

Proyecto básico en Quartus

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2025/01/13/proyecto-basico-en-quartus/>

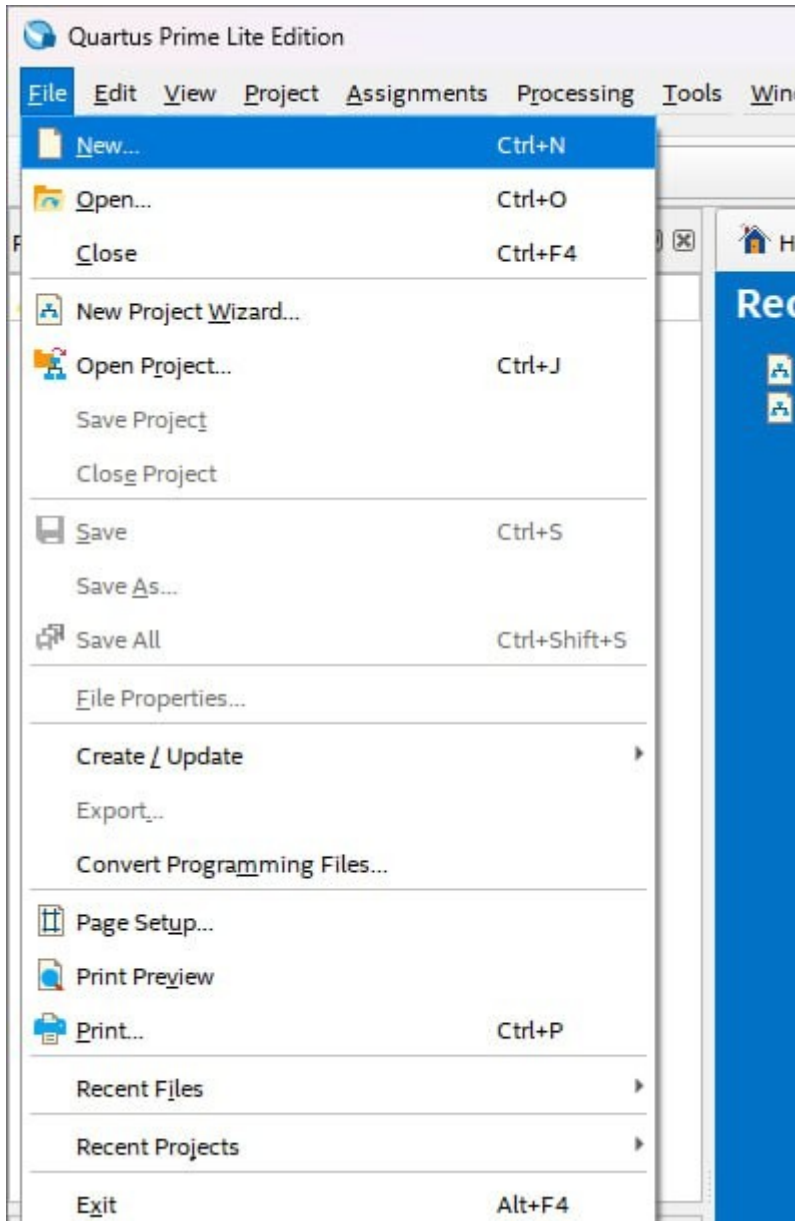
Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

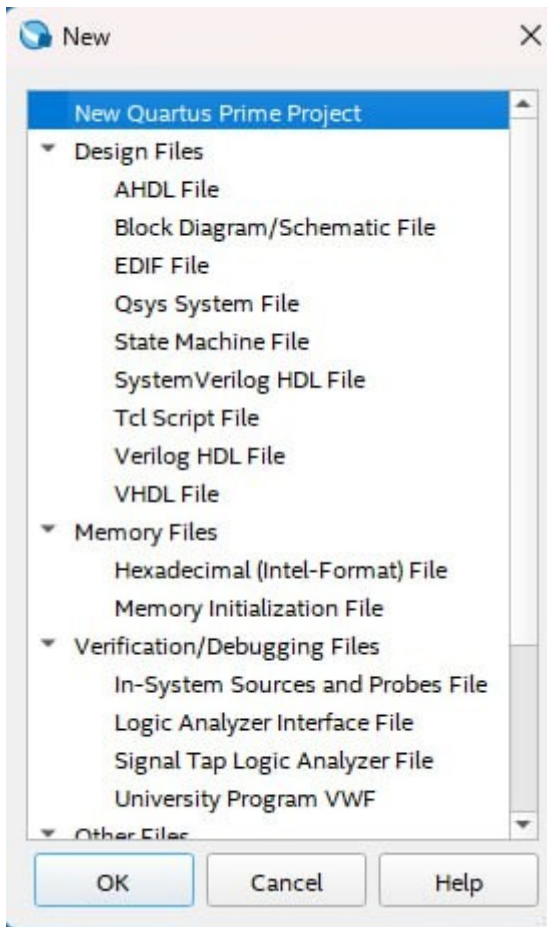
Fecha última modificación: 24/02/2025

En esta entrada se explicará cómo crear un proyecto básico en Quartus.

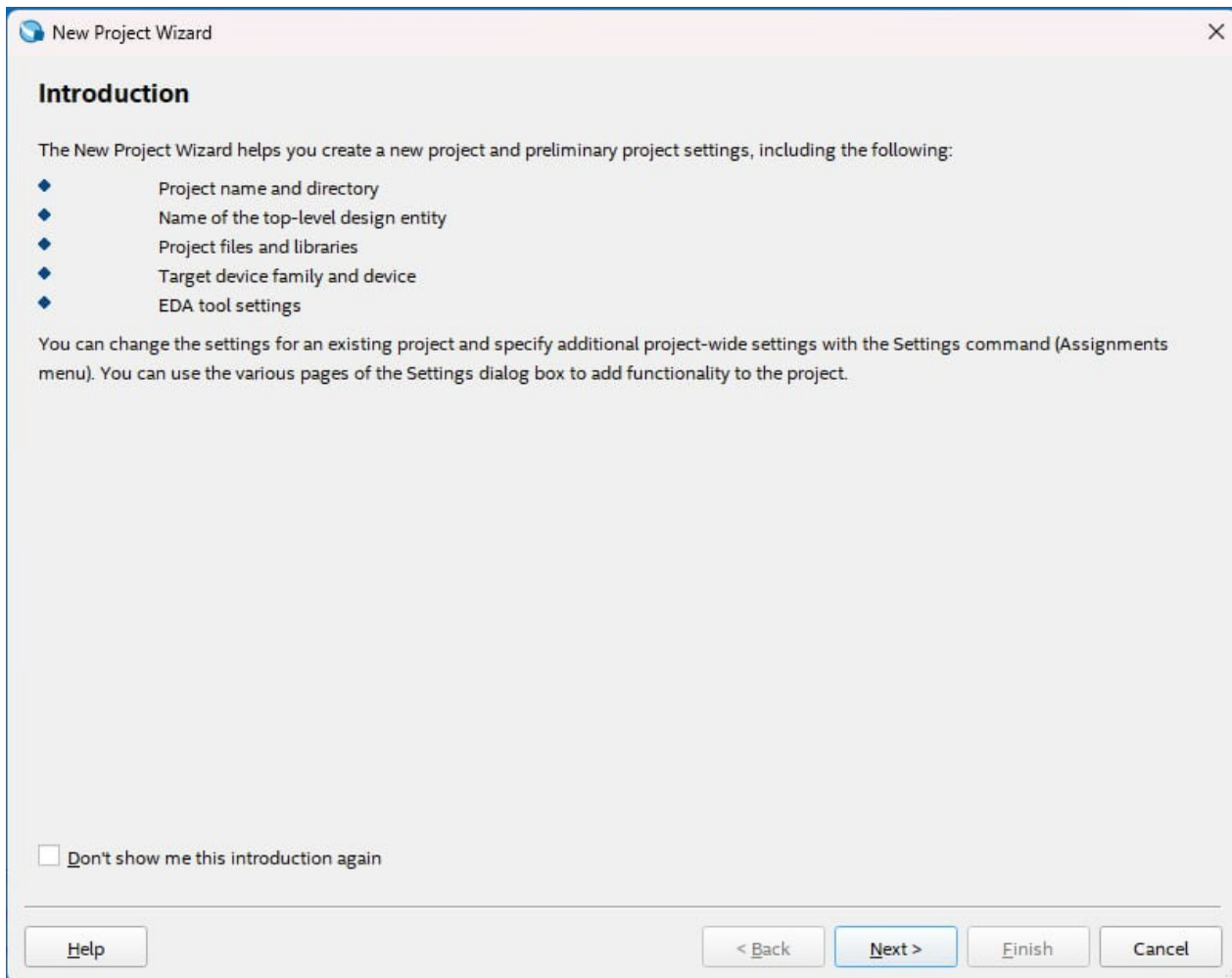
Para empezar, lo primero que hay que hacer es crear un nuevo proyecto, para ello se va a *File* y después a *New...*



Ahora se nos abre una pestaña donde nos pregunta qué queremos crear. Le decimos que un nuevo proyecto en Quartus (*New Quartus Prime Project*).



