
Objetivo general:

Evaluar la capacidad del postulante para construir pipelines de replicación, scraping y actualización de datos, desde bases de datos, APIs y sitios web hacia una base destino en la nube, incluyendo automatización de procesos y modelado básico de datos orientado a BI.

Ejercicio 1 - Replicación de base de datos**Descripción general**

La empresa cuenta actualmente con un sistema transaccional que almacena las ventas y los productos en una base de datos PostgreSQL.

Se requiere replicar estos datos diariamente en una base espejo en la nube, la cual servirá como base para procesos posteriores de Business Intelligence.

El postulante deberá:

1. Crear un pipeline que lea los datos desde la base de datos origen.
2. Cargar los datos en una base destino en la nube.
3. Automatizar el pipeline para que corra una vez al día.

Datos disponibles

Te proveemos las siguientes tablas de la base de datos de origen postgres:

Encontrá acá las siguientes tablas en csv:

- Fechas
- Customer Segment
- Productos
- Ventas

Entregables requeridos

1. Crear la base de datos de origen en Postgres y poblarla con los datos provistos.
2. Crear una base de datos espejo en un servicio en la nube gratuito.
3. Desarrollar un script en Python que:
 - o Se conecte a la base origen (accesos provistos en la entrevista).
 - o Extraiga los datos de ambas tablas.
 - o Los cargue en la base destino, recreando las tablas y relaciones.
4. Automatizar la ejecución diaria del script.
5. Documentar:
 - o El código desarrollado.
 - o El esquema creado en la base destino.
 - o Los pasos de la automatización.
 - o Cómo acceder a la base espejo creada.

Ejercicio 2 - Extracción incremental desde API externa (BCRA)

Descripción general

La empresa requiere incorporar información externa de referencia para enriquecer sus reportes financieros. Específicamente, desea registrar todas las cotizaciones oficiales del tipo de cambio del Banco Central de la República Argentina (BCRA), disponibles en su API pública.

El objetivo de este ejercicio es:

1. Extraer **todas las cotizaciones históricas** desde la API del BCRA.
2. Almacenar los datos en una tabla cotizaciones dentro de una base PostgreSQL en la nube.
3. Diseñar un mecanismo de ingesta **incremental semanal**, de modo que se consulten solo las cotizaciones **posteriores a la última fecha registrada**.
4. Automatizar el script para que se ejecute una vez por semana.

Fuente de datos

API oficial de estadísticas cambiarias del BCRA:

<https://www.bcra.gob.ar/BCRAyVos/catalogo-de-APIs-banco-central.asp>

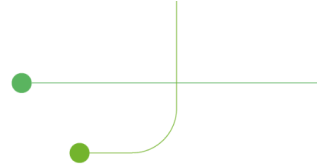
Dataset: Cotización del dólar tipo vendedor (estadística diaria oficial)

Tabla a crear: cotizaciones

Campo	Tipo de dato	Descripción
fecha	DATE	Fecha de la cotización
moneda	TEXT	Nombre de la moneda (ej: Dólar)
tipo_cambio	NUMERIC(10,4)	Valor de la cotización
fuentes	TEXT	Fuente de los datos ("BCRA")

Entregables requeridos

1. Script Python que:
 - o Consuma la API del BCRA.
 - o Cree la tabla cotizaciones si no existe en la BD Postgres del Ejercicio 1 (deseable) o en un csv en la nube.
 - o Inserte todas las cotizaciones históricas inicialmente.
 - o Luego, ejecute modo incremental: sólo cotizaciones posteriores a la última fecha registrada.
2. Automatización del script para que corra una vez por semana.
3. Documentación que incluya:
 - o Cómo fue construida la ingesta incremental.
 - o Ejemplo de ejecución inicial y una incremental.
 - o Acceso a la base PostgreSQL en la nube.



Ejercicio 3 - Scraping de propiedades en venta

Descripción general

La empresa está interesada en recopilar información pública sobre precios del mercado inmobiliario en la ciudad de Posadas, Misiones.

El objetivo de este ejercicio es construir un pipeline que:

1. Realice scraping de anuncios de **venta de terrenos** en la ciudad de Posadas, Misiones, desde algún marketplace público, que permita el scraping entre sus términos y condiciones.
2. Extraiga la mayor cantidad de campos disponibles, incluyendo preferentemente:
 - o Nombre o título del anuncio
 - o Dirección o ubicación aproximada
 - o Precio
 - o Coordenadas geográficas (si están disponibles en el sitio)
3. Almacene la información en una base de datos PostgreSQL en la nube (preferentemente en la misma que el ejercicio 1), o alternatively, en un archivo .csv almacenado en un bucket o carpeta en la nube (Google Drive, Dropbox, etc.).

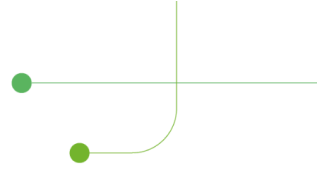
Consideraciones técnicas

- El scraping debe respetar los límites del sitio web (usar headers adecuados, evitar bloqueos, etc.)
- Puede utilizarse requests, BeautifulSoup, Selenium, o cualquier otra herramienta Python.
- No es necesario recorrer todas las páginas: basta con mostrar cómo sería el proceso y scrapeo de al menos 20 resultados.

Entregables requeridos

1. Script Python funcional que realice el scraping solicitado.
2. Código que cargue los datos en la base o guarde el CSV.
3. Documentación que indique:
 - o Sitio utilizado y estructura de scraping.
 - o Campos extraídos y justificación.
 - o Ubicación de los datos (DB o CSV).
 - o Limitaciones conocidas o bloqueos.





Servicios sugeridos para todos los ejercicios

Bases PostgreSQL gratuitas en la nube:

- **Supabase** (<https://supabase.com/>)
- **Render** (<https://render.com/>)
- **ElephantSQL** (<https://www.elephantsql.com/>)
- **Neon** (<https://neon.tech/>)

Automatización de scripts en la nube:

- **GitHub Actions**
- **Render cron jobs**
- **PythonAnywhere (plan gratuito)**
- **Google Cloud Functions (free tier)**
- **HuggingFace Spaces (sólo para ejecución bajo demanda)**

Tecnologías permitidas

- Lenguaje: Python
- Librerías sugeridas: requests, pandas, sqlalchemy, psycopg2, datetime, dotenv, BeautifulSoup, Selenium

Criterios de evaluación

- Correcta extracción y carga de datos.
- Modelado apropiado en base destino.
- Automatización funcional y reproducible.
- Código modular, limpio y documentado.
- Uso apropiado de servicios gratuitos.
- Documentación clara para revisores.

Duración sugerida de la prueba

- Tiempo estimado: 1 semana (viernes 01/08/2025 a 8/08/2025)
- Entrega vía GitHub o Google Drive, incluyendo:
 - o Código fuente.
 - o Instrucciones paso a paso.
 - o Acceso a bases de datos