

# Manual de instalación de ROS 2 Humble Hawksbill en Windows 10

## !! NO FUNCIONA !!

Seguimos la siguiente guía como referencia: <https://docs.ros.org/en/humble/Installation.html>

Y en concreto la dedicada a Windows: <https://docs.ros.org/en/humble/Installation/Windows-Install-Binary.html>

### Prerrequisitos

1. Instalar Chocolatey
2. Instalar Python 3.8.3
3. Instalar Visual C++ Redistributables
4. Instalar OpenSSL
5. Instalar Visual Studio
6. Instalar OpenCV
7. Otras dependencias
8. Instalar Qt5
9. Dependencias de RQt
10. Instalación de la implementación DDS

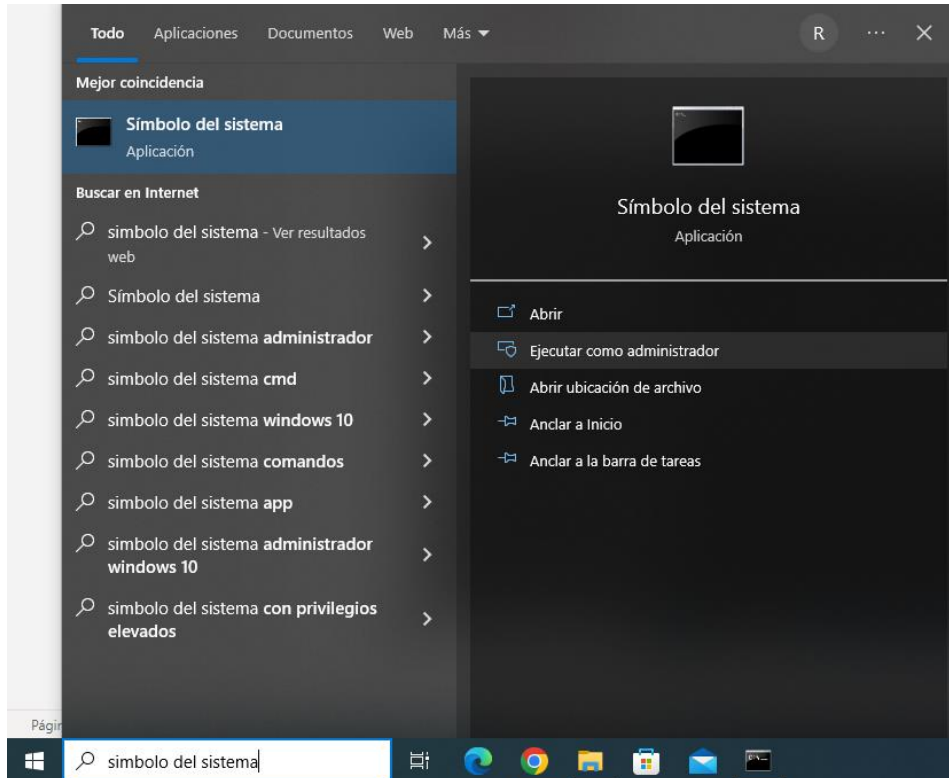
### Instalación de ROS2

1. Descarga el paquete de instalación
2. Setup del entorno
3. Comprobación de la instalación



# Prerrequisitos

Para todo el proceso es necesario trabajar con PowerShell con permisos de administrador. Para ello abrir una ventana de comandos con permisos de administrador:



Una vez dentro de la ventana de comandos lanzar el comando: `powershell.exe`

Ahora ya podemos comenzar la instalación.

## 1. Instalar Chocolatey

<https://chocolatey.org/install>

Comprobar las políticas de ejecución del sistema:

```
Get-ExecutionPolicy
```

Si devuelve Restricted entonces lanzar:

```
Set-ExecutionPolicy AllSigned
```

ó

```
Set-ExecutionPolicy Bypass -Scope Process
```

Después de esta comprobación ya se puede comenzar la instalación (el siguiente comando es una sola línea):

```
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'))
```

Si ya existe una versión de Chocolatey en el equipo se puede actualizar usando el comando:

```
choco upgrade chocolatey
```

Lanzar `refreshenv` o abrir una nueva ventana de comandos como administrador para que se apliquen los cambios. Recuerda lanzar el comando `powershell.exe` cada vez que abra una nueva ventana de comandos para poder seguir con la instalación. De ahora en adelante esta acción se llamará refresque la ventana de comandos.

