



## 3 - Lab. Data visualisation

### Tareas:

1. Para el framework de visualización desarrollado en la práctica anterior, incorpora las modificaciones planteadas por el profesor durante la sesión de evaluación.
2. Analiza y familiarízate con la librería [Seaborn](#).
3. Incorpora al framework que has desarrollado la posibilidad de generar diagramas que permitan mostrar distribuciones de datos: histogramas, correlogramas, boxplots, curvas de densidad, diagramas de violín, ...
4. Con los conjuntos de datos considerados en las prácticas anteriores desarrolla el tratamiento que permita conocer la distribución de datos y extraer información relativa a comportamientos realizando comparativas de algunas variables: edad, país, género, periodos, ...
5. Haciendo uso del componente "servidor" desarrollado en la asignatura Computación en la Nube, convierte en un servicio el framework desarrollado. Para poder acceder a este servicio, deberías desarrollar asimismo un cliente específico que permita al usuario seleccionar el origen de los datos que van a ser representados y que en el servidor se realicen las representaciones que serán mostradas por el cliente. Describe en una página la arquitectura software de la aplicación y las tecnologías desarrolladas.
6. Piensa en un nombre y en algún tipo de aplicación para la que pueda desarrollarse el Framework desarrollado.
7. Desarrolla un documento con 3 transparencias donde recojas los puntos anteriores de esta práctica para realizar una presentación de 10-15 minutos.
8. Desarrolla las tareas asociadas a los "[Ejemplos SciKit-Learn - I](#)" y "[Ejemplos SciKit-Learn - II](#)"

### Working on

#### Incorpora las modificaciones planteadas por el profesor

- Mejoras

#### El rendimiento

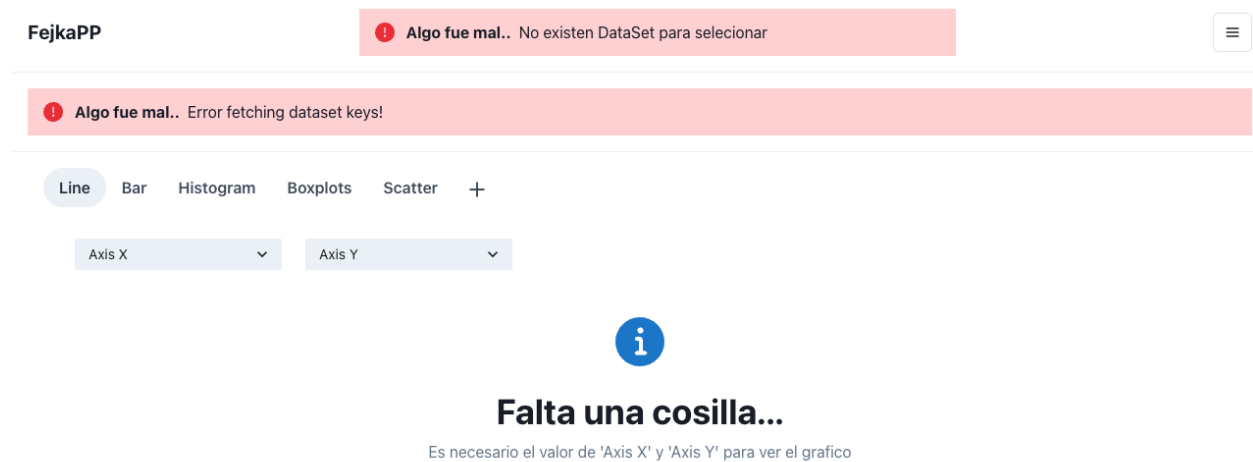
Los gráficos en la ultima version no eran muy buenos y por esa razón se intento pivotar a otras soluciones. Luego que probar algunas alternativas `react google chart` ha sido por mucho la mejor experiencia desarrollador y rendimiento. Ademas de poder personalizar los ejes X y Y. y tener una amplia gama de componentes usando el patron estrategia.

Ejemplo:

```
<Chart
  chartType="Bar"
  width="100%"
  height="400px"
  data={transformDataToGoogleDataTable(data)}
  options={options}
  isStacked={true}
/>
```

Donde `Chart` es la clase "Estrategia" donde se define todo el container, controla las opciones y el parámetro `chartType` se usa para determinar la estrategia (Chart) en concreto que se quiere pintar. es Decir si pasamos el parámetro `chartType = Bar` pintara un grafico de barra pero si el valor es `Line` va pintar un gráfico de lines o si es `Geo` va pintar un mapa.

**Feedback si algo va mal cargando el conjunto de datos.**



## Analiza y familiarízate con la librería Seaborn.

Para esta parte pense que lo mejor era explorar un conjunto de datos usando Seaborn.

