

Seminario de Lenguajes - Python

(SI207)

Año 2022

Informe Final



Aberastain Marzorati, Álvaro.....19950/8
Beraún Centineo, Julián Jorge.....19962/3
Bertón, Julián.....17788/8
de la Vega, Gonzalo.....16811/4

Índice

Introducción	1
Marco teórico	2
Problemas y soluciones durante el desarrollo	3
Consideraciones éticas sobre el desarrollo	4
Referencias	5
Anexo 1 - Guía de usuario	6
Anexo 2 - Guía para el desarrollador	12

Introducción

En este informe se detalla el desarrollo del juego denominado FiguRace, proyecto final de la materia Seminario de Lenguajes, opción Python.

El juego consiste en una serie de rondas; cada una presenta hasta cinco características asociadas a un elemento seleccionado al azar que pertenece a cierta temática. Utilizando esas características, el usuario debe deducir de qué elemento se trata, y seleccionarlo de entre cinco opciones que se disponen en pantalla. De ser correcta la opción seleccionada, se suma una determinada cantidad de puntaje; caso contrario, se resta otra cantidad determinada de puntaje.

Tras haber jugado el total de rondas, se finaliza la partida y se almacenan datos de la partida necesarios para el apartado de estadísticas del proyecto.

Se puede elegir una dificultad de las predeterminadas o, para personalizar la experiencia, en el menú de configuración se pueden ajustar el número de rondas, el tiempo de la misma y el dataset (tabla) del cual se selecciona al azar el elemento, entre otras opciones.

Marco teórico

El proceso de desarrollo se llevó a cabo con la motivación de cumplir todos los objetivos principales propuestos por la cátedra. Estos objetivos comprendían aspectos como la correcta organización de carpetas/paquetes del programa y el correcto funcionamiento de la persistencia de datos y la implementación de las pantallas. Para eso, constantemente se revisaron las clases teóricas y se consultó con los ayudantes asignados, en conjunto con reuniones virtuales grupales cada vez que fue necesario.

Se empleó un conjunto de herramientas que permitieron abordar los distintos problemas de una forma que se asemeja a la práctica general del lenguaje en el mundo laboral. A continuación, se detallan las principales.

Uno de los primeros pasos fue decidir las versiones de Python y de las bibliotecas a utilizar. Para evitar conflictos de compatibilidad y asegurar un trabajo consistente desde todas las máquinas, se trabajó con un archivo de requerimientos que especifica las versiones y paquetes a instalar, y se trabajó sobre un entorno virtual. Los entornos virtuales se pueden definir como *“[...] directorios de instalación aislados. Este aislamiento te permite localizar la instalación de las dependencias de tu proyecto, sin obligarte a instalarlas en todo el sistema.”* [1]. El entorno virtual que se utilizó es Virtualenv, una herramienta para desarrollo en Python, el cual *“[...] proporciona soporte para crear entornos virtuales ligeros con sus propios directorios de ubicación, aislados opcionalmente de los directorios de ubicación del sistema. Cada entorno virtual tiene su propio binario Python [...] y puede tener su propio conjunto independiente de paquetes Python instalados en sus directorios de ubicación.”* [2].

En cuanto al apartado gráfico, se recurrió a la biblioteca PySimpleGUI. Esta biblioteca *“[...] es un paquete de Python que permite a los programadores de Python de todos los niveles crear una GUI. Usted especifica su ventana GUI usando un "diseño" que contiene widgets (se llaman "Elementos" en PySimpleGUI). Su diseño se usa para crear una ventana usando uno de los 4 frameworks compatibles para mostrar e interactuar con su ventana. Los frameworks compatibles incluyen tkinter, Qt, WxPython o Remi. [...]”* [3]. Así, resultó una interfaz gráfica simple de programar y eficaz en su tarea, aunque con limitaciones en el aspecto artístico.