## 2. Instalar Python 3.8.3

<https://www.python.org/downloads/>

Habiendo instalado previamente Chocolatey, lanzar el comando de instalación de miniconda con la flag para que se añada al path:

```
choco install miniconda3 --version=4.8.3 --params="/AddToPath:1"
```

Refresque la ventana de comandos.

A continuación, crear el entorno de conda con la versión de Python deseada, por ejemplo, ROS2:

```
conda create --name ROS2 -c anaconda python=3.8.3
```

Después activar el entorno:

```
conda activate ROS2
```

O desactivar el entorno:

```
conda deactivate ROS2
```

Para verificar la versión de Python en el entorno miniconda lanzar:

```
python --version
```

## 3. Instalar Visual C++ Redistributables

Instalar vía chocolatey:

```
choco install -y vcredist2013 vcredist140
```

## 4. Instalar OpenSSL

Descargar el ejecutable de la última versión de OpenSSL 1 para Windows de 64 bits. No descargar la versión light. Por ejemplo:

Win64 OpenSSL v1.1.1w EXE   MSI	63MB Installer	Installs Win64 OpenSSL v1.1.1w (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the <a href="#">legal agreement</a> of the installation.
------------------------------------	----------------	---

Instalar con los parámetros por defecto.

Es posible que se lance una alerta de seguridad al abrir el ejecutable, en ese caso seleccionar más información y darle a instalar de todas formas.

El siguiente comando asume que se realizó la instalación con los parámetros por defecto:

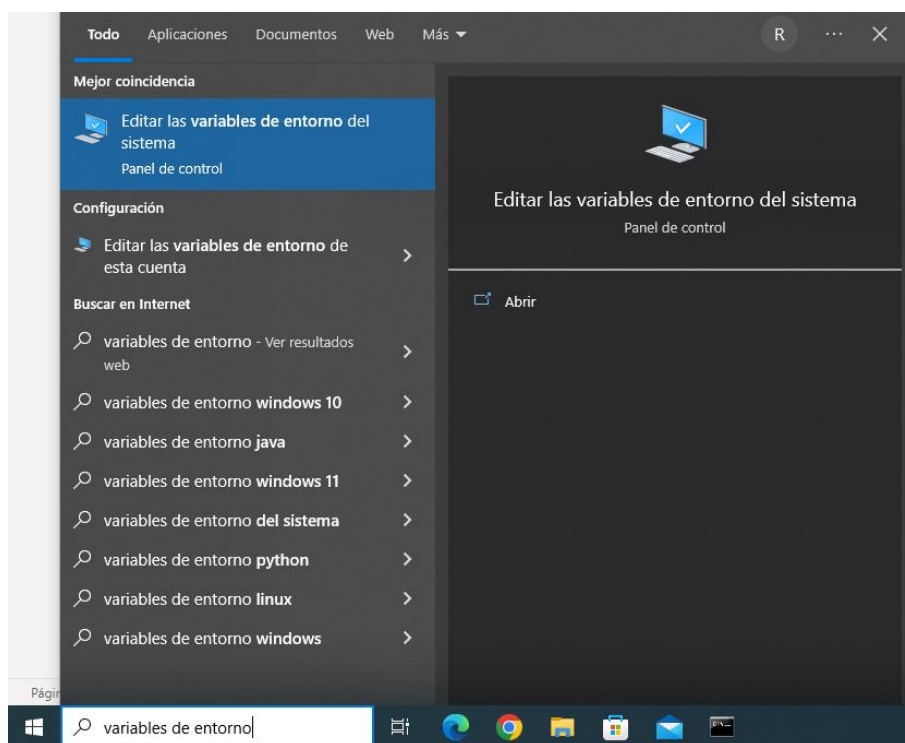
```
setx -m OPENSSL_CONF "C:\Program Files\OpenSSL-Win64\bin\openssl.cfg"
```

