

Solemne 3:

Visualización Interactiva de Datos en Tiempo Real con Streamlit y API REST



Descripción General

En el desarrollo de la solemne 3, los estudiantes aplicarán sus conocimientos de Python para desarrollar un programa que interactúe con una API REST pública. Utilizando las bibliotecas **requests**, **json**, **pandas**, **streamlit**, entre otras, los estudiantes seleccionarán, analizarán y presentarán datos obtenidos de una fuente pública en Internet entregada por el docente y seleccionarán el conjunto de datos, como por ejemplo datos climáticos, económicos, de salud, etc.

Objetivos Generales

1. **Integración y Aplicación de Conocimientos:** Aplicar los conocimientos adquiridos durante el curso en un contexto práctico y real.
2. **Interacción con APIs REST:** Demostrar habilidad en la utilización de APIs REST para la obtención de datos.
3. **Análisis de Datos:** Utilizar Python para analizar y procesar datos.
4. **Presentación de Resultados – Aplicación WEB:** Presentar los datos de manera efectiva y comprensible con el apoyo de gráficos que permitan la visualización de su análisis y conclusiones mediante gráficos y presentaciones desarrolladas mediante streamlit.

Requisitos Técnicos

1. **Lenguaje de Programación:** Python.
2. **Librerías Permitidas:** **requests** para la interacción con APIs, **json** para el manejo de datos en formato JSON, **pandas** para el análisis y presentación de datos, **matplotlib** para graficar resultados y series de datos, **streamlit** para desarrollar la aplicación web final.
3. **Fuente de Datos:** Los datos deben obtenerse a través de la API pública REST Countries (<https://restcountries.com/v3.1/all>). Esta API proporciona información detallada sobre países de todo el mundo, incluyendo variables como población, área, densidad, regiones, idiomas y más. La recolección debe realizarse exclusivamente mediante consultas utilizando el método GET.

Instrucciones Generales

1. Esta solemne se desarrollará de forma grupal, en equipos de 2 a 3 personas. Se mantendrán los mismos grupos formados para la solemne 2, sin posibilidad de realizar cambios. Cualquier inconveniente con la participación de un integrante en las tareas asignadas debe ser informado al docente a la brevedad. No se aceptarán notificaciones de última hora.
2. La evaluación del desempeño de los integrantes del grupo será realizada por el docente durante cada sesión de clase, las cuales estarán destinadas al desarrollo en equipo de la solemne 3. Por lo tanto, la ausencia a estas sesiones implicará una evaluación negativa en la parte correspondiente al trabajo en equipo.
3. El día de entrega de la tarea, se llevará a cabo una presentación oral del trabajo realizado. La presentación deberá tener una duración máxima de 10 minutos y cada integrante del equipo deberá presentar una parte del trabajo

Instrucciones para el desarrollo de la Página Web

Objetivo

- Crear una aplicación web interactiva para visualizar resultados de análisis de datos utilizando Streamlit.

Requisitos obligatorios que debe cumplir su aplicación

- **Uso de Streamlit:** Emplear Streamlit para diseñar la interfaz de usuario y mostrar gráficos interactivos.
- **Método GET:** Utilizar el método GET para recolectar los datos mediante consultas exclusivamente a la API REST Countries (<https://restcountries.com/v3.1/all>). Además, investigar cómo extraer las siguientes variables del JSON proporcionado por la API: nombre común del país, región geográfica, población total, área en kilómetros cuadrados, número de países con frontera, número de idiomas oficiales y número de zonas horarias asociadas a un país.
- **Estructura de la Aplicación:** Investigar cómo crear una aplicación web multipágina utilizando Streamlit, con al menos tres páginas. En cada una de ellas, debe presentarse la siguiente información:
 1. Descripción del proyecto: Proporcione una breve explicación de la base de datos consultada para la creación de su proyecto. Incluya títulos, subtítulos y enlaces cuando sea necesario.
 2. Interacción con los datos: Incluir funcionalidades como (utilizando objetos de Streamlit como `st.selectbox`, `st.slider`, `st.file_uploader`, y `st.button`):
 - Mostrar los datos originales.
 - Seleccionar una columna específica del dataframe con un menú desplegable y luego calcular la media, mediana, desviación estándar de la columna seleccionada.
 - Selecciona una columna del dataframe utilizando un menú desplegable. Luego, incluye controles para ordenar los datos en forma ascendente o descendente según la columna seleccionada. Finalmente, muestra el dataframe ordenado.
 - Filtrar las filas en función de valores en una columna numérica (en este caso elija de forma arbitraria una columna).
 - Agregar un botón para descargar en CSV o Excel los datos filtrados en el paso anterior.
 3. Diseña e implementa una sección de gráficos interactivos en tu aplicación utilizando Streamlit. Asegúrate de incluir al menos tres tipos diferentes de gráficos (por ejemplo, histograma, barra, línea, dispersión, pastel). Tu implementación debe cumplir con las siguientes funcionalidades mínimas:

- **Selección de Variables:** Permite al usuario elegir columnas numéricas de la base de datos que se utilizarán para los ejes X e Y, ubicando este control en un lugar que permita una fácil configuración de los gráficos.
- **Rango Personalizado para Ejes:** Incluye controles que permitan al usuario ajustar dinámicamente los límites de los ejes X e Y en función de los valores mínimos y máximos de las variables seleccionadas.
- **Cambio del Tipo de Gráfico:** Implementa una forma de alternar entre al menos tres tipos de gráficos diferentes, como dispersión, línea, barra, histograma o pastel.
- **Descargar el Gráfico en PNG:** Proporciona una opción para descargar el gráfico generado en formato PNG.

Entregables

1. **Código Fuente:** Archivos de Python con el código desarrollado.
2. **Aplicación Streamlit:** Un enlace a la aplicación desplegada (por ejemplo, utilizando Streamlit sharing o cualquier otro servicio de hosting adecuado) o instrucciones detalladas para ejecutar la aplicación localmente.
3. **Informe del trabajo desarrollado**

El informe debe ser entregado en formato Word e incluir los siguientes apartados:

- **Portada del Trabajo:** Debe contener el título del trabajo, la asignatura, los nombres de los integrantes, la fecha y el nombre del docente.
- **Planteamiento del Problema y Objetivo:** Breve descripción del caso asignado, incluyendo el planteamiento del problema y el objetivo principal del trabajo.
- **Definición teórica del análisis a realizar:** Incluya una descripción del archivo de datos utilizado, destacando las principales variables empleadas. Además, explique las funciones clave de Python que se usaron, especialmente aquellas que no se abordaron en clase.
- **Programa en Python desarrollado:** Inserte imágenes del código desarrollado en Python con una estructura de tipo pregunta-respuesta. Cada imagen debe tener un número, un pie de figura descriptivo, y todas deben tener el mismo tamaño.
- **Resultados:** Presente imágenes que muestren las distintas funcionalidades implementadas en la aplicación web. Cada figura debe incluir un número, un pie de figura descriptivo y respetar un tamaño uniforme.
- **Conclusiones:** Resuma la principal contribución del trabajo, los desafíos enfrentados, las limitaciones encontradas y las posibles líneas de trabajo futuro.
- **Referencias bibliográficas:** Citadas en un formato estandarizado (por ejemplo, APA, IEEE o Vancouver), que respalden la información utilizada, las metodologías aplicadas y cualquier herramienta externa empleada.

Fechas importantes

- **Inicio del Proyecto:** jueves 14 de noviembre.
- **Revisión Intermedia:** 21 de noviembre en horario de clases – Hito de control para revisar el progreso y definición de mejoras.
- **Entrega Final:** 28 de noviembre, incluyendo todos los entregables.

Recursos Adicionales

- Documentación en línea de las librerías **requests**, **json**, **pandas**, **streamlit** y **matplotlib**.
- Ejemplos de APIs REST públicas y streamlit desarrollados en clases dentro de la unidad 3, disponibles en el blackboard.

Apoyo y Consultas

Las consultas sobre el proyecto pueden realizarse durante las horas de avance de los proyectos durante las jornadas de clases con el docente instructor o a través de la plataforma del curso.

Igualmente, se incentiva a los estudiantes a realizar avances en el laboratorio Make iT de la FIAD, en cualquier horario distinto al reservado al curso dentro de su disponibilidad.

Criterios de Evaluación

1. **Calidad del Código:** Claridad, organización y eficiencia del código.
2. **Funcionalidad:** El programa debe ejecutarse sin errores y cumplir con los objetivos propuestos.
3. **Documentación:** Incluir documentación adecuada del código y un informe que explique el desafío, el conjunto de datos seleccionado, el proceso desarrollado (código Python y desarrollo de la aplicación) y los hallazgos y aprendizajes obtenidos.
4. **Análisis de Datos:** Profundidad y relevancia del análisis de datos realizado.
5. **Diseño de la Aplicación:** Calidad del diseño de la interfaz de usuario, enfatizando la usabilidad, estética y experiencia general del usuario. Se valorará especialmente la implementación de funcionalidades interactivas y un diseño intuitivo.
6. **Evaluación del Docente:** Valoración individual de la contribución de cada integrante al desarrollo del trabajo en equipo, considerando su participación activa, aportes técnicos y compromiso durante el desarrollo del proyecto.
7. **Presentación Oral:** Evaluación de la presentación del trabajo por parte de cada integrante, enfocándose en la claridad, estructuración de ideas, dominio del tema y capacidad de responder preguntas. Se valorará el uso de recursos visuales efectivos y el manejo del tiempo.
8. **Informe del Trabajo Desarrollado:** El informe final debe ser claro, organizado y cumplir con los siguientes requisitos solicitados en la sección de entregables.