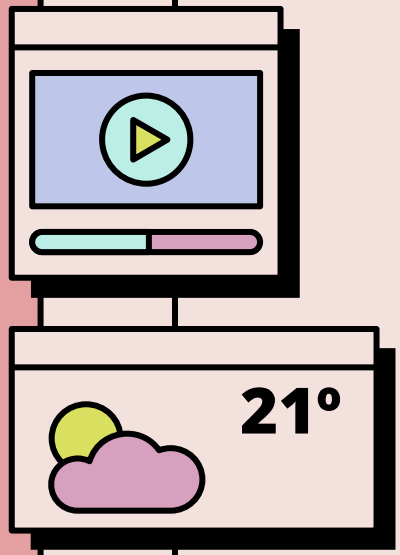
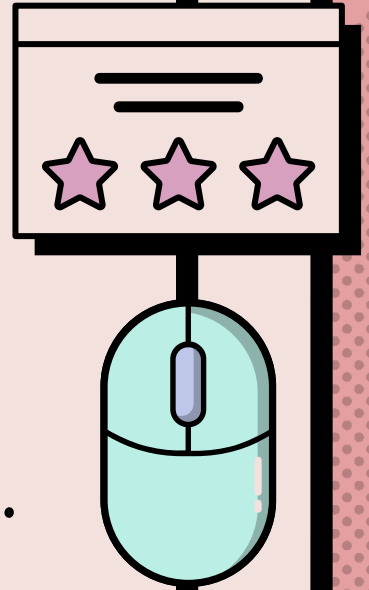
Proyecto Integrador
“Travel Predictor”





EQUIPO 7

CORDERO HERNÁNDEZ - DÍAZ AGUAYO - RODRÍGUEZ CASTRO

Contenidos

01 Introducción

Acerca de Travel Predictor

02 Modelo 1

Cassandra

03 Modelo 2

Neo4j

04 Modelo 3

MongoDB

05 Conclusiones

01

Introducción

A manera de consolidación de los conocimientos de la materia, se ha desarrollado un conjunto de herramientas para la predicción de resultados y recomendaciones para aeropuertos y aerolíneas, visto desde el punto de vista de una agencia de marketing/consultora.

Se hizo uso de los conocimientos y habilidades adquiridas a lo largo del curso, de forma que se aplica la manipulación de *bases de datos no relacionales* a una situación real con el propósito de dar solución a distintos problemas.

Modelos desarrollados

Modelo 1

Modelo que sugiere cuáles son los meses en los que es recomendable introducir campañas de publicidad para empresas de renta de carros para cada aeropuerto.

Modelo 2

Modelo que recomienda en qué aeropuertos es recomendable abrir servicios de alimentos/bebidas.

Modelo 3

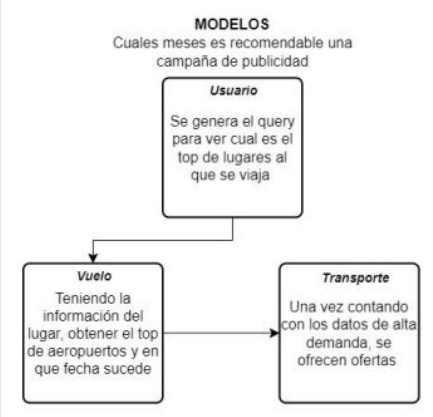
Modelo que determine picos en afluencia de pasajeros en los aeropuertos para valorar la opción de introducir descuentos o distintas estrategias comerciales.

MODELO 1

02

Modelo 1: CassandraDB

Este modelo sugiere cuáles son los meses en los que es recomendable introducir campañas de publicidad para empresas de renta de carros para cada aeropuerto, fue implementado en CassandraDB. A partir de la cantidad de pasajeros registrados por aeropuerto de destino final se cuentan y agrupan la suma para poder mostrar al personal de publicidad cuáles son los meses indicados para lanzar campañas, para esto asumimos que los meses con mayor afluencia de pasajeros serían los meses de interés.



