

**Escuela Politécnica Nacional**

**Análisis de Sistemas Informáticos**

**Programación Avanzada**

**Integrantes:**

**José Cortez**

**Diana Narváez**

**Kelly Soto**

**GR1**

**2018**

**Contenido**

[**Objetivo General** 2](#_Toc521089530)

[**Objetivos Específicos** 2](#_Toc521089531)

[**Introducción** 3](#_Toc521089532)

[**Desarrollo** 3](#_Toc521089533)

[**Importación de paquetes** 3](#_Toc521089534)

[Pygame 3](#_Toc521089535)

[Random 3](#_Toc521089536)

[Time 4](#_Toc521089537)

[**Clases** 4](#_Toc521089538)

[Cursor 4](#_Toc521089539)

[Botón 4](#_Toc521089540)

[Imágenes 4](#_Toc521089541)

[Funciones 4](#_Toc521089542)

[Pintar Alíen 5](#_Toc521089543)

[Iniciar Música 5](#_Toc521089544)

[VerTXT 5](#_Toc521089545)

[Jugador 6](#_Toc521089546)

[**¿Cómo funciona el juego?** 7](#_Toc521089547)

[**Conclusión:** 7](#_Toc521089548)

[[1]**,** [2]**,** [3]**,** [4]**,** [5] 7](#_Toc521089549)

[**Bibliografía** 7](#_Toc521089550)

[Imagen 1 Vista General Del código del juego 3](#_Toc521089566)

[Imagen 2 Importación de paquetes 3](#_Toc521089567)

[Imagen 3 Uso Time suspensión de procesos. 4](#_Toc521089568)

[Imagen 4 Clase Cursor 4](#_Toc521089569)

[Imagen 5 Clase Botón 4](#_Toc521089570)

[Imagen 6 Menú del Juego 4](#_Toc521089571)

[Imagen 7 Clase Imágenes 4](#_Toc521089572)

[Imagen 8 Pantalla para un Jugador 4](#_Toc521089573)

[Imagen 9 Función Tiempo 5](#_Toc521089574)

[Imagen 10 Función Pintar alíen 5](#_Toc521089575)

[Imagen 11 Sprite alíen 5](#_Toc521089576)

[Imagen 12 Fusión iniciar música 5](#_Toc521089577)

[Imagen 13 pantalla menú señalando botón puntaje 5](#_Toc521089578)

[Imagen 14 Código Función verTxt. 5](#_Toc521089579)

[Imagen 15 Resultado del código de verxt 5](#_Toc521089580)

[Imagen 16 Función Jugador Carga Sprite 6](#_Toc521089581)

[Imagen 17 Bloque 2 de la función Jugador 6](#_Toc521089582)

[Imagen 18 Bloque de código 3 6](#_Toc521089583)

[Imagen 19 Bloque de código 4 7](#_Toc521089584)

[Imagen 20 2 Jugadores 7](#_Toc521089585)

# **Objetivo General**

Elaborar un videojuego tipo Arcade en el lenguaje de programación Python con manejo uso de Pygame y manejo de Sprite.

## **Objetivos Específicos**

* Poner en práctica las funciones aprendidas en Python
* Obtener experiencia en la elaboración de distintos programas
* Alcanzar las metas propuestas en el manejo del lenguaje de programación.
* Aprender el manejo de funciones de Pygame para la creación de un juego.

# **Introducción**

La creación de video juegos es algo que hoy en día es muy popular, la gran mayoría los usa más de una vez al día. La implementación de un videojuego va desde solo disfrutar del juego hasta algo tan complejo como el desarrollo de historias.

Los juegos de tipo arcade tiene como objetivo principal ir incrementado la dificulta conforme se elevan los niveles. Para la implementación de siguiente juego se va a usar los módulos contenidos en Pygame. Como punto importante cabe recalcar el beneficio y uso de Sprite dentro del mismo.

# **Desarrollo**

El juego consta principalmente de una estructura básica con clases y funciones que contiene cada una de las funcionalidades e implementaciones realizadas. Vista general del código imagen 1.

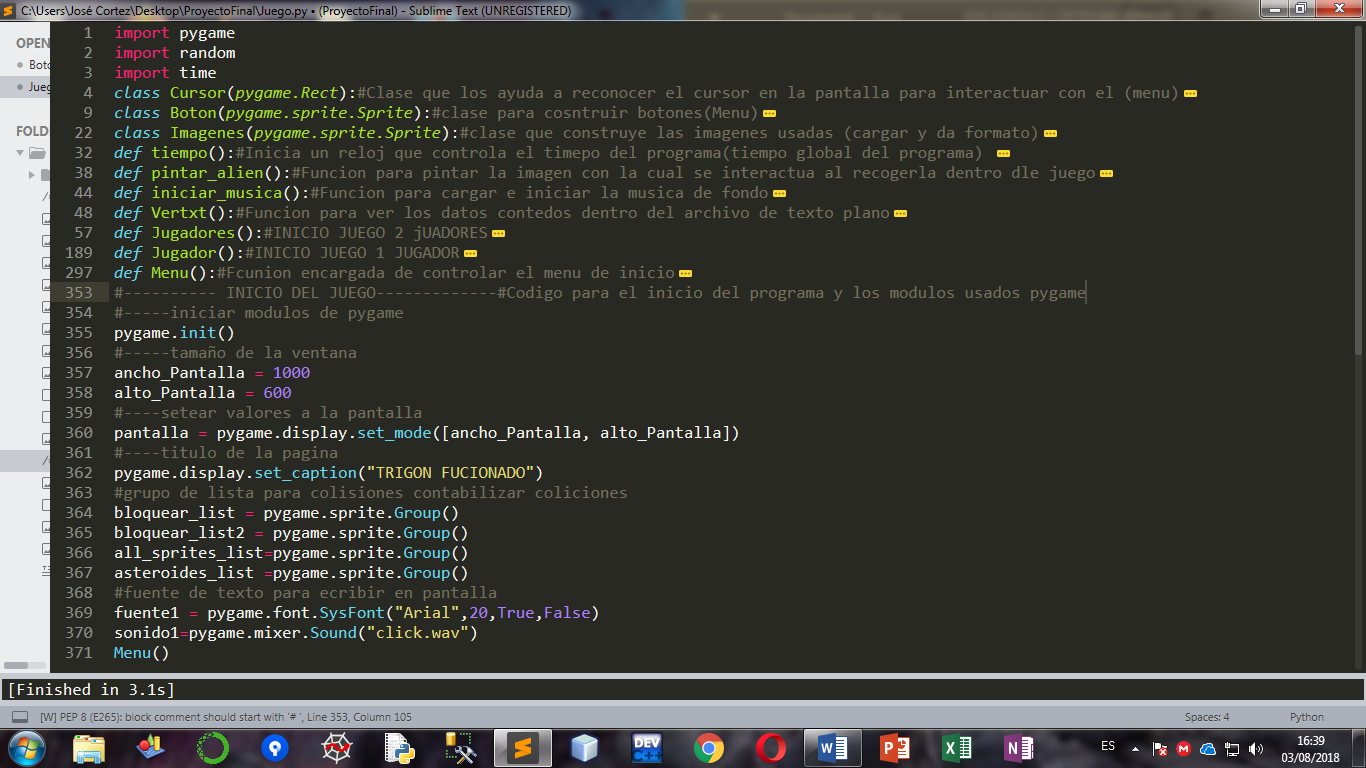


Imagen 1 Vista General Del código del juego

## **Importación de paquetes**

Los paquetes son muy importantes para el desarrollo del juego. Los paquetes usados son 3 Pygame, Time y random. Imagen 2.

