

**`DISEÑO DE SISTEMAS DIGITALES AVANCE #2**

**IMPLEMENTACIÓN DE UN SISTEMA EMBEBIDO BASADO EN FPGA UTILIZANDO UNA TARJETA SD Y UNA CÁMARA PARA ALMACENAR VIDEO DE FORMA CONTINUA.**

**CAPÍTULO DEL CURSO:** PROYECTO DE LA MATERIA DISEÑO DE SISTEMAS DIGITALES

**TEMA DE LA ACTIVIDAD:** UTILIZACION DE LA CÁMARA D5M CON EL FIN DE PODER ALMACENAR UNA SERIE DE IMAGENES EN LA MEMORIA SDRAM ASI COMO LA LECTURA DE ESTAS IMÁGENES A PARTIR UNA TARJETA SD CARD

# OBJETIVOS DE APRENDIZAJE:

* Elaborar bloques de procesamiento para la imagen que facilitara el control de la cámara D5M con la tarjeta DE10-Standard.
* Desarrollar los bloques en VHDL que son parte de un sintetizador standard.
* Enlazar los bloques de manera funcional, en el interior de la FPGA.
* Desarrollar una interfaz de control analógico y salida por monitor HDMI/VGA.
* Implementar la elaboración de un código en lenguaje C++ que facilite la escritura de frames por segundo que son registrados a través de la cámara en la memoria SDRAM, empleando periféricos propios proporcionados por la tarjeta.
* Diseñar un código que permita mostrar en la pantalla VGA la lectura de una secuencia de imágenes guardadas en la memoria SDCARD.

**DURACIÓN:** 60 minutos

# MATERIALES Y HERRAMIENTAS:

* Software Quartus 17.0 versión Estándar.
* Software Eclipse Kepler.
* Tarjeta de desarrollo DE10-Standard.
* Cámara D5M
* Tarjeta SD de 2 GB.

# INTRODUCCIÓN

Hoy en día, el empleo de las cámaras digitales es necesario para los sistemas

dedicados a la vigilancia de automóviles, dado que esto nos facilita

visualizar, así como también registrar en todo momento lo que sucede

a la vez que se encuentre manejando, o no; asimismo, la probabilidad de

conseguir un extracto del video que fue grabado de un suceso dado, por ejemplo:

un accidente entre dos carros, choques, etc.

Debido a esto es fundamental la implementación de un sistema embebido que contenga bloques que faciliten la construcción del hardware, así como del software del sistema, con la posibilidad de hacer uso de la cámara, registrar los frames por segundo y almacenarlos dentro de la memoria SDCARD; para después tener la probabilidad de permitir el acceso a esa información que se encuentra dentro de la memoria.

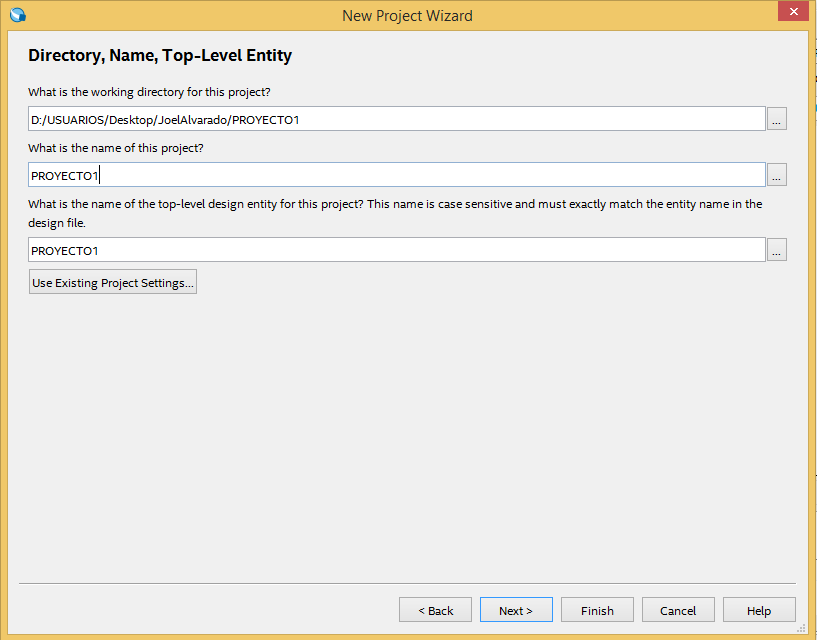
**DESCRIPCIÓN DE LA PRÁCTICA:** Se añadirá el sistema QSYS llamado Sistema.qsys y el bloque vhdl llamado Proyecto1 al proyecto que fue creado para después proceder a compilar el código que facilite la escritura así como la lectura de la SDRAM que se encontrara guardada una sucesión de imágenes captadas a partir la cámara D5M; y al último almacenar esas imágenes dentro de la SDCARD.

# PROCEDIMIENTO:

1. Proceda a crear una carpeta en el escritorio de windows de la PC con el nombre PROYECTO1(escriba sin espacios). Luego copie y pegue todos los archivos que se encuentran en la carpeta compartida, en la carpeta que fue creada en el escritorio de windows de la PC.
2. Ejecute el programa Quartus Prime 17.0 Standard Edition que se encuentra en el escritorio representado por el ícono .

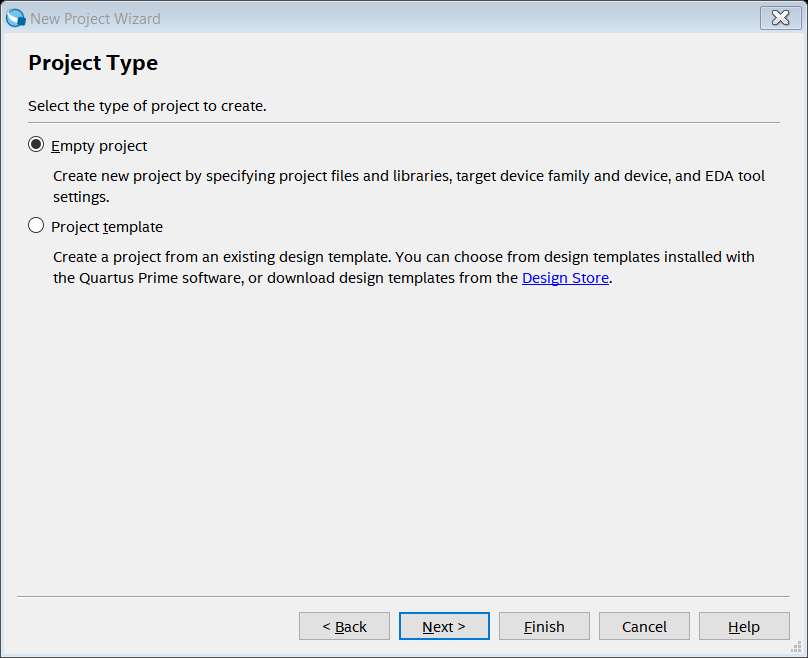
**Figura 1. Creación de un nuevo proyecto.**

1. Proceda a crear un nuevo Proyecto seleccionando la opción **File→New Project Wizard** como se observa en la Figura 1. Aparecerá la ventana de Introducción, luego haga clic en **Next**.
2. En la primera línea escoge la ruta donde va a crear su proyecto, en este caso, la ruta será la carpeta PROYECTO1 que fue creada en el procedimiento 1 y se encuentra en el escritorio de windows de la PC. En la siguiente línea escoja como nombre del proyecto PROYECTO1, tal como se visualiza en la Figura 2. Luego haga clic en Next.



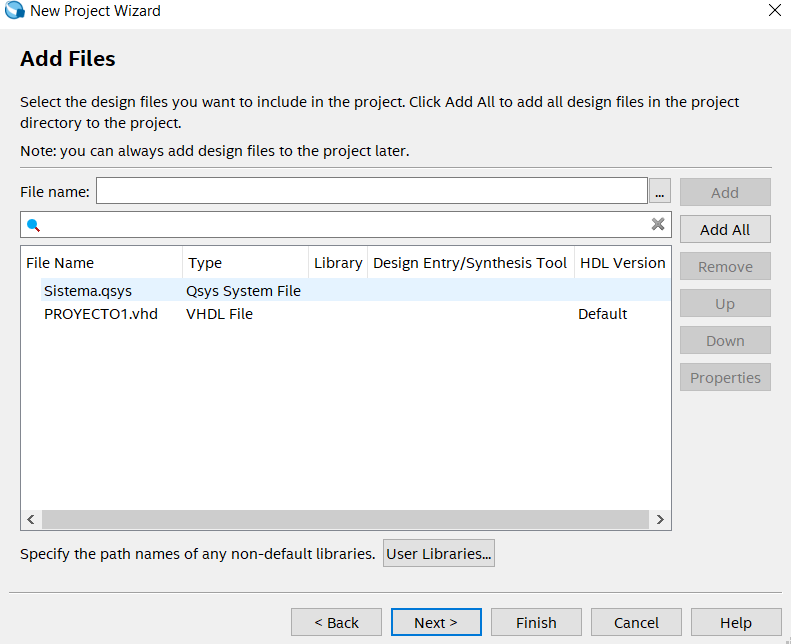
**Figura 2. Nombre del proyecto.**

1. Se mostrará la ventana para seleccionar el tipo de proyecto. Se escogerá la opción de Proyecto vacío (**Empty project**), tal como se visualiza en la Figura 3. Luego haga clic en **Next**.



