

**Universidad Nacional de General Sarmiento**

|  |
| --- |
| **Trabajo Practico de Campo** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Alumna:**

**Melisa Rossi**

**Micaela Benitez de Olivera**

**Alexis Berdud**

**Esequiel Puita**

**Docentes:**

**José Luis Mieres**

**Sergio Santa Cruz**

**Miguel Rodríguez**

**Introducción a la Programación**

**Año 2023**

Contenido

[1. Objetivo 2](#_Toc151312081)

[2. Situación Problemática 2](#_Toc151312082)

[3. Extras 5](#_Toc151312084)

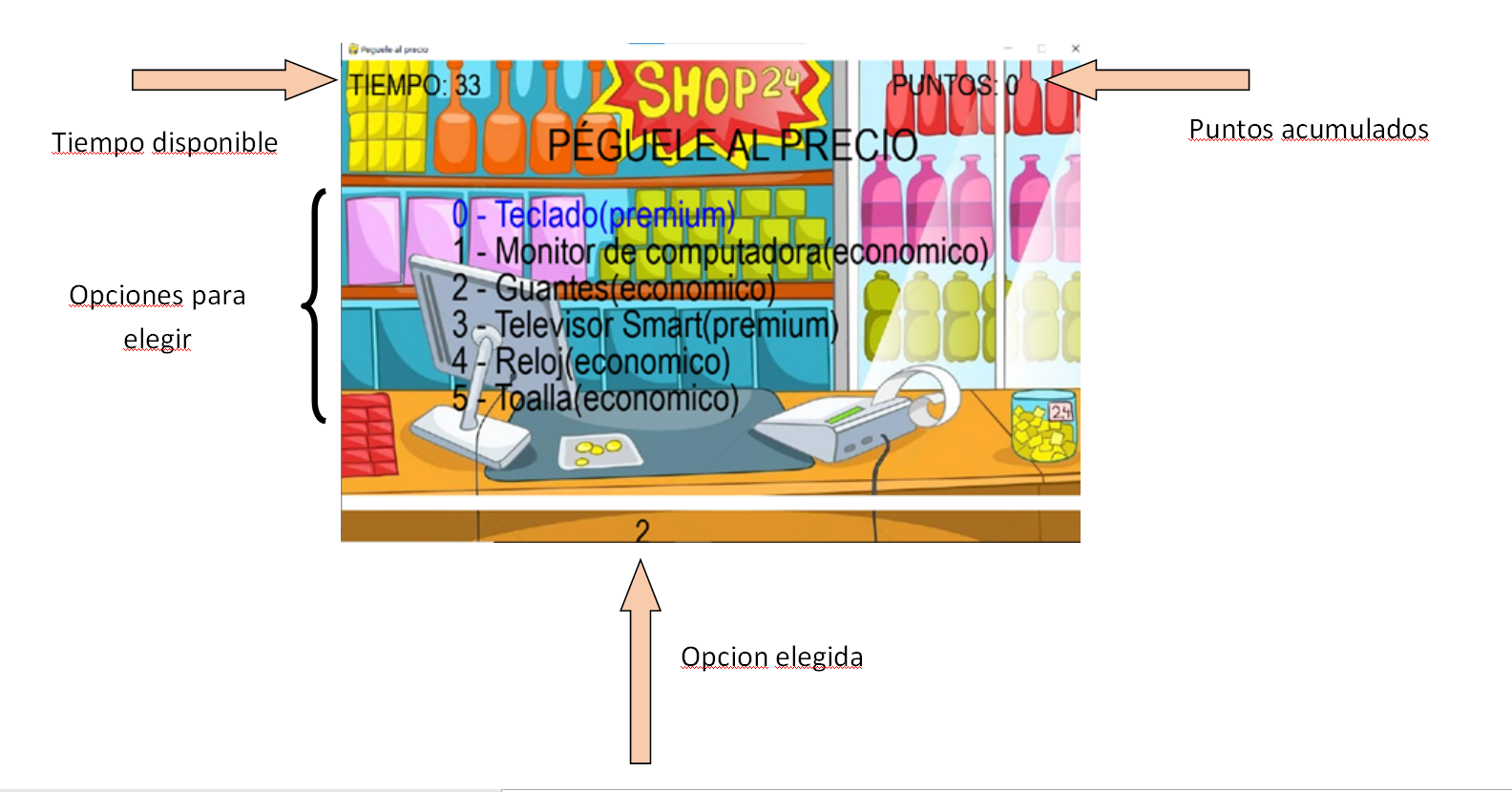
# Objetivo

El objetivo del presente trabajo practico consiste en terminar el desarrollo de un juego en pyscripter, para ello vamos a desarrollar las funciones propuestos por los docentes y otras que consideramos necesarios.

# Presentación del Juego

El juego se llama “PEGUELE AL PRECIO” y consiste en adivinar un producto similar en el precio al producto principal (artículos en color AZUL) en caso de que el usuario adivine este obtendrá puntos y lo tendrá que hacer dentro de un plazo de 60 segundos, la ventana principal ofrecerá 6 artículos totales, de los cuales tendrá 5 opciones más una 6ta opción para hacer la comparación;

A continuación, describimos la principal ventana, donde se detalla las partes a tener en cuenta durante el juego



# Situación Problemática

Para el desarrollo del trabajo, separamos en tres partes, la primera consiste el manejo de datos, la segunda parte en el desarrollo de las funciones y una tercera parte de las funciones extras

Funciones detrás del juego

Tenemos 4 módulos dentro de pyscripter, “CONFIGURACION”, “EXTRAS”, “FUNCIONESVACIAS” y “PRINCIPAL”,

Dentro del módulo “FUNCIONESVACIAS.PY” presentamos las siguientes funciones:

Funciones que nos permite

Una de las primeras funciones propuestas es “LECTURA (PRODUCTOS)” se encarga de hacer una lectura de los artículos del archivo TXT, para que la función sea efectiva es necesario la creación de otras funciones como “SACAULTIMOCARACTER (LISTA)” y “SEPARARPORCOMAS (CADENA)” ambos se encargan de hacer un manejo de la base de txt para generar una lista base de la cual futuras funciones puedan tomar el resultado.

