

### Licenciatura en Sistemas

# Trabajos Práctico

Péguele al precio

### Introducción a la Programación

(2°Semestre, 2023)

La consigna del trabajo es realizar un juego funcional y optimo llamado “Péguele al precio”

Integrantes: Molina Uriel *molinauriel870@gmail.com*

Sosa Gabriel *gabrielsosaeest1@gmail.com*

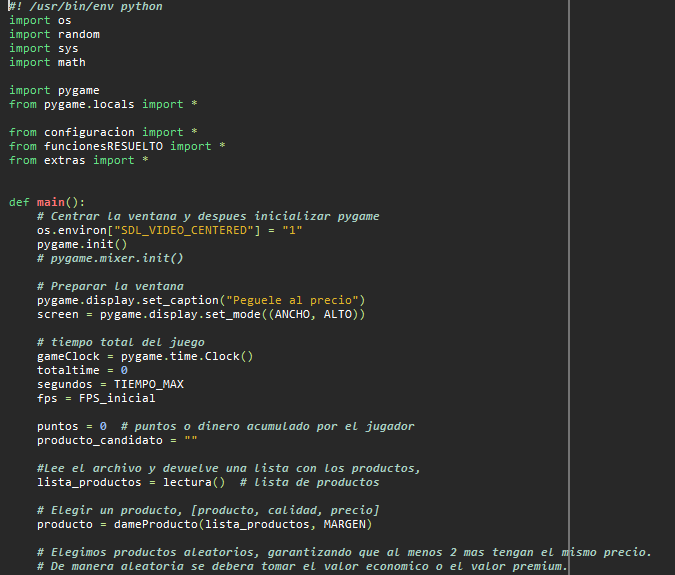
*Barrera Ezequiel* ezequielbarrera2000@gmail.com

**>Introducción<**

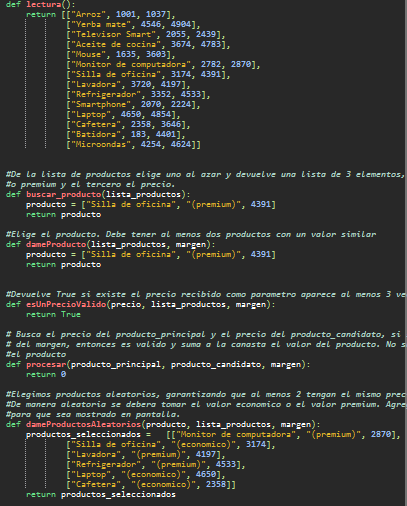
El trabajo practico de Introducción a la Programación, consta de un juego de adivinanza. En este juego, se posee una lista de productos con dos precios diferentes siendo estos el económico, precio menor, y el premium, precio mayor, y en base a esta lista se debe elegir un producto principal y dada una limitada lista de productos, sin mostrar su precio en ningún momento, se debe intentar adivinar el producto que este dentro de un margen especifico, en este caso un margen de 1000.

Por cada acierto se suman puntos y también es importante aclarar que el juego se desarrolla en solo un minuto, es decir, 60 segundos.

El código principal está ya armado, junto con los parámetros del propio juego (tamaño de pantalla, tiempo límite, margen, etc.). Y, además, se da un archivo denominado “funcionesVACIAS” el cual posee las funciones con las que se desarrolla el juego, pero sin ningún tipo de código, debido a que es el grupo de alumnos quien debe programarlas.



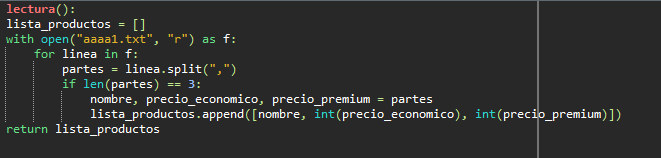
Porción del código principal



"FuncionesVacias"

**>Solución del trabajo<**

A la hora de dar una solución de las consignas principales, se tomo en cuenta dos factores principales: que funcione y que contenga el código necesario en función al problema planteado, ni más, ni menos.

