Infraestructura y Arquitectura Para Big Data

EA4. Proyecto Integrador
Documentación de la Arquitectura y Modelo de Datos

Elizabeth Alzate
PREICA2501B010108

Jimy Mora Russy
PREICA2501B010109

Andrés Felipe Callejas

Docente

INGENIERIA DE SOFTWARE

IUDigital de Antioquia

Medellín, abril de 2025

Descripción General De La Arquitectura

- **Visión Global:**

En el contexto de nuestro proyecto integrador final de la materia arquitectura y estructura para Big Data hacemos un análisis de las diferentes etapas que se incluyeron en el desarrollo de las diferentes actividades del proyecto asignado durante el ciclo académico.

Se siguieron 3 fases claves para el desarrollo de la actividad como fueron la fase de ingesta, preprocesamiento de datos y enriquecimiento.

- Fase de ingesta: Se extrajeron los datos desde un API en nuestro caso el api de "BreweryDB", los cuales se almacenan en una base de datos SQLite local para simular el almacenamiento en la nube. Esta ingesta se realizó utilizando un script de python que emplea bibliotecas como request para extraer los datos y squlite3 para la conexión y almacenamiento en la base de datos.
- Fase de preprocesamiento: En esta fase se realizó la limpieza y transformación de los datos obtenidos del api. Se identificaron y corrigieron errores como valores nulos, valores inconsistentes y se depuraron registros duplicados. Este preprocesamiento simulo un entorno de Big Data en la nube donde los datos son cargados desde la base de datos local para su análisis y depuración.
- Fase de enriquecimiento: En esta fase el conjunto de datos base obtenidos de las actividades anteriores y los cuales fueron preprocesados son enriquecidos con información adicional provenientes de fuentes variadas como archivos XML, JSON, HTML, integrándose nuevas columnas a las ya existentes y manteniendo la estructura para facilitar futuros análisis de los datos de nuestro proyecto.

