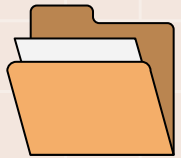
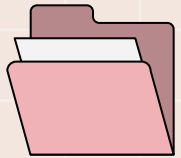
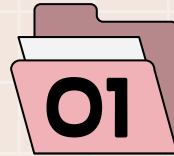


E.T.L. Con DynamoDB

David M.
Elizabeth E.
Izzy D.



CONTENIDO



INTRO



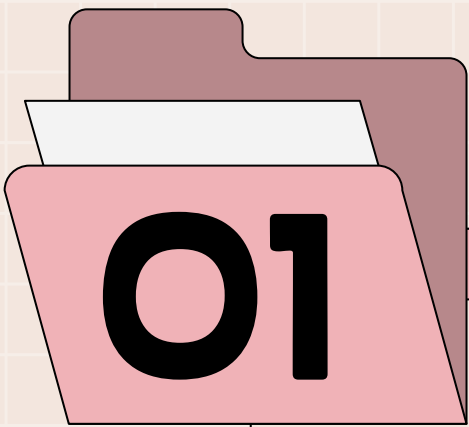
LA BASE DE
DATOS



CONSULTAS



DEMO



INTRO



Planteamiento del Problema

- En el mundo del e-commerce y retail, las tiendas necesitan entender mejor a sus clientes para ofrecer productos personalizados y mejorar la experiencia de compra.
- La recopilación y transformación de datos de consumo es clave para generar estrategias de recomendación eficaces.
- En este proyecto, diseñamos e implementamos un flujo ETL (Extracción, Transformación y Carga) en AWS para procesar datos de consumo y generar recomendaciones inteligentes.



El Objetivo:

- Llevar el historial de compras de un retailer online de tablas csv a una base de datos NoSQL.
- Poder servir datos sobre productos y órdenes recientes a los clientes a través de una página web.
- Implementar un sistema de recomendación basado en ratings de productos, generado con Python.
- Facilitar datos para monitorear el comportamiento de los clientes y tomar decisiones estratégicas para mejorar las ventas.

Explicación del Flujo ETL en AWS

Extracción

- Se reciben datos en formato CSV con información de productos y clientes.
- Estos datos se almacenan en AWS S3 como punto de entrada al sistema.
- También se reciben datos de la web con la puntuación del cliente a su producto adquirido.

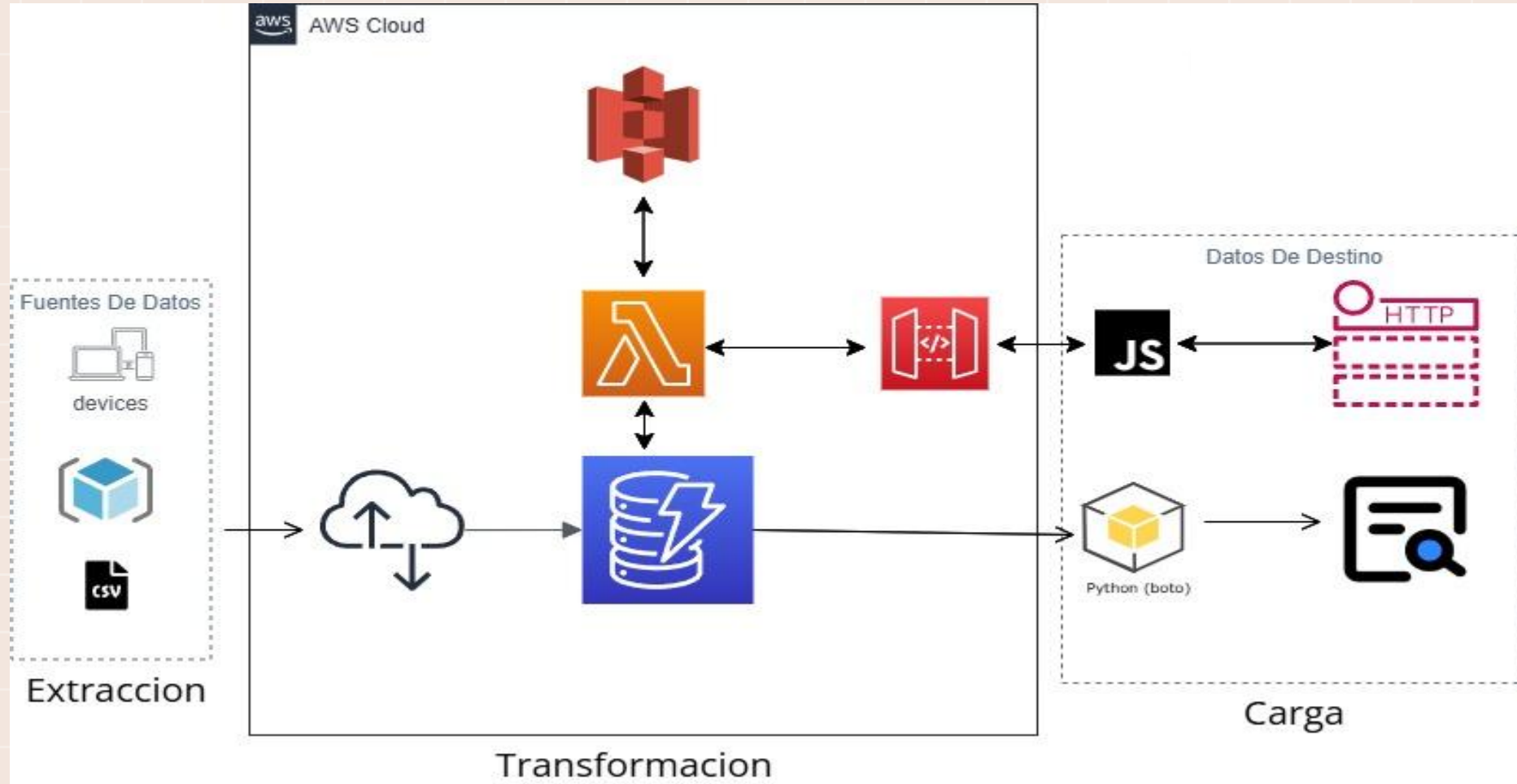
Transformación

- AWS Lambda procesa los datos, interactuando con DynamoDB para almacenar y actualizar la información.
- Se utiliza API Gateway para exponer los datos y permitir su consulta en la web.

Carga

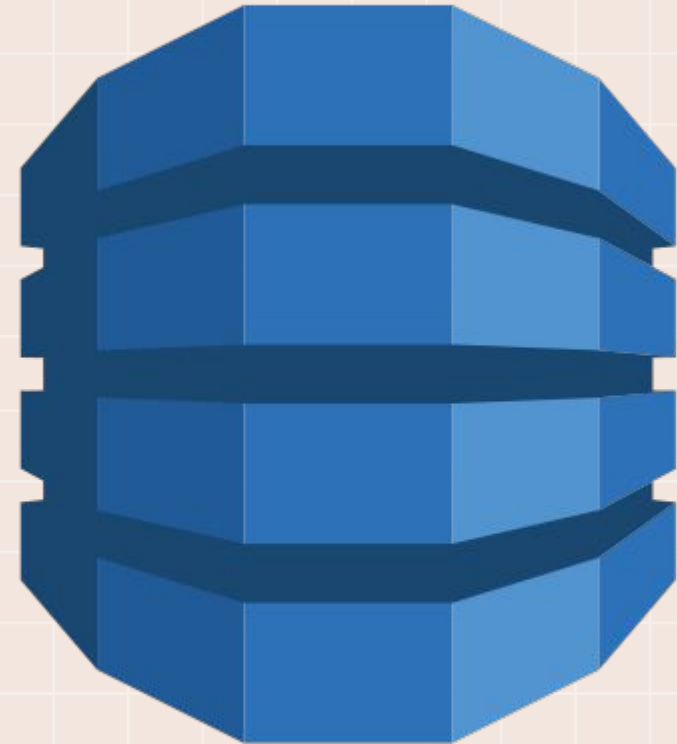
- Se recopilan las calificaciones de los productos a través de la interfaz web.
- Un modelo en Python procesa estos datos y genera recomendaciones basadas en patrones de compra.

