
Seminario de Lenguaje Python

Informe Final

Iñaki Lapeyre - Josué Suarez - Lucas Gallardo - Fabián Martinez

Índice

Introducción (máx. 200 palabras)

En este informe se detalla el proceso de creación del juego interactivo Figurace desarrollado en la materia Seminario de Lenguajes, con el fin de aprender Python desde lo más básico y con la ayuda de la librería PySimpleGUI diseñaremos una interfaz sencilla para la creación de un juego.

El juego consiste en adivinar una persona/lugar según la categoría elegida (Spotify,lagos o FIFA) entre x opciones basado en diversas características dadas según la dificultad, quien más acierte sin errar obtendrá la mejor puntuación. La dificultad del juego se puede personalizar a gusto o elegir entre las ya propuestas(Fácil, Intermedio , Difícil), posee además un sistema de creación y edición de perfiles para registrar las mejores puntuaciones de cada uno y sus estadísticas generales.

Se implementó todo lo aprendido en la materia a lo largo del proyecto y también indagamos sobre un patrón de diseño que se adecue a la aplicación y facilite la comprensión y legibilidad del código. A su vez ayudando a la organización en equipo y separar las tareas de una forma más ordenada.

A continuación profundizaremos más en cómo lo realizamos y qué alternativas se tomaron para tener un ambiente más organizado de trabajo.

 Seminario de Lenguaje Python

Temas estudiados (marco teórico)

Temas estudiados y/o profundizados:

- Git
- MVC
- Nociones del lenguaje
- Tipado estático
- Librerías externas

Git

Es una herramienta de control de versión para el desarrollo de software. Esta permite tener un historial donde se guardan puntos que sirven para recuperar el estado del proyecto en determinado momento de manera que permite trabajar libremente sin perder la seguridad que lo desarrollado previamente se pueda perder.

Además de esto también ofrece otras facilidades como tener versiones del proyecto en diferentes ramas autocontenidas que permiten trabajar en partes del código sin modificar el contenido de otra rama, las cuales luego pueden juntarse, borrarse o solo dejarlas existir.

Para mayor informacion visitar la pagina oficial: https://git-scm.com/

MVC

Las siglas MVC hacen referencia a Modelo Vista Controlador, que es un patrón de diseño de software para la creación de aplicación donde se intenta facilitar el desarrollo siguiendo un simple lineamiento que define cómo interactúa la interfaz del usuario con la lógica y los datos.

Para lograr esto se definen tres partes:

1. Modelo: maneja los datos y la lógica
2. Vista: Se encarga del diseño e interfaz
3. Controlador: comunica los modelos y las vistas

Para mayor informacion una buena referencia:
https://developer.mozilla.org/es/docs/Glossary/MVC

Nociones del lenguaje


Seminario de Lenguaje Python

Sobre el lenguaje manejamos los siguientes conceptos:

- Datos primitivos, secuencias
- Funciones y funciones anónimas
- Decoradores
- Módulos
- Manejo archivos
- Formato JSON y csv
- Excepciones
- Clases
- Iteradores, generadores, map, filter
- Tipado estático

Datos primitivos, secuencias

Los datos primitivos son la manera de representar datos simples

- Int: números enteros
- Float: números reales
- Strings: texto (secuencia de caracteres)
- Boolean: valor de veracidad

Los datos de secuencias son:

- List: conjunto heterogéneo variable de datos, accesibles por un índice entero
- Tuple: conjunto heterogéneo fijo de datos, accesibles por un índice entero
- Set: conjunto heterogéneo fijo de datos únicos, accesibles por un índice entero
- Dict: conjunto heterogéneo de clave (dato inmutable) y valor (dato mutable o no)

Funciones y funciones anónimas

Las funciones (def) son segmentos de código encargados de realizar una tarea con posibles datos de entrada que puede llegar a producir un resultado directo devolviendolo o un resultado indirecto modificando algo fuera del alcance de la función. Estas funciones deben ser explícitamente declaradas en el cuerpo de un módulo, una clase u otra función, en cambio una función anónima (lambda) puede ser declarada en cualquier momento para ser pasada por parámetro o asignada a una variable. Esta ultima es mas limitada y se reduce a unica linea de codigo.

Decoradores

Los decoradores son una manera especial de utilizar funciones que reciben una función, método o clase la cual permite modificar y/o agregar un comportamiento del dato de entrada o extender su funcionalidad.


Seminario de Lenguaje Python

Módulos

En python los módulos son los archivos de python perse, esto brinda una facilidad increíble para la disección de la aplicación en diferentes apartados también así para un sencillo manejo de estado ya que cada módulo contiene su propio estado y sus propias definiciones de variables, funciones y clases.