Después se nos abre el *Project Wizard*. Aquí se configura la estructura del nuevo proyecto. Le damos a Next.



Ahora nos pregunta por el directorio en el que lo vamos a crear y el nombre del proyecto y del fichero top, se recomienda que se llamen igual.

New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/Users

What is the name of this project?

leds

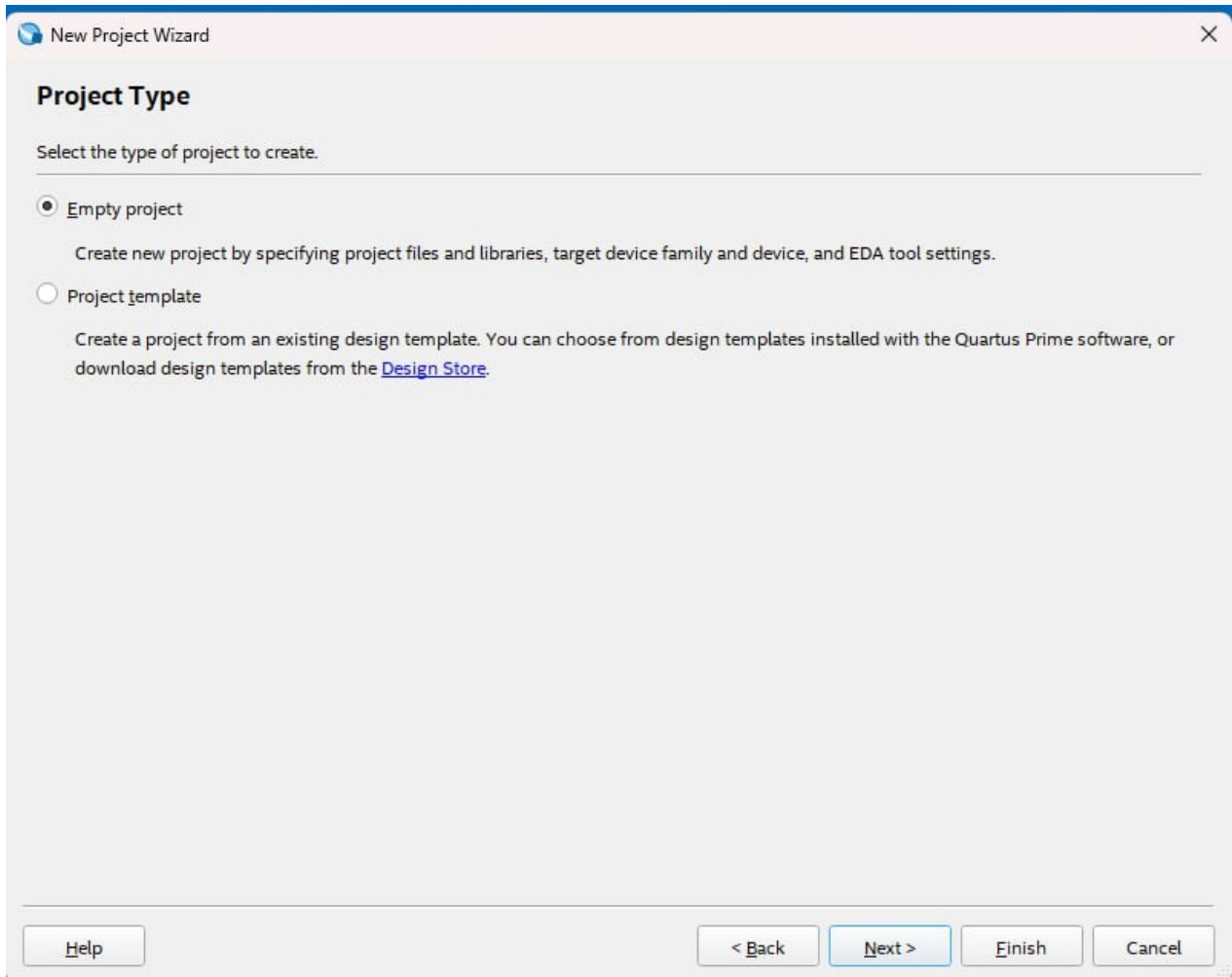
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

leds

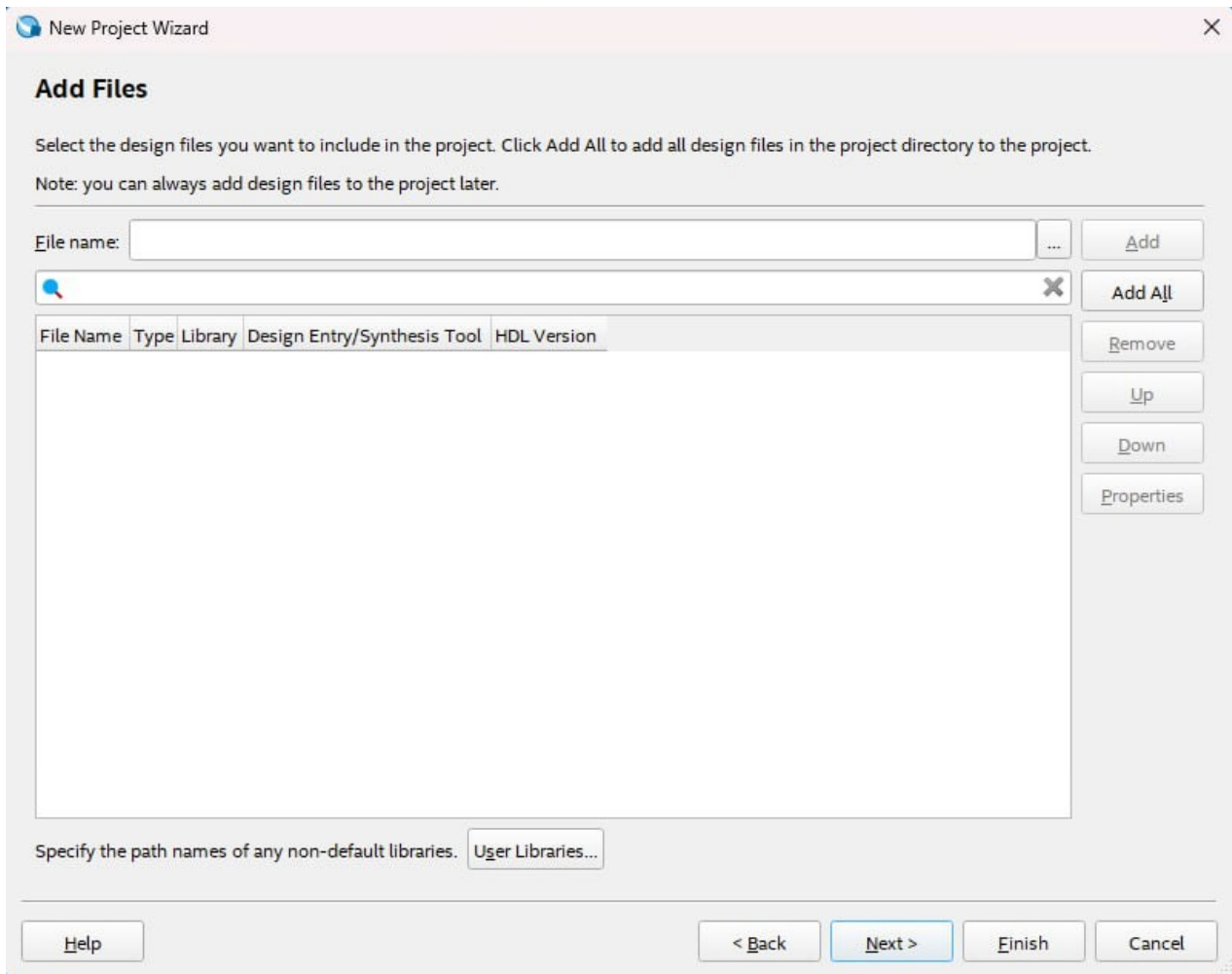
Use Existing Project Settings...

