



RETO GRUPAL

ESTADOS DE ESTADOS UNIDOS

ESCUELA DE TALENTO | FUNDACIÓN NTTDATA



INTRODUCCIÓN

En este documento presentamos nuestro trabajo correspondiente a la resolución de los ejercicios de las distintas áreas que componen el reto grupal “Estados de Estados Unidos”.

El proyecto final, ha sido documentado por Ana, quien se ha encargado de la redacción y edición de imagen, y ha sido revisado, corregido y aprobado por el resto de los componentes del equipo.

En él, a parte de las distintas partes incluidas en la presentación, y un resumen ejecutivo de cada reto, adjuntamos links desde donde se puede tener acceso y descargar toda la documentación correspondiente a cada área, en los que se incluyen todos los detalles y, en los que hemos tratado de solventar, y tener en cuenta todo lo indicado en el feedback recibido tras finalizar cada área.

Solo nos queda agradecer a nuestros profesores toda la ayuda, el conocimiento, la motivación y el fantástico ambiente de este curso, a los mentores y, por supuesto, a nuestra tutora Laura, todos excelentes profesionales.

Muchas gracias por este viaje lleno de retos y de crecimiento, tanto personal y como profesional, que dará paso, sin duda, a una nueva etapa para todos nosotros.



ÍNDICE

1 OBJETIVOS PROPUESTOS VS CONTENIDOS

2 RECURSOS HUMANOS Y UTILIZADOS

- Recursos humanos. El equipo.
- Recursos utilizados:
 - Recursos técnicos
 - Recursos metodológicos

3 RESULTADOS OBTENIDOS

- Área I
- Área II
- Área III
- Área IV

4 RIESGOS MATERIALIZADOS

- Dificultades
- Soluciones

5 CONCLUSIONES



OBJETIVOS PROPUESTOS VS CONTENIDOS

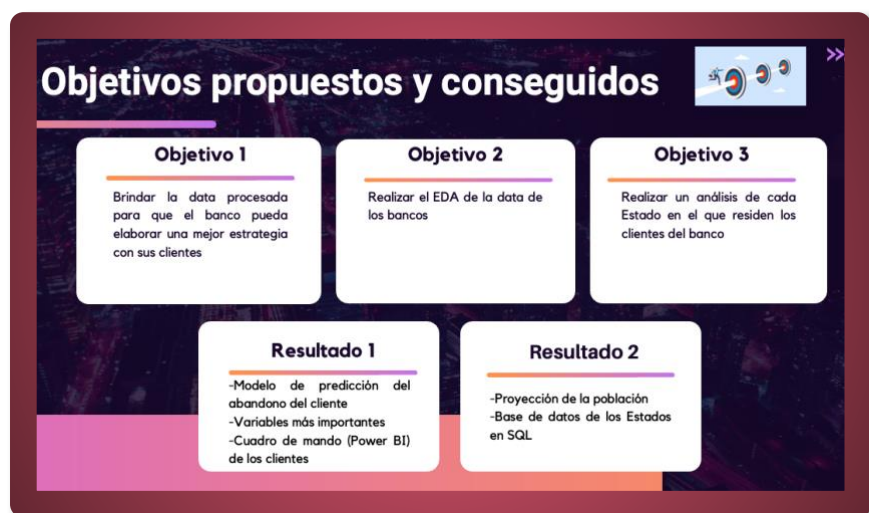
OBJETIVOS PROPUESTOS VS CONTENIDOS

En el reto grupal “Estados de Estados Unidos” se nos plantea un estudio de población de varios estados, que incluye el cálculo del crecimiento de la misma y la creación de bases de datos para procesar y ordenar la información, y posteriormente el uso de los datos de tres estados en concreto, con el objetivo final de poder predecir cuáles son las variables que se han de tener en cuenta, en dichos lugares, para determinar qué es lo que influye o no en la permanencia de los clientes en las entidades bancarias.


Para cumplir este objetivo final, tuvimos que ir cumpliendo objetivos más pequeños, pero no menos importantes, para procesar todos los datos, tales como la creación de una biblioteca con todos los datos correspondientes a los estados que formaban parte del estudio inicial, hacer cálculos sobre la proyección de dicha población, la creación de bases de datos indicadas anteriormente, el análisis de los datos y su limpieza, manejo de los datos, la creación de un modelo predictivo eficaz y poder presentar la información al cliente.

En relación a la primera parte, del análisis por estado y las características propias de la población de los mismos, se cumplieron los objetivos en relación a los cálculos solicitados, ya que se obtuvo una proyección de la población y se crearon las bases de datos requeridas, obteniendo en este proceso los resultados esperados de forma satisfactoria.

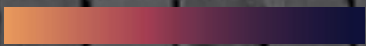
Respecto al objetivo del proyecto solicitado por el cliente, en relación a las entidades bancarias, se realizó un análisis de los tres estados objeto final del estudio, se obtuvo un modelo de predicción de la probabilidad del abandono de un cliente del banco y cuales eran las variables más importantes en el momento del abandono, y se creó un cuadro de mando donde poder presentar los datos de una forma más visual, considerando por tanto que se cumplieron los objetivos de forma exitosa.



Pantallazo de la presentación de Jaime



RECURSOS HUMANOS Y RECURSOS UTILIZADOS



RECURSOS HUMANOS

El equipo formado para el desarrollo del Reto “Estados de Estados Unidos”, estaba conformado inicialmente por Luisa Romero, Jesé Muñoz, Jaime Shimohira, Sergio Bulbarela y Ana Delgado, pero durante el desarrollo del Área II, se incorporó Jéssica Ríos, por lo que el equipo quedó conformado por un total de seis miembros.

En el desarrollo este proyecto, y teniendo en cuenta las valoraciones personales que cada uno ha ido comentado, pudimos ver que han existido roles muy marcados, que han determinado tanto los procesos de trabajo y la consecución de los objetivos, como el desarrollo de nuestros vínculos y la formación del equipo, que ha pasado por todas las fases, incluida la fase de conflicto, de forma exitosa, y que nos ha servido tanto para crear un grupo sólido y consistente, como para crecer personal y profesionalmente.

A continuación, podemos conocer un poco más a los miembros del equipo, su papel dentro del mismo, y aunque su labor individual dentro del trabajo en equipo está detallada en los documentos anexos, todos compartimos conocimiento y recursos con los demás. Indicamos también la puntuación consensuada.