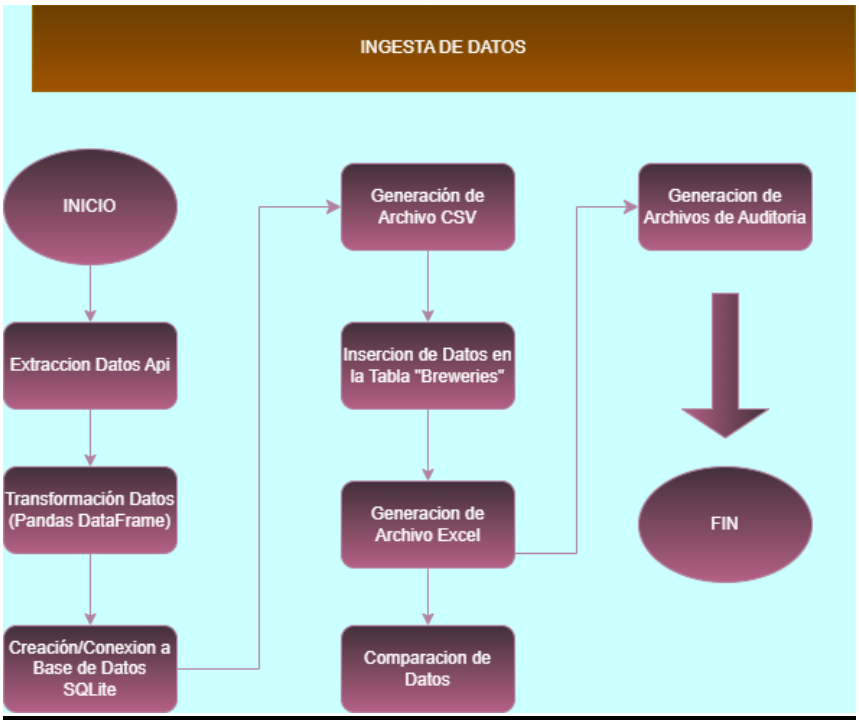
- **Componentes Principales:**

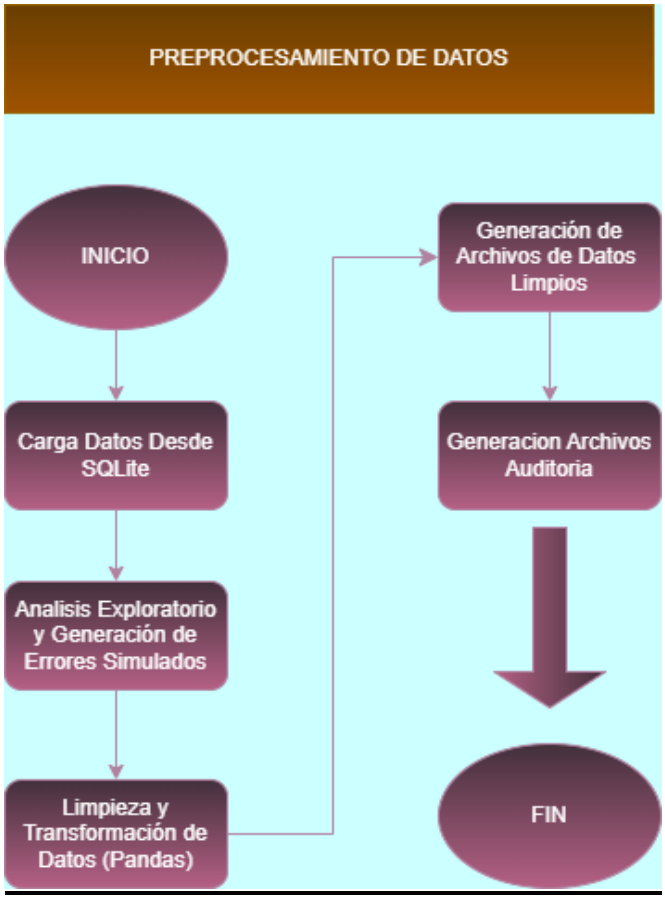
- Base de datos analítica (SQLite): Donde se almacenaron los datos extraídos de la API. SQLite es una base de datos que nos permite simular el almacenamiento de datos en la nube y es utilizada tanto para la ingesta como para el almacenamiento intermedio de datos procesados.

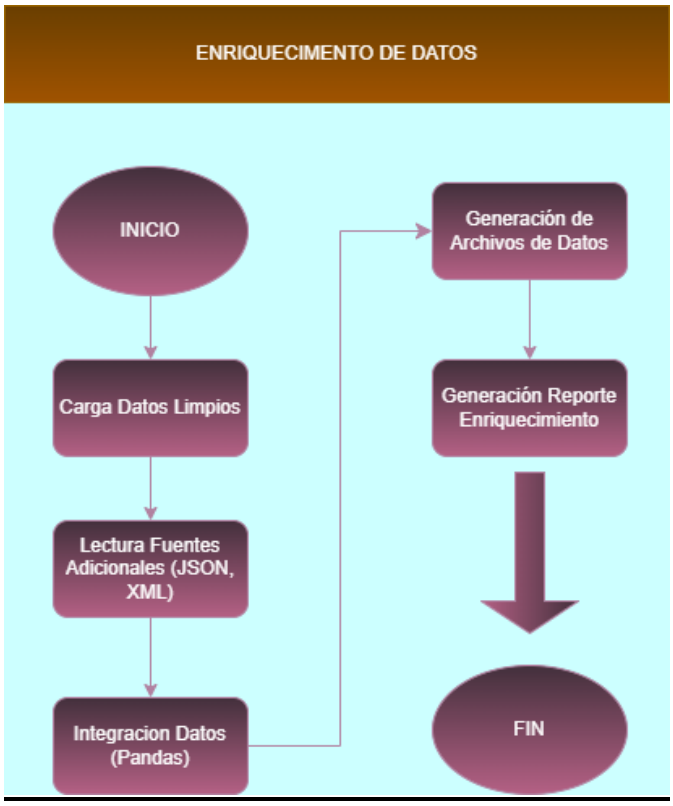
- Scripts de procesamiento (Ingesta, Limpieza, Enriquecimiento): Utilizando diversos scripts de Python ejecutamos las diversas etapas ya descritas del proyecto que permiten automatizar todo el flujo de datos des su extracción, transformación y enriquecimiento.

- Mecanismos de Automatización (GitHub Actions – Workflows): Con GitHub Actions se automatizo todo el flujo de trabajo incluyendo los scripts de ingesta, limpieza, enriquecimiento de los datos, generando archivos de salida y también datos enriquecidos.