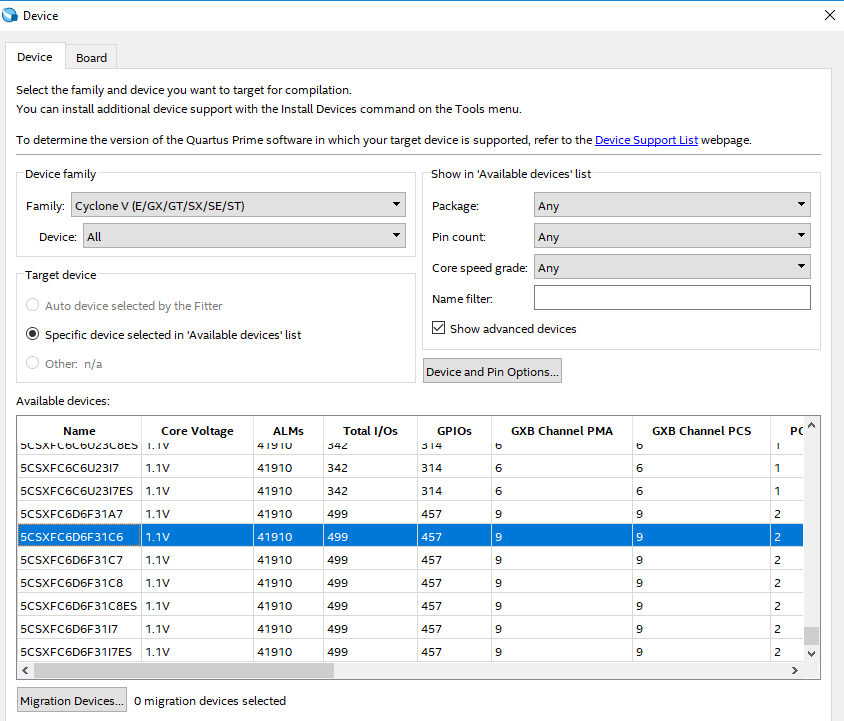
**Figura 3. Tipo del proyecto**

1. Se mostrará la ventana para adhesión de archivos. En esa ventana seleccione la opción **Add All**, para que se agreguen todos los archivos que fueron copiados a la carpeta del proyecto y que se realizó en el procedimiento 1. En la Figura 4 se observan los archivos añadidos. Por último, haga clic en **Next**.



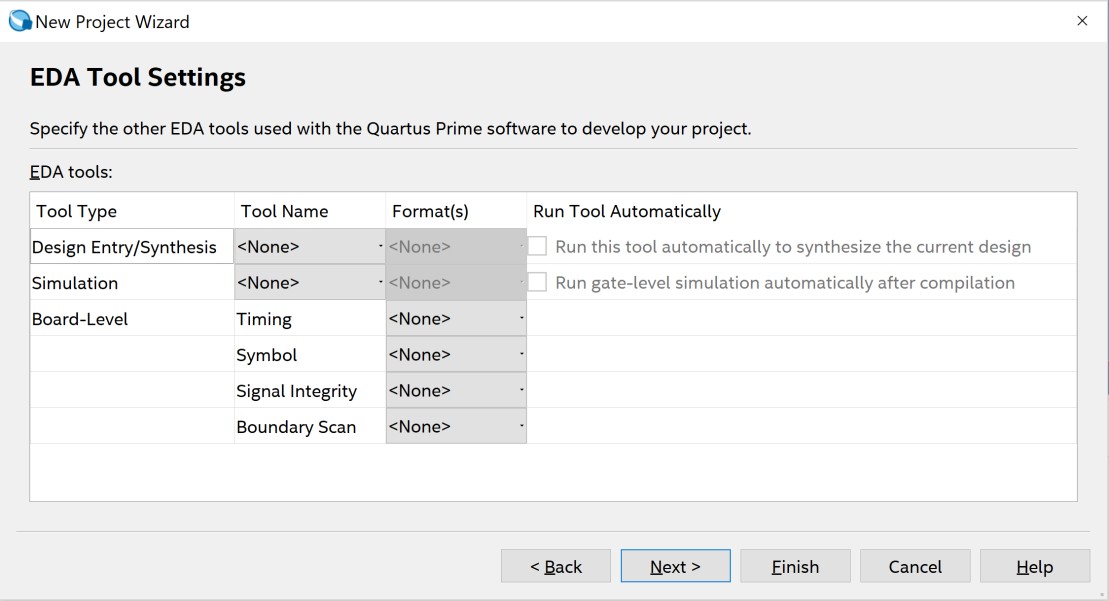
**Figura 4. Adhesión de archivos**

1. En la siguiente ventana aparecerán todas las familias de chips FPGA. Proceda a escoger la familia **Cyclone V (E/GX/GT/SX/SE/ST)** y seleccione el nombre del chip FPGA SoC **5CSXFC6D6F31C6**, tal como se muestra en la Figura 5. Luego haga clic en **Next**.

i

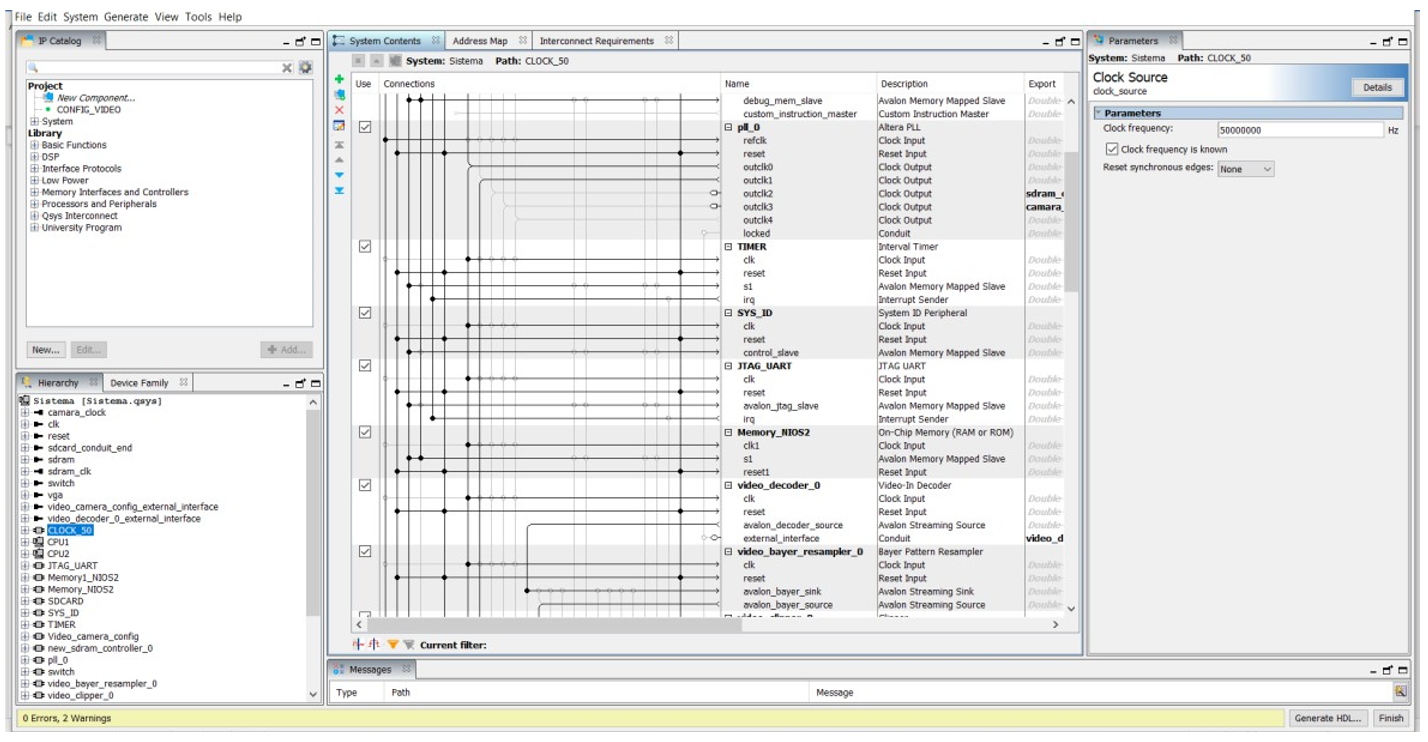
**Figura 5. Familia y modelo del chip FPGA SoC**

1. Debido a que no se realizarán simulaciones en la presente actividad, no se seleccionará alguna herramienta de simulación como se muestra en la Figura 6. Haga clic en **Next**.



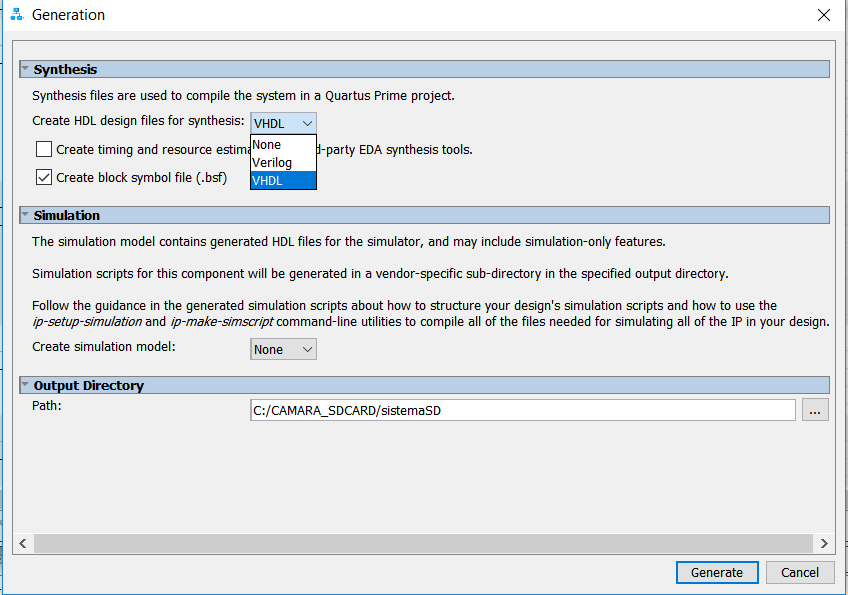
**Figura 6. Herramientas para simulación**

1. Al finalizar la creación del proyecto, aparecerá la ventana **Summary** para resumir todas las especificaciones que tendrá nuestro proyecto. Para finalizar haga clic en **Finish**.
2. Haga doble clic al archivo **sistemaSD.qsys** dentro del programa **Quartus Prime**. Dentro de la ventana **Qsys**, haga clic en **Generate HDL** que se encuentra en la parte inferior derecha, tal como se muestra en la Figura nuermo 7.