Sergio Bulbarela. Data Analyst observador. Su rol es menos participativo, pero revisa los procesos que se llevan a cabo con escucha activa, aporta feedback y puntos de vista fuera de la caja. Su valoración es de 1,25 puntos.



Jesé Muñoz. Data Analyst experto. Mantiene el equilibrio entre independencia y trabajo en equipo, además de tener conocimientos técnicos de valor, como por ejemplo en Power Bi. Su valoración es de 1,83 puntos.



Jaime Shimoshira. Data Analyst investigador de recursos. Rol muy necesario en equipos como el nuestro, buscando siempre la solución de forma eficaz en momentos complejos, y facilitándonos la información de forma concisa y muy bien explicada. Su valoración es de 1,82 puntos.



Jéssica Moreno. Data Analyst organizadora. Es muy proactiva y con habilidad para sintetizar las ideas compartir conocimiento, ordenar y reunir datos. Destaca que, aunque se incorporó más tarde, fue como si siempre hubiese estado. Su valoración es de 2,3 puntos.



Luisa Moreno. Data Analyst implementadora. Destaca su forma práctica de llevar las ideas a cabo y su proactividad, y capacidad de escucha. Resolutiva en cuestiones de código. Su valoración es de 2,3 puntos.



Ana Delgado. Data Analyst coordinadora o líder. Pendiente del trabajo colaborativo, la comunicación, cohesión del grupo, y de organizar que se cumplan los objetivos en los plazos requeridos, uniendo distintas partes, documentando el proceso y creando código. Su valoración es de 2,5 puntos.

RECURSOS UTILIZADOS

Para poder trabajar y resolver los retos planteados, el equipo utilizó recursos técnicos y recursos metodológicos que pasamos a detallar a continuación.

RECURSOS TÉCNICOS

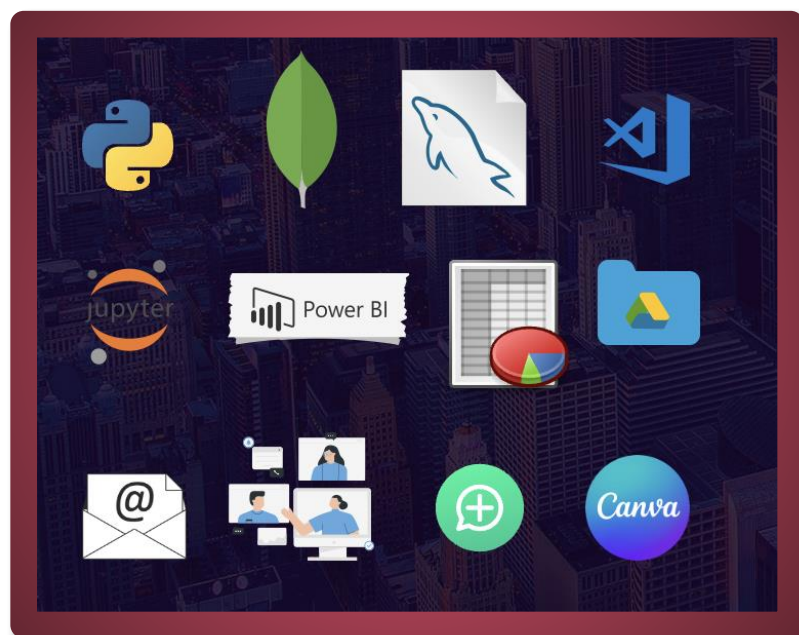
En relación a los recursos técnicos utilizados, podríamos dividirlos en dos grupos, los utilizados para la resolución de los ejercicios y los utilizados para la presentación de los mismos.

En el primer grupo estarían incluidos los distintos programas que hemos utilizado y/o aprendido a utilizar tanto para crear el código, para la creación y gestión de base de datos, la limpieza de datos, presentación de resultados, etc.

En este grupo estaría, por ejemplo, Python y sus librerías, Visual Studio Code, Anaconda y los correspondientes Notebook de Jupyter, MongoDB, MySQL o Powerbi.

En el segundo grupo, incluimos como comentaba anteriormente, otro tipo de recursos necesarios para la presentación de los ejercicios, tales como Word, Excell, PowerPoint o Canva.

También estarían incluidos en este segundo grupo las plataformas utilizadas para realizar videollamada, fundamental en el trabajo colaborativo, como Teams o Google Meet, uso de nubes, en este caso Google Drive, para la compartición de archivos, y finalmente Whatsapp para mantenernos actualizados y en contacto.



Ejemplo de recursos técnicos utilizados de la presentación de Ana

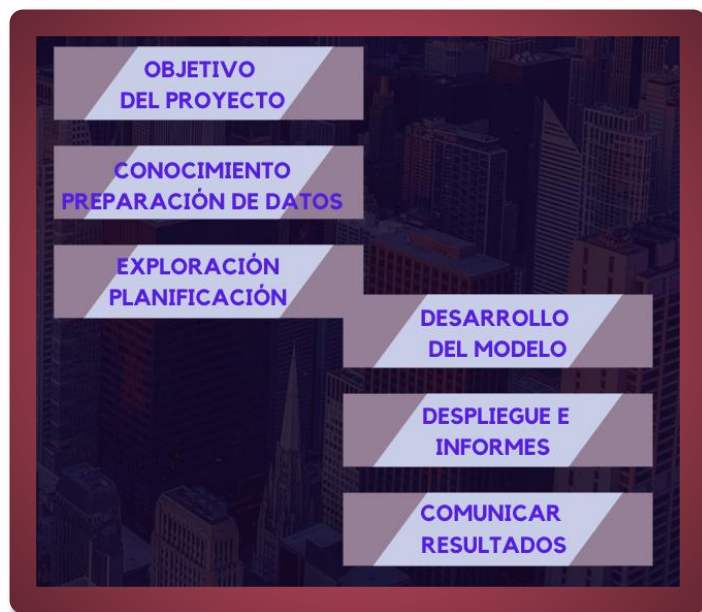
RECURSOS METODOLÓGICOS

Respecto a los recursos metodológicos, todos los retos han sido desarrollados con una parte síncrona en la que hemos tenido lluvia de ideas, nos hemos podido organizar y avanzar de una forma 100% colaborativa y a tiempo real, y cuando esto no ha sido posible, hemos trabajado de forma asíncrona, pero de nuevo basándonos en un trabajo colaborativo en el que los miembros del equipo participaban, enseñaban y daban feedback al resto.


