

- [chocolatey](#)
  - [Que é?](#)
  - [Requisitos](#)
  - [Instalación .](#)
  - [Comandos](#)
- [Casos de uso](#)
  - [Búsqueda dun paquete.](#)
  - [Instalación dun paquete.](#)
  - [Actualización dun paquete.](#)
  - [Información dun paquete](#)
  - [Desinstalación](#)
  - [Exportar a configuración.](#)
- [Anexos](#)
  - [Comandos de choco](#)
  - [Exemplos](#)
    - [Outras opcións.](#)

# chocolatey

---

## Que é?

Chocolatey é un xestor de paquetes e instalador a nivel de máquina para paquetes de software, creado para a plataforma Windows.

O motor de execución utiliza a infraestrutura de empaquetado de NuGet e Windows PowerShell para proporcionar unha ferramenta de automatización e instalar software en máquinas Windows e foi deseñado para simplificar o proceso desde a perspectiva do usuario.

O nome é unha extensión nun xogo de palabras de NuGet (de "nougat" en español, turrón) porque todo o mundo ama Chocolatey (de "nougat chocolat" en español, turrón de chocolate).

## Requisitos

- Windows 7 ou superior / Windows 2003+ (Server Core también, pero no Windows Nano Server)
- Windows PowerShell v2 + (aún no PowerShell Core, también conocido como PowerShell 6)
- .NET Framework 4.x +
- 2 GB de RAM

### [Máis información](#)

## Instalación .

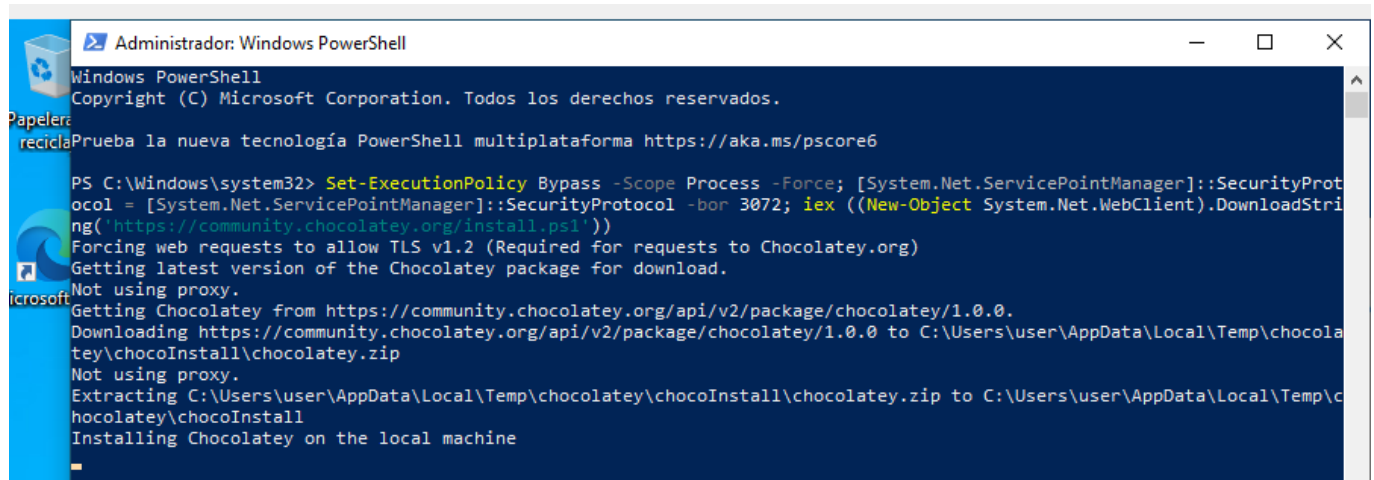
A instalación permite configuralo de dúas maneiras específicas:

- Individual, para un uso "doméstico" o cal empregará os repositorios oficiais.
- Corporativo, empregando repositorios locais no entor de traballo. Estes poden ofrecerse con diferentes tecnoloxías como son: Ansible, Chef, Puppet, PS DSC...

Para instalar a ferramentas teremos que abrir un PowerShell en modo administrador e executar a seguinte liña:

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

Máis información consultar a documentación



## Comandos

Choco ten un conxunto importante de comandos os cales podemos consultar executando

```
choco comand -help
```

A lista completa de comandos é (Fonte: [Documentacion](#)):

- search - searches remote or local packages (alias for list)
- list - lists remote or local packages
- find - searches remote or local packages (alias for search)
- info - retrieves package information. Shorthand for choco search pkgname --exact --verbose
- install - installs packages from various sources
- pin - suppress upgrades for a package
- outdated - retrieves packages that are outdated. Similar to upgrade all --noop
- upgrade - upgrades packages from various sources
- uninstall - uninstalls a package
- pack - packages nuspec, scripts, and other Chocolatey package resources into a nupkg file
- push - pushes a compiled nupkg to a source
- new - generates files necessary for a chocolatey package from a template
- source - view and configure default sources
- sources - view and configure default sources (alias for source)
- config - Retrieve and configure config file settings

- feature - view and configure choco features
- features - view and configure choco features (alias for feature)
- setapikey - retrieves, saves or deletes an apikey for a particular source (alias for apikey)
- apikey - retrieves, saves or deletes an apikey for a particular source
- unpackself - re-installs Chocolatey base files
- support - provides support information
- help - displays top level help information for choco
- export - exports list of currently installed packages
- template - get information about installed templates
- templates - get information about installed templates (alias for template)
- download - downloads packages - optionally internalizing all remote resources
- synchronize - synchronizes against system installed software - generates missing packages
- sync - synchronizes against system installed software - generates missing packages
- optimize - optimizes installation, reducing space usage
- convert - converts packages from one type to another type