Arquitectura ETL



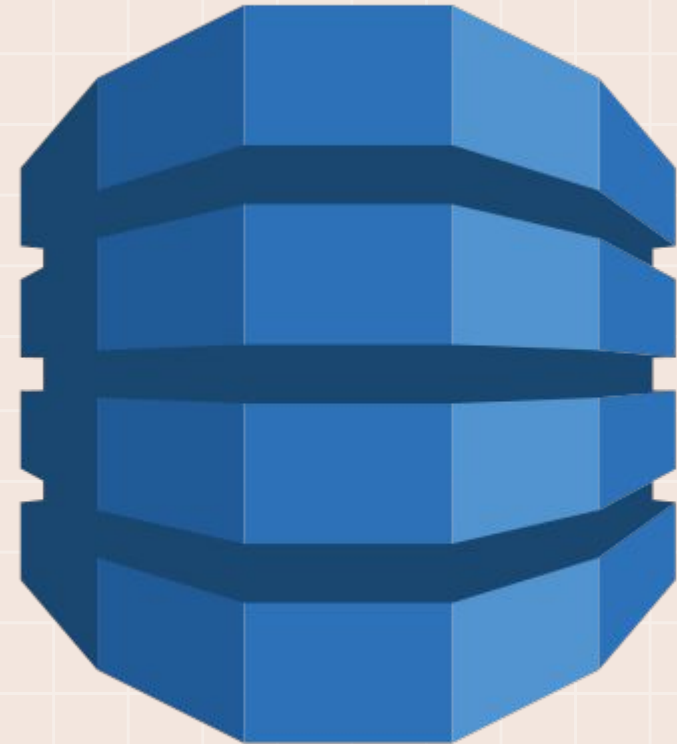
DynamoDB

- Base NoSQL de tablas
 - Partition Key
 - Atributos (string, maps, numbers, binary, bool, sets)
- Schemeless y Serverless
- Pocas Tablas
- Query a través de un SDK y API



DynamoDB

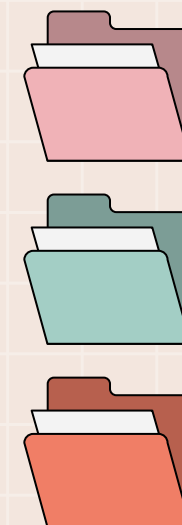
- Rápido acceso a los datos
- Podemos insertar órdenes y ratings en la misma tabla



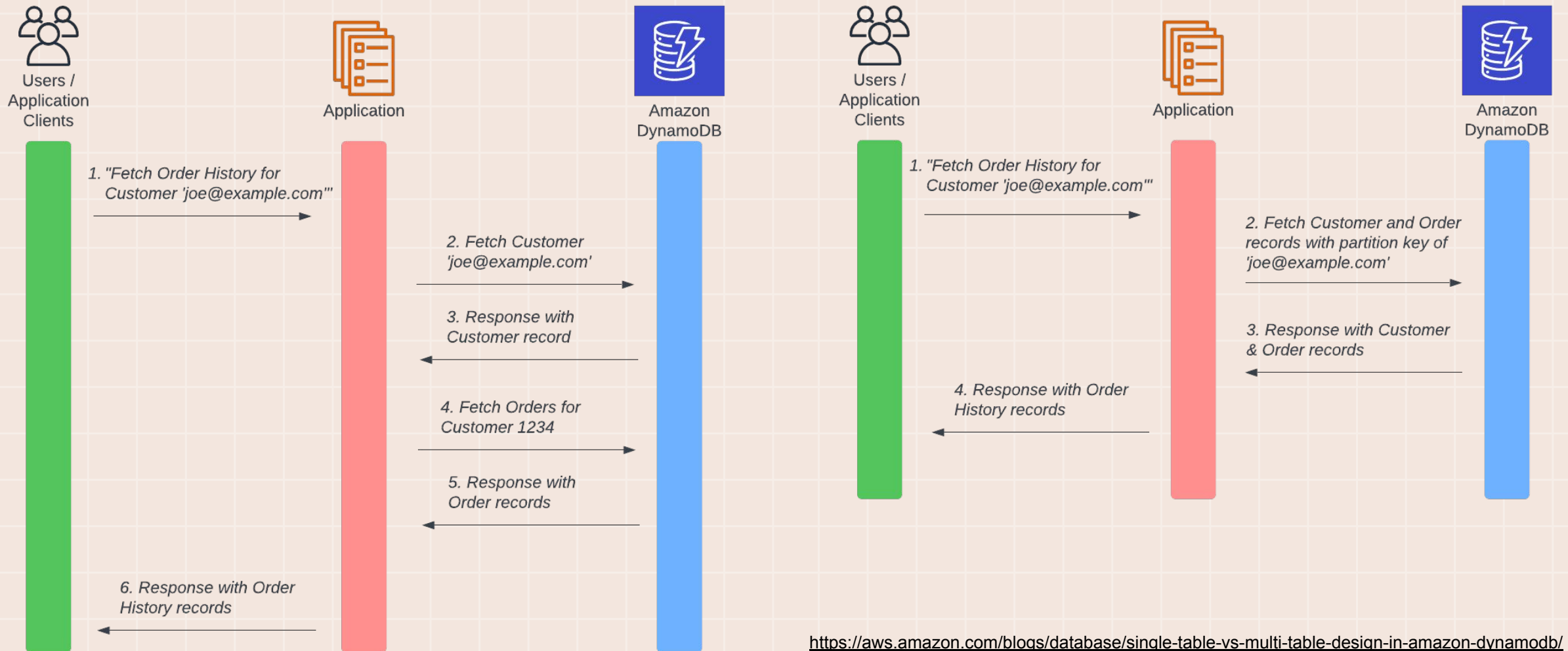


BASE DE DATOS

Esquema y Estructura



TABLAS: UNA O VARIAS



<https://aws.amazon.com/blogs/database/single-table-vs-multi-table-design-in-amazon-dynamodb/>

