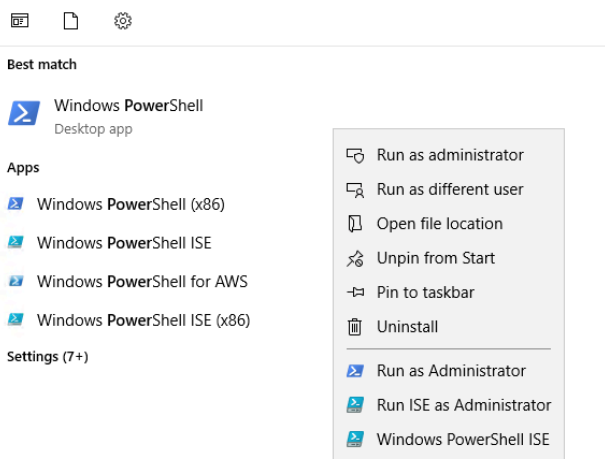


# Instalação do Redis

1. Acessar o site do Chocolatey <https://chocolatey.org/install>

2. Executar o PowerShell como Admin



3. Executar no powershell como administrador

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -  
bor 3072; iex ((New-Object  
System.Net.WebClient).DownloadString('https://chocol  
atey.org/install.ps1'))
```

4. O chocolatey será instalado. Em seguida, instale o redis via PowerShell com o choco:

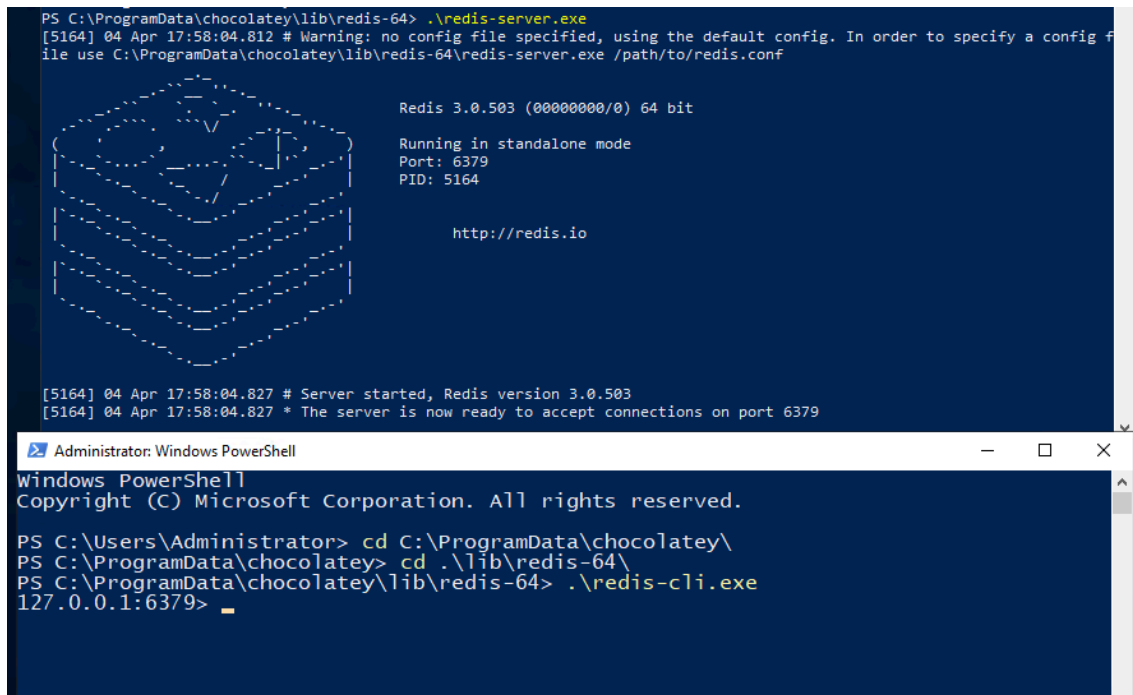
```
choco install redis-64
```

5. Iniciar em um terminal o Redis Server

```
cd C:\ProgramData\chocolatey\lib\redis-64  
.\redis-server.exe
```

6. Iniciar outro power o RedisClient para testar

```
cd C:\ProgramData\chocolatey\lib\redis-64  
.\redis-client.exe
```



```
PS C:\ProgramData\chocolatey\lib\redis-64> .\redis-server.exe
[5164] 04 Apr 17:58:04.812 # Warning: no config file specified, using the default config. In order to specify a config file use C:\ProgramData\chocolatey\lib\redis-64\redis-server.exe /path/to/redis.conf

Redis 3.0.503 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 5164

http://redis.io

[5164] 04 Apr 17:58:04.827 # Server started, Redis version 3.0.503
[5164] 04 Apr 17:58:04.827 * The server is now ready to accept connections on port 6379

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> cd C:\ProgramData\chocolatey\
PS C:\ProgramData\chocolatey> cd .\lib\redis-64\
PS C:\ProgramData\chocolatey\lib\redis-64> .\redis-cli.exe
127.0.0.1:6379>
```

## Conectando com Node.js

Um bom início para conectar via uma linguagem de programação é através do node.js. A partir do powershell, instale o node.js da seguinte forma

```
choco install nodejs
```

Em seguida, atualize as variáveis de ambiente com o comando:

```
refreshenv
```

Agora abra o Visual Studio Code (ou instale se o mesmo não tiver instalado e crie um arquivo com o seguinte conteúdo (exemplo my-redis-sample.js)):

```
const redis = require("redis");
const client = redis.createClient();

client.on("error", function(){
  console.error(error);
});

client.set("key", "value", redis.print);
client.get("key", redis.print);
```

Agora, acesse o terminal do Visual Studio Code e instale a dependência com o gerenciador de pacotes do node:

```
npm install node
```

Em seguida, você pode executar o programa da seguinte forma:

```
node my-redis-sample.js
```

```
c > Users > Administrator > JS redis.js > ...
1  const redis = require("redis");
2  const client = redis.createClient();
3
4  client.on("error", function(){
5    console.error(error);
6  });
7
8  client.set("key", "value", redis.print);
9  client.get("key", redis.print);
```

DEBUG CONSOLE PROBLEMS OUTPUT TERMINAL 1: node

```
PS C:\Users\Administrator> npm install
npm WARN saveError ENOENT: no such file or directory, open 'C:\Users\Administrator\package.json'
npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\Administrator\package.json'
npm WARN Administrator No description
npm WARN Administrator No repository field.
npm WARN Administrator No README data
npm WARN Administrator No license field.

audited 11 packages in 1.3s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities

PS C:\Users\Administrator> node .\redis.js
Reply: OK
```