En relación al proceso en sí, seguido para la consecución de los distintos objetivos hemos seguido una metodología scrum y ágil, con el trabajo en sprint como centro, y en la que todas las fases tienen un comienzo y un final definidos y limitados por el tiempo.

En estos sprints, el trabajo colaborativo es fundamental y responde a dos preguntas básicas, qué trabajo vamos a realizar y cómo se va a realizar, de manera que el equipo pueda crear un plan para finalizar cada elemento de trabajo hasta que se completen todos y se consiga el objetivo.


Y finalmente, en la siguiente imagen, podemos también ver la metodología del trabajo como analistas, que se puede resumir en estos pasos y que culminamos con la presentación final.



Ciclo de vida de un proyecto de análisis de datos de la presentación de Ana



RESULTADOS OBTENIDOS. RESUMEN EJECUTIVO





ÁREA I

FUNDAMENTOS DE PYTHON

El trabajo correspondiente a esta área se centraba principalmente en crear código que nos sirviera para trabajar con datos generales y específicos de la población de varios estados de Estados Unidos. El trabajo desarrollado en esta área se subdividió en varios apartados que explicaremos de una forma más detallada y en los que trabajamos los siguientes puntos:

- Crear diccionarios y listas, actualizar datos incorrectos, introducir nuevas claves, cálculos y desarrollo de programas.
- Calcular ratios.
- Definir funciones.
- Uso de librería Folium.

El trabajo se inició con la creación de una lista con los datos correspondientes a los estados de Alabama, Florida, Georgia y South Carolina, estableciendo como claves “Estado”, “Población 2000”, “Población 2001”, “Residentes menores de 65 años 2000”, “Residentes menores de 65 años 2001”, “Muertes 2000”, “Muertes 2001”, “Fecha de fundación del estado”, “Latitud” y “Longitud”.

Dicha lista posteriormente fue modificada mediante código, convirtiéndose en un diccionario para cada estado. Asimismo, se actualiza el dato correspondiente a la población de Florida en 2001, que constaba incorrecto.

Tras estos cambios se incluyen nuevas claves

- “Días desde fundación nombre_estado”, que se corresponda con el número de días que han pasado desde la fundación del Estado hasta la actualidad, y calculemos el número de meses que han transcurrido.
- “Porcentaje mayores de 65 años nombre_estado”

Tras crear el código que cree los cálculos necesarios para completar dichas claves, pasamos a desarrollar un programa que imprima por pantalla el nombre del Estado más antiguo y el más moderno, y que valga para cualquier número de estados, y que calcule cuántos años de diferencia hay entre ellos.

Una vez sentadas todas las bases para poder trabajar con los datos de cada estado, creamos un código que calculase la tasa de crecimiento de población para 2002 de los estados de Alabama y South Carolina teniendo en cuenta que la tasa de crecimiento será un número aleatorio entre 0 y 0.1 y con el objetivo de contestar a las siguientes preguntas:

- a. ¿Cuántos años tardará el estado de South Carolina en alcanzar en población a Alabama?
- b. ¿En qué año ocurrirá esto?
- c. Si al planteamiento anterior le añadimos la tasa de fallecidos de cada estado. ¿Cuántos años tardará South Carolina en alcanzar en población a Alabama? Vamos a asumir la tasa de fallecidos y la población correspondiente a 2021.

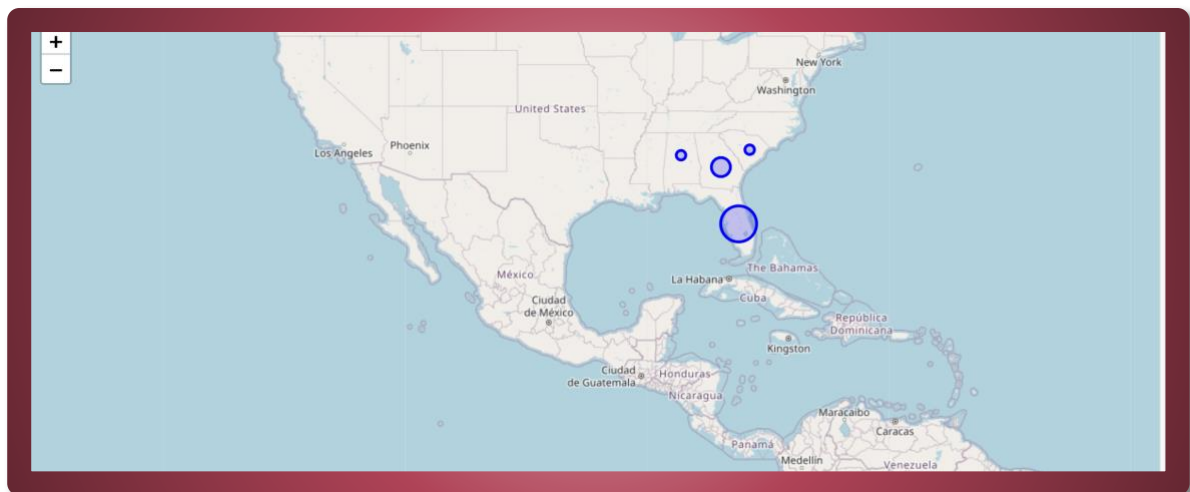
Asimismo, creamos una función para generar una proyección para el año 2002, utilizando como ratio la comparativa entre los años 2000 y 2001. Esta función fue creada tras la revisión del reto inicial:

```
def proyeccion_poblacion():
    for estado in lista_estados:

        # se hace el calculo de la poblacion para 2002 con la formula dada y se actualiza el diccionario
        poblacion_2002 = round(estado['Poblacion 2001'] / estado['Poblacion 2000'] * estado['Poblacion 2001'])
        estado.update({"Poblacion 2002": poblacion_2002 })

        print("La población para el año 2002 en el estado de", estado["Estado"], "será:", poblacion_2002)
        # se hace el calculo para sacar la proyeccion estimada para cda uno de los estados con la formula dada
        calculo_entre_fechas = 1900 - int(estado["Fecha de fundacion del Estado"][-4:])
        formula_de_proyeccion = round((14500 * calculo_entre_fechas + 7000 / (2 * calculo_entre_fechas) + 1))
        estado.update({"Estimacion": formula_de_proyeccion })
        print("La estimación de población en el estado de", estado["Estado"], "será:", formula_de_proyeccion)
```

Finalmente, y para finalizar el cumplimiento de objetivos de esta área, creamos un mapa donde se representan con marcadores de color azul, los datos de población del 2002 para cada uno de los estados, en las coordenadas que se nos facilitan.