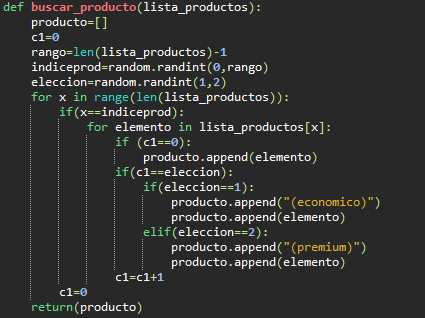
La primer problemática fue la función “lectura()” la cual retorna una lista con listas dentro, las cuales deben tener el nombre del producto, su precio económico y su precio premium. Para lograr esto se hizo uso de una nueva función la cual se llama “split()”, la cual tiene como función dividir una cadena o numero cada que se llega a un carácter el cual figura en su parámetro. 

En este caso, se usa para dividir la línea del archivo de texto. Se divide cada que lee el carácter “ , ”, y una vez se haya dividido en tres partes, se le asigna un nombre a cada parte (nombre, precio económico, precio premium). Por ultimo se toma la lista vacía que se creo al principio y se añade cada parte para así formar una lista, y esto se repite en cada línea del archivo.

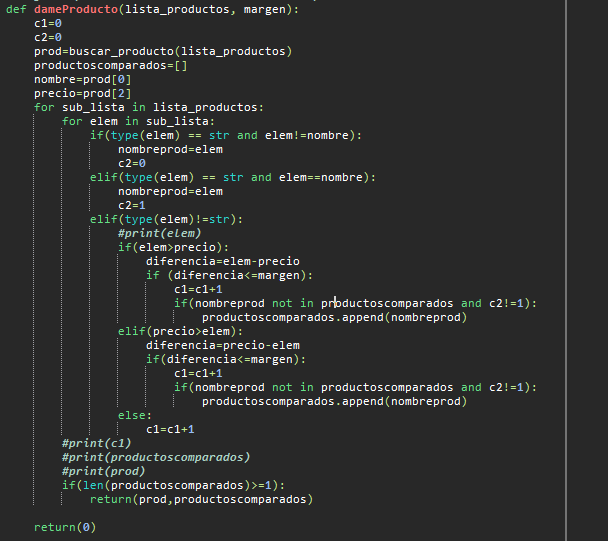
La siguiente función desarrollada fue “buscar\_producto(lista\_productos)”, la cual toma como parámetro la lista de productos devuelta por la función anteriormente mencionada (“lectura()”). La función lo que hace es devolver un producto al azar, ósea, sus tres elementos: nombre, si es económico o premium y por último su precio, el cual va a depender de si es económico o premium.

Lo que hace la función es navegar por la lista, va navegando con su “for x in range” por índice, no por cada elemento por separado, y con la variable “indiceprod” se elige un producto al azar y solo cuando “X” valga ese producto al azar, con otro “for”, va a leer los elementos dentro de ese índice, ósea, la lista con los elementos del producto.

El contador “c1” se usa para, primero, si vale 0 guardar el elemento en la lista “producto”, y al mismo tiempo, como una variable random donde si cae “1” es económico y si es “2” es premium de esa manera se asegura que la elección sea aleatoria. Por ultimo se aumenta el contador para que no lance error el programa y asi la función va a devolver una lista con: nombre del producto, calidad del producto, económico o premium, y su respectivo precio.



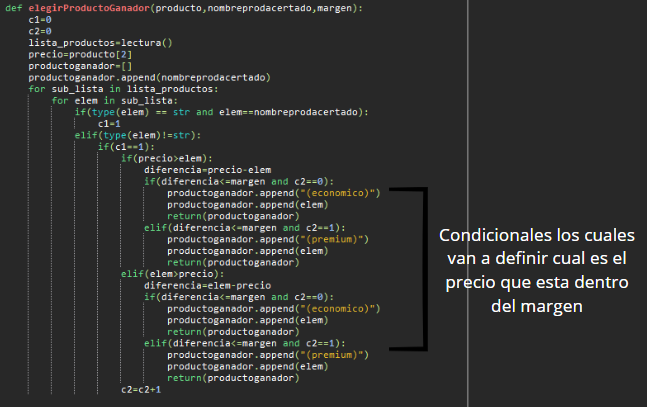
La función que sigue es “dameProducto”, la cual toma como parámetros la lista de productos de la función ya mencionada “lectura()” y el margen, el cual ya esta definido en la configuración del programa.

