**MANUAL TÉCNICO: SABER GANAR**

CARLOS FERNANDO QUINTERO

JUAN PABLO GIRÓN RUIZ

TEMA: PROYECTO FINAL PROGRAMACIÓN

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

INGENIERÍA ELECTRÓNICA

SANTIAGO DE CALI

28/05/2018

**Introducción**

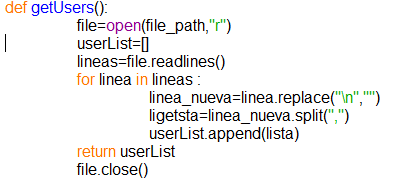
Saber ganar es un juego dirigido para estudiantes desde kínder hasta 11 con el fin de mostrar y afianzar sus estudios respecto a cada tema. De esta manera a continuación se explicará la parte lógica de todo el juego, desde la parte de python hasta la del JavaScript

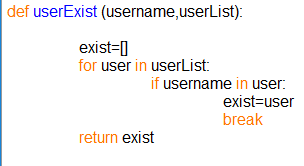
**Módulos Creados (Python)**

**-Ingreso**

Este módulo es el encargado de buscar y generar una lista de todos los estudiantes que han sido registrado, de tal modo que servirá para comparar durante el ingreso de un estudiante su Nickname y contraseña si son correctos. Después de haber dado esta explicación explicare sobre las funciones que están dentro de este modulo

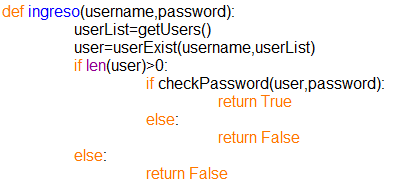
-getUsers():Esta función es la encargada de recorrer todo el archivo plano y generar una lista de todos los estudiantes que han sido inscritos.



-userExist():Verifica en la lista si el usuario esta inscrito de tal modo que crea una lista con los datos del usuario

-checkPassword():Verifica si la contraseña es correcta del usuario

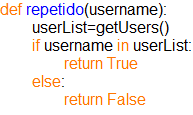


-ingreso():Esta función agrupa las demás funciones mencionadas en el donde verifica si el usuario ha sido encontrado para luego verificar la contraseña del usuario

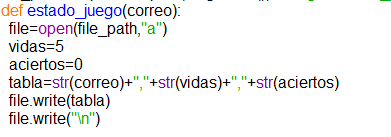
**-Exist\_Usuario**

Este módulo es el encardo de verificar durante el registro del usuario si existe, es decir, nos servirá para no generar otro usuario con el mismo nickname

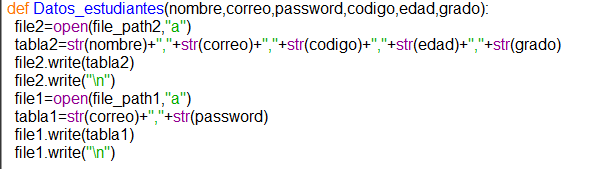
-repetido():Esta función utiliza la función getUsers para verificar si el usuario ya existe entre la lista.



**-Estado\_juego**

Este módulo sirve para cuando el jugador ya ha sido registrado se le asigne las vidas y los aciertos

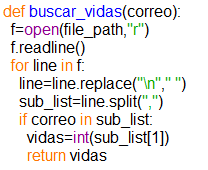
**-Datos\_estudiantes**

Este módulo sirve para escribir en el archivo plano los datos como lo son el nickname, código, edad, grado

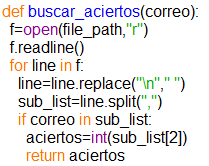
**-Buscar**

Este módulo sirve para buscar el número de vidas, aciertos, grado, nombre para que aparezcan en pantalla cuando un usuario este jugado

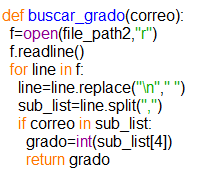
-bucar\_vidas:Busca las vidas en el archivo plano



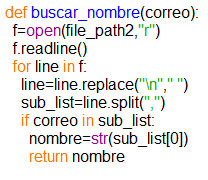
-buscar\_aciertos: Busca los aciertos en el archivo plano



