

# Trabajo Integrador 2024

## Seminario de Lenguajes Opción Python

El trabajo integrador de este año consta del desarrollo de un juego educativo en formato de trivia. El desarrollo de este trabajo involucra el análisis de los datos que alimentarán al juego, el desarrollo de una interfaz gráfica simple para jugar y el análisis de los datos generados por el juego.

El desarrollo del trabajo se divide en 2 etapas que se entregarán de acuerdo al cronograma publicado.

Este documento describe el trabajo a realizar y luego, al final, detalla los requerimientos de cada entrega.

## PyTrivia

El juego a desarrollar implica la presentación de un conjunto de preguntas, a las cuales el jugador o jugadora deberá responder correctamente para acumular puntos. Las preguntas y respuestas se basarán en la información proporcionada por los *datasets* facilitados por la cátedra, dejando a discreción de cada grupo la elección y definición de las preguntas específicas.

La partida concluye cuando el jugador responde todas las preguntas de la ronda, momento en el cual se le informará su puntaje final y su posición en el ranking de puntajes históricos.

En la primera etapa nos enfocamos en los *datasets*, que son la base para generar las preguntas y validar las respuestas. En esta segunda etapa los esfuerzos se centrarán en el desarrollo de la interfaz visual y de la propia lógica del juego.

## Funcionalidades de la segunda etapa

Como previamente fue comentado, para la segunda etapa del trabajo debemos continuar con la implementación de la aplicación **Streamlit** comenzada en la etapa anterior (etapa 1).

La aplicación Streamlit contendrá al menos 5 páginas (o secciones), ellas son:

1. (P1) Inicio
2. (P2) Conociendo nuestros datos
3. (P3) Juego
4. (P4) Formulario de registro (implementada en etapa 1)
5. (P5) Ranking: agregamos una nueva sección donde se podrá observar el ranking de mejores resultados.

6. (P6) Sección de estadísticas: agregaremos una nueva sección de estadística de uso de la aplicación. En esta sección se agregarán consultas sobre la información de las personas que jugaron.

***Aclaración importante: la sección Datos pedida en la etapa 1 fue eliminada.***

A continuación se detalla la funcionalidad de cada página.

## Inicio (P1)

La página de Inicio será aquella que ve el usuario del sistema al ingresar a PyTrivia, es imprescindible diseñar un sitio sencillo que ofrezca información fundamental. La información a incluir es:

- Breve descripción del juego.
- Datos necesarios para comenzar a jugar.
- Instrucciones básicas para hacerlo comenzar a jugar.
- Explicación del funcionamiento del *parámetro* dificultad en el juego.

Cabe aclarar que se puede añadir otros detalles útiles para el jugador.

Asimismo, se requiere la creación de un menú o lista de enlaces que dirija a todas las secciones principales.

## Conociendo nuestros datos (P2)

En esta sección, se presentarán gráficos, tablas y mapas que brinden información útil o interesante extraída de los conjuntos de datos utilizados como fuente para el juego.

La idea es que generen al menos 3 widgets (gráfico, tabla o mapa) de los siguientes datasets:

- Aeropuertos:
  - Generar un mapa con las coordenadas de los aeropuertos diferenciando el color de los puntos según el atributo `elevation_name`
  - Pensar dos widgets más que tengan información útil de los aeropuertos
- Lagos:
  - Generar un mapa con puntos en las coordenadas de cada lago
  - Pensar dos widgets más que tengan información útil de los lagos

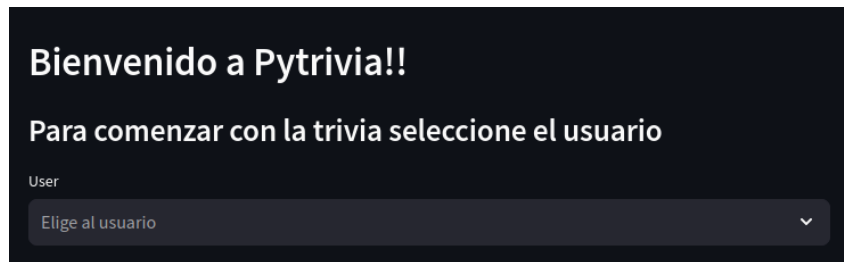
## Juego(P3)

Cómo fue previamente descrito, el juego consiste en una serie de preguntas que el jugador debe responder. La premisa básica es que, al iniciar el juego, se generará de manera aleatoria una lista de 5 preguntas para responder. Estas preguntas se derivan de la información contenida en los datasets analizados y procesados en la etapa 1.