ÁREA II

BASE DE DATOS

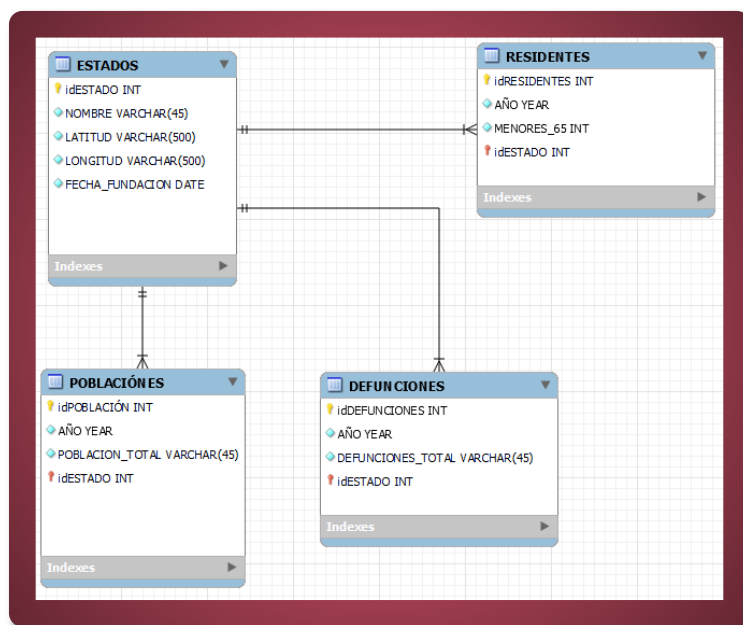
Tras conocer los datos con los que íbamos a trabajar y después de crear y definir funciones que nos permitieran hacer cálculos con los mismos, procedimos con la creación de las bases de datos necesarias para continuar con el proyecto.

Creamos diagramas entidad-relación con DIA, entre cuatro tablas, Estados que sería la tabla principal y las tablas Poblaciones, Residentes y Defunciones.

Una vez concretados los diagramas, creamos por palabras la estructura de manera que las tablas tuvieran con los siguientes atributos y cardinalidades

- ESTADOS: ID (PK), Nombre, Latitud, Longitud, Fecha_fundación.
 - POBLACIONES: ID (PK), Año, Poblacion_total, idEstado (FK).
 - RESIDENTES: ID (PK), Año, Menores_65, idEstado (FK).
 - DEFUNCIONES: ID (PK), Año, Defunciones_totales, idEstado (FK).
- La relación de ESTADOS con POBLACIONES tiene cardinalidad de (1, N), es decir, un Estado puede tener muchas poblaciones, aunque cada población sólo podrá pertenecer a un Estado específico.
 - La relación de ESTADOS con RESIDENTES tiene cardinalidad de (1, N), es decir, un Estado puede tener muchos residentes, pero a su vez cada residente sólo podrá pertenecer a un Estado específico.
 - La relación de ESTADOS con DEFUNCIONES tiene cardinalidad de (1, N), es decir, un Estado puede tener muchas defunciones entre sus habitantes, aunque cada defunción sólo podrá contabilizarse en un Estado específico.