Para empezar, usa la función “buscar\_producto” para poder comparar ese producto con toda la lista de productos, para poder buscar dos productos diferentes con precio dentro del margen, el cual es 1000. Se definen dos variables nuevas, “nombre” con la posición del nombre del producto elegido para ser comparado y “precio” con la posición de su precio. Luego se abren dos “for”, uno para ir viendo índice por índice en la lista de productos y otro adentro de ese mismo, pero para ver elemento por elemento.

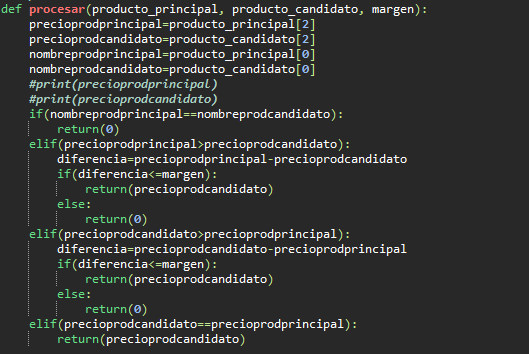
Los “if” sirven para comprobar si el elemento es "str", ósea es el nombre o su calidad, el nombre del producto va a tomar el valor del elemento, es decir, el nombre del producto al que se esta comparando. Luego, si el elemento no es un "str",ósea, en este caso es un int/número entero, se verifica si su precio esta dentro del margen. Una vez se haya comprobado todo, se guarda el nombre del producto en una lista con los productos comparados.

Por último, si la longitud de la lista de los productos comparados en mayor o igual a 1, se devuelve esa lista la cual contiene nombre del producto y sus precios, económico/premium, y otra variable para su futuro uso. Se exige que la longitud de la lista sea mayor o igual a 1 debido a que la consigna exige al menos un producto dentro del margen.

Otra función la cual va de la mano de la anterior es “elegirProductoGanador”. Esta función toma el precio de los productos que retornó la función anterior y comprueba cual de los dos precios es el acertado dentro del margen establecido, debido a que la función “dameProducto” devuelve el nombre y sus diferentes precios pero no cual precio esta dentro del margen establecido.



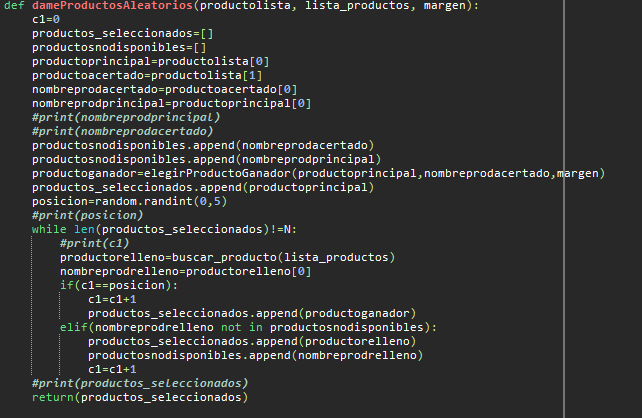
La próxima función presenta es "procesar" esta usa los parámetros producto\_principal, producto\_candidato y margen. Toma el nombre y los precios de los productos en la lista "producto\_principal" y "producto\_candidato" ubicados en la posición (0) y (2) de las respectivas listas ya mencionadas. Con esos datos pone las siguientes condiciones, la primera es si el nombre del producto principal es igual al del producto candidato retorna a (0); la siguiente condición son que si el precio principal es mayor que el precio candidato se resta el precio principal con el precio candidato y de ahí si la diferencia es menor o igual que el margen retorna el precio candidato, de caso contrario retorna a (0); la próxima condición es en este caso que si el precio candidato es mayor que el precio principal se resta el precio candidato con el precio principal y si la diferencia es menor o igual que el margen retorna el precio candidato de caso contrario retorna a (0) y la última condición pide que si el precio candidato es igual al precio principal retorne el precio candidato. En resumidas cuentas la función en el juego compara tu producto elegido, producto candidato, en el juego con el producto con el que se compara, producto principal, y decide si cumple con las condiciones para sumarte puntos o no.