También se utilizó la biblioteca pandas para dos cuestiones fundamentales. Primero, para filtrar y simplificar información contenida en las tablas que iban a ser utilizadas; y luego para el procesamiento de datos con el fin de efectuar análisis estadísticos: *“[...] pandas es una biblioteca de software escrita como extensión de NumPy para manipulación y análisis de*

datos para el lenguaje de programación Python. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.” [4]. De acuerdo a sus creadores “pretende ser el bloque de construcción fundamental de alto nivel para realizar análisis prácticos de datos del mundo real en Python. Además, tiene el objetivo más amplio de convertirse en la herramienta de manipulación/análisis de datos de código abierto más poderosa y flexible disponible de cualquier lenguaje.” [5]

Junto a pandas, se utilizó la librería Matplotlib para complementar los resultados estadísticos con una adecuada visualización. “[...] Matplotlib produce figuras con calidad de publicación en una variedad de formatos y entornos interactivos en todas las plataformas. Matplotlib se puede usar en scripts de Python, shells de Python/IPython, servidores de aplicaciones web y varios kits de herramientas de interfaz gráfica de usuario. [...]” [6].

Problemas y soluciones durante el desarrollo

Durante el desarrollo del juego, surgieron problemas de distinta índole y complejidad. Los que eran más sencillos fueron solucionados en base al criterio del grupo ya que no afectaban drásticamente al futuro del proyecto. Por otro lado, para los más difíciles se requirió de orientación por parte de los ayudantes para optar por el camino más conveniente. A continuación, se detallan los problemas más complejos surgidos a lo largo del desarrollo.

El primer inconveniente surgió con el dataset de películas. El mismo tenía un conflicto con el uso de comas que causaba líneas vacías al procesarlo, por lo que se optó por preprocesar el archivo para resolverlo.

Otro problema significativo que ocurrió fue la creación de dificultades. Se tenía en claro cuáles iban a ser las cuatro dificultades por defecto, sin embargo, el debate surgió tras notar que la pantalla de configuración iba a estar a disposición del usuario, la cual permitiría modificar los aspectos de la partida por fuera de los valores predeterminados de las mismas. Esto sería fruto de inconsistencias para el futuro análisis de datos, entre otros inconvenientes. Tras el debate se llegó a la conclusión de que cualquier conjunto de valores de configuración que no coincidiera con alguno de los niveles definidos iba a ser considerado como una dificultad “Personalizada”, y la misma no se tendría en cuenta para los cálculos de los puntajes y estadísticas.

Para finalizar, cabe destacar el obstáculo que significó la interpretación de los enunciados para los cálculos de las estadísticas. Algunos de ellos no resultaron lo suficientemente

específicos y surgieron distintos puntos de vista, así que se optó por elegir aquellos que parecían más coherentes.

Consideraciones éticas sobre el desarrollo

Para el desarrollo del juego no fue necesario utilizar un software que requiriera de una licencia. Si bien el desarrollo del mismo fue realizado sobre ordenadores con sistema operativo (SO) Windows, su ejecución no requiere tener instalado dicho SO en la computadora utilizada. Se efectuaron pruebas en una máquina virtual con sistema Linux para comprobar su correcto funcionamiento.

Como se mencionó con anterioridad, para la creación de FiguRace, se utilizaron diversas herramientas con el fin de facilitar múltiples tareas y evitar potenciales conflictos. Tanto Virtualenv como PySimpleGUI, pandas, y Matplotlib no tienen costo económico alguno y son denominados software libre, es decir que se tiene acceso a su código fuente y documentación para su aplicación, y permiten la posibilidad de mejorar el software y reportar errores que se encuentren.

Aunque en ocasiones el tiempo de entrega resultó acotado, y a pesar de que no se llegaron a aplicar todas las ideas opcionales que se propusieron; sí se alcanzó un funcionamiento pleno de los puntos principales. A modo de observación, se puede agregar que en ninguna instancia se planteó la idea de implementar un sistema que, por ejemplo, pueda adaptarse al tipo de daltonismo del usuario, o que se pueda jugar sólo basándose en guías auditivas.

Conclusiones y trabajos futuros

De este proyecto hubo mucho aprendizaje, no sólo de aspectos técnicos, sino también de organización y planificación.

Esta fue una experiencia de desarrollo de software de mayor escala a la habitual y para todos los integrantes fue la primera vez que se trabajó en un proyecto en grupo. Fue primordial mantener una constante comunicación y un respeto mutuo en los intercambios de opiniones.

Una de las mejoras que quedó pendiente debido a falta de tiempo fue la del aspecto visual del juego, mediante una mejor distribución de los menús y diseño de los elementos que los componen, además de la accesibilidad para los usuarios con algún tipo de daltonismo.

