# TECNICAS DIGITALES 2

Actividad de Formación Practica 1

GRUPO 1

Profesor:

Ing. Rubén Darío Mansilla

ATTP:

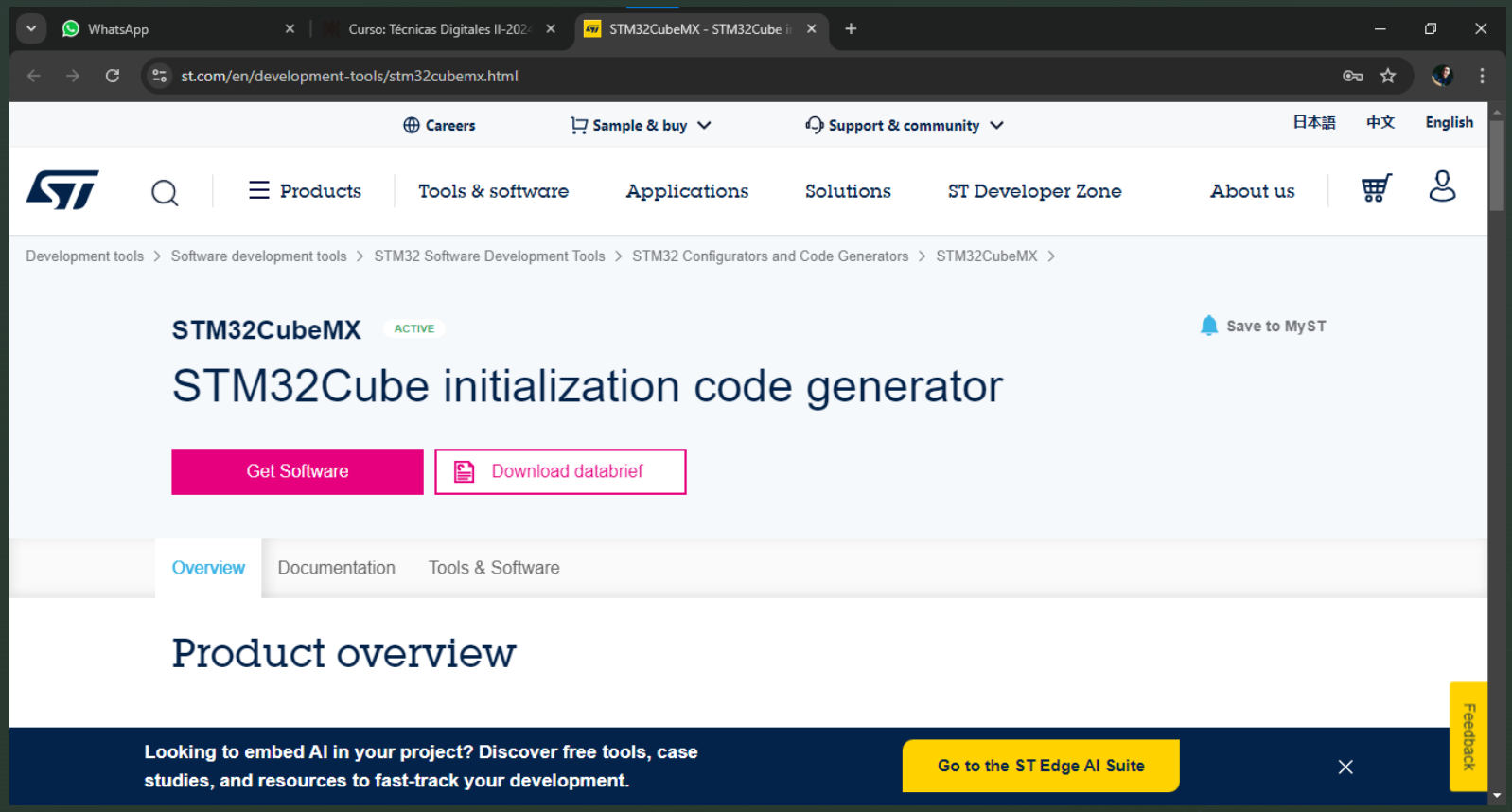
Ing. Lucas Abdala

- tutorial en soporte digital, que explique como se aborda el uso del STM32CubeIDE,

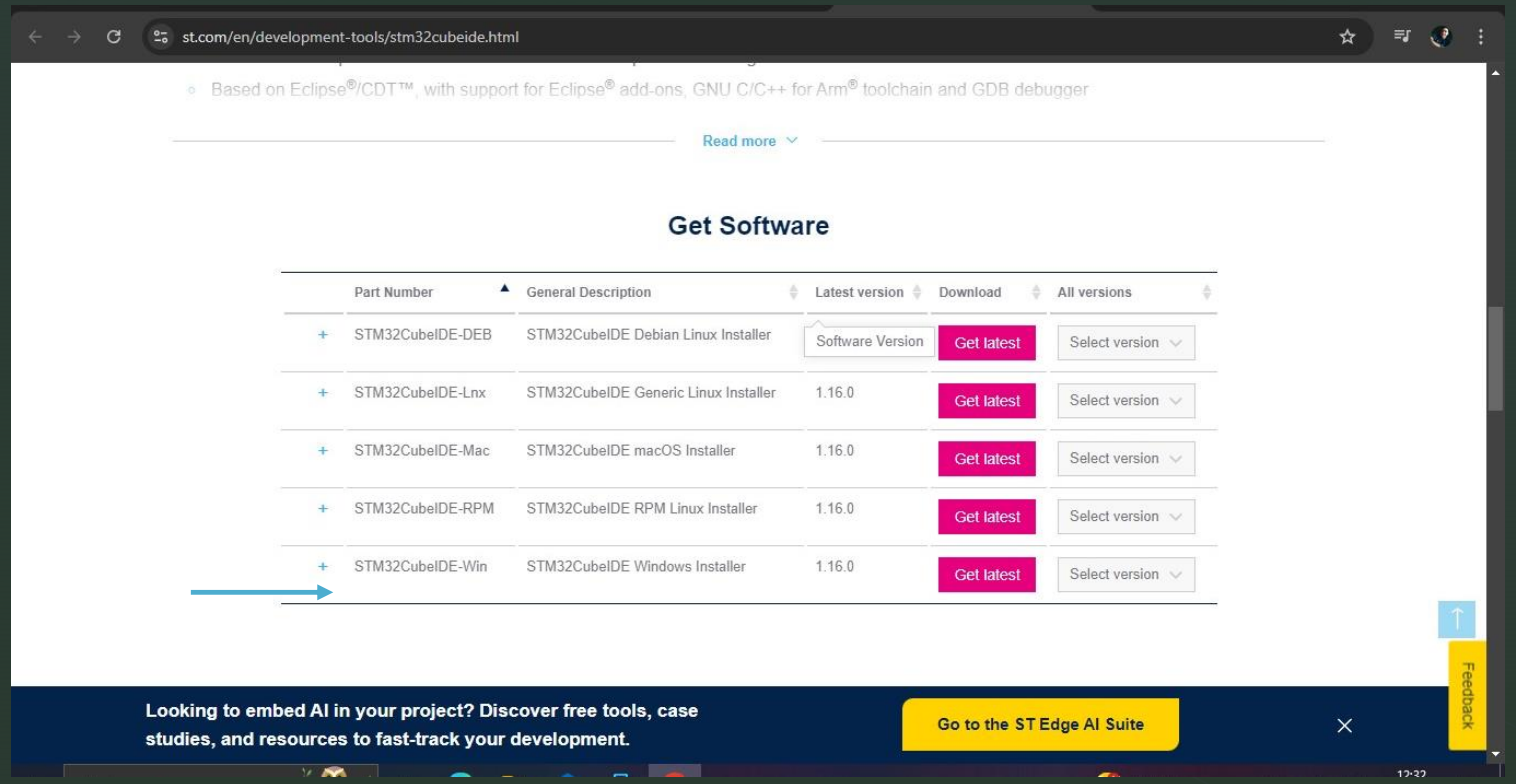
  Instalación del software STM32cubeide

  link de descarga del software:
  https://www.st.com/en/development-tools/stm32cubeide.html

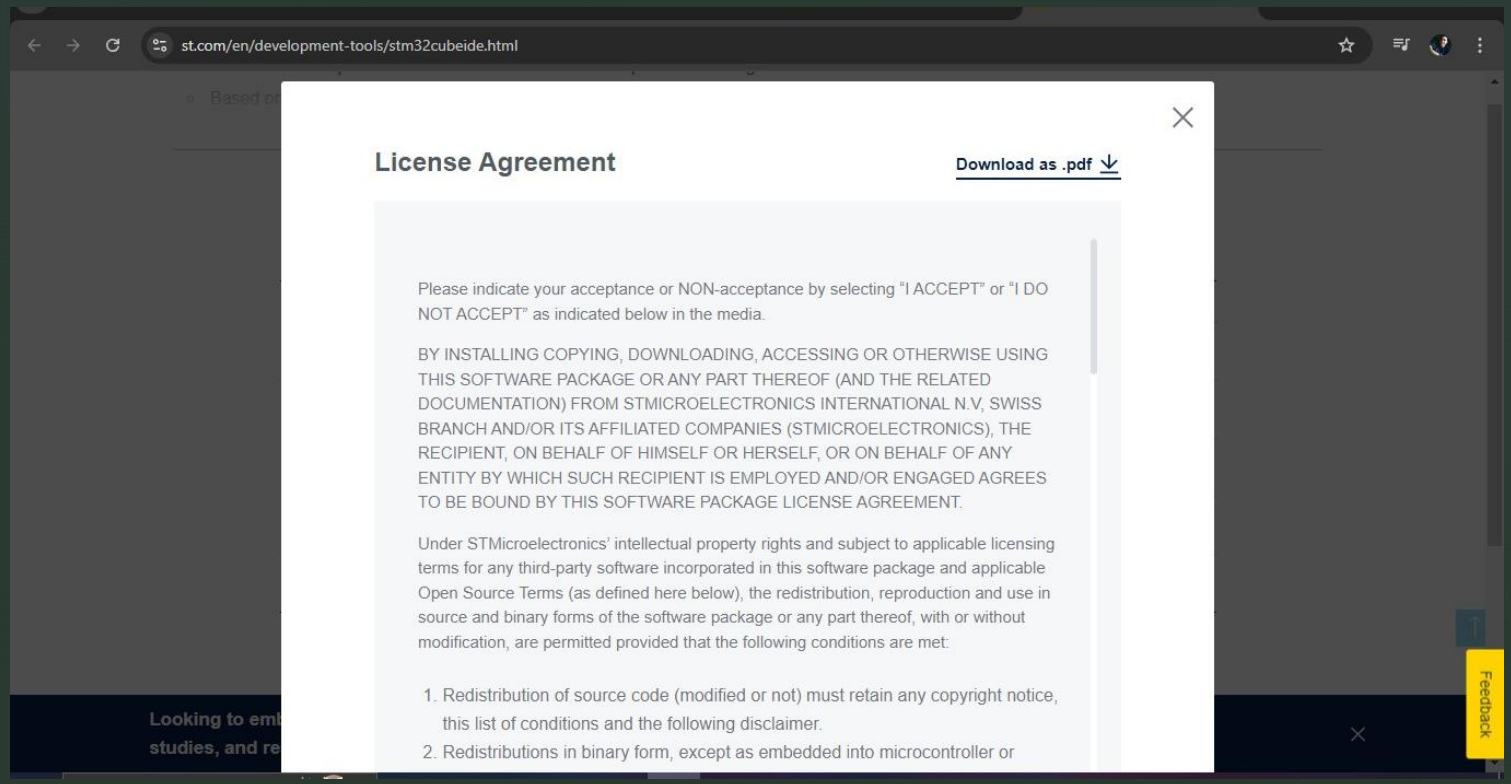
# Descripción general de la plataforma de descarga de IDE
aquí tenemos (descripción general, documentación, herramienta y software)