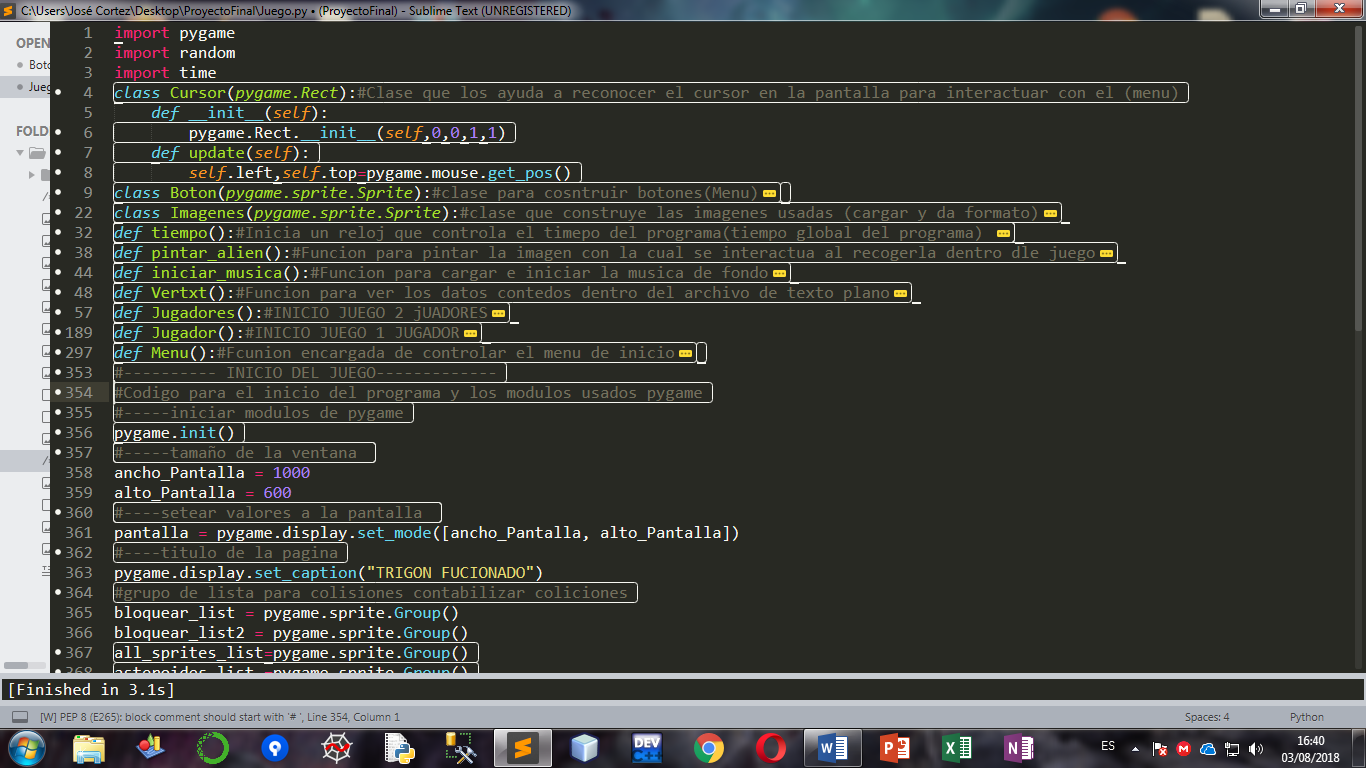


Imagen 2 Importación de paquetes

Pygame es el paquete mas usado dentro del juego. Lista de usos:

-Uso del reloj para tiempo general del programa y para mantener los fotogramas por segundo del juego

-Uso de grupos de Sprite y colisiones de los mismos también usada para cargar el fondo de la pantalla de juego.

-Cargar imagen y usarlas en Sprite.

-Cargar música y sonidos para fondo y eventos.

Random usado para generar posiciones (x,y) de los Sprite usada para ubicar en pantalla el marciano con el que el jugador interactúa.

### Time

usado para la suspensión de programa ocasionalmente. Principal uso es para poder observar los puntajes de los jugadores, para lo cual se da un máximo de 3 segundos usado en la función vertxt, línea específica 56. Imagen 3.

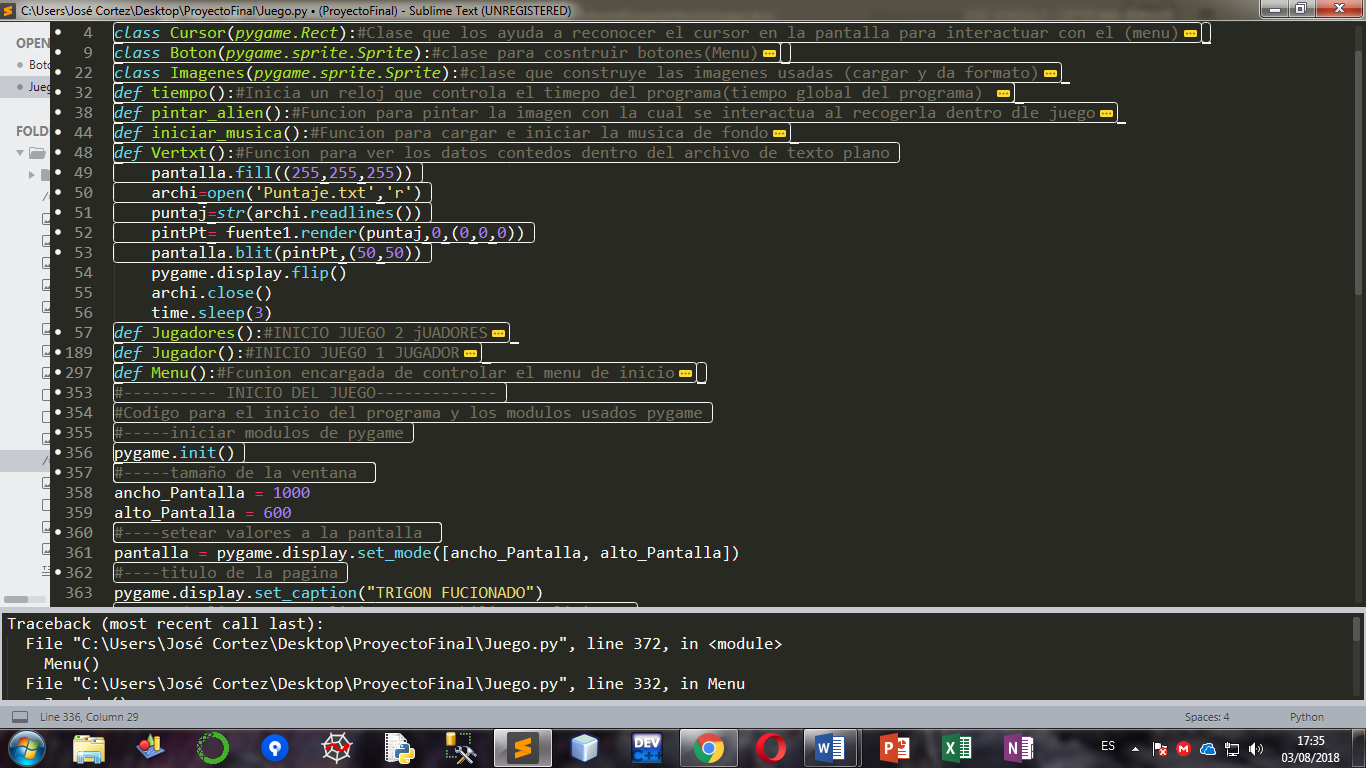


Imagen 3 Uso Time suspensión de procesos.

## **Clases**

Las clases son fragmentos de código independientes que realiza una función dentro de un programa, hacen modular a un programa.

Clases usadas:

### Cursor

Usada para detectar la posición del ratón dentro de la pantalla, del menú y colisión de algún botón. Código perteneciente a esta función. Imagen 4.

