



Reseñas y Recomendaciones

Proyecto



Sprint 2



Índice

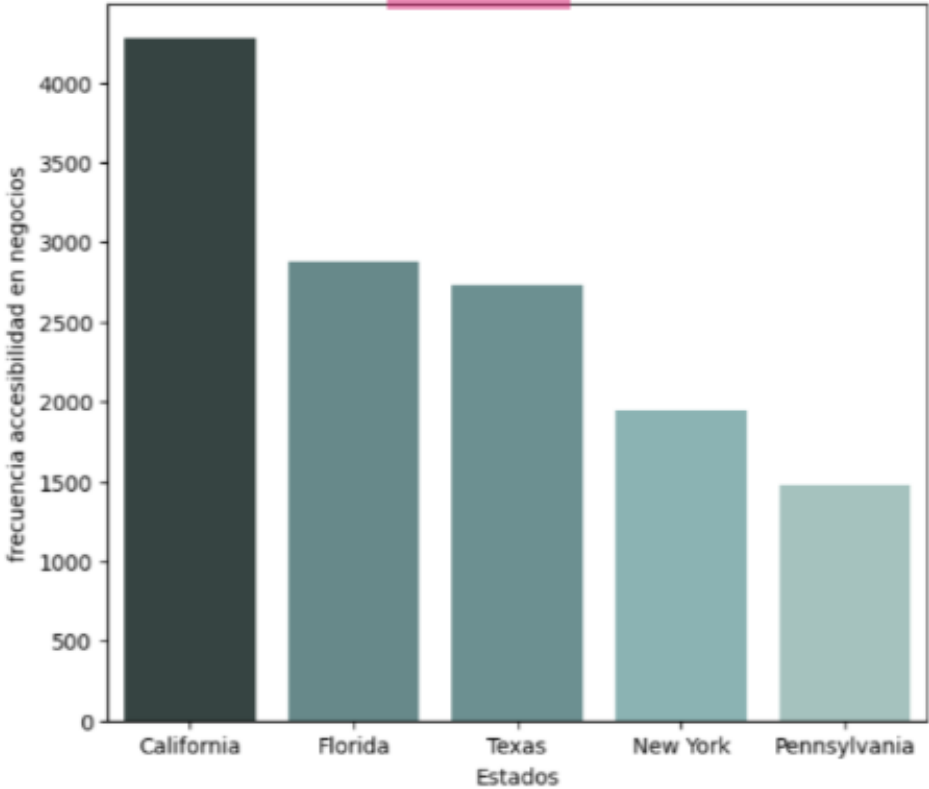


- Portada
- Índice
- EDA
- Pipeline ETL Automatizado - Workflow
- API (video)
- MVP Dashboard
- MVP Model ML (6 páginas)
- Modelo E/R
- Documentación

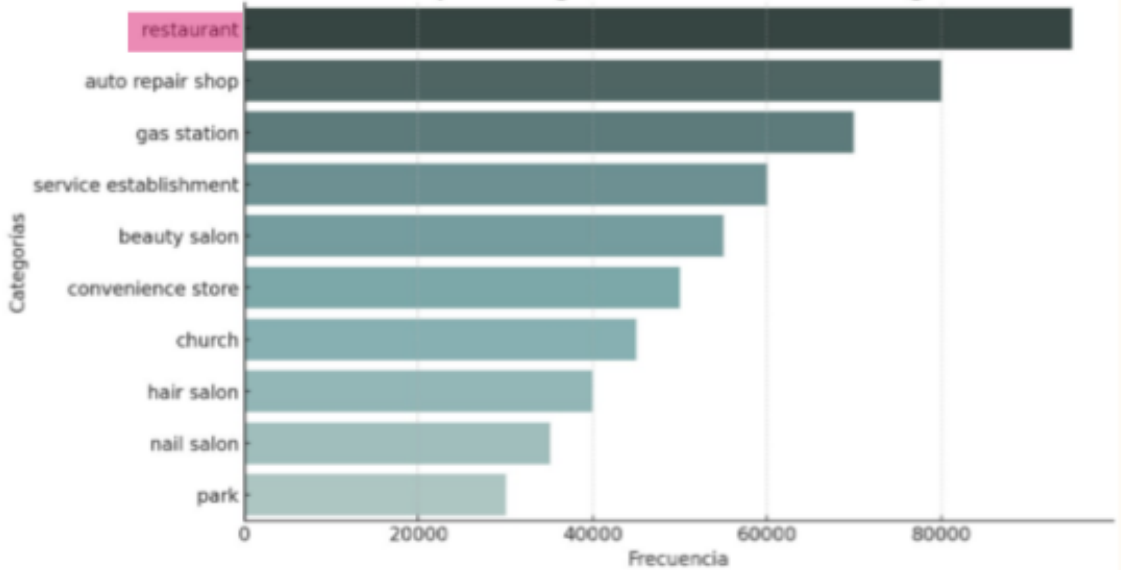
Nuestro KPI ...