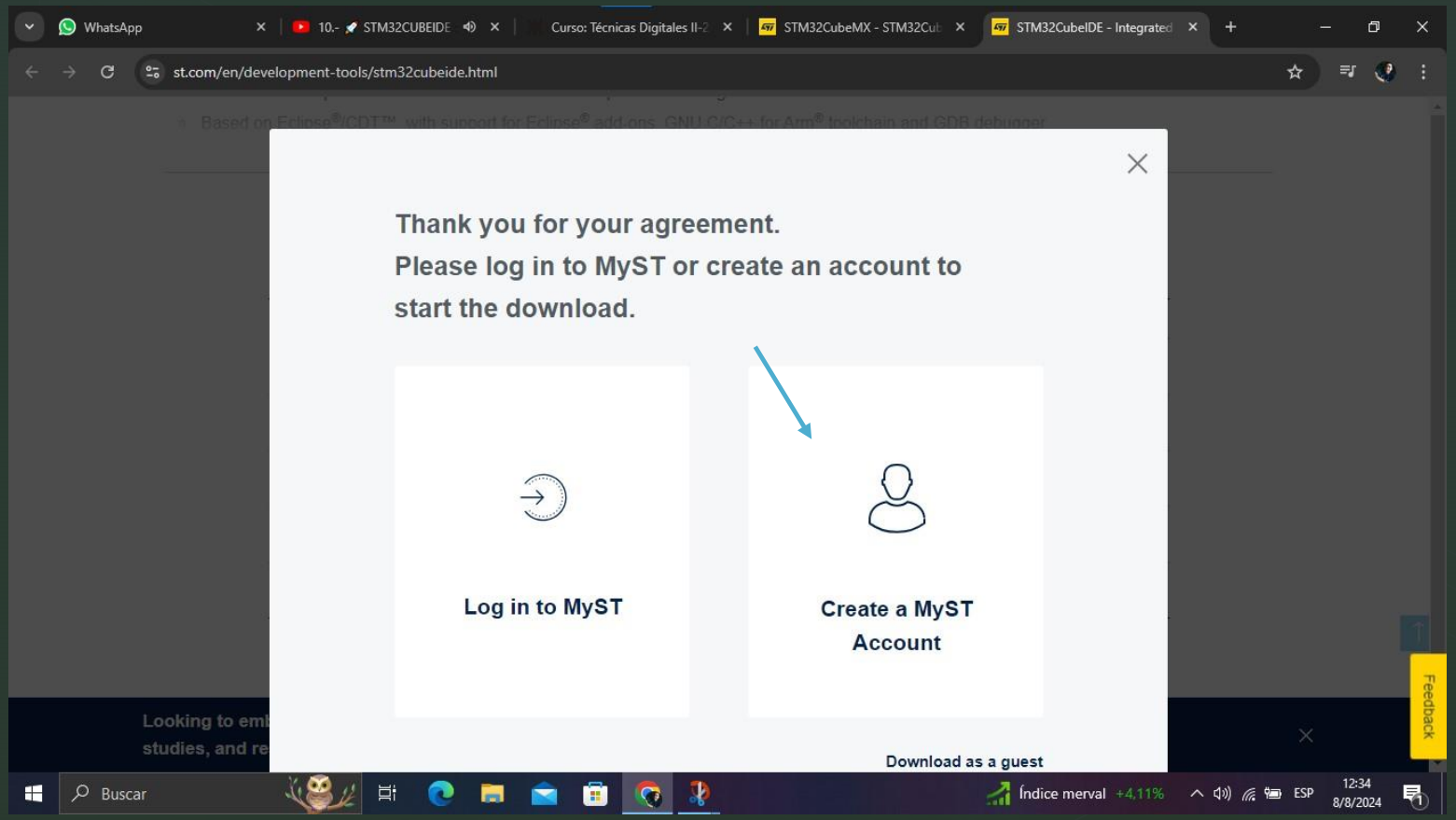
- La plataforma presenta distintos tipos de software para distinto tipo de sistema operativo:



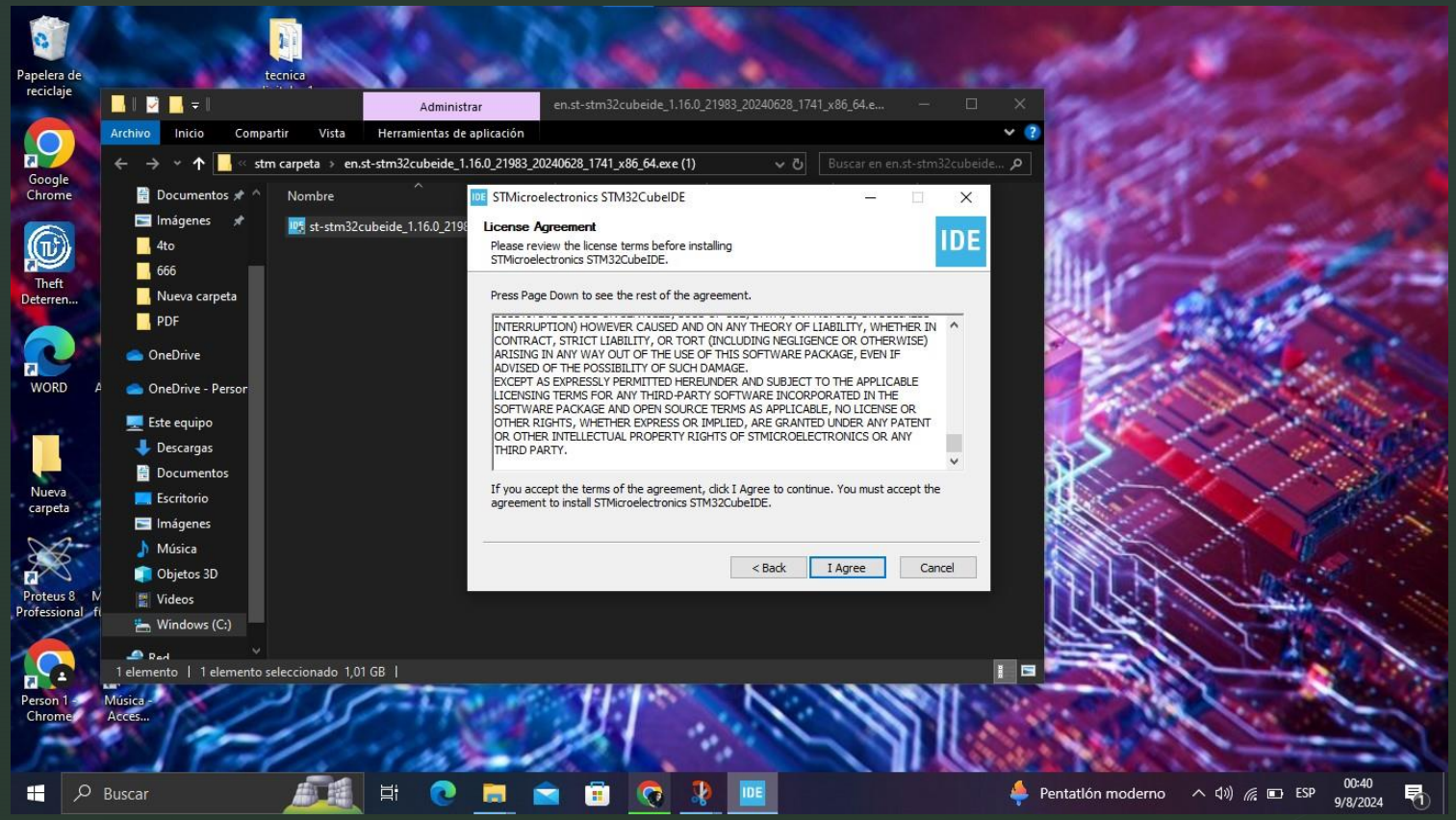
# Una ves elegido el software de acuerdo a su sistema operativo nos va aparecer, términos y condiciones:



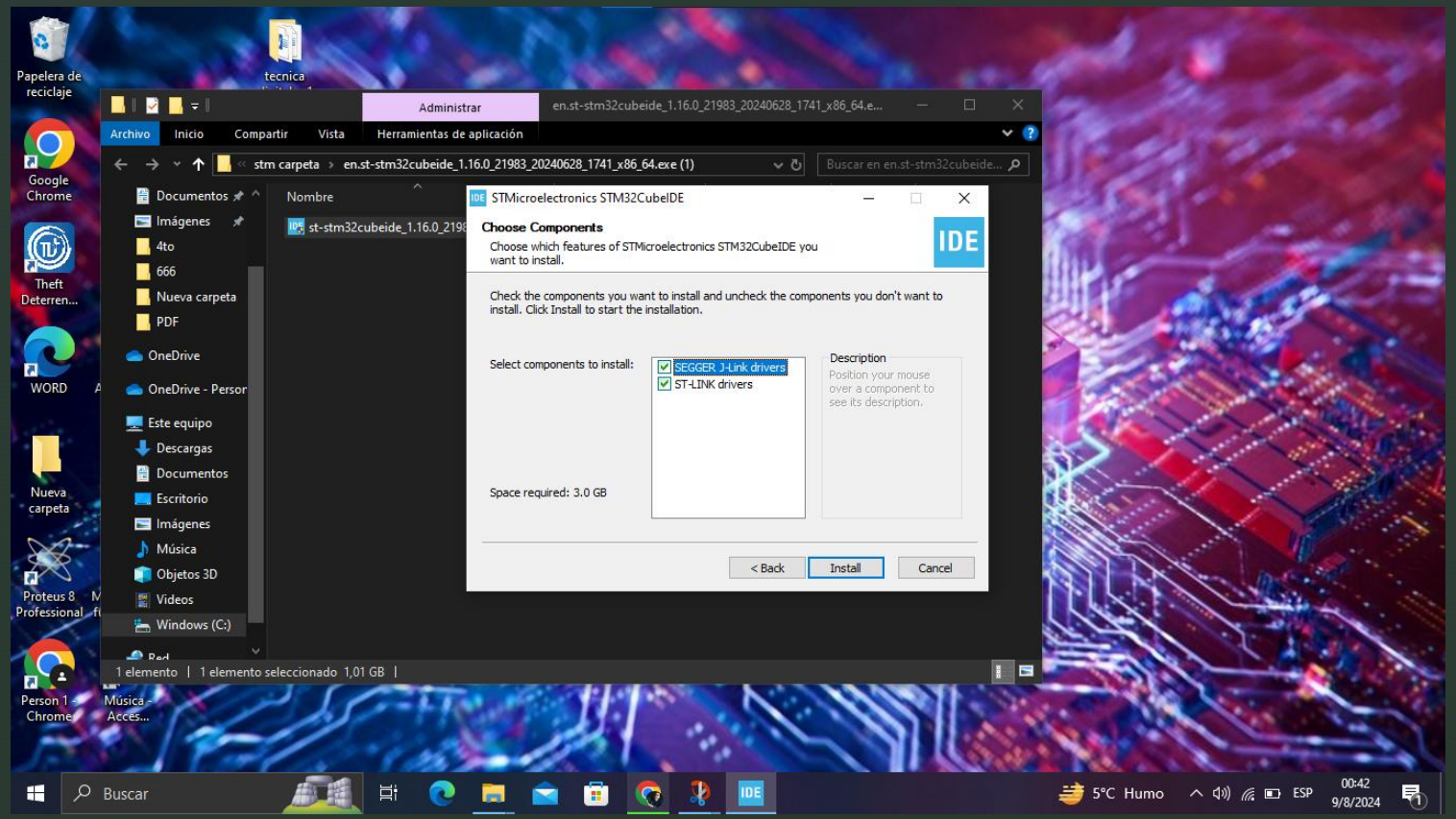
Una vez aceptados los términos y condiciones la plataforma nos da la opción de descargar el software con invitado o crear una cuenta en MyST (recomendación CREAR UNA CUENTA MyST)



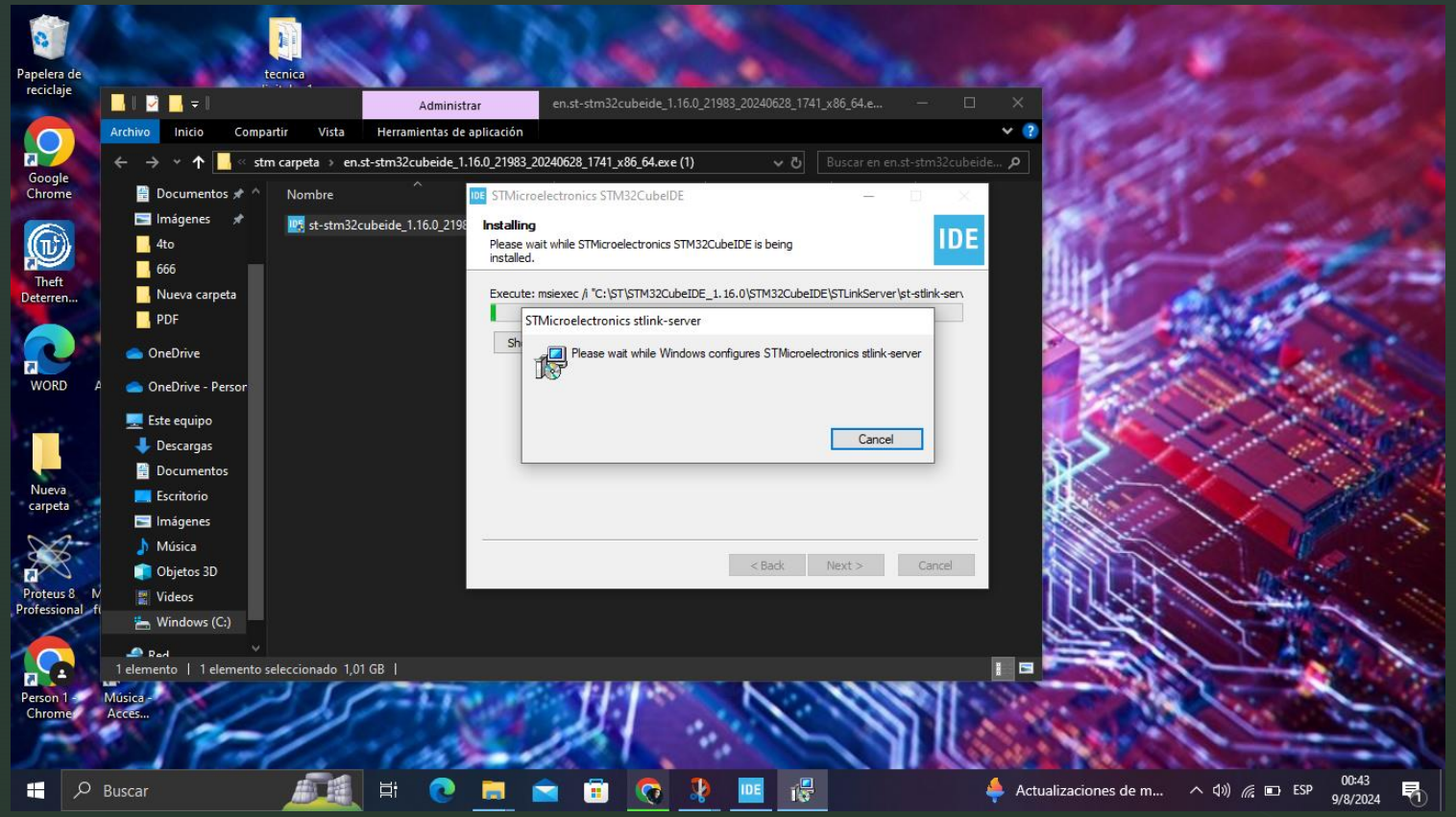
# Una ves descargado pasamos a la instalación 1)