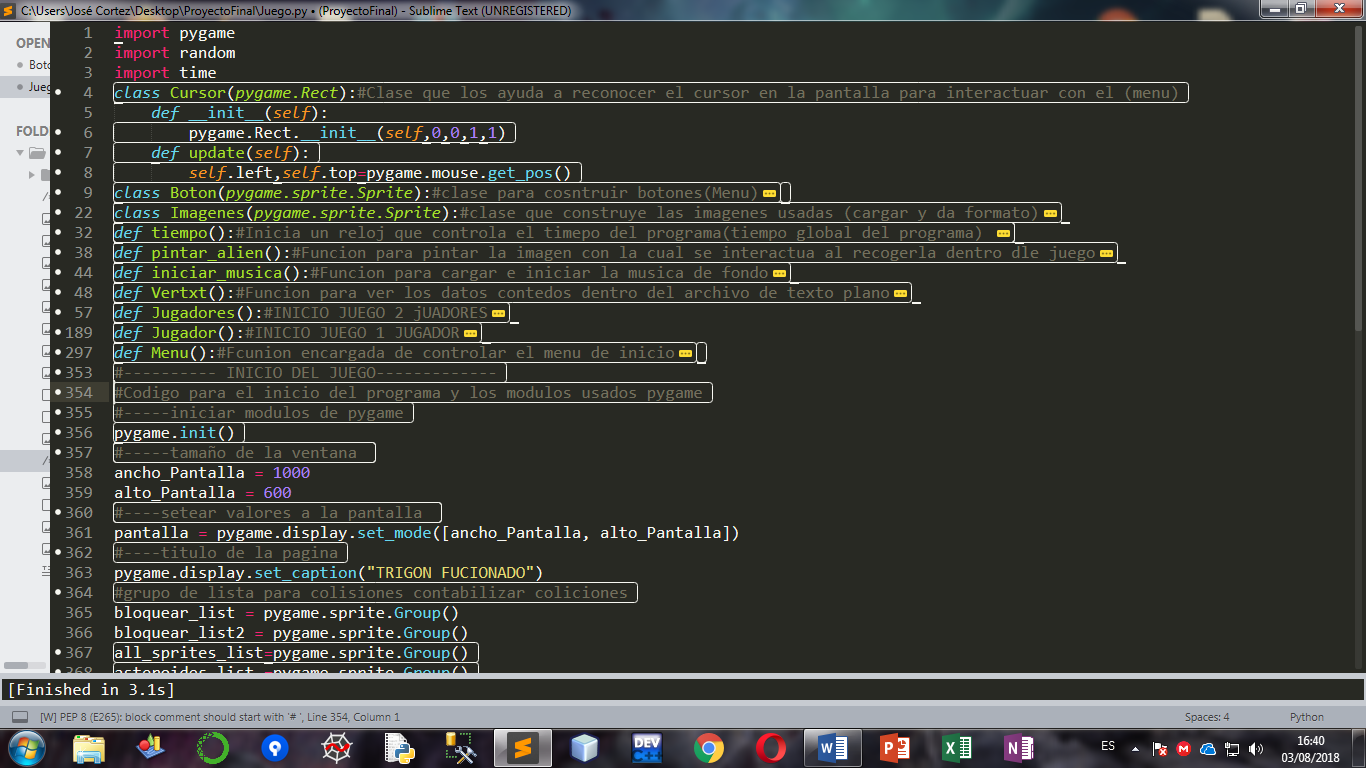


Imagen 4 Clase Cursor

### Botón

Usada para cambio de imágenes contenidas en el botón, efectos de los botones. También la función actualizar(update) se usa para detectar una posible colisión del mouse con algún botón.

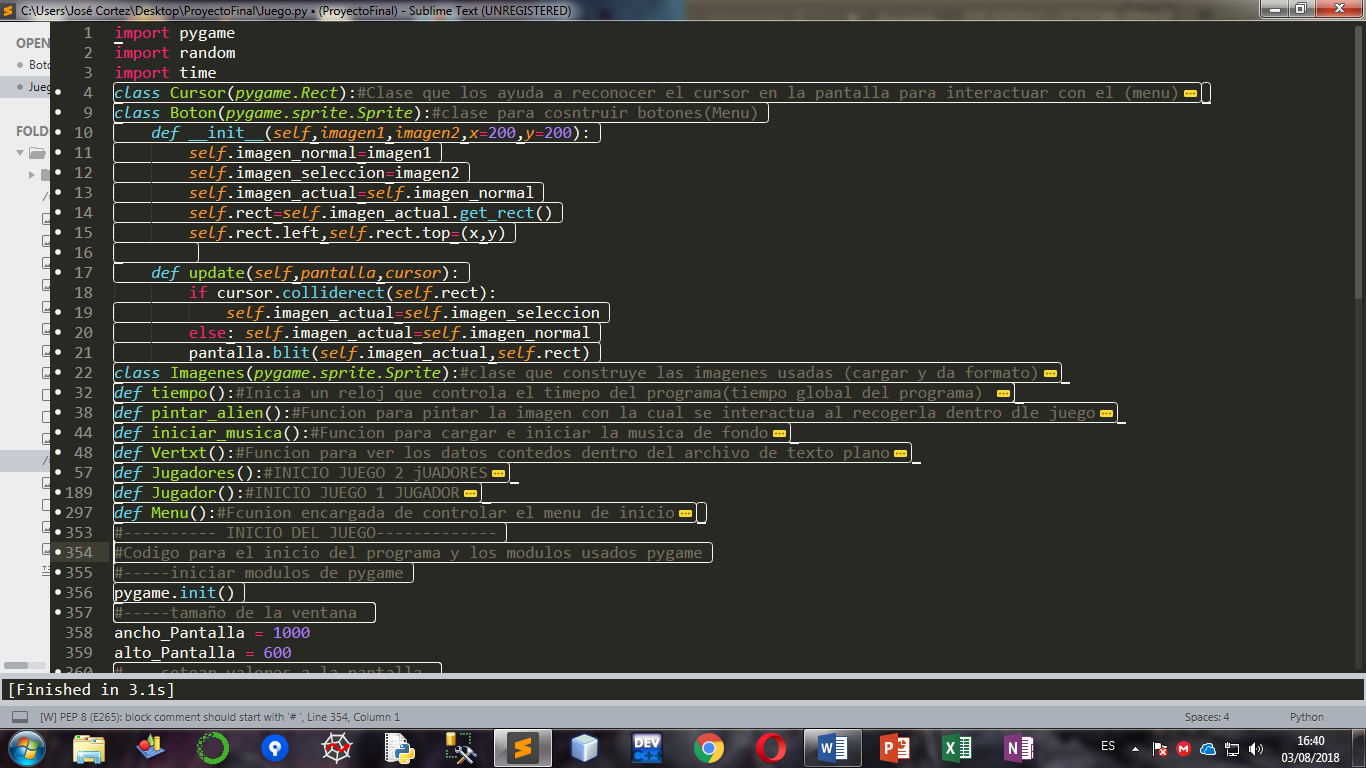


Imagen 5 Clase Botón

Estos códigos con ayuda de la función Menú despliegan la pantalla inicial que se presenta en el juego. Imagen 6. La función Menú se explicará más adelante.

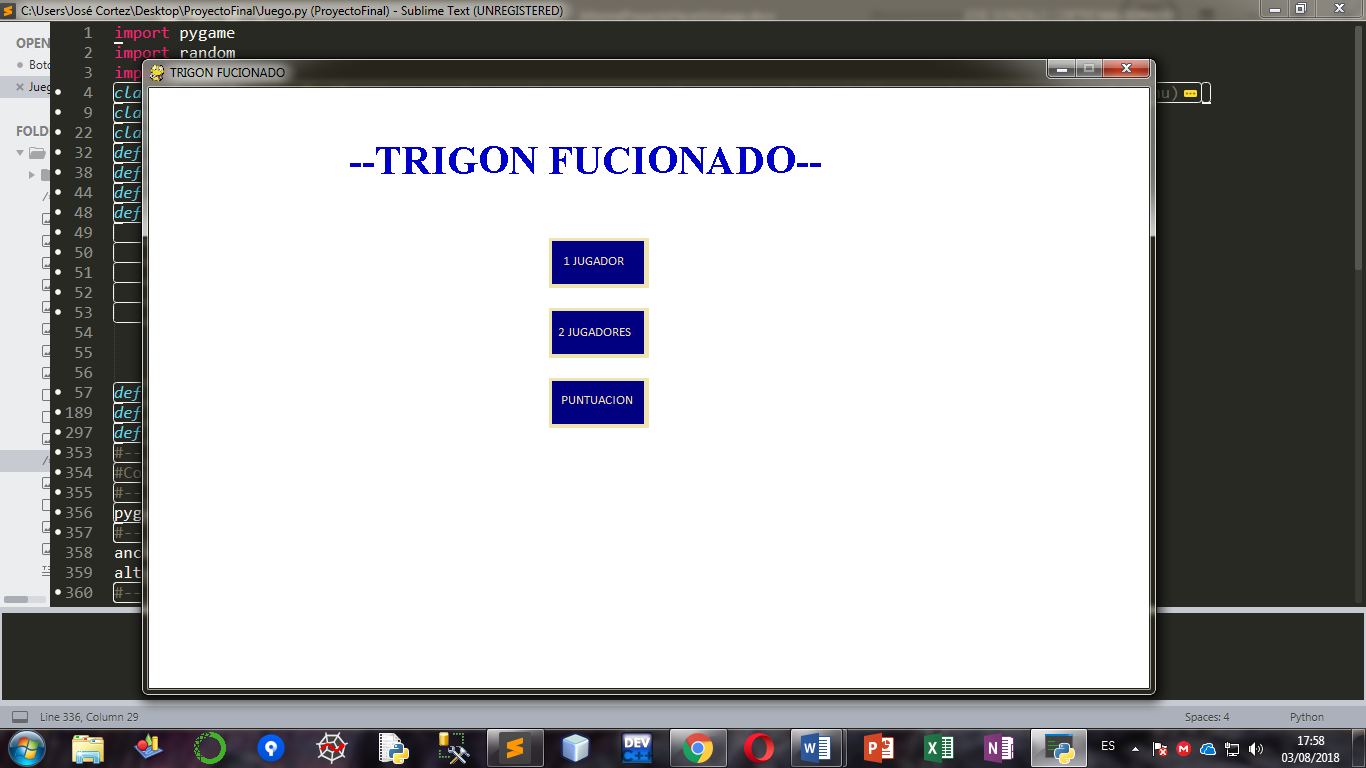


Imagen 6 Menú del Juego

### Imágenes

Esta clase dentro del constructor de la clase llama a los constructores de la clase padre. Esta clase es la encargada de cargar las imágenes de todo el juego asignarles una superficie dentro de la pantalla. La clase es usada para ser Pintar y colisionar imágenes. La imagen 7 muestra el código.