Manejo archivos

El manejo de archivos es la forma de interacción del programa con datos conservados en memoria secundaria. Esto es útil para la preservación de información. Esto puede simplificarse cuando se trabaja con manejadores de contexto los cuales facilitan la limpieza de recursos al finalizar su uso.

Formato JSON y csv

Las siglas JSON hacen referencia a JavaScript Object Notation las cuales quieren decir que es un formato para la notación de objetos de javascript en texto plano. Este formato es habitualmente utilizado en el ambiente Web.
Las siglas csv hacen referencia a comma separated values las cuales quieren decir que es un formato donde la información está separada por comas dentro de una misma línea de texto y múltiples líneas, al igual que en JSON los valores están en texto plano. Este formato es habitualmente utilizado en el ambiente de trabajo con grandes conjuntos de datos (DATASETS).

Excepciones

Las excepciones son la manera en que se manejan resultados imprevistos al realizar ciertas operaciones. Cuando ocurre un error en la ejecución del programa se levanta una excepción la cual se va a propagar esperando que haya un manejador de esta misma para tratar de recobrar la funcionalidad del programa y/o finalizarlo de manera correcta.

Clases

Cuando se intenta modelar un cosa normalmente a esta se asocian información (datos) y comportamientos, para hacer esto se puede definir el esquema de un objeto (clase) el cual sirve como molde para generar diferentes copias del mismo donde puede variar el estado interno de manera que cada uno representa una instancia distinta de lo que se quiere modelar teniendo las mismas capacidades en términos de comportamientos para cada uno.

Iteradores, generadores

Los iteradores son funciones las cuales se utilizan para aplicar una función a todos los elementos de una secuencia de manera de producir una nueva secuencia con todos los elementos transformados en el caso del map, con una parte de los elementos en el caso del filter, o con un resultado en el caso de reduce.
Los generadores son una manera especial de obtener los valores de una secuencia donde cada valor es computado por única vez cuando es accedido, esto permite que los datos realmente no están siendo almacenados innecesariamente siendo más eficiente en este aspecto.

Tipado estático

Si bien python es un lenguaje fuertemente tipado y dinámico, lo cual indica que los tipos son inferidos durante la ejecución del programa, esto no quita que el lenguaje goza de la posibilidad de agregar anotaciones al código para que herramientas de análisis de código estáticas o editores de código inteligentes ofrecen un mejor soporte para el lenguaje además de facilitar y mejorar la experiencia de desarrollo y mantención del programa.

Problemas y soluciones surgidas durante el desarrollo

Al ser la primera vez que trabajamos en grupo con GIT hubo ciertos inconvenientes por ejemplo:

- Se realizó un commit con varios cambios en la rama principal

donde un integrante estaba testeando el código en una carpeta local (modificando bastante el código), y a la hora de subir los cambios de la carpeta en la que estaba trabajando normalmente, se confunde y realiza un push de los cambios sobre la carpeta que él estaba testeando. Por suerte se pudo solucionar con los siguiente:

git branch rama-pruebas
git reset HEAD~ --hard
git checkout rama-pruebas

- Me equivoque a la hora de nombrar un commit (ya sea por el propósito del comit o simplemente faltas de ortografía)

Se soluciono escribiendo: git commit --amend

Consideraciones éticas sobre el desarrollo

Al ser software libre cualquier desarrollador que quiera puede modificar, copiar, distribuir, estudiar,y mejorar el software.
Decidimos utilizar la licencia MIT ya que es una Licencia de software libre permisiva lo que significa que impone muy pocas limitaciones en la reutilización y por tanto posee una excelente Compatibilidad de licencia. La licencia MIT permite reutilizar *software* dentro de Software propietario.

Conclusiones y trabajos futuros

Este proyecto cumplió con su propósito de aprender lo básico sobre Python , permitiéndonos implementar por completo el temario del seminario de lenguajes y a su vez pulir nuestras habilidades para desempeñarse en grupo y organizar un proyecto más serio junto a los beneficios y consecuencias que esto puede llevar. Cabe destacar que agudizó nuestra capacidad de buscar información en la documentación oficial al investigar tanto de python como de la librería utilizada PySimpleGUI. Siendo este proyecto una buena práctica a futuro tanto individual como grupalmente.