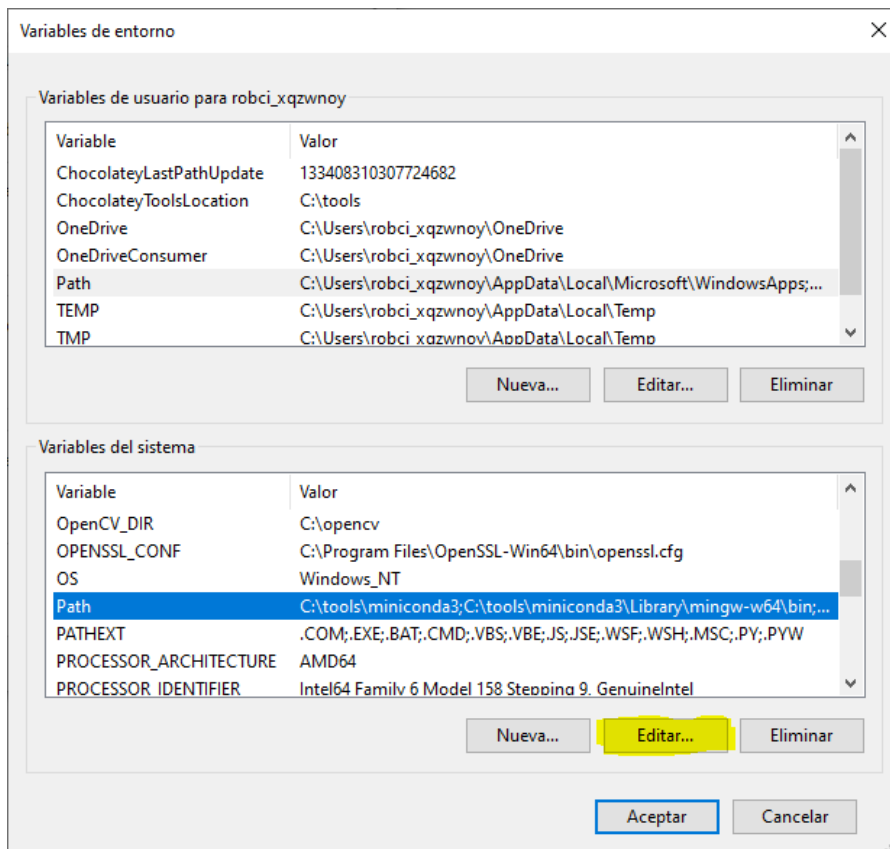
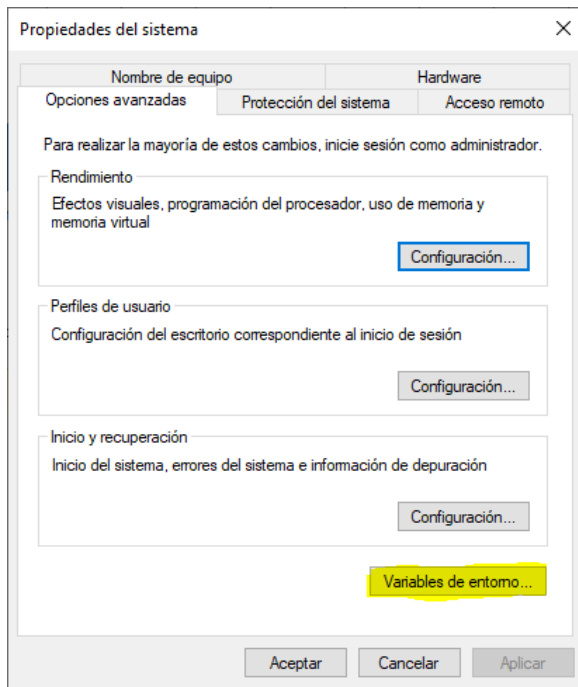
Después es necesario añadir la ruta al PATH. Para ello buscar Variables de Entorno en el buscador de Windows y editar la variable de entorno del sistema llamada Path o PATH.