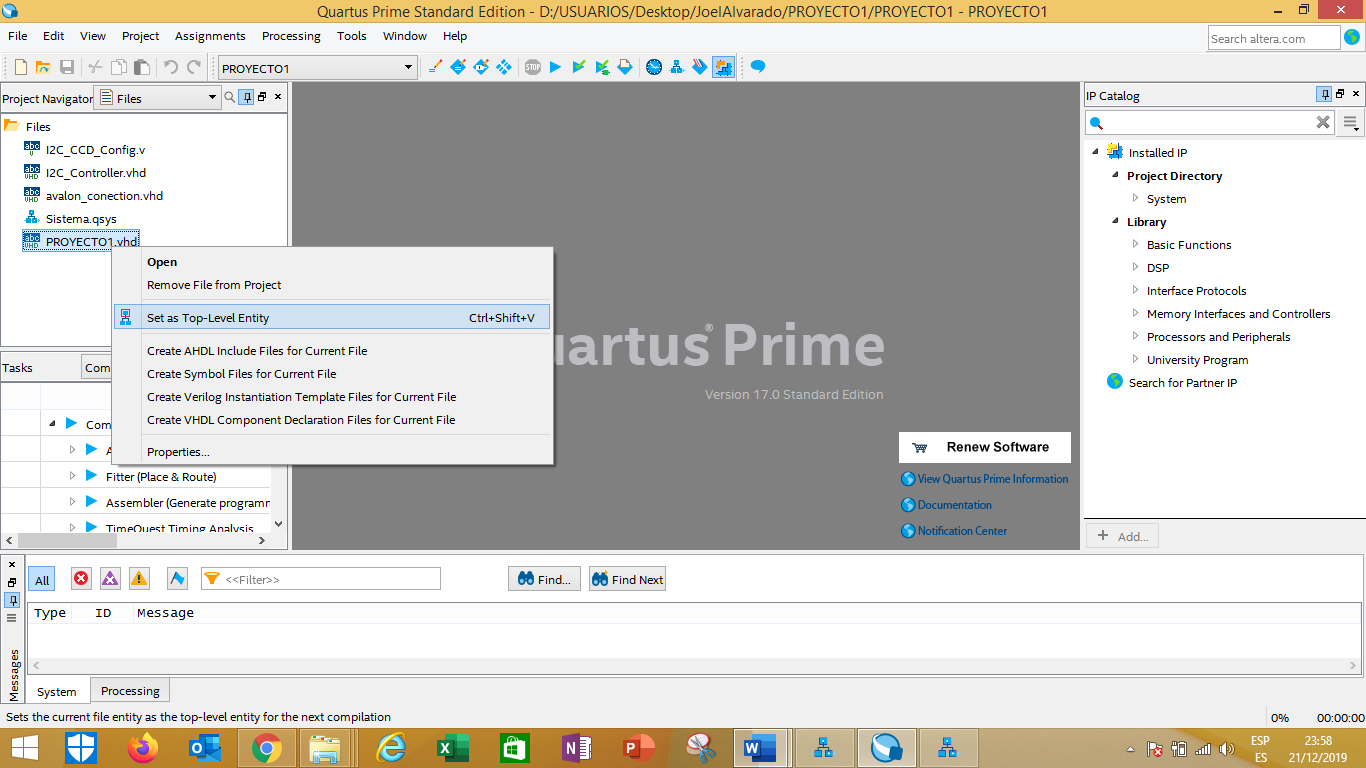
**Figura 7. Ventana Qsys**

1. En la ventana **Generation**, seleccione la creación de archivos VHDL en la sección **Synthesis**, tal como se muestra en la Figura 8. Por último, haga clic en **Generate**, y cuando finalice haga clic en **Finish**.

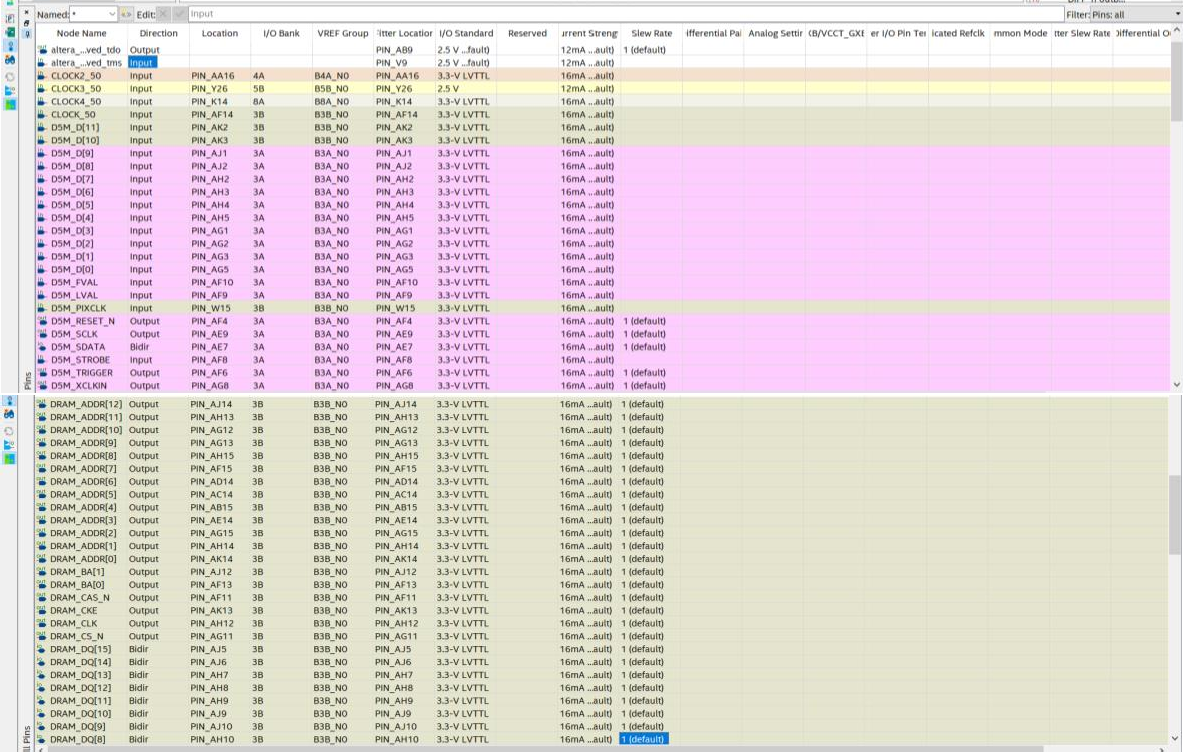


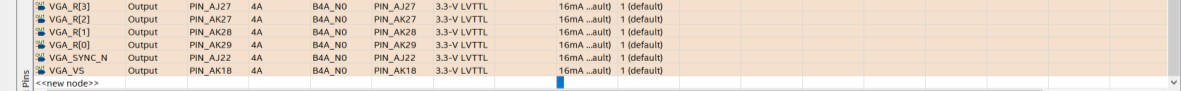
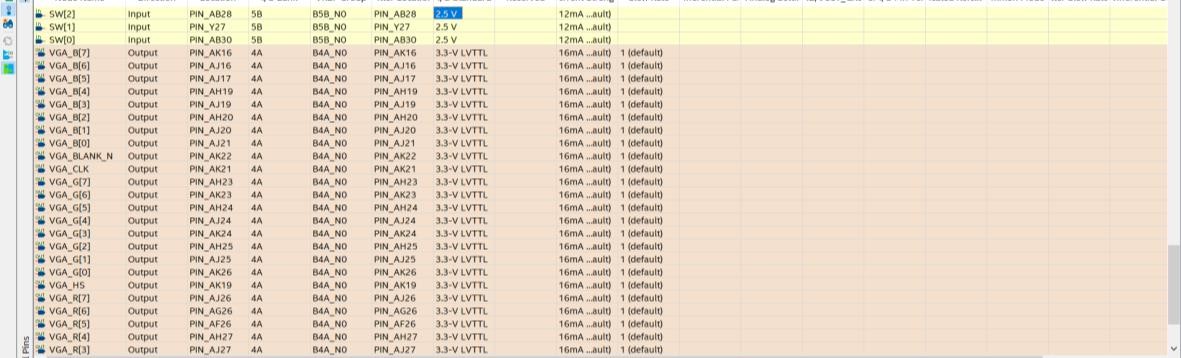
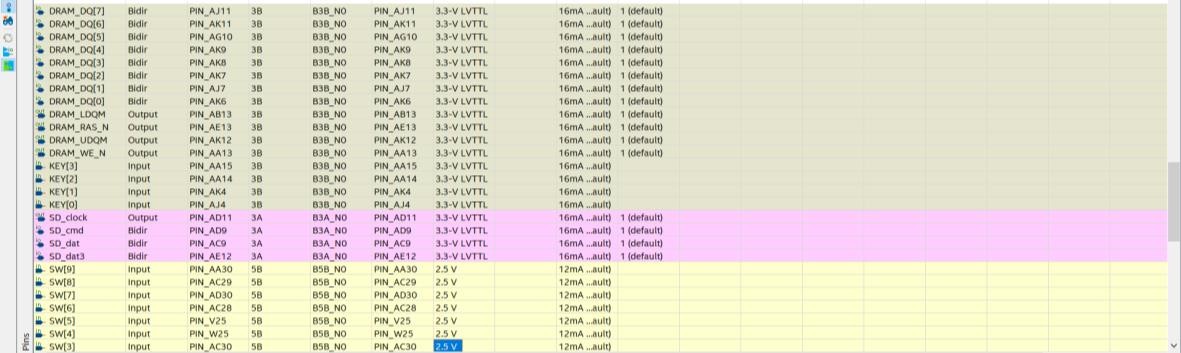
**Figura 8. Ventana Generation**

1. Proceda a compilar el archivo **PROYECTO1.vhd** estableciéndolo previamente como más alta jerarquía (dando clic derecho en el mismo y seleccionando la opción **Set as Top-Level Entity** Luego realice la compilación **Start Analysis & Synthesis** haciendo clic en el ícono  como se muestra en la Figura 9.