Posteriormente se modelaron las tablas en Workbench



Para finalizar el ejercicio, se crearon bases de datos relacionales SQL con Python y se insertaron los datos, y también se creó en PHPMYADMIN:

```
1 #Conectar con BBDD
2 import mysql.connector
3
4 conexion=mysql.connector.connect(host="localhost", user="root", password="", database="")
5
6 #Crear cursor
7 cursor = conexion.cursor()
8
9 #Ejecutar código SQL
10 sql = """
11 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
12 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
13 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
14
15 -- Schema RETO_ESTADOS
16 --
17 --
18 CREATE SCHEMA IF NOT EXISTS 'RETO_ESTADOS' ;
19 USE 'RETO_ESTADOS' ;
20
21 --
22 -- Table 'RETO_ESTADOS'. 'ESTADOS'
23 --
24 CREATE TABLE IF NOT EXISTS 'RETO_ESTADOS'. 'ESTADOS' (
25   'idESTADO' INT NOT NULL,
26   'NOMBRE' VARCHAR(45) NOT NULL,
27   'LATITUD' VARCHAR(500) NOT NULL,
28   'LONGITUD' VARCHAR(500) NOT NULL,
29   'FECHA_FUNDACION' DATE NOT NULL,
30   PRIMARY KEY ('idESTADO'))
31 ENGINE = InnoDB;
32
33 --
34 -- Table 'RETO_ESTADOS'. 'POBLACIONES'
35 --
36 CREATE TABLE IF NOT EXISTS 'RETO_ESTADOS'. 'POBLACIONES' (
37   'idPOBLACION' INT NOT NULL,
38   'AÑO' YEAR NOT NULL,
39   'POBLACION_TOTAL' VARCHAR(45) NOT NULL,
40   'idESTADO' INT NOT NULL,
41   PRIMARY KEY ('idPOBLACION'),
42   FOREIGN KEY ('idESTADO')
43     REFERENCES 'RETO_ESTADOS'. 'ESTADOS' ('idESTADO')
44     ON DELETE NO ACTION
45     ON UPDATE NO ACTION)
46 ENGINE = InnoDB;
47
48 --
49 -- Table 'RETO_ESTADOS'. 'DEFUNCIONES'
50 --
51 CREATE TABLE IF NOT EXISTS 'RETO_ESTADOS'. 'DEFUNCIONES' (
52   'idDEFUNCIONES' INT NOT NULL,
53   'AÑO' YEAR NOT NULL,
54   'DEFUNCIONES_TOTAL' VARCHAR(45) NOT NULL,
55   'idESTADO' INT NOT NULL,
56   PRIMARY KEY ('idDEFUNCIONES'),
57   FOREIGN KEY ('idESTADO')
58     REFERENCES 'RETO_ESTADOS'. 'ESTADOS' ('idESTADO')
59     ON DELETE NO ACTION
60     ON UPDATE NO ACTION)
61 ENGINE = InnoDB;
62
63 --
64 -- Table 'RETO_ESTADOS'. 'RESIDENTES'
65 --
66 CREATE TABLE IF NOT EXISTS 'RETO_ESTADOS'. 'RESIDENTES' (
67   'idRESIDENTES' INT NOT NULL,
68   'AÑO' YEAR NOT NULL,
69   'MENORES_65' VARCHAR(45) NOT NULL,
70   'idESTADO' INT NOT NULL,
71   PRIMARY KEY ('idRESIDENTES'),
72   FOREIGN KEY ('idESTADO')
73     REFERENCES 'RETO_ESTADOS'. 'ESTADOS' ('idESTADO')
74     ON DELETE NO ACTION
75     ON UPDATE NO ACTION)
76 ENGINE = InnoDB;
77 """
78
79 # Dividir el código SQL en consultas individuales
80 queries = sql.split(';')
81
82 # Ejecutar cada consulta individualmente
83 for query in queries:
84     cursor.execute(query)
85
86 # Cerrar la conexión
87 conexion.close()
```

```

1 #Conectar con BBDD
2 import mysql.connector
3
4 conexion=mysql.connector.connect(host="localhost", user="root", password="", database="")
5
6 #Crear cursor
7 cursor = conexion.cursor()
8
9 #Ejecutar código SQL
10 sql = """
11 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
12 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
13 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
14
15 -- Schema RETO_ESTADOS
16
17 -- Table "RETO_ESTADOS"."ESTADOS"
18 CREATE SCHEMA IF NOT EXISTS "RETO_ESTADOS" ;
19 USE "RETO_ESTADOS" ;
20
21 -- Table "RETO_ESTADOS"."ESTADOS"
22
23 CREATE TABLE IF NOT EXISTS "RETO_ESTADOS"."ESTADOS" (
24   "idESTADO" INT NOT NULL,
25   "NOMBRE" VARCHAR(45) NOT NULL,
26   "LATITUD" VARCHAR(500) NOT NULL,
27   "LONGITUD" VARCHAR(500) NOT NULL,
28   "FECHA_FUNDACION" DATE NOT NULL,
29   PRIMARY KEY ("idESTADO"))
30 ENGINE = InnoDB;
31
32 -- Table "RETO_ESTADOS"."POBLACIONES"
33
34 CREATE TABLE IF NOT EXISTS "RETO_ESTADOS"."POBLACIONES" (
35   "idPOBLACION" INT NOT NULL,
36   "AÑO" YEAR NOT NULL,
37   "POBLACION_TOTAL" VARCHAR(45) NOT NULL,
38   "idESTADO" INT NOT NULL,
39   PRIMARY KEY ("idPOBLACION"),
40   FOREIGN KEY ("idESTADO")
41     REFERENCES "RETO_ESTADOS"."ESTADOS" ("idESTADO")
42     ON DELETE NO ACTION
43     ON UPDATE NO ACTION)
44 ENGINE = InnoDB;
45
46 -- Table "RETO_ESTADOS"."DEFUNCIONES"
47
48 CREATE TABLE IF NOT EXISTS "RETO_ESTADOS"."DEFUNCIONES" (
49   "idDEFUNCIONES" INT NOT NULL,
50   "AÑO" YEAR NOT NULL,
51   "DEFUNCIONES_TOTAL" VARCHAR(45) NOT NULL,
52   "idESTADO" INT NOT NULL,
53   PRIMARY KEY ("idDEFUNCIONES"),
54   FOREIGN KEY ("idESTADO")
55     REFERENCES "RETO_ESTADOS"."ESTADOS" ("idESTADO")
56     ON DELETE NO ACTION
57     ON UPDATE NO ACTION)
58 ENGINE = InnoDB;
59
60 -- Table "RETO_ESTADOS"."RESIDENTES"
61
62 CREATE TABLE IF NOT EXISTS "RETO_ESTADOS"."RESIDENTES" (
63   "idRESIDENTES" INT NOT NULL,
64   "AÑO" YEAR NOT NULL,
65   "MENORES_65" VARCHAR(45) NOT NULL,
66   "idESTADO" INT NOT NULL,
67   PRIMARY KEY ("idRESIDENTES"),
68   FOREIGN KEY ("idESTADO")
69     REFERENCES "RETO_ESTADOS"."ESTADOS" ("idESTADO")
70     ON DELETE NO ACTION
71     ON UPDATE NO ACTION)
72 ENGINE = InnoDB;
73 """
74
75 # Dividir el código SQL en consultas individuales
76 queries = sql.split(';')
77
78 # Ejecutar cada consulta individualmente
79 for query in queries:
80   cursor.execute(query)
81
82 # Cerrar la conexión
83 conexion.close()

```

✓ Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0004 segundos.)

SELECT * FROM `estados`

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: | Ordenar según la clave:

Opciones extra

<div><div><div><div></div><div></div></div><div><div></div><div></div></div></div><div></div></div>				idESTADO	NOMBRE	LATITUD	LONGITUD	FECHA_FUNDACION
<div><div><div></div></div></div>	<div><div><div></div></div><div>Editar</div></div>	<div><div><div></div></div><div>Copiar</div></div>	<div><div><div></div></div><div>Borrar</div></div>	1	Alabama	33.258.882	-86.829.534	1819-12-14
<div><div><div></div></div></div>	<div><div><div></div></div><div>Editar</div></div>	<div><div><div></div></div><div>Copiar</div></div>	<div><div><div></div></div><div>Borrar</div></div>	2	Florida	27.756.767	-81.463.983	1845-03-03
<div><div><div></div></div></div>	<div><div><div></div></div><div>Editar</div></div>	<div><div><div></div></div><div>Copiar</div></div>	<div><div><div></div></div><div>Borrar</div></div>	3	Georgia	32.329.381	-83.113.737	1733-02-12
<div><div><div></div></div></div>	<div><div><div></div></div><div>Editar</div></div>	<div><div><div></div></div><div>Copiar</div></div>	<div><div><div></div></div><div>Borrar</div></div>	4	South Carolina	33.687.439	-80.436.374	1776-03-26

Seleccionar todo | Para los elementos que están marcados: Editar Copiar Borrar Exportar

Para terminar, creamos una base de datos no relacional en MongoDB a través de Visual Studio Code y finalizamos comprobando que se visualizaba todo correctamente y que los datos estaban insertados.