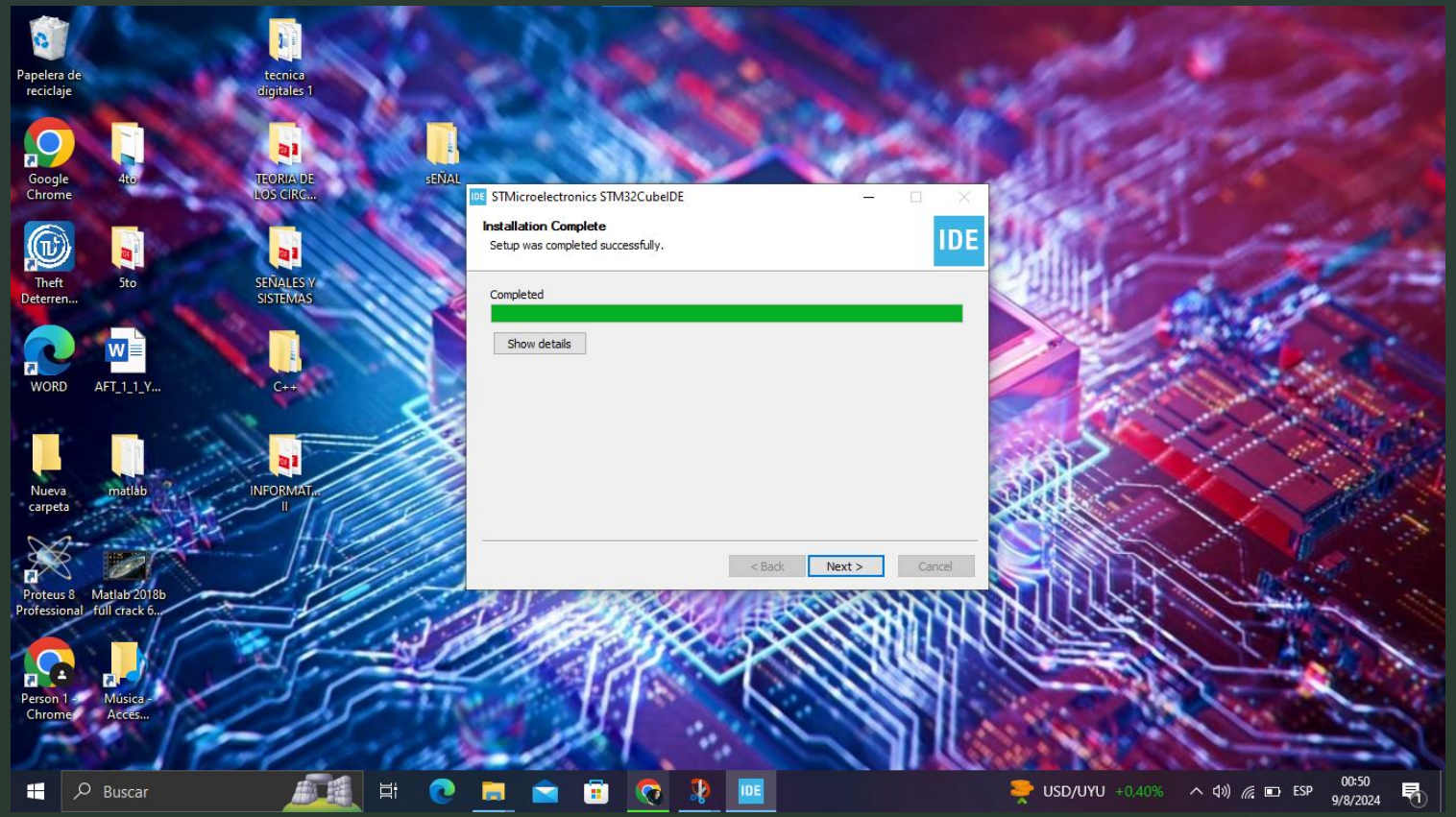
# 2)


Papelera de reciclaje
tecnica
Google Chrome
Theft Deterren...
WORD
Nueva carpeta
Proteus 8 Professional
Person 1 - Chrome
Música - Acces...
Administrar
en.st-stm32cubeide_1.16.0_21983_20240628_1741_x86_64.e...
Archivo
Inicio
Compartir
Vista
Herramientas de aplicación
stm carpeta
en.st-stm32cubeide_1.16.0_21983_20240628_1741_x86_64.exe (1)
Buscar en en.st-stm32cubeide...
Documentos
Imágenes
4to
666
Nueva carpeta
PDF
OneDrive
OneDrive - Persor
Este equipo
Descargas
Documentos
Escritorio
Imágenes
Música
Objetos 3D
Videos
Windows (C:)
Nombre
st-stm32cubeide_1.16.0_219
1 elemento | 1 elemento seleccionado 1,01 GB
STMicroelectronics STM32CubeIDE
Choose Components
Choose which features of STMicroelectronics STM32CubeIDE you want to install.
IDE
Check the components you want to install and uncheck the components you don't want to install. Click Install to start the installation.
Select components to install:
SEGGER J-Link drivers
ST-LINK drivers
Description
Position your mouse over a component to see its description.
Space required: 3.0 GB
< Back
Install
Cancel
Buscar
5°C Humo
ESP
00:42
9/8/2024

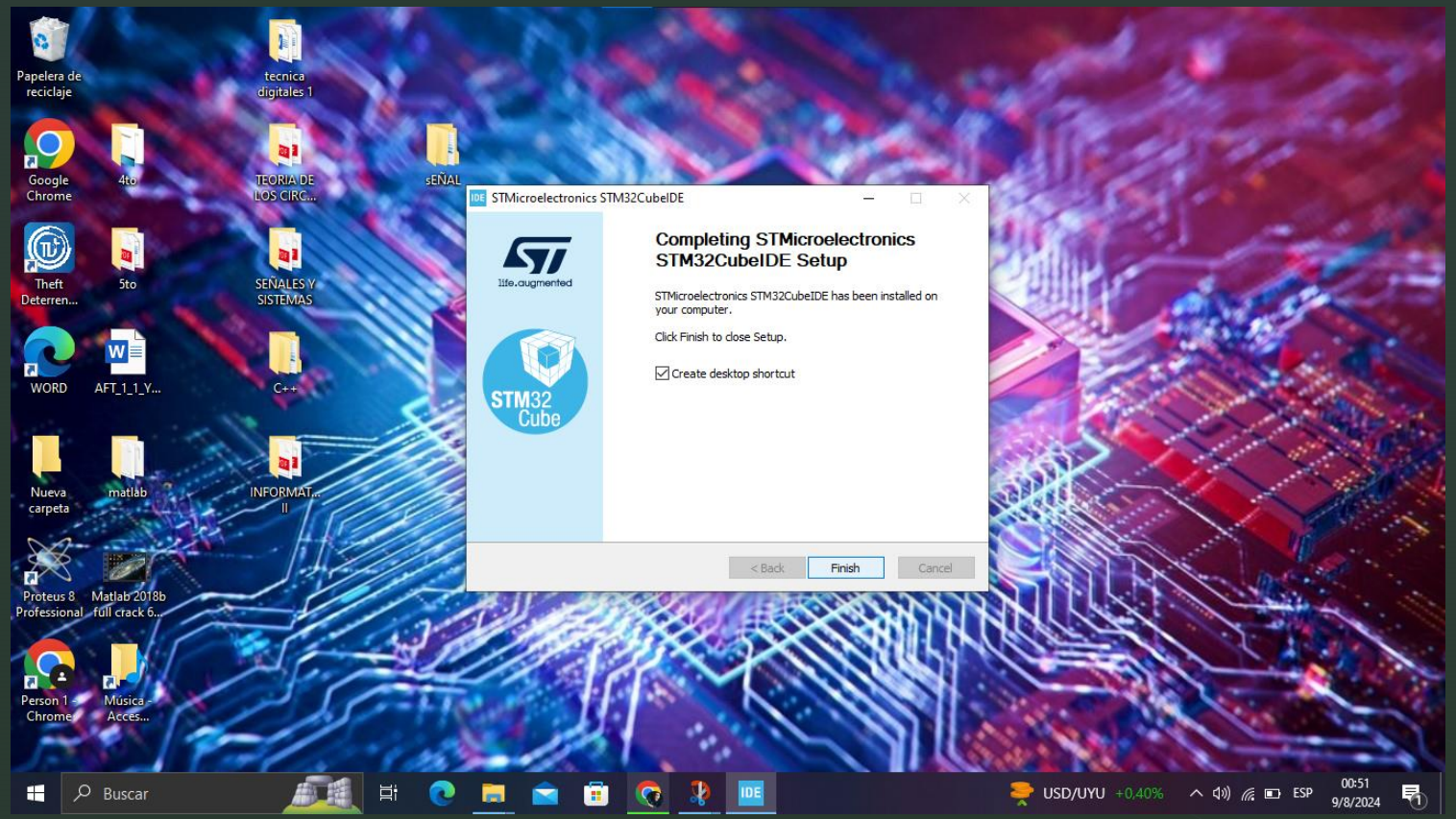
# 3)


Papelera de reciclaje
tecnica
Google Chrome
Theft Deterren...
WORD
Nueva carpeta
Proteus 8 Professional
Person 1 - Chrome
Música - Acces...
Administrar
en.st-stm32cubeide_1.16.0_21983_20240628_1741_x86_64.e...
Archivo
Inicio
Compartir
Vista
Herramientas de aplicación
stm carpeta
en.st-stm32cubeide_1.16.0_21983_20240628_1741_x86_64.exe (1)
Buscar en en.st-stm32cubeide...
Documentos
Imágenes
4to
666
Nueva carpeta
PDF
OneDrive
OneDrive - Persor
Este equipo
Descargas
Documentos
Escritorio
Imágenes
Música
Objetos 3D
Videos
Windows (C:)
Nombre
st-stm32cubeide_1.16.0_2198
1 elemento | 1 elemento seleccionado 1,01 GB
STMicroelectronics STM32CubeIDE
Installing
Please wait while STMicroelectronics STM32CubeIDE is being installed.
IDE
Execute: msiexec /i "C:\ST\STM32CubeIDE_1.16.0\STM32CubeIDE\STLinkServer\st-stlink-ser\
STMicroelectronics stlink-server
Please wait while Windows configures STMicroelectronics stlink-server
Cancel
< Back
Next >
Cancel
Buscar
Actualizaciones de m...
ESP
00:43
9/8/2024

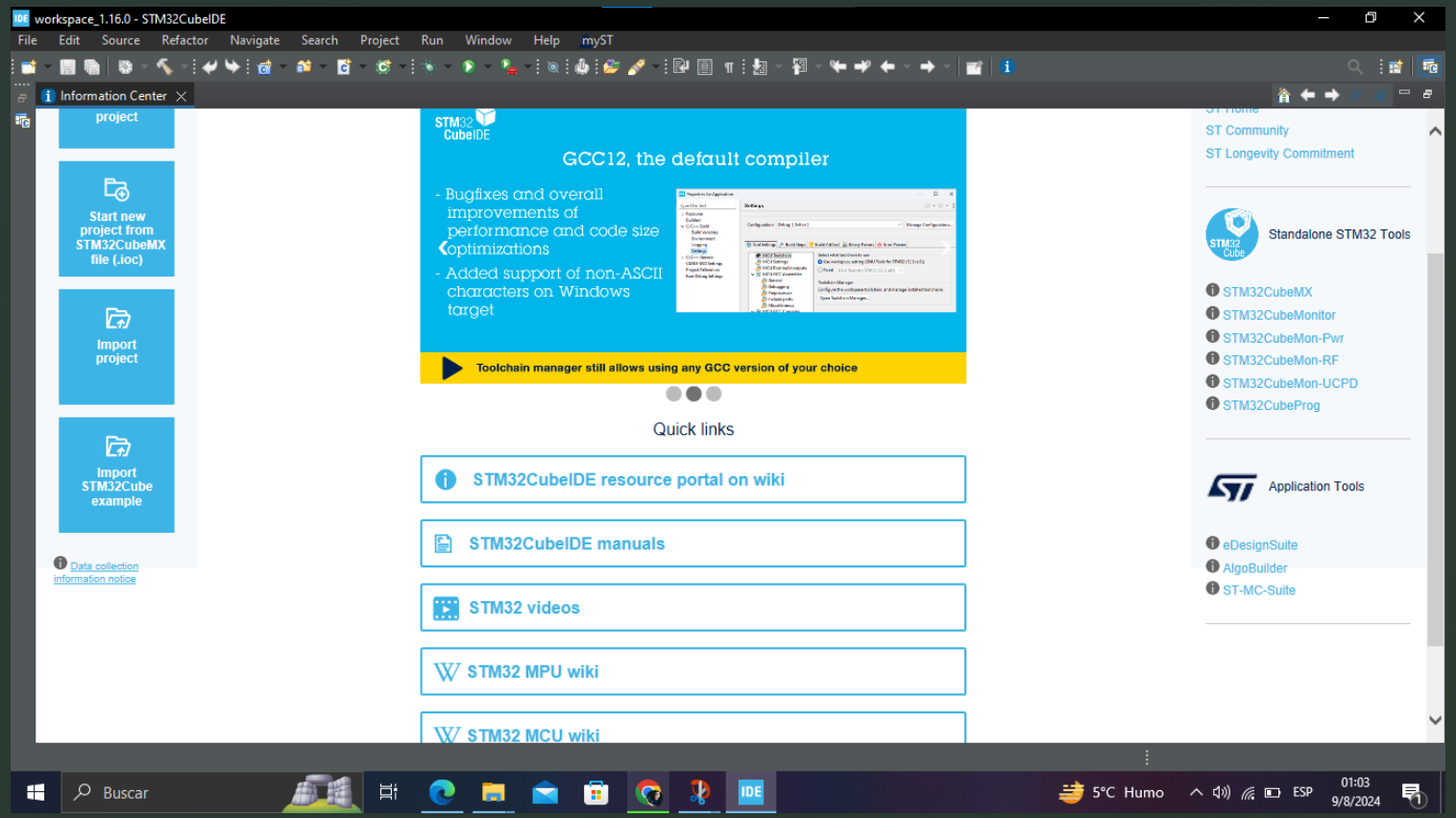
# 4)