ETL

Order ID	Order Date	Product ID	Product Category	Buyer Gender	Buyer Age	Order Location	International Shipping	Sales Price	Shipping Charges	Sales per Unit	Quantity	Total Sales	Rating	Review	ClientID
189440	2024-07-21	BF8238	Clothing	Male	30	New Jersey	No	100	0	100	1	100	4	The delivery team handled the product with care.	509-47-7760
187385	2024-07-20	BF2152	Clothing	Male	32	Las Vegas	No	100	0	100	1	100	3	Had slight delays but the product was in good shape.	334-69-6217
181844	2024-07-21	BF8238	Other	Female	26	Cardiff	Yes	9	40	49	1	49	2	Waste of Money.	357-58-7458
197934	2024-08-19	BF4831	Other	Male	28	Pittsburgh	No	9	0	9	2	18	3	Had slight delays but the product was in good shape.	709-11-7996
122470	2024-01-06	BF6975	Other	Female	19	Miami	No	10	0	10	3	30	5	Lack of delivery delays is greatly appreciated.	458-54-7302
199680	2024-06-01	BF6693	Clothing	Male	29	Sydney	Yes	118	100	218	1	218	1	The product had an underwhelming design.	725-09-9845
130929	2024-03-06	BF8238	Ornaments	Male	32	Memphis	No	32	0	32	3	96	4	Premium quality as usual.	687-46-1501
164427	2024-04-16	BF0060	Ornaments	Male	26	Miami	No	32	0	32	2	64	3	Not as expected.	887-24-1826
149510	2024-05-22	BF0809	Other	Female	26	New York	No	10	0	10	1	10	4	Lack of delivery delays is greatly appreciated.	844-94-2027
176280	2024-02-27	BF0060	Clothing	Male	20	Sydney	Yes	65	100	165	1	165	4	Great design is one of the major positives of this product.	203-21-0359
196659	2023-11-05	BF2370	Clothing	Female	34	Montreal	Yes	118	25	143	1	143	5	The product was delivered quickly.	328-37-9085
146245	2024-09-24	BF0103	Clothing	Male	21	Sacramento	No	65	0	65	1	65	1	There is clear evidence of subpar detailing.	220-57-5020
150560	2024-10-04	BF6604	Clothing	Male	29	Las Vegas	No	118	0	118	3	354	5	The product was delivered quickly.	864-15-9356
138348	2024-01-26	BF2887	Clothing	Male	26	New Jersey	No	130	0	130	2	260	2	The product had an underwhelming design.	543-38-1711
138790	2024-05-13	BF1681	Ornaments	Male	24	Montreal	Yes	32	25	57	2	114	3	Average product	555-45-3873
170025	2023-12-03	BF4831	Clothing	Male	34	Paris	Yes	97	50	147	1	147	5	Exquisite craftsmanship here is absolutely appreciated.	339-54-7906

ID	Name	Email
396-56-6891	Hertha Jiles	hjiles0@ed.gov
330-37-2285	Brittani Markwelley	bmarkwelley1@nytimes.com
235-27-7937	Devonna Weekley	dweekley2@flickr.com
409-93-9196	Candace Conrart	cconrart3@canalblog.com
472-06-9675	Dew Johannes	djohannes4@symantec.com
522-63-1669	Falkner Blenkiron	fblenkiron5@reverbnation.com
272-12-1718	Shermie Gynni	sgynni6@independent.co.uk
370-83-3379	Brunhilda Charles	bcharles7@cbsnews.com
453-96-9291	Leelah Kineton	lkineton8@amazonaws.com
497-35-8268	Steve Leeming	sleeming9@ehow.com
221-87-8673	Aprilette Titman	atitmana@slashdot.org
584-98-1355	Ceciley Rizzello	crizzello@ifeng.com
756-86-1836	Bev Mulcahy	bmulcahy@narod.ru
171-84-0997	Jeanne Mellhuish	jmellhuishd@xinhuanet.com
324-83-1265	Isacco Senyard	isenyarde@goo.gl
542-53-8550	Denis Lumsdall	dlumsdallf@wordpress.com
246-22-6728	Kateen Ofener	kofenerg@abc.net.au
414-04-8076	Juan Vayro	jvayroh@discovery.com
197-92-8096	Dal Mosson	dmossoni@prweb.com
550-76-7280	Marv Meigh	mmeigh@mlb.com
778-84-4011	Victor McSaul	vmcsaulk@illinois.edu
324-84-7749	Adele Dwane	adwanel@virginia.edu
358-67-8296	Astra Downey	adowneym@barnesandnoble.com
485-10-0773	Akvis Pavicur	apavicurm@ask.com

ProductID	Producto	descripcion
BF1414	Camisa Estampada	Camiseta en algodón con estampado blanco fresco y combinable para cualquier ocasión casual.
BF2887	Zapatos Beige	Los zapatos tienen un diseño moderno y están hechos con material transpirable
BF0103	Pantalones Claros	vaqueros de talla alto en color azul claro con un corte recto. Los jeans tienen un diseño clásico de cinco bolsillos y están ligeramente enrollados en los puños.
BF7814	Gafas Transparentes	Gafas con una montura transparente o de color claro. Las gafas tienen una forma clásica
BF6604	Collar Acero	Collar delicado y versatil para el día a día que daran un toque a tu outfit.
BF9708	Bufanda Tejida	Bufanda de tela con un patrón de cuadros en tonos de gris y negro. La bufanda tiene un diseño de tejido entrelazado y bordes deshilachados
		Falda en jean con un bordado de flores que añade un elemento decorativo y estiloso.
BF2370	Falda Jean	
BF8063	Vestido Rojo Estampado	Vestido rojo con puntos blancos con un estilo clasico pero juvenil a su vez.
BF4831	Camisa Sin Mangas	Camiseta en algodón sin mangas muy apropiada para días mas calurosos que permiten mayor comodidad y frescura.
BF9878	Zapatos Altos Blancos	Zapatos tipo tenis blancos con cordones anchos cómodos y versátiles.
BF1772	Pantalones Skinny Oscuro	jeans de color gris oscuro doblados. Los jeans tienen un bolsillo trasero visible y una etiqueta en la pretina.
BF6693	Gafas Clásicas	Estas gafas destacan por ser un accesorio elegante y estilo* combinando diseño estético e identidad de marca.
BF3942	Collar Perlas	Collar tipo perlas sofisticado que daran un toque de elegancia a tu estilo.
BF0060	Bufanda Rayas	La bufanda presenta un elegante diseño suave y ligera lo que la hace cómoda y adecuada para el uso diario. Perfecta para darle un toque de estilo a cualquier atuendo
BF2152	Falda Estampada Flores	falda roja con flores subliminadas de talla alto y corta.
BF6975	Vestido Cinturon	Vesddio color mostaza con un cinturón que le da un toque delicado.
BF8238	Camisa Estampada B/W	Camiseta en algodón con estampado de color con un contraste que crean una sinergia perfecta.
BF0809	Zapatos Multicolor	zapatos con una combinación de colores perfecta y cómodos para el día a día.
BF1681	Pantalones Bota Ancha	jeans de mezclilla azul con cintura alta y piernas anchas. Los jeans tienen dos grandes bolsillos frontales y un cierre de botón con cremallera.
BF6885	Gafas Rosados	gafas de sol redondas con lentes de color rosa. La montura* con un puente y patillas delgadas. Las lentes rosas le dan a las gafas un aspecto elegante y moderno.
BF5238	Collar Choker	Collar tipo choker atrevido y perfecto para complementar tu estilo.
BF6167	Bufanda Tejida Beige	Bufanda tejida perfecta para días fríos con un diseño muy versatil que permitira su uso diario.
BF9724	Falda Cuero	Falda tipo cuero cintura alta y presenta pliegues que añaden volumen a la prenda.
BF9289	Vestido Flores	Vestido florecido con una combinación de colores perfecta.