Help < Back Next > Finish Cancel

Después, nos pregunta como queremos crear el proyecto, sobre un proyecto vacío o sobre una proyecto plantilla.



Ahora nos pregunta si queremos añadir ficheros al proyecto.



El siguiente paso es elegir la familia de la FPGA que se quiere programar.

New Project Wizard

Family, Device & Board Settings

Device | Board

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone V (E/GX/GT/SX/SE/ST)
Device: All

Target device

☐ Auto device selected by the Fitter
☒ Specific device selected in 'Available devices' list
☐ Other: n/a

Show in 'Available devices' list

Package: Any
Pin count: Any
Core speed grade: Any
Name filter:
☒ Show advanced devices

Available devices:

Name	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Channel PMA	GXB Channel PCS	PC
5CGXFC7C6F2317	1.1V	56480	268	240	6	6	1

Help < Back Next > Finish Cancel

Ahora nos pregunta si queremos utilizar herramientas externas a Quartus para hacer los análisis, síntesis, etc.

EDA Tool Settings

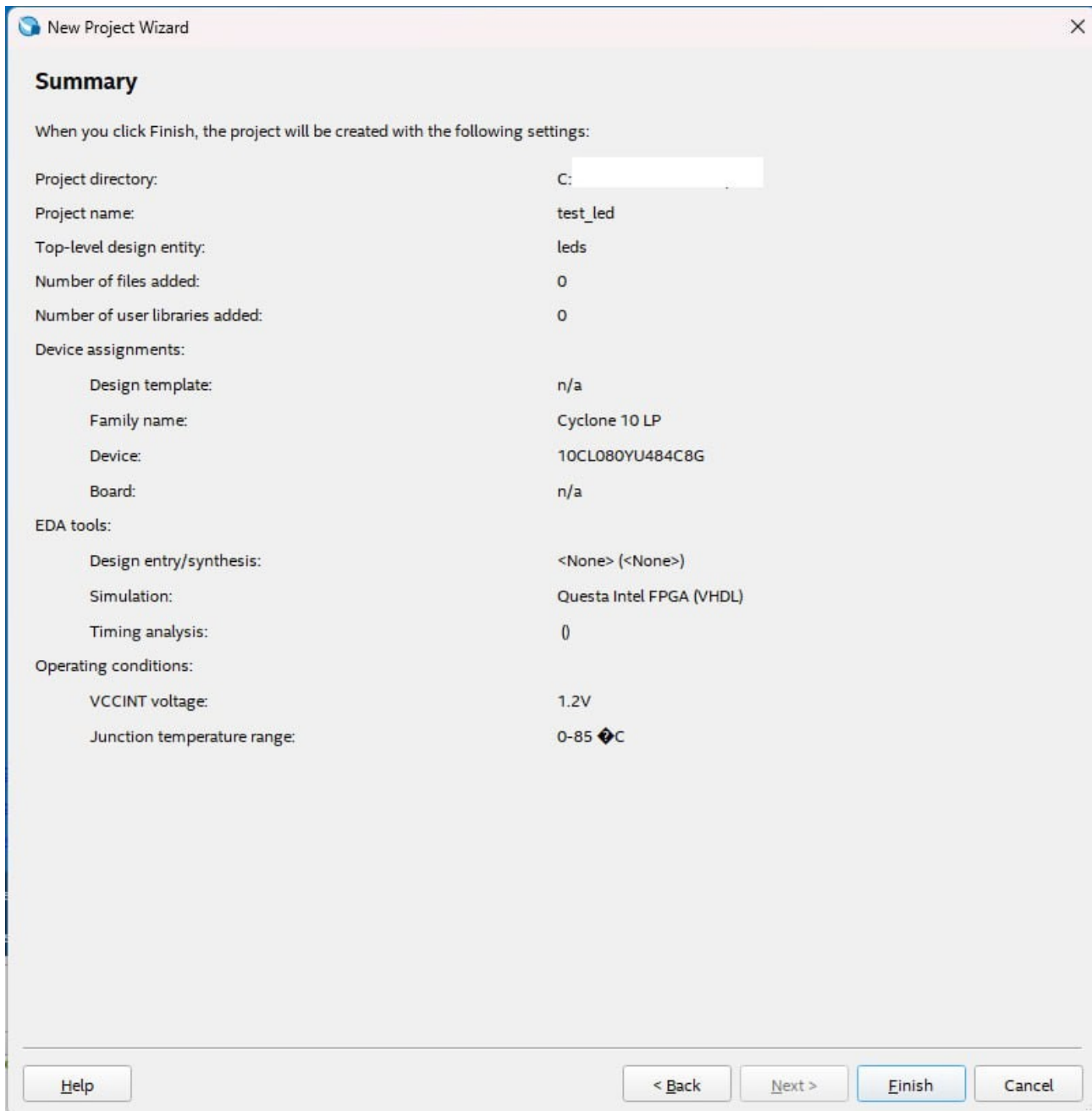
Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

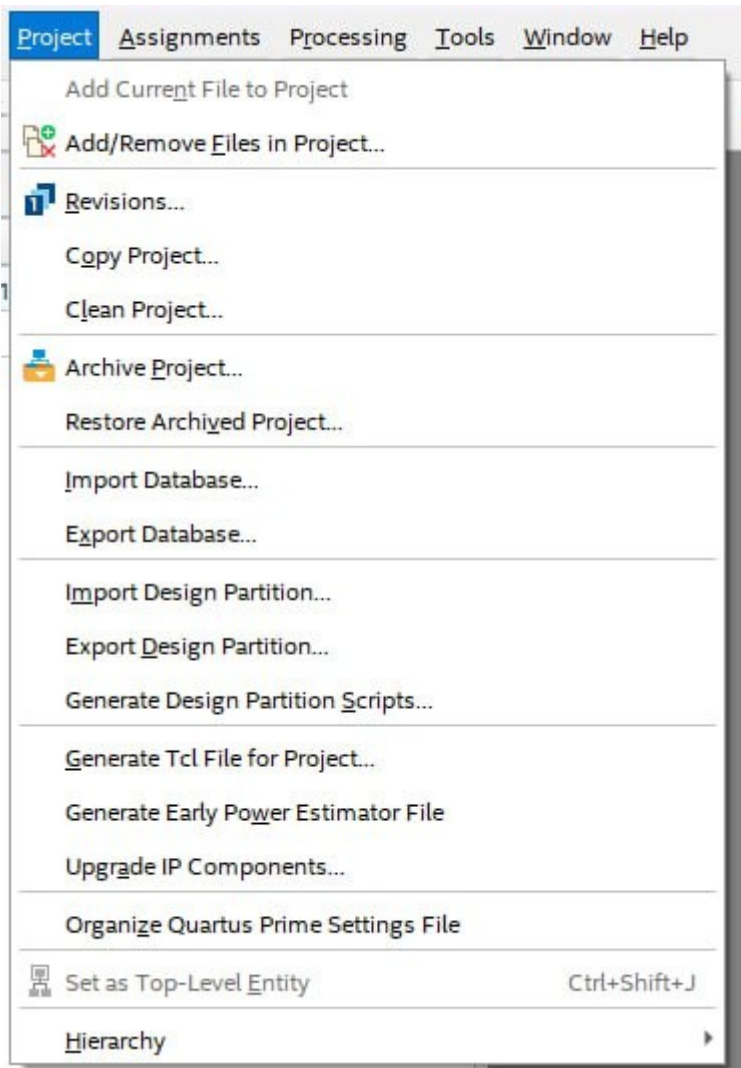
Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synth...	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	Questa Intel FPGA	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