Papelera de reciclaje
tecnica digitales 1
Google Chrome
4to
TEORIA DE LOS CIRC...
sEÑAL
Theft Deterren...
5to
SEÑALES Y SISTEMAS
WORD
AFT_1_1_Y...
C++
Nueva carpeta
matlab
INFORMAT... II
Proteus 8 Professional
Matlab 2018b full crack 6...
Person 1 - Chrome
Música - Acces...
STMicroelectronics STM32CubeIDE
Installation Complete
Setup was completed successfully.
IDE
Completed
Show details
< Back
Next >
Cancel
Buscar
USD/UYU +0,40%
ESP
00:50
9/8/2024

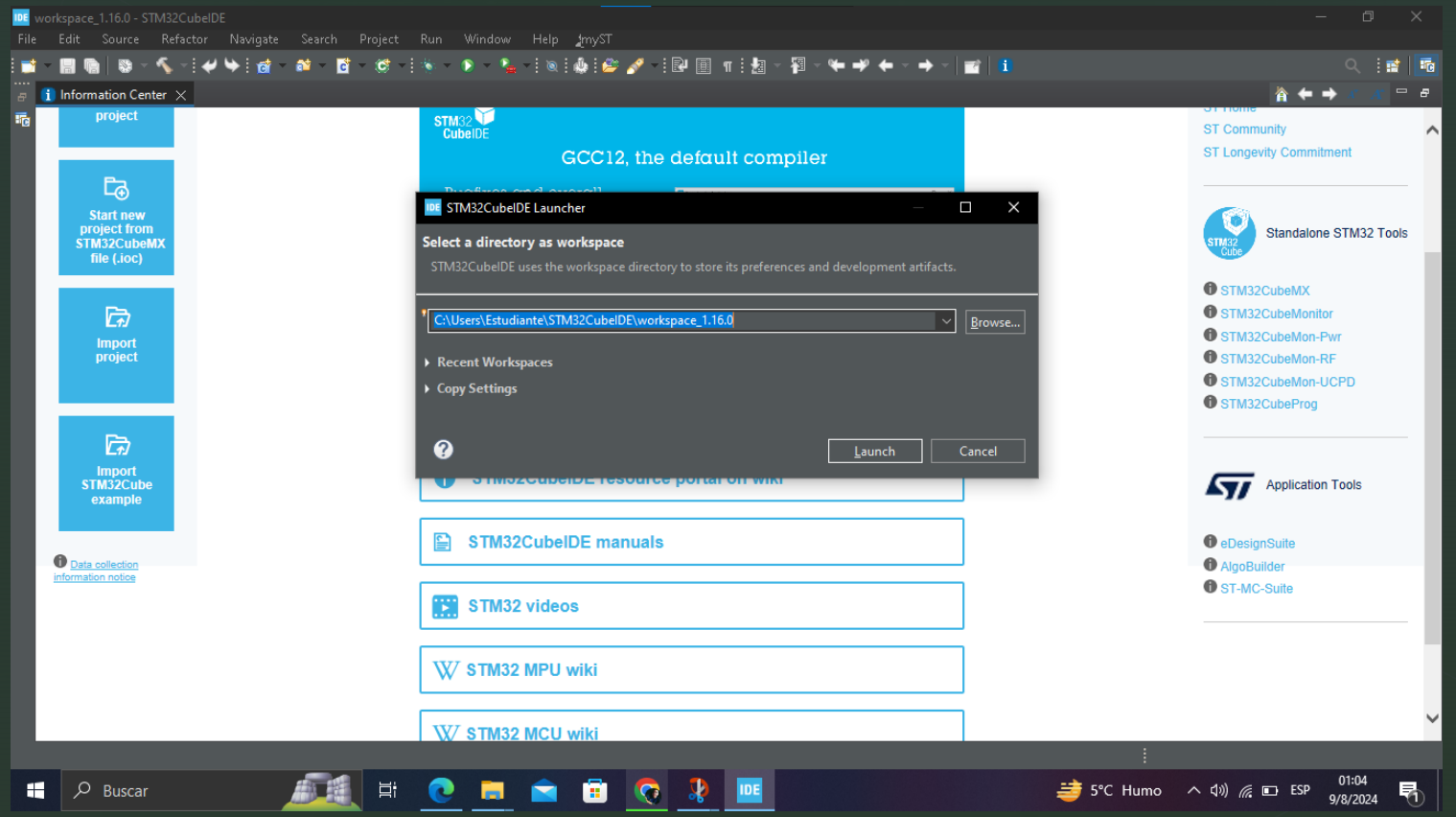
# 5)


Papelera de reciclaje
tecnica digitales 1
Google Chrome
4to
TEORIA DE LOS CIRC...
sEÑAL
Theft Deterren...
5to
SEÑALES Y SISTEMAS
WORD
AFT_1_1_Y...
C++
Nueva carpeta
matlab
INFORMAT... II
Proteus 8 Professional
Matlab 2018b full crack 6...
Person 1 - Chrome
Música - Acces...
STMicroelectronics STM32CubeIDE
life.augmented
STM32 Cube
Completing STMicroelectronics STM32CubeIDE Setup
STMicroelectronics STM32CubeIDE has been installed on your computer.
Click Finish to close Setup.
Create desktop shortcut
< Back
Finish
Cancel
Buscar
USD/UYU +0,40%
ESP
00:51
9/8/2024

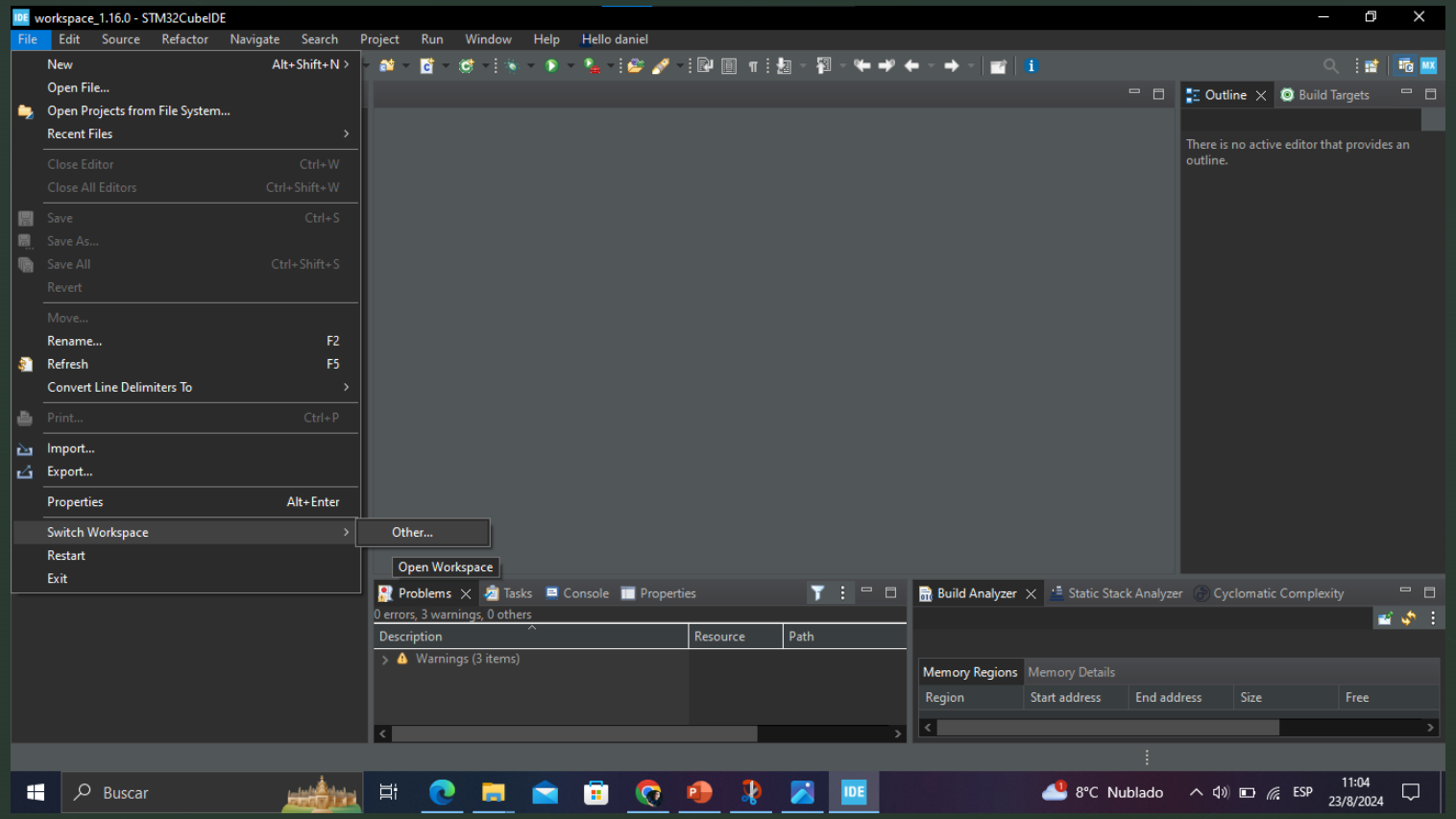
# Uso del IDE
-una vez instalado el software pasamos a iniciar el programa



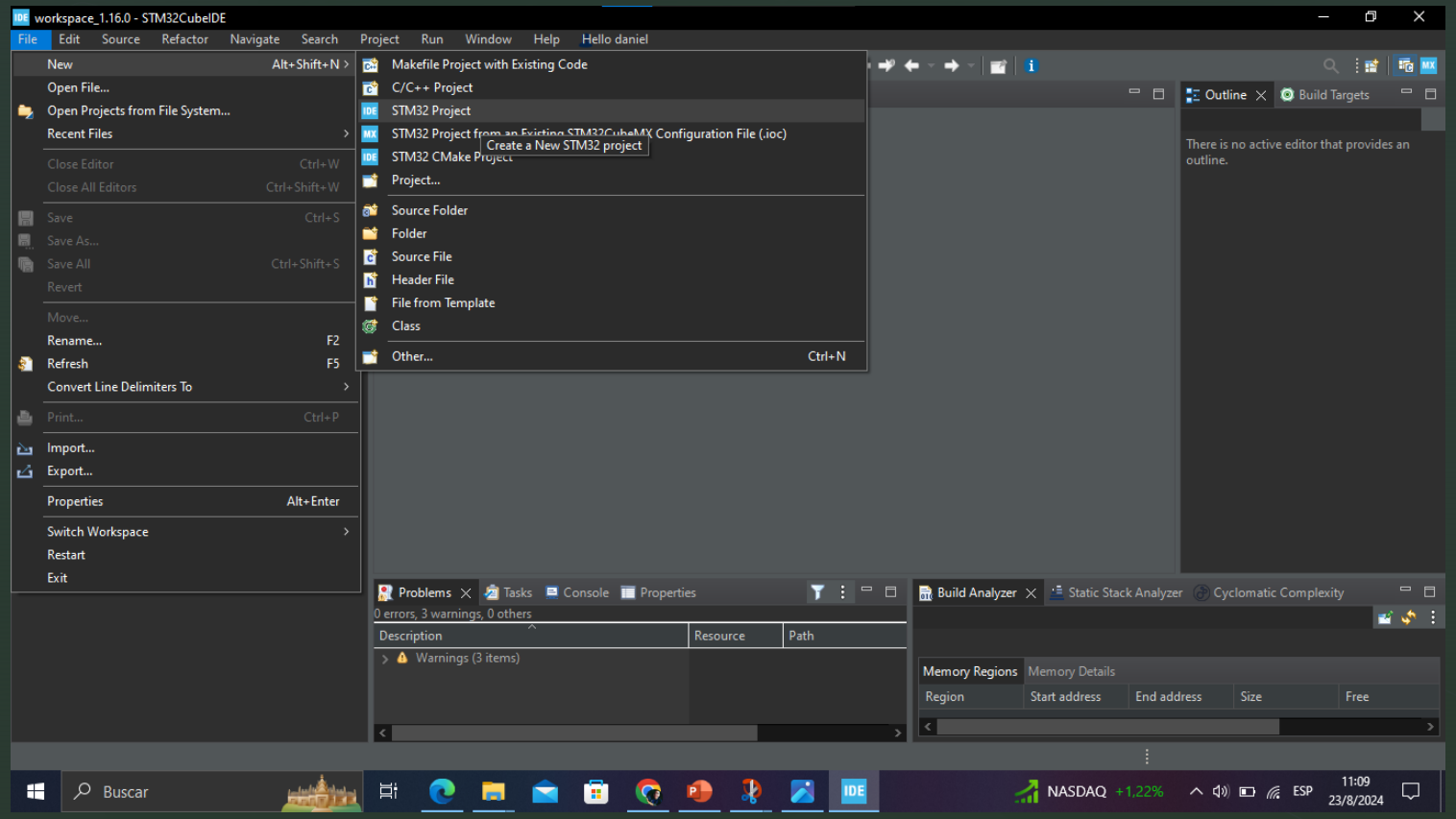
Una vez que abrimos la app lo primero que nos va aparecer es que configuremos nuestro workspace (lugar en donde se van a guardar nuestros proyectos)