En primer lugar, creamos esta función para eliminar el caracter “\n” para luego poder trabajar más cómodos y poder manipular el archivo con listas (que es lo que pide la consigna)

A screen shot of a computer code

Description automatically generated

Ahora lo que se nos ocurrió fue recorrer esas cadenas que nos dejó la función sacaUltimoCaracter(lista) a través de la función lectura(productos) e ir pasándoselas a separarPorComas(cadena). Esta función recibe una cadena, por ejemplo ‘yerba mate, 1730,1890’. Entonces, cada vez que vea una coma la voy a ir concatenando a una cadena vacía y cuando encuentre una coma, voy a appendear esa cadena que se formó a una nueva lista vacía, reinicio la cadena vacía y vuelvo a recorrer otra vez hasta encontrar una coma. Luego, la función separarPorComas (cadena) retorna una lista de la siguiente forma: [‘yerba mate’,’1730’,’1890’]

A screen shot of a computer

Description automatically generated

“BUSCAR\_PRODUCTO (LISTA\_PRODUCTOS)” se encarga de tomar un artículo de manera aleatoria, la cadena se conforma con el nombre del artículo, seguido de la calidad del producto y finalmente el precio (en función de la calidad del producto). Para elegir este elemento utilizamos una función ya implementada en Python, llamada ‘random.choice (nombre de la variable)’ que lo que hace es devolver un valor aleatorio extraído desde la variable que la pasamos como parámetro. Esta información la sacamos desde el siguiente enlace: <https://interactivechaos.com/es/python/function/randomchoice>

A black screen with white text

Description automatically generated

Para poder realizar las restas y que el valor no quede en negativo al restar el primer número con el segundo, creamos una función que se llama valorAbsoluto(num) que lo que hace es dado un número, verificar si es negativo. En caso de ser negativo, se lo multiplicara por (–1) para que se vuelva positivo, en caso contrario dejo el numero tal cual esta y devuelvo el número absoluto.