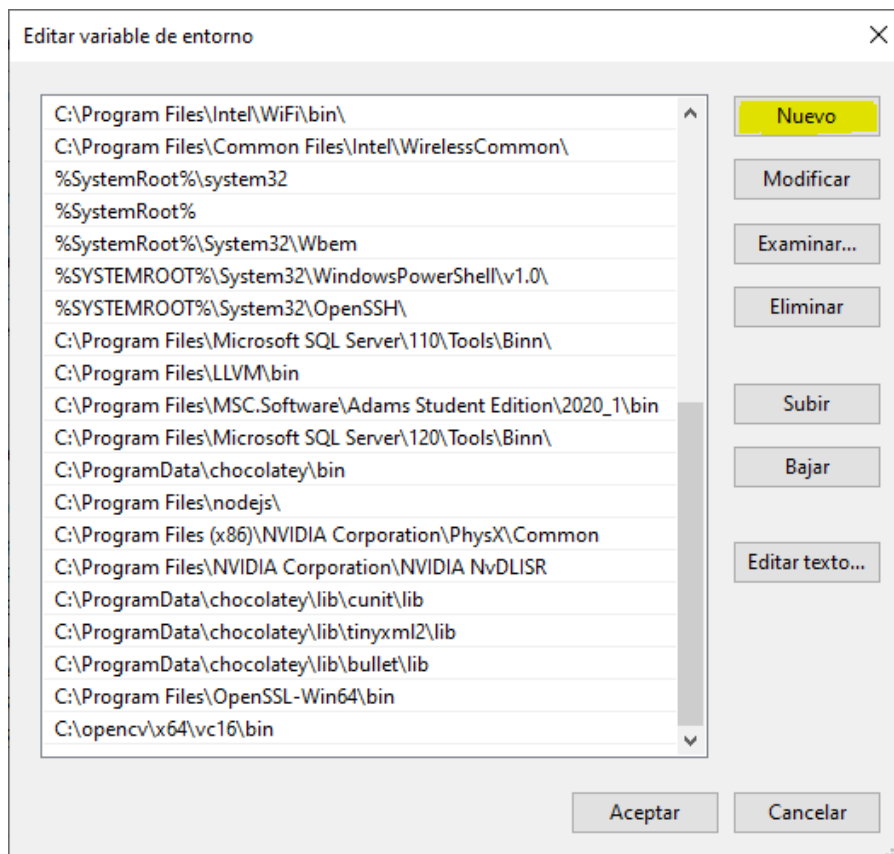
En la pestaña de Opciones Avanzadas hacer click sobre el botón Variables de entorno... que está abajo a la derecha. Seleccionar la variable Path, editar y hacer click en nuevo. Copiar y pegar la ruta hasta el archivo openssl.cfg:

```
C:\Program Files\OpenSSL-Win64\bin
```

Siga la secuencia de imágenes:







Click en aceptar en todos los menús para salir.

Al terminar refresque la ventana de comandos.

## 5. Instalar Visual Studio

Para comprobar si está ya instalado y conocer la versión lanzar:

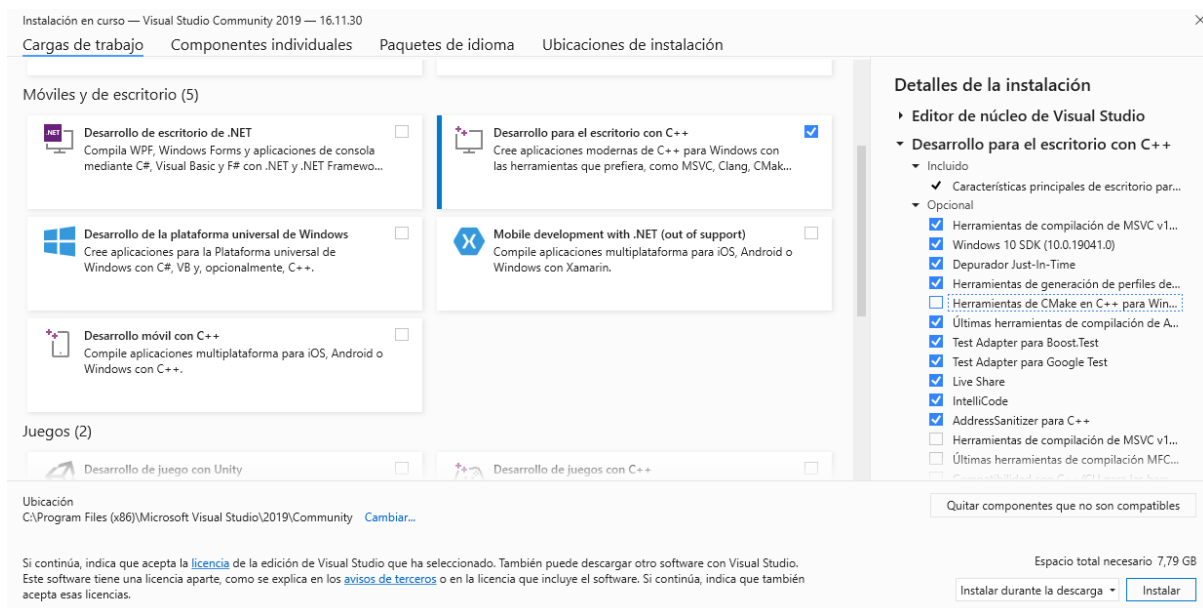
```
"C:\Program Files (x86)\Microsoft Visual Studio\Installer\vswhere.exe" -property catalog_productDisplayVersion
```

Para ROS2 Humble Hawksbill necesitaremos la versión 2019. Ir a

<https://visualstudio.microsoft.com/es/vs/older-downloads/>

Seleccionar Visual Studio 2019 y otros productos y darle a descarga. En la nueva página que se acaba de abrir seleccionar Visual Studio Community 2019.