He realizado estas dos gráficas siguiente los pasos del tutorial en Kaggle de "Seaborn Tutorial for Beginners"

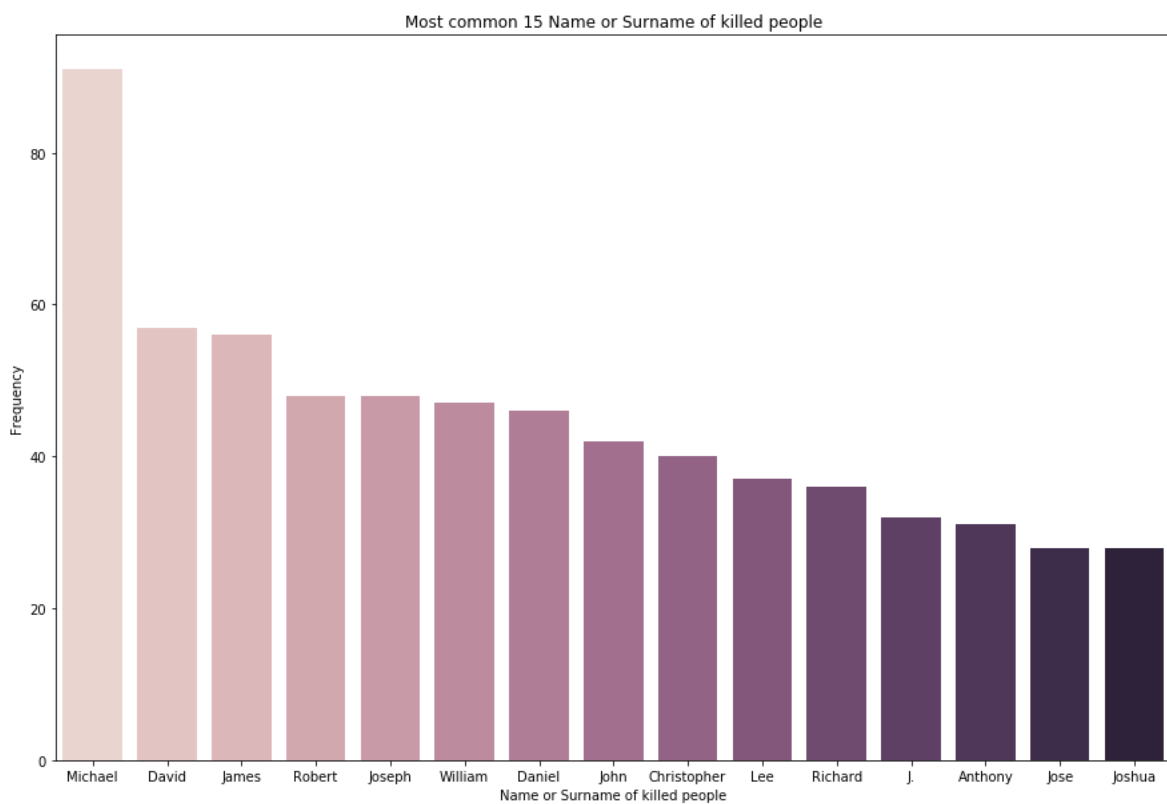
```
import seaborn as sns
```

```

separate = kill.name[kill.name != 'TK TK'].str.split()
a,b = zip(*separate)
name_list = a+b
#Contador de apellidos
name_count = Counter(name_list)
most_common_names = name_count.most_common(15)
#Calculamos la ubicación
x,y = zip(*most_common_names)
x,y = list(x),list(y)

#Tamaño de la figura
plt.figure(figsize=(15,10))
# Creamos un bar plot
ax= sns.barplot(x=x, y=y,palette = sns.cubehelix_palette(len(x)))
plt.xlabel('Name or Surname of killed people')
plt.ylabel('Frequency')
plt.title('Most common 15 Name or Surname of killed people')

```



```

f,ax = plt.subplots(figsize=(5, 5))
sns.heatmap(data.corr(), annot=True, linewidths=0.5, linecolor="red", fmt= '.1f',ax=ax)
plt.show()

```

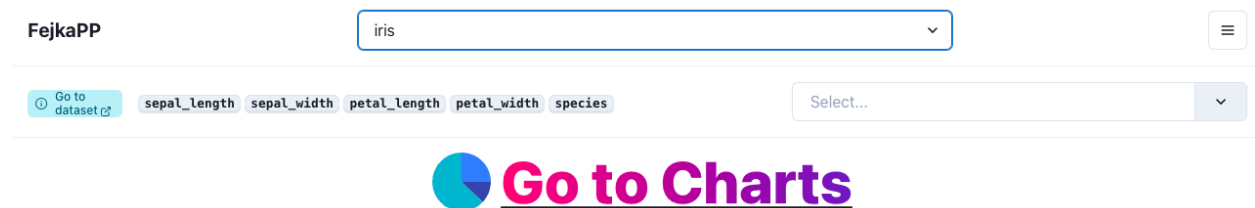
Correlación de los datos (heatmap)



