

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. Además de la sección principal donde se lleva a cabo el juego, habrá otras secciones necesarias para su funcionamiento, las cuales definiremos más adelante y serán implementadas en la segunda etapa del proyecto.

En esta fase inicial nos enfocaremos en los *datasets*, que serán la base para generar las preguntas y validar las respuestas. Estos *datasets* requerirán un procesamiento previo para simplificar la construcción del juego.

Funcionalidades de la primera etapa

En esta primera parte del trabajo, se deben cumplir dos objetivos principales:

1. Procesar los conjuntos de datos originales para dejarlos listos para su uso en la implementación del juego.
2. Iniciar el proyecto **Streamlit**, que servirá de base para el juego y se continuará en la segunda parte del trabajo integrador.

Procesamiento de los datasets

- Se requiere la creación de uno o más Jupyter Notebook que contenga las operaciones destinadas al procesamiento, análisis y limpieza de los conjuntos de datos originales (**cada uno con sus respectivos comentarios**).
- Se deberá desarrollar una función para cada dataset. Estas funciones deberán recibir la ruta completa del dataset original como primer argumento y la ubicación donde se almacenará el dataset procesado como segundo argumento.

Datasets

- (A) Aeropuertos de Argentina¹: *ar-airports.csv*
- (B) Conectividad en Argentina²: *Conectividad_Internet.csv*
- (C) Lagos en Argentina³: *lagos_arg.csv*
- (D) Población del censo 2022 sobre el total de cada Provincia⁴: *c2022_tp_c_resumen_adaptado.csv*
- (E) Información de la composición de cada provincia con el listado de ciudades de cada una⁵: *ar.csv*

Procesamiento

1. En el dataset de Aeropuertos (A), se debe generar una columna llamada **'elevation_name'** que contenga datos cualitativos indicando la elevación de cada aeropuerto. Esta columna se completará con las palabras: "bajo", "medio" o "alto", según la elevación de cada aeropuerto. Para definir estos grupos, se utilizarán los siguientes valores:
 - a. bajo: aeropuertos con elevación menor o igual a 131 ft.
 - b. medio: aeropuertos con elevación mayor que 131 ft y menor o igual a 903 ft.
 - c. altos: aeropuertos con elevación mayor a 903 ft.

Además, se realizará otra modificación en este dataset, consistente en agregar una columna llamada **'prov_name'**, donde se incluirá el nombre de la provincia correspondiente a cada aeropuerto. Esta información se obtendrá consultando los nombres de las ciudades en el dataset (E).

2. En el dataset de Conectividad (B), se **realizará un reemplazo** en las celdas que contengan el carácter '-' con la palabra 'NO'.

Para el mismo dataset se debe generar una nueva columna denominada **'posee_conectividad'**, la misma puede tomar uno de dos valores posibles: SÍ o NO.

¹ Aeropuertos en Argentina: <https://data.humdata.org/dataset/ourairports-arg>

² Conectividad en Argentina: https://datos.gob.ar/dataset/enacom-conectividad-al-servicio-internet/archivo/enacom_CONEC-AL-SE-RVI-DE-INTER_0

³ Lagos Argentina: <https://www.ign.gob.ar/NuestrasActividades/Geografia/DatosArgentina/Lagos>

⁴ Censo 2022 Argentina: https://censo.gob.ar/index.php/datos_definitivos_total_pais/

⁵ Listado de ciudades de Argentina con localización <https://simplemaps.com/data/ar-cities>

El valor será NO si todos los campos ADSL, CABLEMODEM, DIALUP, FIBRAOPTICA, SATELITAL, WIRELESS, TELEFONIAFIJA, 3G y 4G poseen el valor --. Caso contrario el valor será SI

3. En el dataset de Lagos (C), se creará una nueva columna llamada '**Sup Tamaño**' que contendrá datos cualitativos indicando el tamaño de cada lago en función de su superficie en kilómetros cuadrados (km²). Esta columna se completará con las palabras: "chico", "medio" o "grande", según los siguientes criterios:
 - a. Lagos con una superficie menor o igual a 17 km² serán clasificados como "chico".
 - b. Lagos con una superficie mayor que 17 km² y menor o igual a 59 km² serán clasificados como "medio".
 - c. Lagos con una superficie mayor a 59 km² serán clasificados como "grande".