Model.py y app.py

La aplicación está dividida en dos partes, model.py es donde se definen las consultas a realizar como creación del keyspace, tablas, consultas e inserciones.

App.py es la aplicación con la que el usuario interactúa, ésta tiene 2 opciones de ejecución

1. Se muestran los meses donde más demanda en los aeropuertos de turistas para identificar las mejores fechas para lanzar las campañas de publicidad de renta de autos
2. Se carga un dataset y se ingresa la información al cluster de CassandraDB

```
1 -- Show most travel popular months
2 -- Load dataset
3 -- Exit
```

```
1 -- Show most travel popular months
2 -- Load dataset
3 -- Exit
Enter your choice: 1
```

```
How many months: 2
```

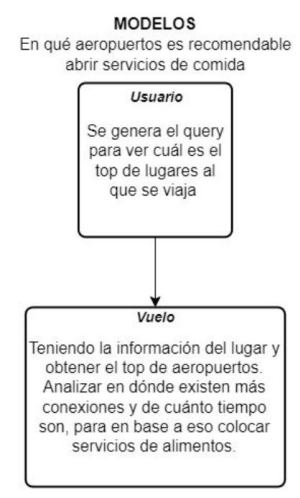
```
Retrieving the 2 most popular months for travel
=== Month: 7 ===
- Total passengers: 55
=== Month: 1 ===
- Total passengers: 50
```

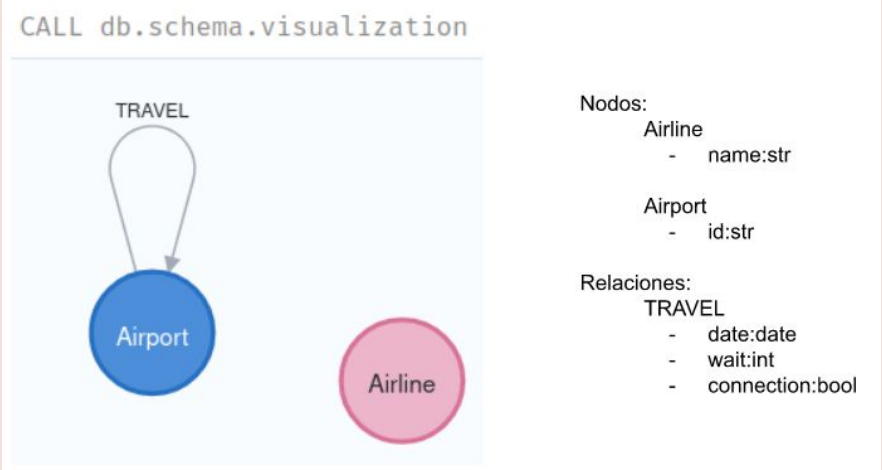
MODELO 2

Recomendación para apertura de nuevos servicios de alimento

El siguiente modelo analiza el tipo de viaje y sus conexiones directas para determinar si sería óptimo instaurar nuevos establecimientos de consumibles en los aeropuertos. Se pensó este modelo como algo fuertemente conectado, por lo tanto, la herramienta predilecta ha sido Neo4j, por la posibilidad de ofrecer consultas eficientes para el análisis de relaciones, siendo esto un concepto elemental de los grafos, es decir, el fundamento sobre el cual está construida esta base.

La forma en la que los datos son procesados hacen posible la omisión de algunas columnas en el conjunto de datos inicial, tales como género, edad, etc.