A black screen with white text

Description automatically generated

La siguiente función también es creada por nosotros. Esta función similar (lista, margen) lo que hace es devolver un valor booleano, True o False. Se fija en los precios de las dos listas que le pasamos y hace la resta. Una vez hecha la resta, llamo a la función absoluto y le paso el número de la resta para que me devuelva su módulo, una vez que tengo el módulo me pregunto si es menor al margen (1000). Si es menor, retorno True, en caso contrario False.

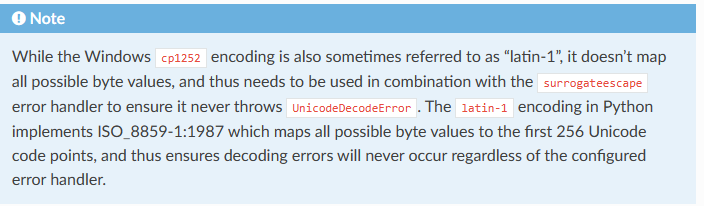
A screen shot of a computer code

Description automatically generated  
En el desarrollo de la siguiente función (dameProducto (lista\_productos, margen)) tuvimos varios inconvenientes cuando llamábamos a similar (lista, margen). Lo que sucedía es que si me daba False, nunca encontraba dos productos similares y el programa tiraba errores. El profesor José nos habló sobre ‘while else’ el cual funciona casi como un if else, pero en vez de if use while, para que mientras de False la función similar, entre en un bucle infinito hasta que se vuelva True. Una vez que se hace True, retorno el producto aleatorio de los dos. Adjunto el enlace del video donde explica el uso de ‘while-else’: [https://www.youtube.com/watch?v=omlz4KYpCE8Python desde Cero: ¿Cómo usar el While-Else? - YouTube](https://www.youtube.com/watch?v=omlz4KYpCE8Python%20desde%20Cero:%20¿Cómo%20usar%20el%20While-Else?%20-%20YouTube) . Por otra parte, respecto al código de esta función (es muy repetitiva, ósea se repite muchas veces el mismo código en varias partes de la función). El profesor Sergio nos recomendó eliminar la parte de if parecido == True y meter un return productoSimilar al final de todo, a la altura del while. Lo probe y no me funciono, también intente poder cambiarlo para que no quede así pero me tiraba errores, por lo que decidí dejarlo así, porque así funciona correctamente.

A computer screen shot of a code

Description automatically generated

Otro de los problemas con lo que nos encontramos fue en como leer el archivo txt(encoding) ya que cuando el archivo tenía una letra ‘ñ’ o alguna letra con tilde en la pantalla se presentaba con caracteres especiales. Después de buscar en varias páginas y probar diferentes tipos de codificación, lo pudimos solucionar. La información se obtuvo desde el siguiente enlace: [https://python-notes.curiousefficiency.org/en/latest/python3/text\_file\_processing.html#files-in-an-ascii-compatible-encoding-best-effort-is-acceptablean (curiousefficiency.org)](https://python-notes.curiousefficiency.org/en/latest/python3/text_file_processing.html#files-in-an-ascii-compatible-encoding-best-effort-is-acceptable)



# Extras

Como extras que se le agregan al juego son los sonidos y una imagen como fondo. Para poder implementar el sonido al juego nos ayudamos desde un video tutorial en YouTube con el siguiente enlace: [https://www.youtube.com/watch?v=Ib5TQaWwznYbe](https://www.youtube.com/watch?v=Ib5TQaWwznY) . No nos presentó muchas dificultades ya que en el video fue muy claro. Además, pudimos poner sonidos cuando el usuario acierta y de cuando no acierta. Desde el siguiente enlace: <https://www.youtube.com/watch?v=J473At9fuSk> pudimos poner una imagen como fondo de pantalla para el juego. Lo mismo que en el sonido, no se nos presentó mayor dificultad poder incorporarlo ya que el video fue muy claro. El único inconveniente fue poder encontrar una imagen que no se haga contraste con las letras para que éstas se puedan visualizar de manera correcta y además tuvimos que modificar el ancho y el alto de la ventana ya preestablecida por los profesores, adaptándola al tamaño de la imagen para que ésta no esté cortada.

