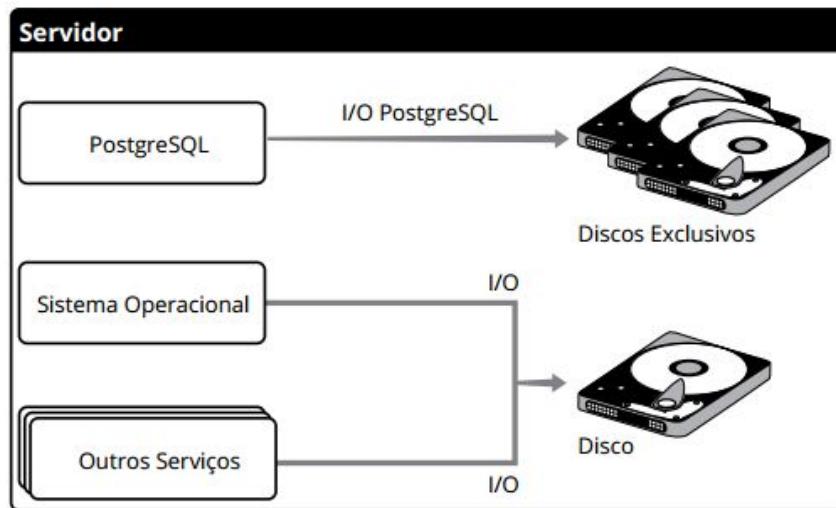


# Configuração do Banco - Diretório de Dados

- Tradicionalmente, os arquivos de dados e configurações usados por um cluster de banco de dados, são armazenados juntos, dentro de um diretório de dados - **PGDATA**.
  - Um lugar comum para a inicialização é:
    - `/var/lib/pgsql/data`
  - Múltiplos clusters, gerenciados por diferentes instâncias de servidor, podem existir na mesma máquina.
- O diretório PGDATA contém vários subdiretórios e arquivos de controle e configuração.

# Configuração do Banco - Diretório de Dados

- É recomendável separar o PGDATA em uma partição exclusiva:
  - Evita-se concorrência com operações de I/O do S.O e outras aplicações;
  - Melhor desempenho;
  - Manutenção facilitada;





# Configuração do Banco - PGDATA

- Vamos simular um disco exclusivo em nossa máquina virtual:

- Adicionar novo disco SATA a VM (alocação dinâmica);
- Inicializar VM;
- Instalar GParted;
- Criar tabela de partições GPT para o novo disco;
- Criar partição

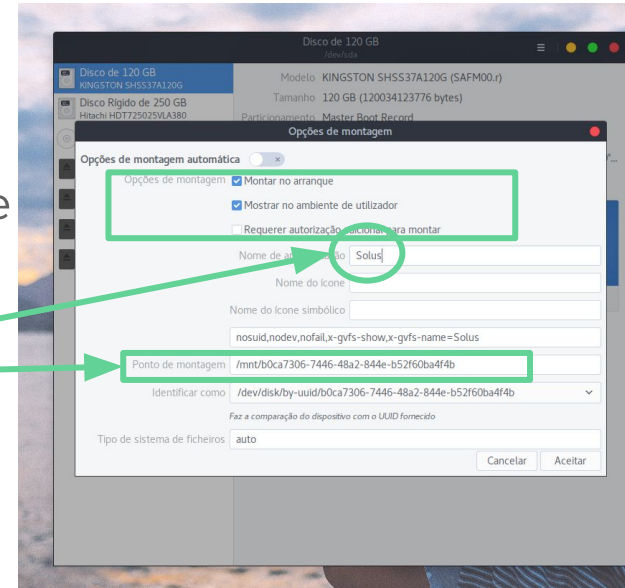
- Utilizar utilitário Discos para configurar as opções de montagem.

- Criar diretório para montagem do novo disco:

■ /media/db

- Rótulo:

■ db



# Configuração do Banco - PGDATA

- Para que o serviço funcione adequadamente é preciso que o usuário responsável tenha permissões em /media/db:

```
Chown -R postgres /media/db/
```

- Esse comando torna o usuário postgres proprietário do diretório

# Configuração do Banco - Variáveis de Ambiente

Para tornar prática a execução dos programas e a definição da área de dados, é uma boa prática definir esses caminhos nas variáveis de ambiente.

1. Conectar com usuário postgres:

```
su - postgres
```

2. Abrir o arquivo `.bashrc` em um editor:

```
nano ~/.bashrc
```

3. Inserir as variáveis `PATH` e `PGDATA`:

```
PATH=$PATH:/usr/lib/postgresql/10/bin:$HOME/bin
```

```
PGDATA=/media/db/data/
```

```
Export PATH PGDATA
```

Atenção para o subdiretório.  
Crie-o!

# Configuração do Banco - Variáveis de Ambiente

Para tornar prática a execução dos programas e a definição da área de dados, é uma boa prática definir esses caminhos nas variáveis de ambiente.