Frecuencia de solo **accesibilidad** en negocios por estados.



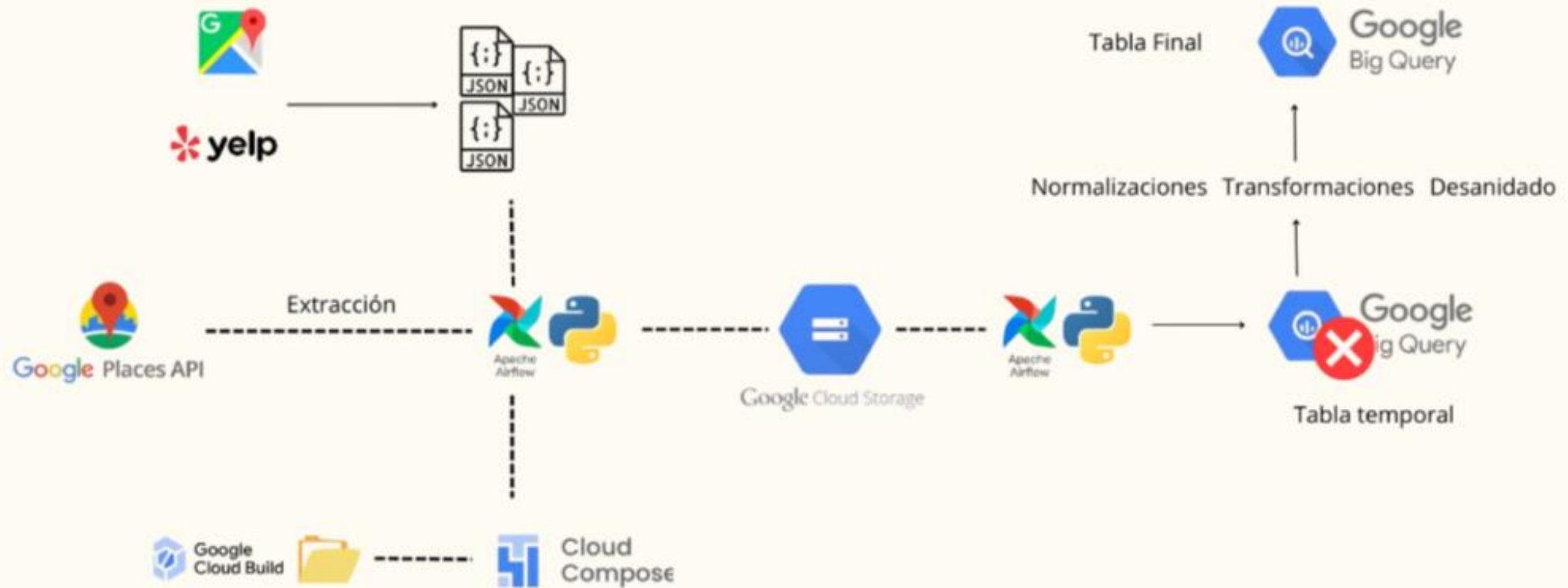
Top 10 categorías más comunes en Google



Nuestro alcance ...