Help < Back Next > Finish Cancel

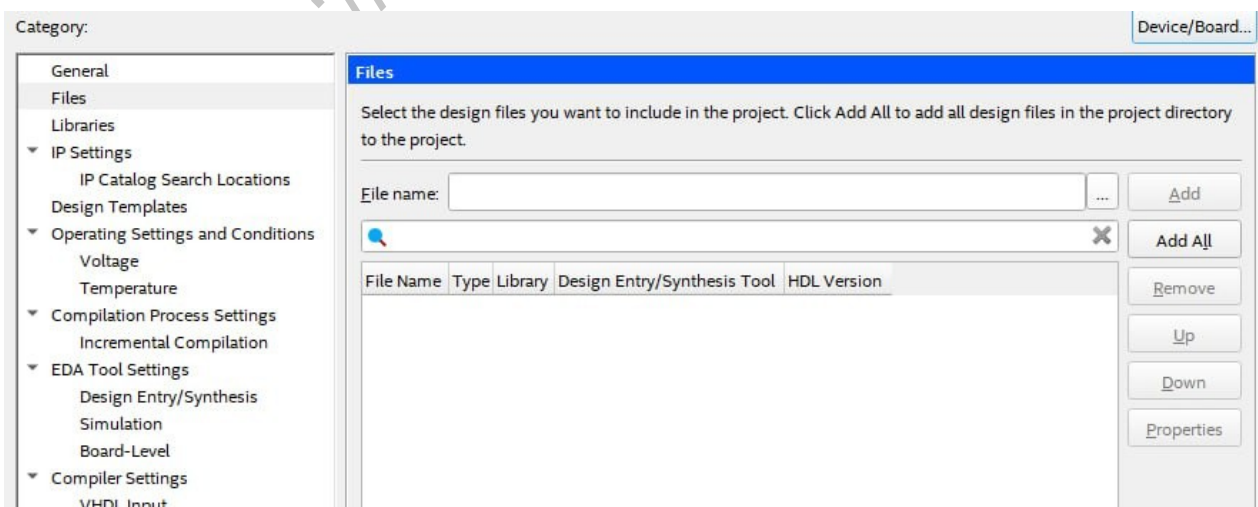
Lo último que nos da es un resumen de la configuración previa.



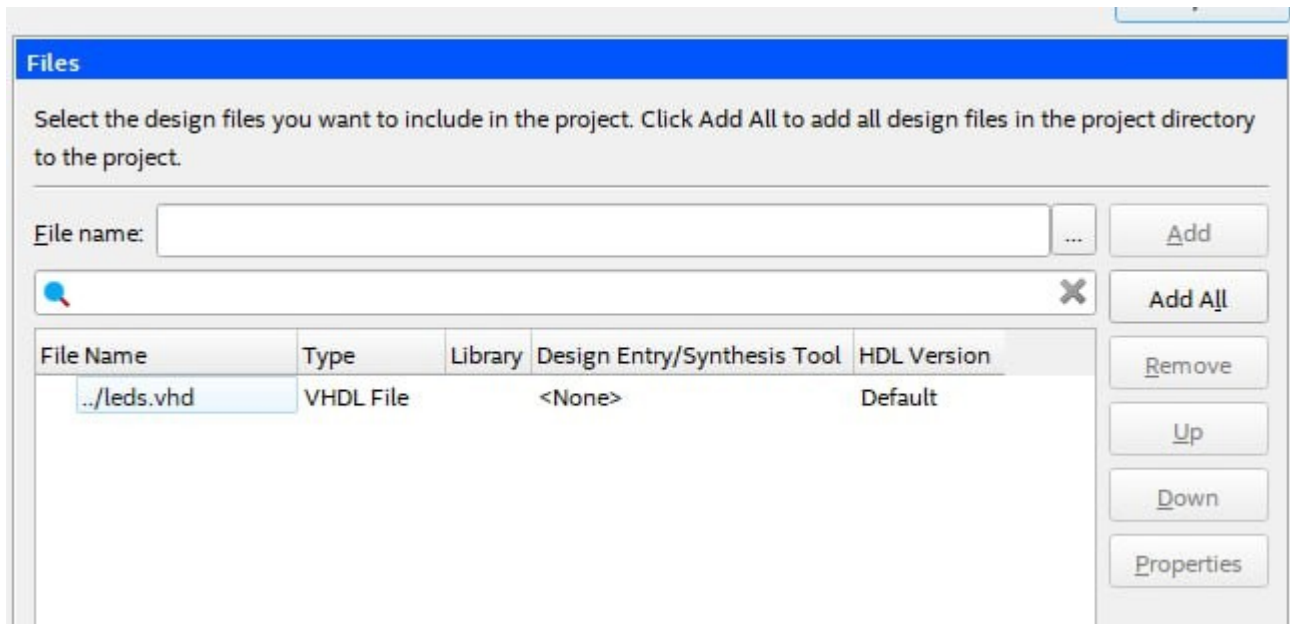
Para importar ficheros al proyecto, lo único que hay que hacer es irse a *Project* y después darle a *Add/Remove Filer in Project...*



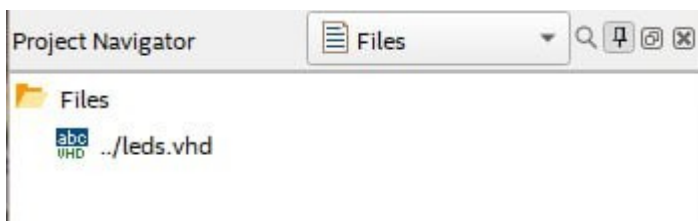
Al darle se nos abre una pestaña donde se puede cambiar la configuración del proyecto. Aquí le damos a los tres puntos al lado de *File name*.



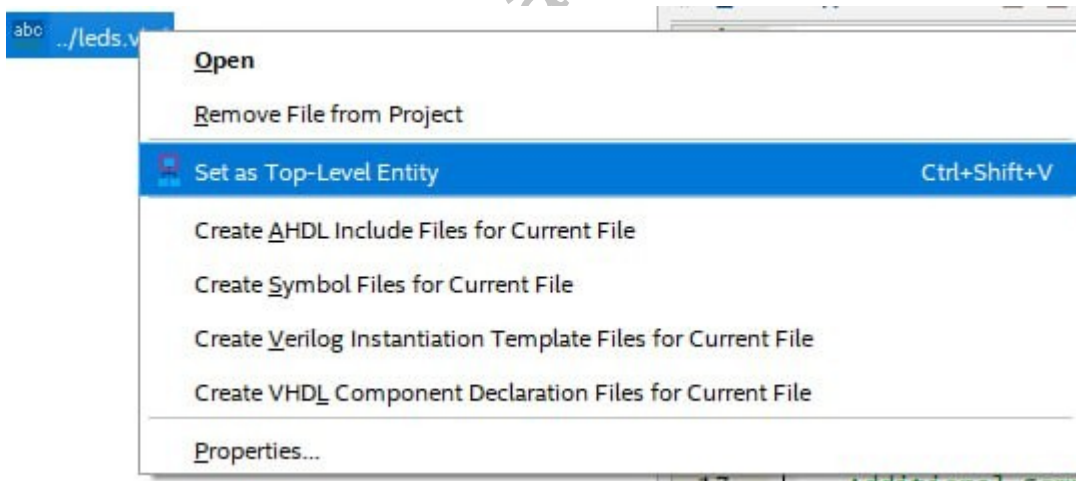
Ahora solo hay que importar el/os ficheros deseados al proyecto.