1. Abrir o arquivo `profile` em um editor:

```
nano /etc/profile
```

2. Inserir a variável `PGDATA`:

```
PGDATA="/media/db/data/"
```

```
Export PGDATA
```

3. Acrescentar o path do postgresql a variável `PATH`:

```
PATH="(...):/usr/lib/postgresql/10/bin"
```

# Configuração do Banco - Variáveis de Ambiente

4. Recarregar o .bashrc:

```
source .bashrc
```

5. Executar initdb (está referenciado pelo PATH definido):

```
initdb
```

**Explore os diretórios de instalação e o de dados !!!**



# INICIALIZAÇÃO DO POSTGRESQL

# Configuração do Banco - Iniciar o Serviço

- Uma opção é chamar o executável postgres:

```
postgres -D /media/db/data
```

- -D define o diretório de dados a ser utilizado.

```
postgres
```

- Se a variável PGDATA foi corretamente definida, não há necessidade desse parâmetro.

- A forma recomendável (mais adequada) de iniciar o banco é através do executável `pg_ctl`:

```
pg_ctl start
```

# Configuração do Banco - Parar o Serviço

- Para encerrar o processo pode se utilizar o comando kill do Linux:
  - Para isso é necessário ter o Id do Processo (PID) principal do PostgreSQL;
  - Pode se utilizar o comando ps do linux, filtrando pelos processos postgres:

```
ps -ef | grep postgres
```

- Pode se consultar o arquivo postmaster.pid:

```
cat /media/db/postmaster.pid
```

```
postgres@pg01:~$ ps -ef | grep postgres
postgres 815 1 0 22:59 ? S 0:00 /usr/local/pgsql/bin/postmaster -D /d
b/data
postgres 927 815 0 22:59 ? Ss 0:00 \_ postgres: logger process
postgres 948 815 0 22:59 ? Ss 0:00 \_ postgres: checkpointer process
postgres 950 815 0 22:59 ? Ss 0:00 \_ postgres: writer process
postgres 951 815 0 22:59 ? Ss 0:00 \_ postgres: wal writer process
postgres 952 815 0 22:59 ? Ss 0:00 \_ postgres: archiver process
postgres 953 815 0 22:59 ? Ss 0:00 \_ postgres: stats collector
process
postgres@pg01:~$
```

```
postgres@pg01:~$ cat /db/data/postmaster.pid
815
/db/data
1396317578
5432
/tmp
*
5432001 0
postgres@pg01:~$
```



# Configuração do Banco - Parar o Serviço

- O kill envia um (dentre três) sinais possíveis para o processo:
  - **TERM:** Smart Shutdown - O banco não aceitará novas conexões e aguardará as existentes terminarem.  

```
$ kill -TERM 1010
```
  - **INT:** Fast Shutdown - O banco não aceitará conexões e enviará um sinal TERM para todos os processos backends fecharem (abortam as transações);  

```
$ kill -INT 1010
```
  - **QUIT:** Immediate Shutdown - O processo envia um QUIT para todos os backends terminarem imediatamente;  

```
$ kill -QUIT 1010
```

# Configuração do Banco - Parar o Serviço

- A forma recomendável (mais adequada) de parar o banco é através do executável `pg_ctl` (o pid é dispensável):

```
$ pg_ctl stop
```

- É possível determinar o modo de parada passando -m e a inicial do modo desejado (smart, fast, immediate):

```
$ pg_ctl stop -ms
```

```
$ pg_ctl stop -mf
```

```
$ pg_ctl stop -mi
```

# Configuração do Banco - Teste

- Verificar status utilizando o `pg_ctl`:  
`pg_ctl status`
- Reiniciar o serviço (stop - start) recarregando definições, memória, configurações:  
`pg_ctl restart`
- Recarregar as configurações do banco sem parar o serviço:  
`pg_ctl reload`

# Configuração do Banco - Teste

- Primeira conexão no servidor...
- `$ psql -h localhost -p 5432 -d curso -U aluno`
  - h é o servidor (host)
  - p a porta
  - d a base (database)
  - U o usuário

Comando	Função
<code>psql -U postgres -h localhost</code>	Abre conexão com o servidor especificando usuário e host.
<code>su - postgres</code> <code>psql</code>	Altera o usuário local para postgres Inicia conexão com o servidor local utilizando o usuário de domínio.
<code>psql -U postgres NomeBanco</code>	Inicia conexão com o servidor em um banco específico.
<code>\?</code>	Lista comandos psql;
<code>\h</code>	Lista todos os comandos sql;
<code>\du</code>	Lista usuários
<code>ALTER USER postgres WITH password 'exemplo';</code>	Altera a senha de um usuário do SGBD;

# Bibliografia Utilizada

CAIUT, Fábio. Administração de banco de dados. 1ª Edição. Rio de Janeiro. RNP/ESR, 2015.

PostgreSQL 10.5

Documentation. Disponível em:  
<<https://www.postgresql.org/docs/10/static/index.html>>. Acesso em 09 Ago. 2018.

