

```
pacman::p_load(RPostgres, keyring)

tryCatch({
  drv <- Postgres()
  print("Connecting to Database...")
  connec <- dbConnect(drv,
    dbname = "dvdrental",
    host = Sys.getenv("psql_eq_ip"),
    port = Sys.getenv("psql_eq_port"),
    user = key_list("psql-eq")[1,2],
    password = key_get(service = "psql-eq",
      username = key_list("psql-eq")[1,2]))

  print("Database Connected!")
},
error=function(cond) {
  print(cond)
  print("Unable to connect to Database.")
})
```

```
[1] "Connecting to Database..."
[1] "Database Connected!"
```

```
df <- dbGetQuery(connec, "SELECT first_name, last_name FROM actor")

head(df)
```

	first_name	last_name
1	Penelope	Guinness
2	Nick	Wahlberg
3	Ed	Chase
4	Jennifer	Davis
5	Johnny	Lollobrigida
6	Bette	Nicholson

```
dbDisconnect(connec)
```

Preparar WSL y PostgreSQL en Windows

Ahora Windows incluye la opción de habilitar una versión de linux virtual que convive con el ambiente Windows, la característica que denomina *Windows Subsystem Linux (WSL)*. Hay que habilitarlo desde la panel de control como *características de Windows* como se ilustra en la [?@fig-wsl](#), o mediante comandos desde una ventana de **Powershell** de la manera siguiente.

```
wsl --install
wsl --version
wsl --install -d ubuntu
```

Una vez activado **WSL**, tendrás, en este ejemplo, un *linux Ubuntu* en marcha y para acceder a su consola de sistema bastará con que escribas **wsl** en una terminal de **Powershell**, que puede ser la misma que usaste para instalar todo.

En una ventana de *Powershell* escribir

```
sudo apt-get install postgresql postgresql-contrib
```

Para iniciar tu interacción con PostgreSQL inicialmente necesitará entrar a la interfaz del sistema con el siguiente comando.

```
sudo -u postgres psql

sudo -u postgres createuser <username>

sudo -u postgres createdb <dbname>

$ sudo -u postgres psql
psql=# alter user <username> with encrypted password '<password>';
grant all privileges on database <dbname> to <username> ;
```