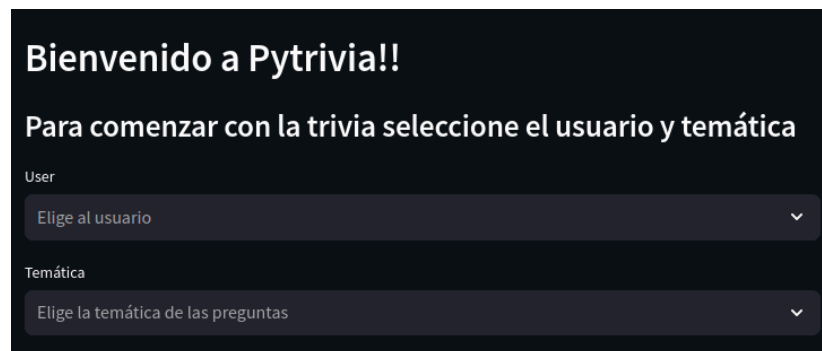
El objetivo del jugador es adivinar una característica específica obtenida de un atributo en una fila del conjunto de datos, utilizando el resto de las filas como información para hacer su suposición.

El flujo esperado es el siguiente:

1. La persona que ingresa al sistema selecciona el usuario que le corresponda. El sistema mostrará una lista con los usuarios del sistema y a su vez brindará la posibilidad de ir al registro de un nuevo usuario (P4).



2. El usuario selecciona una temática de 4 posibles: Aeropuertos, Lagos, Conectividad y Censo 2022.



3. El usuario selecciona una de 3 dificultades: Fácil, Media, Alta. Cada grupo decidirá cómo implementar estas dificultades. Existen muchas formas de implementarlo, a modo de ejemplo:
  - Dificultad alta: no se brindan ayudas.
  - Dificultad media: se le informa la cantidad de palabras de la respuesta.
  - Dificultad fácil: se le brindan tres respuestas de las cuales una es la correcta.



**Bienvenido a Pytrivia!!**

Para comenzar con la trivia seleccione el usuario, temática y dificultad

User  
Elige al usuario

Temática  
Elige la temática de las preguntas

Dificultad  
Elige la dificultad de las preguntas

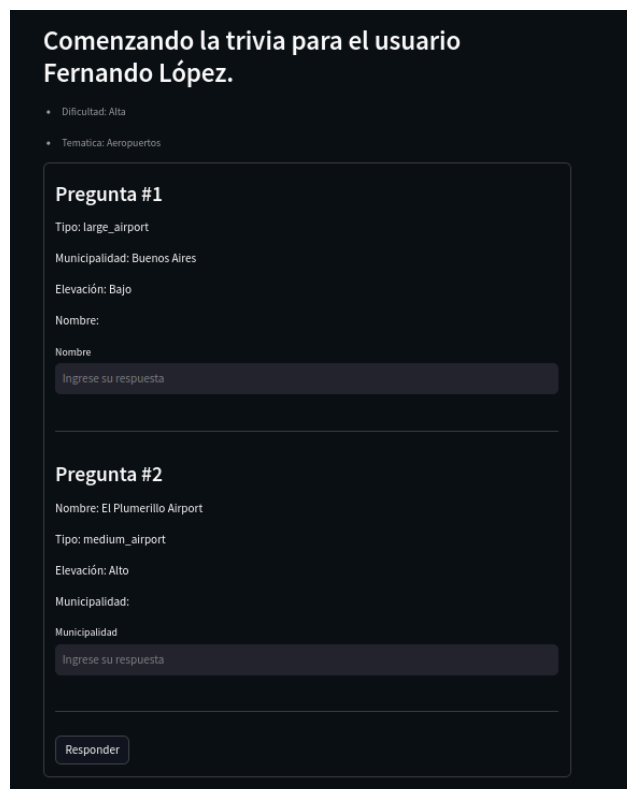
4. A partir de la temática elegida por el usuario se generan de manera aleatoria 5 preguntas.

En cada pregunta al usuario se le muestran 3 atributos de un lugar y debe responder cuál es el cuarto atributo.

Por ejemplo, si se elige la temática Aeropuerto, una de las preguntas generadas podría ser:

- Tipo: large\_airport.
- Municipalidad: Buenos Aires
- Elevación: Bajo
- Nombre: ??