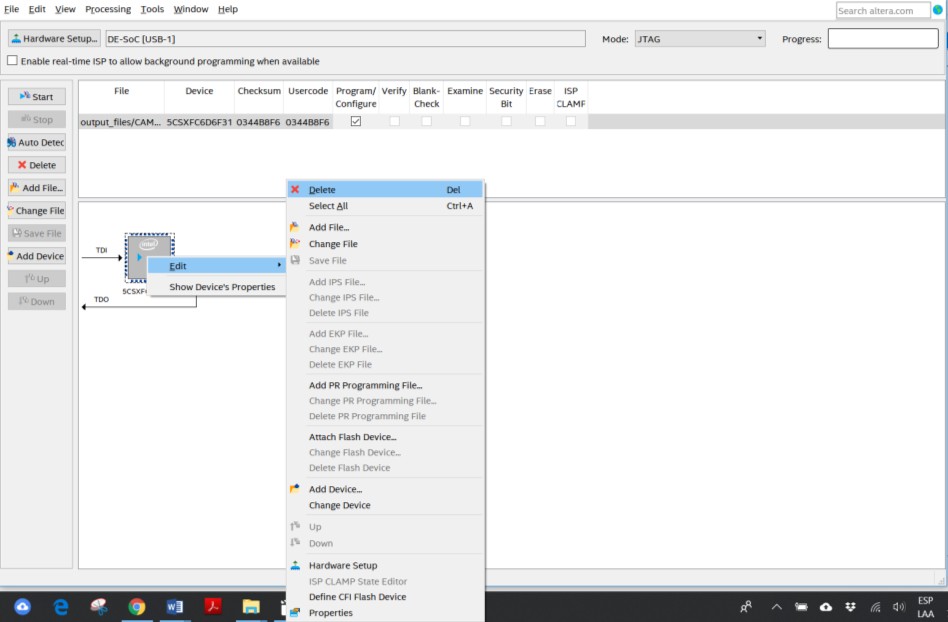
**Figura 9. Compilación del proyecto**

1. Diríjase al menú **Assignments** de la ventana Quartus y seleccione **Pin Planner**. Asigne los pines de las señales de entrada y salida en la columna **Location**, tal como se muestra en la Figura 10.



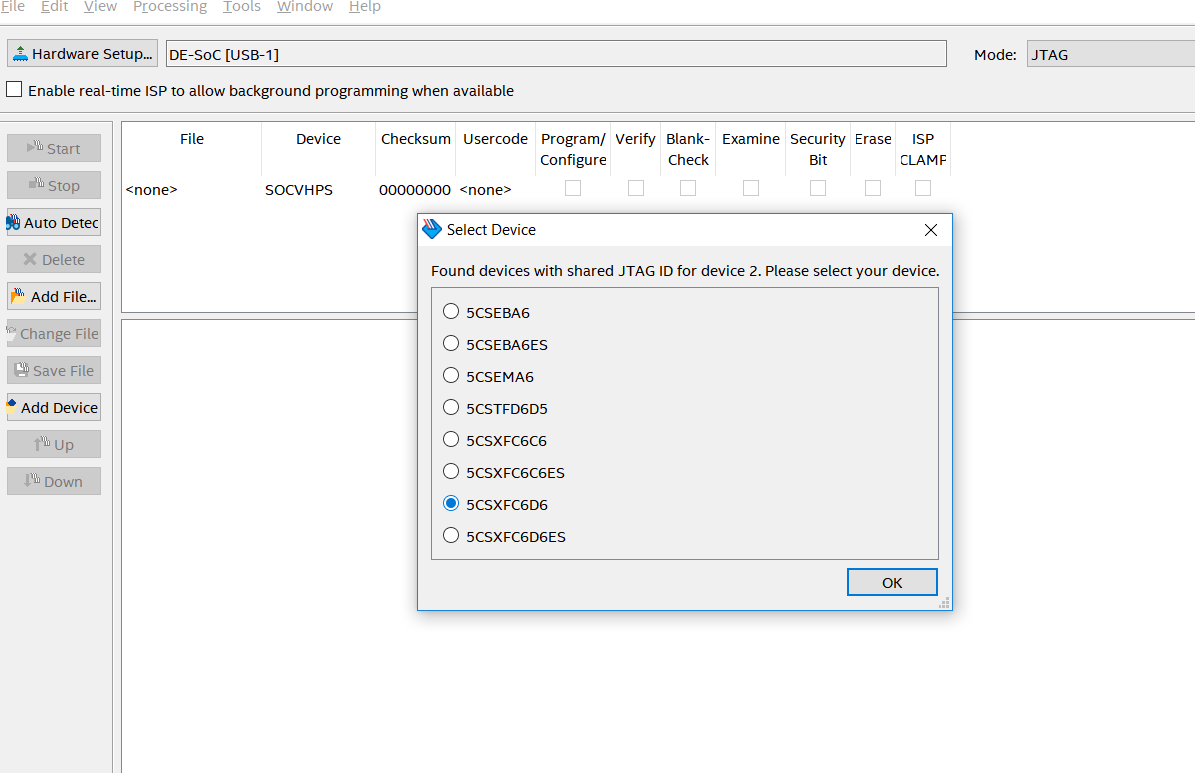
**Figura 14. Asignación de Pines**

1. Una vez que terminó de asignar los pines, proceda a cerrar la ventana **Pin Planner**, y luego compile nuevamente el archivo **PROYECTO1.vhd,** pero en esta ocasión seleccione el ícono **Start Compilation .**
2. Proceda a conectar la tarjeta de desarrollo **DE10-Standard** tanto a la PC (cable **USB-Blaster**) como a la fuente de alimentación.
3. Una vez que se haya completado la compilación, diríjase al ícono  que se encuentra en la barra de tareas del programa **Quartus Prime**, y servirá para programar la tarjeta de desarrollo **DE10-Standard**.
4. En la ventana Programmer se visualizará un archivo, haga clic derecho en el mismo y elimínelo escogiendo la opción **Delete**, tal como se visualiza en la Figura 11.



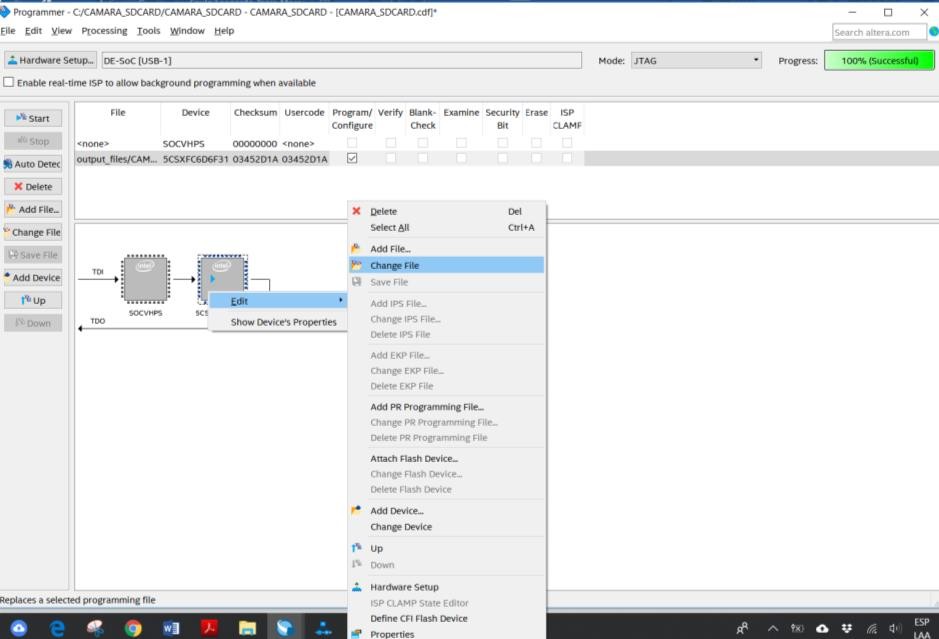
**Figura 11. Ventana Programmer**

1. Una vez eliminado el archivo y que no se encuentre ningún otro archivo en la ventana Programmer, proceda a escoger la opción **Auto Detect** representada por la siguiente botonera . Seleccione la opción **5CSXFC6D6** tal como se muestra en la Figura 12 Haga clic en OK.

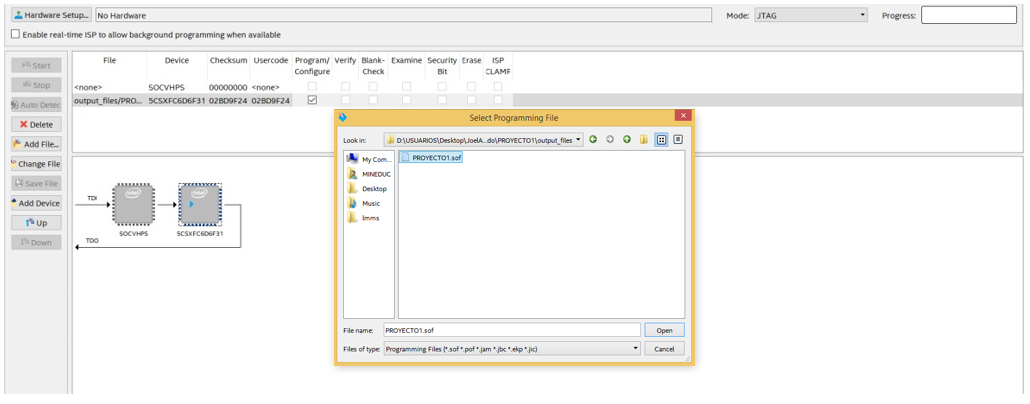


**Figura 12. Ventana Auto Detect**

1. Aparecerán dos archivos: **SOCVHPS** y **5CSXFC6D6**. Haga clic derecho en el segundo y escoja la opción Change File tal como se muestra en la Figura 13, y reemplácelo por el archivo **PROYECTO1.sof** que se encuentra en la carpeta **Output\_Files** como se muestra en la Figura 14.