ETL: TRANSFORMACIÓN TABLA ORDENES

```
import pandas as pd

# Leer los archivos CSV
clientes_df = pd.read_csv(r"C:/Users/DAVID MARTINEZ/OneDrive/Desktop/maestria/00 Bases de datos de nueva generacion/Trabajos/Tarea Final Grupal/MOCK_CLIE
ordenes_df = pd.read_csv(r"C:/Users/DAVID MARTINEZ/OneDrive/Desktop/maestria/00 Bases de datos de nueva generacion/Trabajos/Tarea Final Grupal/OrdenesOri

# Realizar el merge en base a la columna 'ClientID'
result_df = pd.merge(ordenes_df, clientes_df[['ClientID', 'Name', 'Email']], on='ClientID', how='left')

# Guardar el resultado en un nuevo archivo CSV
result_df.to_csv(r"C:/Users/DAVID MARTINEZ/OneDrive/Desktop/maestria/00 Bases de datos de nueva generacion/Trabajos/Tarea Final Grupal/OrdenesConClientes

# Verificar las primeras filas del resultado
print(result_df.head())
```

ETL

Attributes

Add new attribute ▾

Attribute name	Value	Type	
OrderID - Partition key	163287	String	
<input type="text" value="BuyerAge"/>	<input type="text" value="29"/>	Number	<input type="button" value="Remove"/>
<input type="text" value="BuyerGender"/>	<input type="text" value="Female"/>	String	<input type="button" value="Remove"/>
<input type="text" value="ClientID"/>	<input type="text" value="601355057"/>	String	<input type="button" value="Remove"/>
<input type="text" value="Email"/>	<input type="text" value="edemeadaa@skyrock.com"/>	String	<input type="button" value="Remove"/>
<input type="text" value="InternationalShipping"/>	<input type="text" value="Yes"/>	String	<input type="button" value="Remove"/>
<input type="text" value="NameClient"/>	<input type="text" value="Eachelle Demead"/>	String	<input type="button" value="Remove"/>
<input type="text" value="OrderDate"/>	<input type="text" value="7/03/2024"/>	String	<input type="button" value="Remove"/>
<input type="text" value="OrderLocation"/>	<input type="text" value="Sydney"/>	String	<input type="button" value="Remove"/>
<input type="text" value="ProductID"/>	<input type="text" value="BF2370"/>	String	<input type="button" value="Remove"/>
<input type="text" value="Quantity"/>	<input type="text" value="1"/>	Number	<input type="button" value="Remove"/>
<input type="text" value="Rating"/>	<input type="text" value="4"/>	Number	<input type="button" value="Remove"/>
<input type="text" value="Review"/>	<input type="text" value="Exquisite craftsmanship here is absolutely appreciated."/>	String	<input type="button" value="Remove"/>
<input type="text" value="SalesPerUnit"/>	<input type="text" value="115"/>	Number	<input type="button" value="Remove"/>
<input type="text" value="SalesPrice"/>	<input type="text" value="15"/>	Number	<input type="button" value="Remove"/>
<input type="text" value="ShippingCharges"/>	<input type="text" value="100"/>	Number	<input type="button" value="Remove"/>
<input type="text" value="TotalSales"/>	<input type="text" value="115"/>	Number	<input type="button" value="Remove"/>


ETL

Attributes

Add new attribute ▼

<div><div></div>Attribute name</div>	Value	Type	
ProductID - Partition key	BF1772	String	
categoria	Prenda	String	Remove
descripcion	jeans de color gris oscuro, doblados. Los jeans tienen un bolsillo trasero visible y una etiqueta en la pretina.	String	Remove
precio	41	Number	Remove
Producto	Pantalones Skinny Oscuro	String	Remove
urls3	https://productosizzzyshop.s3.us-east-1.amazonaws.com/panatalon2.JPG	String	Remove

ETL

 query - python query.py

```
ordenes = dynamodb.Table('Ordenes')
```


ETL

BDNG final / AllItems

Save Share

GET

https://7s98pr5chh.execute-api.eu-west-3.amazonaws.com/tienda/catalogo/?query=AllItems

Send

Params

Authorization

Headers (6)

Body


Scripts

Settings

Cookies

Response

History



Click Send to get a response

Postbot

Runner

Start Proxy

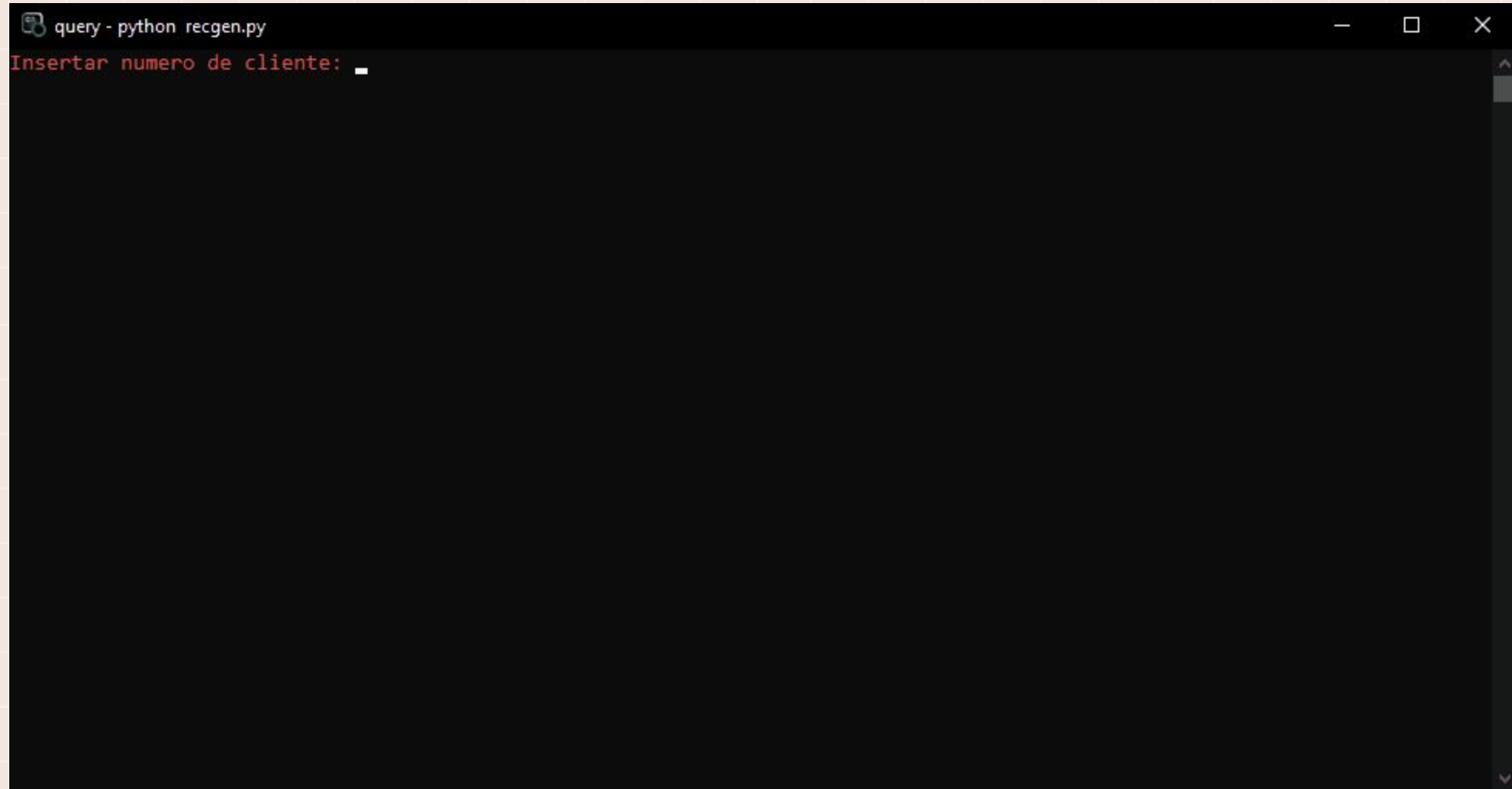
Cookies

Vault

Trash

?

