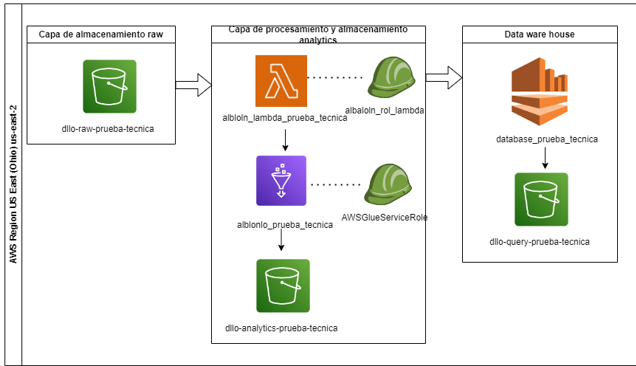
Prueba técnica

# Objetivo

Disponibilizar la información de los archivos PruebaTecnica.xml y Ventas.csv

# Arquitectura

Se utiizará AWS en la región US East (Ohio) us-east-2



## BUCKETS

### dllo-raw-prueba-tecnica

Se almacena la información de la data sin procesar

s3://dllo-raw-prueba-tecnica/clientes/PruebaTecnica.xml

s3://dllo-raw-prueba-tecnica/clientes/Ventas.csv

### dllo-analytics-prueba-tecnica

Se almacena la información transformada

s3://dllo-analytics-prueba-tecnica/financiera/dim\_cliente/

s3://dllo-analytics-prueba-tecnica/financiera/dim\_fact\_venta/

s3://dllo-analytics-prueba-tecnica/financiera/dim\_producto/

s3://dllo-analytics-prueba-tecnica/financiera/fact\_venta/

### dllo-query-prueba-tecnica

Se almacena los queries generados desde Athena.

## [**IAM**](https://us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-2#/home) **Roles**

### albaloln\_rol\_lambda

Tiene los permisos para ejecutar la lambda

### AWSGlueServiceRole

permite la ejecución del job de glue.

## Lambda

### albloln\_lambda\_prueba\_tecnica

Permite la ejecución del job de glue

## Glue

### alblonlo\_prueba\_tecnica

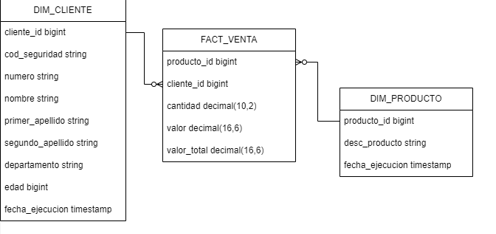
Script para procesar la información

## Athena

### database\_prueba\_tecnica

Servicio donde se disponibiliza la información.

# Modelo



## dim\_cliente

Contiene la información del cliente

* cliente\_id bigint: Indicador autogenerado. Cruza con Fact\_Venta.cliente\_id
* cod\_seguridad string: Relacionado con el xml Cliente.Informe.codSeguridad
* numero string: Relacionado con el xml Cliente.Informe.Identificacion.numero y con el csv columna ID\_CLIENTE
* nombre string: Relacionado con el xml Cliente.Informe.NaturalNacional.nombres
* primer\_apellido string: Relacionado con el xml Cliente.Informe.NaturalNacional.primerApellido
* segundo\_apellido string: Relacionado con el xml Cliente.Informe.NaturalNacional.segundoApellido
* ciudad string: Relacionado con el xml Cliente.Informe.Identificacion.ciudad
* departamento string: Relacionado con el xml Cliente.Informe.Identificacion.departamento
* edad bigint: Relacionado con el xml Cliente.Informe.Edad valor
* fecha\_ejecucion timestamp: Fecha que se guardó el registro

## dim\_producto

Contiene la información del producto.

* producto\_id bigint: Indicador autogenerado. Cruza con Fact\_Venta.producto\_id
* desc\_producto string: Relacionado con el csv columna ARTICULO
* fecha\_ejecucion timestamp: Fecha que se guardó el registro

## fact\_venta

Contiene la información de la venta

* producto\_id bigint: Cruza con Dim\_Producto.producto\_id
* cliente\_id bigint: Cruza con Dim\_Cliente.cliente\_id
* cantidad decimal(10,2): Relacionado con el csv columna CANTIDAD
* valor decimal(16,6): Relacionado con el csv columna VALOR\_UNITARIO
* valor\_total decimal(16,6): Es la multiplicación de CANTIDAD \* VALOR\_UNITARIO

# Script

## scripts\_base\_datos\_athena

Creación de las tablas en la base de datos.

## script\_creacion\_defecto

Creación del script con los valores por defecto, se debe ejecutar después de la primera corrida.

# Clases

### Constante

Clase que permite declarar las variables constantes del proceso

### Xml

Clase que permite declarar las funciones relacionadas con XML

### Util

Clase que permite declarar las funciones relacionadas con pyspark y lo que se puede cargar con este csv

### Dim

Clase que permite declarar las funciones relacionadas con Dim

### Fact

Clase que permite declarar las funciones relacionadas con Fact

### Main

Clase que permite declarar las funciones relacionadas con el procesamiento

# Proceso

Inicia con la función main.process\_information()

Carga la información del xml y csv del s3, lo lleva a un pyspark.dataframe y crea una vista en pyspark sql.

Borra la información que haya final que hay en el bucket para sobre escribirla.

Guarda en parquet la información de las dimensiones y las facts en el s3.

# Mejoras

* Con Air Flow realizar un script para que ejecute los requisitos que se requiere para que el proceso ejecute correctamente, como los scripts de base de datos ya que actualmente se ejecutan manual.
* Generar que cada vez que se actualice un archivo en AWS se ejecute la lambda y por consiguiente se ejecute el job de glue.
* Se pueda llenar la información de cliente también desde el archivo de ventas.csv