

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
DISEÑO DIGITAL MODERNO  
SEMESTRE 2020 - 2  
  
PROYECTO FINAL  
SNAKE GAME

*ALUMNO:*

Murrieta Villegas Alfonso

*PROFESORA:*

Elizabeth Fonseca Chávez

*FECHA DE ENTREGA:*

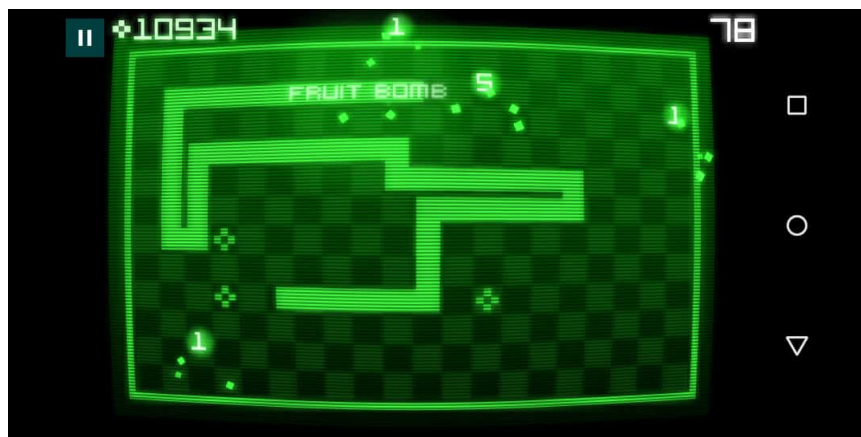
15 de mayo del 2020

*GRUPO TEORÍA: 4*

*GRUPO LABORATORIO: 3*

## Argumento del Juego

Snake es un juego donde el jugador o usuario controla una larga y delgada criatura, semejante a una serpiente, que vaga alrededor de un plano delimitado, recogiendo alimentos (o algún otro elemento), tratando de evitar golpear a su propia cola o las "paredes" que rodean el área de juego. (Ver imagen inferior, juego hecho por Nokia emulado)



Cada vez que la serpiente se come un pedazo de comida, la cola crece más, provocando que aumente la dificultad del juego. El usuario controla la dirección de la cabeza de la serpiente (arriba, abajo, izquierda o derecha) y el cuerpo de la serpiente la sigue. Además, el jugador no puede detener el movimiento de la serpiente, mientras que el juego está en marcha.

## Introducción

El presente proyecto es el resultado de todo lo aprendido a lo largo del semestre acerca del Diseño Digital Moderno, desde la planificación de algoritmos y arquitecturas de software mediante Cartas ASM y Diagramas de estados hasta el desarrollo y pruebas de escritorio mediante un lenguaje dedicado al manejo de FPGA's y hardware como es VHDL.

El proyecto en el apartado de desarrollo tendrá en primera instancia todo el apartado de diseño y lógica previa a la codificación, esto sobre todo para poder entender cómo es que se llevó a cabo de forma metódica y ordena el desarrollo del juego, por otro lado, el código como diagrama RTL del proyecto.

Por último, el apartado de pruebas de escritorio y los correspondientes resultados obtenidos.

## Objetivo

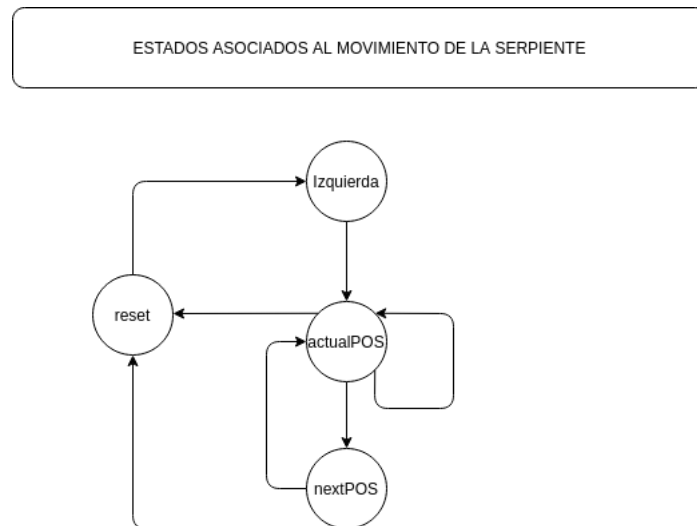
- 1) Mediante todo los conocimientos adquiridos a lo largo de este semestre realizar y culminar un proyecto a través del Diseño Digital Moderno
- 2) Utilizar recursos como Diagramas de Estados, Cartas ASM o mapas de Karnaugh para desarrollar y realizar el diseño previo a la codificación
- 3) Desarrollar un videojuego mediante el lenguaje de programación VHDL

## Desarrollo

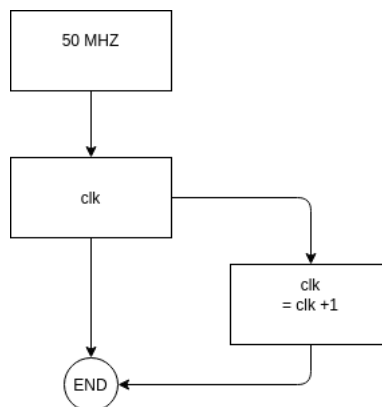
### Diagramas

Antes de escribir el código en VHDL , fue necesario realizar diagramas auxiliares para la comprensión de este trabajo, es por ello que a continuación se muestran tanto Diagramas de Estados como Cartas ASM de varios procesos del snake:

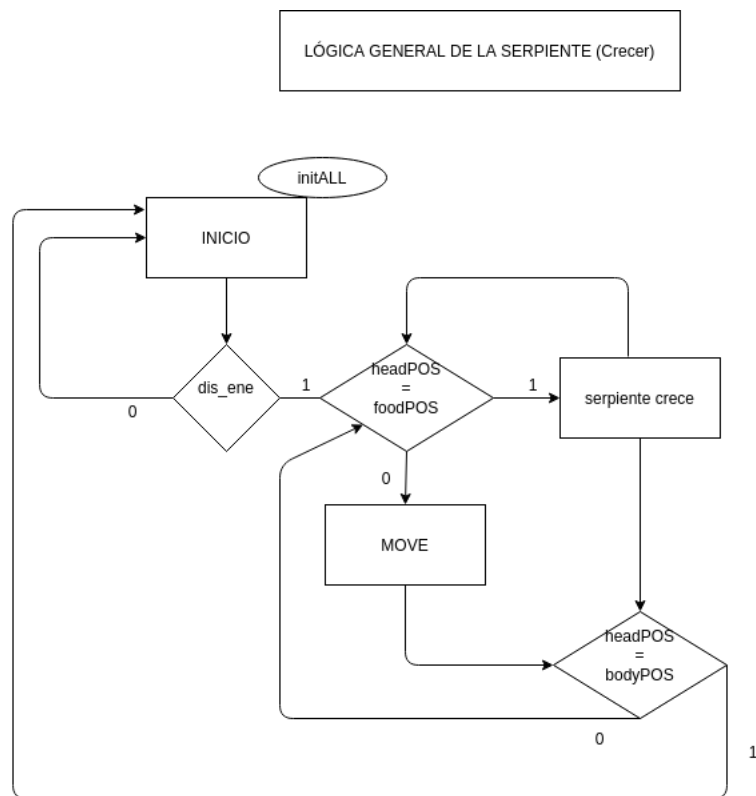
A continuación se muestra la carta ASM asociada a los estados de movimiento de la serpiente:



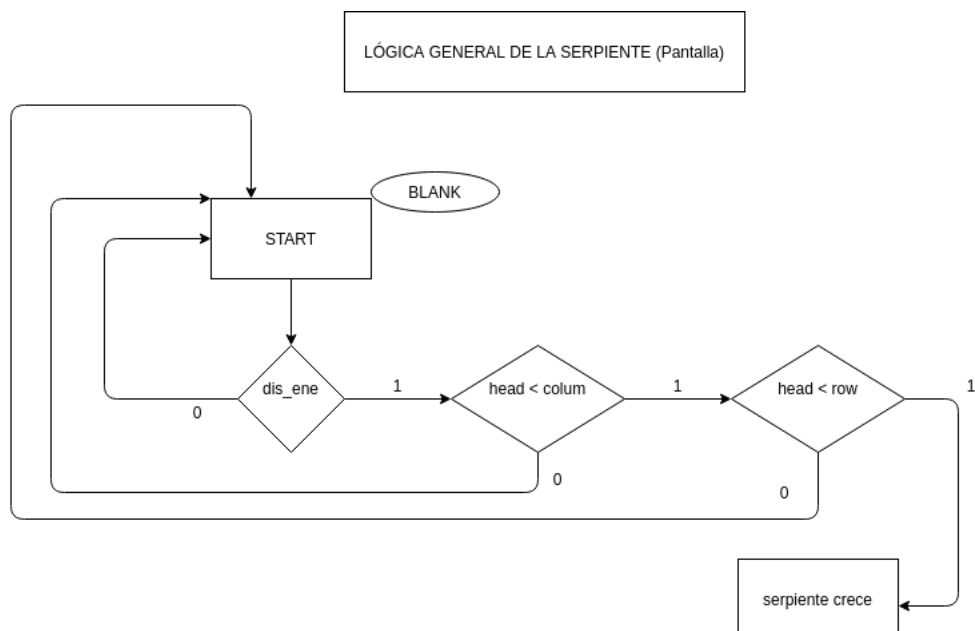
A continuación se muestra el diagrama de estados asociado al divisor de frecuencia:



A continuación se muestra el diagrama de estados asociado al crecimiento de la serpiente:



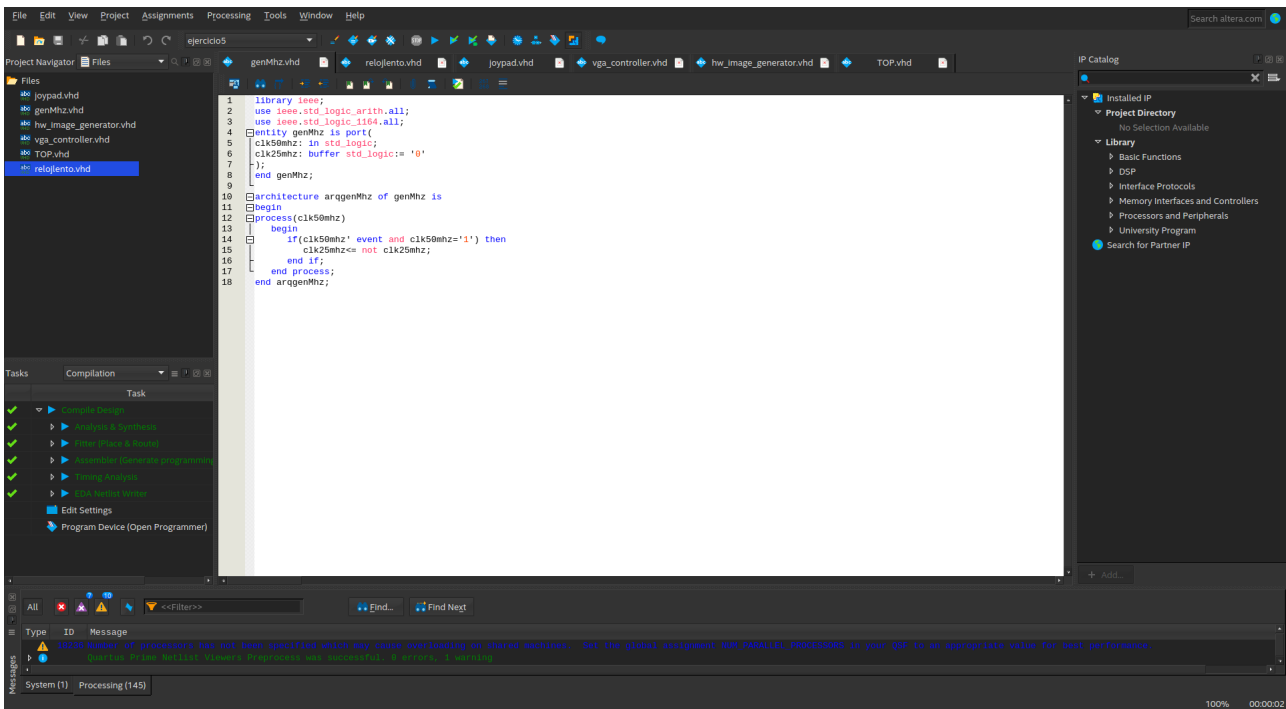
A continuación se muestra el diagrama de estados asociado a las limitaciones de movimiento de la serpiente en la pantalla empleando el VGA:



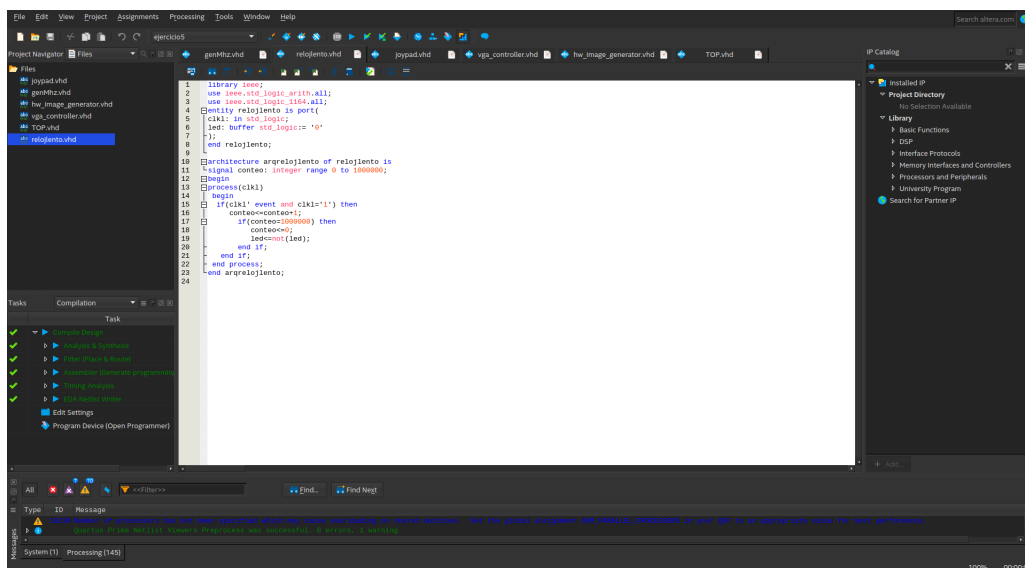
## Código y Funcionamiento

Para comprender el funcionamiento del proyecto es necesario destacar la modularidad que se tiene dentro del mismo para ello se hacen explícitos 6 códigos, el primero

A continuación se muestra el código asociado al divisor de frecuencia el cual fue utilizado principalmente para el controlador del VGA:

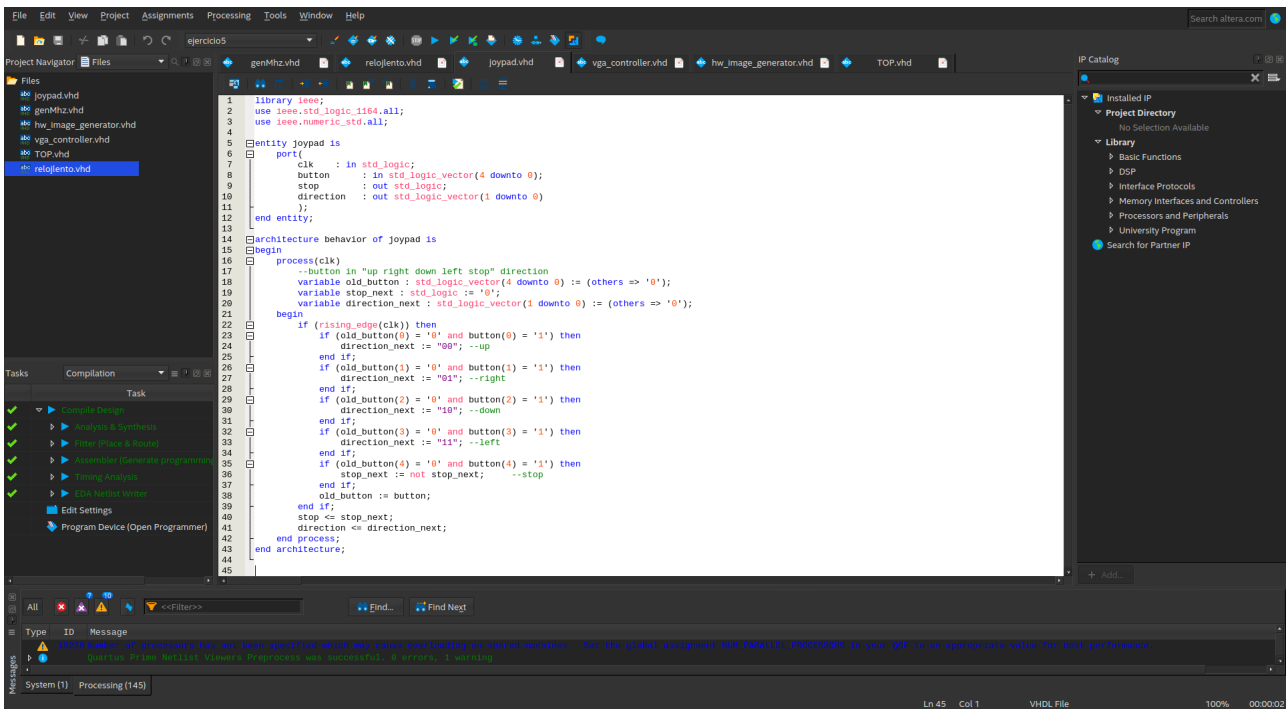


A continuación se muestra el código asociado al reloj lento usado principalmente para el movimiento de la serpiente

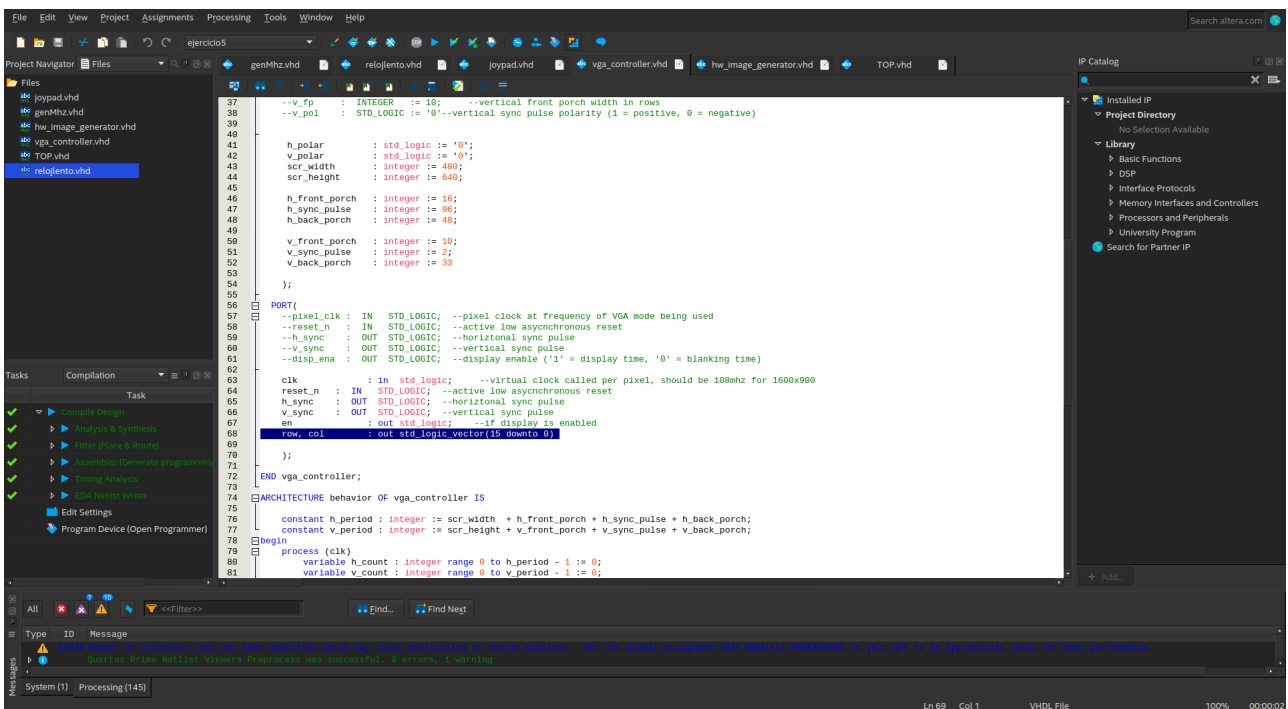


A continuación se muestra el código asociado al guardado de los movimientos de la serpientes denominado como 'joypad', como se observa solamente se tiene 3 variables uno para el guardado de la dirección en el

momento, otro pque es un vector que guarda toda la información asociada a los estados de los switches y del stop.