Consultas utilizadas - Predicciones

```
// Creación de constraints
CREATE CONSTRAINT airline_constraint IF NOT EXISTS FOR (al:Airline) REQUIRE al.name IS UNIQUE
CREATE CONSTRAINT airport_constraint IF NOT EXISTS FOR (ap:Airport) REQUIRE ap.id IS UNIQUE

// Limpieza inicial de la base
MATCH (n) DETACH DELETE n

// Creación de nodo tipo aerolínea
CREATE (al:Airline {name: $name})

// Creación de nodo tipo aeropuerto
CREATE (ar:Airport {id: $id})

// Creación de relación de tipo viaje entre aeropuertos
MATCH (org:Airport {id: $id_from}), (dest:Airport {id: $id_to})
MERGE (org)-[:TRAVEL {date: date($date), connection: $connection, wait: $wait}]->(dest)

// Búsqueda de resultados para predicción
MATCH (org:Airport)-[t:TRAVEL]->(dest:Airport) WHERE t.connection = "True"
[AND date >= date($date)]
[AND date <= date($date)]
RETURN org.id as ORIGIN, COUNT(dest) AS CONNECTIONS, avg(t.wait) AS WAIT_AVG
ORDER BY [WAIT_AVG, CONNECTIONS | CONNECTIONS, WAIT_AVG] DESC
[LIMIT N]
```

Consultas utilizadas - GDS

```
//* --- GDS --- *//
// Page Rank
CALL gds.graph.project(
  'mod2_page_rank',
  'Airport',
  'TRAVEL',
  {
    relationshipProperties: 'wait'
  }
)

CALL gds.pageRank.stream('myGraph')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).id AS name, score
ORDER BY score DESC, name ASC

// Closeness Centrality
CALL gds.graph.project('mod2_centrality', 'Airport', 'TRAVEL')

CALL gds.beta.closeness.stream('mod2_centrality')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).id AS id, score
ORDER BY score DESC
```

Uso de la herramienta

```
python3 travel_predictor_mod2.py
usage: travel_predictor_mod2.py [-h] [-f FILE] [-t TOP] [-r REVERSE] [-s START] [-e END] [-l LINKS] [-c CENTRALITY] {fill,predict,stats}

positional arguments:
  {fill,predict,stats}  Available application actions

options:
  -h, --help            show this help message and exit
  -f FILE, --file FILE  Dataset for database filling (csv format)
  -t TOP, --top TOP     Limit results to [top] records
  -r REVERSE, --reverse REVERSE
                        Reverse sorting order of (Waiting avg, Connection amount) factor
  -s START, --start START
                        Starting date in format (dd-mm-yyy)
  -e END, --end END     Ending date in format (dd-mm-yyy)
  -l LINKS, --links LINKS
                        Execute Page Rank algorithm for link between aiports
  -c CENTRALITY, --centrality CENTRALITY
                        Execute Closeness Centrality algorithm for node inspection
```

Ejecución de la herramienta - Predicción

```
(venv) marcordero@mrch-ubuntu:~/.../NoSQL_P2023/Proyecto/Neo4j$ python3 travel_predictor_mod2.py predict
The top 5 most suitable airports to open more establishments are the following
1 - Airport ID: DEN
        - Connections: 64
        - Waiting time average per connection: 313.66 minutes

2 - Airport ID: LAX
        - Connections: 73
        - Waiting time average per connection: 334.25 minutes

3 - Airport ID: SJC
        - Connections: 72
        - Waiting time average per connection: 349.85 minutes

4 - Airport ID: IZT
        - Connections: 82
        - Waiting time average per connection: 354.99 minutes

5 - Airport ID: MTY
        - Connections: 89
        - Waiting time average per connection: 360.93 minutes

(venv) marcordero@mrch-ubuntu:~/.../NoSQL_P2023/Proyecto/Neo4j$ python3 travel_predictor_mod2.py predict -s 08/06/2020 -e 01/01/2023 -t 3 -r True
The top 3 most suitable airports to open more establishments starting from 08-06-2020 up to 01-01-2023 are the following
1 - Airport ID: DEN
        - Connections: 12
        - Waiting time average per connection: 293.58 minutes

2 - Airport ID: PDX
        - Connections: 16
        - Waiting time average per connection: 299.00 minutes

3 - Airport ID: GDL
        - Connections: 17
        - Waiting time average per connection: 333.24 minutes
```