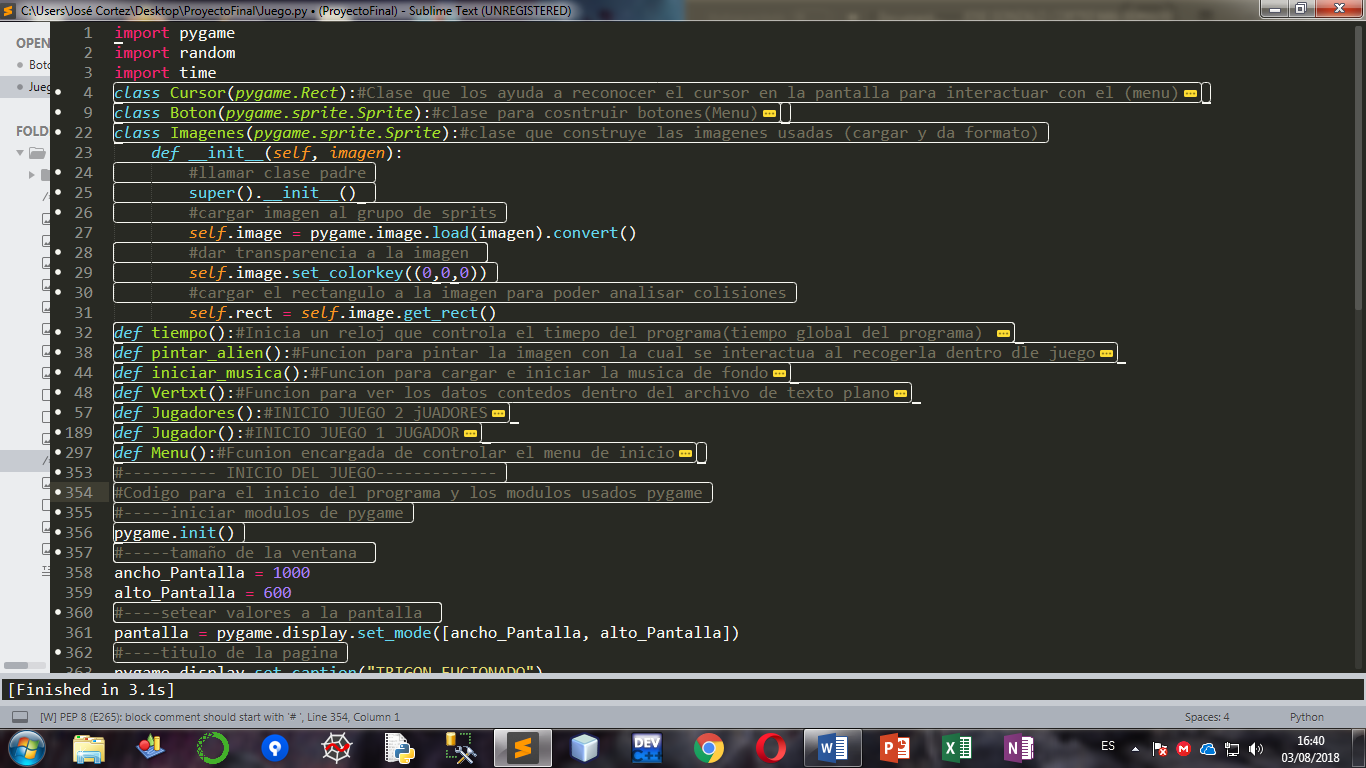


Imagen 7 Clase Imágenes

En la línea 29 se asigna transparencia al color de fondo negro por el cual las imágenes a excepción de meteoro no se ven con fondo. Vista general del juego imagen 8.

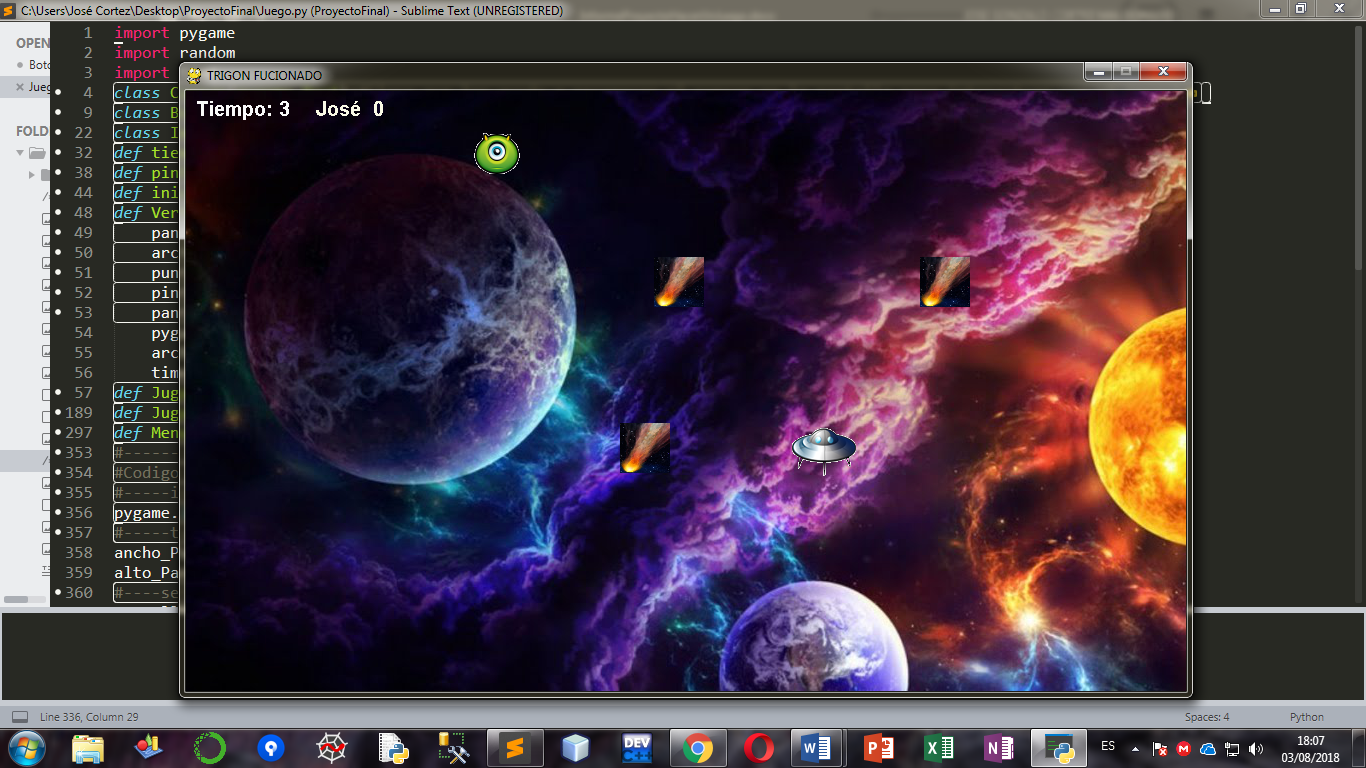


Imagen 8 Pantalla para un Jugador

## Funciones

Cada función se destina para implementar un fragmento del código.

Funciones usadas:

Tiempo

Usada para devolver el tiempo general del programa en milisegundos, al dividirle para 1000 los dará en segundos. Esta lista para pintarse en la pantalla, pero no se la usa mas que para identificar la diferencia del tiempo dentro del juego y el tiempo general del programa.