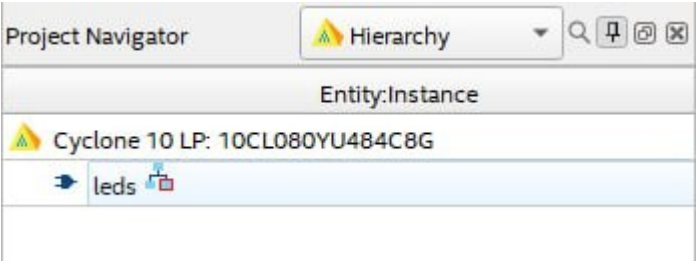
Ahora volvemos a Quartus, y en Files aparece el fichero que hemos introducido.



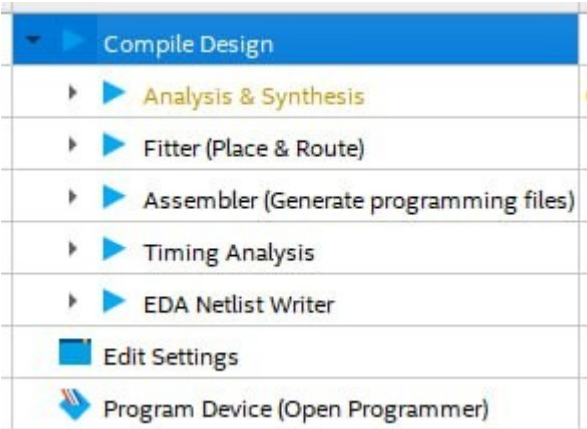
Ahora le damos clic derecho y marcamos la opción de *Set as Top-Level Entity*.



Ahora vamos a *Hierarchy* y podemos ver el nuevo fichero top.



Ahora en la parte inferior aparecen las siguientes opciones de Síntesis e Implementación (Compile Design).



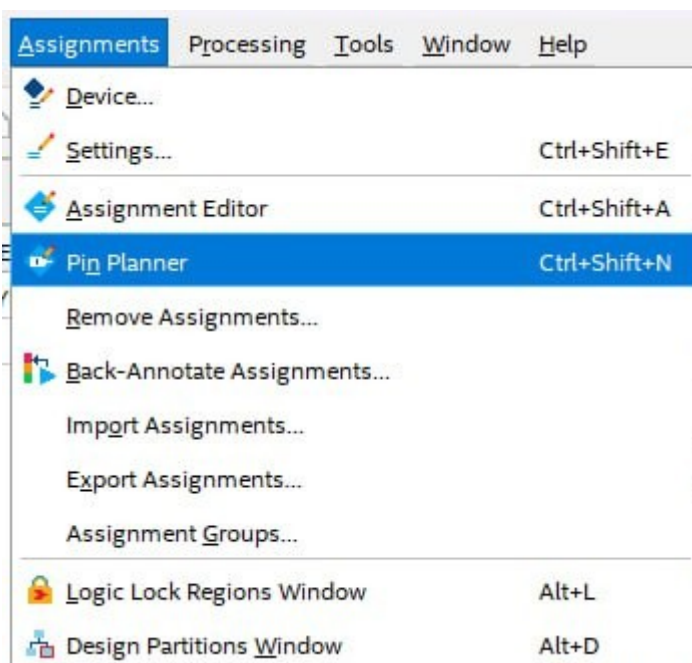
Ahora lo único que habría que hacer es darle a *Compile Design* y este ejecutará todos los análisis.

	Task	Time
✓	▶ ▶ Compile Design	00:00:26
✓	▶ ▶ ▶ Analysis & Synthesis	00:00:10
✓	▶ ▶ ▶ Fitter (Place & Route)	00:00:09
✓	▶ ▶ ▶ Assembler (Generate programming files)	00:00:03
✓	▶ ▶ ▶ Timing Analysis	00:00:03
✓	▶ ▶ ▶ EDA Netlist Writer	00:00:01
	■ Edit Settings	
	■ Program Device (Open Programmer)	

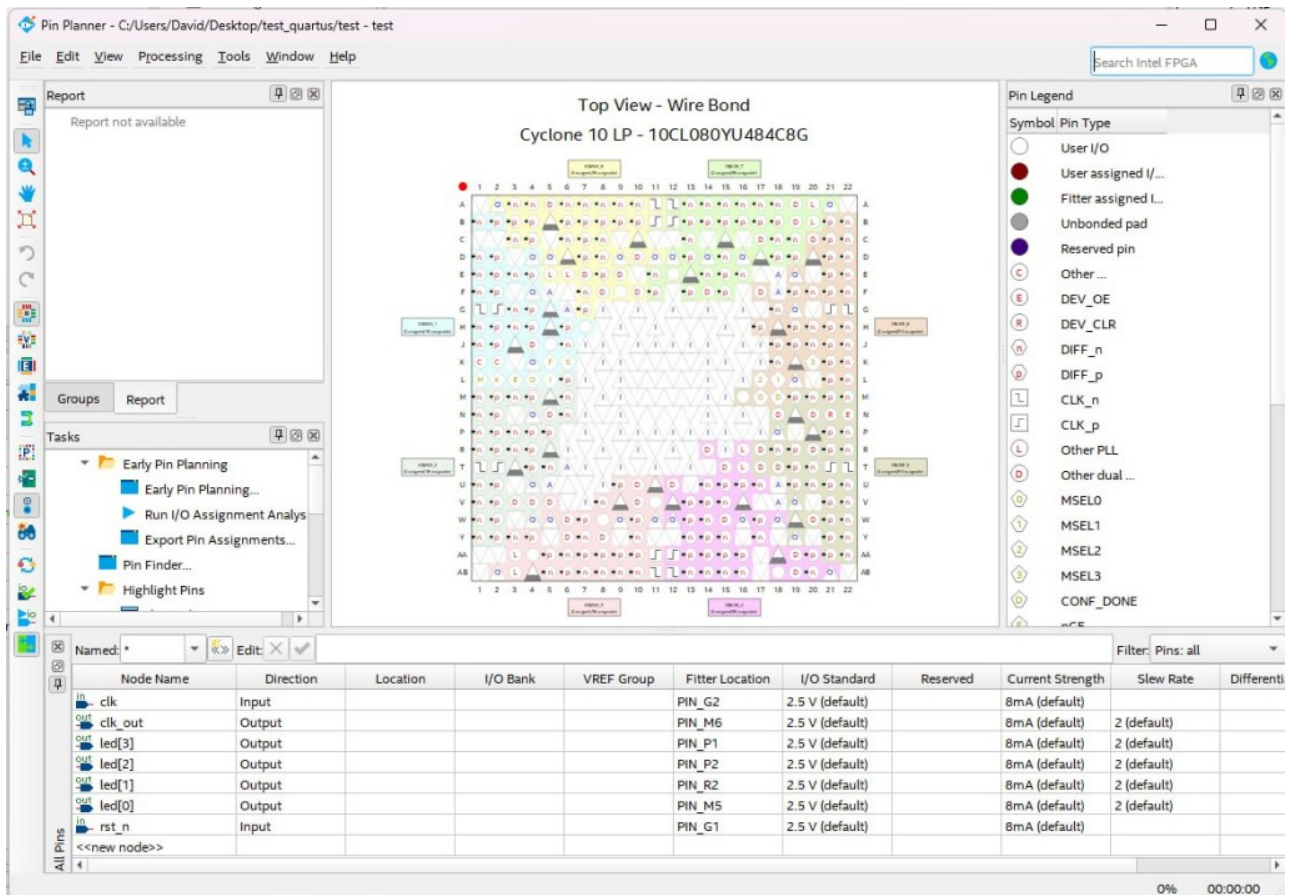
Al terminar aparece en la pestaña los resultados de los diferentes análisis hechos.