Abrir el instalador. Una vez terminada la preparación automática asegurarse de que las herramientas de Visual C++ se instalan. Para ello, en el menú Cargas de Trabajo, seleccionar Desarrollo para escritorio con C++ y en la lista que aparece a la derecha asegurarse de que no se instalan las herramientas de C++ CMake deseleccionando la casilla correspondiente.



Refresque la ventana de comandos.

## 6. Instalar OpenCV

Descargar la versión 3.4.6 (compatible con Visual Studio 2019) precompilada de este enlace:

<https://github.com/ros2/ros2/releases/download/opencv-archives/opencv-3.4.6-vc16.VS2019.zip>

También puedes buscar el binario precompilado en el gitHub de opencv:

<https://github.com/opencv/opencv/releases>

Extrae el fichero .zip en C:\ de forma que la carpeta opencv quede en la siguiente ruta: C:\opencv.

Lanzar en la ventana de comandos:

```
setx /m OpenCV_DIR C:\opencv
```

Añadir C:\opencv\x64\vc16\bin al PATH en las variables de entorno. Para ello siga el ejemplo en el apartado 4. Instalar OpenSSL.

Después refresque la ventana de comandos.

## 7. Otras dependencias

Instalar CMake:

```
choco install -y cmake
```

Añadir C:\Program Files\CMake\bin al PATH en las variables de entorno. Para ello seguir los pasos vistos anteriormente.

Refrescar la ventana de comandos.

Descargar los siguientes ficheros de este repositorio: <https://github.com/ros2/choco-packages/releases/tag/2022-03-15>



- asio.1.12.1.nupkg
- bullet.3.17.nupkg
- cunit.2.1.3.nupkg
- eigen-3.3.4.nupkg
- tinyclang-0.12.0.nupkg
- tinyclang2.6.0.0.nupkg

Después instalarlos con el siguiente comando:

```
choco install -y -s <PATH\TO\DOWNLOADS\> asio cunit eigen tinyclang tinyclang2 bullet
```

Donde <PATH\TO\DOWNLOADS\> es la ruta hasta la carpeta donde has descargado los ficheros.

Refrescar la ventana de comandos.

Activar el entorno de miniconda, por ejemplo, ROS2:

```
conda activate ROS2
```

Ahora hacer upgrade de pip y setuptools:

```
python -m pip install -U pip setuptools==59.6.0
```

Por último, instalar dependencias adicionales de python (el siguiente comando es una única línea):

```
python -m pip install -U catkin_pkg cryptography empy importlib-metadata lark==1.1.1 lxml
matplotlib netifaces numpy opencv-python PyQt5 pillow psutil pycairo pydot pyparsing==2.4.7
pyyaml rosdistro
```