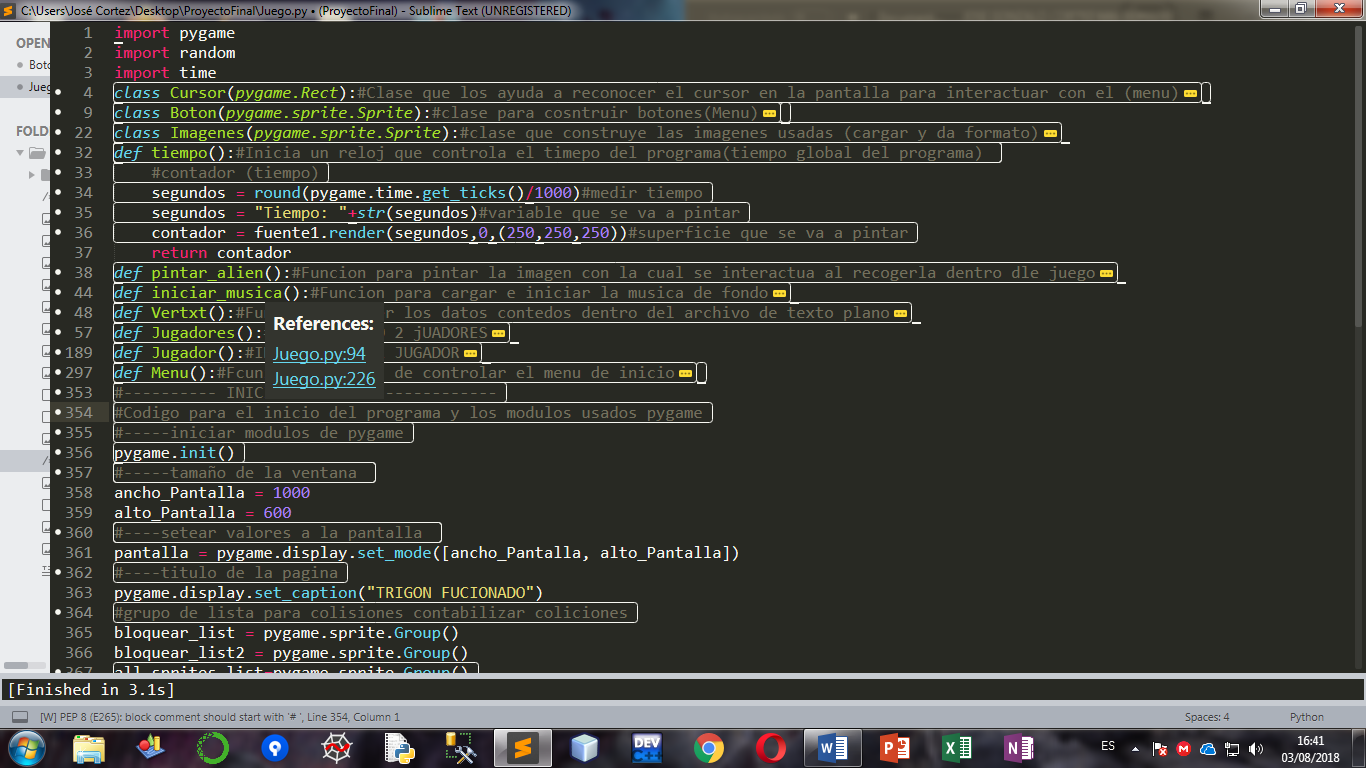


Imagen 9 Función Tiempo

### Pintar Alíen

Función usada para pintar el Sprite con de un alíen para que se pueda interactuar como elemento a recoger por los Sprite de la naves y platillo. Código Imagen 10.

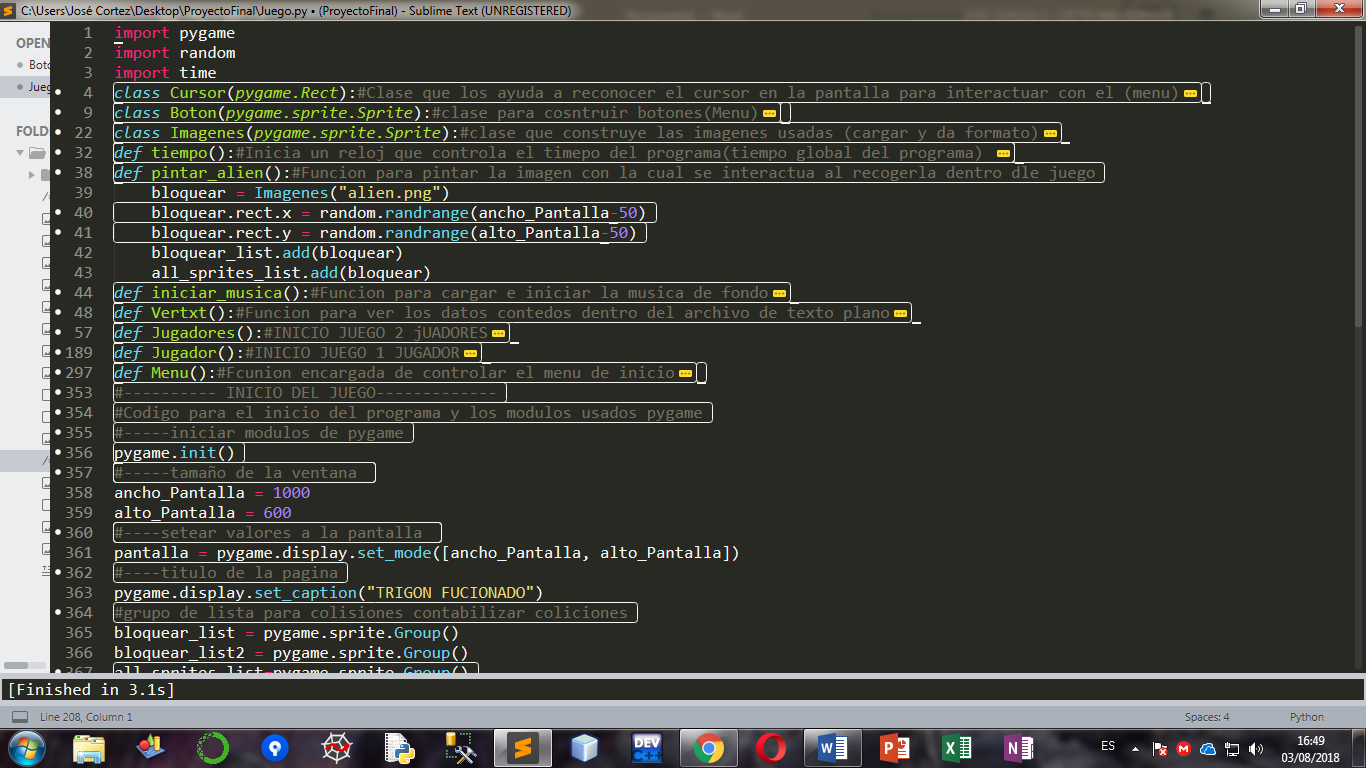


Imagen 10 Función Pintar alíen

El Sprite usado en esta clase se puede ver en la imagen 11 dentro de un rectángulo verde.



Imagen 11 Sprite alíen

### Iniciar Música

Esta función cargar la música de fondo y la inicia para que empieza a sonar al inicio del juego. Código usado imagen 12.

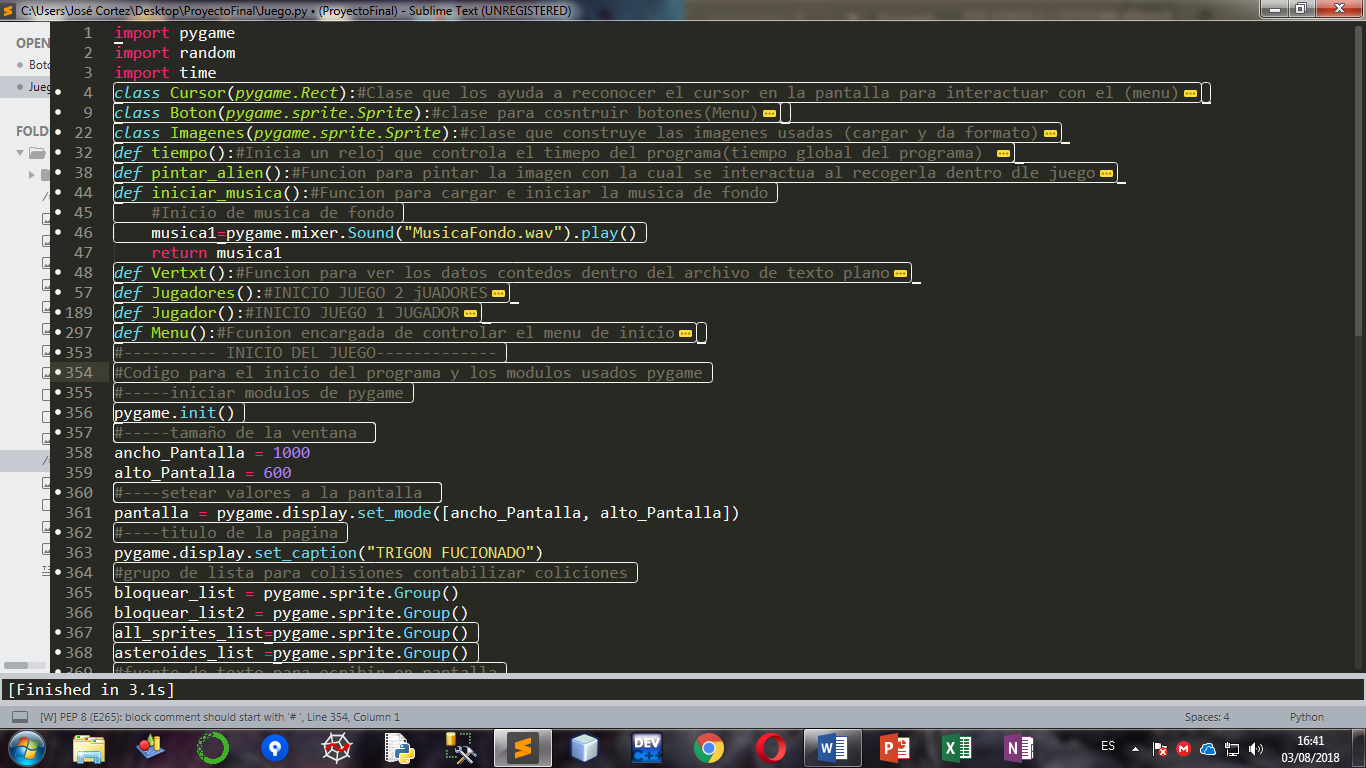


Imagen 12 Fusión iniciar música

Pygame es capaz de reconocer algunos tipos archivo tipo audio en este caso se usa una con formato wap.