-buscar\_grado:Busca el grado en el archivo plano



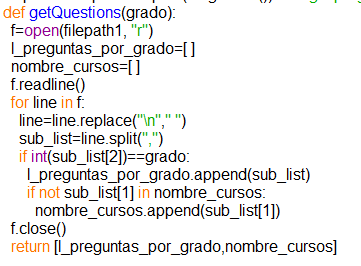
-buscar\_nombre: Busca el nombre del usuario en el archivo plano



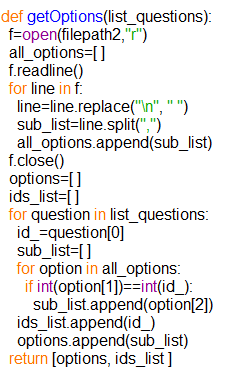
**-Diccionario**

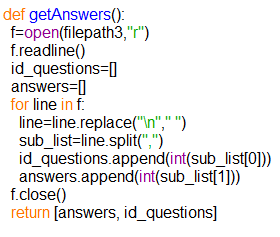
Este módulo genera un diccionario con las preguntas, las opciones y respuestas según el grado. A través de funciones que se encargaran de buscar las preguntas, las opciones y las respuestas. Así mismo contiene una función la cual se encargara de verificar si las preguntas ya se han respondido.

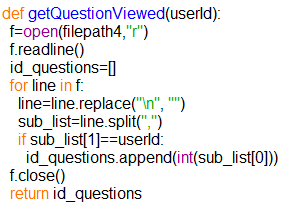
-getQuestions: Esta función recorrerá el archivo plano almacenando las preguntas según el grado, dentro de una lista



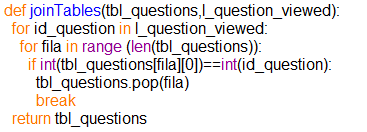
-getOptions: Esta función recorrerá el archivo plano almacenando las opciones según dentro de una lista, es decir que las opciones de una pregunta se almacenaran dentro de una lista, generando asi una lista de listas



-getAnswers: Esta función recorrerá el archivo plano almacenando las respuestas y el id de la pregunta según el grado, dentro de una lista

-getQuestionViewed:Sirve para almacenar en una lista las preguntas que ya visto el usuario

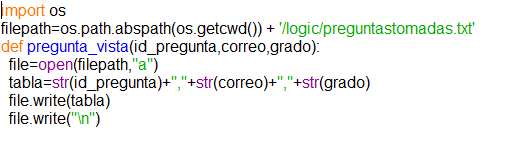
-joinTables: Esta función nos sirve para quitar las preguntas ya vistas por el usuario



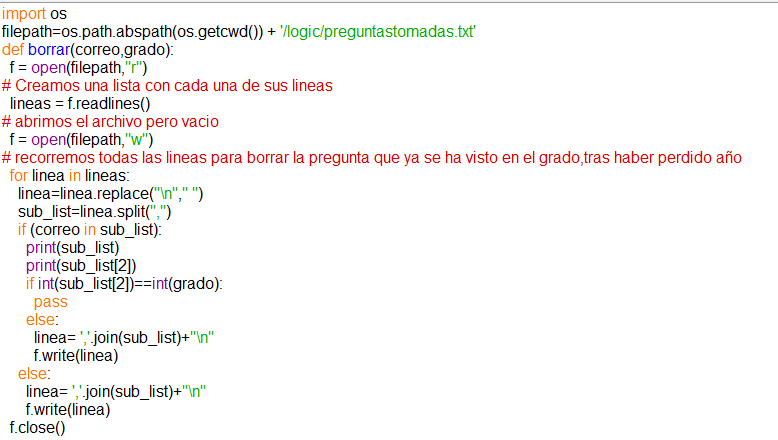
-getCourseDiccionary: Esta función genera un diccionario con las preguntas, las opciones y respuestas según el grado. A través del uso de las anteriores funciones, las cómo hemos mencionado se encargaran de buscar las preguntas, las opciones y las respuestas