Table of Contents		Flow Summary	
Flow Summary		<<Filter>>	
Flow Settings		Flow Status	Successful - Mon Jan 13 19:38:50 2025
Flow Non-Default Global Settings		Quartus Prime Version	23.1std.1 Build 993 05/14/2024 SC Lite Edition
Flow Elapsed Time		Revision Name	test
Flow OS Summary		Top-level Entity Name	leds
Flow Log		Family	Cyclone 10 LP
Analysis & Synthesis		Device	10CL080YU484C8G
Fitter		Timing Models	Final
Assembler		Total logic elements	72 / 81,264 (< 1 %)
Timing Analyzer		Total registers	29
EDA Netlist Writer		Total pins	7 / 290 (2 %)
Flow Messages		Total virtual pins	0
Flow Suppressed Messages		Total memory bits	0 / 2,810,880 (0 %)
		Embedded Multiplier 9-bit elements	0 / 488 (0 %)
		Total PLLs	0 / 4 (0 %)

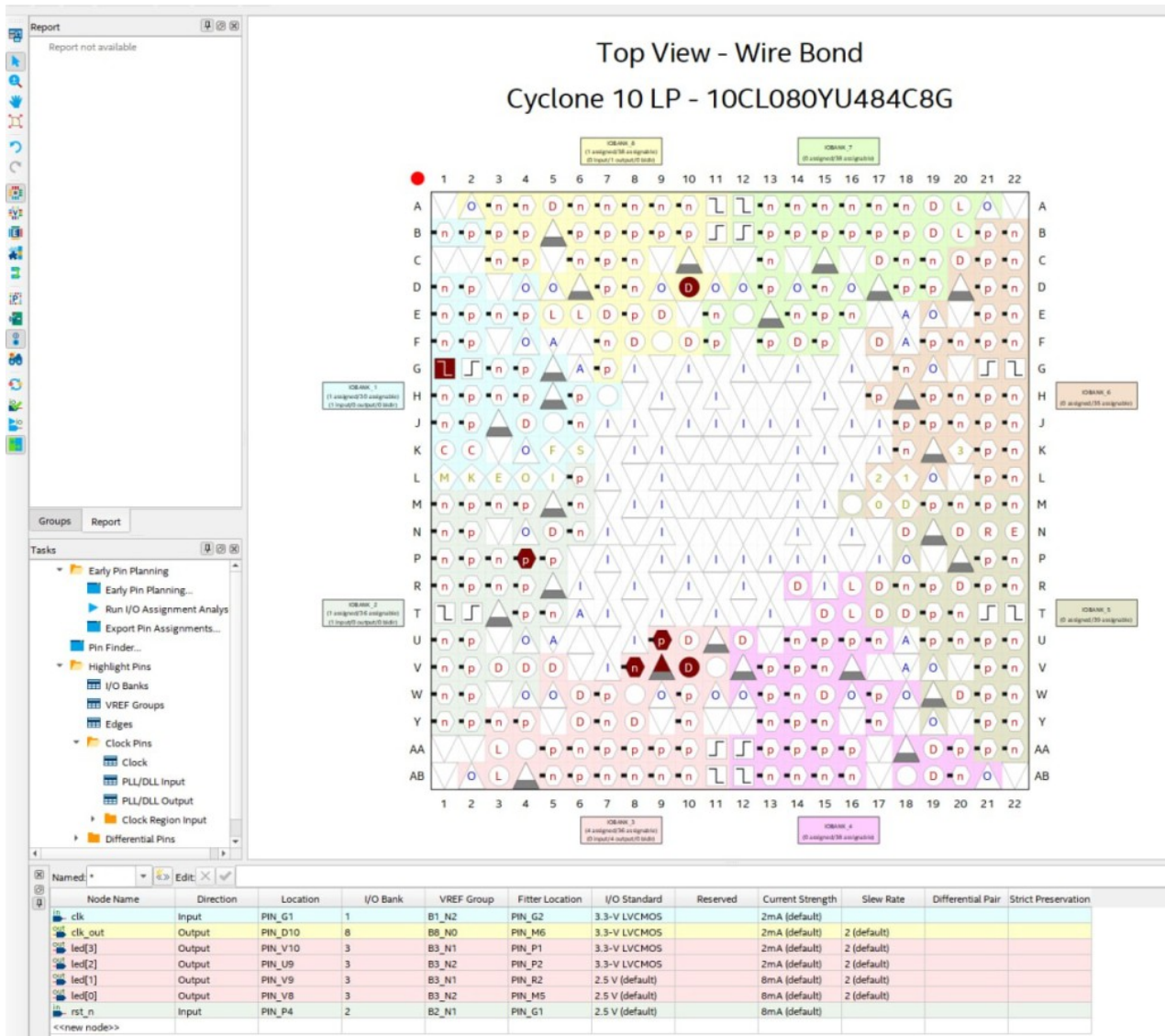
Ahora vamos a asignarle los pines de la FPGA a los puertos. Para ello le damos a *Assignments* y a *Pin Planner*.



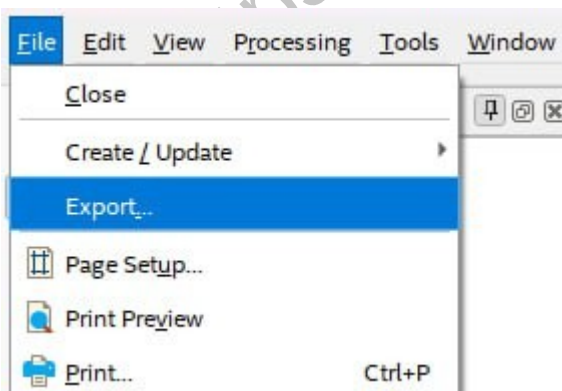
Se nos abre la pestaña para la configuración de los pines.



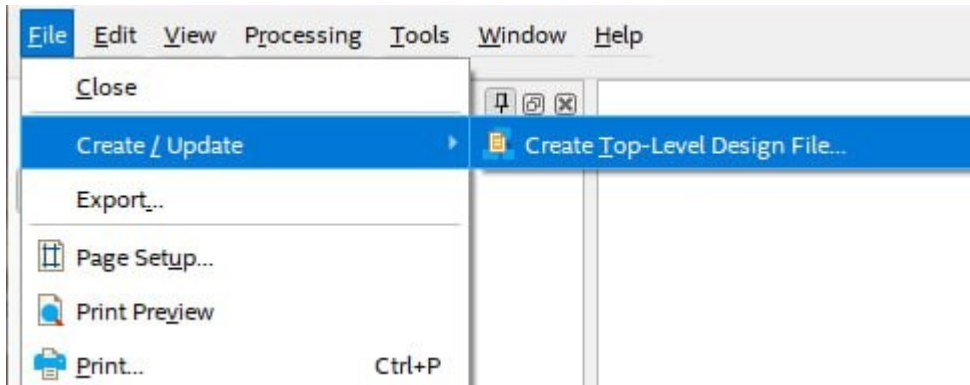
Ahora solo tenemos que asignarle los pines que queremos, para ello se tiene que rellenar la casilla *Location* y después cambiar la tensión en *I/O Standard*. Esta configuración se quedará guardada dentro del proyecto, por lo que no se hace necesario tener un fichero de pines.



NOTA: se pueden exportar los puertos en formato CSV, para ello se le da a la opción de *Export* en *File*



NOTA 2: existe una opción de con estos puertos crear un fichero top para nuestro proyecto. Solo hay que darle a *Create/Update* y después a *Create Top-Level Design File*.