### VerTXT

Esta función se usa para mostrar el puntaje de los jugadores, tiene un botón específico dentro del menú principal. El botón se puede ver en la Imagen 13 encerrada en un rectángulo color verde.

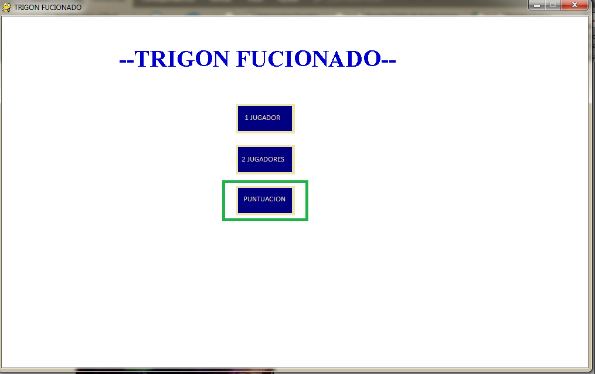


Imagen 13 pantalla menú señalando botón puntaje

El código de esta función se muestra en la imagen 14.

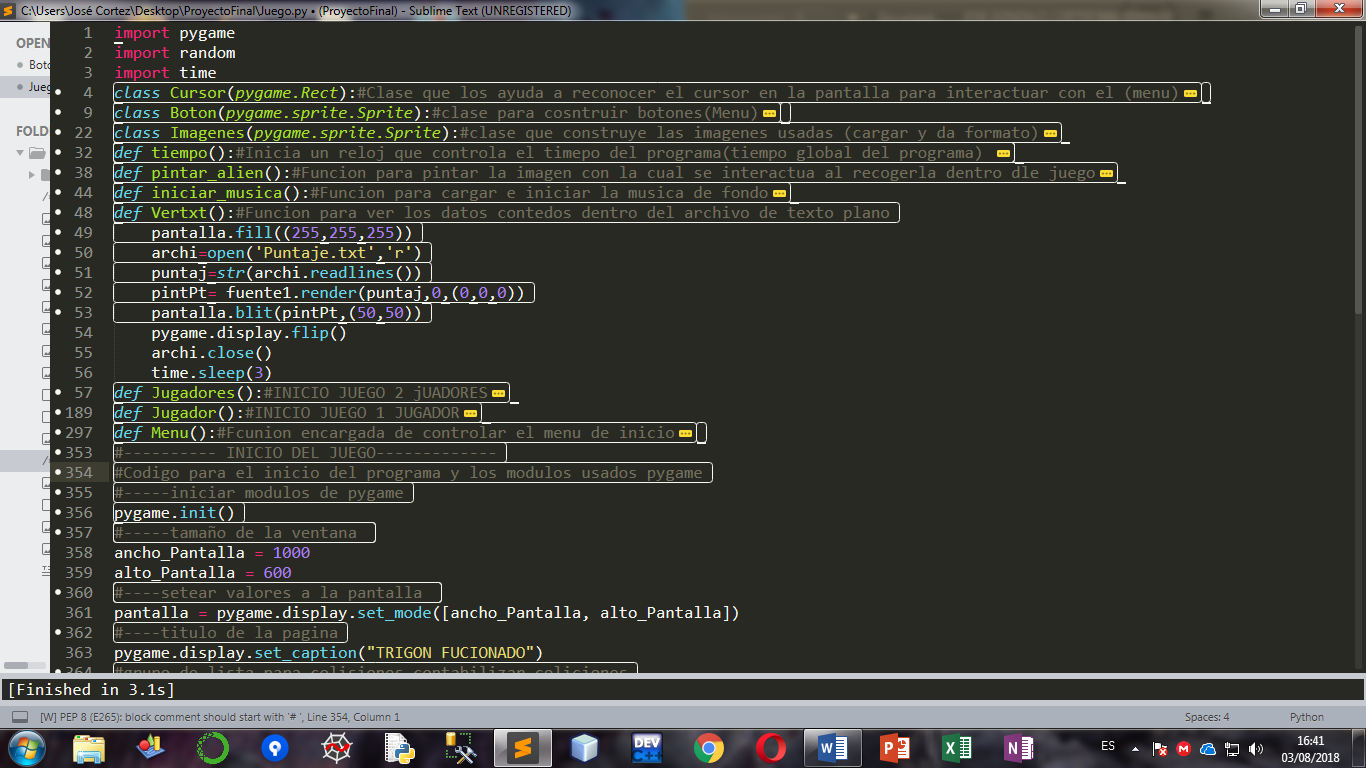


Imagen 14 Código Función verTxt.

El resultado de la función vertxt se muestra en la imagen 15.

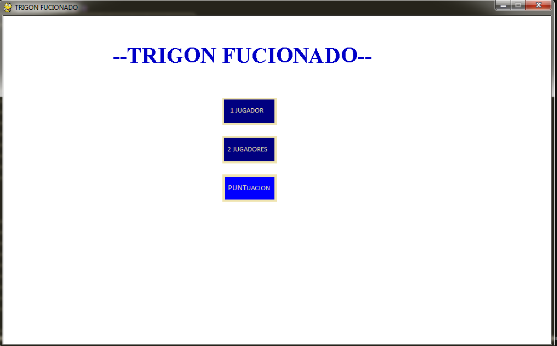


Imagen 15 Resultado del código de verxt

El resultado de los jugadores se mostrara por 3 segundos antes de actualizar la pantalla al menú por la función time.sleep(3) línea 56 imagen 14.

### Jugador

El bloque de codgo con mayor cantidad de instrucciones.

Para la implementacion de dos jugadores solo se repite varios bloques de codigo con la diferencia de Sprite cargados y la funcionalidad de botones distintos. Bloque de codigo jugador imagen 16.Pertenece a la carga de Sprites del jugador y los asteroides.

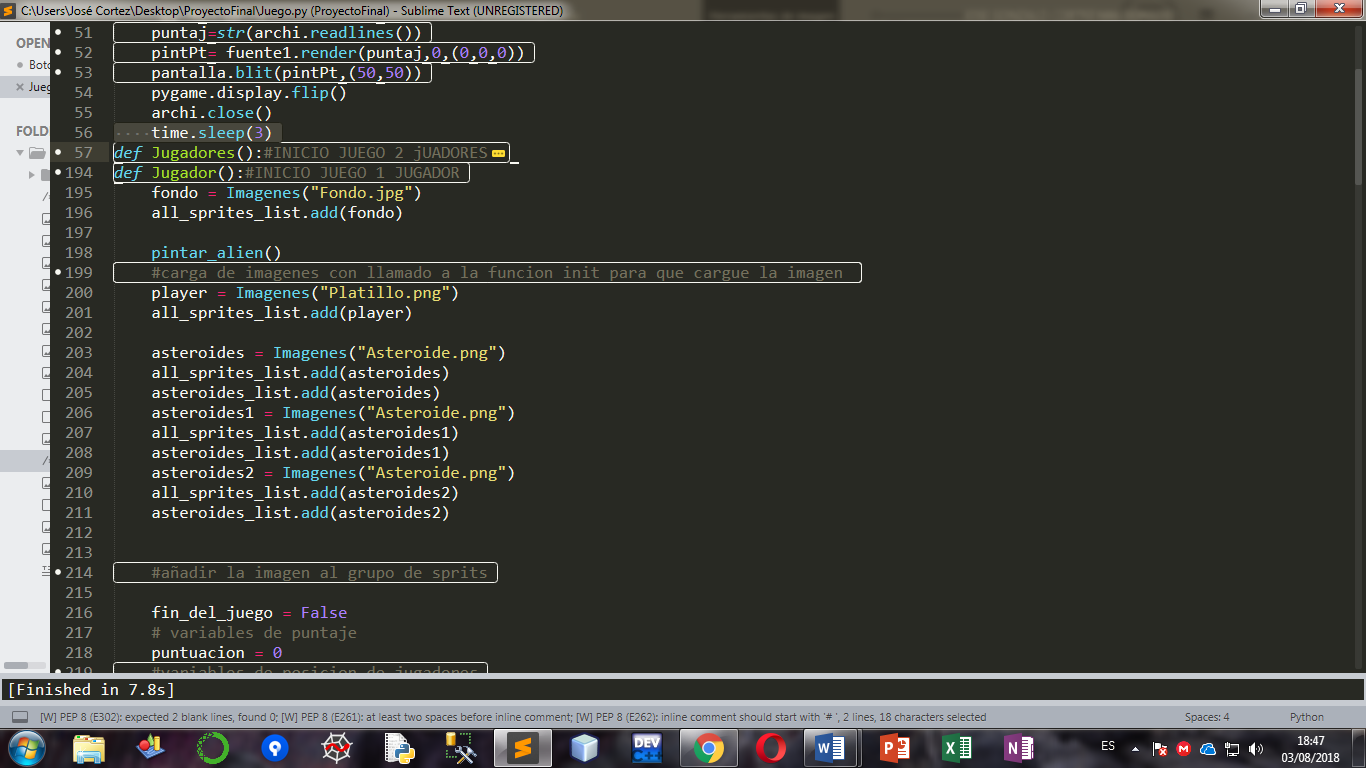


Imagen 16 Función Jugador Carga Sprite

En la imagen 17 se puede ver la variable de salida del bucle línea 211. El inicio de las variables usadas para movimiento del jugador y asteroides líneas 215 a 223. Variables para uso del reloj y muestra de tiempo en pantalla de jugador líneas 224 y 225. Por último, se ve la llamada a la función iniciar música para empezar la reproducción de la pista línea 226.

