



Prueba Técnica

Integrations Engineer

Capital Humano, 2025

¡Muchas Felicidades!

Bienvenido a la siguiente fase del proceso para la posición: **Integrations Engineer.**

Tendremos varios casos en donde se le pedirá diseñar y construir un flujo de integración de datos que recolecte información de múltiples fuentes, la transforme y la consolide en un único destino. Esto nos permitirá evaluar tus habilidades técnicas en el manejo de diferentes formatos de datos, la interacción con APIs y tu capacidad para estructurar una solución de integración coherente y funcional.

Para empezar, nos gustaría que tenga en cuenta los siguientes criterios:

- Fecha de entrega de la prueba
- Ejecución y resolución
- Presentación y sustentación



Casos a resolver....



Prueba Técnica para Ingeniero de Integración



El objetivo de esta prueba es diseñar y construir un flujo de integración de datos que recolecte información de múltiples fuentes, la transforme y la consolide en un único destino. Esto nos permitirá evaluar tus habilidades técnicas en el manejo de diferentes formatos de datos, la interacción con APIs y tu capacidad para estructurar una solución de integración coherente y funcional.

Nota:

En caso que no resuelvas todos los puntos, no te preocupes, la idea es que puedas sustentar los que logres resolver.



Fuentes de Datos



Para esta prueba, utilizarás una combinación de APIs públicas y un archivo de datos estático. Esto simulará un escenario real donde un ingeniero de integración necesita combinar información de diferentes sistemas.

- **API de Usuarios (JSON):** JSONPlaceholder (<https://jsonplaceholder.typicode.com/>)
 - Endpoint a utilizar: /users para obtener una lista de usuarios de prueba.
 - Formato: REST/JSON.
- **API del Clima (JSON):** OpenWeatherMap (<https://openweathermap.org/api>)
 - Deberás registrarte para obtener una API Key gratuita.
 - Endpoint a utilizar: Current weather data.
 - Formato: REST/JSON.
 - **Nota:** Puedes usar las coordenadas geográficas (lat y lon) de los usuarios obtenidos de JSONPlaceholder para consultar el clima de su ubicación.
- **Datos de Pedidos (CSV):**
 - Puedes descargar un archivo CSV de ejemplo desde este enlace: Sample CSV Orders (<https://www.google.com/search?q=https://www.contextures.com/SampleData.zip>). Dentro del archivo ZIP, utiliza SampleData.xlsx y guárdalo como orders.csv.
 - Formato: CSV.



Herramientas Recomendadas



Tienes la libertad de elegir las herramientas con las que se sienta más cómodo. Sin embargo, aquí hay algunas sugerencias de herramientas de código abierto o con capas gratuitas que son relevantes para este tipo de tareas:

- **Lenguaje de Programación:** Con el que te sientas más cómodo, sin embargo, Python o Node.js son excelentes opciones por sus librerías para el manejo de datos y peticiones HTTP.
- **Orquestación de Flujos (Opcional, pero un plus):**
 - Apache Airflow: Si el candidato tiene experiencia, puede definir el flujo como un DAG.
 - N8n u Pipedream: Alternativas más sencillas y visuales para la orquestación de flujos de trabajo.
- **Contenerización (Opcional, pero un plus):**
 - Docker: Entregar la solución en un Dockerfile que facilite la ejecución del proceso es una excelente práctica.
- **Control de Versiones:**
 - Git: El proyecto final debe ser entregado en un repositorio de Git (puede ser en GitHub, GitLab, etc.).



Puntos a Resolver



Deberás construir un script o un flujo de trabajo que realice los siguientes pasos:

- Extracción de Datos:
 - Conectarse a la API de JSONPlaceholder y obtener la lista completa de usuarios.
 - Para cada usuario obtenido, extraer su id, name, username, email, y los datos de geolocalización (lat y lon) desde el objeto address.
 - Leer los datos del archivo orders.csv.
- Enriquecimiento de Datos:
 - Utilizando los datos de geolocalización de cada usuario, realizar una llamada a la API de OpenWeatherMap para obtener el clima actual de su ubicación.
 - Del resultado del clima, te interesa obtener la descripción principal (weather[0].main) y la temperatura (main.temp). La temperatura deberá ser convertida de Kelvin a Celsius. La fórmula es: $Celsius = Kelvin - 273.15$.
- Transformación y Agregación de Datos:
 - Debes encontrar una manera de asociar los pedidos del archivo CSV con los usuarios. Puedes asumir que la columna Rep (Representante) en el archivo CSV corresponde al username de los usuarios de la API.
 - Para cada usuario, calcula el total de unidades vendidas (Units) y el ingreso total (Total) sumando los pedidos que le corresponden.



Puntos a Resolver



- Carga de Datos:
 - El resultado final debe ser un único archivo JSON llamado `integration_output.json`.
 - Este archivo debe contener un array de objetos, donde cada objeto represente a un usuario y tenga la siguiente estructura:

```
[
  {
    "userId": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "weather": {
      "main": "Clear",
      "temperature_celsius": 25.5
    },
    "sales_summary": {
      "total_units_sold": 500,
      "total_revenue": 7500.00
    }
  },
  ...
]
```



Entregables



Para considerar la prueba completa, el candidato debe proporcionar:

- Un repositorio de Git con todo el código fuente de la solución.
- Un archivo README.md claro y conciso que explique:
 - Cómo configurar el entorno (instalación de dependencias, variables de entorno como la API Key de OpenWeatherMap, etc.).
 - Cómo ejecutar la solución.
 - Cualquier decisión de diseño importante que haya tomado.
- El archivo de salida integration_output.json generado por su proceso.
- (Opcional) Un Dockerfile para ejecutar la solución en un contenedor.





¡Gracias!



Fecha y hora máxima de envío:

Lunes, 20 de octubre del 2025, 8:00 am.

Correo Electrónico:

vcastaneda@banesco.com