**Pregunta\_vista**

Este módulo se encargará de almacenar cada vez que el usuario responda una pregunta, escriba el id de la pregunta con el correo y el grado en archivo plano

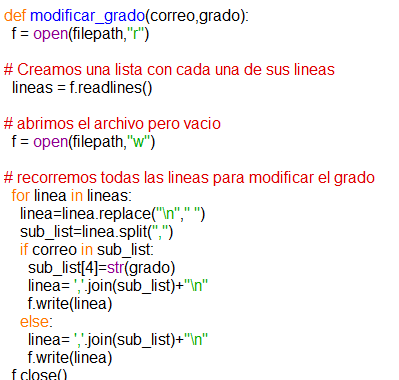
**Borrar**

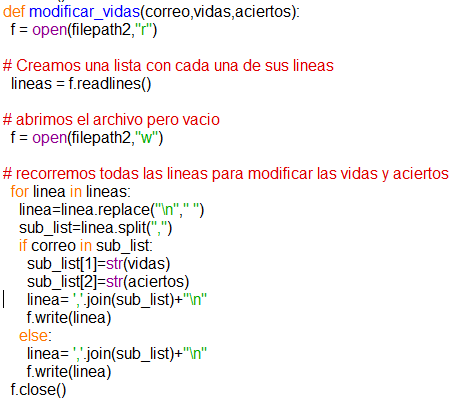
Este módulo se encargará de borrar todas la preguntas que haya respondido el usuario cuando pierda un grado, este módulo tendrá como parámetros el correo y el grado

**Modificar**

Este módulo tendrá dos funciones las cuales se encargarán una para modificar el grado, y otra para modificar las vidas y aciertos. La función que cumple este módulo es guardar los datos cuando el usuario haya decidido salir.

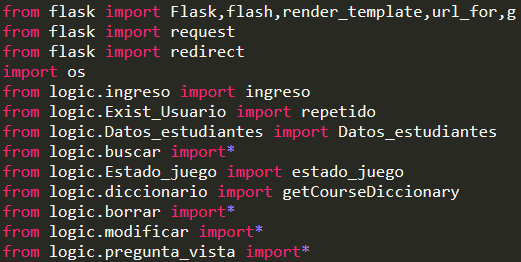
-modificar\_grado: Esta función servirá para guardar el grado cada vez que el usuario logre pasar de grado



-modificar\_vidas: Esta función servirá para guardar las vidas y acierto cada vez que el usuario logre pasar de grado

**Flask y Html**

Para la elaboración del juego primero es necesario importar los siguientes módulos, los cuales servirán para que el juego funciones correctamente. A continuación, explicaremos cada ruta de Flask mostrando el archivo HTML