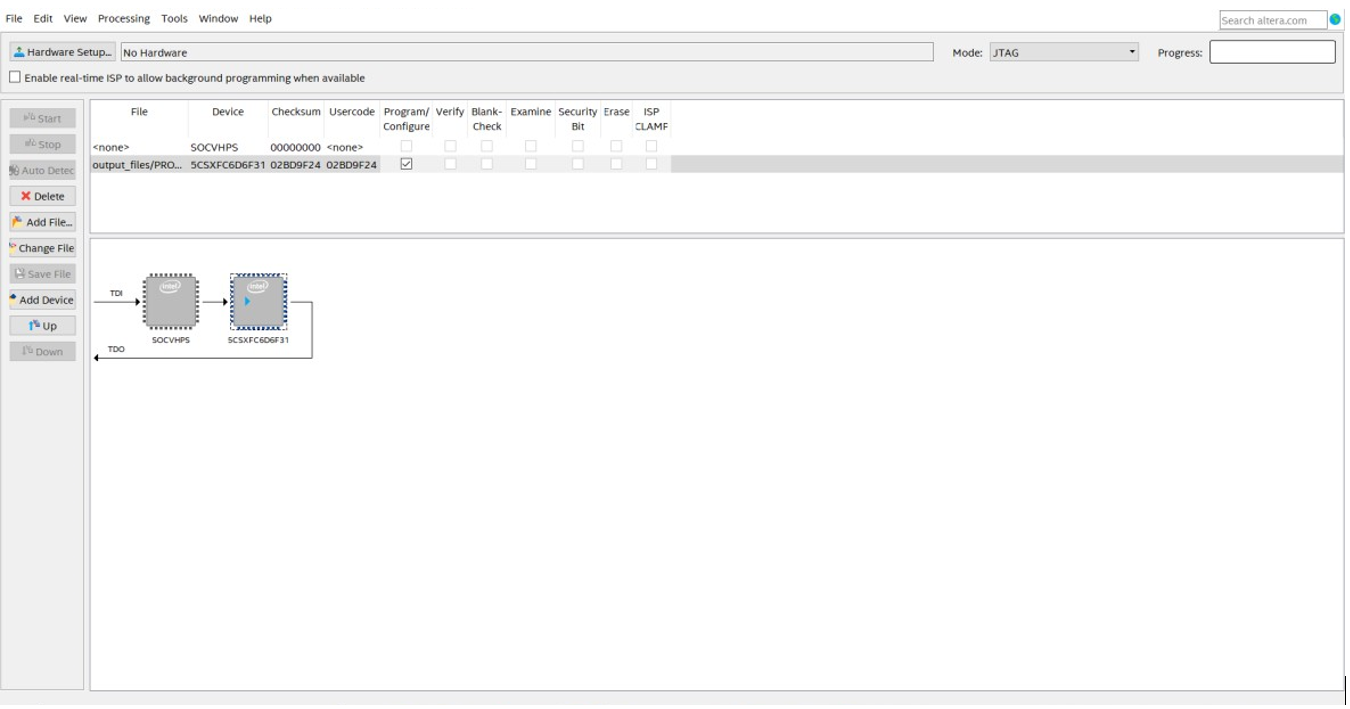


**Figura 13. Cambio de archivo en la ventana Programmer**



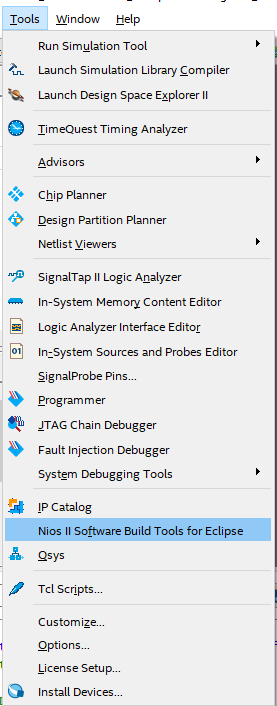
**Figura 14. Buscando el archivo .sof**

1. Haga clic en la casilla correspondiente a la columna **Program/Configure** del archivo **PROYECTO1.sof**, y luego seleccione **Start** para que se programe el chip FPGA. Para que la programación sea exitosa se debe de alcanzar el 100%, tal cual se muestra en la Figura 15.



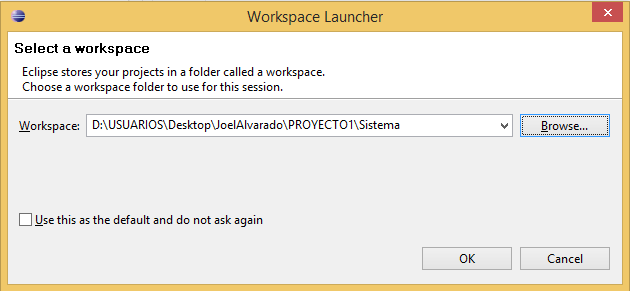
**Figura 15. Programación exitosa del chip FPGA**

1. En este momento se acaba de construir físicamente la computadora embebida con el microprocesador Nios II en el chip FPGA.
2. Lo siguiente que se realizará será la programación del microprocesador **Nios II** utilizando lenguaje C++. En el menú **Tools**, seleccione la opción **Nios II Software Build Tools for Eclipse**, como se muestra en la Figura 16.



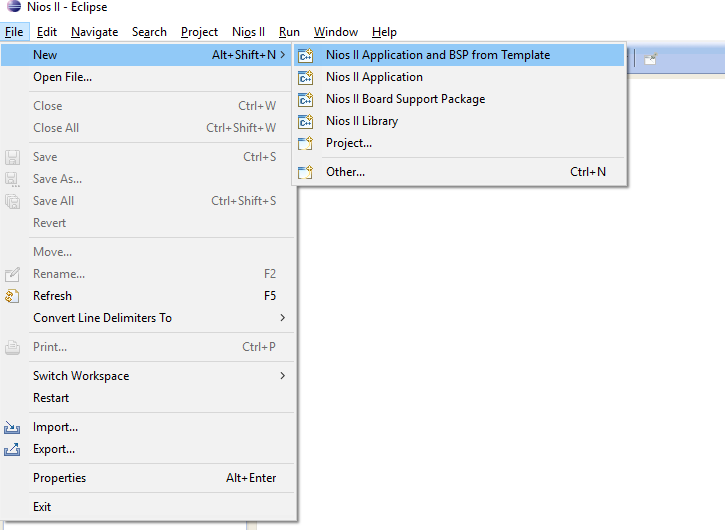
**Figura 16. Selección de la herramienta Eclipse**

1. Lo primero que pedirá Eclipse será ubicar en donde va a estar alojado el proyecto que vayamos a crear, seleccione la carpeta **Sistema** que se encuentra dentro de la carpeta del proyecto, como se encuentra en la Figura 17. Haga clic en OK.



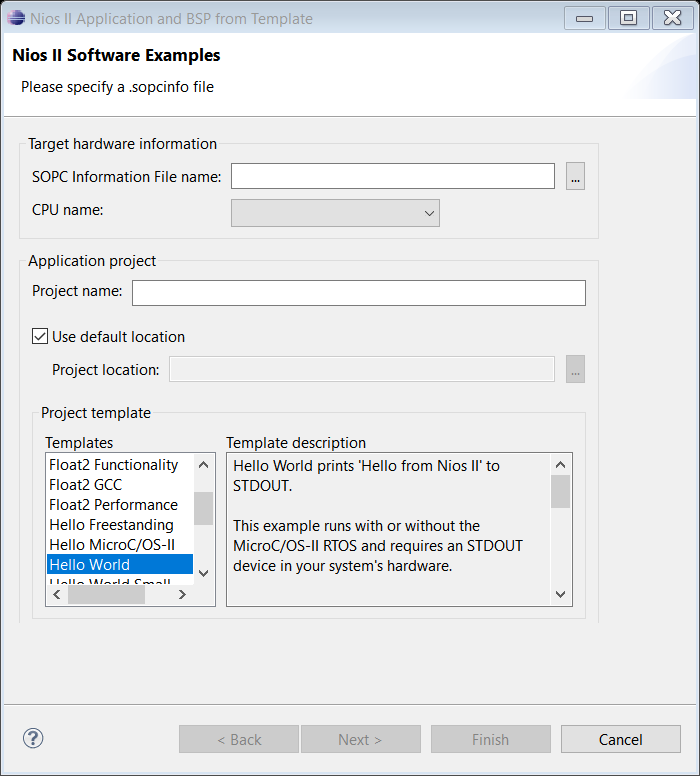
**Figura 17. Ubicación del proyecto en Eclipse**

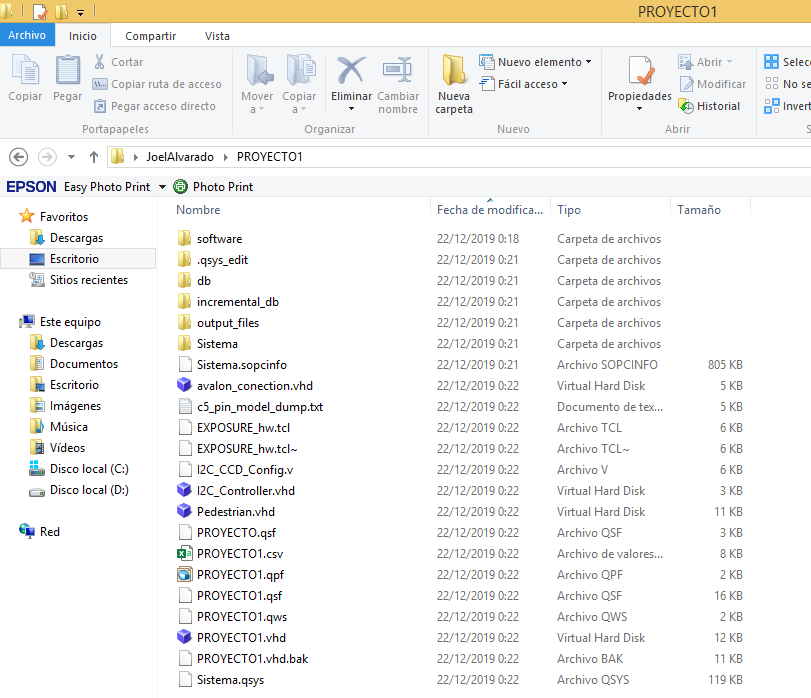
1. Se abrirá el entorno de programación de Eclipse. Se procederá a crear un nuevo proyecto, para ello seleccione **File→New→Nios II Application and BSP from Template** como se visualiza en la Figura 18.



**Figura 18. Selección de una nueva Aplicación para Nios II**

1. En la opción **SOPC Information File name**, proceda a buscar el archivo **Sistema.sopcinfo** que tendrá toda la información de la computadora embebida, como se visualiza en la Figura 19.