Por último, la función “dameProductosAleatorios” toma tres parámetros, productolista (viene de “dameProducto”), la lista de productos y el margen. Se abren dos listas, productos\_seleccionados y productosnodisponibles, estas sirven para saber cuales productos deben mostrarse en pantalla para que no haya repetidos, debido a que una vez aparezcan se vuelven “no disponibles”. Tambien existe la variable posición, la cual define en que posición estará el producto correcto (0-5).

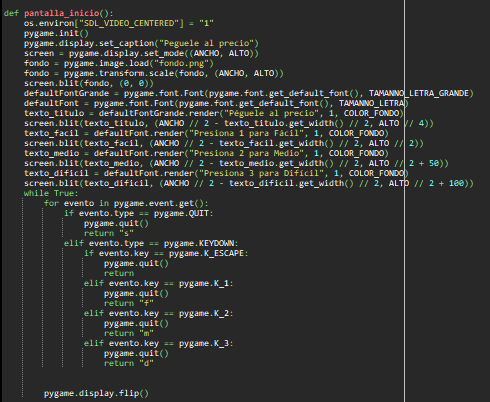
El código muestra un código “while” el cual se cierra una vez se llegue a la cantidad de productos en la lista “productos\_seleccionados” sea la misma que la variable N, que en "configuración.py" esta definida como N=5, luego empieza un "if" donde si el contador cae en la posición aleatoria, se añade a la lista el producto ganador y sino se suma 1 al contador.

Luego hay un "elif" para añadir productos de relleno y al final se aumenta el contador devuelta. Por último, retorna la lista de productos\_seleccionados, los cuales son los que se van a ver en pantalla.



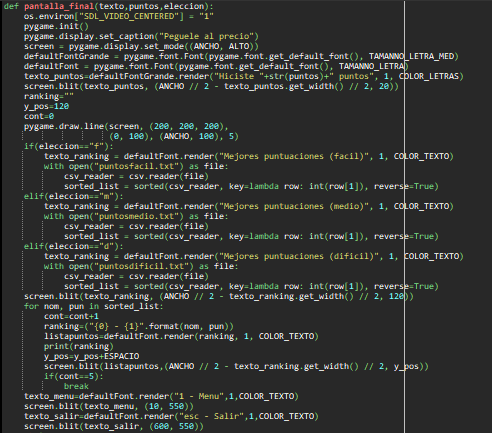
**>Extras<**

Ahora con el juego siendo completamente operativo optamos por agregarle al mismo diversas funciones extras. De las cuales se puede destacar el fondo de pantalla personalizado, efectos de sonidos, pantalla de inicio y fin del juego, seleccion de dificultad y mejores puntuaciones.

La primera función es pantalla\_inicio() la cual funciona como menu principal, esta funcion es llamada desde main y retorna tres posibles valores dependiendo de la eleccion del usuario, si la eleccion es 1, devuelve “f”, si la eleccion es 2, devuelve “m” y si la eleccion es 3, devuelve “f”, esto sirve para seleccionar el nivel de dificultad el cual se desea para jugar.

Dependiendo el nivel de dificultad elegido, se modifican algunas funciones y variables anteriores para modificar el juego, si se escoge la dificultad facil, el margen se queda en 1000, en medio, el margen es 800 y en dificil, el margen es 500. Ahora procesar() devuelve dos variables, una son los puntos, como anteriormente hacia, y ahora los segundos que se le suman o restan al contador si la respuesta es correcta o incorrecta, en dificultad facil, el contador suma 1 segundo y no resta ninguno, en dificultad medio, el contador no suma ni resta segundos, y en dificultad dificil, el contador suma 5 segundos y resta 5 segundos. Además, la cantidad total de productos tambien es afectada por las dificultades, en facil, son 4 productos, en medio son 6 productos y en dificil son 7 productos.

Otra funcion es ingresar\_nombre(puntos,eleccion). Esta funcion se encarga de preguntarle el nombre al usuario que acaba de jugar para si guardarlo en un archivo de texto junto a su puntuacion, por ello, toma los parametros puntos y eleccion para obtener que .txt se debe escribir. El usuario puede escribir su nombre y cuando presiona enter, lo lleva a la pantalla final junto a las mejores puntuaciones.