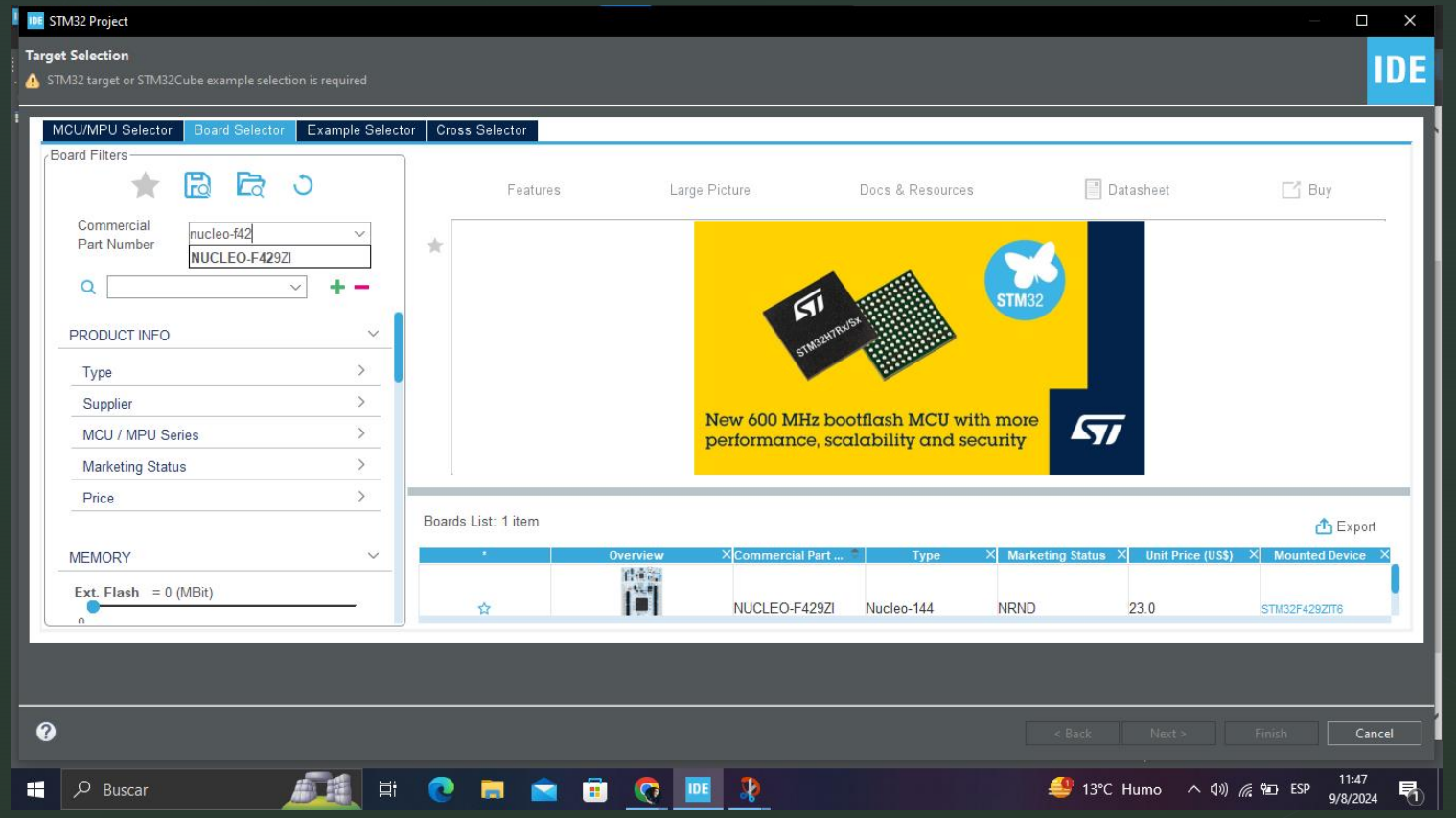
# En el caso de que queramos cambiar nuestro workspace voy a file y busco la opción q dice switch workspace



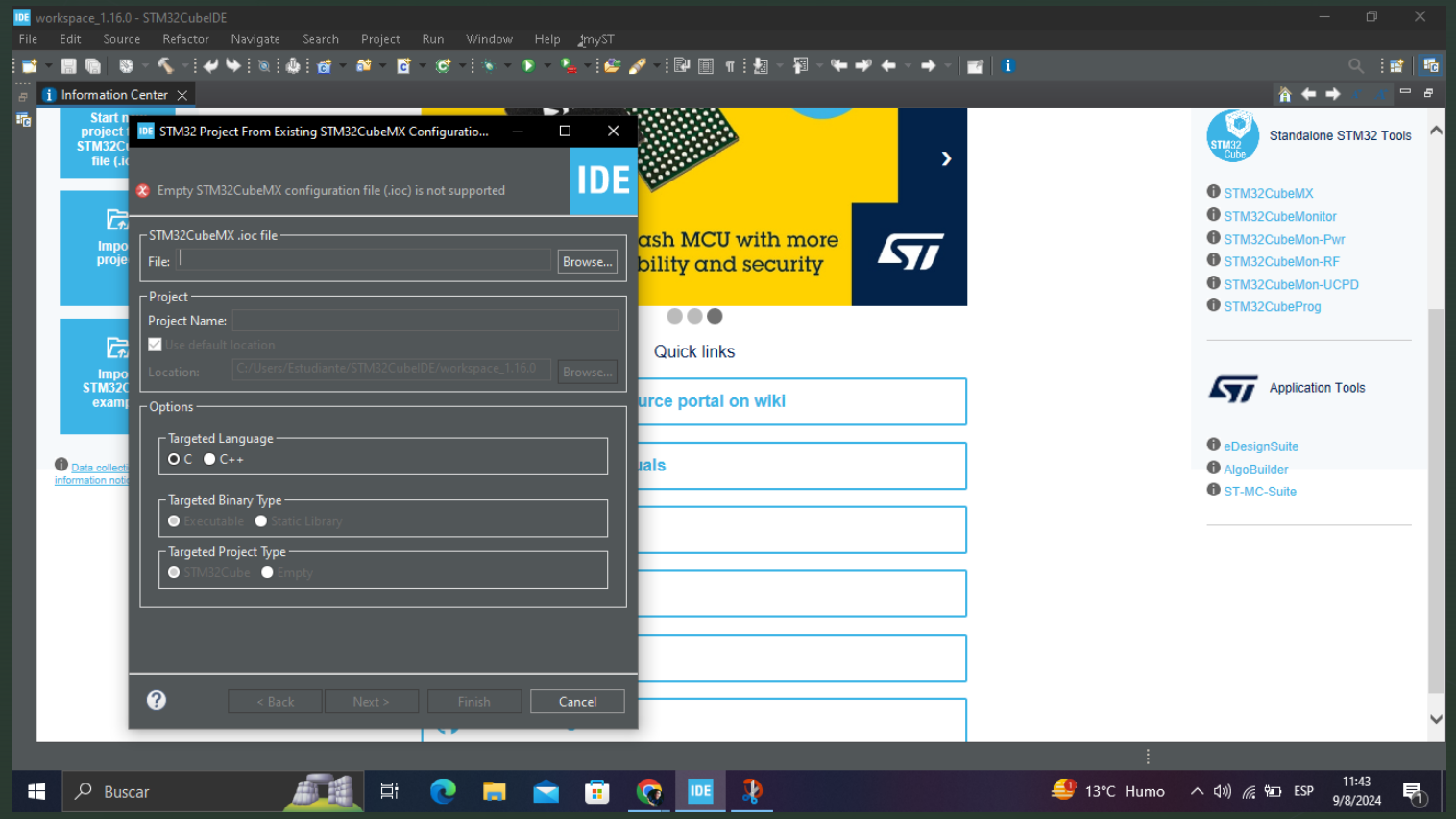
# Para comenzar con la creación de un proyecto vamos a file, luego a new y seleccionamos donde dice STM32 proyect



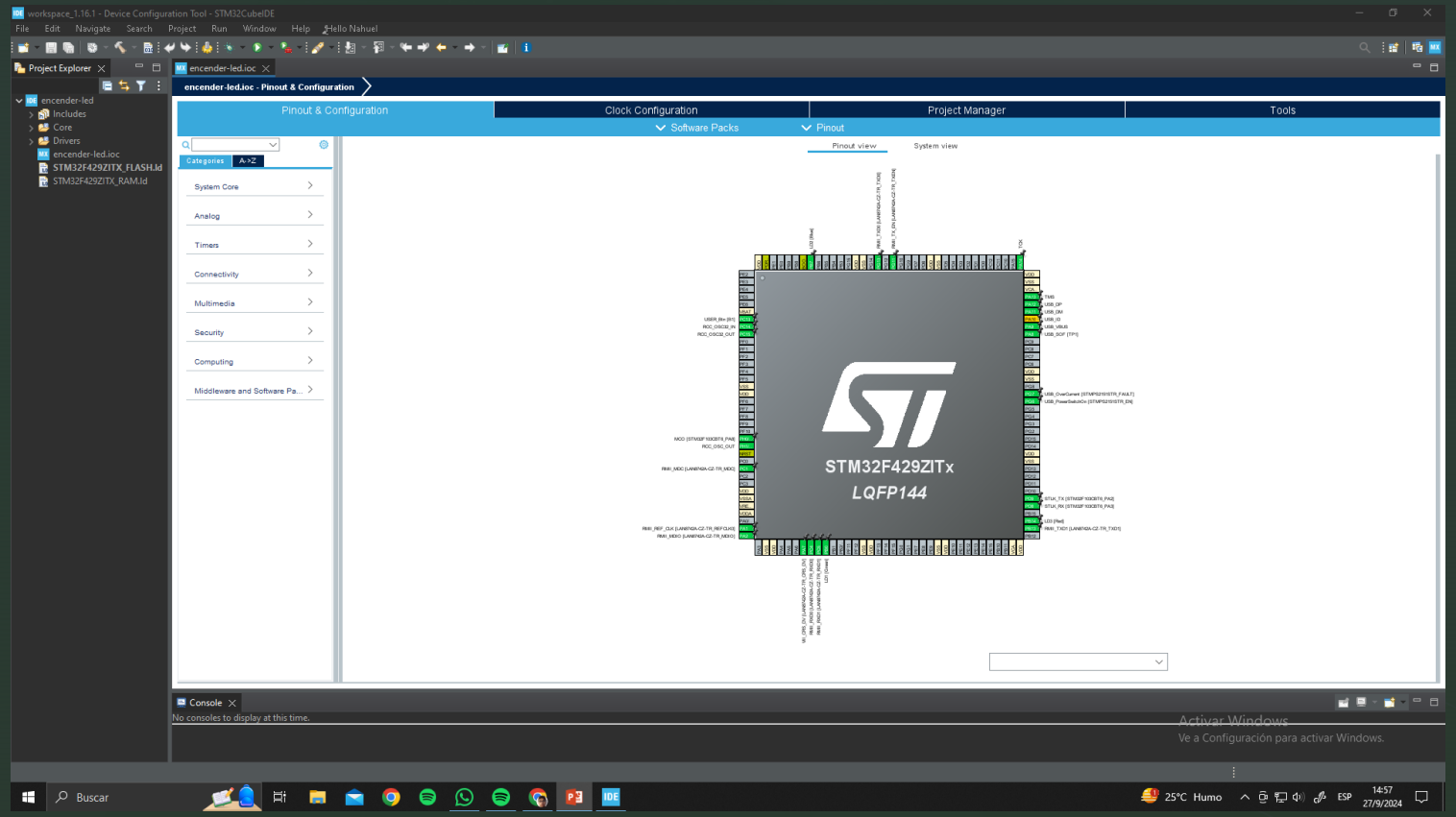
# El siguiente paso es elegir la placa de desarrollo con la que vamos a trabajar