**Figura 19. Ubicación del archivo que contiene la computadora embebida**

1. En la opción **CPU name** se seleccionará **nios2\_gen2\_0.**
2. En **Project name** se elegirá el nombre del proyecto, utilice el nombre

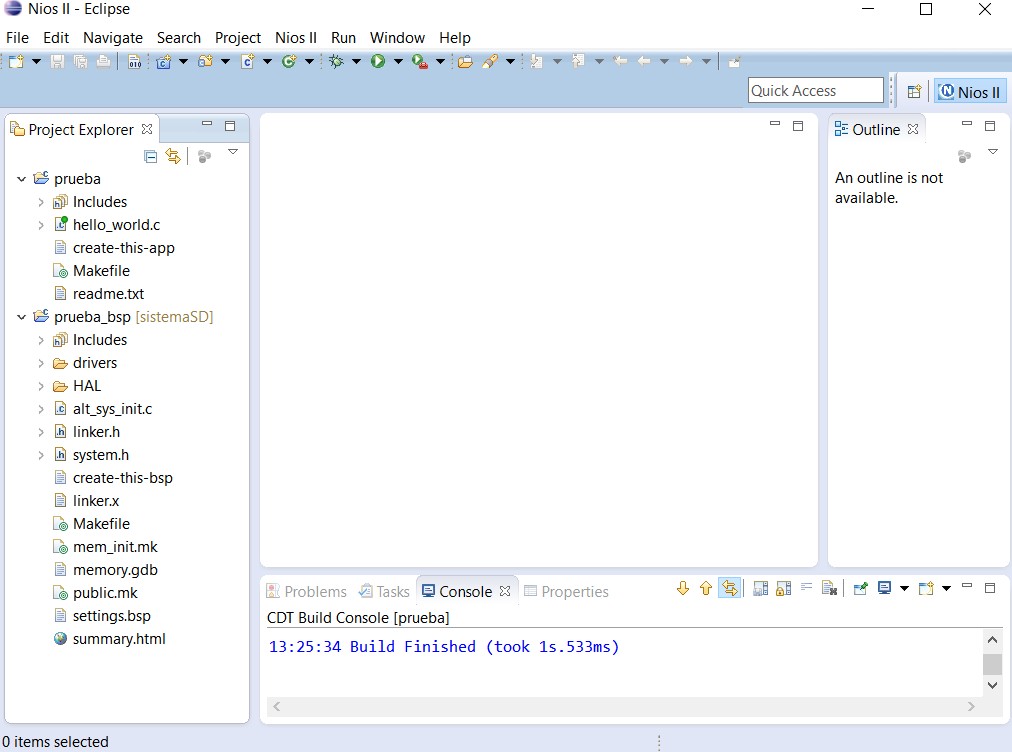
**prueba** para efectos de la práctica.

1. En **Project Template**, seleccione la opción **Hello World** para crear nuestro proyecto vacío. Haga clic en **Finish**. Todo lo anteriormente dicho se lo observará en la Figura 20.



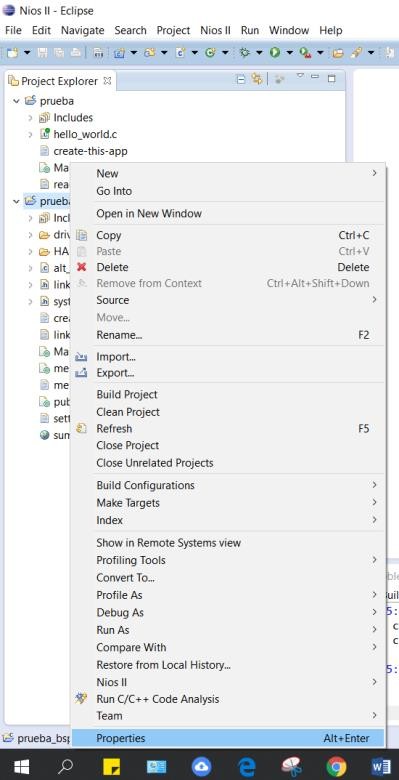
**Figura 20. Ubicación del archivo que contiene la computadora embebida**

1. En la subventana de Project Explorer podrán visualizar dos carpetas: **prueba** y **prueba\_bsp**. En la primera se podrán crear todos los archivos de programación en lenguaje C/C++ para nuestro microprocesador, y la segunda carpeta contiene todos los componentes del hardware, como se visualiza en la Figura 21.



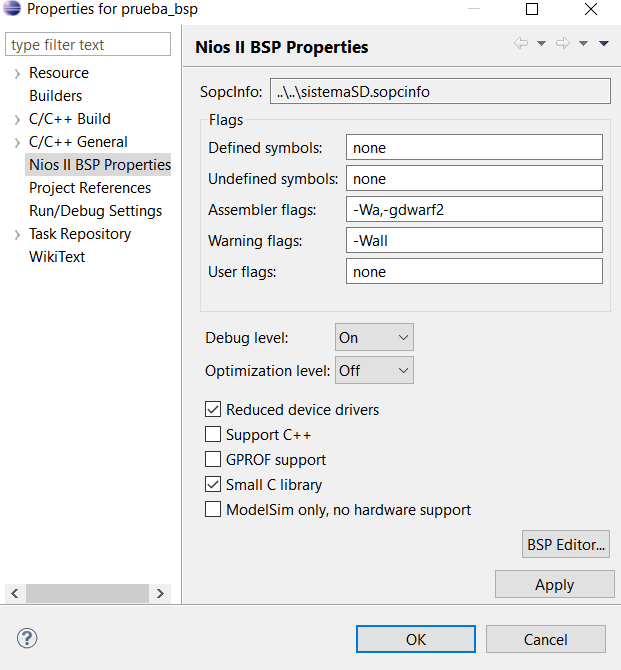
**Figura 21. Ubicación del archivo que contiene la computadora embebida**

1. Copie la programación del archivo **PROYECTO1.txt** que se encuentra en la carpeta compartida, en el **hello\_world.c** de Eclipse (sustituya a lo que esté ahí). Parte de la programación fue obtenida de la siguiente referencia [1].
2. Proceda a hacer clic derecho a la carpeta **prueba\_bsp[sistemaSD]** y seleccione la opción **Properties**, tal como se muestra en la Figura 22.



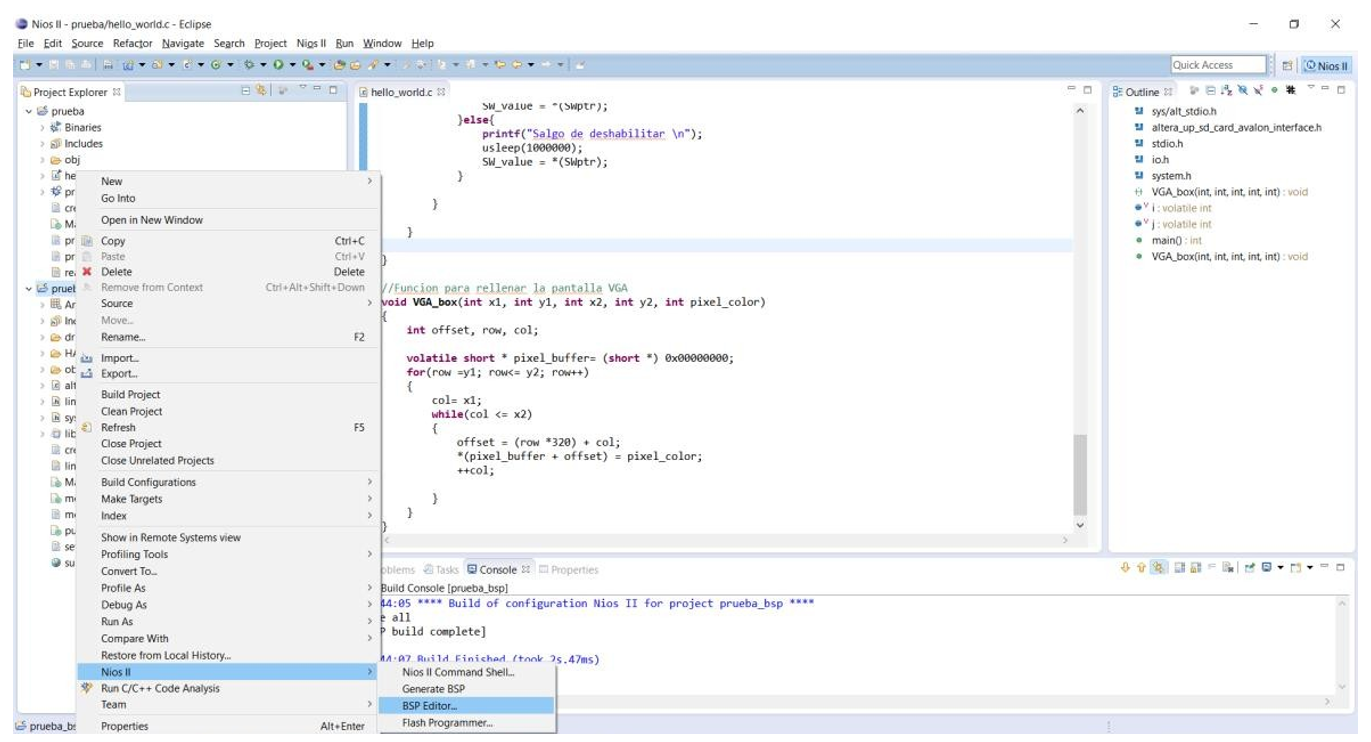
**Figura 22. Propiedades del archivo prueba\_bsp**

1. En la ventana de propiedades que se visualizará, diríjase a la categoría **Nios II BSP Properties**, deshabilite la casilla **Support C++** y habilite las casillas **Reduced device drivers** y **Small C library**, tal como se muestra en la Figura 23. Luego haga clic en **OK**.