Y esto crearía un nuevo fichero top. ¡Cuidado! porque reescribe el fichero, por lo que puedes borrar todo tu proyecto.

```
library ieee;
use ieee.std_logic_1164.all;
library altera;
use altera.altera_syn_attributes.all;

entity leds is
  port
  (
    -- {ALTERA_IO_BEGIN} DO NOT REMOVE THIS LINE!

    clk : in std_logic;
    clk_out : out std_logic;
    led : out std_logic_vector(3 downto 0);
    rst_n : in std_logic
    -- {ALTERA_IO_END} DO NOT REMOVE THIS LINE!

  );

  -- {ALTERA_ATTRIBUTE_BEGIN} DO NOT REMOVE THIS LINE!
  -- {ALTERA_ATTRIBUTE_END} DO NOT REMOVE THIS LINE!
end leds;

architecture ppl_type of leds is
  -- {ALTERA_COMPONENTS_BEGIN} DO NOT REMOVE THIS LINE!
  -- {ALTERA_COMPONENTS_END} DO NOT REMOVE THIS LINE!
begin
  -- {ALTERA_INSTANTIATION_BEGIN} DO NOT REMOVE THIS LINE!
  -- {ALTERA_INSTANTIATION_END} DO NOT REMOVE THIS LINE!

end;
```

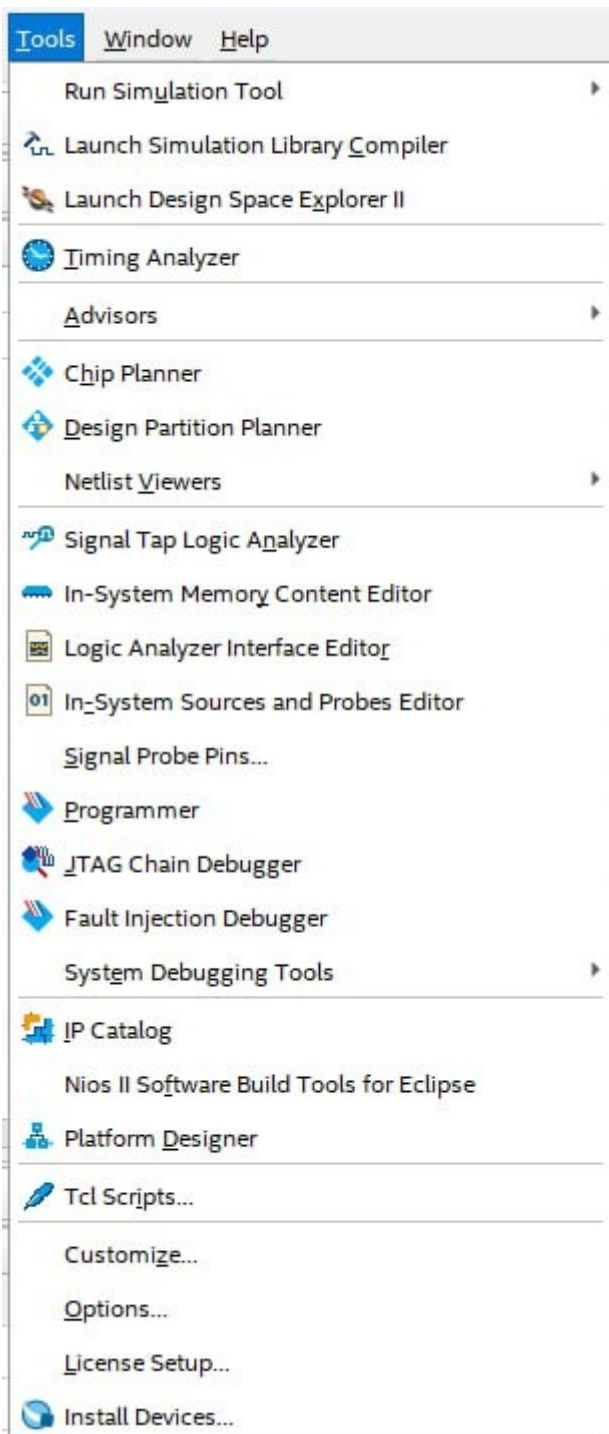
Cuando ya tengamos los pines asignados, lo siguiente es volver a generar la Implementación, que se hace como antes.

	Task	Time
?	▶ Compile Design	
✓	▶ ▶ Analysis & Synthesis	00:00:10
?	▶ ▶ Fitter (Place & Route)	00:00:10
?	▶ ▶ Assembler (Generate programming files)	00:00:03
?	▶ ▶ Timing Analysis	00:00:02
?	▶ ▶ EDA Netlist Writer	00:00:01
	■ Edit Settings	
	🔧 Program Device (Open Programmer)	

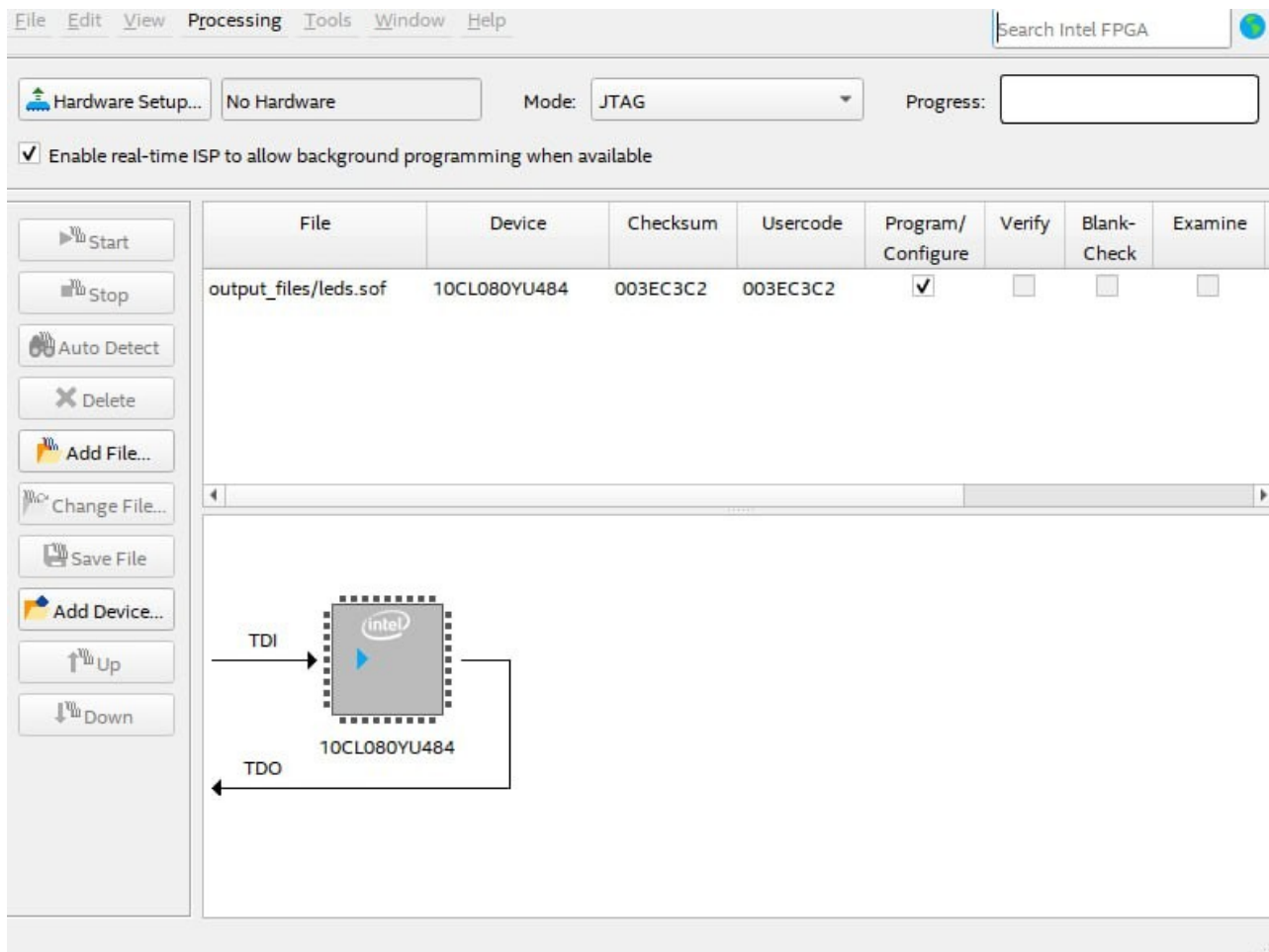
Una vez terminada esta implementación ya se puede pasar a programar la FPGA.

