# Documentación:

## Estructura de las carpetas:

* **Datos/**: tus archivos de entrada originales.
* **SQLScripts/**: todo lo que necesitas para crear y poblar esquemas en la base de datos.
* **SSISPackege/**: implementación del ETL en SSIS.
* **SSASModel/**: definición del cubo y dimensiones en Analysis Services.
* **README.md**: guía de alto nivel para cualquiera que clone el repo.

## Estructura base de datos:

**1. StgMobility (staging)**

| **Columna** | **Descripción** |
| --- | --- |
| **sub\_region\_1** | Nombre de la región (por ejemplo, departamento o estado). |
| **date** | Fecha original en formato YYYY-MM-DD. |
| **retail\_and\_recreation\_percent\_change\_from\_baseline** | % de cambio en movilidad hacia sitios de ocio y recreación, respecto a la línea base. |
| **grocery\_and\_pharmacy\_percent\_change\_from\_baseline** | % de cambio en movilidad hacia tiendas de abarrotes y farmacias. |
| **parks\_percent\_change\_from\_baseline** | % de cambio en movilidad hacia parques y espacios abiertos. |
| **transit\_stations\_percent\_change\_from\_baseline** | % de cambio en movilidad hacia estaciones de transporte público. |
| **workplaces\_percent\_change\_from\_baseline** | % de cambio en movilidad hacia lugares de trabajo. |
| **residential\_percent\_change\_from\_baseline** | % de cambio en movilidad hacia áreas residenciales. |

* **Propósito**: Aquí se vuelca “tal cual” la información cruda que trae Google Mobility (CSV, JSON, etc.).
* **Uso**: Desde un paquete SSIS o un proceso ETL se lee esta tabla para poblar la Fact y Dimensiones.

**2. DimRegion (dimensión)**

| **Columna** | **Descripción** |
| --- | --- |
| **RegionID** | Clave surrogate o natural de la región. |
| **NombreRegion** | Nombre legible de la región (misma que sub\_region\_1). |

* **Grano**: Una fila por cada región (sub\_region\_1) única.
* **Relación**: FactMovilidad.RegionID → DimRegion.RegionID.

**3. DimFecha (dimensión)**

| **Columna** | **Descripción** |
| --- | --- |
| **FechaID** | Clave surrogate de la fecha. |
| **Año** | Año (por ejemplo, 2020). |
| **Mes** | Mes numérico (1–12). |
| **Día** | Día del mes (1–31). |
| **Trimestre** | Trimestre (1–4). |
| **FechaKeyInt** | Fecha en formato YYYYMMDD (entero). |

* **Grano**: Una fila por cada día del calendario.
* **Relación**: FactMovilidad.FechaID → DimFecha.FechaID.

**4. FactMovilidad (tabla de hechos)**

| **Columna** | **Descripción** |
| --- | --- |
| **FactID** | Clave surrogate de la fila de hecho. |
| **FechaID** | FK a DimFecha (día al que corresponden los porcentajes). |
| **RegionID** | FK a DimRegion (región a la que aplican los porcentajes). |
| **ParksPct** | Copia de parks\_percent\_change\_from\_baseline. |
| **RetailRecPct** | Copia de retail\_and\_recreation\_percent\_change\_from\_baseline. |
| **GroceryPharmPct** | Copia de grocery\_and\_pharmacy\_percent\_change\_from\_baseline. |
| **TransitPct** | Copia de transit\_stations\_percent\_change\_from\_baseline. |
| **WorkplacesPct** | Copia de workplaces\_percent\_change\_from\_baseline. |
| **ResidentialPct** | Copia de residential\_percent\_change\_from\_baseline. |

* **Grano**: Un registro por combinación *(región, fecha)*.
* **Hechos**: Son los valores de porcentaje de cambio de movilidad.
* **Relaciones**:
  + Con DimFecha a través de **FechaID**.
  + Con DimRegion a través de **RegionID**.

**Flujo típico ETL**

1. **Extraer** de la fuente (CSV/JSON) a **StgMobility**.
2. **Limpiar/Transformar**:
   * Normalizar nombres de región y fechas.
   * Convertir date → FechaKeyInt y luego lookup en **DimFecha**.
   * Lookup de sub\_region\_1 en **DimRegion** para obtener RegionID.
3. **Cargar** en **FactMovilidad** los valores numéricos y las claves foráneas.

Con ese diseño ya tienes un cubo SSAS bien modelado: las dimensiones de Fecha y Región filtrarán tus medidas de movilidad por distintos ejes.