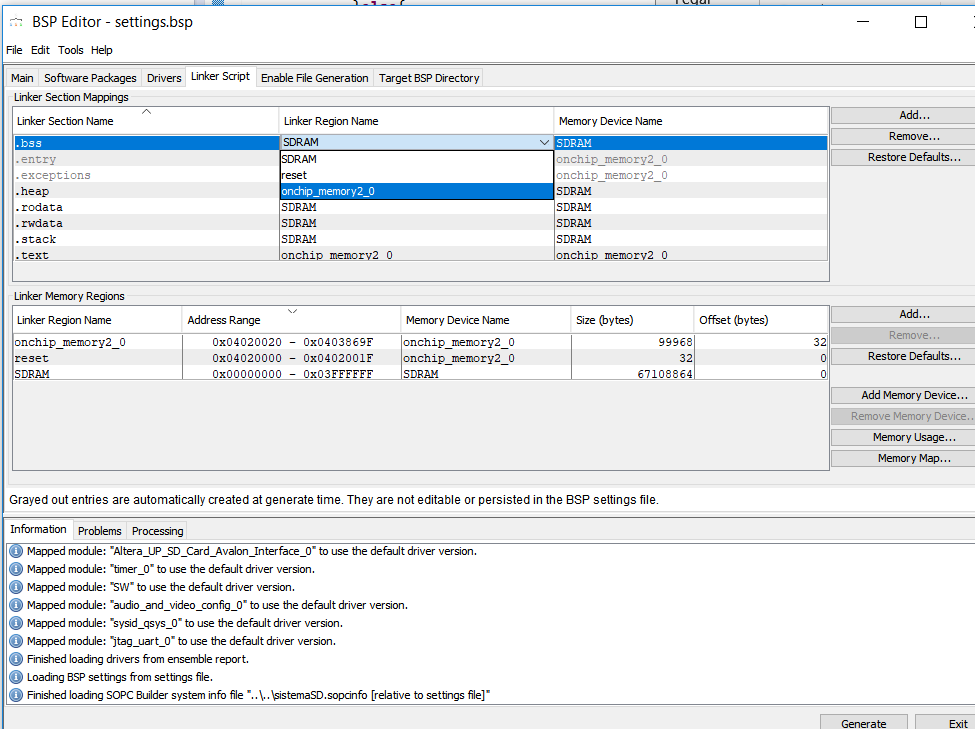
**Figura 23. Selección para librerías reducidas.**

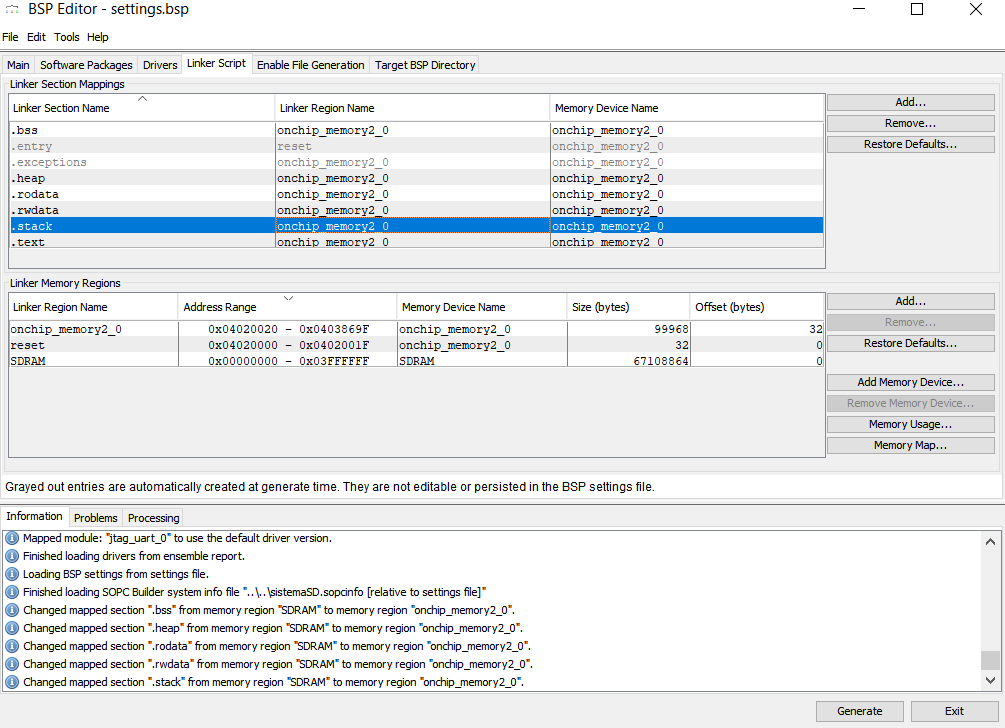
1. Procedemos a editar el BSP del archivo prueba\_bsp. Hacemos clic derecho en prueba\_bsp y seleccionamos **NiosII>> BSP Editor…**Tal como se muestra en la Figura 25.



**Figura 25. Editando el BSP Editor de prueba\_bsp.**

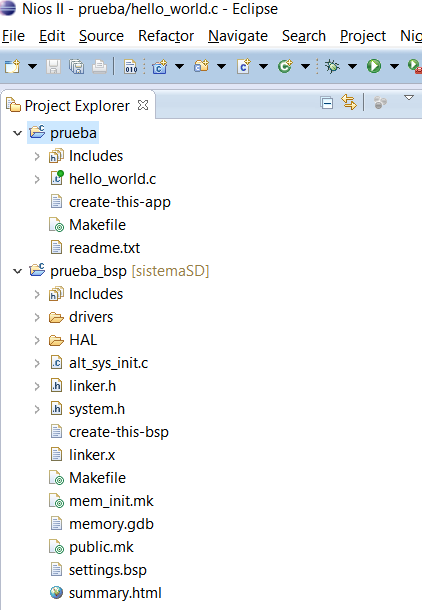
1. Ahora pasaremos a editar en dónde se va a almacenar la imagen cargada de la tarjeta SD. En la ventana de BSP Editor señalamos la pestaña **Linker Script** donde en la columna **Linker Region Name** pasaremos a seleccionar **onchip\_memory2\_0** para todos los dispositivos., luego pasamos a dar clic en **Generate y Exit**. Tal como se muestra en la Figura 26.





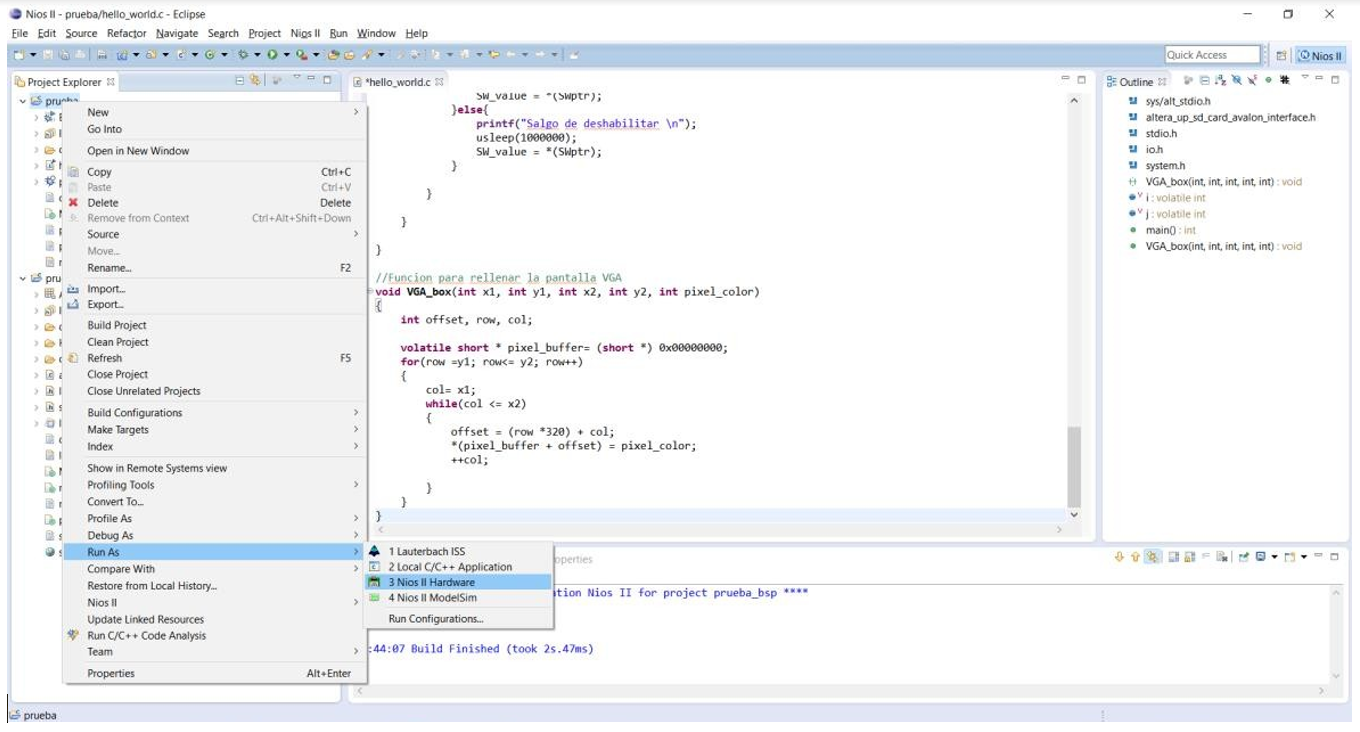
**Figura 26. Editando el BSP Editor de prueba\_bsp.**

1. Luego proceda a hacer clic en el ícono de **Build All ** como se muestra en la Figura 24.



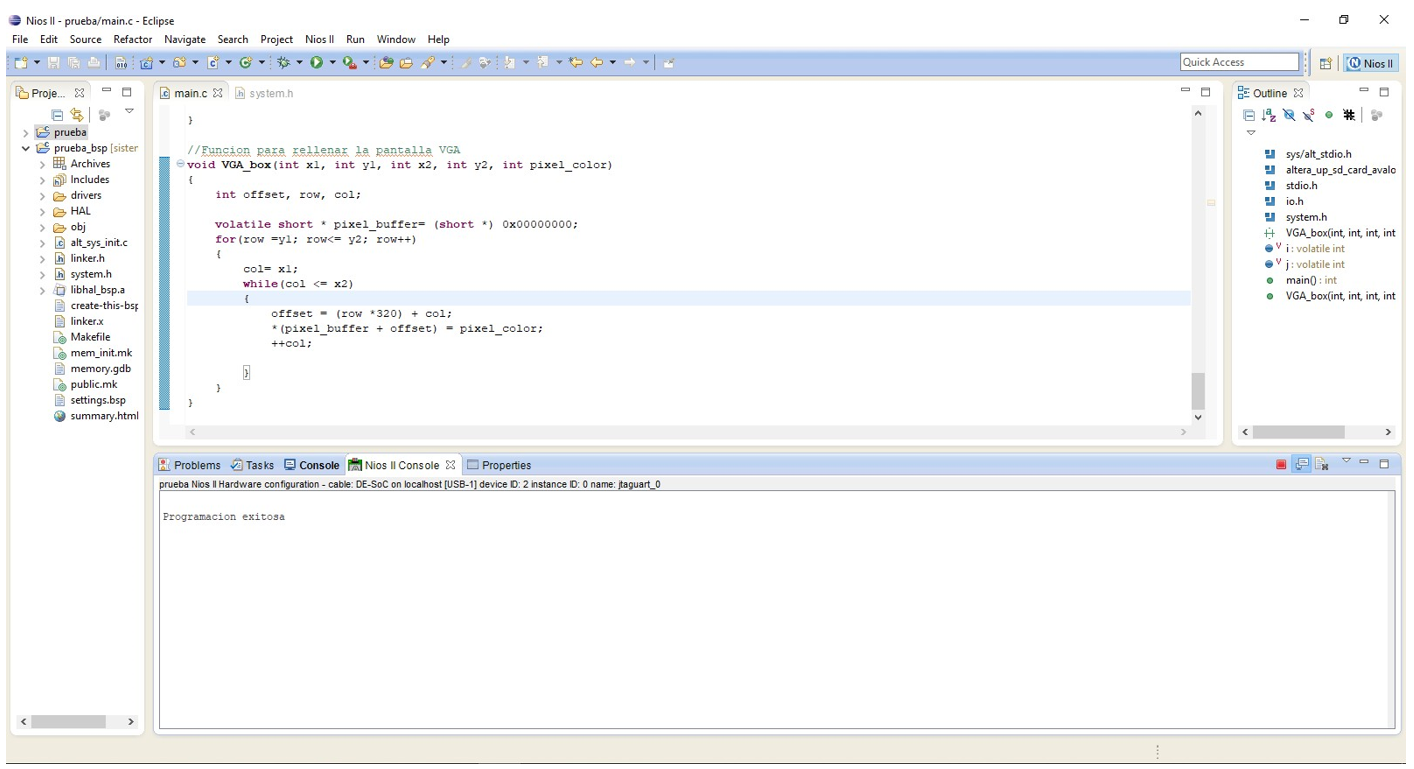
**Figura 24. Compilación del proyecto.**