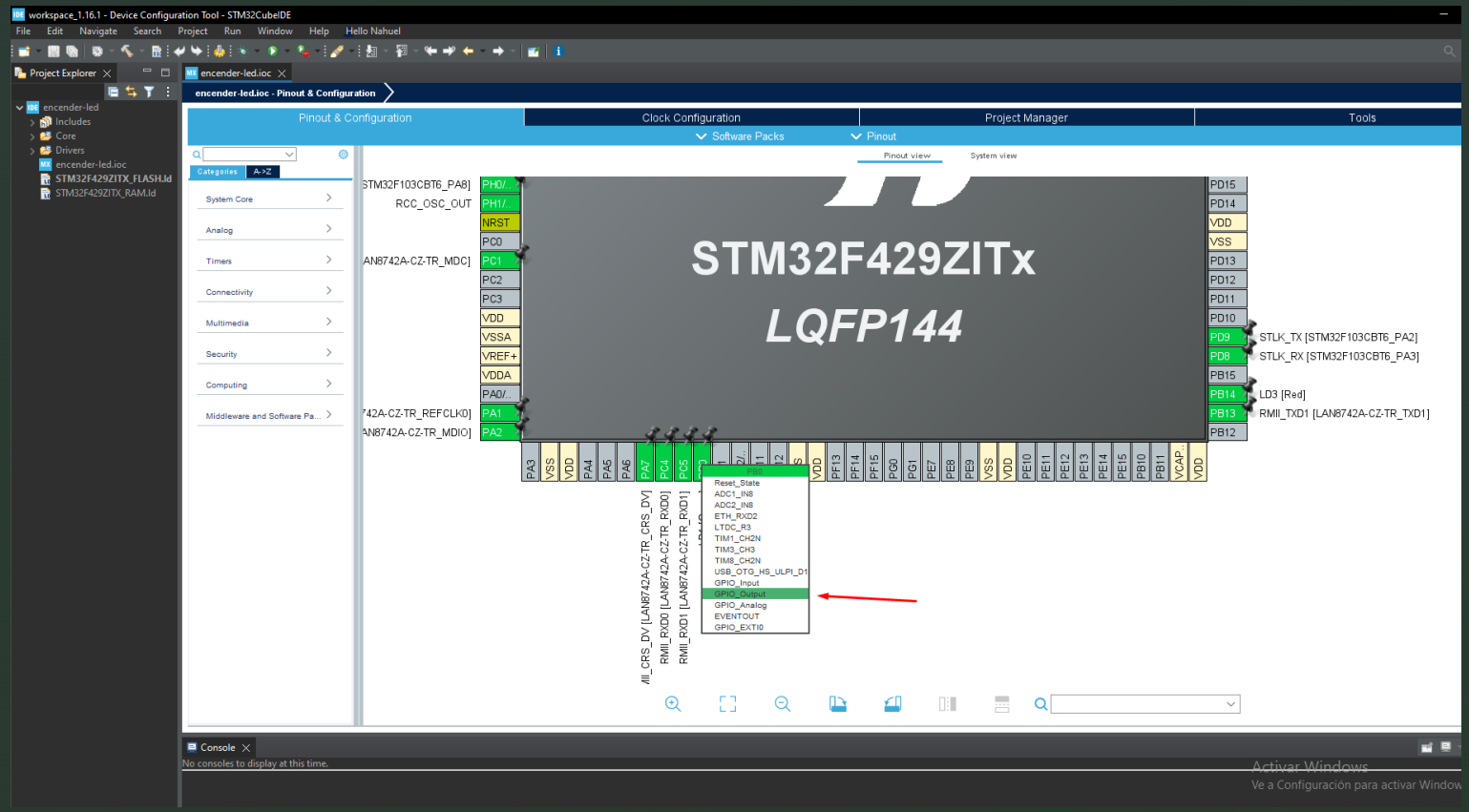
# Aquí ingresamos el nombre del proyecto y elegimos en que lenguaje vamos a trabajar además el tipo de proyecto que vamos a realizar



- Una vez creado el proyecto se abrirá una intefaz grafica, en ella vamos a configurar los pines a utilizar

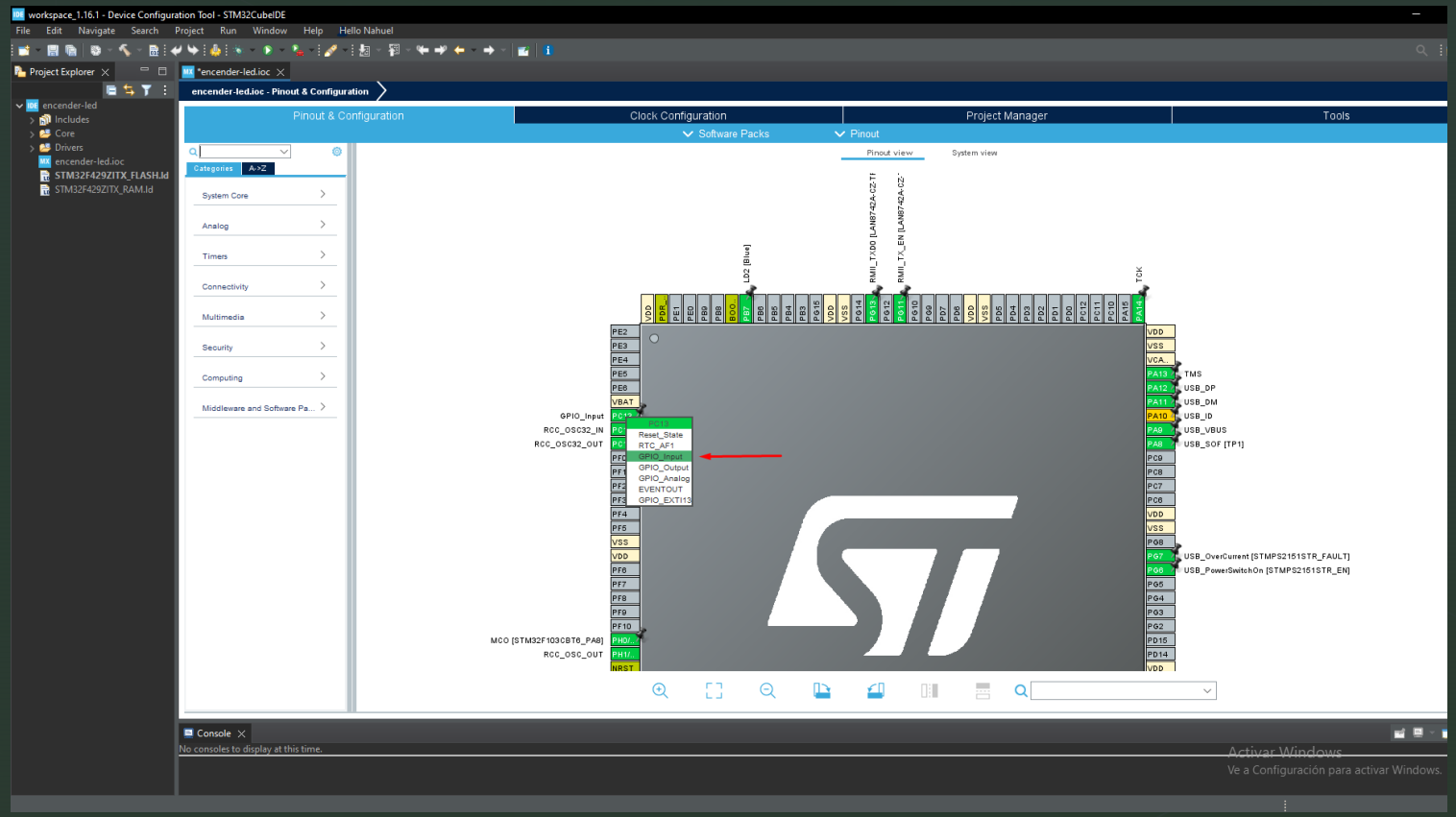


LED1= pin PB0, LED2= pin PB7, LED3= pin PB14 hacemos clic en el pin a configurar y seleccionamos GPIO_OutPut.

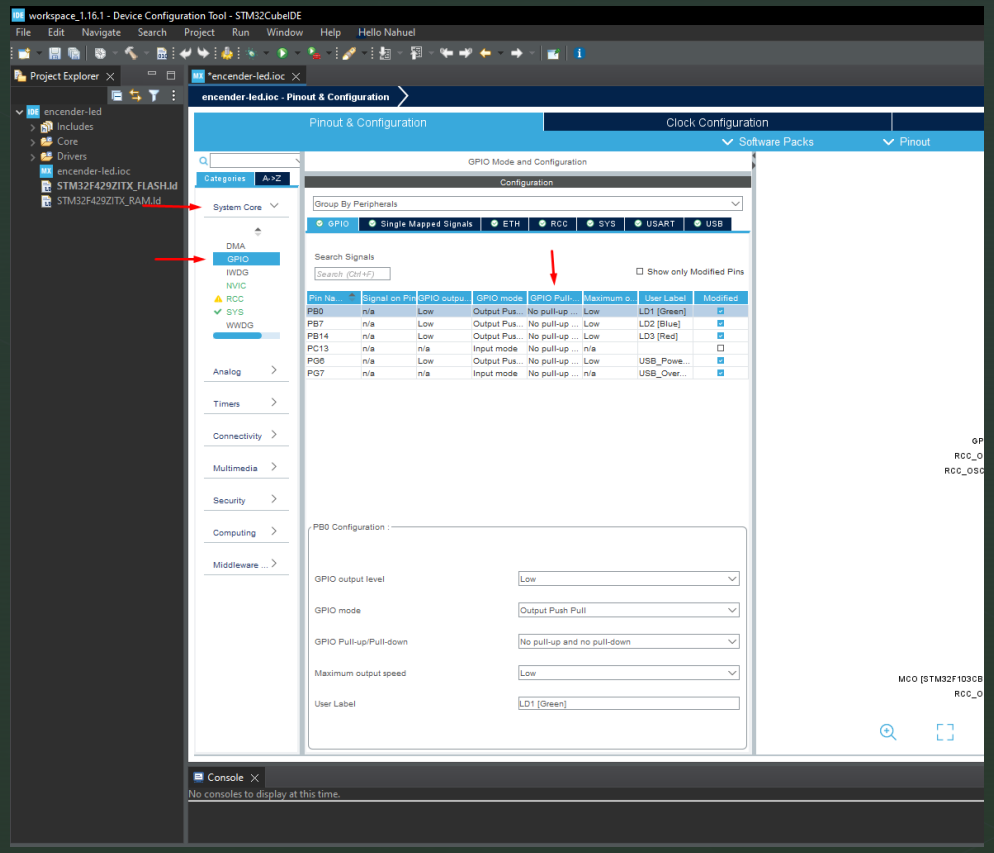


El pulsador de usuario B1 esta conectado al pin PC13

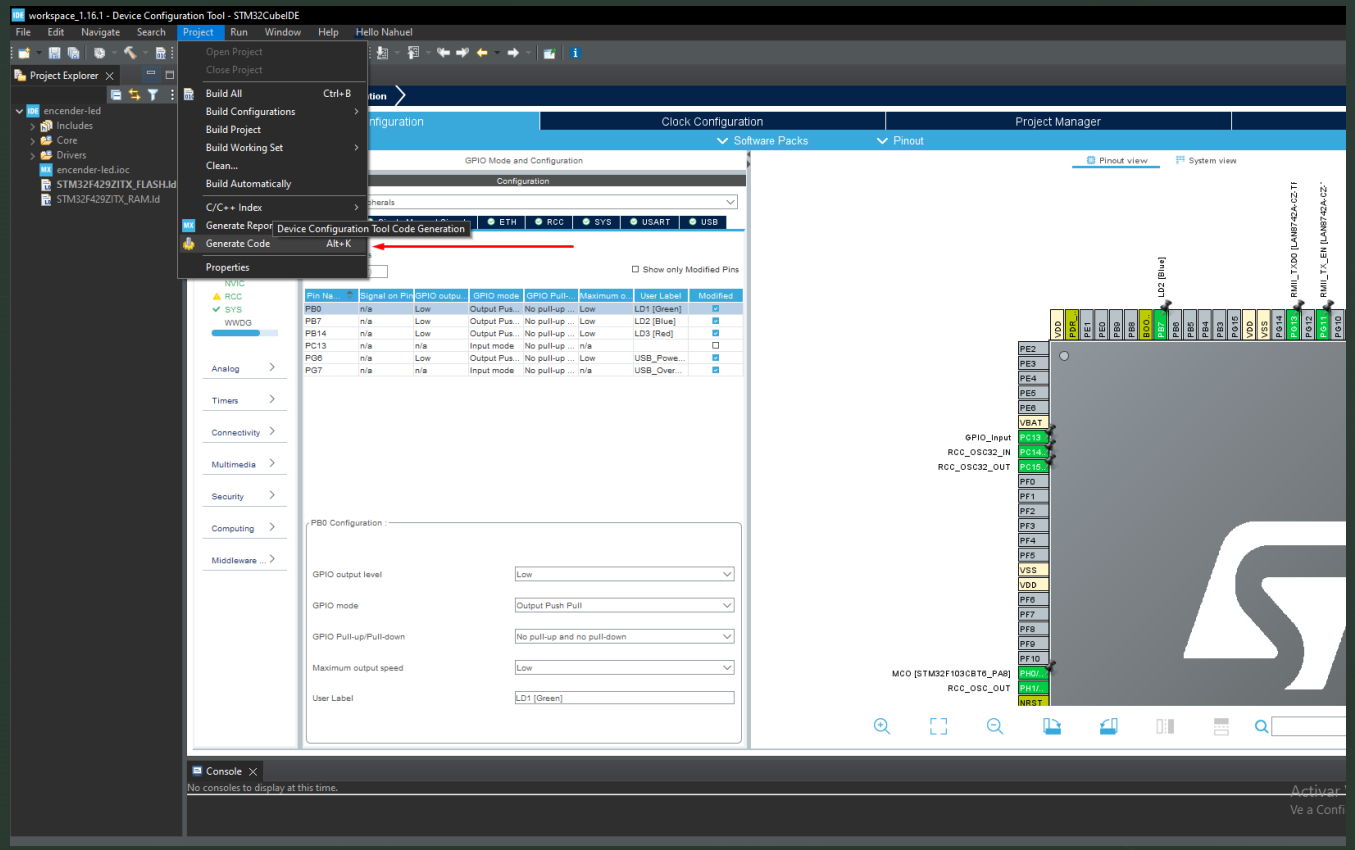
En la configuración de pines, se hace clic en el pin PC13 y seleccionamos GPIO_OUTPUT.



