

**Universidad de Costa Rica**

**Herramientas de Ciencia de Datos II**

**CA – 0305**

**Proyecto Individual**

**Desarrollo de Gráficos Interactivos con  
Plotly.express e Integración en Dash**

**Eyeri Méndez Méndez – C24765**

**Profesor: Luis Alberto Juárez Potoy**

**Escuela de Matemática**

**I Semestre – 2024**

## 1. Introducción

En el ámbito de la Ciencia de Datos, la visualización de información desempeña un papel importante en la comprensión y comunicación de patrones, tendencias y relaciones dentro de los datos. En este proyecto, se explorará la utilidad de Plotly Express, una biblioteca de Python especializada en la creación de gráficos interactivos, para la generación de visualizaciones dinámicas y atractivas. Este documento detalla el proceso de utilizar Plotly Express para desarrollar una variedad de gráficos interactivos, los cuales posteriormente se integrarán en una aplicación utilizando Dash, un marco de trabajo para crear aplicaciones web interactivas con visualizaciones de datos.

El objetivo principal es mostrar cómo Plotly Express simplifica la creación de gráficos complejos mediante una sintaxis concisa y accesible, permitiendo a los usuarios explorar y analizar datos de manera intuitiva. A través de ejemplos prácticos y explicaciones detalladas, se destacará la versatilidad de Plotly Express para visualizar datos en histogramas, gráficos de dispersión, de barras, de líneas y más, todos ellos interactuando con la capacidad de Dash para construir interfaces de usuario interactivas.

De esta manera, este proyecto no solo busca presentar una serie de visualizaciones estáticas, sino también demostrar cómo estas pueden mejorarse significativamente con elementos interactivos que permitan explorar diferentes aspectos y detalles de los datos con facilidad. Al finalizar, se espera haber proporcionado un recurso de utilidad para utilizar Plotly Express en la creación de gráficos interactivos y su práctica integración en aplicaciones web utilizando Dash.

## 2. Desarrollo

El proyecto se divide en dos componentes principales: un módulo de Python hecho en Spyder llamado 'GenerarGraficos' y un archivo de Python en Jupyter Notebook en el que se creará una aplicación Dash en donde estarán los gráficos interactivos, haciendo uso del módulo mencionado anteriormente.

En el módulo 'GenerarGraficos', se define una clase que interactúa con la biblioteca Plotly Express para la creación de gráficos interactivos. Esta clase se configura para recibir una instancia de una aplicación Dash como parámetro en su constructor. Los métodos de la clase, tales como 'barras', 'histograma', 'lineas', 'dispersion', o 'cajas', se encargan de generar gráficos específicos (como gráficos de barras, de dispersión, de líneas, histogramas, etc.) a

partir de un DataFrame proporcionado como entrada. Estos gráficos se diseñan con capacidades interactivas utilizando widgets Dash, como listas desplegables, cajas de texto y controles deslizantes, que permiten modificar aspectos visuales o parámetros estadísticos de los gráficos.

Estos widgets van a permitir al usuario realizar diferentes manipulaciones en el gráfico en cuestión, tales como cambiar el color, imputar los valores nulos por el promedio, la mediana, el máximo o el mínimo, agrupar por cierta categoría, cambiar el título o, en general, mostrar diferentes características del gráfico interactivo.

En este sentido, dado que cada gráfico se puede crear con diferentes parámetros usando `plotly.express`, los métodos pueden recibir distintos tipos de estos. Por dar una idea, algunos gráficos pueden recibir o no un parámetro para el eje y del gráfico (una columna del DataFrame dado), por lo que, en caso de que el método no recibiera un parámetro para esto, se crearía un gráfico de frecuencia, en el caso de los gráficos de barras o histogramas.

Lo anterior también es posible con un widget creado en la aplicación Dash que se está trabajando, por lo que, en el caso mencionado de gráficos de barras o histogramas, es posible dar una variable a agrupar y luego en la aplicación dar clic en el widget y mostrar los mismos gráficos, pero con frecuencias en el eje y. También es posible, en algunos casos, crear gráficos interactivos sin una variable a agrupar, por lo que varios métodos pueden no recibir este parámetro y crear el gráfico sin una agrupación, o bien, dar clic en un widget y generar el gráfico sin una agrupación, a pesar de haber dado una variable para agrupar.

Asimismo, en la mayoría de los métodos es posible dar listas como parámetros con los nombres de los ejes, de las leyendas (en caso de agrupar), de las categorías de una variable a agrupar o de columnas numéricas del DataFrame dado, con el objetivo de poder personalizar las etiquetas del gráfico interactivo correspondiente al método. Todo lo mencionado anteriormente se puede extender a más widgets y más casos en particular, pero como se mencionó anteriormente, cada gráfico interactivo es diferente a la hora de crearlo.