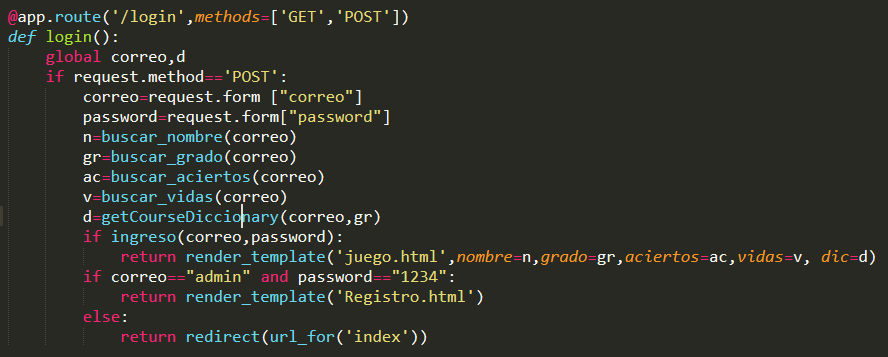


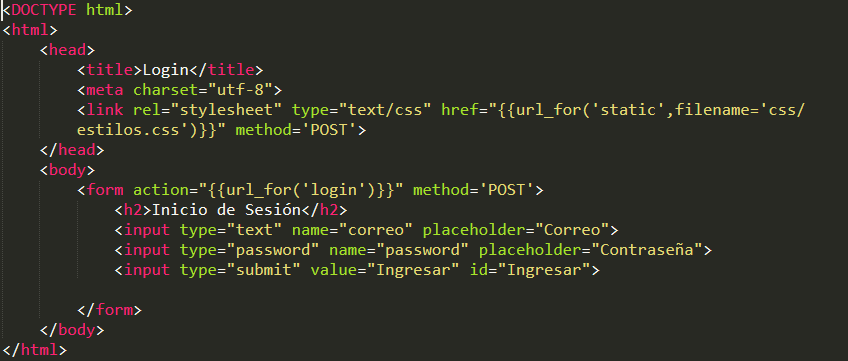
**Rutas**

**Login**

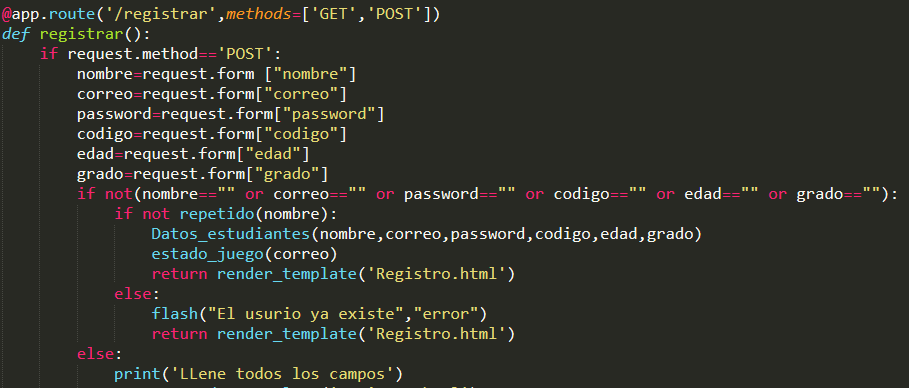
Esta ruta se encargará de retornar al juego en caso tal de que el usuario (Estudiante) ingrese su Nickname de manera correcta. A su vez cuando haya cargado la página, a través del módulo de buscar y del diccionario enviará al juego el nombre, el grado, los aciertos, las vidas, y el diccionario en formato json

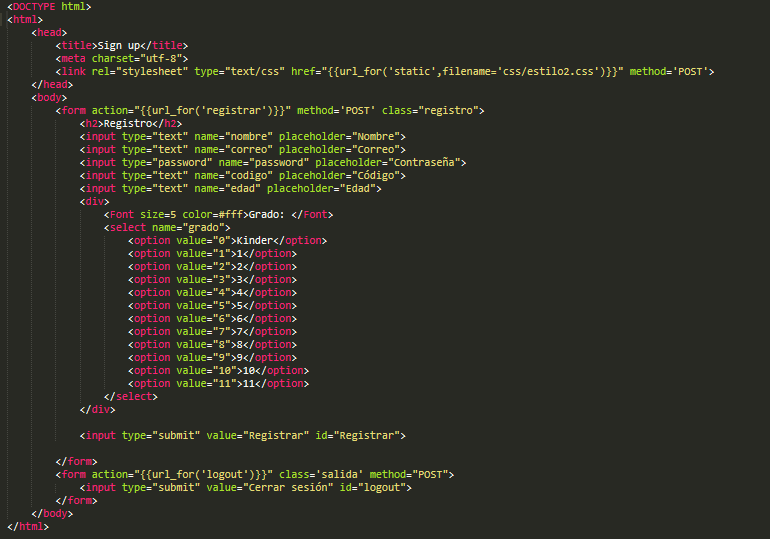
Nota: Si el usuario es el administrador se le redirigirá hacia la página de registrar estudiantes (Ver Registrar)





**Registrar**

Esta ruta se encargará de recibir los datos que se envíen desde la página registrar, de tal modo que los recibirá si todos los campos están llenos. Como podemos ver también se encargar de almacenar en un archivo plano los datos de los estudiantes, y en otro el estado del juego.