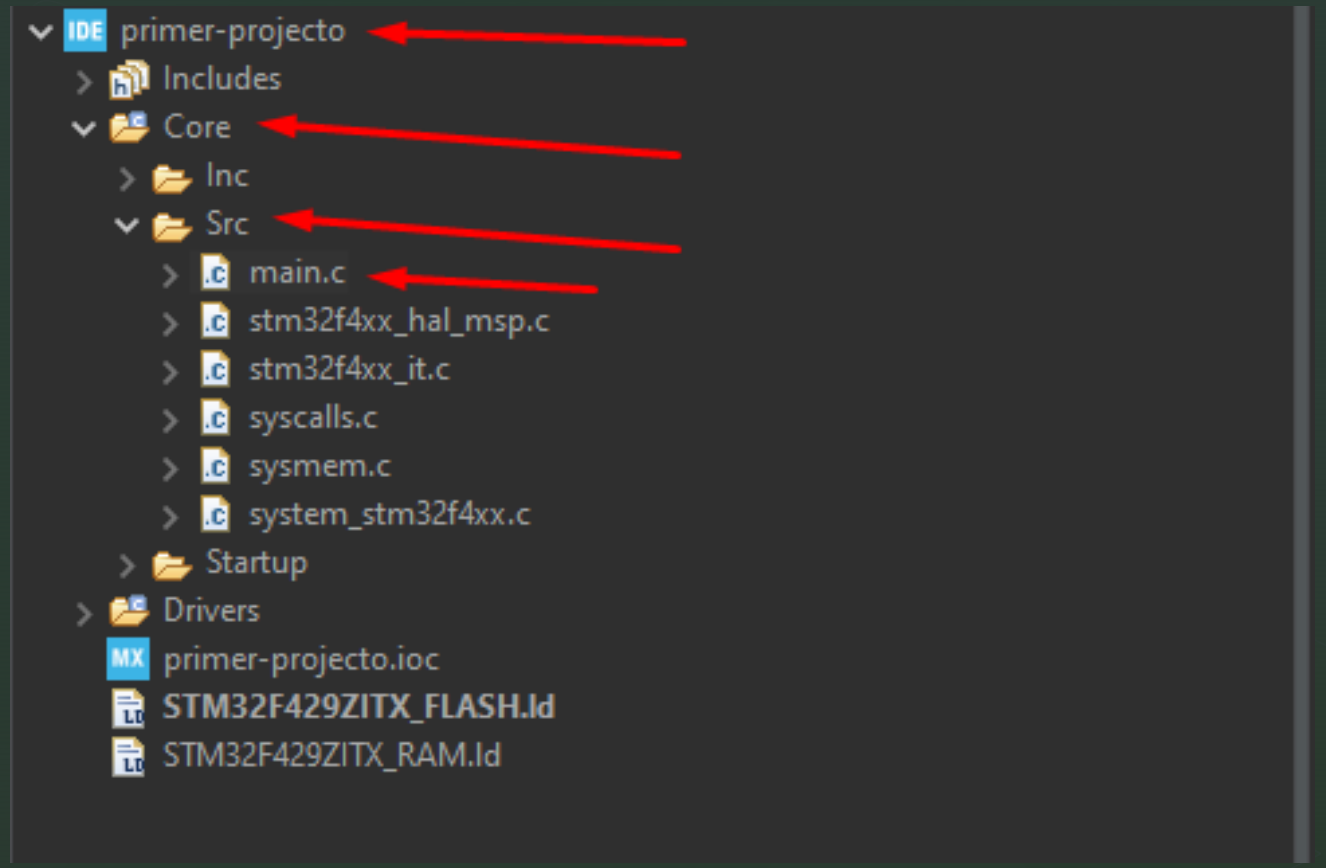
- Configuración adicional, en el panel de configuración de GPIO, nos aseguramos de que los modos de los pines estén configurados correctamente: LED:GPIO_OutPut con Push-Pull Pulsador:GPIO_InPut



# En el apartado “Project” seleccionamos generate code



# Navegamos hasta el archivo main.c en la raíz del proyecto



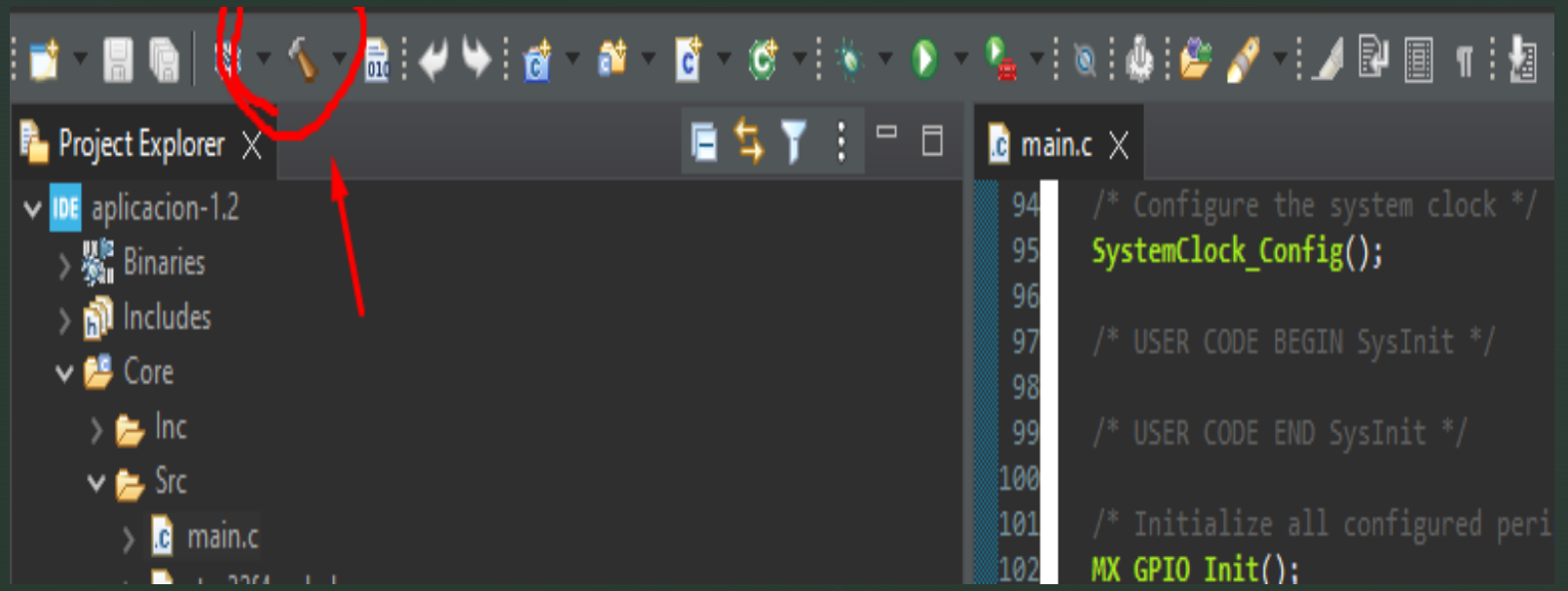
Dentro de main.c debemos realizar nuestro código dentro de los apartados indicados como se ve en la imagen

```
88    HAL_Init();
89
90    /* USER CODE BEGIN Init */
91
92    /* USER CODE END Init */
93
112   while (1)
113   {
114     /* USER CODE END WHILE */
115
116     /* USER CODE BEGIN 3 */
117   }
118   /* USER CODE END 3 */
```

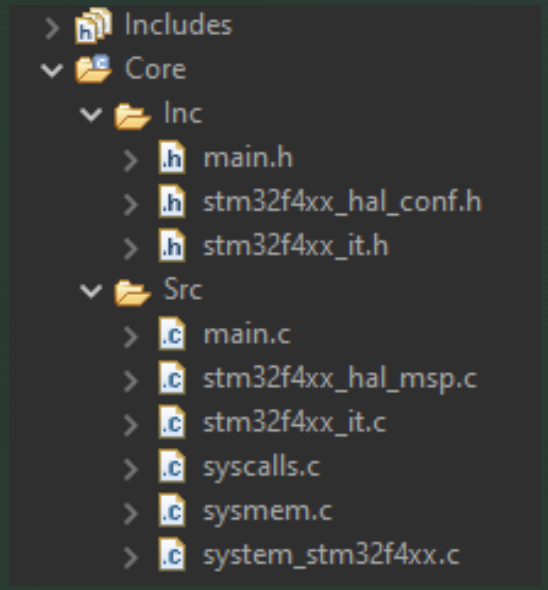
Este código hará que el led 1 de la placa parpadeé cada 250 milisegundos

```
112   while (1)
113   {
114     /* USER CODE END WHILE */
115           HAL_GPIO_WritePin(D1_GPIO_Port, LD1_Pin, GPIO_PIN_SET);
116           HAL_Delay(250);
117           HAL_GPIO_WritePin(D1_GPIO_Port, LD1_Pin, GPIO_PIN_RESET);
118           HAL_Delay(250);
119     /* USER CODE BEGIN 3 */
120   }
121   /* USER CODE END 3 */
```

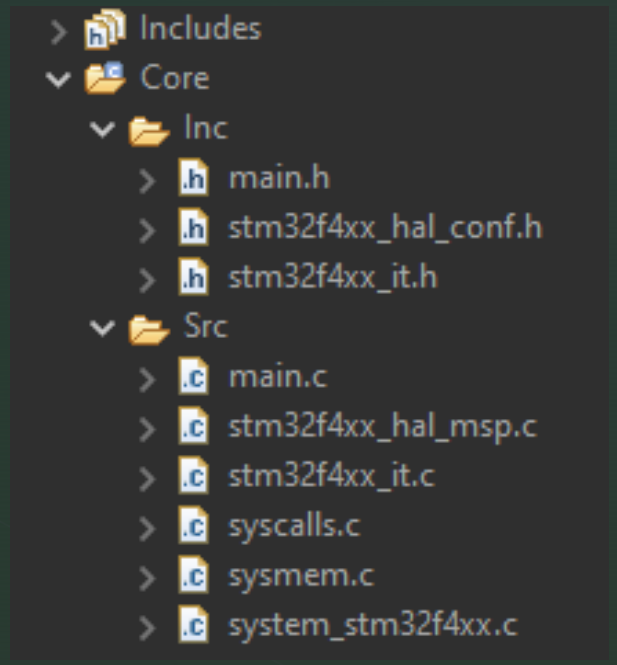
# Para finalizar damos en build



# Estructura del Árbol de archivos en STM32CubeIDE




1. Includes
○ Esta carpeta suele contener los archivos de cabecera (headers) comunes a todo el proyecto.
2. Core
○ Inc
■ main.h: Archivo de cabecera principal del proyecto. Suele contener definiciones globales, prototipos de funciones y declaraciones de variables.
■ stm32f4xx_hal_conf.h: Archivo de configuración de la HAL (Hardware Abstraction Layer) específico para la familia de microcontroladores STM32F4.
■ stm32f4xx_it.h: Archivo de cabecera que declara los manejadores de interrupciones para el microcontrolador STM32F4.