**Incorpora la posibilidad de generar diagramas que permitan mostrar distribuciones de datos: histogramas, correlogramas, boxplots, curvas de densidad, diagramas de violín, ...**

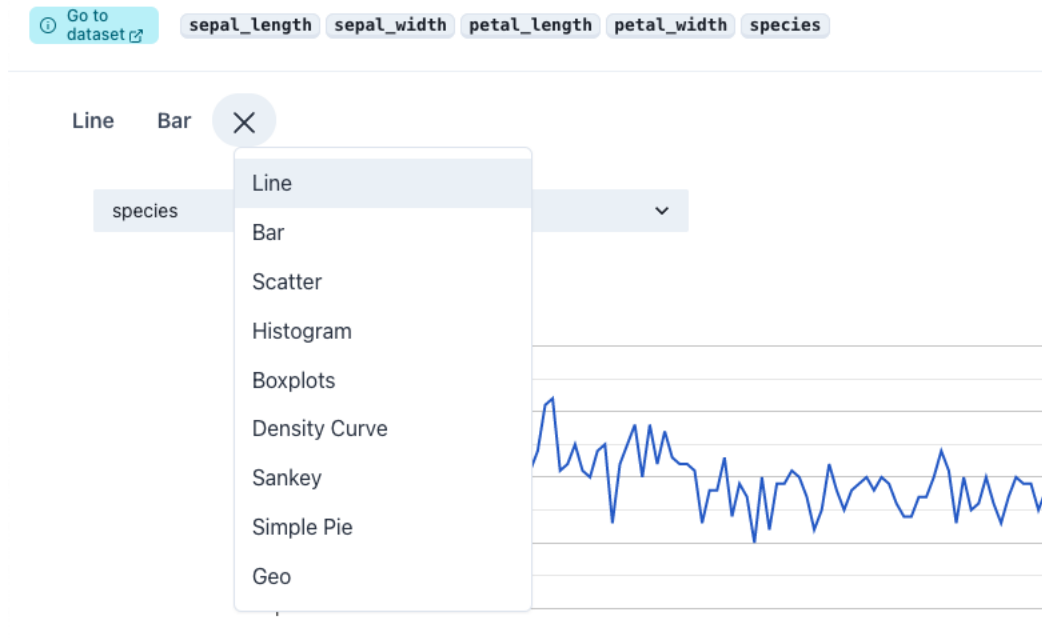
Para esto he añadido una sección exclusiva para los gráficos y funciona de esta manera:

1. Vamos a Go to Charts desde la pagina principal.



2. Por defecto vamos al gráfico de Líneas pero no se pintaría nada hasta seleccionar los ejes X y Y.

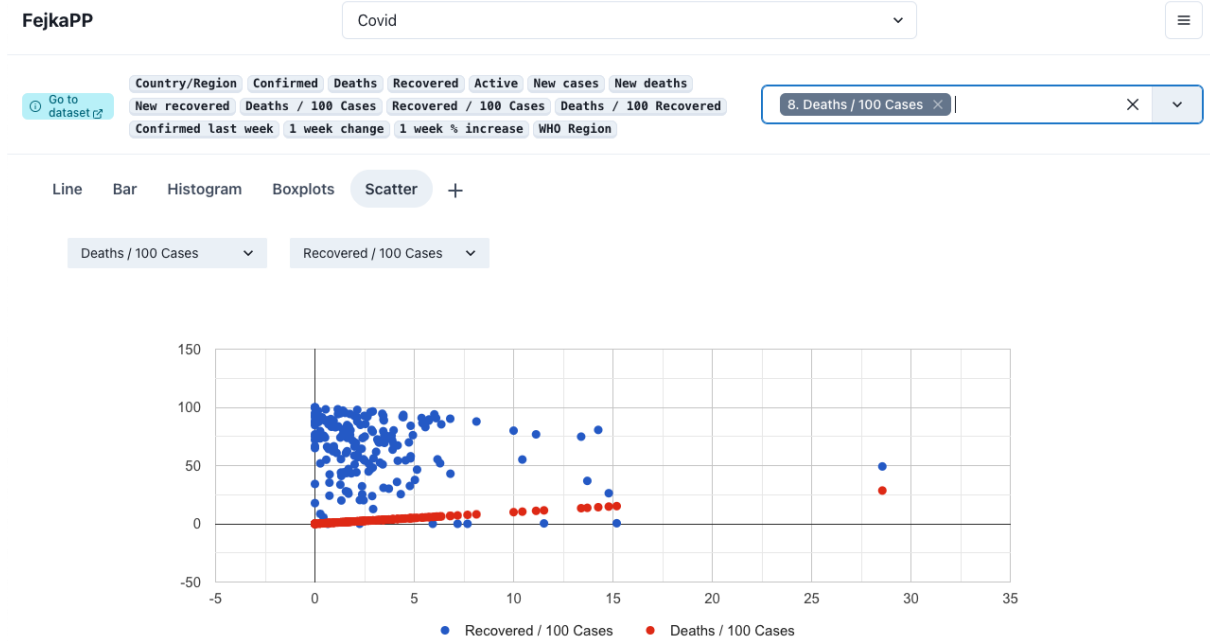




5. Si escogemos el Histograma por ejemplo. en un conjunto de dato diferente.



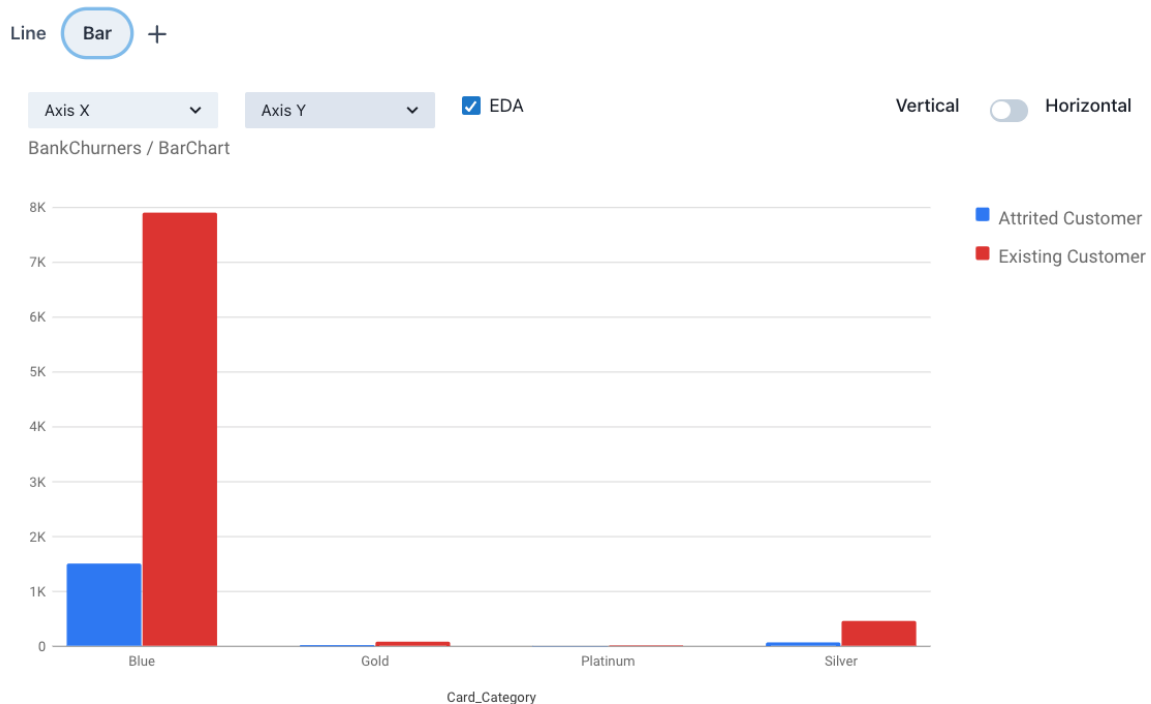
6. Si escogemos el Scatter por ejemplo.