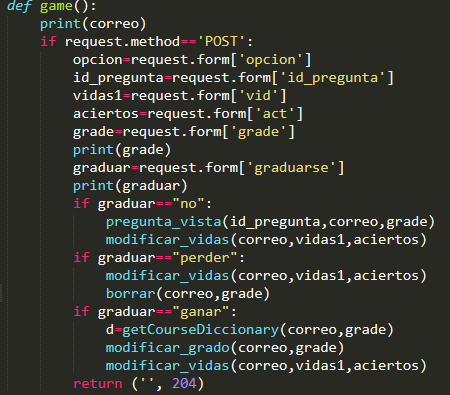
**Game**

La ruta game se encarga de recibir todas las variables como lo son las opciones, el id de la pregunta, las vidas, los aciertos, y el grado. Esta función servirá para guardar cada dato que se está modificando en el juego, es así como a través de una bandera, la cual se llama graduar, se encargará de cumplir varias funciones:

-graduar==”no”: Esta condición nos servirá para saber si el jugador está todavía en el juego respondiendo preguntas, de tal modo que ira almacenando el id de la pregunta en el archivo plano de preguntas tomadas, y a su vez ira modificando las vidas y los aciertos que el jugador tiene

-graduar==” perder”: Esta condición hace eliminar las preguntas que han sido contestadas según el grado, de tal modo que hace que el jugador comience de nuevo

-graduar==” ganar”: Esta condición hace que el jugador pase al siguiente grado

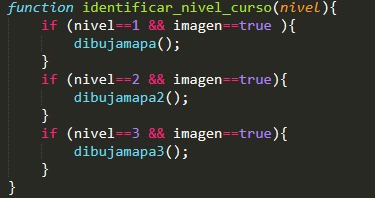


**Juego (Html)**

**¿Cómo se cargaban las imágenes?**

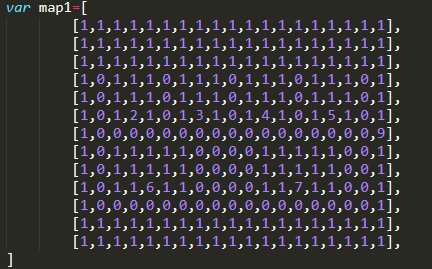
Las imágenes se cargaban a través de canvas, en el donde cumplían unas condiciones para así poder cargarse.

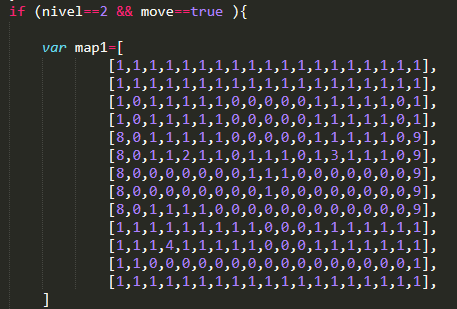


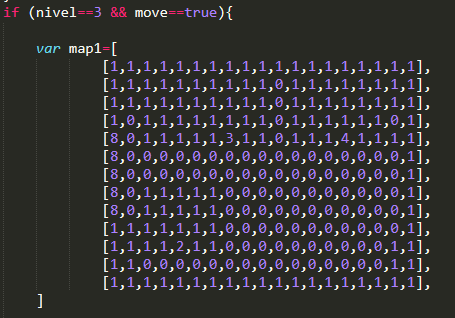


**¿Cómo fue construido los mapas?**

Los mapas fueron construidos a través de una matriz, en la cual se delimitaba Los espacios en el que el jugador se podía mover. De este modo si el jugador se encontraba un uno no podía moverse hacia ese lugar, pero si el jugador se encontraba un numero diferente al 0 este se podía mover hacia ese lugar