○ Src

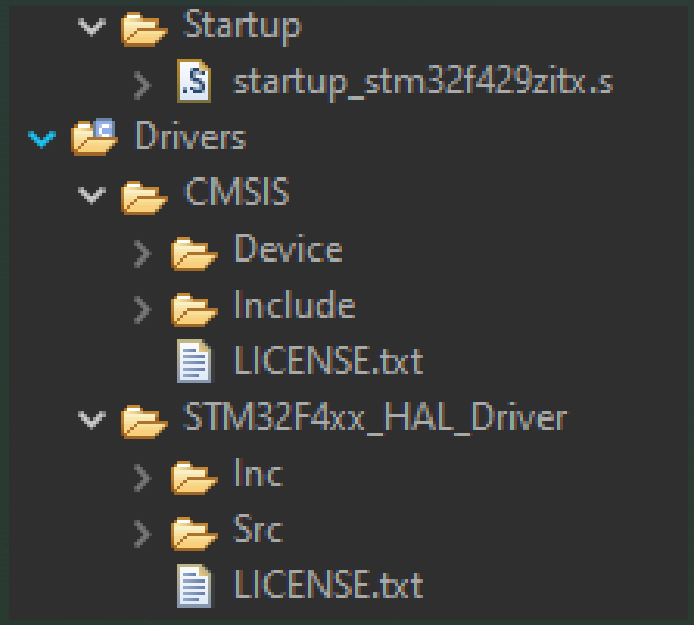
■ main.c: Archivo fuente principal del proyecto. Contiene la función main y el flujo principal del programa.

■ stm32f4xx_hal_msp.c: Archivo fuente que contiene funciones de inicialización del Sistema de Soporte del Microcontrolador (MSP).

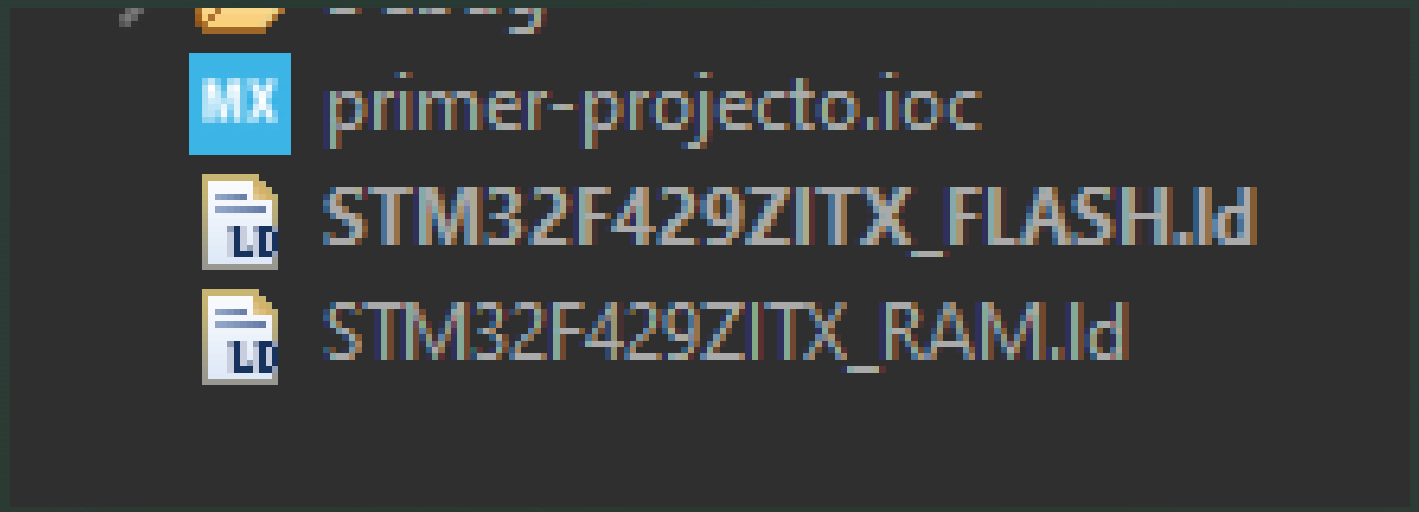
■ stm32f4xx_it.c: Archivo fuente que implementa los manejadores de interrupciones declarados en stm32f4xx_it.h.

■ syscalls.c: Archivo fuente que implementa las llamadas al sistema (system calls) para la integración con las librerías estándar de C.

■ sysmem.c: Archivo fuente que implementa la gestión de memoria para el proyecto. ■ system_stm32f4xx.c: Archivo fuente que inicializa el sistema y el reloj del microcontrolador.




3. Startup
○ startup_stm32f429zitx.s: Archivo ensamblador que contiene el código de arranque (startup) para el microcontrolador STM32F429ZITx.
4. Drivers
○ CMSIS
■ Esta carpeta contiene los archivos del CMSIS (Cortex Microcontroller Software Interface Standard), una librería estándar para microcontroladores Cortex-M.
○ STM32F4xx_HAL_Driver
■ Inc: Archivos de cabecera para los drivers HAL específicos para la serie STM32F4.
■ Src: Archivos fuente que implementan los drivers HAL específicos para la serie STM32F4.
■ LICENSE.txt: Archivo que contiene la licencia de uso de los drivers HAL.



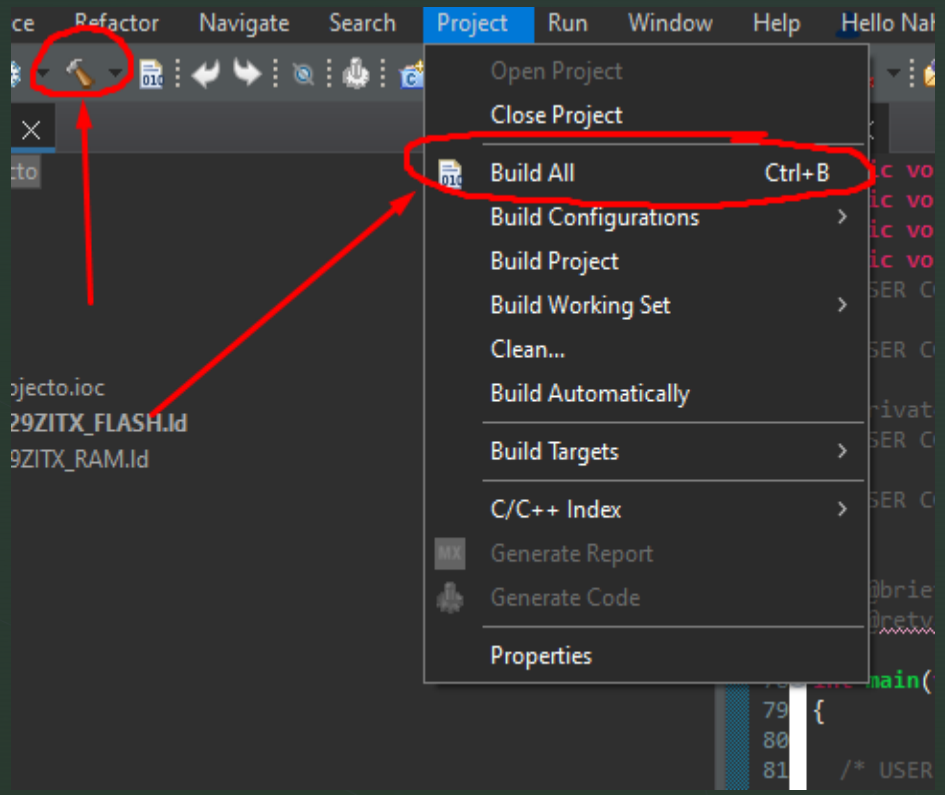

5. primer-proyecto.ioc
○ Archivo de configuración del proyecto generado por STM32CubeMX. Contiene toda la configuración de los periféricos y pines del microcontrolador.
6. STM32F429ZITX_FLASH.ld
○ Archivo de script de enlace (linker script) que define la organización de la memoria FLASH del microcontrolador.
7. STM32F429ZITX_RAM.ld
○ Archivo de script de enlace (linker script) que define la organización de la memoria RAM del microcontrolador.

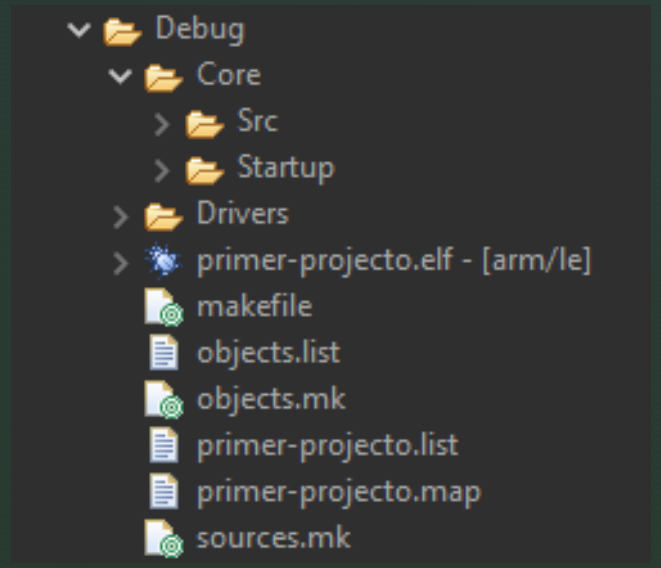
# Compilación del Proyecto en STM32CUBEIDE

- 1. Seleccionar el Proyecto: En el panel de Project Explorer, seleccionar el proyecto que se desea compilar.
- 2. Compilar el Proyecto: Se puede compilar el proyecto de dos maneras:

○ Haciendo clic derecho en el proyecto y seleccionando Build Project.

○ O usando el atajo de teclado Ctrl + B.





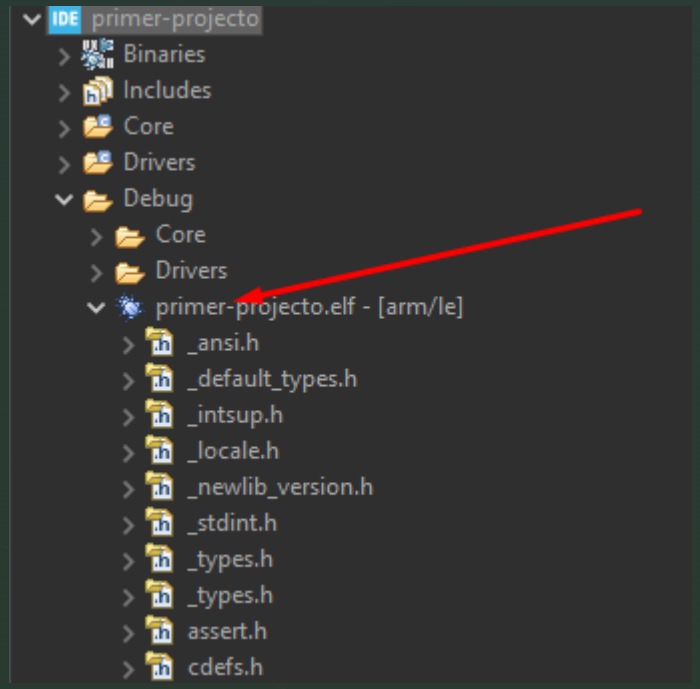

Archivos Generados

Durante la compilación, STM32CUBEIDE genera varios archivos, incluyendo el archivo objeto y otros archivos de salida:

1. Código Objeto:

○ Archivo OBJ: Los archivos objeto tienen la extensión .o (o .obj en algunos sistemas). Estos archivos contienen el código máquina generado a partir de tu código fuente y son usados por el enlazador para crear el archivo binario final.

○ Ubicación: Los archivos objeto se encuentran en el directorio Debug o Release, dependiendo del modo de compilación. En la estructura de archivos de tu proyecto, la ruta suele ser algo como:




Archivo Binario Final:

- Archivo ELF: El archivo final es el archivo ejecutable en formato ELF (.elf). Este archivo contiene el código ejecutable y datos necesarios para la programación del microcontrolador.

- Ubicación: El archivo ELF se encuentra en el mismo directorio Debug o Release.

# Configuración del Debugger en STM32CubeIDE y Descarga del Código Objeto a la Placa Target

1. Conectar la Placa:

○ Asegurarse de que la placa NUCLEO-F429ZI esté conectada a la computadora a través del cable micro USB. Verificar que la placa sea detectada por el sistema operativo.

2. Abrir el Proyecto:

○ Abrir el proyecto en el que se está trabajando en STM32CubeIDE.

# Descarga del Código Objeto a la Placa Target

Una vez que se haya configurado el debugger, el siguiente paso es descargar el código objeto (archivo .elf) a la placa NUCLEO-F429ZI:

1. Compilación del Proyecto:

○ Asegurarse de que el proyecto esté compilado correctamente. Para hacerlo, seleccionar Project > Build Project o presiona Ctrl + B.

2. Cargar el Código en la Placa:

○ Con la placa conectada y detectada por STM32CubeIDE, seleccionar Run > Debug o presiona F11.

○ STM32CubeIDE comenzará el proceso de descarga del código objeto a la placa target. Una vez finalizada la descarga, la ejecución del programa comenzará automáticamente en modo de depuración.

# LINK DE GITHUB

REPOSITORIO: https://github.com/NahuelFSotelo/Grupo-1-TDII-2024..git

App1.1: https://github.com/NahuelFSotelo/Grupo-1-TDII-2024./tree/4e7bc031c2ce53a6ec8f7671349a9cfd007e589f/aplicacion-1.1

App1.2: https://github.com/NahuelFSotelo/Grupo-1-TDII-2024./tree/4e7bc031c2ce53a6ec8f7671349a9cfd007e589f/aplicacion-1.2

App1.3: https://github.com/NahuelFSotelo/Grupo-1-TDII-2024./tree/4e7bc031c2ce53a6ec8f7671349a9cfd007e589f/aplicacion-1.3

App1.4: https://github.com/NahuelFSotelo/Grupo-1-TDII-2024./tree/4e7bc031c2ce53a6ec8f7671349a9cfd007e589f/aplicacion-1.4