ETL



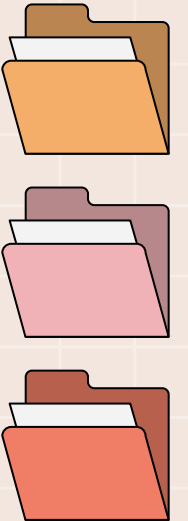
A terminal window titled "query - python recgen.py" is displayed. The prompt "Insertar numero de cliente:" is shown in red text, followed by a white cursor. The terminal has a black background and a vertical scrollbar on the right side.

```
query - python recgen.py
Insertar numero de cliente: _
```



CONSULTAS

Analisis, API calls y Modelo de Recomendaciones



QUERY

Productos Más Vendidos

```
Ordenes = dynamodb.Table('Ordenes')

# Escanear la tabla para obtener Product ID y Quantity
response = Ordenes.scan(
    ProjectionExpression="ProductID, Quantity"
)

# Diccionario para almacenar el total vendido por producto
ventas_por_producto = defaultdict(Decimal)

# Sumar la cantidad vendida de cada producto
for item in response['Items']:
    product_id = item['ProductID'] # ID del producto
    quantity = Decimal(item['Quantity']) # Convertir a número decimal
    ventas_por_producto[product_id] += quantity # Sumar la cantidad vendida

# Ordenar los productos por cantidad vendida (de mayor a menor)
productos_ordenados = sorted(ventas_por_producto.items(), key=lambda x: x[1], reverse=True)

# Mostrar los productos más vendidos
print("Productos más vendidos:")
for product, total in productos_ordenados[:10]: # Mostrar solo los 10 primeros
    print(f"Producto {product}: {total} unidades vendidas")
```

Productos más vendidos:

```
Producto BF9878: 274 unidades vendidas
Producto BF9724: 270 unidades vendidas
Producto BF1414: 264 unidades vendidas
Producto BF8063: 261 unidades vendidas
Producto BF7814: 261 unidades vendidas
Producto BF9289: 260 unidades vendidas
Producto BF1772: 260 unidades vendidas
Producto BF0060: 257 unidades vendidas
Producto BF8238: 256 unidades vendidas
Producto BF6975: 252 unidades vendidas
```

QUERY

Total Comprado Por Cliente

```
Ordenes = dynamodb.Table('Ordenes')

# Escanear la tabla 'Ordenes' y obtener ClientID y Total Sales
response = Ordenes.scan(
    ProjectionExpression="ClientID, NameClient, TotalSales"
)

# Procesar resultados
ventas_por_cliente = defaultdict(Decimal) # Diccionario para almacenar sumas

for item in response['Items']:
    client_id = item['ClientID'] # ID del cliente
    total_sales = Decimal(item['TotalSales']) # Convertir a número decimal
    ventas_por_cliente[client_id] += total_sales # Sumar ventas

# Mostrar total de ventas por cliente
print("Total de ventas por cliente:")
for client, total in ventas_por_cliente.items():
    print(f"Cliente {client}: ${total}")
```

```
Total de ventas por cliente:  Cliente 645989146: $397
                             Cliente 601355057: $475
                             Cliente 190499590: $948
                             Cliente 679633181: $766
                             Cliente 153154348: $1217
                             Cliente 220575020: $441
                             Cliente 612161326: $998
                             Cliente 464759072: $65
                             Cliente 824847749: $1079
                             Cliente 331534109: $694
                             Cliente 848979759: $590
                             Cliente 862285465: $427
                             Cliente 750662495: $772
                             Cliente 623388320: $563
                             Cliente 749416055: $449
                             Cliente 482320644: $1050
                             Cliente 617513041: $886
                             Cliente 243789138: $190
                             Cliente 445703916: $1021
                             Cliente 428373966: $77
                             Cliente 896030670: $130
                             Cliente 119084313: $107
                             Cliente 209633074: $475
                             Cliente 710450932: $993
                             Cliente 462958476: $1767
                             Cliente 364434994: $359
                             Cliente 478570128: $749
                             Cliente 600465334: $487
                             Cliente 308589279: $215
                             Cliente 604165697: $637
                             Cliente 800348499: $784
                             Cliente 784418358: $1384
                             Cliente 358678296: $375
```


QUERY

Total Ventas Por Producto

```
Ordenes = dynamodb.Table('Ordenes')

# Escanear la tabla 'Ordenes' y obtener Product ID y Total Sales
response = Ordenes.scan(
    ProjectionExpression="ProductID, TotalSales"
)

# Procesar resultados
ventas_por_producto = defaultdict(Decimal) # Diccionario para almacenar sumas

for item in response['Items']:
    product_id = item['ProductID'] # Ajustado al nombre correcto
    total_sales = Decimal(item['TotalSales']) # Convertir a número decimal
    ventas_por_producto[product_id] += total_sales # Sumar ventas

# Mostrar total de ventas por producto
print("Total de ventas por producto:")
for product, total in ventas_por_producto.items():
    print(f"Producto {product}: ${total}")
```

Producto BF6693: \$17173	Producto BF7814: \$18083
Producto BF2887: \$15647	Producto BF5238: \$15927
Producto BF6975: \$15771	Producto BF7814: \$18083
Producto BF2152: \$14706	Producto BF5238: \$15927
Producto BF9878: \$20009	Producto BF5238: \$15927
Producto BF0060: \$17611	Producto BF9724: \$18790
Producto BF9878: \$20009	Producto BF0809: \$13553
Producto BF0060: \$17611	Producto BF0809: \$13553
Producto BF0060: \$17611	Producto BF6604: \$18315
Producto BF6167: \$18636	Producto BF1772: \$19514
Producto BF3942: \$17243	Producto BF6885: \$14402
Producto BF7814: \$18083	Producto BF4831: \$11463
Producto BF5238: \$15927	Producto BF9708: \$14962
Producto BF3942: \$17243	