Diagrama

El contenido generado por IA puede ser incorrecto.

Tratamiento de los datos:  
Interfaz de usuario gráfica, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

**1. DFT\_CargaInicial\_Mobility**

**Propósito:**  
– Extraer los datos crudos (por ejemplo, CSV o API) y volcarlos en la tabla de staging (StgMobility).  
– Realizar validaciones básicas (tipo de datos, campos obligatorios) y limpieza mínima para asegurar que el origen esté listo para transformaciones posteriores.

**Objetivos clave:**

* Centralizar el punto de entrada de los datos de movilidad
* Aislar al detalle de cambios en el esquema del origen
* Facilitar auditoría y debugging (esta tabla conserva el histórico original)

**2. DFT\_Populate\_DimFecha**

**Propósito:**  
– Construir la dimensión de fechas (DimFecha) a partir de los valores de fecha únicos de StgMobility.  
– Generar atributos adicionales (año, mes, día, trimestre, FechaKeyInt) necesarios para análisis temporal.

**Objetivos clave:**

* Garantizar que cada fecha tenga su propio surrogate key (FechaID)
* Precalcular jerarquías y claves enteras para optimizar el cubo y joins
* Evitar regenerar fechas duplicadas en ejecuciones posteriores (usar “lookup + insert if not exists”)

**3. DFT\_Populate\_DimRegion**

**Propósito:**  
– Llenar la dimensión de región (DimRegion) con los valores únicos de subregión (departamento) del staging.  
– Normalizar nombres (US-ASCII vs. caracteres acentuados) y asignar surrogate keys (RegionID).

**Objetivos clave:**

* Estandarizar la denominación de regiones (e.g. “Usulután” vs. “Usulután”)
* Mantener un catálogo maestro de regiones que nunca cambie
* Permitir cambios futuros solo vía mantenimiento de dimensión sin tocar el hecho

**4. DFT\_Load\_FactMovilidad**

**Propósito:**  
– Construir la tabla de hechos (FactMovilidad) combinando los datos de StgMobility con las dimensiones pobladas.  
– Usar componentes **Lookup** para recuperar FechaID y RegionID a partir de los valores de fecha y región.  
– Cargar los porcentajes de cambio (ParksPct, TransitPct, etc.) como medidas aditivas.

**Objetivos clave:**

* Asegurar integridad referencial: cada registro de hechos apunta a dimensiones preexistentes
* Convertir formatos y tipos (varchar→date, string→int surrogate key)
* Dejar los datos listos para el procesamiento del cubo SSAS

## Procedimiento cubo:

## Captura de pantalla de computadora El contenido generado por IA puede ser incorrecto.

**Cube Structure**

* **Measures**:
  + Parks Pct, Retail Rec Pct, Grocery Pharm Pct, Transit Pct, Workplaces Pct, Residential Pct  
    – Todos son los porcentajes de cambio que importas desde FactMovilidad.
  + Fact Movilidad Count  
    – Un contador de filas de la tabla de hechos.
* **Dimensions**:
  + **Dim Fecha**  
    – Debes marcarla como **Time Dimension** y crear al menos una jerarquía (Año → Trimestre → Mes → Día).
  + **Dim Region**  
    – Te permite filtrar los datos por región (departamento, estado, etc.).

**Data Source View**

* **FactMovilidad**  
  – Tabla de hechos con columnas FechaID, RegionID y tus cinco campos de porcentaje.
* **DimFecha**  
  – Tabla de dimensión de tiempo con atributos Año, Mes, Día, Trimestre, FechaKeyInt.
* **DimRegion**  
  – Tabla de dimensión con RegionID y NombreRegion.

**Error List (para resolver antes de desplegar)**

* **Semi-additive measure requires a time dimension**  
  – Tu medida de recuento está configurada como semi-aditiva; necesita que Dim Fecha esté marcada como dimensión de tiempo.
* **Create hierarchies in non-parent child dimensions**  
  – Construye al menos la jerarquía Año→Mes (o Año→Trimestre→Mes→Día) en DimFecha.

**Solution Explorer**

* **Data Sources**: Conexión “Movilidad COVID” hacia tu base relacional.
* **Data Source Views**: El .dsv que reúne FactMovilidad, DimFecha y DimRegion.
* **Cubes**: El archivo Movilidad COVID.cube donde defines medidas y vinculas las dimensiones.
* **Dimensions**: Los dos archivos de dimensión (Dim Fecha.dim, Dim Region.dim) que luego asignas al cubo.