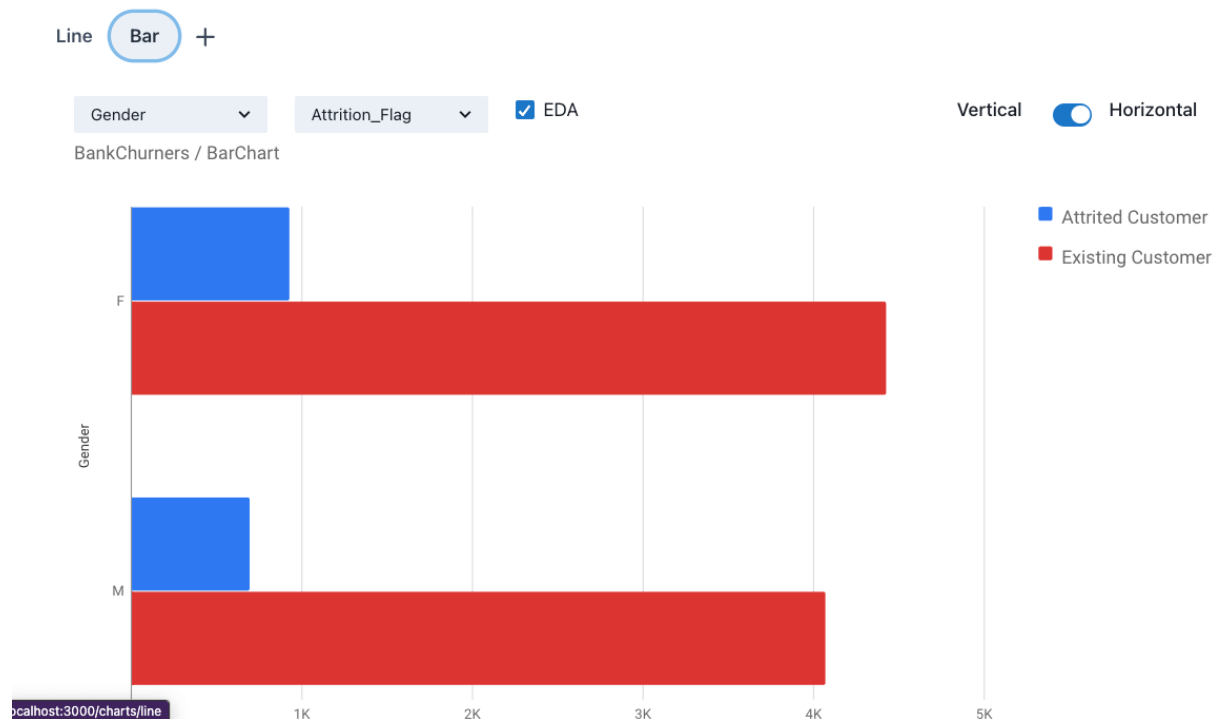
7. De esa manera con cada Chart de la lista.

**Con los conjuntos de datos considerados en las prácticas anteriores desarrolla el tratamiento que permita conocer la distribución de datos y extraer información relativa a comportamientos realizando comparativas de algunas variables: edad, país, género, periodos..**

Las categorías de tarjetas blue tienen mas abandono que el resto por mucho.

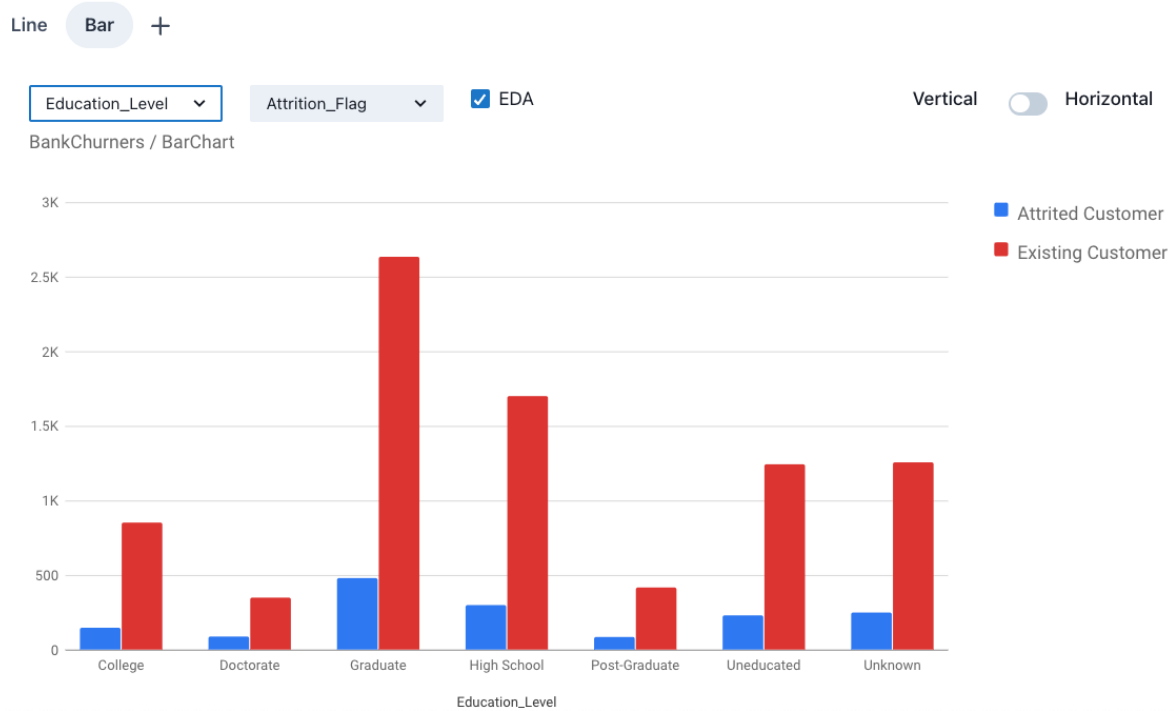


Las mujeres tienen un menor numero de abandono.