Otra de las ideas fue el agregado de imágenes para aportar, en conjunto con las características, otra pista acerca del elemento presentado, aunque sería un trabajo arduo obtener y adaptar las imágenes correspondientes.

Como un detalle, también en un principio se buscaba mostrar más información acerca de la última partida jugada en el menú principal, como el puntaje que había logrado y la fecha de la misma.

Por último, agregar más datasets/tablas al repertorio ya disponible, con el fin de diversificar las temáticas y abarcar más usuarios.

Se puede destacar del proceso de desarrollo el compromiso que hubo por parte de cada miembro del grupo por hacer el proyecto no solo de la manera más prolija posible, sino también con entusiasmo y con el propósito de obtener el mejor funcionamiento posible.

Este seminario y proyecto dejaron en todo el grupo un deseo de profundizar en las diversas áreas de la informática en las que se emplea este lenguaje como el análisis de datos o el desarrollo web, y brindaron un abanico de ejemplos y herramientas que sin dudas van a inspirar y formar parte del uso personal y/o profesional.

Referencias

- [1] <https://www.freecodecamp.org/espanol/news/entornos-virtuales-de-python-explicados-con-ejemplos/#:~:text=Los%20entornos%20virtuales%20se%20pueden,dos%20aplicaciones%20App1%20y%20App2>
- [2] <https://docs.python.org/es/3.8/library/venv.html>
- [3] <https://github.com/PySimpleGUI/PySimpleGUI#what-is-pysimplegui->
- [4] [https://es.wikipedia.org/wiki/Pandas_\(software\)](https://es.wikipedia.org/wiki/Pandas_(software))
- [5] <https://pandas.pydata.org/about/>, extracto de "Mission"
- [6] <https://github.com/matplotlib/matplotlib>, primer párrafo del readme

Anexo 1 - Guía de usuario

Para poder ejecutar el juego es requisito tener Python 3.10 o una versión posterior instalada. Desde una terminal, mediante pip se deben instalar los módulos necesarios mediante el comando `pip install -r requirements.txt`, y, para correr la aplicación, el usuario se debe ubicar dentro de la carpeta raíz de Figurace y ejecutar el comando: `py figurace.py`. En caso de que el juego esté siendo ejecutado por primera vez, se muestra una ventana que pide la información correspondiente con el fin de crear un perfil de usuario tal como se muestra en la figura 1. El usuario debe completar esta información para proseguir a la ventana del menú principal.

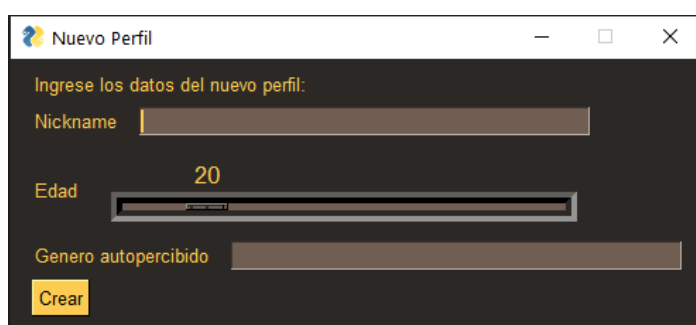


Figura 1. Primera creación de perfil

Menú principal

En la figura 2 se muestra el menú principal, que contiene el nombre del perfil de usuario que jugó la última partida, un listado de perfiles seleccionables y uno de dificultades, y un grupo de botones que llevan a las diferentes ventanas del programa.