	Task	Time
✓	▶ Compile Design	00:00:26
✓	▶ ▶ Analysis & Synthesis	00:00:10
✓	▶ ▶ Fitter (Place & Route)	00:00:09
✓	▶ ▶ Assembler (Generate programming files)	00:00:03
✓	▶ ▶ Timing Analysis	00:00:03
✓	▶ ▶ EDA Netlist Writer	00:00:01
	■ Edit Settings	
	🔧 Program Device (Open Programmer)	

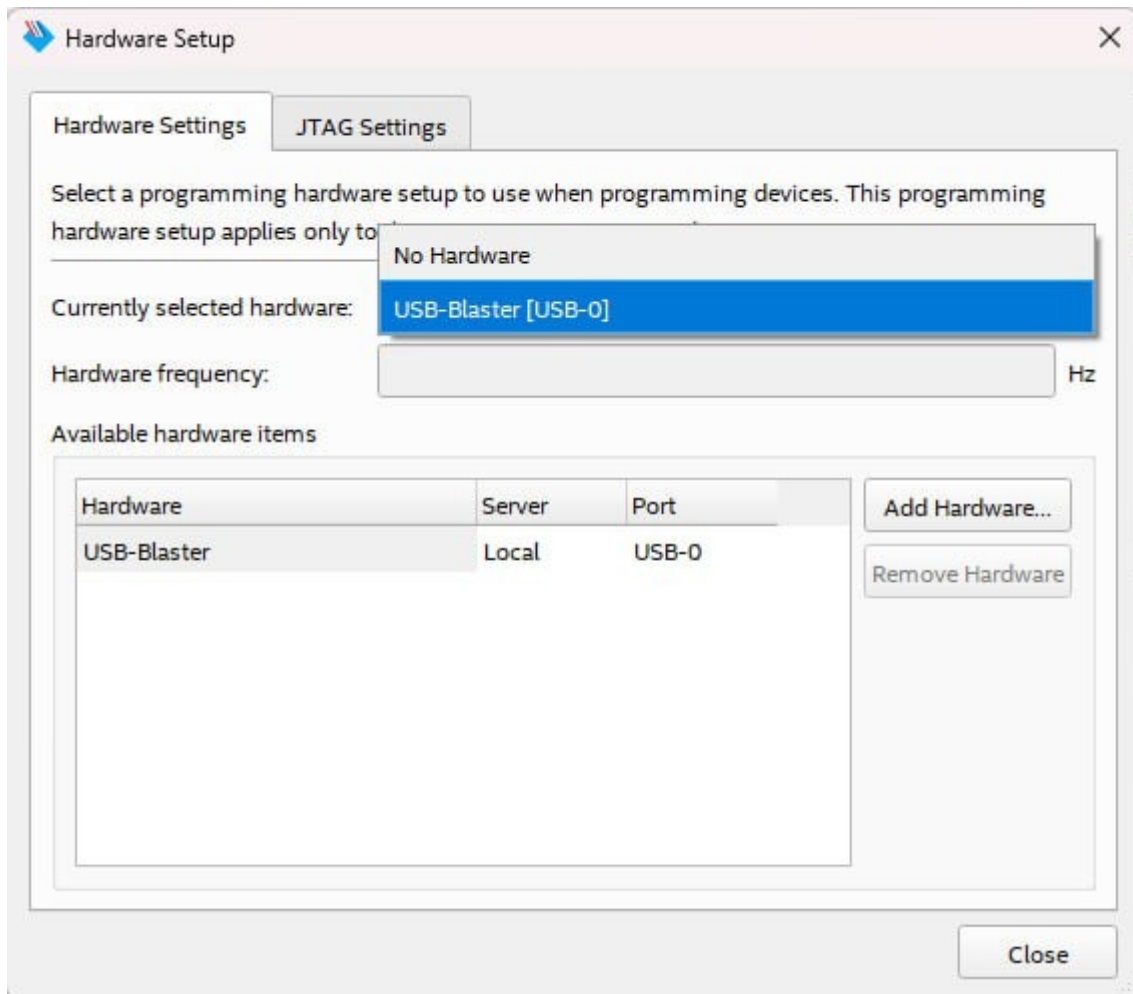
Ahora nos vamos a *Tools* y abrimos la herramienta *Programmer*.



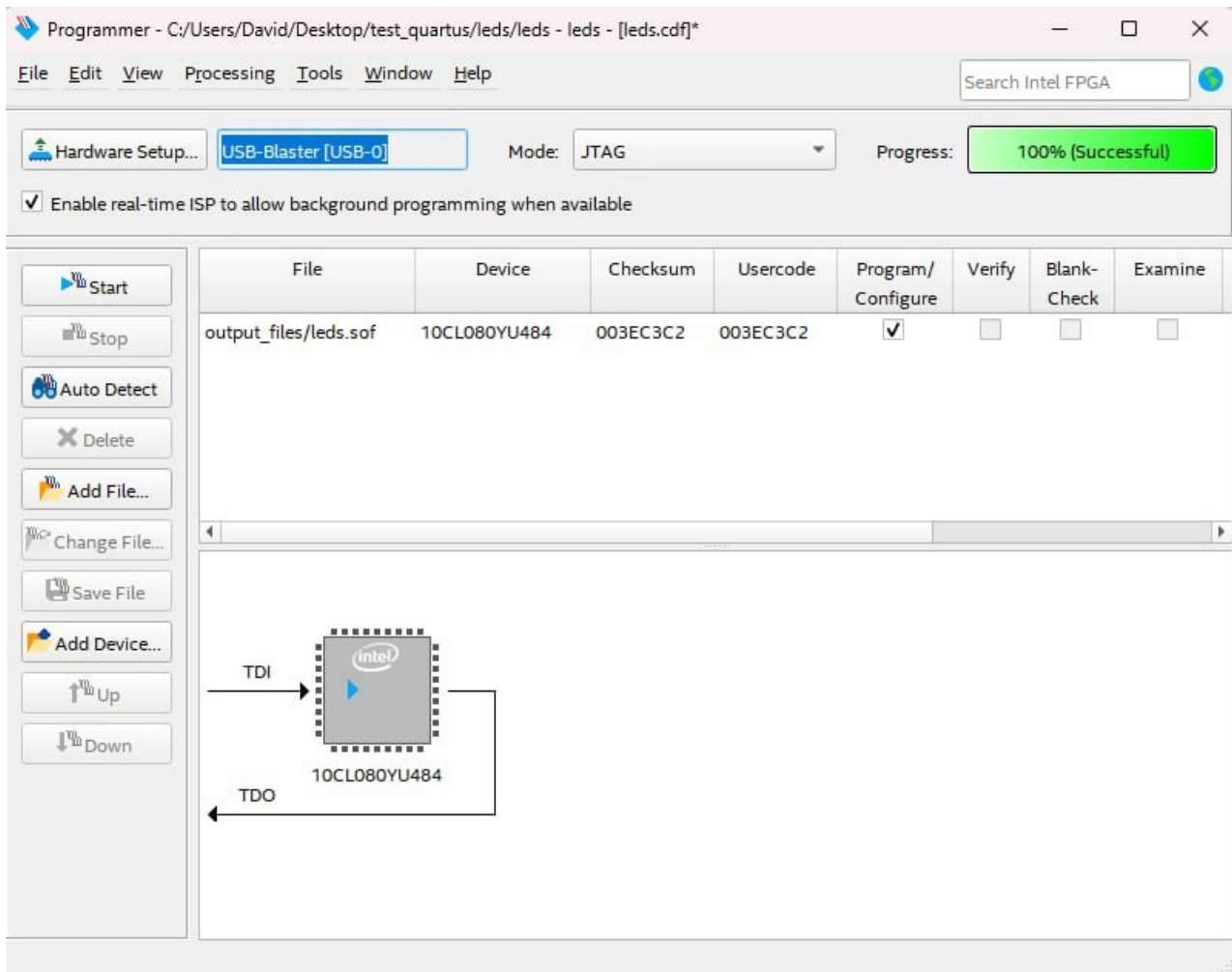
Ahora se nos abre el programa que graba la FPGA, y al estar en el mismo proyecto incluye automáticamente el bitstream. *El bitstream es un fichero de tipo SOF.*



Lo primero que hay que hacer es configurar el JTAG que se va a utilizar.



Después lo único que hay que hacer una vez tenemos configurado el JTAG es darle al botón de *Start*, esto inicia automáticamente la grabación de la FPGA.



Si está todo correcto saldrá un mensaje de *Successful* y la FPGA estará programada.