En este caso el usuario debería responder: Jorge Newbery Airpark



**Comenzando la trivia para el usuario Fernando López.**

- Dificultad: Alta
- Temática: Aeropuertos

**Pregunta #1**

Tipo: large\_airport  
Municipalidad: Buenos Aires  
Elevación: Bajo  
Nombre:  
Nombre

Ingrese su respuesta

**Pregunta #2**

Nombre: El Plumerillo Airport  
Tipo: medium\_airport  
Elevación: Alto  
Municipalidad:  
Municipalidad

Ingrese su respuesta

Responder

5. El usuario responderá a las 5 preguntas y luego de ello se pasará a la página de ranking (P5) donde se podrá ver el ranking de mejores resultados (con la información del usuario de cada resultado), el resultado de la partida jugada y las preguntas junto a su respuesta correcta.

***Aclaración importante: todas las imágenes son a modo ilustrativo. La disposición y estética de los elementos de la interfaz queda a consideración y gusto de cada grupo con su respectivo ayudante.***

Características y consideraciones a tener en cuenta al momento de desarrollar el juego:

- Selección de **temática**: como previamente fue explicado, la temática establece de cierto modo el dataset del cual se generarán las preguntas. Esto no es una limitante, para una misma temática se pueden generar preguntas cruzando más de un dataset.
- Selección de **dificultad**: como previamente fue explicado, la dificultad establece qué ayudas recibirá el usuario al momento de jugar. Su implementación es libre.
- Tanto el **atributo** que se debe descubrir como aquellos que forman parte de la información que se da a conocer son **libres**. Sin embargo, existen atributos en los datasets que no son descriptivos o útiles para el propósito del juego. Se debe considerar mostrar atributos que realmente ayuden al usuario a responder

Por ejemplo en dataset de aeropuertos si los atributos mostrados son:

- iso\_country: Ar.
- iso\_region: AR-C
- country\_name: Argentina
- Nombre: ??

Es una tarea muy difícil descubrir el nombre del aeropuerto. Por tal motivo se debe desarrollar la **lógica necesaria para que solo ciertos atributos del dataset sean elegibles como parte de las “preguntas”**.

- Se debe almacenar en un archivo (formato y organización a elección de cada grupo) la información de cada partida. Es de interés conocer al menos:
  - Fecha y hora: momento en el cual se responde la trivia.
  - Usuario: identificador del usuario (mail).
  - Dificultad: dificultad elegida por el usuario.
  - Temática: temática elegida por el usuario.
  - Cantidad de respuestas correctas.
- Considerar que dados 3 atributos puede existir **más de una respuesta correcta**.
- Se valora si para una misma partida el atributo a descubrir no es siempre el mismo para todas las preguntas.
- Para obtener el resultado de una partida se debe sumar un punto por respuesta correcta (las incorrectas no restan ni suman).
- La dificultad NO solo establece las ayudas que recibirá el usuario al momento de jugar. También afecta el cálculo de puntaje de una partida:
  - Al resultado de un partida con dificultad baja no se le de aplicar ningún cambio.

- Al resultado de una partida con dificultad media se le debe sumar un 50%.
- Al resultado de una partida con dificultad alta se le debe sumar un 100%.

## Formulario de registro (P4)

Esta sección fue solicitada en la etapa 1, pero si se consideran mejoras o cambios, pueden llevarse a cabo en esta fase.

## Ranking (P5)

A esta sección se puede llegar de forma directa, es decir desde el *menú/sidebar/botones* implementados. En este caso se debe mostrar el ranking (primeros 15) de mejores resultados junto con el usuario asociado.

Otra forma de llegar es luego de haber terminado una partida, en este caso además del ranking de mejores resultados se debe mostrar el resultado de la partida:

- Preguntas de la partida en conjunto con la respuesta correcta y la respuesta del usuario.
- Cantidad de preguntas respondidas correctamente.
- Puntaje de la partida.
- Posición en el ranking.

### Finalizó su partida

- 01/05/2024 a las 14:50
- 1 respuesta correcta de 2
- Resultado final de 1 punto
- Su posición en el ranking histórico es 4 de 4 partidas

#### Pregunta #1

Tipo: large\_airport  
Municipalidad: Buenos Aires  
Elevación: Bajo  
Respuesta correcta para: Nombre

Jorge Newbery Airport

#### Pregunta #2

Nombre: El Plumerillo Airport  
Tipo: medium\_airport  
Elevación: Alto  
Respuesta correcta para: Municipalidad

Mendoza

Respuesta del usuario: Córdoba

### Ranking histórico de puntajes

Puesto	Usuario/a	Puntaje	Email
1	Sofia Martin	5	sofia@gmail.com
2	Federico Otaran	4.1	fedo@gmail.com
3	Facundo Diaz Gira	4	facu@gmail.com
4	Fernando López	1	fer@gmail.com

