

# Trabajo Integrador 2022

## Seminario de Lenguajes Opción Python

Teniendo en cuenta lo realizado en la primera parte de la entrega se va a solicitar que completen ciertas funcionalidades sobre el juego FiguRace además de las descritas en la primer entrega.

A continuación vamos a describir distintos aspectos del juego que vamos a mejorar para esta segunda parte.

### FiguRace: carrera de datos

El juego comienza cuando se presiona el botón “Comenzar” en la ventana del tablero de juego. Como ya sabemos el juego trata de encontrar la respuesta correcta de las posibles respuestas. En cada turno se mostrarán características de objeto o persona, tendrá un tiempo definido según el nivel para adivinar la respuesta correcta, luego del tiempo que pasó de la ronda se pasará a la siguiente, según la cantidad total que tenga configurado el nivel.

En caso que acierte se considera como respuesta correcta y sumarán los puntos que se hayan definido y acumulará con los puntos correspondientes en las siguientes rondas. El puntaje quedará relacionado al usuario y al nivel.

El juego termina cuando completa todas las rondas definidas. Además, al terminar el juego se debe registrar el tiempo que duró la partida.

En cualquier momento puede salir del juego con un botón que será parte de la interfaz del juego. Cuando esto sucede se le mostrará al usuario la pantalla de puntos en un popup o nueva ventana y luego pasará a la pantalla “Menú de inicio del juego”.

### Sistema de ayudas (Opcional)

Además de implementar toda la dinámica del juego descrita en el punto anterior se puede opcionalmente implementar un sistema de ayudas o pistas que puede solicitar el jugador para adivinar la tarjeta en un turno.

El mismo consiste en mostrar más características del objeto o respuesta que el usuario debe encontrar en caso que hayan o relevar respuestas erróneas. Todas las opciones son tomadas del dataset de forma random.

El sistema debe ser activado para el nivel correspondiente en donde se quiera usar.

Hacer uso de las ayudas debería impactar en el cálculo de la puntuación de alguna manera dado que adivinar haciendo uso del mismo simplifica el juego.

Los grupos podrán agregar o modificar el comportamiento de las ayudas si lo consideran mejor para el juego por alguno de complejidad similar. En este caso consulten con el ayudante si aún sigue cumpliendo con los requisitos que se necesitan para cumplir con este punto.

## Carga de datos para el juego

Las tarjetas de FiguRace se generan a partir de un conjunto de datos que deberán ser preparados previamente a diferencia de la primera parte se deberá utilizar la librería **Pandas** para procesar los archivos originales csv. Al igual que en la primera parte el procesamiento se deberá desarrollar en un/unos cuaderno/s de **Jupyter Notebook** que extraiga los datos necesarios para el juego de alguno de los datasets provistos y muestre en forma detallada el proceso realizado con los mismos.

## Pantalla de puntajes

La primera columna contendrá los 20 puntajes más altos ordenados de mayor a menor, mostrando el nick correspondiente a cada puntaje.

Recuerde que tiene que mostrarse una tabla por cada uno de los niveles configurados en el juego.

La segunda contendrá los 20 puntajes promedio más altos ordenados de mayor a menor mostrando el nick correspondiente a cada promedio. De la misma forma que la primera columna debería haber una tabla por cada nivel. El promedio se calcula de la siguiente forma: Se toma el puntaje acumulado total de cada usuario y se divide por la cantidad de veces que el usuario jugó.

## Cálculo de Puntuación

El cálculo de la puntuación queda abierta a la consideración del grupo en acuerdo con el ayudante asignado al grupo. Sólo deben tenerse en cuenta algunas consideraciones:

- Las partidas abandonadas, no deben tenerse en cuenta para la puntuación ni para los rankings.
- El criterio elegido tiene que tener sentido dentro de la jugabilidad. Es decir, la puntuación no debe cambiar el objetivo del juego que es tratar de adivinar las fichas lo antes posible. En caso de alguna duda consultar bien esta parte con el ayudante asignado.

## Registro de jugadas y partidas

En este apartado vamos a dar una guía un poco más detallada de como guardar información del juego para poder realizar luego un análisis. En este punto se dará una descripción de los datos necesarios a registrar.

Para poder hacer análisis de datos sobre las partidas de nuestro juego es necesario poder registrar cada evento que ocurre mientras se está jugando.

Por este motivo se requerirá que muchas de las acciones desarrolladas en cada partida sean registradas como eventos.

En todos los eventos se deben registrar los siguientes campos:

- Marca de tiempo (timestamp)
- Número de Partida (autoincremental o UUID)
- Nombre de evento
- Cantidad total de objetos o personas a adivinar
- Usuario - nick
- Estado
- Objeto o respuesta
- Nivel dificultad

**Deben registrar los siguientes eventos:**

- **Comienzo de partida:** el nombre del evento es "inicio\_partida" y los campos de texto ingresado, respuesta (correspondiente a la respuesta correcta de la ronda) y estado no se completan. En este momento se ingresa el número de partida que debe ser único para cada partida.

timestamp	id	evento	usuario	estado	texto ingresado	respuesta	nivel
1654492780	33ff0802-e53f-11ec-8fea-0242ac120002	inicio_partida	sofi		-	-	normal

- **Chequeo de resultado de turno (se terminó el tiempo o seleccionó la respuesta correcta):** el nombre del evento es "intento", y se completan todos los campos. En el caso de estado los posibles valores son: "error", "ok" o "timeout".