La ultima función extra es pantalla\_final(texto,puntos,eleccion), esta pantalla es invocada cuando se aprieta la tecla enter en ingresar\_nombre(), consta de tres parametros: texto, puntos y eleccion. El primer parametro es el nombre del usuario, el segundo los puntos que hizo y el tercero la eleccion para poder leer el archivo de texto correcto. Se hace uso de la librería csv aprovechando el formato de los archivos de texto “Nombre,Puntos” esta librería se usa para separar los parametros con comas, además, la funcion sorted los ordena de menor a mayor, por eso al seleccionar reverse=True, los ordena de menor a mayor. Finalmente con un for, se recorre la lista ordenada de mayor a menor y se muestran las 5 mejores puntuaciones. Además, se puede volver al menu apretando la tecla 1 y salir del programa apretando la tecla esc.

Además de eso, cabe destacar que en la funcion dameProductosAleatorios se controla que no hayan productos repetidos para cumplir con el requisito extra de que no hayan repetidos.

**>Conclusion<**

Para concluir, hay que aclarar que la realizacion del trabajo fue dividida entre todos los integrantes donde cada uno del grupo pudo cooperar de una forma u otra, logrando hacer el trabajo que se ha entregado. Salvo en algunos casos debido a ciertos problemas de tiempo/disponibilidad, el trabajo se logró realizar exitosamente y en el tiempo pre-establecido en los requisitos de entrega. Y hemos logrado aparte incluir varias funciones extras, esto último debido a una exhaustiva búsqueda en distintas páginas y videotutoriales en Internet que nos permitieron entender mejor la biblioteca, PyGame, que estábamos manejando y su amplia gama de posibilidades.

**>Enunciado<**

Un jugador tiene 60 segundos para ir adivinando productos que tienen precios similares y así ir acumulando dinero. El juego actualmente consta de un archivo con el programa principal, otro de configuración y uno de extras. Estos se encargan de capturar la entrada del teclado, llevar la cuenta del tiempo, así como también de dibujar en la pantalla. También cuenta con un archivo, funcionesVACIAS, que debe completar. Se brinda como ejemplo un archivo de texto con un listado con productos y precios. Recomendamos utilizar otro archivo con pocos productos para realizar los test. Para la mayoría de estas tareas, el programa hace uso de una biblioteca de código llamada PyGame. Una biblioteca de código es un conjunto de subprogramas utilizados para desarrollar software. En particular PyGame es una biblioteca especialmente diseñada para el desarrollo de juegos interactivos en Python.

**Lo que falta implementar**

Lo primero que se debe hacer es leer el archivo. Luego deben hacer funcionar el juego, y para ello se deben completar las funciones obligatorias, que son las funciones indicadas en el archivo funcionesVACIAS.py. Son las que aún faltan desarrollar y que son llamadas desde el programa principal

**Consignas**

Implementar las funciones requeridas para el correcto funcionamiento del juego. Pensar e implementar funciones auxiliares que resuelvan tareas intermedias, de forma tal que el código sea más claro, sencillo, ordenado, legible y fácil de corregir. Las funciones que reciben listas como parámetros deberán también chequear que dichas listas permanezcan en el estado correcto luego de utilizada la función. Sugerimos fuertemente probar y corregir las funciones más sencillas antes de encarar las funciones más complicadas.

Requisitos de Aprobacion

El presente trabajo debe realizarse en grupo, consultar la cantidad máxima a sus docentes. Para aprobar el trabajo, se deberá cumplir los siguientes items:

● El juego debe funcionar correctamente.

● El código debe ser claro. Es decir, las variables y funciones deben tener nombres que hagan fácil entender el código a quien lo lea, y debe haber comentarios que contribuyan al fácil entendimiento de cada porción de código. Además, deben hacer funciones siempre que consideren necesario, y se evaluará el buen uso de las mismas.

● El código debe ser coherente. No debe haber variables definidas que luego no se usen, ni funciones que tomen parámetros que no necesitan, ni ciclos innecesarios, etc.

● Los casos de prueba deben ser completos y debe ser posible correrlos de nuevo por cualquiera.

Nota: El correcto funcionamiento del juego no es suficiente para la aprobación del trabajo, son necesarios todos los items mencionados arriba.