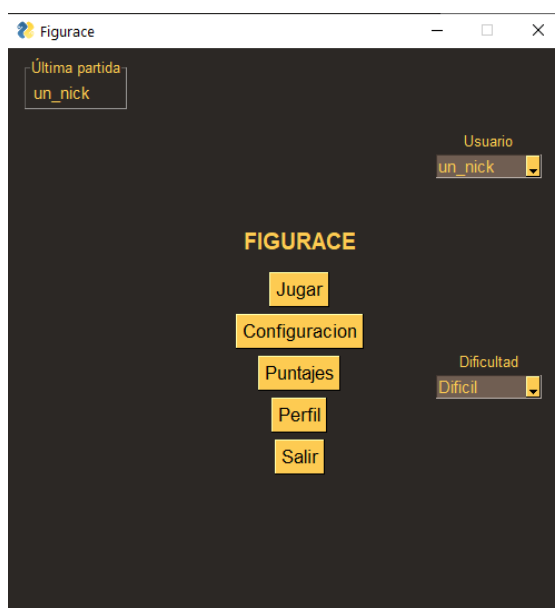


Figura 2. Menú principal

Pantalla de configuración

En la pantalla de configuración se disponen los parámetros que definen a la partida. Como se muestra en la figura 3, este menú se conforma por:

- Un listado desplegable que muestra los conjuntos/datasets disponibles
- Un conjunto de controles deslizadores (del inglés, “sliders”), los cuales permiten configurar, dentro de un rango definido, la cantidad de rondas, el tiempo de ronda, el puntaje que se sumará y restará tras la selección de una opción, la cantidad de características a mostrar del elemento.
- Botones para guardar los cambios y para volver al menú principal

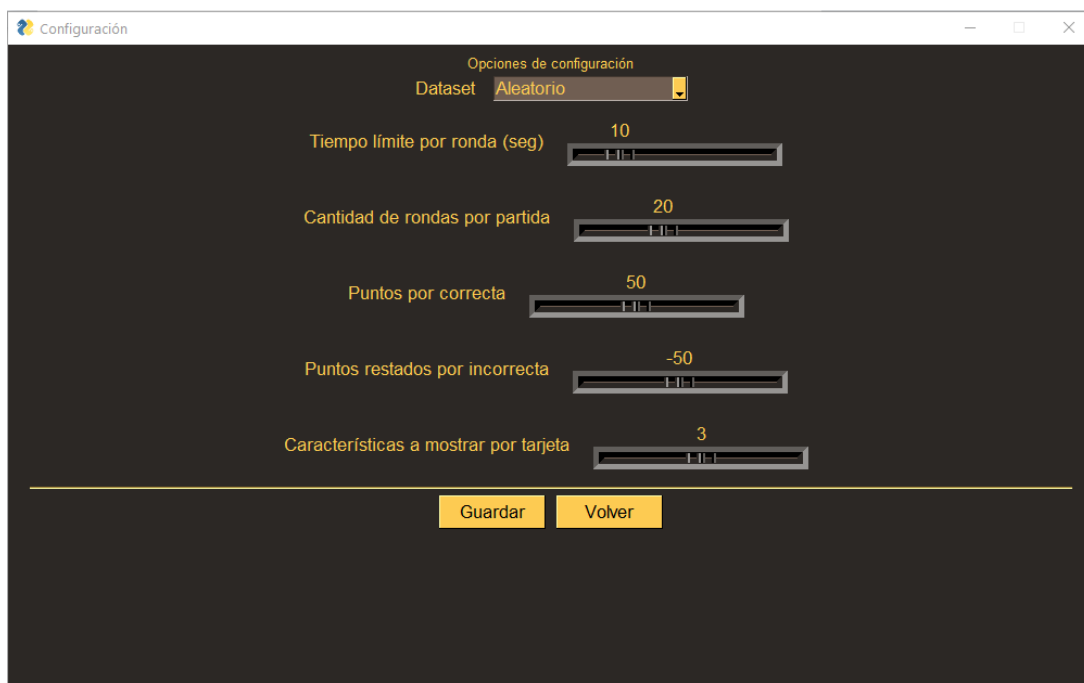


Figura 3. Pantalla de configuración

Si desea modificar las características del juego, se debe ajustar en cada elemento el valor deseado y presionar el botón “Guardar” para hacer efectivos los cambios.

Cabe destacar que al guardar cualquier variación de los atributos de la partida, se establece el juego en una dificultad por fuera de las predeterminadas, denominada “personalizada”, incluso cuando estos coincidan con los valores por defecto de las dificultades ya definidas.