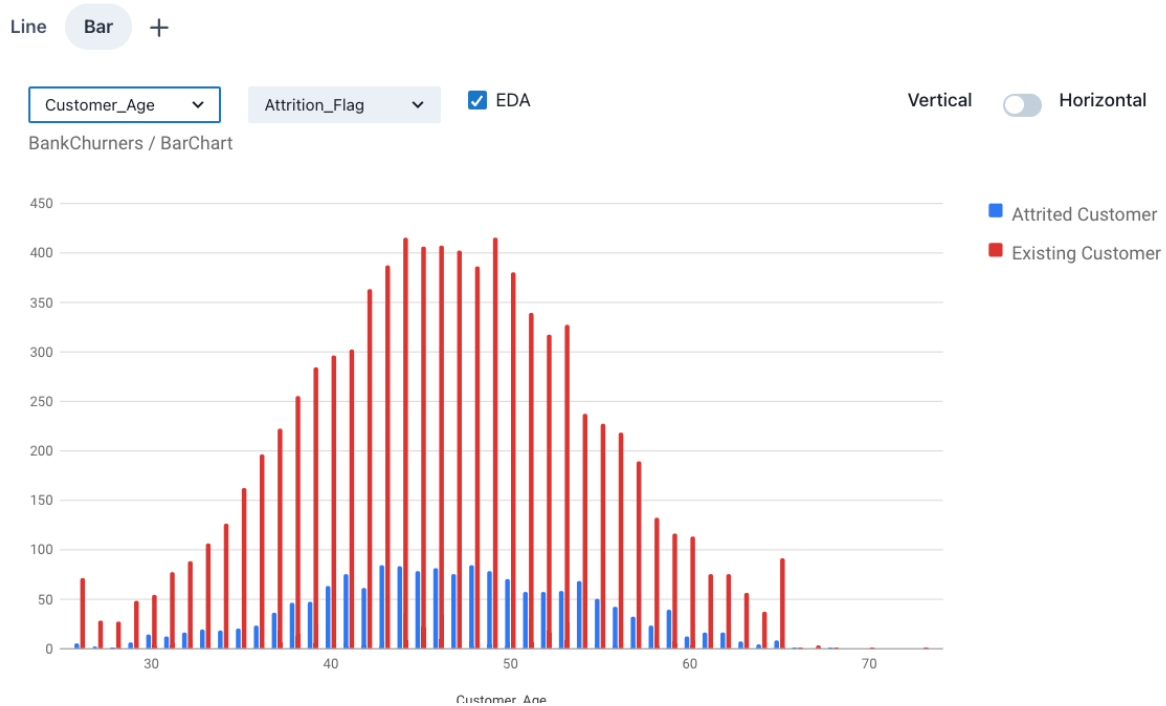


Al parecer las personal con estudios universitarios tienen una mayor tasa de abandono.