3. Pipeline ETL automatizado - Workflow Tecnologías





4. Pipeline para alimentar DW (API)

The screenshot displays the Apache Airflow web interface. The top navigation bar includes links for Airflow, DAGs, Cluster Activity, Datasets, Browse, Admin, Docs, and Composer. The main header shows the DAG name 'Procesamiento ETL Google' and the execution time '2024-10-27, 00:00:00 UTC'. Below the header, there are tabs for Details, Graph, Gantt, Code, and Audit Log. The left sidebar lists the DAG's tasks, each with a status icon (green for success, red for failure). The central area shows the DAG's graph, with tasks represented as boxes connected by arrows. The tasks include: 'obtener_datos', 'transformar_datos', 'cargar_datos', 'validar_datos', 'generar_reporte', 'enviar_reporte', 'limpiar_datos', 'actualizar_datos', 'generar_reporte', 'enviar_reporte', 'limpiar_datos', 'actualizar_datos', 'generar_reporte', 'enviar_reporte', 'limpiar_datos', 'actualizar_datos'. The bottom left corner features the 'genially' logo.

PIPELINE ETL GOOGLE

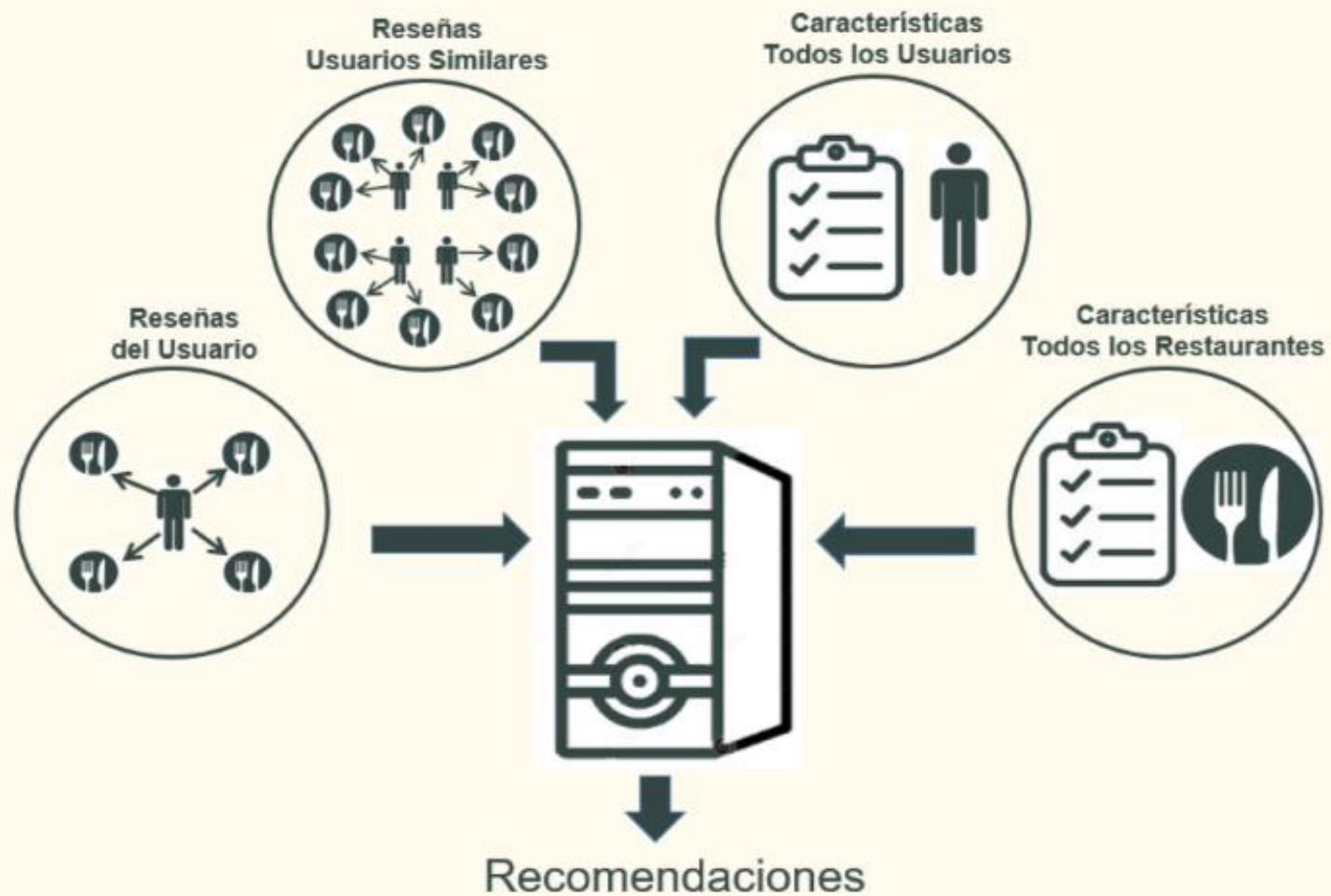


5. MVP Dashboard





Modelo de Recomendación Híbrido





Algoritmo LightFm

```
model = LightFM(loss='warp', item_alpha=1e-10, user_alpha=1e-10)  
model.fit(interactions, item_features=item_features, user_features=user_features)
```

```
scores = model.predict(user_id, business_ids, item_features, user_features)  
top_items = np.argsort(-scores)[:top_n]
```

DONDE:

user_id: **int**

business_ids: **list[int]**

item_features: **scipy.sparse**

user_features: **scipy.sparse**

interactions: **scipy.sparse**

scores: **numpy.ndarray**