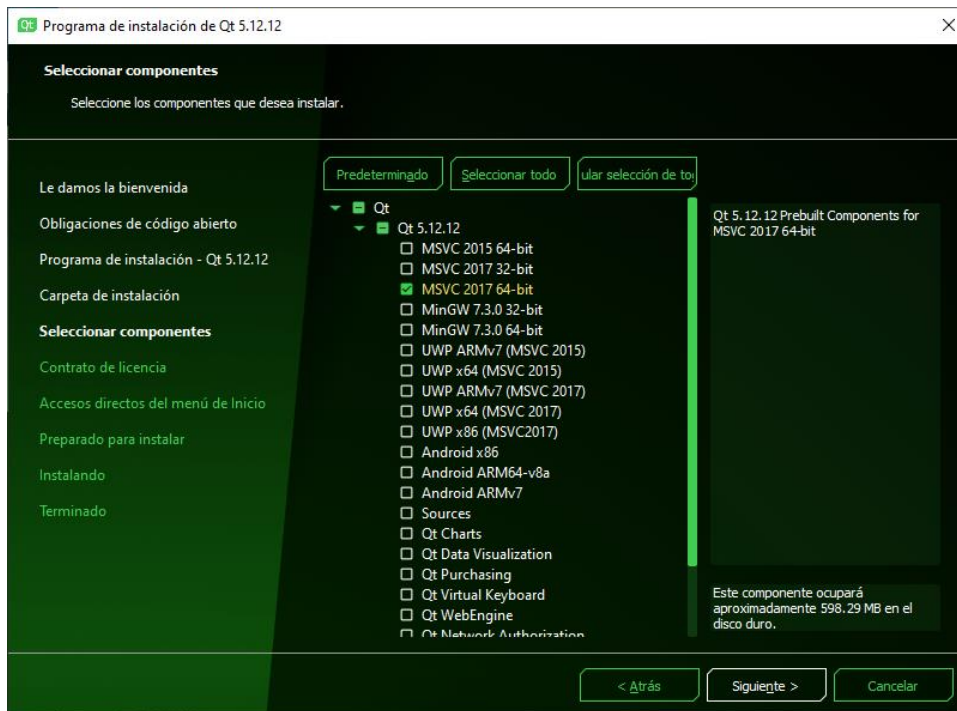
## 8. Instalar Qt5

Selecciona y descarga el instalador offline de Qt5 para la versión 5.12.x de este enlace:

<https://www.qt.io/offline-installers>

Ejecuta el archivo descargado y sigue las instrucciones. Necesitarás registrarte en Qt, yo he usado el correo de la universidad y el número de teléfono de la ETSII.

Durante la instalación asegurarse de instalar el paquete MSVC 2017 64-bit:



Una vez terminada la instalación y asumiendo que Qt se ha instalado en C:\, en una ventana de comandos con permisos de administrador lanzar:

```
setx /m Qt5_DIR C:\Qt\Qt5.12.12\5.12.12\msvc2017_64
setx /m QT_QPA_PLATFORM_PLUGIN_PATH C:\Qt\Qt5.12.12\5.12.12\msvc2017_64\plugins\platforms
```

## 9. Dependencias de RQt

Para usar `rqt_graph` hay que descargar e instalar Graphviz. Se puede descargar el ejecutable desde:

<https://graphviz.org/download/>

ó

Instalar directamente via Chocolatey:

```
choco install graphviz
```

## 10. Instalación de la implementación DDS

Instalaremos RTI Connex DDS pues es el que se utiliza por defecto en ROS2 Humble:

<https://docs.ros.org/en/foxy/Releases/Release-Humble-Hawksbill.html#rmw-connexdds>

Referencia: <https://community.rti.com/documentation/rti-connex-dds-601>

Y: <https://docs.ros.org/en/humble/Installation/DDS-Implementations/Install-Connex-University-Eval.html>

Antes de acceder a la descarga, entrar en: <https://www.rti.com/free-trial> y rellenar el formulario. Una vez rellenado seleccionar en las opciones que aparecen abajo la casilla de *Connex license for Research and Academia*.

A continuación, recibirás un email con una licencia de evaluación de 30 días con la que podrás instalar la versión que se propone a continuación. Se recomienda solicitar una licencia de investigación en la página de contacto de RTI mientras tanto para que antes de que cumpla el plazo de los 30 días se nos conceda una licencia a más largo plazo: <https://www.rti.com/company/contact>

Accederemos a la descarga desde el siguiente portal y seleccionaremos la versión 6.0.1 para Windows: <https://community.rti.com/page/rti-connext-dds-installers-ros-2>

Instalar el ejecutable. Seleccionar la opción de instalar el paquete de Excel spreadsheet puede ser interesante.

Una vez instalado iniciar el launcher descargar y añadir la licencia recibida por correo electrónico seleccionando la opción *Copy to RTI install location (all users)*. **Repetir este paso cuando se reciba la licencia académica de duración 1 año (renovable).**

Ya se puede pasar al siguiente paso *Instalación de ROS2*. De todas formas, se añade como alternativa Fast DDS de eProsima por si RTI Connex DDS diera problemas.