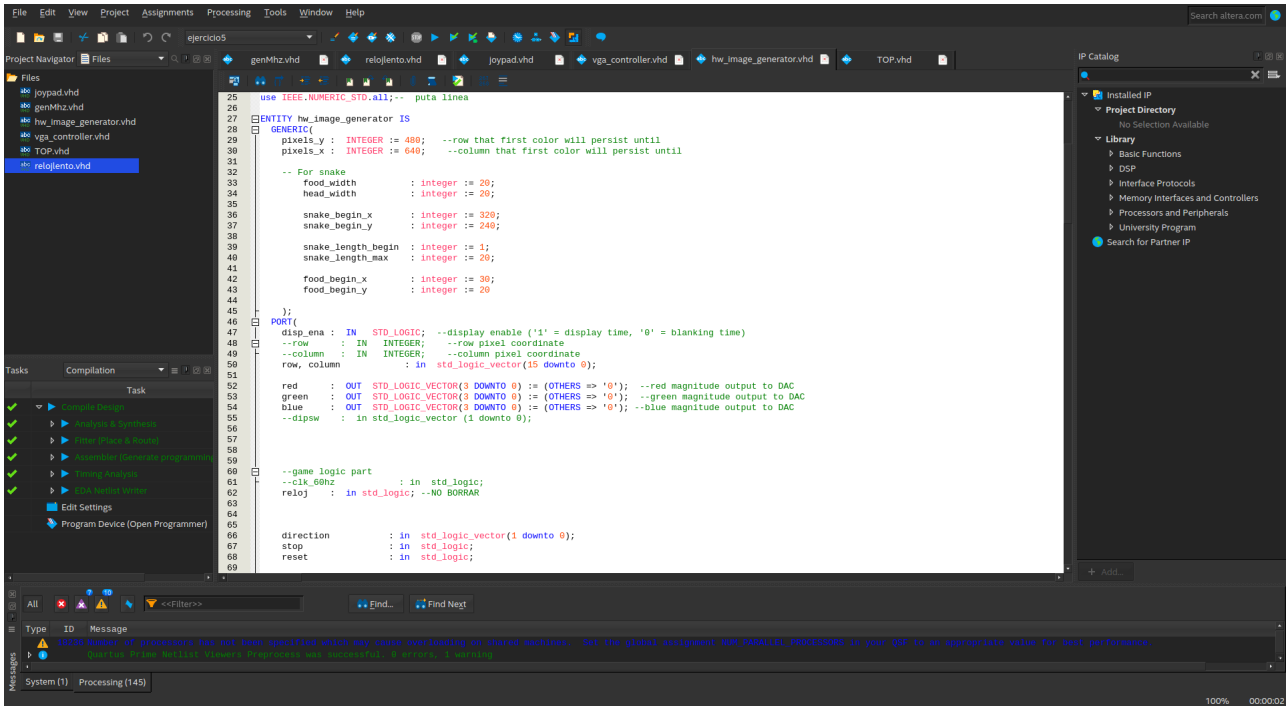


A continuación se muestra el código asociado al controlador del puerto VGA (Este código fue compartido por la profesora y tiene su licencia de uso y de derechos e la parte superior)

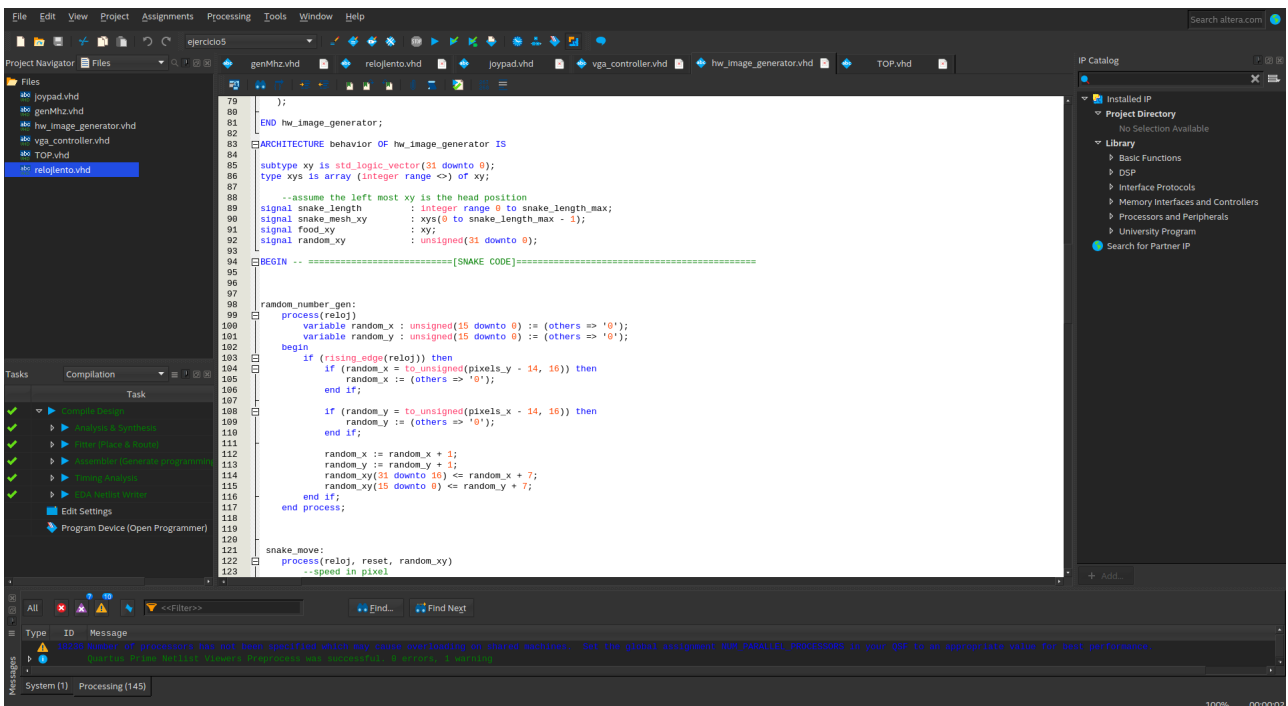


A continuación se muestra el código asociado a toda la lógica principal del juego denominado como 'hw\_image-generator',