Ejecución de la herramienta - GDS

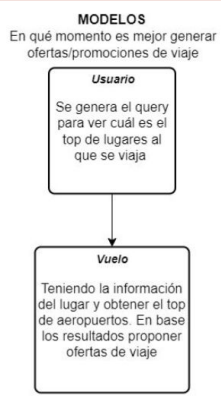
```
(venv) marcordero@mrch-ubuntu:~/.../NoSQL_P2023/Proyecto/Neo4j$ python3 travel_predictor_mod2.py stats -l 1 -c 1
-- PAGE RANK --
Aeropuerto: TPQ - Puntuación: 1.0149778466995731
Aeropuerto: JFK - Puntuación: 0.9569508379589623
Aeropuerto: PVR - Puntuación: 0.9628429593865205
Aeropuerto: GDL - Puntuación: 1.0225198006291152
Aeropuerto: PDX - Puntuación: 1.0087654978014295
Aeropuerto: DEN - Puntuación: 0.9358625858591483
Aeropuerto: SJC - Puntuación: 0.921809675058331
Aeropuerto: IZT - Puntuación: 0.9207295041544812
Aeropuerto: TLC - Puntuación: 0.9729850408760293
Aeropuerto: MTY - Puntuación: 1.0022126930572688
Aeropuerto: LAX - Puntuación: 0.8539887165894766

-- CENTRALITY --
Aeropuerto: TPQ - Puntuación: 1.0
Aeropuerto: JFK - Puntuación: 1.0
Aeropuerto: PVR - Puntuación: 1.0
Aeropuerto: GDL - Puntuación: 1.0
Aeropuerto: PDX - Puntuación: 1.0
Aeropuerto: DEN - Puntuación: 1.0
Aeropuerto: SJC - Puntuación: 1.0
Aeropuerto: IZT - Puntuación: 1.0
Aeropuerto: TLC - Puntuación: 1.0
Aeropuerto: MTY - Puntuación: 1.0
Aeropuerto: LAX - Puntuación: 1.0
```

MODELO 3

Recomendación para campañas promocionales

El último modelo alberga un funcionamiento un tanto sencillo, pero no por ello lo hace menos útil para el propósito planeado. La forma en que se manipulan los datos es mediante la obtención de dos extremos distintos de flujo de gente: puntos de mayor y menor cantidad de viajes. Según estas interpretaciones, se proponen distintas estrategias de descuentos a implantar a conveniencia de la situación.

Al requerir varias etapas de procesamiento para datos con relaciones no tan fuertes, MongoDB fue la herramienta de preferencia para concretar este requerimiento del proyecto. Su sencilla integración con Python han hecho de su uso algo bastante intuitivo y de suma utilidad.



Ajustes iniciales y consultas generales

```
// Limpieza inicial

db.[coleccion].insertMany({})

db.[coleccion].deleteMany({})


// Índices

db.travel.createIndex({'date':1})

db.travel.createIndex({'date':1, 'from':1})

db.travel.createIndex({'date':1, 'airline':1})

db.travel.createIndex({'date':1, 'from': 1, 'airline':1})

db.travel.createIndex({'from':1})

db.travel.createIndex({'airline':1})

db.travel.createIndex({'from': 1, 'airline':1})


// Obtener todos los aeropuertos

db.airport.aggregate({'$project': {'_id':0, 'id':'$id'}})


// Obtener todas las aerolíneas

db.airline.find({}, {'_id':0})
```