Se aportan pantallazos de parte de estos procesos, el proceso completo puede comprobarse en la documentación facilitada en el anexo:

```
1 use("Reto_Estados")
2 db.createCollection("Estados")
```

```
3 db.Estados.insertMany([
4   {
5     "Nombre": "Alabama",
6     "2000": {"Poblacion": 4447100,
7              "Residentes < 65 años":3870598,
8              "Muertes":10622,
9            },
10    "2001": {"Poblacion": 4451493,
11             "Residentes < 65 años":3880476,
12             "Muertes":15912,
13            },
14    "Latitud": 33.258882,
15    "Longitud": -86.829534,
16    "Fecha fundación": "14-12-1819"
17  },
18  {
19    "Nombre": "Florida",
20    "2000": {"Poblacion": 15982378,
21             "Residentes < 65 años":13237167,
22             "Muertes":38103,
23            },
24    "2001": {"Poblacion": 17054000,
25             "Residentes < 65 años":13548077,
26             "Muertes":166069,
27            },
28    "Latitud": 27.756767,
29    "Longitud": -81.463983,
30    "Fecha fundación": "03-03-1845"
31  },
32 ])
```



The screenshot shows the MongoDB Compass interface. On the left, the 'Reto_Estados' database is selected, and the 'Estados' collection is visible. The main panel displays the details of the 'Estados' collection, including a table with the following data:

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
Estados	4	975B	244B	20KB	1	20KB	20KB

QUERY RESULTS: 1-4 OF 4

```

▼ 2000: Object
  Poblacion: 4447100
  Residentes < 65 años: 3870598
  Muertes: 10622
▼ 2001: Object
  Poblacion: 4451493
  Residentes < 65 años: 3880476
  Muertes: 15912
  _id: ObjectId('645e7af2514cb3e80844b79d')
  Nombre: "Alabama"
  Latitud: 33.258882
  Longitud: -86.829534

▼ 2000: Object
  Poblacion: 15982378
  Residentes < 65 años: 13237167
  Muertes: 38103
▼ 2001: Object
  Poblacion: 17054000
  Residentes < 65 años: 13548077
  Muertes: 166069
  _id: ObjectId('645e7af2514cb3e80844b79e')
  Nombre: "Florida"
  Latitud: 27.756767
  Longitud: -81.463983
  Fecha fundación: "03-03-1845"

▼ 2000: Object
  Poblacion: 8186453
  Residentes < 65 años: 7440877
  Muertes: 14804
▼ 2001: Object
  Poblacion: 8229823
  Residentes < 65 años: 7582146
  Muertes: 15000
  _id: ObjectId('645e7af2514cb3e80844b79f')
  Nombre: "Georgia"
  Latitud: 32.329381
  Longitud: -83.113737
  Fecha fundación: "12-02-1733"

▼ 2000: Object
  Poblacion: 4012012
  Residentes < 65 años: 3535770
  Muertes: 8581
▼ 2001: Object
  Poblacion: 4023438
  Residentes < 65 años: 3567172
  Muertes: 9500
  _id: ObjectId('645e7af2514cb3e80844b7a0')
  Nombre: "South Carolina"
  Latitud: 33.687439
  Longitud: -80.436374

```

ÁREA III

LIMPIEZA DE DATOS

En esta área realizamos una limpieza de datos para su posterior análisis con objeto de determinar cuales eran las variables determinantes para realizar un modo predictivo, que arrojase información sobre qué llevaba a un cliente a permanecer o no en una entidad bancaria.

En las conclusiones finales, se determinó que las variables “Age” y “Balance” eran determinantes para la permanencia o abandono de los clientes, y que realizar el estudio por separado, en función de diferentes zonas geográficas arrojaba información más precisa.

No obstante, se observó que había datos que no estaban balanceados en dos de las variables objeto del estudio “Exited”, que era la variable objetivo y “NumOfProducts”, por lo que tras el informe inicial se ha procedido con la corrección del notebook aplicando SMOTETomek ya que el oversampling sugerido por Ana nos pareció la forma más correcta de tratar dicho desbalanceo, ya que mejoraría el rendimiento de los modelos predictivos, conserva la información original, reduciendo el sesgo hacia la clase mayoritaria.

Teniendo en cuenta este cambio en la forma de tratar los datos, se aplica el código de SMOTETomek facilitado por Jéssica y a partir de ahí Ana realiza todas las modificaciones en el resto de pasos para calcular de nuevo los modelos, la relación entre las variables y también el análisis realizado por estados.

Aportamos un pantallazo del código utilizado para el balanceo:

```
# Importar las librerías necesarias
from imblearn.combine import SMOTETomek
from sklearn.model_selection import train_test_split
from collections import Counter

# Crear el DataFrame balanceado
df_balanceo = pd.concat([df7.loc[df7.Exited == 1].iloc[:5000], df7.loc[df7.Exited == 0].iloc[:1000]])

# Separar las características (X) y la variable objetivo (y)
X = df_balanceo.drop('Exited', axis=1)
y = df_balanceo['Exited']

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7)

# Aplicar SMOTETomek para balancear los datos
os_us = SMOTETomek()
X_train_res, y_train_res = os_us.fit_resample(X_train, y_train)

# Mostrar la distribución antes y después del balanceo
print("Distribución antes del balanceo: {}".format(Counter(y_train)))
print("Distribución después del balanceo: {}".format(Counter(y_train_res)))

Distribución antes del balanceo: Counter({1: 1177, 0: 715})
Distribución después del balanceo: Counter({0: 930, 1: 930})
```

```
# CONTINUAMOS CON LA VARIABLE NUMOFPRODUCTS

# Importar las librerías necesarias
from imblearn.combine import SMOTETomek
from sklearn.model_selection import train_test_split
from collections import Counter

# Crear el DataFrame balanceado
df_balanceo = pd.concat([df7.loc[df7.NumOfProducts == 1].iloc[:500],
                        df7.loc[df7.NumOfProducts == 2].iloc[:500],
                        df7.loc[df7.NumOfProducts == 3].iloc[:500],
                        df7.loc[df7.NumOfProducts == 4].iloc[:500]])

# Separar las características (X) y la variable objetivo (y)
X = df_balanceo.drop('NumOfProducts', axis=1)
y = df_balanceo['NumOfProducts']

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7)

# Aplicar SMOTETomek para balancear los datos
os_us = SMOTETomek()
X_train_res, y_train_res = os_us.fit_resample(X_train, y_train)

# Mostrar la distribución antes y después del balanceo
print("Distribución antes del balanceo: {}".format(Counter(y_train)))
print("Distribución después del balanceo: {}".format(Counter(y_train_res)))

Distribución antes del balanceo: Counter({1: 365, 2: 337, 3: 181, 4: 45})
Distribución después del balanceo: Counter({4: 282, 3: 257, 2: 243, 1: 236})
```