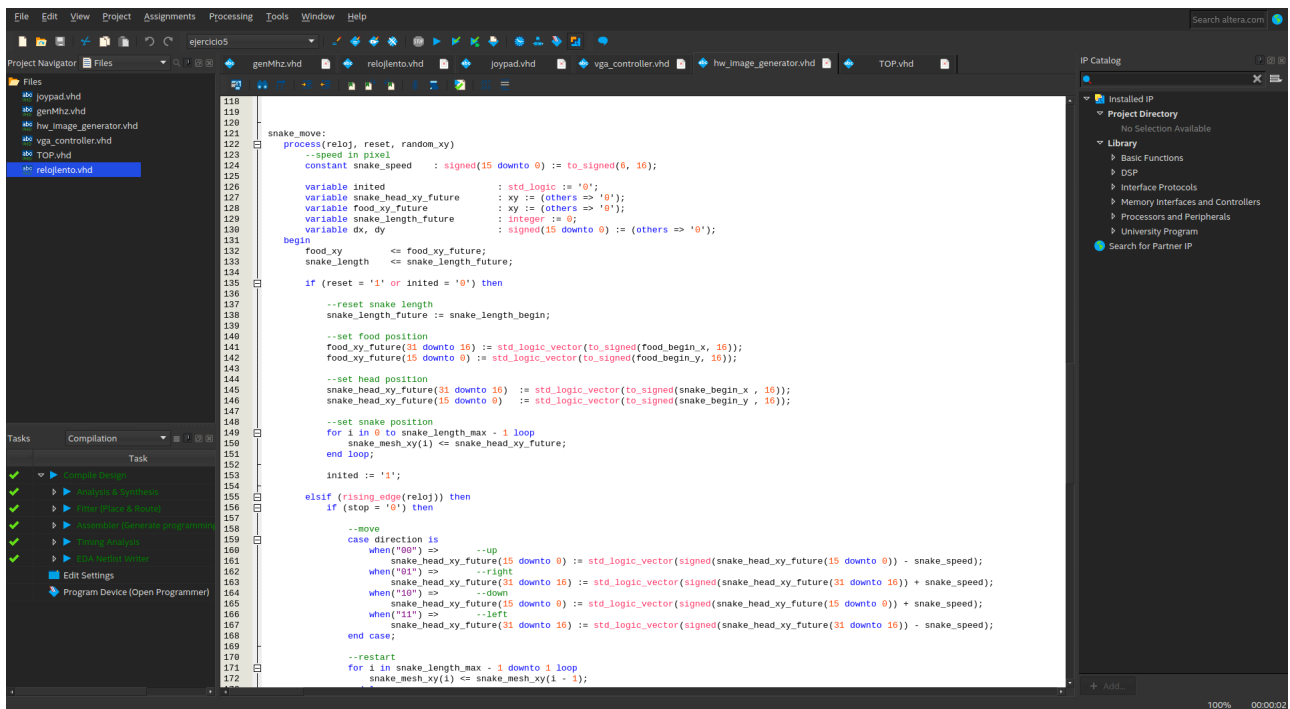
está conformado principalmente de 3 procesos de los cuales 'draw', snake\_move y random\_number\_generator:



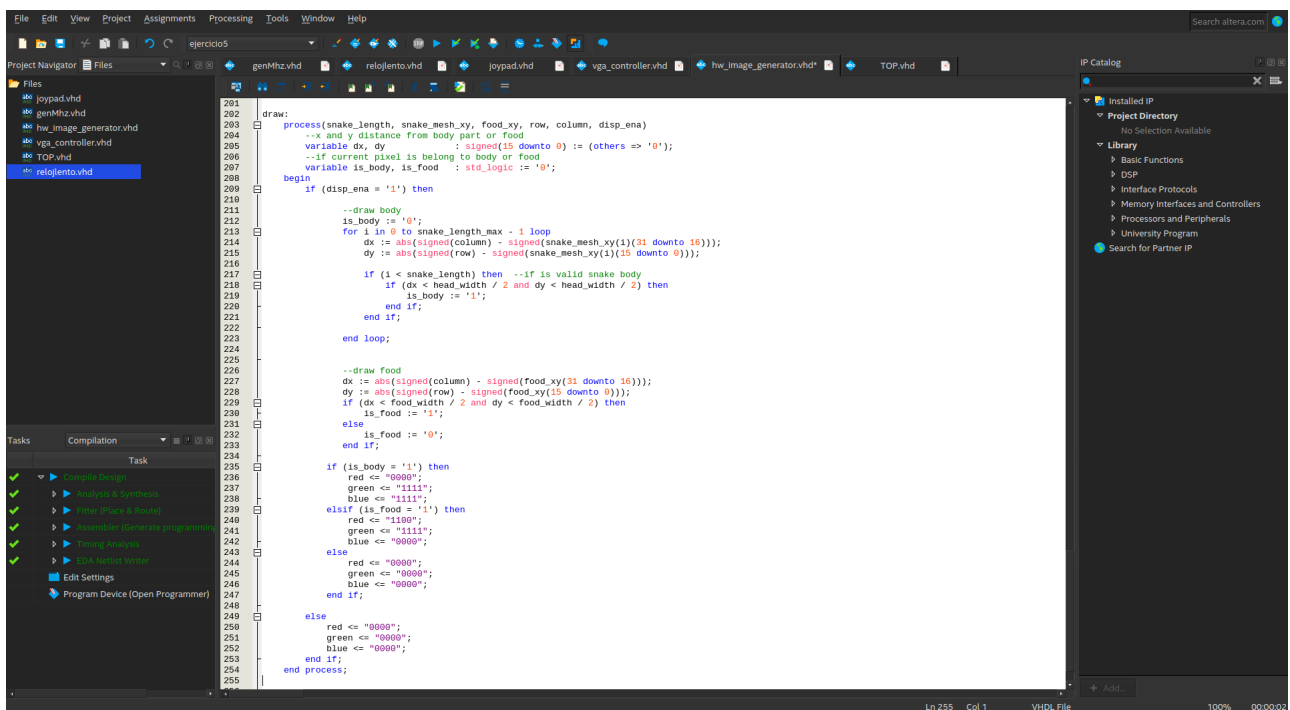
En este apartado se muestra a detenimiento del código asociado a la generación de numerosa aleatorios



En este apartado se muestra a detalle el código del proceso dedicado únicamente a la validación, colisión y creación de todos los objetos del juego que son el cuerpo de la serpiente, la cabeza de la serpiente y la comida.