Anexo 1: guía de usuario

- **Selección de perfiles**

En la figura 1 se muestra nuestra Interfaz que nos permite guardar, editar o eliminar un perfil de usuario, de este quedarán registrados tanto los progresos, como las configuraciones personalizadas del mismo.

figura 1: Seleccionar perfiles

- **Crear Perfil**

Una vez le demos a crear nuestro perfil, tenemos que completarlo con los datos que deseados (como se muestra en la figura 2)

figura 2: Crear Perfil

- **Pantalla de inicio**

Pantalla de inicio en la que además de iniciar partida con nuestro perfil, podemos, volver a la edición de perfiles, también podemos configurar los datos tanto de la dificultad como la de los datos personales.

figura 3 : Pantalla de Inicio

- **Configuración**

Pantalla para poder personalizar las configuraciones de dificultad a gusto del usuario

figura 4: Configuración

- **Configuración de partida**

Una vez que iniciamos partida, tenemos una cantidad de opciones en la que podemos elegir la dificultad de nuestra partida y el dataset con el que queremos jugar

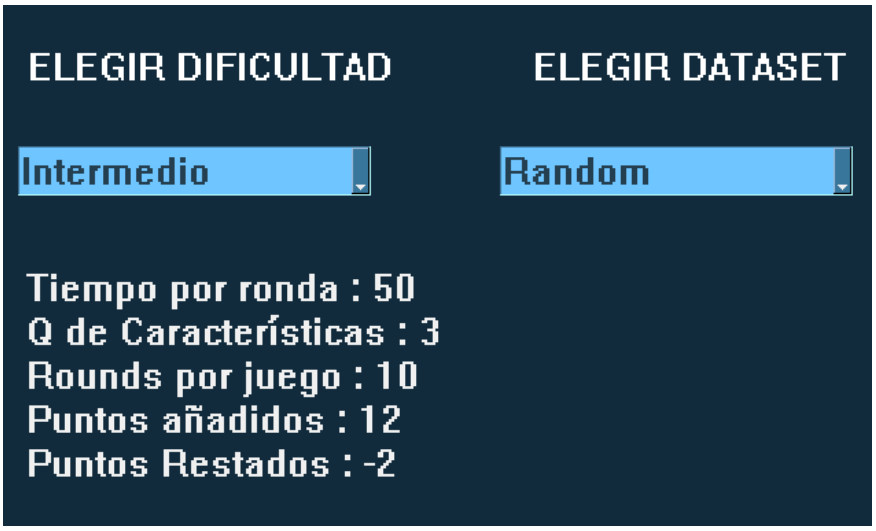

figura 5: Configuración de Partida

- **A jugar!!**

Una vez que elegimos el dataset y dificultad, ya podemos jugar de forma normal

Anexo 2: guía para el desarrollador

1. **Requerimientos de sistema**
   Se necesita tener python versión mayor o igual a 3.10
   En caso de que no, una de estas guías puede ser de ayuda:

   - Windows
   - Linux
   - Mac
2. **Obtener el repositorio**

   Existen dos maneras:

   - Clonar el repositorio por medio de SSH o HTTPS
   - Descargar el .zip (luego descomprimirlo)

3. **Instalación de dependencias**

Las dependencias son:

- Juego: PySimpleGUI
- Datasets: notebook, pandas
- Analysis: notebook, pandas, matplotlib

4. **Para su instalación:**
   - Primero abrir una terminal/consola en la ubicación donde se descargó el contenido del repositorio
   - Luego ejecutar el siguiente comando

     pip install -r requirements.

5. **Ejecución**

   Ejecución de los diferentes apartados
   Se asume que se encuentra en una terminal/consola en la ubicación donde se descargó el contenido del repositorio y realizo los pasos previos de la guía

   - **Juego**
     py figurace.py
   - **Procesamiento Datasets**
     Esta sección se encuentra en la carpeta dataset_section
     El procesamiento de los datasets se encuentra como un script de python o un cuaderno interactivo de JupyterNotebook en la carpeta second_assignment
     Los datasets a procesar en la carpeta base_datasets
   - **Análisis de eventos**
     Esta sección se encuentra en la carpeta analysis_section
     El análisis de los eventos generados al jugar partidas se encuentra como un cuaderno interactivo de JupyterNotebook.

Modo desarrollador:
Se puede iniciar el juego en modo dev (desarrollador), este modo fue pensado para desarrollar la aplicación de una manera más fácil. Este modo se activa al ejecutar la aplicación con el argumento extra --dev.

Por defecto el modo dev inicia la aplicación en la pantalla de selección de perfiles y a su vez pone un tiempo de inactividad máximo de 5 segundos para el cierre automático de la aplicación (si se juega una partida se desactiva el cierre automático).

Los argumentos adicionales para este modo siguen el siguiente formato:

--(nombre argumento)=(valor para el argumento)

Los argumentos posibles son:

Nombre	**Valor/es**	**Descripción**	**Ejemplo**
help		Informa en consola sobre los argumentos posibles en el modo dev	--help
to	duración (segundos)	El tiempo de inactividad para el cierre de la aplicación	--to=10
is	pantalla inicial (SCREEN_NAME)	La pantalla en la cual iniciar la aplicación	--is=-MENU-
el	booleano (true o false)	Habilita el logeo en consola información sobre los eventos	--el=true