Consulta para predicción

```
// Consulta principal
db.travel.aggregate([
    {
        $match:
            {
                'connection': {'$ne': 'true'},
                'stay': {'$in': ['Hotel','Short-term homestay']},
                'reason': {'$in': ['On vacation/Pleasure','Back Home']},
                'airline': '$QUERY_AIRLINE$',
                'from': '$QUERY_ORIGIN_AIRPORT$',
                'date': {'$gte': ISODate('$QUERY_START_DATE$'), '$lt': ISODate('$QUERY_END_DATE$')}
            }
    },
    {
        $group: {
            '_id': {'$month': "$date"},
            'qty': {'$sum': 1}
        }
    },
    {
        $project:
            {
                '_id': 0,
                'month': '$_id',
                'qty': '$qty'
            }
    },
    {
        $sort:
            { 'qty' : -1 } }
])
```

Uso de la herramienta

```
python3 travel_predictor_mod3.py
usage: travel_predictor_mod3.py [-h] [-f FILE] [-s START] [-e END] {fill,predict}

positional arguments:
  {fill,predict}        Acciones disponibles

options:
  -h, --help            show this help message and exit
  -f FILE, --file FILE  Set de datos para el llenado de la base (formato csv)
  -s START, --start START
                        Año de inicio para rango de búsqueda
  -e END, --end END     Año de término para rango de búsqueda
```

Ejecución de la herramienta - Sin parámetros

```
Ejecución sin parámetros
python3 travel_predictor_mod3.py predict

¿Deseas un aeropuerto en específico [S/n]: n

¿Deseas una aerolínea específica? [S/n]: n

Mostrando posibles ofertas


Resultados:
Aeropuerto "TPQ"
        Posibilidad de introducción de paquetes grupales o similar
        -> Junio, Noviembre, Octubre

        Posibilidad de descuentos por baja demanda
        -> Septiembre, Febrero, Julio

Aeropuerto "JFK"
        Posibilidad de introducción de paquetes grupales o similar
        -> Febrero, Enero, Octubre

        Posibilidad de descuentos por baja demanda
        -> Abril, Junio, Marzo

Aeropuerto "PVR"
        Posibilidad de introducción de paquetes grupales o similar
        -> Marzo, Junio, Enero

        Posibilidad de descuentos por baja demanda
        -> Mayo, Octubre, Diciembre

Aeropuerto "GDL"
        Posibilidad de introducción de paquetes grupales o similar
        -> Junio, Enero, Abril

        Posibilidad de descuentos por baja demanda
```

```
        -> Febrero, Agosto, Junio

Aeropuerto "SJC"
        Posibilidad de introducción de paquetes grupales o similar
        -> Junio, Enero, Mayo

        Posibilidad de descuentos por baja demanda
        -> Abril, Noviembre, Febrero

Aeropuerto "IZT"
        Posibilidad de introducción de paquetes grupales o similar
        -> Julio, Noviembre, Enero

        Posibilidad de descuentos por baja demanda
        -> Diciembre, Abril, Mayo

Aeropuerto "TLC"
        Posibilidad de introducción de paquetes grupales o similar
        -> Febrero, Julio, Octubre

        Posibilidad de descuentos por baja demanda
        -> Noviembre, Marzo, Agosto

Aeropuerto "MTY"
        Posibilidad de introducción de paquetes grupales o similar
        -> Mayo, Marzo, Abril

        Posibilidad de descuentos por baja demanda
        -> Diciembre, Agosto, Noviembre

Aeropuerto "LAX"
        Posibilidad de introducción de paquetes grupales o similar
        -> Septiembre, Mayo, Enero

        Posibilidad de descuentos por baja demanda
        -> Febrero, Noviembre, Junio

Mes con mejor posibilidad de introducción de promociones: Febrero
```