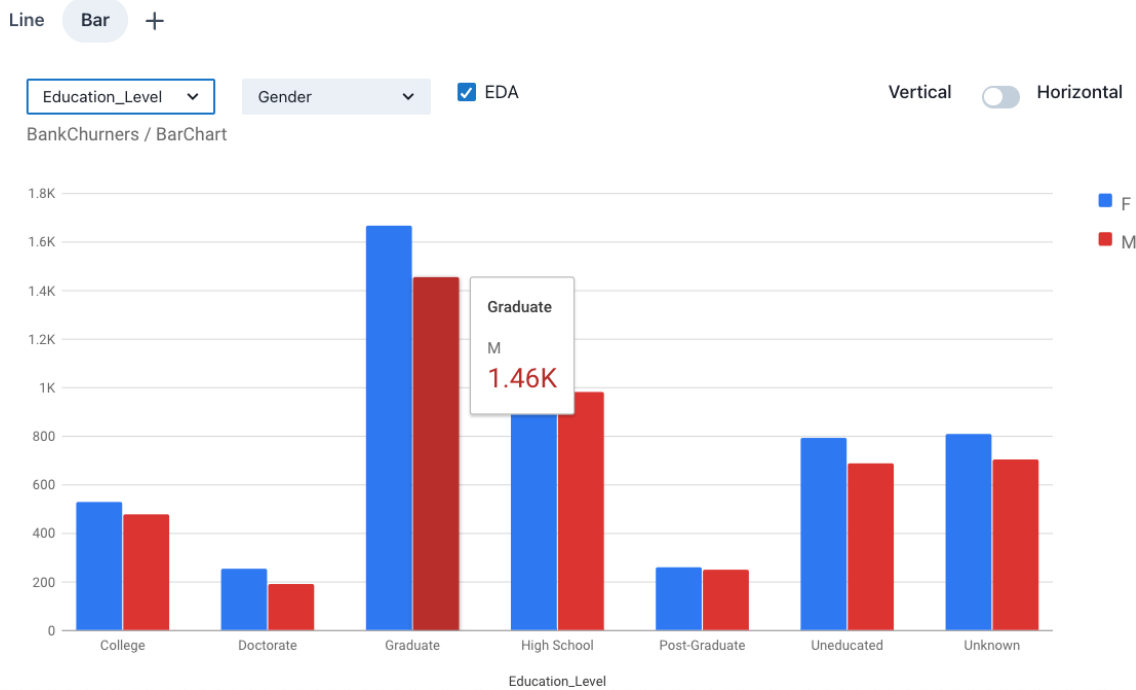




El rango de edad que tiene más abandono esta entre los 40 y 50 años de edad.



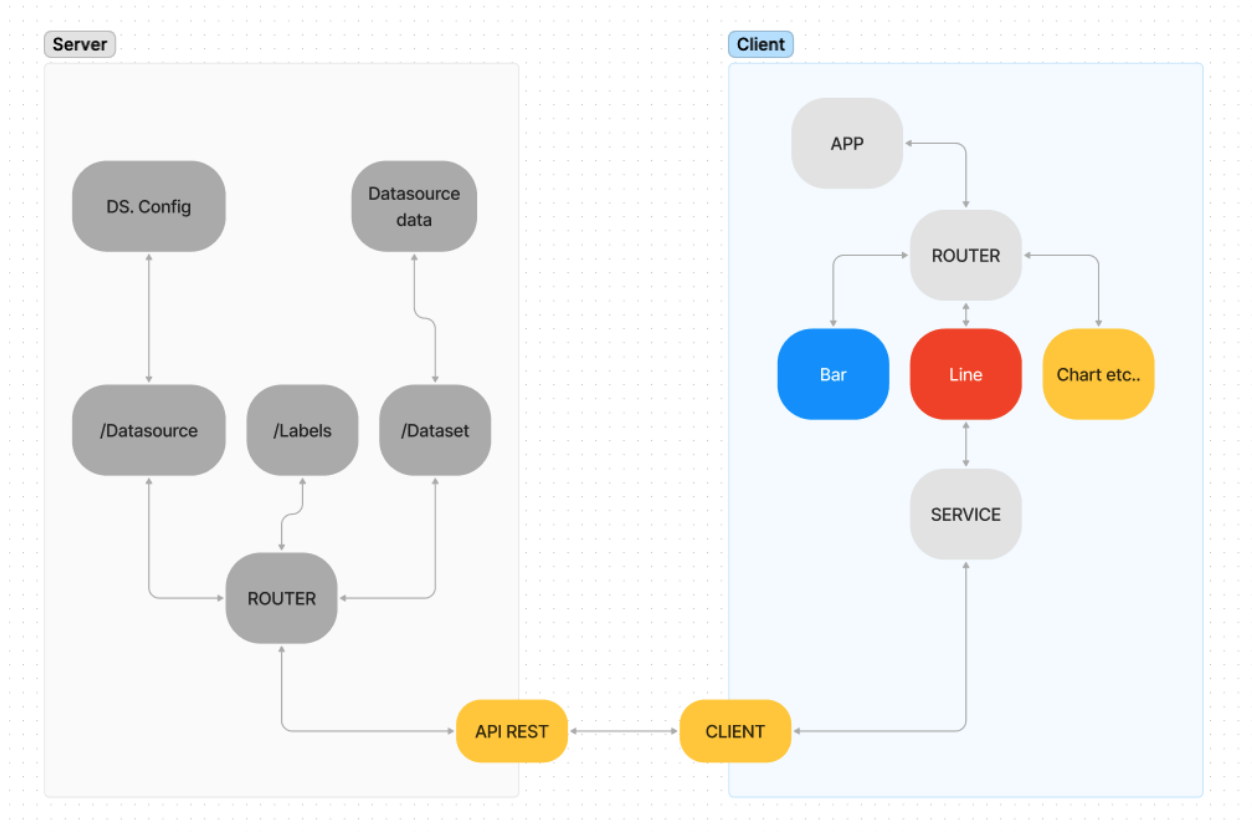
Mujeres graduadas son el grupo más grande del conjunto de dato.