Entrenamiento del Modelo

- . 6000 **Locales**
- . 36000 **Usuarios**
- . 57000 **Reseñas**

Prueba 1:

- . 3 Peticiones Variando sólo el Usuario

Prueba 2:

- . 1 Petición mal hecha (Probando la Api)



Probando el Modelo con la Api

Petición 1

```
url = "http://127.0.0.1:8000/get_recomendations/"
payload = {"state": "PA",
           "user_id_str": "A-urIi0zR2sAcnN3VccI50",
           "category": "Normal",
           "top_n": 2}

response = requests.get(url=url, json=payload)

print(f'Status Code --> {response.status_code}')
print('*'*30)
if response.status_code == 200:
    for i in range(len(response.json()['recommendations'])):
        response.json()
```

[34] ✓ 0.0s

```
... Status Code --> 200
*****
Gia Pronto | Red Hook Coffee & Tea

... {'recommendations': [{'business_id': '1bsQMS1SeLxcqXrJQVvfGQ',
    'name': 'Gia Pronto',
    'city': 'Philadelphia',
    'address': '3400 Civic Center Blvd.',
    'latitude': 39.9476279,
    'longitude': -75.192598},
    {'business_id': 'bTve2mwLk5Zc01vRKqc2KQ',
    'name': 'Red Hook Coffee & Tea',
    'city': 'Philadelphia',
    'address': '765 S 4th St',
    'latitude': 39.9385518,
    'longitude': -75.1496356}]}
```

Petición 2

```
url = "http://127.0.0.1:8000/get_recomendations/"
payload = {"state": "PA",
           "user_id_str": "96cTtpuU3rJ9jXgJ70oR8A",
           "category": "Normal",
           "top_n": 2}

response = requests.get(url=url, json=payload)
print(f'Status Code --> {response.status_code}')
print('*'*30)
if response.status_code == 200:
    for i in range(len(response.json()['recommendations'])):
        response.json()
```