Por otro lado, en el notebook de Jupyter, se carga y preprocesa varios DataFrames y conjuntos de datos utilizando bibliotecas como Pandas y Seaborn. Posteriormente, se crea una aplicación Dash, la cual sirve como lienzo para los gráficos interactivos generados por la instancia de la clase `GenerarGráficos`. Cada método de esta clase se invoca con los diferentes DataFrames mencionados para añadir gráficos y widgets a la interfaz de la aplicación Dash, permitiendo a los usuarios explorar y analizar los datos de manera dinámica y personalizada.

Este enfoque facilita la integración de visualizaciones complejas y adaptativas dentro de aplicaciones web interactivas, proporcionando herramientas poderosas para la exploración y comunicación efectiva de patrones y relaciones en conjuntos de datos complejos.

### **3. Resultados**

La creación de una aplicación Dash totalmente funcional que integre varios tipos de gráficos interactivos generados dinámicamente a partir de datos proporcionados puede ser significativo desde el punto de vista práctico, ya que esta aplicación podría ser utilizada como una herramienta flexible para explorar y visualizar datos de manera interactiva.

En este sentido, los múltiples gráficos que se pueden generar con los métodos del módulo 'GenerarGraficos' estarán personalizados con widgets interactivos, lo que permite a los usuarios ajustar parámetros visuales como títulos, colores, escalas, imputación y/o seleccionar datos específicos para visualizar, lo que facilita el análisis exploratorio de datos. De esta manera, los usuarios podrán explorar y analizar datos complejos de manera más efectiva debido a la capacidad de interactuar directamente con los gráficos. Esto incluye la capacidad de filtrar datos, ajustar la visualización y obtener detalles específicos al pasar el cursor sobre puntos o barras en los gráficos.

De esta manera, la combinación de Plotly Express y Dash facilita la creación de un entorno interactivo con visualizaciones de datos avanzadas que pueden ayudar en la comprensión de patrones, tendencias y relaciones ocultas dentro de conjuntos de datos grandes y complejos. Además, la inclusión de widgets como listas desplegables, cajas de texto y controles deslizantes no solo mejora la interactividad de los gráficos, sino que también mejora la usabilidad general de la aplicación, permitiendo a los usuarios personalizar su experiencia de visualización de múltiples DataFrames.

En síntesis, el presente proyecto incluye como resultados una aplicación web interactiva con capacidades avanzadas de visualización de datos, que mejoran la accesibilidad y el análisis de información compleja a través de gráficos interactivos y herramientas de personalización.

### **4. Recomendaciones**

Dado que no fue posible guardar o compartir la aplicación Dash con los gráficos interactivos creados usando los métodos de la clase 'GenerarGraficos', se recomienda implementar una solución alternativa para guardar o exportar los gráficos cuando sea necesario. Una opción podría ser capturar las configuraciones actuales de los gráficos y datos, y permitir la exportación a formatos como imágenes estáticas o archivos HTML independientes utilizando las funciones de Plotly. Otra alternativa a este problema podría ser crear un archivo ejecutable (.exe) con la aplicación Dash creada con los gráficos deseados, permitiendo que cualquier persona pueda abrir la app y visualizar dichos gráficos.

Debido a la restricción de Plotly Express en cuanto a la generación de gráficos con IDs únicos, es recomendable asegurar que cada método de creación de gráficos en la clase 'GenerarGraficos' no sea invocado más de una vez por cada instancia de la aplicación Dash, ya que esto causa errores y problemas. Esto implica estructurar el flujo de creación de gráficos de manera que se evite la duplicación inadvertida de IDs de gráficos, lo cual podría afectar la integridad y el funcionamiento de la aplicación.

Por último, si se crea una aplicación Dash y se le añaden gráficos interactivos usando la clase 'GenerarGraficos', y posteriormente se crea otra aplicación Dash nueva a la cual también se le agregan gráficos con la misma clase, entonces la primera aplicación creada se actualiza a la segunda; es decir, no es posible tener dos aplicaciones Dash abiertas con diferentes gráficos interactivos. Es por esto por lo que se debe considerar la gestión cuidadosa de las instancias de la clase y las aplicaciones Dash asociadas. Esto implica diseñar la aplicación para que sea consciente del contexto de la aplicación actual y asegurar que cada instancia de 'GenerarGraficos' se utilice de manera aislada o con métodos de control adecuados para manejar múltiples instancias.

## 5. Anexos

Link al GitHub: [https://github.com/Eyeri1120/Proyecto\\_individual\\_CA0305](https://github.com/Eyeri1120/Proyecto_individual_CA0305)