MODELO

ClientID	BF1772	BF2152	BF1681	BF2370	BF9289	BF8238	BF0060	BF9878	BF0809	BF7814	BF8063	BF6885	BF5238	BF2887	BF4831	BF9724	BF6693	BF1414	BF0103	BF6167	BF6975	BF6604	BF3942	BF9708
100071193	0	0	0	5	0	0	0	4	0	4	0	5	0	0	0	4	4	2	4	0	0	3	1	2
102228703	0	0	0	0	4	0	0	5	0	0	0	3	0	0	0	4	5	0	0	0	0	4	0	0
102320254	0	5	5	0	0	0	0	5	0	0	0	0	4	0	0	0	5	5	0	0	0	5	0	0
103076257	1	5	0	0	5	0	0	0	0	4	4	0	5	0	0	4	3	5	0	1	0	0	0	0
103692255	0	0	4	0	0	0	5	0	0	0	0	0	0	4	0	3	1	0	0	0	0	0	5	0
105120468	0	0	0	0	0	1	0	0	0	0	3	0	0	0	0	1	0	0	0	0	3	0	0	0
109928560	0	3	4	0	3	0	0	0	0	0	0	5	0	0	1	0	0	5	3	0	0	5	2	0
110171128	0	3	0	4	1	0	0	0	0	0	3	1	4	4	5	0	4	0	0	0	3	5	0	0
110552209	0	0	0	0	0	0	5	0	5	5	5	4	0	0	0	0	0	0	0	4	0	0	0	4
111080176	0	3	0	0	0	1	3	4	0	0	0	1	0	3	0	1	0	0	0	0	0	4	1	0
111370443	0	0	0	3	5	0	2	0	0	5	0	0	0	0	3	0	0	0	5	3	5	5	0	0
111532088	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	5	4	1	0	1	1	2	0	0
112087671	0	0	5	0	0	5	4	0	5	3	4	0	0	5	2	0	5	0	0	4	0	0	1	3
112137859	0	0	0	4	0	3	0	0	0	3	0	0	0	0	0	1	0	4	1	1	0	0	0	0
113848143	0	4	0	0	5	0	0	0	0	0	0	0	0	0	5	2	4	0	5	5	5	0	5	0
114084722	4	0	0	0	0	3	4	0	0	5	0	0	0	4	0	0	4	0	5	0	1	0	4	3
114147579	0	3	5	0	4	0	4	2	0	0	0	0	4	0	0	0	0	0	0	0	4	0	0	0
114940045	4	0	5	4	0	0	0	0	0	0	0	5	0	5	5	0	0	0	5	0	0	5	0	4
116579465	0	0	0	0	5	0	0	5	0	0	0	0	5	3	0	4	0	1	3	0	0	2	0	4
117425588	0	4	0	0	5	4	0	0	0	5	4	3	0	0	5	1	4	0	0	0	0	0	0	2
118893962	0	0	0	0	0	0	2	0	0	0	3	0	0	0	4	4	0	4	1	0	1	4	4	0
119084313	3	0	0	0	0	0	5	0	0	5	0	0	3	4	0	0	5	0	5	0	0	3	0	0
119934948	0	5	5	0	5	4	0	0	0	1	2	0	0	0	0	4	0	5	0	0	5	0	0	0
120309319	4	0	0	5	5	0	4	0	0	2	0	0	4	0	0	0	3	5	5	0	0	4	0	0
120912556	0	1	0	0	0	4	0	3	0	0	0	0	1	0	1	4	5	0	5	0	0	0	0	1
122144442	0	0	2	0	0	0	5	0	4	0	4	0	3	0	0	0	0	3	0	4	0	0	5	0
123491556	5	0	0	5	0	5	0	0	0	0	0	0	0	0	1	5	0	4	1	5	0	3	0	0
124309706	1	1	4	4	5	0	0	0	0	0	0	0	0	0	4	2	0	5	0	4	0	5	0	0
125149462	0	1	2	4	0	0	1	0	0	0	0	0	2	0	0	4	0	0	0	1	4	5	0	0
125799118	4	4	0	5	5	0	0	0	0	0	0	0	4	0	0	0	2	3	4	5	4	0	0	0
127169052	5	0	0	5	4	0	0	0	0	5	0	4	4	0	5	0	4	5	2	4	0	3	4	0
129120976	3	5	1	0	0	5	0	0	5	4	1	4	0	5	0	0	0	5	0	1	1	5	0	0
129645212	0	4	1	5	0	2	0	0	4	0	0	5	0	0	0	0	0	0	0	5	5	4	0	0
129944407	0	0	0	0	0	0	0	2	5	4	4	0	0	0	0	0	0	0	0	4	0	0	0	0
130368545	0	0	0	0	5	5	0	0	0	1	1	0	0	0	0	4	0	0	0	0	0	0	0	0
132435420	0	0	2	0	4	0	5	0	0	0	0	4	0	0	5	0	0	1	0	0	3	0	4	2
133445159	0	0	0	4	0	0	4	0	0	0	0	0	5	0	0	0	5	0	0	0	0	1	5	0
134382923	0	2	5	3	0	0	5	0	0	3	0	0	1	3	0	3	2	3	0	0	0	3	0	0
134500410	3	5	4	3	4	0	4	0	4	0	0	5	0	0	2	4	0	5	0	0	0	3	1	0
136522981	0	4	3	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	0	0	0
136792598	0	0	0	0	0	0	5	0	0	1	0	0	0	1	0	0	0	1	0	0	0	5	0	0
136800865	3	0	0	0	4	0	0	5	0	4	0	0	5	0	0	0	0	0	0	0	0	0	0	4
136889015	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0	0	0	5
137088565	4	0	0	5	0	3	4	0	5	5	4	0	4	0	0	4	1	0	0	0	1	3	5	0
137880967	0	0	0	2	0	0	4	0	3	3	0	0	0	0	0	4	4	1	0	0	0	0	0	0
139056963	4	0	0	4	0	0	5	4	2	0	0	0	0	5	0	0	5	0	0	0	4	0	0	0
141537827	0	0	0	0	0	5	0	0	0	0	0	0	5	0	0	0	5	0	0	0	5	0	3	0
141960488	0	0	4	0	4	0	0	2	0	5	0	0	0	0	0	0	0	0	4	1	0	4	4	5
143937463	0	0	0	3	0	5	0	0	0	3	3	0	0	0	3	0	0	0	4	5	0	0	0	5
147191794	0	5	0	4	0	0	0	0	4	5	0	0	0	5	0	2	0	4	0	0	0	0	4	0
148802088	0	0	5	0	0	0	0	0	0	4	0	4	1	4	5	0	5	0	0	5	0	0	0	4
151207704	0	0	5	5	0	1	0	0	0	3	0	0	0	4	0	0	5	5	0	0	0	4	5	0
153154348	3	4	4	0	0	5	4	0	4	3	0	4	0	5	0	1	0	0	0	0	4	0	4	0
153732125	4	0	0	0	4	1	4	0	1	0	0	2	4	0	4	0	0	0	0	0	0	0	0	4
154082394	0	0	5	4	5	0	0	0	1	0	0	4	0	1	0	2	0	2	0	0	0	0	0	4
154222290	4	0	0	4	0	0	5	0	2	0	0	0	0	0	0	0	4	0	0	0	4	0	0	4
154965625	0	0	0	0	0	0	0	0	2	1	5	4	0	5	0	5	0	0	0	0	3	0	0	0
155589871	0	0	0	0	0	4	0	0	0	4	0	0	4	1	5	0	0	5	0	0	3	0	0	0
156890316	0	0	1	3	0	0	0	5	0	4	0	0	4	5	4	5	0	0	4	0	0	2	0	0
158152614	0	0	5	5	0	5	0	0	3	0	0	0	0	0	0	0	1	0	0	5	3	4	0	0
158814357	0	1	4	0	0	2	4	0	0	0	5	0	0	0	0	0	1	0	0	3	1	0	0	0
160040916	0	0	0	0	0	0	2	0	0	0	3	5	1	0	4	0	0	0	0	3	0	2	0	0
160091981	0	4	3	0	0	4	0	0	0	0	0	0	0	0	5	0	0	0	1	0	2	0	0	0
162476207	0	1	0	4	3	0	0	0	0	0	2	3	5	5	0	5	0	5	0	0	3	0	0	5
164295475	0	0	2	0	0	0	5	0	0	0	0	4	4	0	0	5	0	0	0	4	5	5	3	4
164456478	0	0	0	3	5	0	5	0	0	0	0	0	0	0	2	0	0	0	0	0	5	0	0	3

Item
Embedding
Matrix

1	1	0
0	0	1

User Preference
Matrix

1	0
1	0
0	1

Superman Batman Frozen

User 1

1	1	0
1	1	0
0	0	1

User 2

User 3

<https://towardsdatascience.com/collaborative-filtering-simplified-the-basic-science-behind-recommendation-systems-1d7e7c58cd8>