[15] ✓ 0.0s

```
... Status Code --> 200
*****
Saxbys | Wawa

... {'recommendations': [{'business_id': 'PCmoZ6i5fXLkti6SmykDNg',
    'name': 'Saxbys',
    'city': 'Philadelphia',
    'address': '65 N 34th St',
    'latitude': 39.9574168,
    'longitude': -75.1907202},
    {'business_id': 'kka9G6czM7e7QdXFQbRo9A',
    'name': 'Wawa',
    'city': 'Woodlyn',
    'address': '1920 West Macdade Blvd',
    'latitude': 39.87065,
    'longitude': -75.353596}]}
```



Petición 3

```
url = "http://127.0.0.1:8000/get_recomendations/"
payload = {"state": "PA",
           "user_id_str": "26BRT0-E8XX9-Hher7Evzg",
           "category": "Normal",
           "top_n": 2}
response = requests.get(url=url, json=payload)
print(f'Status Code --> {response.status_code}')
print('*'*30)
if response.status_code == 200:
    for i in range(len(response.json()['recomendations'])):
        response.json()
```

[16] ✓ 0.0s

... Status Code --> 200

McDonald's | Paris Baguette

... {'recomendations': [{'business_id': 'KN0IJPxZ6bwVYmqLcAfrdg',
'name': 'McDonald's',
'city': 'North Wales',
'address': '1200 Welsh Rd',
'latitude': 40.2273628769,
'longitude': -75.2591022849},
{'business_id': 'CqmPurpyPzwGs9chhruC3Q',
'name': 'Paris Baguette',
'city': 'Philadelphia',
'address': '600 Washington Ave',
'latitude': 39.9344835,
'longitude': -75.1549941}]}

Petición 4

```
url = "http://127.0.0.1:8000/get_recomendations/"
payload = {"state": "XX",
           "user_id_str": "26BRT0-E8XX9-Hher7Evzg",
           "category": "Normal",
           "top_n": 2}
response = requests.get(url=url, json=payload)
print(f'Status Code --> {response.status_code}')
print('*'*30)
if response.status_code == 200:
    for i in range(len(response.json()['recomendations'])):
        response.json()
```

[17] ✓ 0.0s

... Status Code --> 400

... {'detail': 'BAD REQUEST'}



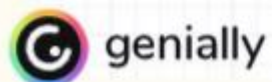
Utilización de FastApi para desplegar Modelo

Petición 1: user_id = "A-uri0zR2sAcnN3Vccl5Q"
Petición 2: user_id = "96cTtpuU3rJ9jXgJ70oR8A"
Petición 3: user_id = "26BRT0-E8XX9-Hher7Evzg"
Petición 4: Se introdujo MAL el Estado

Recomendación: Gia Pronto | Red Hook Coffee & Tea
Recomendación: Saxbys | Wawa
Recomendación: McDonald's | Paris Baguette
Resultado: **BAD REQUEST**

```
aliskair@LaptopAliskair: ~/pro × + v
aliskair@LaptopAliskair:~$ cd proyecto_grupal/
aliskair@LaptopAliskair:~/proyecto_grupal$ source venv/bin/activate
(venv) aliskair@LaptopAliskair:~/proyecto_grupal$ uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['/home/aliskair/proyecto_grupal']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [106453] using WatchFiles
INFO: Started server process [106455]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:50898 - "GET /get_recomendations/ HTTP/1.1" 200 OK
INFO: 127.0.0.1:42284 - "GET /get_recomendations/ HTTP/1.1" 200 OK
INFO: 127.0.0.1:33162 - "GET /get_recomendations/ HTTP/1.1" 200 OK
INFO: 127.0.0.1:44780 - "GET /get_recomendations/ HTTP/1.1" 400 Bad Request
```

} Logs
Petición





'¿Dónde encontrar más información?'

13. Documentación

En el **repositorio de GitHub** se explica todo el proceso, las tecnologías escogidas, y más detalles de nuestro proyecto, tanto en el **README**, como en cada una de las carpetas . Allí podrás ver:

- Diagrama de Gantt con la distribución de tareas.
- Diccionario de datos detallado.
- Pipelines y diagramas.
- EDA completo.
- Análisis detallado con insights obtenidos.
- Enlaces a información estratégica.