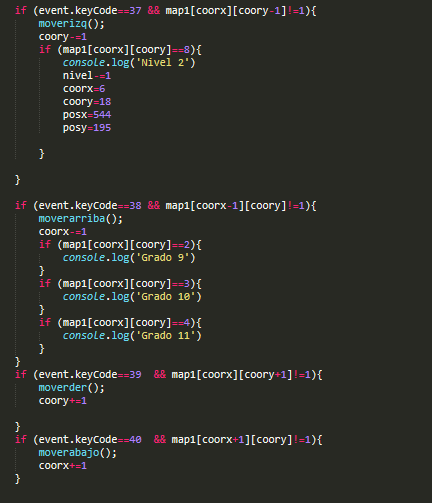
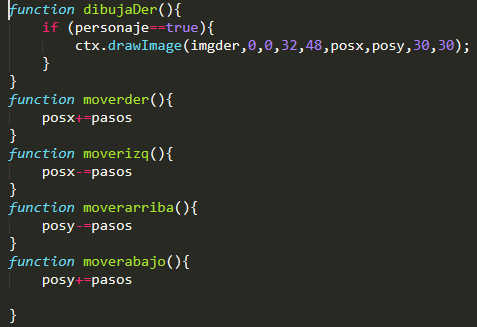
-Nivel 1:

-Nivel 2:

-Nivel 3:

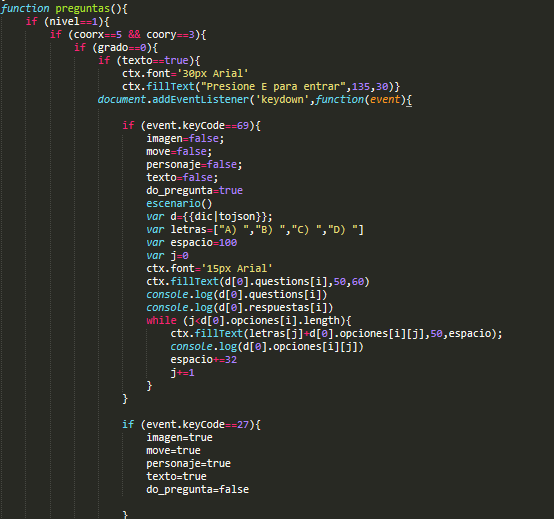
**¿Cómo se puede mover el personaje?**

El jugador se mueve con las flechas, de tal modo que la posición del personaje se vaya desplazando.

**¿Cómo el jugador entra a una casa?**

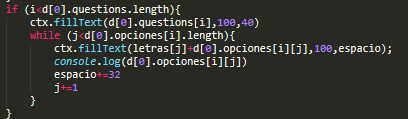
El jugador puede entrar a una casa según el grado en el que se encuentra, de modo que el jugador cuando este tocando “la puerta” se le activará un mensaje diciéndole que puede entrar a la casa, dándole al botón E:



**¿Cómo se el jugador responde las preguntas?**

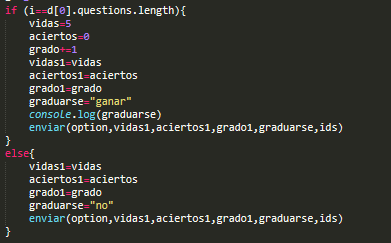
El jugador cuando haya entrado a la casa, inmediatamente le aparecerán las preguntas a través del diccionario que se le ha enviado. Así mismo, a través de un ciclo hará que aparezcan las preguntas, todo esto se ira activando cada vez que el jugador haya contestado las preguntas.

Las preguntas se responden con las teclas A, B,C,D

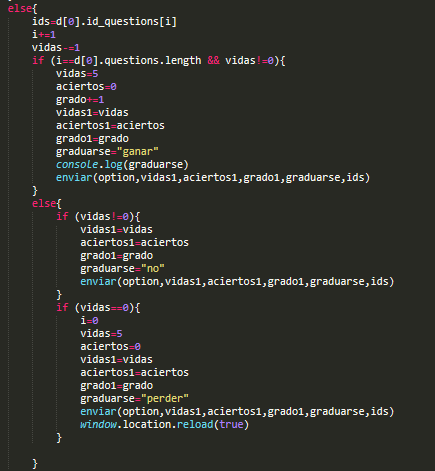


**¿Cómo gana?**

El jugador gana o pasa de curos cuando haya contestado todas sus preguntas, sin haber agotado sus 5 vidas



**¿Cómo pierde?**

El jugador pierde cuando ha agotado sus siete vidas, de tal modo que tiene que repetir nuevamente el curso

**Enlace proyecto:** <https://github.com/CarlosQ18/Proyecto>