Pantalla de perfiles

La pantalla de perfiles ofrece la posibilidad de visualizar, modificar y crear nuevos perfiles. Este apartado del programa, como se presenta en la figura 4, se muestra de la siguiente forma:

- Una tabla en la que se dispone de todos los perfiles ya existentes
- Botones para crear un nuevo perfil y para regresar al menú principal



Figura 4. Pantalla de perfiles

Al presionar sobre el botón “Crear jugador”, se muestra una nueva ventana (figura 5), en la que se pide ingresar los datos del usuario de tal forma que no se utilice un nombre de usuario (o nickname) vacío o ya existente, y no se ingrese un género autopercibido vacío. Una vez que ingresada la información, al presionar el botón “Crear perfil”, se hace la verificación de las condiciones mencionadas anteriormente y se crea si las cumple.

Para modificar un perfil, simplemente se debe presionar sobre uno de los perfiles de la tabla. Al hacer esto se abre una nueva ventana, como la de la figura 6, en la que se muestra la información del mismo en los distintos campos como si fuera la ventana de creación de perfiles. Allí se pueden hacer las modificaciones deseadas, a excepción del nombre de usuario. Una vez efectuados los cambios, se presiona el botón “Editar perfil” para hacerlos efectivos.

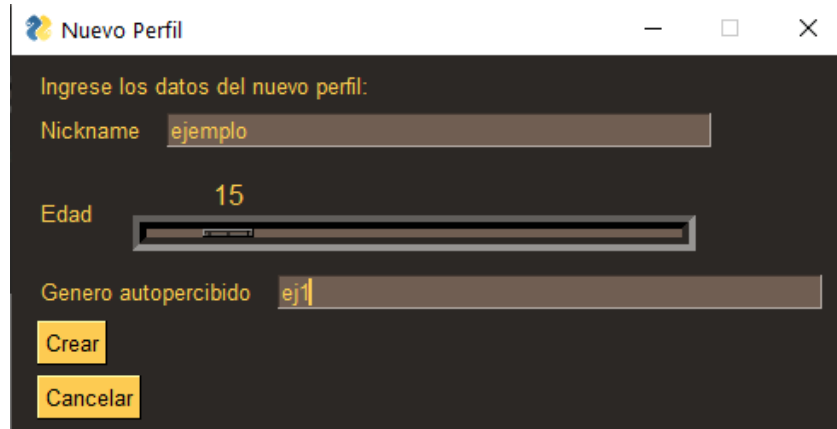


Figura 5. Crear perfil



Figura 6. Editar perfil

Pantalla de puntajes

La pantalla de puntajes presenta cuatro columnas de tablas que representan cada una de las dificultades. Como se ve en la figura 7, la ventana se divide en dos secciones. En la mitad superior se encuentran las tablas correspondientes a los puntajes obtenidos por los usuarios en cada partida; mientras que en la mitad inferior se hallan los puntajes promedio de cada usuario entre todas las partidas.

Puntajes

Top 20 puntajes

Fácil		Normal		Difícil		Einstein	
Usuario	Puntos	Usuario	Puntos	Usuario	Puntos	Usuario	Puntos
otro_Usuarie	70.0	whatsername	200.0	un_nick	-100.0	Sin datos	-
otro_Usuarie	70.0	whatsername	-175.0	un_nick	-200.0		
whatsername	70.0	whatsername	-250.0	whatsername	-400.0		
otro_Usuarie	10.0			un_nick	-500.0		
otro_Usuarie	10.0			whatsername	-800.0		
whatsername	10.0			un_nick	-900.0		
whatsername	10.0						
otro_Usuarie	-50.0						
whatsername	-50.0						

Top 20 promedios

Fácil		Normal		Difícil		Einstein	
Usuario	Puntos	Usuario	Puntos	Usuario	Puntos	Usuario	Puntos
otro_Usuarie	22.0	whatsername	-75.0	un_nick	-425.0	Sin datos	-
whatsername	0.0			whatsername	-600.0		
un_nick	-50.0						

Volver

Figura 7. Pantalla de puntajes

Pantalla de juego

Presionar el botón “Jugar” en el menú principal da inicio a una partida. Aunque esta pantalla es la más dinámica de todas, la estructura de la misma a lo largo de todas las rondas es la misma. Tal como se ve en la figura 8, en la mitad superior se muestran las características del elemento escogido al azar y en la mitad inferior se encuentra un conjunto de botones y el contador de tiempo.