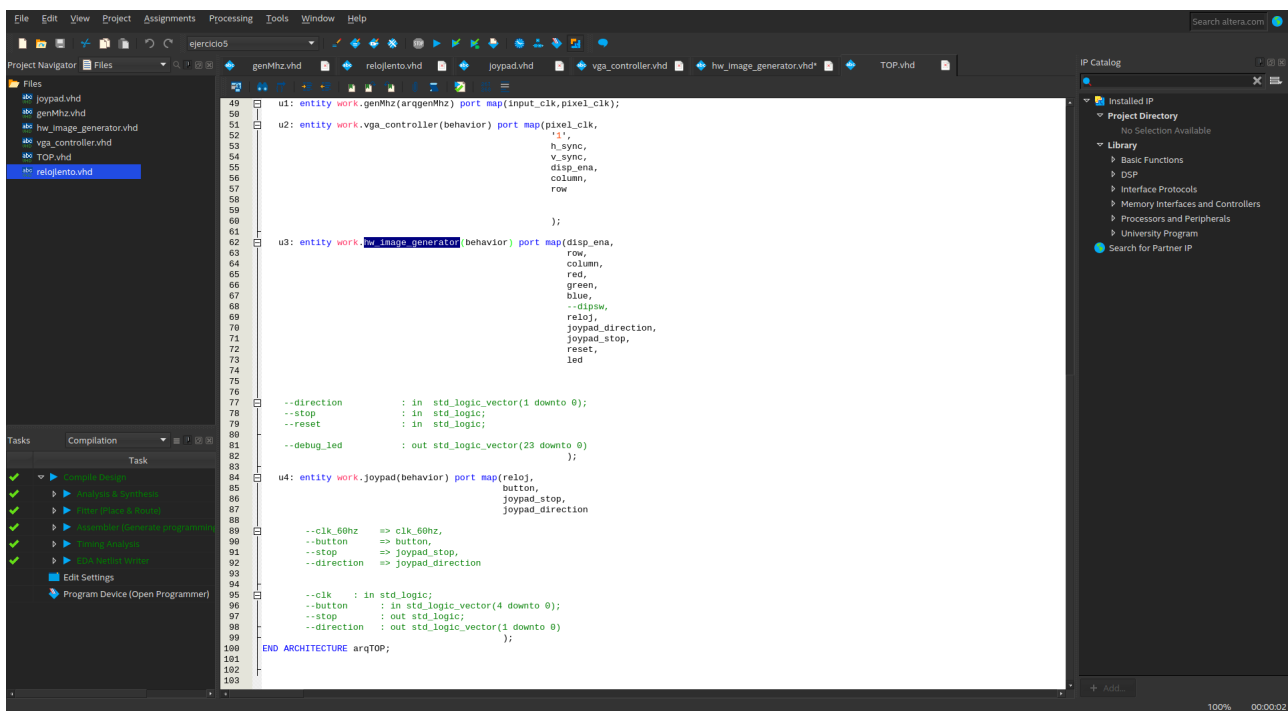


En este apartado se muestra a detalle el código del proceso dedicado únicamente al "dibujado." despliegue de datos en la pantalla:

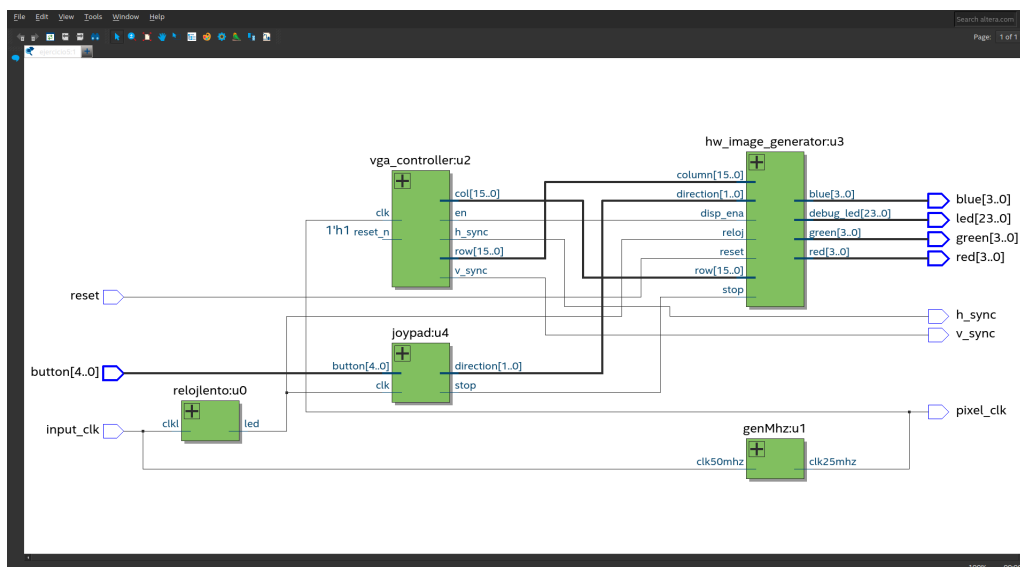


Para terminar el apartado de código, se muestra el código TOP el cual solamente hace la llamada a los códigos anteriores:





Por último, el diagrama RTL obtenido de Quartus, aquí principalmente observamos como es que se unen los componentes anteriores entre sí:



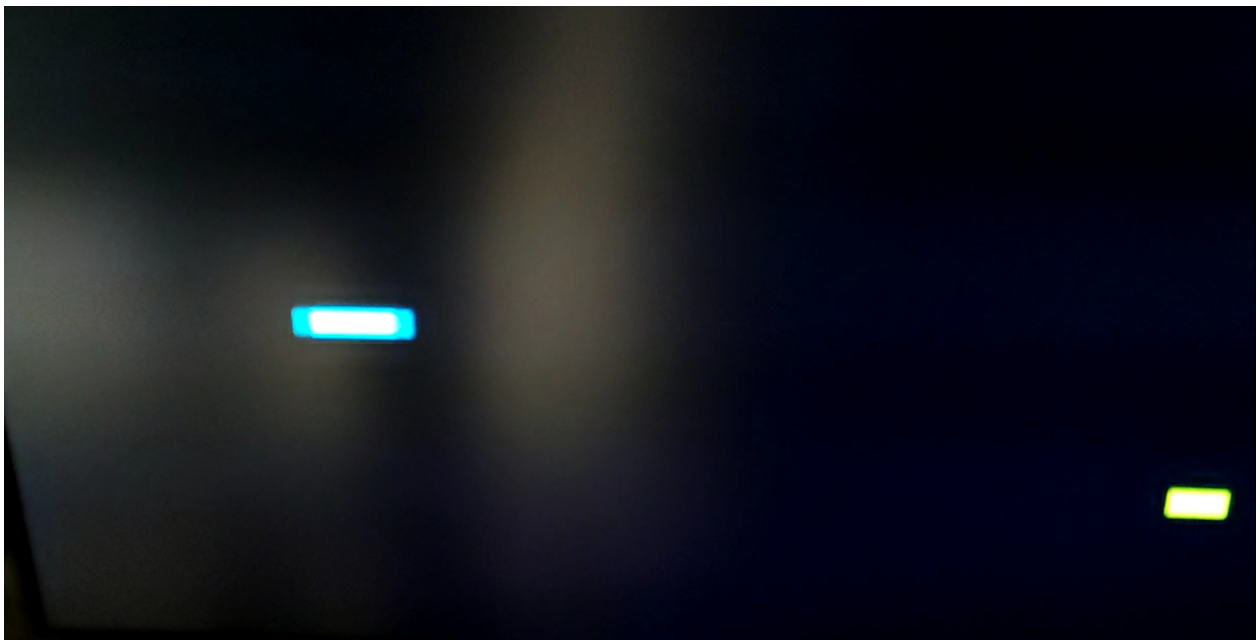
## Pruebas de Escritorio

A continuación seccionaré los diferentes casos oportunos del proyecto para poder ver todo lo que se pudo lograr:

En la siguiente imagen se aprecia el juego ya ejecutado, de color amarillo la comida, de color azul el snake:



Por otro lado, aquí se aprecia el crecimiento de la serpiente debido a que ya ha colisionado con comida y por ende ha crecido:



Para mayor detalle, recomiendo el ver el vídeo final de este proyecto (Estará en mi canal a partir del 16 de mayo de 2020).

## Conclusiones

En el presente trabajo a través de recursos gráficos como la carta ASM o el diagrama de estados es como se pudo entender y realizar el código en VHDL de un contador de bits, sin duda, este tipo de recursos facilitan la comprensión de proyectos de diseño digital, tal vez en este caso sea muy obvio lo que hace y cómo funciona el contador, sin embargo, en proyecto más grandes observaremos que a veces un recurso gráfico como estos son de mucha ayuda.

Sin duda el presente proyecto final fue un arduo trabajo que es sin duda el resultado de todo un semestre de haber aprendido Diseño Digital Moderno, la realidad es que a pesar de que visualmente el juego no se vea nada llamativo además de no contar con algunos elementos esenciales de muchos juegos, satisfactoriamente puedo decir que es un buen resultado incluso personal al haber logrado llevar a cabo algo como esto en un lenguaje como VHDL.

Por otro lado, algunas modificaciones y cosas que podrían agregarse a este proyecto son: 1) La inclusión de un contador visual en la pantalla donde se lleven los puntos de la partida

2) El guardado de perfiles y puntos de cada jugador a través del tiempo, esto sobre todo es una curiosidad que resolveré pronto pues a pesar de que en la FPGA se puede crear una unidad dedicada a la ROM, me gustaría sobre todo saber si hay alguna forma de extraerle datos a dispositivos externos.

3) Notablemente un mejor diseño para el juego, como es visible el diseño (Aspecto estético) del juego es prácticamente nulo, sin embargo, como objetivos del proyecto no eran necesario este aspecto.

## Referencias

- 1) Youtube Alfonso Murrieta Villegas. Recuperado el 20 de marzo de 2020, de <https://www.youtube.com/watch?v=O.KOQBGj-Aw>
- 2) Manual de Usuario. Recuperado el 2 de abril de 2020, de [https://www.intel.com/content/dam/altera-www/global/en\\_US/portal/dsn/42/doc-us-dsnbk-42-2912030810549-de10-lite-user-manual.pdf](https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-2912030810549-de10-lite-user-manual.pdf)
- 3) Morris, M. Diseño Digital. México, México. Pearson Educación.
- 4) Práctica 9 del Laboratorio de Diseño Digital. Recuperado el 2 de abril de 2020, de [https://bloglabdsd.files.wordpress.com/2015/02/practica9\\_parte2\\_-asm.pdf](https://bloglabdsd.files.wordpress.com/2015/02/practica9_parte2_-asm.pdf)