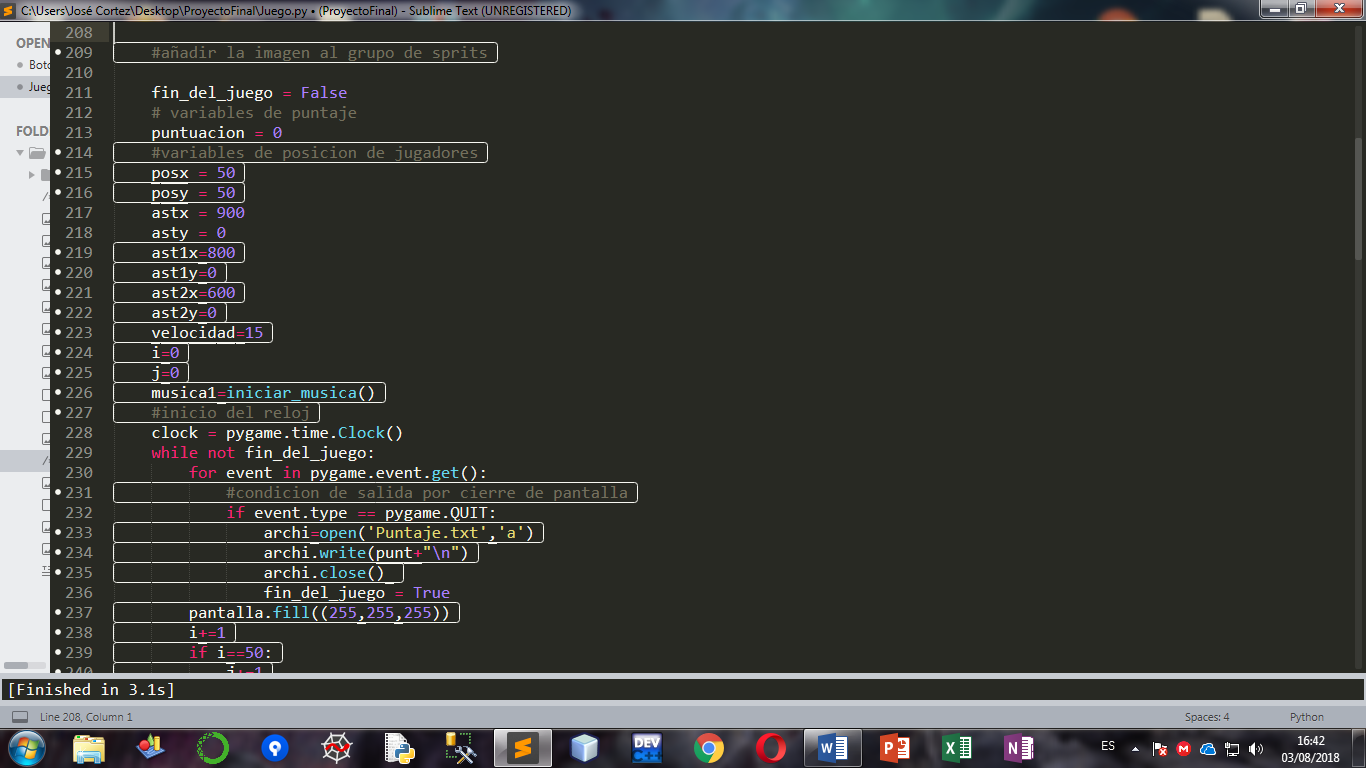


Imagen 17 Bloque 2 de la función Jugador

La imagen 18, muestra el inicio de clock para control de los fotogramas por segundo dentro del bucle principal del juego línea 227.

Bucle principal del juego línea 229. Línea 230 inicio del bucle para reconocimiento de eventos dentro del juego. Línea 232 condición de comparación de enveto con cierre de pantalla para salir del bucle del juego. La línea de 233 a 235 sirven para guardar el puntaje del jugador en el archivo de texto plano. Línea 237 pintar pantalla de fondo blanco. Línea 238 a 241 usada para contabilizar el tiempo en dentro del juego. Línea 243 a 248 variables de posiciones de asteroides.

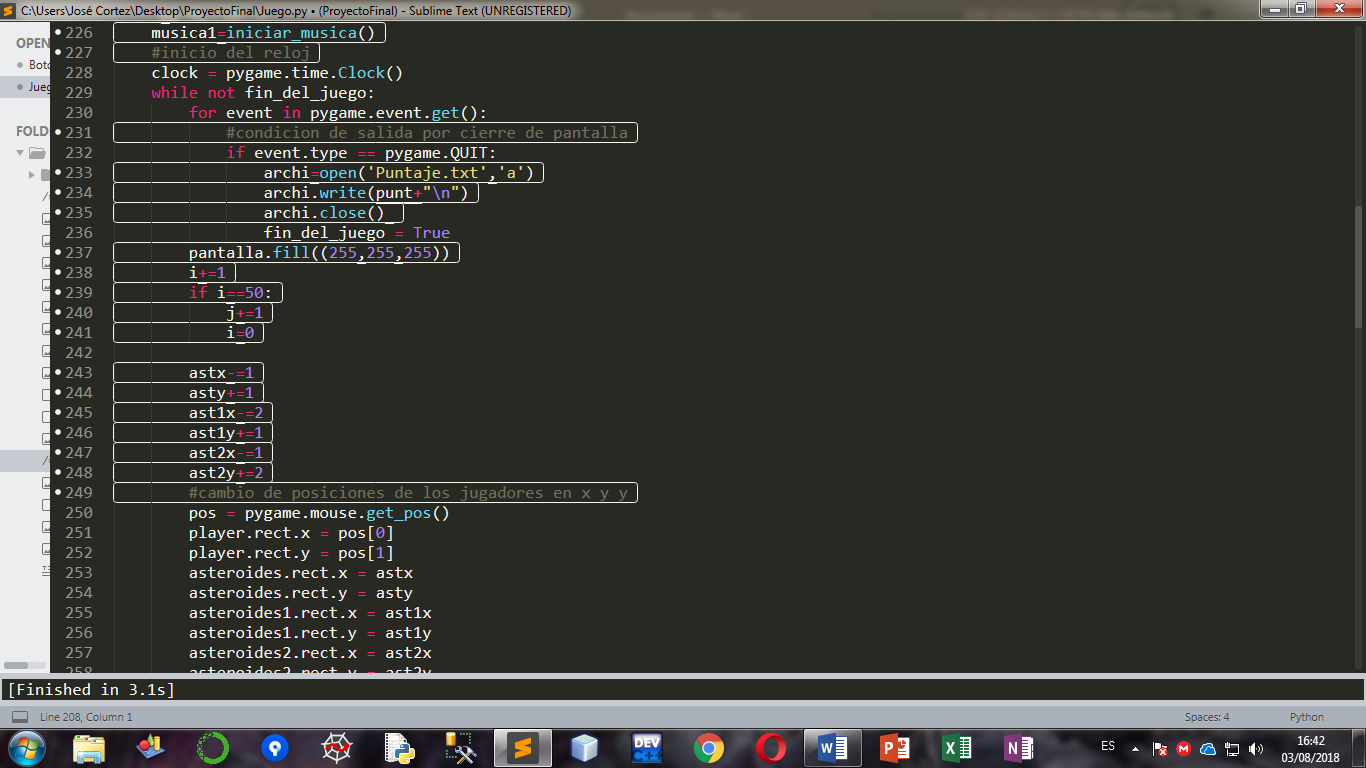


Imagen 18 Bloque de código 3

Líneas de código de 271 y 272 detección de evento de colisión con el marciano y con el asteroide respectivamente. Líneas 274 a 279 for que detectan una variable en este caso la de las colisiones en su interior esta el código que se ejecutara si se cumple la condición de colisión. Línea 281 pinta todos los Sprite que están dentro del grupo de Sprite. Líneas 283 a285 asignación a una variable reloj el valor de j contador reloj del juego 287 a 289 pintar puntaje del jugador en pantalla. Línea 291 muestra la cantidad de fotogramas por segundo(fps). Linea293 realiza una actualización de pantalla. Línea 294 parar música de fondo. Línea 295 Ingreso al menú por colisión. Línea 296 fin y salida de la pantalla por cierre de ventana. Imagen 19.