Diagramas de Arquitectura




PREPROCESAMIENTO DE DATOS
INICIO
Carga Datos Desde SQLite
Analisis Exploratorio y Generación de Errores Simulados
Limpieza y Transformación de Datos (Pandas)
Generación de Archivos de Datos Limpios
Generacion Archivos Auditoria
FIN


ENRIQUECIMENTO DE DATOS
INICIO
Carga Datos Limpios
Lectura Fuentes Adicionales (JSON, XML)
Integracion Datos (Pandas)
Generación de Archivos de Datos
Generación Reporte Enriquecimiento
FIN

Modelo de Datos

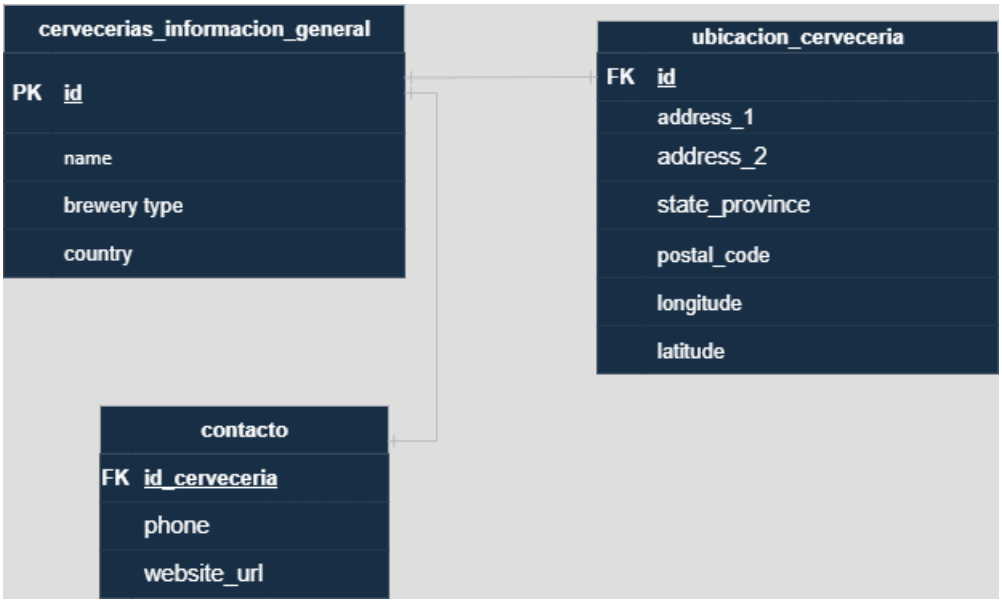
Para poder realizar este punto tuvimos que dividir la única tabla que tiene la API de Breweries en 3 subtablas como se visualiza en la gráfica para poder hacer el punto solicitado y crear el modelo entidad – relación.

Estructura de la tabla Breweries: Principal tabla de la activad a partir de la cual se inicia el proyecto de Big Data.

Nombre del campo	Descripción del campo	Tipo de dato
id	Identificador único de la cervecera	TEXT
name	Nombre de la cervecera	TEXT
brewery_type	Tipo de cervecera (micro, nano, regional, etc.)	TEXT
address_1	Dirección principal de la cervecera	TEXT

Nombre del campo	Descripción del campo	Tipo de dato
address_2	Segunda dirección (opcional)	TEXT
city	Ciudad donde se ubica la cervecera	TEXT
state_province	Estado o provincia	TEXT
postal_code	Código postal	TEXT
country	País de origen	TEXT
longitude	Longitud geográfica (ubicación)	REAL
latitude	Latitud geográfica (ubicación)	REAL
phone	Teléfono de contacto	TEXT
website_url	Sitio web oficial de la cervecera	TEXT
state	Estado o región puede duplicar state_province)	TEXT
street	Calle donde se ubica la cervecera	TEXT

Subtablas: Grafico de las subtablas generadas a partir de la tabla Breweries.



1. **cervecerias_informacion_general**: Contiene la información básica de cada cervecera, como el nombre, tipo y país.
2. **ubicacion_cerveceria**: Contiene la información de ubicación de cada cervecera, como dirección, estado, código postal, longitud y latitud.
3. **contacto**: Contiene la información de contacto de la cervecera, como teléfono y URL del sitio web.

Relaciones:

- La relación entre las tablas se establece a través de la **clave primaria** id de la tabla **cervecerias_informacion_general** y las **claves foráneas** en las tablas **ubicacion_cerveceria** y **contacto**.