1. Por último, proceda a hacer clic derecho en la carpeta **prueba1** y seleccione la opción **Run As → Nios II Hardware** como se muestra en la Figura 25.



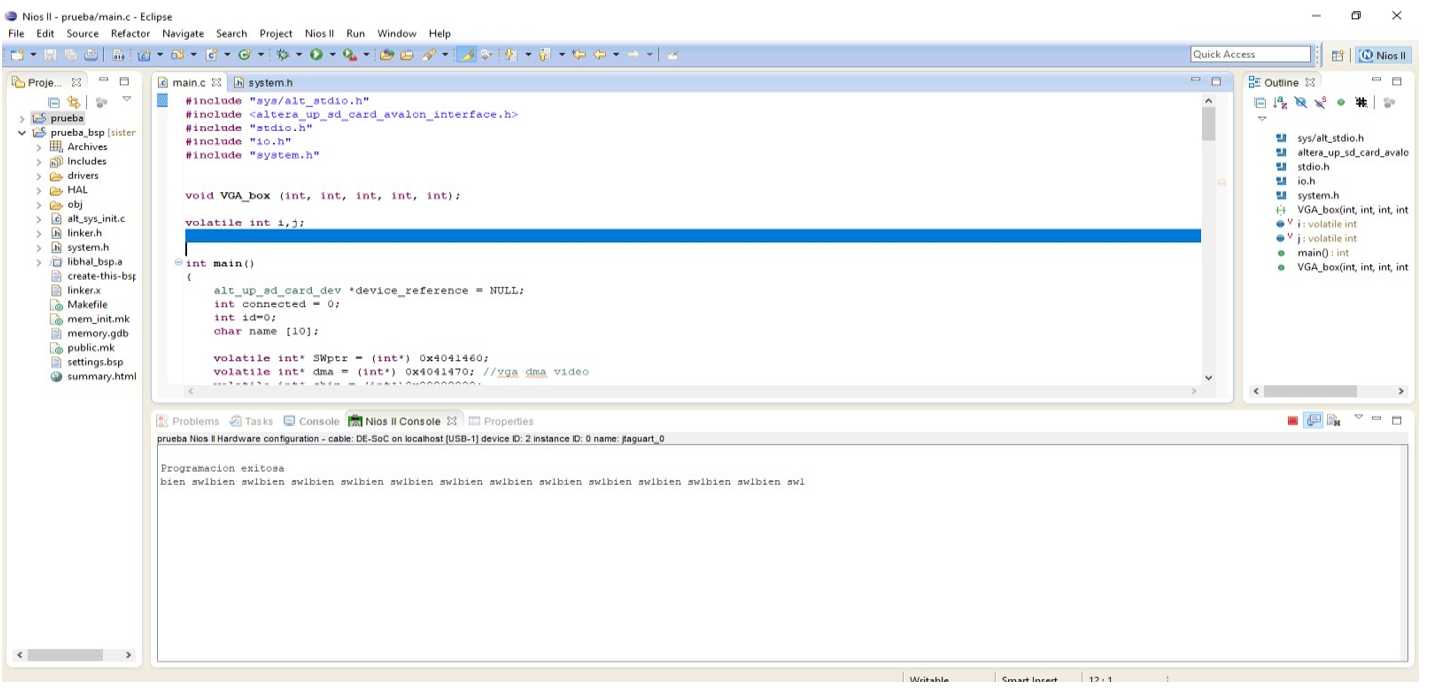
**Figura 25. Programación del microprocesador Nios II en la FPGA.**

1. Si la tarjeta se programó exitosamente, se procederá a observar el siguiente mensaje por consola. Tal como se muestra en la Figura 26. También podremos observar la imagen que se encuentre almacenada en la tarjeta SD.

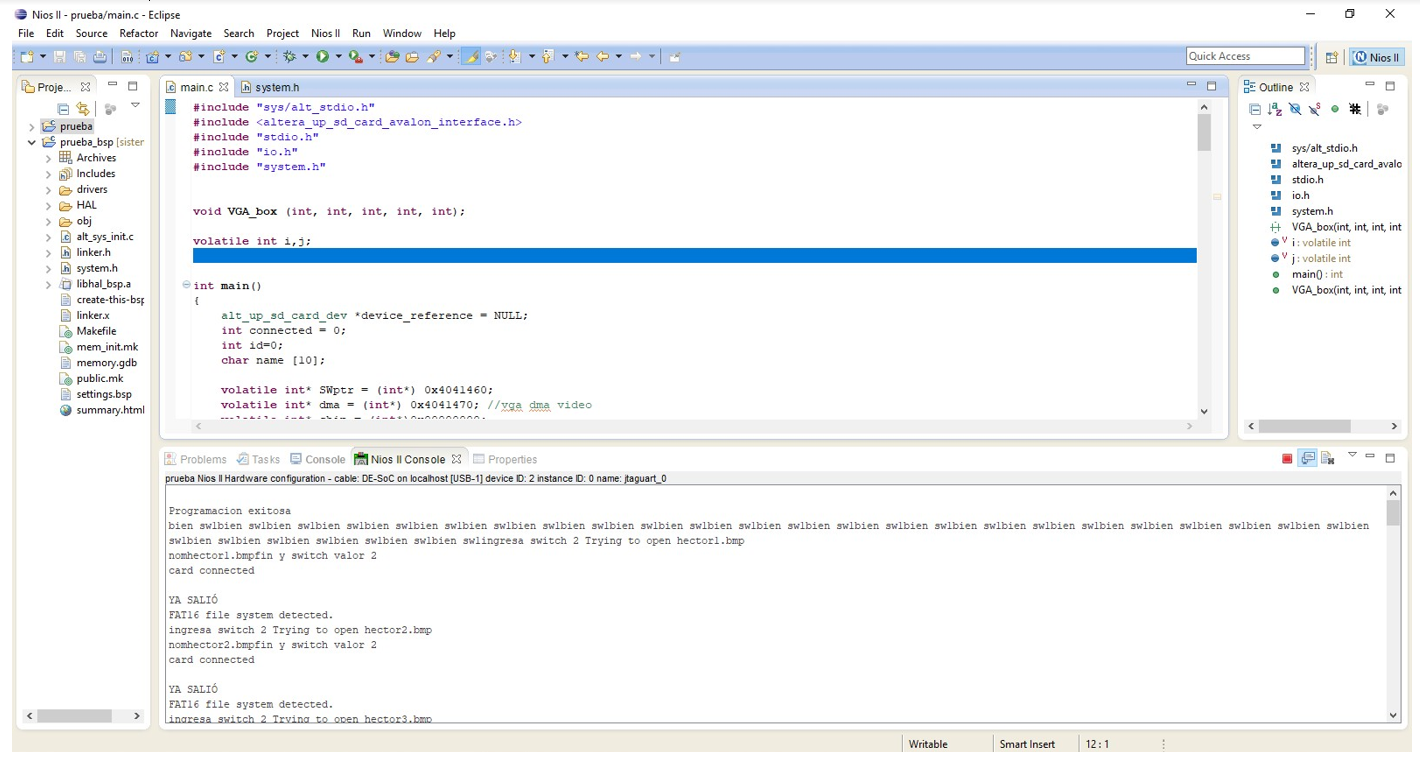


**Figura 26. Salida por consola del programa Eclipse.**

1. Si se **activa** el **switch 1** procedemos a escribir y leer en la SDRAM los frames por segundo captado por la cámara. Luego si lo **desactivamos** se detendrá la escritura y lectura en la SDRAM y el programa estará esperando por una instrucción. Tal como se visualiza en la Figura 27.



**Figura 27. Captura de los frames captados por la cámara y mostrando la salida por consola del programa Eclipse al activar SW1**

1. Si se **activa** el **switch 2** nos aparecerá el mensaje que se observa en la figura 28, el cual nos permite leer la secuencia de imágenes guardadas en la SDCARD. Al desactivar se detiene la lectura y solo se observará por pantalla lo que la cámara está captando.

**Figura 28. Salida por consola del programa Eclipse al activar SW2**

# ENLACE DE DRIVE :

# PROYECTO EN QUARTUS

# https://drive.google.com/open?id=1cHdkOzCvG4aaczt4SagbIajbm7JqKgqq

# REFERENCIA

[1]. Design of Qsys system showcasing Nios II processor - SD card interface, Intel FPGA. Fecha de consulta: Jueves 7 de Noviembre del 2019.

[2]. User Manual for the Intel DE10-Standard Board. Intel.

Fecha de consulta: Martes 5 de Noviembre de 2019.

|  |  |  |  |
| --- | --- | --- | --- |
| **☐** | **☐** | **☐** | **☐** |
| **INICIAL** | **EN DESARROLLO** | **DESARROLLADO** | **EXCELENTE** |