Seguimos la documentación en: <https://docs.ros.org/en/humble/Installation/DDS-Implementations/Working-with-eProsima-Fast-DDS.html>

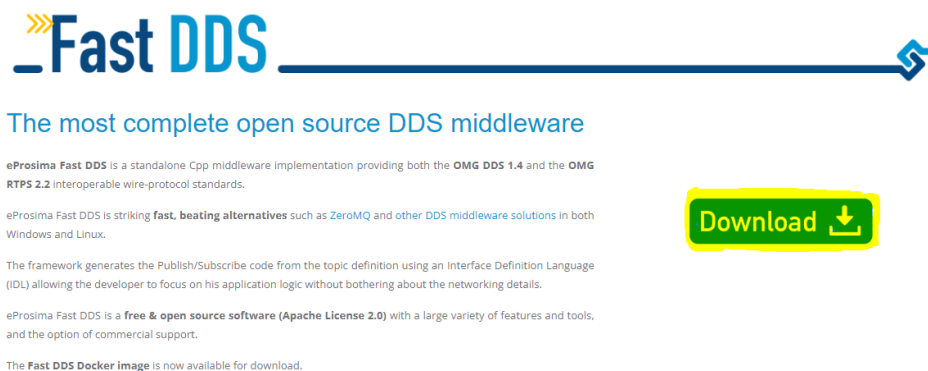
Instalaremos eProsima Fast DDS que es el DDS por defecto que usa ROS2.

Primero instalaremos rosdep ([https://wiki.ros.org/rosdep#Installing\\_rosdep](https://wiki.ros.org/rosdep#Installing_rosdep)) en una ventana de comandos con permisos de administrador:

```
conda activate ROS2
pip install -U rosdep
```

A continuación, descargaremos Fast DDS del siguiente enlace:  
<https://www.eprosima.com/index.php/products-all/eprosima-fast-dds>

Darle a Download y registrarse con el email de la universidad:



**Fast DDS**

The most complete open source DDS middleware


eProsima Fast DDS is a standalone Cpp middleware implementation providing both the **OMG DDS 1.4** and the **OMG RTPS 2.2** interoperable wire-protocol standards.

eProsima Fast DDS is striking **fast, beating alternatives** such as ZeroMQ and other DDS middleware solutions in both Windows and Linux.

The framework generates the Publish/Subscribe code from the topic definition using an Interface Definition Language (IDL) allowing the developer to focus on his application logic without bothering about the networking details.

eProsima Fast DDS is a **free & open source software (Apache License 2.0)** with a large variety of features and tools, and the option of commercial support.

The **Fast DDS Docker image** is now available for download.

**Download** 

Buscar la descarga de eProsima Fast DDS:

## Official Releases

### eProsima DDS suite

eProsima DDS suite Downloads

The eProsima DDS world in one docker!

[View releases in this category](#)

### eProsima Fast DDS

eProsima Fast DDS/RTFS Downloads

The most complete open-source DDS middleware!

[View releases in this category](#)

### eProsima DDS Router

Seleccionar la release estable más reciente y hacer clic en view files:

**2.11.2** Stable

**Maturity:** Stable

**Released on:** Tuesday, 22 August 2023 02:00

**Viewed:** 928 times

DESCRIPTION

RELEASE NOTES

eProsima Fast DDS is a high performance publish subscribe framework to share data in distributed systems using a decoupled model based on Publishers, Subscribers and Data Topics.

eProsima Fast DDS is striking fast, beating alternatives such as ZeroMQ and other pubsub middleware in both Windows and Linux.

For more information also visit the [official documentation of the eProsima Fast DDS](#).

[View files](#)

Seleccionar la versión para windows:

**eProsima Fast DDS 2.11.2 - Win (32 & 64)**

**Downloaded:** 1830 times

**File size:** 155,14 Mb

**MD5 Signature:** 7203bc5bde9cb120b883b9a64e16ead2

**SHA1 Signature:** 7d8693a27d3294ec41263f3b436dbdc9e726277f

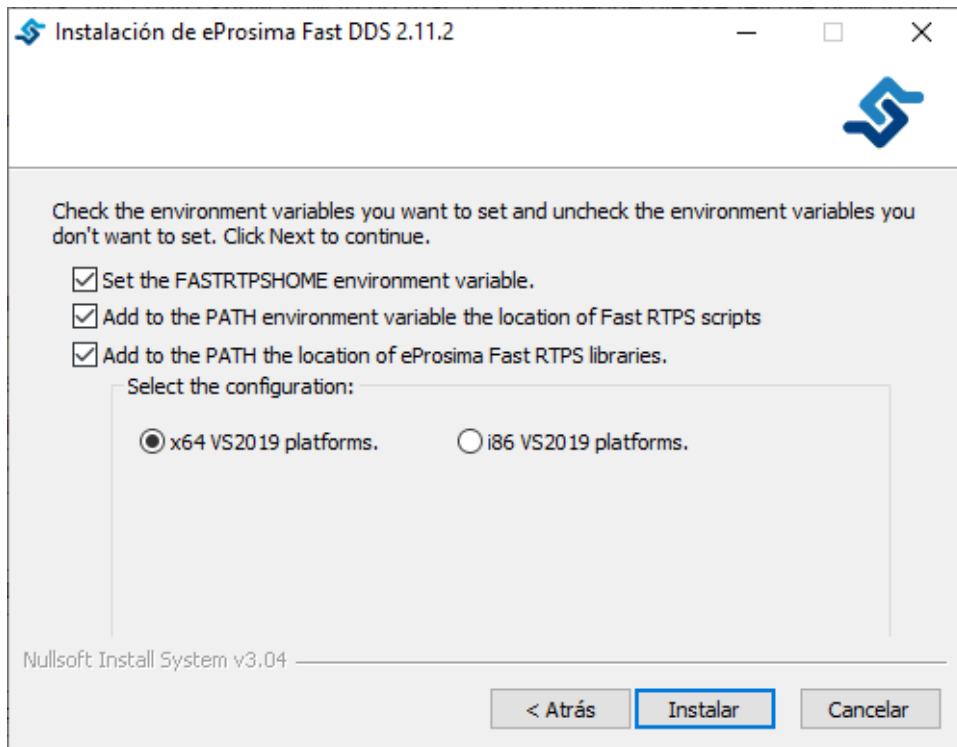
Windows Installer of eProsima Fast DDS 2.11.2 - Windows (32 & 64)

VS2015, VS2017, VS2019 Support

Instructions on how to install Fast DDS on Windows can be found [here](#).

[Download now](#)

Por último, ejecutar el instalable e instalar con la siguiente configuración:



Nota: i86 es para 32-bits.

Más info sobre eProxima Fast DDS en: <https://fast-dds.docs.eprosima.com/en/latest/>

## Instalación de ROS2

Antes de continuar con la instalación de ROS2 es necesario reiniciar el sistema.

Referencia: <https://docs.ros.org/en/humble/Installation.html>

### 1. Descarga el paquete de instalación

Referencia: <https://docs.ros.org/en/humble/Installation/Windows-Install-Binary.html#downloading-ros-2>

Accede a: <https://github.com/ros2/ros2/releases> , busca el paquete deseado, de la forma: ROS 2 Humble Hawksbill - Patch Release x, y descarga la versión para Windows `ros2-humble-*-windows-release-amd64.zip`

Descomprime el archivo en algún sitio, por ejemplo, en la carpeta `C:\dev\ros2_humble`

### 2. Setup del entorno

En una ventana de comandos activamos nuestro entorno de conda y llamamos al archivo de configuración del entorno de ROS2:

```
call C:\dev\ros2_humble\local_setup.bat
```

Según la documentación es normal que el comando anterior, si todo ha ido bien, devuelva "The system cannot find the path specified". exactamente una vez. Si alguna cosa falló durante el setup no se mostrará el mensaje anterior.

En mi caso cuando ejecuté el comando no se devolvió ningún mensaje.

Por ejemplo, un error de la forma especificada en la documentación que puede ocurrir es con la configuración DDS: "[rti\_connext\_dds\_cmake\_module][warning] RTI Connex DDS environment script not found (\resource\scripts\rtisetenv\_x64Win64VS2017.bat). RTI Connex DDS will not be available at runtime, unless you already configured PATH manually."

Para solucionarlo usar la respuesta del 1 de noviembre de 2022 de gakutasu en este hilo:

<https://answers.ros.org/question/379164/rti-connext-dds-environment-script-not-found-ros2-on-windows/>

Que se basa en esta guía: <https://docs.ros.org/en/humble/How-To-Guides/Working-with-multiple-RMW-implementations.html>

### 3. Comprobación de la instalación

Porbar en una ventana:

```
ros2 run demo_nodes_cpp talker
```

Y en otra:

```
ros2 run demo_nodes_py listener
```

No funciona, igual: probando con esta solución se arregla:

<https://github.com/ros2/ros2/issues/525#issuecomment-490069329>

Creo que esto no hace falta:

Si se deseara usar ROS2 fuera del entorno de miniconda habría que instalar también las librerías en la versión de python que haya fuera del entorno que hemos creado. Para ello habría que ejecutar las mismas operaciones, pero sin hacer `conda activate`.