Si bien es cierto que las variables que consideramos en un inicio eran muy importantes, el nuevo estudio confirma que la variable “NumOfProducts” era sumamente determinante, tanto de forma individual como relacionada con otras variables, ya que cuanto mayor era el número de productos contratados, mayor era la relación con la permanencia de los clientes en la entidad.

Dado que los datos balanceados implicaron cambios en las gráficas y el proceso de análisis en general, en la documentación del reto hemos adjuntado un anexo con el nuevo estudio y la documentación del mismo, y también las correcciones en las conclusiones y en los procesos previos.

Aportamos los links con el enlace a la documentación indicada, la primera versión del notebook del análisis de datos y el notebook actual.

ÁREA IV

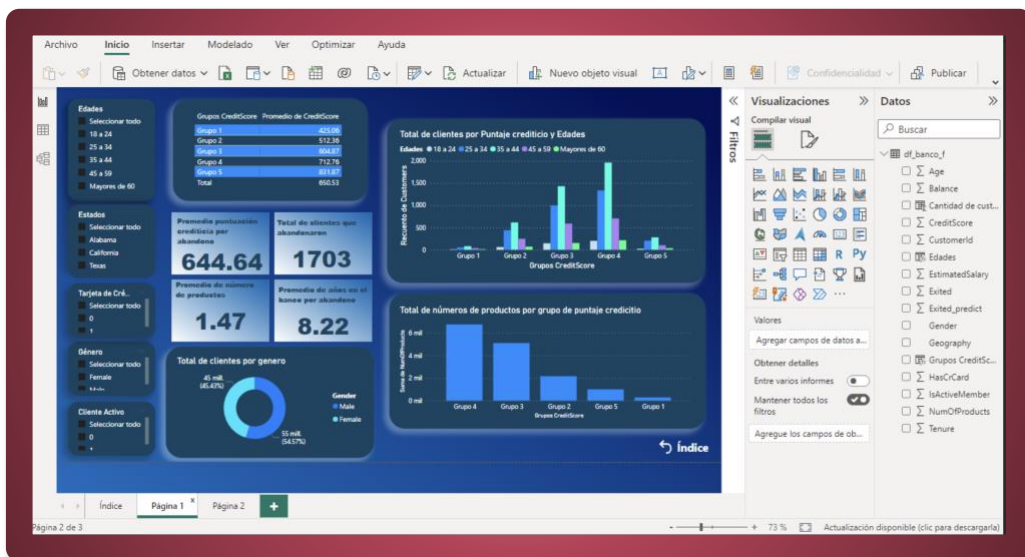
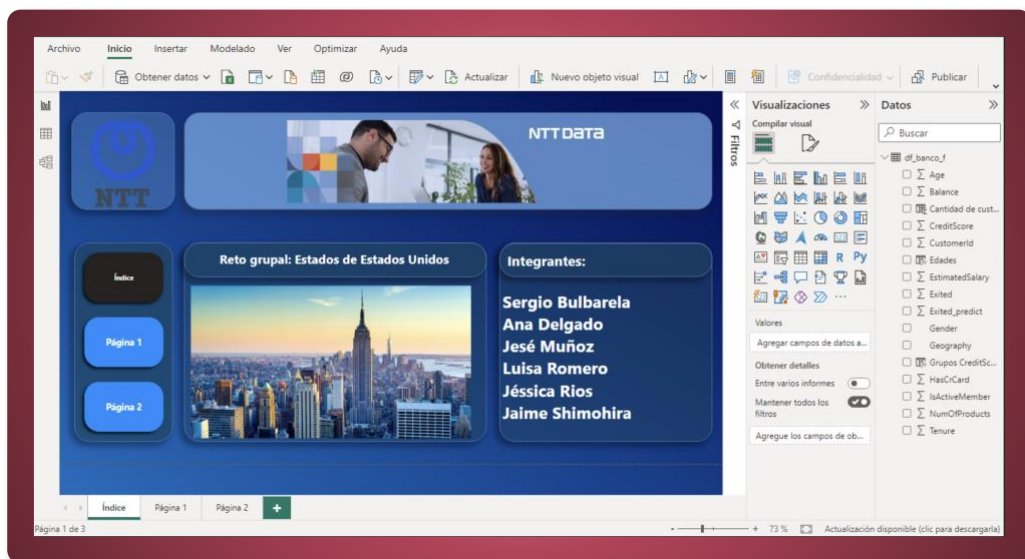
VISUALIZACIÓN DE DATOS

En el apartado de visualización de datos, hemos tratado de mostrar de una forma sencilla pero también llamativa, los resultados del análisis de datos realizado.

Para ello hemos utilizado tres hojas de POWER BI, una de ellas como página de inicio, y con botones configurados para moverse entre las mismas. El resultado ha sido un compendio de información traducida en gráficas de muy sencilla interpretación.

Adjuntamos en los anexos toda la documentación asociada al Reto IV, en donde se puede apreciar como fue el proceso de creación de este archivo de POWER BI. Asimismo, adjuntamos dicho archivo y un documento con un GIF animado donde se pueden ver los cambios de página.