Descripción de tablas involucradas, campos y tipos de datos

Tabla cervecerías_informacion_general:

- Esta tabla almacena los **detalles básicos** de la cervecera.

Campo	Descripción	Tipo de Dato
id (PK)	Identificador único de la cervecera (clave primaria)	TEXTO
name	Nombre de la cervecera	TEXTO
brewery_type	Tipo de la cervecera (por ejemplo, micro, nano, etc.)	TEXTO
country	País donde está ubicada la cervecera	TEXTO

Tabla ubicación_geografica:

Esta tabla almacena la **ubicación geográfica** de cada cervecera, incluyendo la dirección y las coordenadas geográficas.

Campo	Descripción	Tipo de Dato
id (PK, FK)	Referencia a id de la tabla **cervecerias_informacion_general**	TEXTO
address_1	Dirección principal de la cervecera	TEXTO
address_2	Dirección secundaria (opcional)	TEXTO
state_province	Estado o provincia donde está la cervecera	TEXTO
postal_code	Código postal de la cervecera	TEXTO
longitude	Longitud geográfica de la ubicación de la cervecera	REAL
latitude	Latitud geográfica de la ubicación de la cervecera	REAL

Tabla contacto:

Esta tabla almacena los datos de **contacto** de la cervecera, como el teléfono y la URL de su sitio web.

Campo	Descripción	Tipo de Dato
id (PK, FK)	Referencia a id de la tabla **cervecerias_informacion_general**	TEXTO
phone	Número de teléfono de la cervecera	TEXTO
website_url	URL del sitio web de la cervecera	TEXTO

Relaciones de las tablas:

En este modelo de datos, las tablas están **relacionadas por la clave primaria** id en la tabla **cervecerias_informacion_general**.

1. **Relación entre cervecerias_informacion_general y ubicacion_cerveceria**:
   - La tabla **ubicacion_cerveceria** tiene una **clave foránea (id)** que hace referencia a la **clave primaria (id)** de la tabla **cervecerias_informacion_general**.
   - Esto establece una **relación uno a uno (1:1)** entre las dos tablas: cada cervecera tiene una única ubicación asociada.

2. **Relación entre cervecerias_informacion_general y contacto**:
   - La tabla **contacto** también tiene una **clave foránea (id)** que hace referencia a la **clave primaria (id)** de la tabla **cervecerias_informacion_general**.
   - Esta es otra relación **uno a uno (1:1)**, lo que significa que cada cervecera tiene un único número de teléfono y una URL asociada.

Justificación: El modelo de datos se diseñó así para almacenar información detallada sobre cervecerías facilitando así el análisis y la integración de datos. La columna id como llave primaria se utiliza para la correcta relación de la información.

Justificación de Herramientas y Tecnologías

SQLite:

- Base de datos ligera para simular un entorno de nube.
- Permite el almacenamiento y consulta de datos.

Pandas:

- Librería para la manipulación y análisis de datos en Python.
- Facilita la limpieza, transformación y exportación de datos.

GitHub Actions:

- Plataforma de CI/CD para la automatización del flujo de trabajo.
- Garantiza la reproducibilidad y trazabilidad.

Simulación del Entorno Cloud:

- SQLite y Pandas simulan el almacenamiento y procesamiento en la nube.

Flujo de Datos y Automatización

- **Explicación del Flujo:**

  El flujo de datos sigue los siguientes procesos:

  - Ingesta: Los datos se extraen de la API y archivos locales y se almacenan en SQLite.
  - Preprocesamiento: Los datos se limpian y transforman con Pandas.
  - Enriquecimiento: Se enriquecen los datos con información de otras fuentes.
  - Generación de resultados: Los datos procesados y enriquecidos se exportan a formatos como Excel, CSV para su análisis posterior.
  - Generación de informes de auditoría.
  - GitHub Actions automatiza todo el proceso.

Conclusiones y Recomendaciones

- **Beneficios:**

  - Automatización del flujo de trabajo con GitHub Actions.
  - Uso de herramientas eficientes como Pandas y SQLite.
  - Generación de informes de auditoría.

- **Limitaciones:**

  - La simulación con SQLite puede no ser escalable para grandes volúmenes de datos.

- **Recomendaciones:**

  - Considerar servicios de nube reales para entornos de producción.
  - Implementar pruebas exhaustivas de calidad de datos.