También como extras quisimos implementar un menú de inicio con categorías. Desde otro archivo creamos una pantalla de inicio con sus respectivas categorías y dependiendo de lo que elegía el usuario el juego iniciaba. Pero tuvimos un problema y no lo pudimos implementar. El problema fue que mientras el juego estaba en la pantalla de inicio, los segundos del juego corrían igual (tenía que empezar recién en la pantalla de juego). Intentamos de varias formas, creando un archivo nuevo y llamando a principal, dentro del archivo principal, adentro de la f unción main(). También intentamos con varios bucles, uno que diga intro\_juego y otro comenzar\_juego inicializando intro\_juego en True y comenzar\_juego en False, hacer las sentencias necesarias para la pantalla principal, después cuando se apretaba en la categoría se cambiaba la variable comenzar\_juego a True e intro\_juego a False, para que termine el bucle de la pantalla de inicio y entre al del juego, pero cuando entraba, el tiempo seguía corriendo desde la primera ejecución.

También intentamos solucionarlo con la función time.sleep (tiempo), pero igualmente el tiempo corría, no hubo maneras de solucionarlo. Adjunto el código de la pantalla principal (es muy básico ya que era de prueba, solo tiene fondo, un título y dos opciones de categorías)

import os

import random

import sys

import math

import pygame

from pygame.locals import \*

from configuracion import \*

from funcionesVACIAS import \*

from extras import \*

from Boton import \*

from principal import \*

pygame.init()

negro = (0,0,0)

blanco = (255,255,255)

verde = (0,255,0)

rojo = (255,0,0)

azul = (0,0,255)

size = (1280,720)

aceptar = pygame.Rect (510,220,360,50)

salir = pygame.Rect (510,320,360,50)

fuente = pygame.font.SysFont("Helvetica",TAMANNO\_CATEGORIAS)

#creo la ventana

pantalla = pygame.display.set\_mode(size)

pygame.display.set\_caption("PEGUELE AL PRECIO")

fuenteInicio = pygame.font.SysFont ("Helvetica",60)

tituloInicio = fuenteInicio.render("PÉGUELE AL PRECIO",True, COLOR\_NEGRO)

def ppal():

run = True

while run:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

if event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:

if aceptar.collidepoint(pygame.mouse.get\_pos()):

return 0

if salir.collidepoint(pygame.mouse.get\_pos()):

return 10

keys = pygame.key.get\_pressed()

if (keys[pygame.K\_ESCAPE]):

pygame.quit()

sys.exit()

#pinto el fondo

pantalla.fill((255,255,255))

#dibujo los rectangulos

pygame.draw.rect(pantalla,(100,100,100),aceptar,0)

pygame.draw.rect(pantalla,(100,100,100),salir,0)

hardware = fuente.render("Productos de computacion",1,(255,255,255))

limpieza = fuente.render("Productos de limpieza",1,(255,255,255))

#dibujo en la pantalla

pantalla.blit(tituloInicio,(400,80))

pantalla.blit(hardware,(aceptar.x+(aceptar.width - hardware.get\_width())/2,aceptar.y+(aceptar.height - hardware.get\_height())/2))

pantalla.blit(limpieza,(salir.x+(aceptar.width - limpieza.get\_width())/2,salir.y+(salir.height - limpieza.get\_height())/2))

pygame.display.flip()

pygame.display.quit()

if ppal() == 0:

main\_electrodomesticos()

else:

main\_limpieza()



Por ejemplo, si tardas 5 segundos en poner una categoría, en el juego principal esos 5 segundos ya pasaron.  