**Haciendo uso del componente "servidor" convierte en un servicio el framework desarrollado. Para poder acceder a este servicio, deberías desarrollar asimismo un cliente específico que permita al usuario seleccionar el origen de los datos. Describe en una página la arquitectura software de la aplicación y las tecnologías desarrolladas.**

### Arquitectura

La arquitectura es un cliente servidor que se comunican por el protocolo http a través de una API REST. En el servidor se descargan las csv, se procesan y se envían los datos en formato JSON y el cliente recibe ese JSON y lo transforma en modelos de datos que la clase Charts pueda pintar.




## Client side

El cliente es una aplicación React. Esta escrita en JavaScript y usa dos framework para el desempeño.

- React Google Charts

React Google Charts | React Google Charts

A thin, typed, React wrapper for Google Charts

 <https://www.react-google-charts.com/>



- Chakra UI

Chakra UI - A simple, modular and accessible component library that gives you the building blocks you need to build your React applications.

Simple, Modular and Accessible UI Components for your React Applications. Built with Styled System

 <https://chakra-ui.com/>



## Server side

El servidor esta escrito en Python y usa un framework llamado "FastApi" para el servicio API REST y panda para el manejo de los Datasets

### FastAPI

FastAPI framework, high performance, easy to learn, fast to code, ready for production Documentation: <https://fastapi.tiangolo.com> Source Code: <https://github.com/tiangolo/fastapi> FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. The key



### pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. <https://pandas.pydata.org/>

**Piensa en un nombre y en algún tipo de aplicación para la que pueda desarrollarse el Framework desarrollado.**

El nombre de mi app se llama **Fejka** y esta inspirado en una plata del **ikea** porque tiene forma de ramas y gráficos, ademas son artificiales

## 5. Recursos y referencias

Repo de la practica:

### GitHub - juanchojbarroso/fejka

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/juanchojbarroso/fejka>

juanchojbarroso/  
**fejka**



Contributor 0 Issues 0 Stars 0 Forks

### How to Use Google Charts in React

Google Charts is a free Javascript library that allows you to visualize data in many types of charts and graphs. It's very useful and easy to use in your projects. In this tutorial, we'll see how to use Google Charts in React by creating a simple React app with Create React App (CRA).

<https://blog.shahednasser.com/how-to-use-google-charts-in-react/>



### Pandas tips and tricks


This post includes some useful tips for how to use Pandas for efficiently preprocessing and feature engineering from large datasets. Pandas has an apply function which let you apply just about any function on all the values in a column. Note that apply is just a little bit faster than a python for loop!

<https://towardsdatascience.com/pandas-tips-and-tricks-33bcc8a40bb9>

1	Chandler Bing	party	2017-08-04 08:00:00
6	Harry Kane	football	2017-08-04 13:00:00
0	John Doe	beach	2017-08-04 14:00:00
3	Joey Tribbiani	party	2017-08-04 10:00:00
2	Monica Geller	travel	2017-08-04 07:00:00

#### 4 Pandas GroupBy Tricks You Should Know

As one of the most popular libraries in Python, Pandas has been utilised very commonly especially in data EDA (Exploratory Data Analysis) jobs. Very typically, it can be used for filtering and transforming dataset just like what we usually do using SQL queries. They share a lot of similar concepts such as joining tables.

 <https://towardsdatascience.com/4-pandas-groupby-tricks-you-should-know-9e5b9870693e>



## 6. Mejoras

#### Recharts is slow with large data · Issue #1146 · recharts/recharts


Do you want to request a feature or report a bug? a feature What is the current behavior? I use Recharts in my application. My application often uses large data (> 3 MB). But the drawing is very slow. For example, ofk/recharts-large-data...

 <https://github.com/recharts/recharts/issues/1146>

recharts/recharts

#### #1146 Recharts is slow with large data

 3 comments

 ofk opened on January 18, 2018



#### How to completely rerender the graph? · Issue #1624 · recharts/recharts


What is the current behavior? When I change the data to LineChart, the only curve is rerendered. But the initial animation(showing the drawing of the graph) is not there. I am using this into a ResponsiveContainer. Here is my state and s...

 <https://github.com/recharts/recharts/issues/1624>

recharts/recharts

#### #1624 How to completely rerender the graph?

 6 comments

 aditya81070 opened on January 27, 2019