Los cinco primeros botones representan las opciones de respuesta, de las cuales el usuario debe escoger una. Además, hay otros dos botones: el de “Pasar ronda”, que hace que se avance al siguiente elemento sin dar una respuesta ni alterar el puntaje; y el de “Abandonar partida”, que finaliza la misma, mostrando en pantalla (figura 9) el puntaje conseguido. Si se abandona la partida, el puntaje no es almacenado y, por consiguiente, no se tiene en cuenta para los cálculos y estadísticas.

Cuando el tiempo del contador llega a cero, se pasa automáticamente a la siguiente ronda, sin variar el puntaje, tal como lo hace el botón “Pasar ronda”.



Figura 8. Pantalla de juego



Figura 9. Volver al menú

En caso de que se llegue al límite de rondas, se presenta en pantalla el aviso de partida finalizada (figura 10) junto con el puntaje total obtenido, el cual es almacenado y se tiene en cuenta para los cálculos y estadísticas.

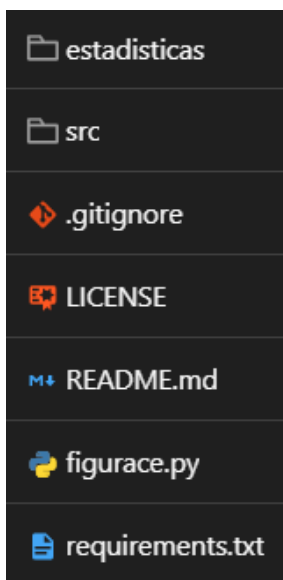


Figura 10. Partida finalizada

Anexo 2 - Guía para el desarrollador

A continuación se detalla la organización de carpetas, archivos y módulos del trabajo para facilitar posibles modificaciones y desarrollos en el mismo.

La estructura de carpetas se visualiza en la figura 1.



Lo más importante a entender de la organización es el contenido de las dos carpetas `estadísticas` y `src` y el archivo `figurace.py`: este último es el módulo `__main__` y tiene como función lanzar el videojuego iniciando la pantalla de menú principal; en `estadísticas` simplemente se encuentran los Jupyter del análisis estadístico; y por otro lado, en `src` (figura 2) se localiza: una carpeta `datos` para los archivos necesarios, como los perfiles, la configuración, los puntajes guardados, etc.; también la carpeta de los `datasets`, que guarda tanto los Jupyter como los `.csv`, y por último y lo más importante a desarrollar, el directorio `core`.

Figura 1. Estructura principal del repositorio

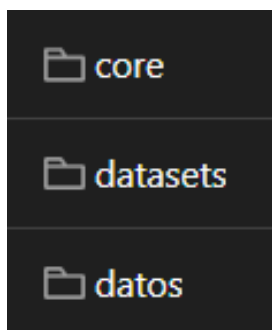


Figura 2. Directorio `src`

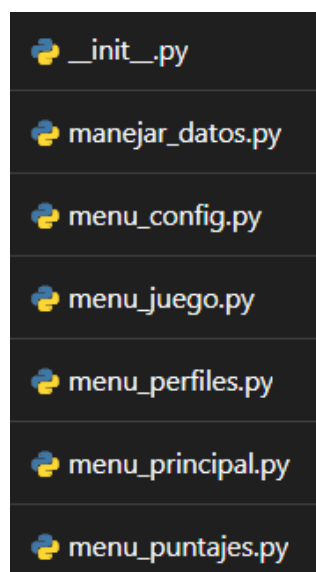


Figura 3. Directorio `core`

Este paquete `core` define el paquete principal del programa. En él se encuentran los módulos de cada pantalla, que se encargan de estructurarlos visualmente y definir su comportamiento; y además se halla un módulo que contiene la lógica del manejo de datos (creación, lectura y modificación de archivos).

Todos los módulos se comunican entre sí para lograr las funcionalidades. Por ejemplo, el módulo de la pantalla de configuración se comunica con el de manejar datos para leer la configuración actual y sobrescribir el archivo si se realizan cambios, y el módulo de la pantalla de juego lee la configuración actual para construir la partida y en ciertas ocasiones envía luego los datos de los eventos ocurridos y del puntaje final para que se persistan.

El código fue escrito de tal forma que resulte lo más claro y prolijo a la vista, y está documentado con docstrings para facilitar el entendimiento de las funciones.