MODELO

ClientID	BF1772	BF2152	BF1681	BF2370	BF9289	BF8238	BF0060	BF9878	BF0809	BF7814	BF8063	BF6885	BF5238	BF2887	BF4831	BF9724	BF6693	BF1414	BF0103	BF6167	BF6975	BF6604	BF3942	BF9708
100071193	1.59661	0.07264	0.29492	4.49799	-0.4702	-0.323	-0.8223	3.17246	-0.0438	3.30722	0.51161	4.27721	0.21359	1.31145	-0.1897	2.16639	3.28696	3.72769	4.2211	2.31031	1.00984	1.40428	0.69484	1.77191
102228703	1.39438	0.57387	1.23734	0.5726	1.52652	0.32973	0.46592	2.84331	0.08075	-0.3733	0.47847	2.88836	0.99152	0.29877	-0.2833	1.42985	1.94042	2.41497	1.64339	0.71778	0.96204	0.23524	0.50739	2.12395
102320254	-0.7327	3.80374	1.96117	1.14809	0.79909	0.99934	0.16214	4.97584	0.75317	-1.6356	0.99853	1.99671	2.5886	0.89198	1.26424	-0.4801	2.00919	4.40749	1.93034	-0.6051	2.07413	2.00014	2.46816	0.22141
103076257	1.67698	3.03996	0.64106	1.94308	3.99568	-1.4057	0.57372	1.29756	1.62947	2.83771	2.89556	0.88456	2.35658	0.10321	2.08049	3.14299	1.04633	4.06292	-0.9964	-0.3916	2.71972	-0.3299	1.55188	1.64412
103692255	0.61554	-0.8223	2.81731	-0.5896	0.82275	1.73879	4.41694	-0.1417	1.15479	0.52412	0.44361	0.43022	0.54122	3.08819	1.44689	0.97612	2.58553	1.02082	0.14882	0.12022	-0.188	-0.866	2.87545	-1.1597
105120468	-0.3511	0.6204	0.10408	0.71015	-0.0353	0.49624	0.14351	-0.0775	-0.1924	2.32963	-0.1365	0.56185	0.68343	0.40507	0.23103	0.56628	-0.1265	-0.4309	-0.556	0.77118	2.06144	-0.1734	-0.3159	-0.2888
109928560	1.74828	2.65246	3.41265	2.28417	2.73138	1.71939	-0.7727	1.28839	-0.8579	-1.2539	0.48762	3.39006	0.06086	-0.7021	1.55405	0.1241	1.05729	3.85269	2.44719	0.11545	0.64431	2.82112	2.68636	-0.4913
110171128	-0.1457	3.68608	1.4992	3.50228	1.40564	0.04443	0.98907	1.02359	0.77285	2.04213	3.71526	-0.0525	2.63501	2.71422	2.80887	-1.4491	1.63552	0.19056	2.48264	-0.1849	2.98952	3.11671	1.1743	0.40427
110552209	2.10759	0.86074	-0.6006	-0.5592	-0.4662	-0.144	-1.2602	4.92905	-0.3943	4.5319	2.50546	3.78114	3.06827	1.59876	1.16341	1.15775	1.06824	-0.7179	0.56393	2.19621	1.53292	0.10532	0.41102	4.56079
111080176	1.24587	2.0731	-0.3221	0.44526	-0.6042	2.23523	2.63635	3.09103	0.6135	-0.0958	0.51088	0.95272	1.46812	2.13368	0.07062	-1.0227	0.07044	1.07358	0.35614	0.11511	0.62128	2.33183	0.92159	0.0785
111370443	1.06046	0.70283	1.33858	4.80822	2.67435	0.56501	1.74219	-0.8053	-0.9952	3.56095	0.5746	-0.2023	2.28862	0.2498	3.20423	1.35221	-0.4256	0.03143	4.36546	2.9566	4.10265	3.64453	-0.9442	0.14992
111532088	1.28458	-0.1901	0.3241	0.26139	0.10796	0.50964	4.82866	-0.0307	1.82475	-0.029	-0.2256	0.47132	0.17703	0.21626	-0.0899	2.77123	2.15825	1.41973	1.24666	1.69534	1.72385	-0.3582	0.29809	-0.3953
112087671	0.29396	0.14301	3.75369	0.28028	0.4047	4.39668	3.44467	-0.3553	5.3093	2.26522	2.70637	0.71004	0.9437	5.96729	1.38277	1.61685	4.43084	-1.6877	0.9645	2.58227	0.5405	0.17069	3.064	2.67168
112137859	0.2233	-0.0194	-0.0218	3.68508	0.55027	1.34534	-0.3863	0.27735	0.70502	2.635	-1.2369	0.49591	-0.2374	1.11826	-0.0323	1.61801	-0.1313	3.13699	0.68694	1.02928	0.37529	1.17994	0.51643	-0.5129
113848143	0.43311	2.91245	2.34379	1.01719	3.00897	0.90695	0.72006	0.20427	2.46062	0.91729	0.171	1.58491	1.20103	-2.1633	1.7185	4.58738	1.34543	4.19572	-0.0923	1.83456	4.02832	-0.3336	1.91362	0.08414