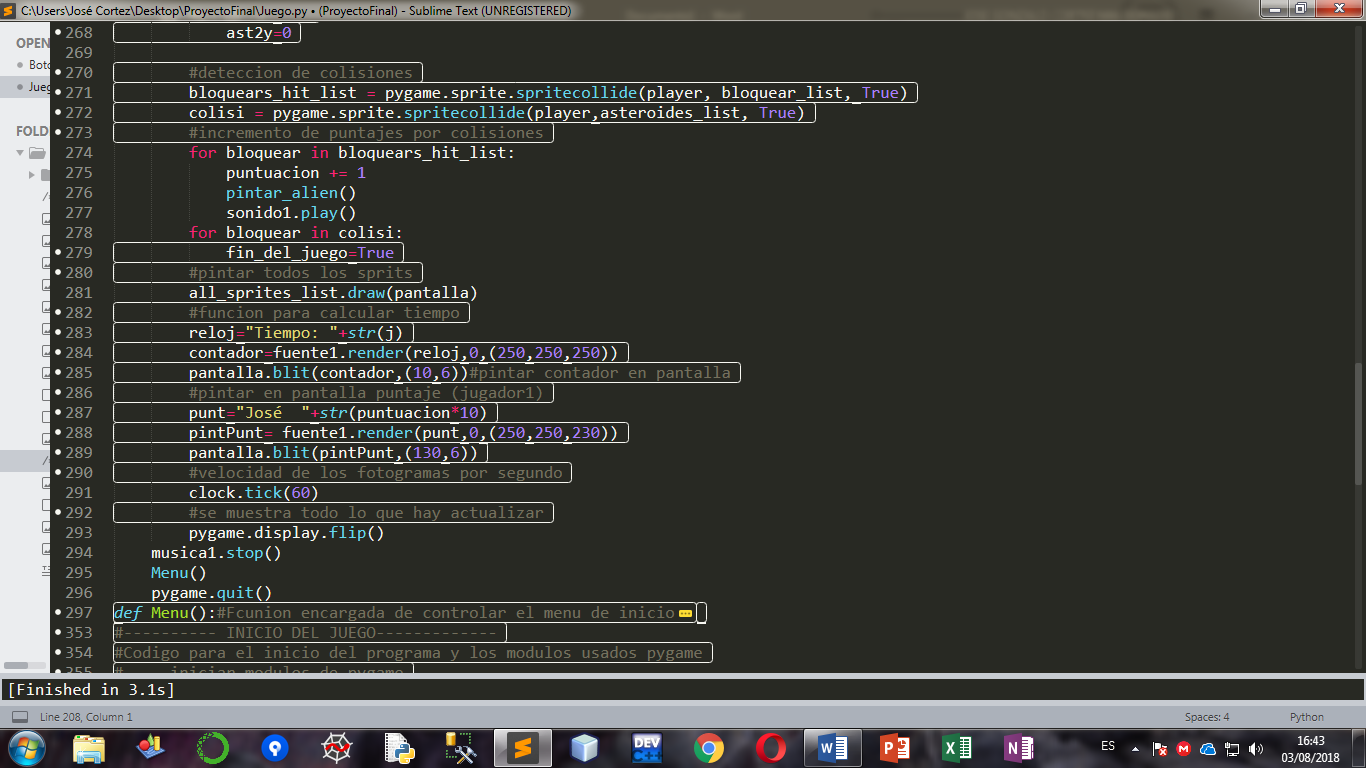


Imagen 19 Bloque de código 4

## **¿Cómo funciona el juego?**

Presentación principal del menú de inicio con 3 opciones 1 jugador, 2 jugadores y puntuación.

Funcionalidad de 1 jugador. Imagen 8.

Funcionalidad de 2 jugadores imagen 20.

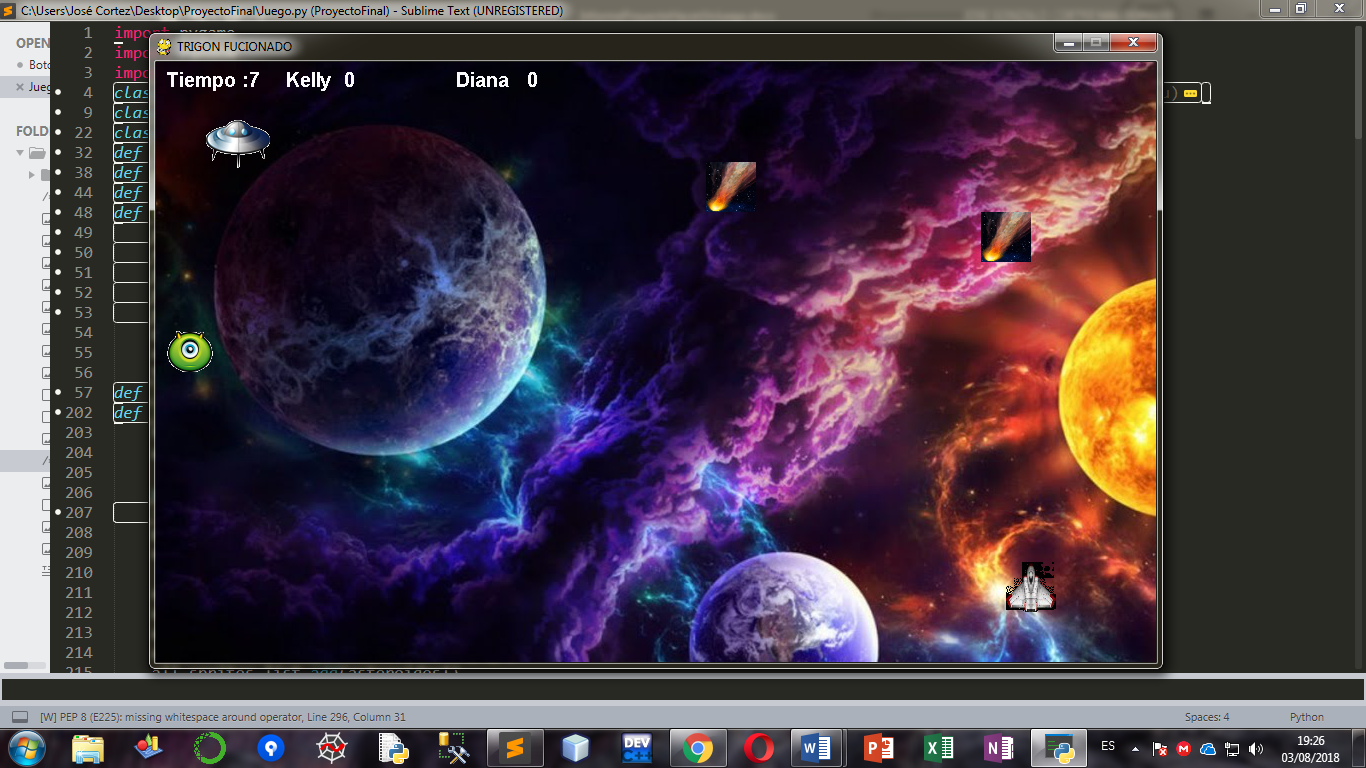


Imagen 20 2 Jugadores

# **Conclusión:**

El juego cumple con las instrucciones requeridas para su diseño y cada. Se debe realizar un análisis más exhaustivo en la función jugador para obtener una modularización óptima. El juego puede seguir siendo desarrollado con el tiempo añadiendo funcionalidades para una mejor experiencia de los jugadores.

[1], [2], [3], [4], [5]

**Bibliografía**:

|  |  |
| --- | --- |
| [1] | Pildoras Informaticas, «Pildoras Informaticas,» [En línea]. Available: https://www.youtube.com/user/pildorasinformaticas. [Último acceso: 30 05 2018]. |
| [2] | Código Facilito, «Código Facilito,» [En línea]. Available: https://www.youtube.com/user/codigofacilito. [Último acceso: 07 06 2018]. |
| [3] | PyDev, [En línea]. Available: http://www.pydev.org/manual\_101\_install.html. [Último acceso: 03 05 2018]. |
| [4] | Pygame, [En línea]. Available: https://www.pygame.org/docs/ref/mouse.html. [Último acceso: 21 06 2018]. |
| [5] | F. Jurado, J.J Castro, J. Albusac, D. Villa, D. Vallejo y G. Simmoros, Desarrollo de Video Juegos, vol. 4, Escuela Superior de Informatica, 2010. |