Ejecución de la herramienta - Con parámetros

```
Ejecución con parámetros
python3 travel_predictor_mod3.py predict -s 2019 -e 2022

¿Deseas un aeropuerto en específico [S/n]:
Selecciona un aeropuerto para realizar la predicción
1 - TPQ
2 - JFK
3 - PVR
4 - GDL
5 - PDX
6 - DEN
7 - SJC
8 - IZT
9 - TLC
10 - MTY
11 - LAX
Selección: TPQ

¿Deseas una aerolínea específica? [S/n]: s
Selecciona una aerolínea
1 - Aeromar
2 - Alaska
3 - Aeromexico
4 - Lufthansa
5 - Volaris
6 - Viva Aerobus
7 - Delta Airlines
8 - American Airlines
Selección: 7
No hay recomendaciones disponibles para el conjunto de parámetros actual.
```

```
Ejecución con parámetros
python3 travel_predictor_mod3.py predict -s 2023

¿Deseas un aeropuerto en específico [S/n]: s
Selecciona un aeropuerto para realizar la predicción
1 - TPQ
2 - JFK
3 - PVR
4 - GDL
5 - PDX
6 - DEN
7 - SJC
8 - IZT
9 - TLC
10 - MTY
11 - LAX
Selección: 10

¿Deseas una aerolínea específica? [S/n]: s
Selecciona una aerolínea
1 - Aeromar
2 - Alaska
3 - Aeromexico
4 - Lufthansa
5 - Volaris
6 - Viva Aerobus
7 - Delta Airlines
8 - American Airlines
Selección: 5

Mostrando posibles ofertas
        Parámetros:
                - Desde 2023
                - Aerolínea "Volaris"
                - Aeropuerto "MTY"

Resultados:
        Posibilidad de introducción de paquetes grupales o similar
        -> Enero, Febrero

        Posibilidad de descuentos por baja demanda
        -> Enero, Febrero
```

Ejecución de la herramienta - Con parámetros

```
Ejecución con parámetros
python3 travel_predictor_mod3.py predict -e 2021

¿Deseas un aeropuerto en específico [S/n]: n

¿Deseas una aerolínea específica? [S/n]: s
Selecciona una aerolínea
1 - Aeromar
2 - Alaska
3 - Aeromexico
4 - Lufthansa
5 - Volaris
6 - Viva Aerobus
7 - Delta Airlines
8 - American Airlines
Selección: 3

Mostrando posibles ofertas
        Parámetros:
                - Hasta 2021
                - Aerolínea "Aeromexico"

Resultados:
Aeropuerto "TPQ"
        Posibilidad de introducción de paquetes grupales o similar
        -> Noviembre, Mayo, Junio

        Posibilidad de descuentos por baja demanda
        -> Mayo, Junio, Septiembre

Aeropuerto "JFK"
        Posibilidad de introducción de paquetes grupales o similar
        -> Febrero, Diciembre

        Posibilidad de descuentos por baja demanda
        -> Febrero, Diciembre
```

```
Aeropuerto "SJC"
        Posibilidad de introducción de paquetes grupales o similar
        -> Mayo

        Posibilidad de descuentos por baja demanda
        -> Mayo

Aeropuerto "IZT"
        Posibilidad de introducción de paquetes grupales o similar
        -> Junio

        Posibilidad de descuentos por baja demanda
        -> Junio

Aeropuerto "TLC"
        Posibilidad de introducción de paquetes grupales o similar
        -> Abril, Diciembre, Mayo

        Posibilidad de descuentos por baja demanda
        -> Mayo, Octubre, Enero

Aeropuerto "MTY"
        Posibilidad de introducción de paquetes grupales o similar
        -> Junio, Octubre, Mayo

        Posibilidad de descuentos por baja demanda
        -> Octubre, Mayo, Marzo

Aeropuerto "LAX"
        Posibilidad de introducción de paquetes grupales o similar
        -> Septiembre, Enero

        Posibilidad de descuentos por baja demanda
        -> Septiembre, Enero

Mes con mejor posibilidad de introducción de promociones: Mayo
```

05

Conclusiones

Las herramientas utilizadas, en conjunto de las tecnologías que las soportan, han sido de suma utilidad para lograr el cometido del proyecto.

Más importante que lo anterior, contar con las habilidades requeridas para llevar a cabo este desarrollo, serán útiles en el futuro, en un contexto laboral real para dar solución a problemáticas del mundo moderno.

Código completo

Muchas Gracias