Aportamos también estas imágenes correspondientes al proceso de creación del archivo final:



ANEXO: DOCUMENTACIÓN DEL DESARROLLO DE LAS ACTIVIDADES

Para finalizar, adjuntamos los enlaces correspondientes al desarrollo de cada reto y también del código creado para cada área, con las correcciones y actualizaciones que han sido necesarias:

ÁREA I:

- [RETO DOCUMENTADO](#)
- [NOTEBOOK](#)

ÁREA II:

- [RETO DOCUMENTADO](#)
- [SQL.PY](#)
- [INSERCCIÓN DE DATOS.PY](#)
- [MONGODB.PY](#)

ÁREA III:

- [RETO DOCUMENTADO CON ENLACE AL NOTEBOOK ORIGINAL](#)
- [NOTEBOOK BALANCEO](#)

ÁREA IV:

- [RETO DOCUMENTADO](#)
- [ARCHIVO POWER BI](#)
- [GIF](#)



El resumen ejecutivo fue presentado por Luisa

A grayscale photograph of a stone wall with a desk and a potted plant in the foreground. The wall is made of rough, rectangular stones. In the foreground, there is a desk with a computer monitor and a potted plant. The text "RIESGOS MATERIALIZADOS" is overlaid on the wall in white, with two horizontal bars above and below it.

RIESGOS MATERIALIZADOS

RIESGOS MATERIALIZADOS

Trabajar en equipo siempre supone un reto, y, más aun, cuando no se domina la materia y los componentes no se conocen o no han trabajado nunca juntos. Durante el desarrollo del trabajo para los distintos retos, nos hemos enfrentado a distintas dificultades que consideramos que han sido superadas con éxito.

DIFICULTADES Y SOLUCIONES

La primera de las dificultades se relaciona con las diferencias horarias no solo por estar en distintos países sino por cuestiones personales, como por ejemplo horarios de trabajo o estudio.

También encontramos diferencias de carácter y de opinión, tanto sobre el desarrollo de los ejercicios en sí, como sobre que cosas mínimas debemos de cumplir para que se conforme un buen trabajo o exista una buena comunicación, y también la forma de entender o no las necesidades de los demás.

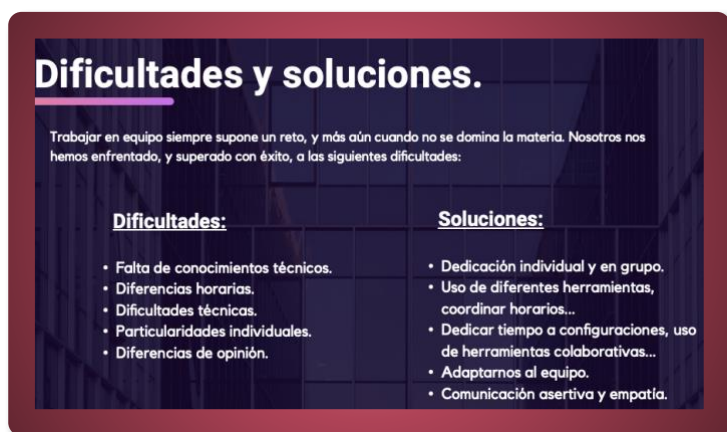
A parte de estas casuísticas que tienen, en mayor o menor medida, mucho componente personal, nos enfrentamos a que éramos un equipo en la que la mayoría de los miembros no tenía conocimientos previos relacionados con los temas tratados en los retos y también se presentaron dificultades técnicas al haber varios miembros del equipo que no trabajaban con entorno Windows, sino con Linux o macOS.

Teniendo en cuenta lo expuesto anteriormente, encontramos solución para cada una de las situaciones que se nos fueron planteando, en primer lugar, facilitando que todos estuviéramos en contacto independientemente de las diferencias horarias, buscando espacios y tiempos comunes, y cuando no era posible, teniendo acceso todos al trabajo de los demás, por ejemplo, con el uso de Google Drive y otros sistemas para compartir archivos.

También dedicamos tiempo a ayudarnos unos a otros cuando alguien encontraba un problema técnico o de configuración, de manera que los propios compañeros facilitaban el trabajo a quien presentaba la dificultad o investigaban y buscaban una solución.

La mayoría de dificultades se fueron solucionando por sí mismas con un aumento de la comunicación asertiva entre los miembros del equipo, teniendo empatía y confianza para comentar las dificultades de cada uno y apoyarse en los demás.

En términos generales, podríamos confirmar que tanto la comunicación, como la dedicación y compromiso, individual y del grupo, han sido claves para superar cualquier dificultad y para evolucionar y conformar un equipo sólido.



Presentación de Jéssica



CONCLUSIONES



CONCLUSIONES

Tras realizar el estudio de los datos correspondientes al reto grupal “Estados de Estados Unidos”, y teniendo en cuenta el objetivo final, que era poder predecir la permanencia o no de un cliente en una entidad bancaria, se determinó que era necesario estudiar en profundidad las siguientes variables y la relación entre las mismas y la variable objetivo “Exited”:

- CreditScore: Puntuación crediticia del cliente.
- Geography: Estado al que pertenece el cliente.
- Gender: En el caso de esta base de datos, si el cliente era hombre o mujer.
- Age: Edad del cliente.
- Tenure: Antigüedad del cliente en el banco.
- Balance: Balance económico del cliente
- NumOfProducts: Número de productos que el cliente tiene en el banco.
- HasCrCard: Si el cliente tiene o no tarjeta de crédito.
- IsActiveMember: Si es un miembro activo o no de la entidad.
- EstimatedSalary: Importe ingresado mensualmente por el cliente.

Asimismo, pudimos verificar que para poder alcanzar el objetivo de una forma eficaz era necesario tener en cuenta:

- Segmentación de clientes: El análisis de datos puede ayudar a segmentar a los clientes en diferentes grupos según su comportamiento, necesidades y preferencias, o incluso lugar geográfico, lo que permite al banco diseñar estrategias específicas para cada grupo.
- Modelos predictivos: El análisis de datos puede permitir la creación de modelos predictivos que permitan al cliente predecir qué clientes tienen más probabilidades de abandonar la institución, y por tanto plantear estrategias al respecto o medidas preventivas para retener a esos clientes.
- Mejora de la calidad del servicio: El análisis de datos puede identificar áreas de mejora en la calidad del servicio, lo que permite al cliente mejorar la satisfacción del cliente y reducir la tasa de deserción.

En conclusión, un análisis de datos sobre el problema de si un cliente abandona o no un banco puede proporcionar información valiosa para la toma de decisiones y la mejora de la retención de clientes, como fue en este caso el saber que tanto la edad, el número de productos contratados y el balance el cliente eran determinantes.



Imagen de la presentación de Jesé