## Casos de uso

---

### Búsqueda dun paquete.

Podemos consultar a xuda para o comando **search**

```
choco search -?
```

Neste comando destan as opcións:

- -d : Debug, que mostra todas as mensaxes.
- -y ou --yes : Que confirma todas as preguntas co valor "yes"
- --timeout : Configura o tempo máximo de resposta dun comando. Exemplo **--timeout, --execution-timeout=VALUE**

```
choco search hugo
```

Con este comando podemos consultar o **id** de cada aplicación no caso anterior retorna 5 resultados.

```
PS C:\Windows\system32> choco search hugo
Chocolatey v1.0.0
hugo 0.95.0 [Approved] Downloads cached for licensed users
hugo-extended 0.95.0 [Approved] Downloads cached for licensed users
doxybook2 1.4.0 [Approved] Downloads cached for licensed users
pcwrunas 0.4.0.20161129 [Approved] Downloads cached for licensed users
zola 0.15.3 [Approved] Downloads cached for licensed users
5 packages found.
```

### Instalación dun paquete.

A sintaxe máis sinxela para instalar será:

```
choco install idpaquete
```

No exemplo anterior:

```
choco install hugo
```

Vainos preguntar se desexamos executar o script polo cal será preciso escribir Y (yes). Unha alternativa para evitar estas cuestións e confirmalo por defecto. Isto sería:

```
choco install hugo -y
```

## Actualización dun paquete.

```
choco upgrade <pkg|all> [ ] [<options/switches>]
```

- [Documentación](#)

Teremos que empregar o comando **upgrade** o cal podemos actualizar todas as aplicacións ou unha concreta.

Exemplo para actualizar a aplicación hugo:

```
choco upgrade hugo
```

Exemplo para actualizar todas as aplicacións:

```
choco upgrade all
```

Exemplo actualizar varias aplicacións nunha liña:

```
choco upgrade notepadplusplus googlechrome atom 7zip
```

Exemplo, actualizar todo menos unhas aplicacións concretas:

```
choco upgrade all --except="skype,conemu"
```

Exemplo, actualizar a unha versión en concreto:

```
choco upgrade nodejs.install --version 0.10.35
```

## Información dun paquete

```
choco info chocolatey
```

```
choco info googlechrome
```

```
choco info powershell
```

## Desinstalación

```
choco uninstall hugo
```

## Exportar a configuración.

Unha ferramenta útil é exportar as aplicacións instaladas nun equipo, isto vainos permitir aplicar a mesma instalación noutro equipo. O ficheiro que xera por defecto a aplicación é de tipo **XML**.

```
choco export [<options/switches>]
```

Exemplo:

```
choco export "packages.config"
```

O contido do ficheiro será:

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="chocolatey" />
  <package id="hugo" />
  <package id="winrar" />
</packages>
```

## Anexos

### Comandos de choco

Command	Description
<a href="#">list</a>	lists remote or local packages
<a href="#">search</a>	searches remote or local packages
<a href="#">info</a>	retrieves package information
<a href="#">install</a>	installs packages from various sources
<a href="#">pin</a>	pins a package to suppress upgrades
<a href="#">outdated</a>	retrieves packages that are outdated
<a href="#">upgrade</a>	upgrades packages from various sources
<a href="#">uninstall</a>	uninstalls a package

See also the [Chocolatey Command Reference](#) for a complete list.

### Exemplos

Command	Description
<b>Búsqueda de paquetes</b>	
<code>choco list</code>	List all chocolatey packages
<code>choco search zip</code>	Search packages mentionning "zip"

Command	Description
<code>choco search --by-tag compression</code>	Search packages by their tags
<code>choco search --by-id-only zip</code>	Search packages with "zip" in the package name
<code>choco search --order-by-popularity zip</code>	Filter and sort by package results by popularity
<code>choco search --approved-only zip</code>	Only return approved packages
<code>choco info 7zip</code>	Get information about 7zip package

### Instalación

<code>choco install 7zip</code>	Install
<code>choco install 7zip --install-directory=P:\7z</code>	Install to a specific directory

### Mantenimiento

<code>choco list --localonly</code>	Listar todos os paquetes instalados
<code>choco outdated</code>	List upgradable packages
<code>choco upgrade all -y</code>	Upgrade all packages

### Pinning

<code>choco pin list</code>	List pinned packages
<code>choco pin add --name 7zip</code>	Suppress upgrades for 7zip
<code>choco pin remove --name 7zip</code>	Supress upgrades supression for 7zip 😊

Prefered search method: `choco search --by-id-only --order-by-popularity --approved-only searchstring`

Outras opcións.

Switches	Description
<code>--help</code>	Prints out the help menu.
<code>--install-directory=VALUE</code>	Install Directory Override
<code>-y, --yes</code>	Confirm all prompts.
<code>-f, --force</code>	Force the behavior.
<code>--what-if</code>	Don't actually do anything, simulate.

Switches	Description
<code>--execution-timeout=VALUE</code>	The time to allow a command to finish.
<code>--fail-on-standard-error</code>	Fail on standard error output (stderr).
<code>--use-system-powershell</code>	Execute PowerShell using an external process.
<code>--debug</code>	Show debug messaging.
<code>--verbose</code>	Show verbose messaging.
<code>--trace</code>	Show trace messaging.
<code>--log-file=VALUE</code>	Log File to output to in addition to regular loggers.
<code>--no-color</code>	Do not show colorization in output.
<code>--limit-output</code>	Limit the output to essential information.
<code>--no-progress</code>	Do not show download progress percentages.
<code>--allow-unofficial-build</code>	Allow the use of an unofficial build.

Podes consultar máis exemplos de uso na [Documentación](#)