timestamp	id	evento	usuario	estado	texto ingresado	respuesta	nivel
1654632898	bd7f4ee8-e53f-11ec-8fea-0242ac120002	intento	sofi	error	Cristiano Ronaldo	Lionel Messi	normal
1654632999	6c1bae73-a8d8-4ded-97db-636d59af0ee2	intento	sofi	ok	Lionel Messi	Lionel Messi	normal
1654732392	03aea3f6-890c-4194-9613-9c76981083ec	intento	sofi	error	Kylian Mbappé	Alisson	normal

1654742291	a1d62ee7-4f35-48a9-9b3e-4fa2f04a6645	intento	sofi	timeout	-	Alisson	normal
------------	--------------------------------------	---------	------	---------	---	---------	--------

- **Cada vez que se termina una partida:** el nombre del evento es “fin” y el estado puede ser “finalizada” o “cancelada”.

timestamp	id	evento	usuaria	estado	texto ingresado	respuesta	nivel
1654482693	5fb49e99-53a9-48b1-b103-ecbeae12f581	fin	sofi	finalizada	-	-	normal

Los datos de cada evento pueden variar si en el equipo consideran guardar algún dato adicional importante. En caso de alguna duda consultar con el ayudante también el dato que se quiera agregar.

Con estos datos y realizar un procesamiento de los mismos de la forma descrita en la sección **Análisis de datos y estadísticas**.

## Análisis de datos y estadísticas

Dada la información que tenemos acumulada de nuestros juegos, vamos a realizar un archivo en Jupyter Notebook que muestre estadísticas del mismo.

### Estadísticas a mostrar y gráficos

La sección de estadísticas deberá mostrar como mínimo la siguiente información en base a los eventos guardados de todas las partidas:

- Top 10 de palabras que se encuentran primero de todas las partidas.
- Gráfico que muestre el porcentaje de partidas por estado (fin, cancelada, error, timeout). También debe tener en cuenta las partidas que se terminaron inesperadamente o de forma repentina (partidas sin evento de fin).
- Gráfico que muestre el porcentaje de partidas finalizadas según género.
- Gráfico que muestre el porcentaje de partidas finalizadas según nivel.
- Duración total de la partida (desde que el usuaria empieza a jugar hasta que presiona salir o pasan las N rondas definidas para el nivel de dificultad actual).
- Tiempo de la respuesta exitosa más rápida.
- Promedio de tiempo de respuesta para respuestas exitosas.
- Promedio de tiempo de partida general.

- Porcentaje de aciertos por usuario.
- Cantidad de tarjetas para las que el usuario no dió respuesta (timeout).
- Cantidad de tarjetas en las que el usuario dió una respuesta errónea.

Se podrán agregar opcionalmente:

- Gráfico que muestre el porcentaje de partidas que se juegan para cada día de la semana.
- Promedio de tiempo de partidas finalizadas por nivel.
- Porcentaje de palabras encontradas en las partidas "timeout, es decir que finalizaron por falta de tiempo para completar.

## Segunda Entrega

Funcionalidades pedidas:

- Reimplementar el parseo de los datasets originales con Pandas. No debe cambiar el resultado, solamente deben utilizar la librería para el parseo.
- Definición e implementación de sistema de puntajes para cada nivel.
- Terminar la implementación de ventana de puntajes mostrando los puntajes para cada nivel de dificultad.
- Realizar el registro de eventos cumpliendo como mínimo con la información solicitada en [Registro de jugadas y partidas](#).
- Realizar uno o más notebooks con todos los gráficos y cálculos solicitados.
- Terminar los detalles de interfaz que faltaron en la entrega anterior.
- Todas las funcionalidades del juego deben estar implementadas.

## Consideraciones generales de implementación

Debe continuar respetando las consideraciones generales de la entrega anterior.

El software debe correr correctamente tanto en Windows como en Linux.

Se deberá agregar al README.md lo que considere importante a agregar para esta segunda entrega. Tener en cuenta los nuevos notebooks agregados.

Recuerde:

- Utilizar Pandas para el análisis de datos.
- Generar los archivos CSV que se requiera.

- Documentar el código usando docstrings en las funciones y clases.
- Tener en cuenta las guías de estilo de Python para la escritura del código.
- Mantener una estructura de carpetas prolija.

## Consideraciones de la entrega y defensa

Se mantiene la misma metodología de defensa de la entrega pasada. Si bien el trabajo es grupal, **la defensa es INDIVIDUAL**.

La defensa se realiza dentro del horario de consulta inmediatamente posterior a la fecha de entrega. La misma consiste en tener un encuentro virtual o presencial con el ayudante asignado donde el ayudante va a realizar una serie de preguntas sobre la entrega. Estas preguntas serán destinadas a los distintos miembros del grupo. La idea es que se distribuyan las preguntas entre todos los miembros teniendo cada uno su espacio para responder. El desempeño en las respuestas de las mismas es el que define finalmente la nota de la entrega para cada integrante.

**Esto quiere decir que por más que sea una entrega grupal, las notas entre los miembros de los grupos pueden ser diferentes.**

Es importante también tener participación activa en las consultas prácticas y en el repositorio de GitLab para que el ayudante tenga un panorama conceptual del conocimiento de cada participante del grupo.

## Fechas de entrega

La fecha de entrega es la semana del **24 de Junio** y la **defensa individual** semana del **27 de Junio** en horario de la consulta práctica.

## Criterios de evaluación

- Funcionalidad implementada de acuerdo al enunciado.
- Cumplimiento de las consideraciones planteadas.
- Código subido al repositorio de GitLab indicado.
- Participación durante el desarrollo del trabajo. Esto incluye tanto la asistencia a las consultas (virtual o presencial) y las contribuciones en el repositorio de GitLab
- En la defensa deberán demostrar los conocimientos utilizados para la realización del trabajo.

## Puntos de la primera entrega (en la defensa individual)

- **Puntos de cursada:** 40 puntos.