## Estadísticas (P6)

En esta sección se deben visualizar información estadística obtenida del uso del juego, el objetivo es utilizar la información almacenada de cada partida y de los usuarios del sistema para informar:

1. Gráfico de torta de usuarios que hayan jugado alguna vez el juego agrupados por género.
2. Gráfico de torta con porcentaje de partidas que tienen una puntuación superior a la media (promedio de calificaciones).
3. Gráfico de barras que para cada día de la semana muestre la cantidad de partidas realizadas.
4. Promedio de preguntas acertadas mensuales entre un rango de dos fechas insertadas en dos inputs.
5. Top 10 de usuarios con mayor cantidad de puntos acumulados entre un rango de dos fechas insertadas por input.
6. Ordenar los datasets por dificultad donde primero se debe ubicar el dataset que tiene mayor número de errores en las respuestas.
7. Gráfico de líneas que permita seleccionar dos usuarios para compararlos. Al seleccionarlos se debe mostrar la evolución de su puntaje a lo largo del tiempo.
8. Listar para cada género cuál es la temática en la cual demuestra mayor conocimiento.
9. Listar cada dificultad de juego junto con el puntaje promedio obtenido en cada una y con la cantidad de veces que fue elegida.
10. Listado de usuarios en racha. Lista los usuarios que registran una partida con un puntaje mayor a cero en todos los días durante los últimos 7 días.

## Consideraciones generales de implementación

El software implementado deberá funcionar correctamente tanto en Windows, Linux o Mac. Se deberá armar una estructura de directorios organizando los archivos en carpetas y subcarpetas de manera tal que mantengan el código organizado haciendo que sea fácil de actualizar y de corregir. Se evaluará tanto el código como su organización.

El código se deberá subir a un repositorio de GitLab designado por la cátedra. En este repositorio, cada equipo deberá incluir un archivo denominado **README.md** que contenga el nombre de los integrantes del equipo y las instrucciones para ejecutar cada aplicación. Además, se solicitará un archivo que enumere las bibliotecas necesarias, siguiendo la metodología explicada en las prácticas para este propósito.

Para el desarrollo del código se deberá cumplir con los siguientes requerimientos:

- En esta segunda etapa del trabajo integrador **deben utilizar la librería de Python pandas**.
- Usar la herramienta **Streamlit** para el desarrollo de la interfaz gráfica.
- Generar los *datasets* pre-procesados necesarios.

- Documentar el código usando docstrings en funciones y clases.
- Incluir un archivo **LICENSE** con la licencia del código.
- Se debe tener en cuenta las [guías de estilo de Python](#) para la escritura de código.
- Definir una estructura de carpetas que permita estructurar el código de forma prolija.

## Consideraciones de la entrega y defensa

Si bien el trabajo es grupal, **la nota de la defensa es INDIVIDUAL**.

La defensa se lleva a cabo durante el horario de consulta inmediatamente posterior a la fecha de entrega. Este proceso implica la realización de un encuentro virtual o presencial con el ayudante asignado, durante el cual se formulan una serie de preguntas relacionadas con la entrega. Dichas preguntas están dirigidas a los distintos miembros del grupo, distribuyendo equitativamente la participación entre todos ellos. La finalidad es que cada integrante tenga la oportunidad de responder. La evaluación se basa en el desempeño de las respuestas, lo que determina finalmente la nota asignada a cada miembro del grupo en relación con la entrega realizada.

**Esto implica que, aunque se trate de una entrega grupal, las calificaciones entre los miembros del grupo pueden variar.**

Tener en cuenta que es fundamental mantener una participación activa tanto en las consultas prácticas como en el repositorio de GitLab. Esto permitirá al ayudante obtener una comprensión conceptual del conocimiento de cada participante del grupo.

## Detalles de entrega

La fecha de entrega de la primera etapa es el día **19 de junio a las 23:59 hs.** **La defensa será la semana del 24 de junio en horario de la consulta práctica.**

La segunda entrega puede otorgar al estudiante un máximo de **40 puntos**, los cuales se basan en su desempeño durante la defensa.

## Criterios de Evaluación

- Funcionalidad implementada de acuerdo al enunciado.
- Cumplimiento de las consideraciones planteadas.
- Código subido en tiempo y forma al repositorio de GitLab indicado.
- Participación activa durante el desarrollo del trabajo, que incluye asistencia a las consultas (tanto virtuales como presenciales) y contribuciones al repositorio de GitLab.



- En la defensa, se espera que demuestren los conocimientos utilizados para la realización del trabajo.