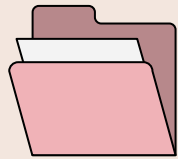


DEMO

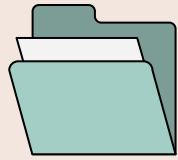
Shop



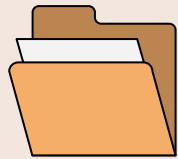
INTERACCIÓN FLUJO WEB



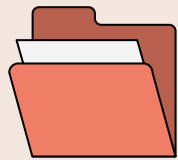
LOGIN



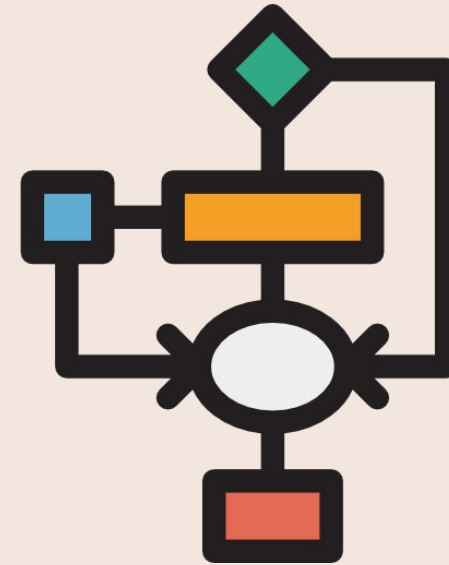
PRODUCTOS



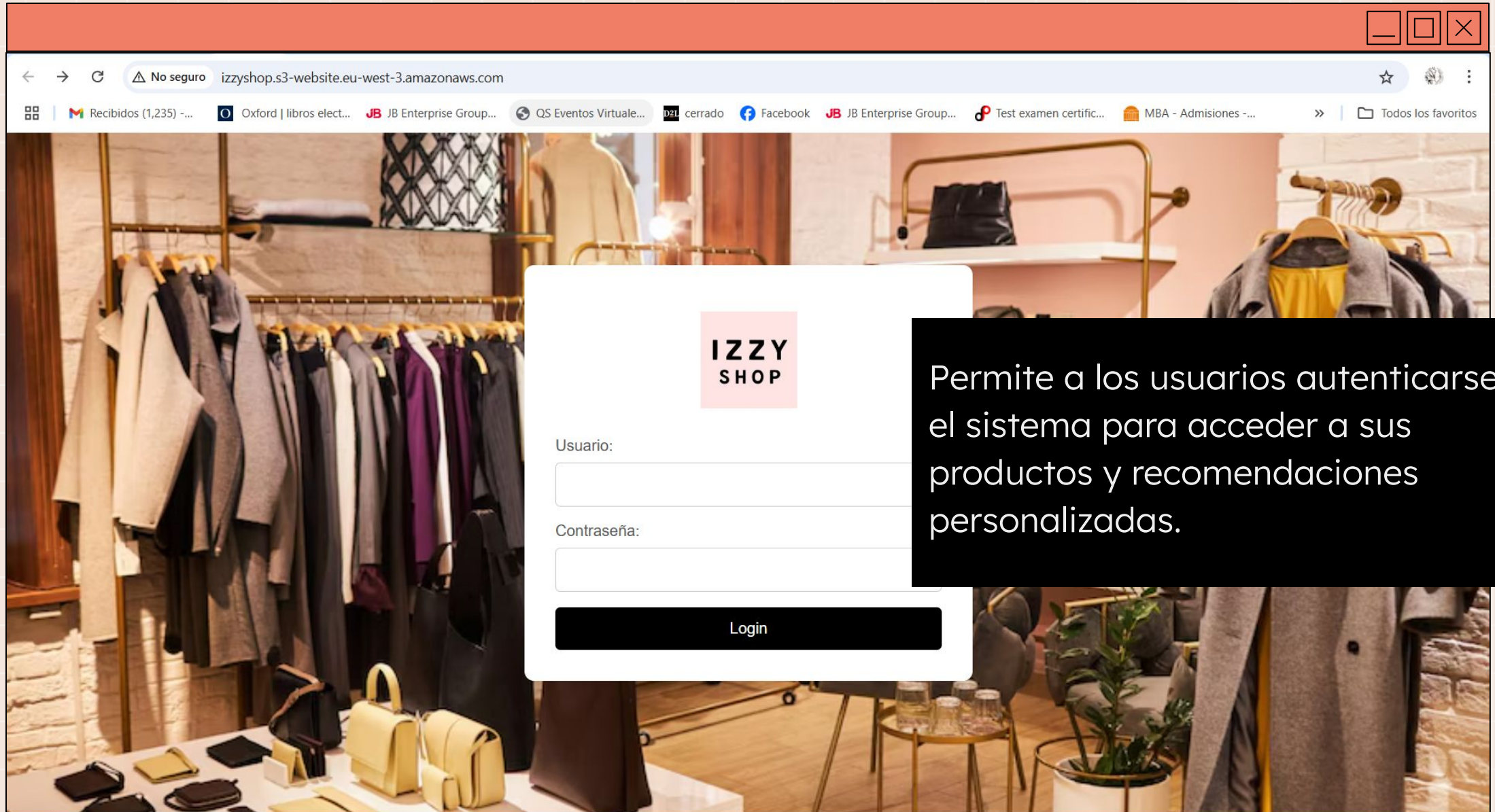
PEDIDOS





RECOMENDACIONES






Pantalla 1: LOGIN



Pantalla 2: PRODUCTOS



Envios a todo España




Tienda de Ropa

Inicio Productos Pedidos Contacto Compras


Hola, Bertha Mosson! Bienvenido a la tienda.

Todos ▾




Bufanda Rayas

La bufanda presenta un elegante diseño suave y ligera lo que la hace cómoda y adecuada para el uso diario. Perfecta para




Bufanda Tejida

Bufanda de tela con un patrón de cuadros en tonos de gris y negro. La bufanda tiene un diseño de tejido entrelazado y bordes



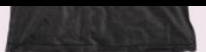
Bufanda Tejida Beige

Bufanda tejida perfecta para días fríos con un diseño muy versátil que permitira su uso diario.




Camisa Estampada

Camiseta en algodón con estampado blanco fresco y combinable para cualquier ocasión causal.



Camisa Estampada B/W

Camiseta en algodón con estampado de color con un contraste que crean una sinergia perfecta.




Camisa Sin Mangas

Camiseta en algodón sin mangas muy apropiada para días mas calurosos que permiten mayor comodidad y frescura.


- Consulta en tiempo real el total de productos consumidos por cliente y su distribución por categorías.
- Usa una API para obtener los datos desde DynamoDB.

Pantalla 3: PEDIDOS

Pedido del Cliente


Bertha Mosson
bmosson@gmail.com

Pedidos del Cliente


Bufanda Rayas

La bufanda presenta un elegante diseño suave y ligera lo que la hace cómoda y adecuada para el uso diario. Perfecta para darle un toque de estilo a cualquier atuendo.


Precio: \$28.7

Nivel de satisfacción (1-5)


Enviar Satisfacción

- Muestra el último pedido del cliente, permitiendo calificar los productos adquiridos.
- Los ratings ingresados se almacenan en DynamoDB mediante una API.

Pantalla 4: RECOMENDACIONES



Perfil del Cliente


 Cerrar Sesión

Datos del Cliente

Nombre: Bertha Mosson


Email: bmosson@gmail.com

Recomendaciones Basadas en Compras Anteriores




Zapatos Beige

Agregar al carrito



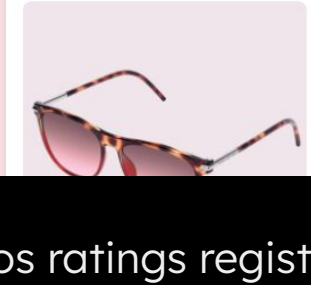

Vestido Rojo Estampado

Agregar al carrito



Zapatos Altos Blancos

Agregar al carrito



Basado en los ratings registrados, un modelo en Python analiza preferencias y sugiere productos similares dentro de la misma categoría.

RECOMENDACION SEGUN RATING

```
response = ordenes.scan(  
    ProjectionExpression = 'ClientID, ProductID, Rating'  
)  
  
orders = response['Items']  
last = response['LastEvaluatedKey']  
  
for order in orders:  
    product = order['ProductID']  
    Client = order['ClientID']  
    rating = float(order['Rating'])  
    R.loc[len(R)] = {'ClientID': Client, 'product': rating}  
  
paginating = True  
while paginating == True:  
    response = ordenes.scan(  
        ExclusiveStartKey = last,  
        ProjectionExpression = 'ClientID, ProductID, Rating'  
    )  
  
    orders = response['Items']  
  
    for order in orders:  
        product = order['ProductID']  
        Client = order['ClientID']  
        rating = float(order['Rating'])  
        R.loc[len(R)] = {'ClientID': Client, 'product': rating}  
  
    try:  
        last = response['LastEvaluatedKey']  
    except:  
        paginating = False
```

IZZY
SHOP

Perfil del Cliente



Cerrar Sesión

Datos del Cliente

Nombre: Bertha Mosson

Email: bmosson@gmail.com

Recomendaciones Basadas en Compras Anteriores



Zapatos Beige



Vestido Rojo Estampado

Análisis Predictivo y Beneficios del Sistema

Mejora la experiencia del usuario al sugerir productos de su interés basados en comportamientos previos.

BENEFICIOS

- Automatización del proceso ETL, reduciendo carga manual.
- Escalabilidad y almacenamiento eficiente con AWS DynamoDB y S3.
- API Gateway y Lambda para integración rápida y eficiente.
- Análisis predictivo que ayuda a mejorar la toma de decisiones comerciales.

CONCLUSIONES

- La implementación de un sistema ETL en AWS permite gestionar datos de clientes y productos de manera eficiente.
- El modelo de recomendación mejora la personalización de ofertas en e-commerce.
- Se logró estructurar un flujo de datos que optimiza la toma de decisiones basada en análisis predictivo.

FUTURO DEL PROYECTO

- Mejorar los algoritmos de recomendación con técnicas avanzadas de Machine Learning.
- Integración con plataformas de Business Intelligence para visualización de datos en dashboards.
- Ampliar la cantidad de datos analizados para mejorar la precisión de las recomendaciones.

REFERENCIAS

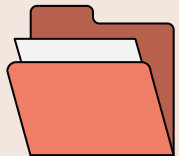
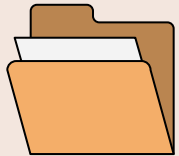
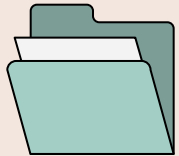
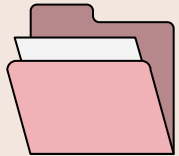
Datos: Adarsh Anil Kumar via Kaggle; Mockaro

Documentación: DynamoDB; boto3; numpy

Google. “Collaborative Filtering”

Template Presentación: SlidesMania

Fonts : Lexend and **Belanosima**.





GRACIAS

This is where you section ends. Duplicate this set of slides as many times you need to go over all your sections.