Además de transformar el campo de coordenadas actual, que sigue el formato estándar de grados, minutos y segundos (GMS), por ejemplo, 42°9'3"S 71°38'59"O, en dos campos separados para la latitud y longitud, añade dos nuevos campos para representar la latitud y longitud en formato de grados decimales (GD). Asegúrate de proporcionar la conversión correcta de GMS a GD para ambos campos. En el ejemplo mencionado el valor resultante debería ser -42.150833 para la latitud y -71.649722 para la longitud.

4. Para el dataset de Población del censo 2022 (D) reemplazar los valores "///" y "-" por cero en los campos que corresponda.

Además agregar un nuevo campo que tenga el porcentaje de población en situación de calle. Tener en cuenta el total general (NO tener en cuenta los totales por sexo registrado al nacer).

Consulta

Consultando estos datasets ya modificados en el punto anterior, se debe poder informar utilizando Jupyter notebook:

1. Los diferentes tipos de aeropuertos según la columna '**type**'.
2. Los aeropuertos de una elevación, (bajo, medio, alto) cuya especificación se permita modificar fácilmente, evaluando la columna creada '**elevation_name**'.
3. Los aeropuertos que tienen una mayor o menor elevación con respecto al valor numérico dado.
4. Los aeropuertos, lagos y tipo de conectividad en provincias con población mayor o menor a un valor que se pueda especificar fácilmente.
5. Mostrar los aeropuertos en las capitales de cada provincia.
6. Mostrar los Lagos de una superficie según la columna '**Sup Tamaño**' donde el criterio (chico, medio, grande) se puede indicar fácilmente.
7. Listar las 5 Jurisdicciones con mayor porcentaje de población en situación de calle. Mostrar el nombre de la jurisdicción y el porcentaje.

8. Obtener la jurisdicción donde la brecha entre las poblaciones diferenciadas según sexo registrado al nacer es mayor. Informar nombra de Jurisdicción y cantidad de la brecha poblacional.
9. Mostrar los diferentes tipos de conectividades.
10. Mostrar las cantidades de localidades con cada tipo de conectividad.
11. Mostrar las provincias para las cuales todas sus ciudades poseen FIBRA ÓPTICA.
12. Mostrar para cada provincia su capital y, si se conoce la información para dicha capital, informar si posee conectividad (campo '**posee_conectividad**' creado previamente). En caso de no conocer la información mostrar el texto "conectividad desconocida".

Ayuda: se recomienda leer la documentación del dataset Listado de ciudades de Argentina con localización (<https://simplemaps.com/data/ar-cities>)

Aclaración importante: En los incisos 2, 3, 4 y 6 se espera que el usuario del jupyternotebook sea capaz de ingresar el parámetro de búsqueda que desee (elevación ordinal, elevación cuantitativa, población y tamaño ordinal respectivamente). Es decir, no debe encontrarse fijado el parámetro de búsqueda en el código.

Comenzar aplicación Streamlit

Se solicita que generen un menú en el sidebar con las siguientes páginas:

- (P1)Inicio
- (P2) Datos
- (P3) Juego
- (P4) Formulario de registro

En las páginas P1 P2 y P3 no es necesario agregar funcionalidad, sólo los datos de los integrantes en la página (P1).

La página P4 consiste en generar un formulario capaz de pedir al usuario o usuaria que ingrese la información necesaria para registrarse en el sistema. La misma es:

- Nombre de usuario
- Nombre Completo
- Mail
- Fecha de nacimiento
- Género

Una vez completados todos los campos (todos los atributos son obligatorios) se debe permitir a la persona registrarse y su información se debe almacenar en un archivo. Con cada nuevo registro se debe ir actualizando el mencionado archivo.

Es importante destacar que si ya se encuentra registrado el usuario o usuaria con el mismo mail se deben actualizar el resto de datos y NO agregar un nuevo usuario.

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 primera etapa del trabajo integrador NO se puede 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 **3 de mayo a las 23:59 hs.** La defensa será la semana del **6 de mayo en horario de la consulta práctica.**

La primera entrega puede otorgar al estudiante un máximo de **35 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.