IMF Business School

# Hoja de respuestas

INTELIGENCIA DE NEGOCIO Y VISUALIZACIÓN

Violeta Chamosa Fondevila

20-09-2022

Hoy en día es imprescindible para las empresas analizar los datos generados en el negocio para poder extraer conocimiento de los mismos y usarlos como ventaja competitiva. Esto es lo que se conoce como inteligencia de negocio o Business Inteligence (BI).

La inteligencia de negocio ayuda a las compañías a tomar decisiones basadas en hechos e identificar oportunidades de negocio haciendo mejorar aspectos como las ventas o las compras, entre muchos otros. También establece un único lenguaje y nos proporciona una única verdad de la realidad de la compañía. En resumen, nos permite tener una imagen clara de la posición de la empresa respecto a sus competidores y al mercado.

En este ejercicio vamos a hacer uso de esta práctica empresarial en un caso basado en datos del Grupo World Bank, en concreto, se utilizará un conjunto de indicadores sobre la burocracia mundial.

Los indicadores de la burocracia mundial (WWBI) son un conjunto de datos sobre el empleo y los salarios del sector público que pueden ayudar a los investigadores y profesionales del desarrollo a obtener una mejor comprensión de las dimensiones del personal contratado por los estados, la huella del sector público en el mercado laboral general y en el fiscal y las implicaciones de la factura salarial del gobierno.

## 1. ¿QUÉ DATOS SE USARÁN?

En el caso se van a usar dos documentos estructurados, es decir, con una estructura interna identificable, en formato CSV. Antes de ponernos a trabajar con dichos datos de origen, vamos a realizar un pequeño análisis de los mismos para evaluar su formato, lo que nos ayudará a comprender mejor su contenido y facilitar su análisis.

El primero de los documentos, nombrado WWBICountry, cuenta con 29 columnas, entre las cuales podemos encontrar el código de país, el nombre corto del país, el nombre completo, la unidad de la moneda de curso legal en el país, la región a la que pertenece dentro del continente, a que grupo de riqueza pertenece, entre otros muchos campos.

El segundo documento nos proporciona una serie de valores para unos indicadores o métricas según el país y el año en el que fueron recogidos los datos. Contiene columnas para cada país con el nombre del país y su código, otras dos con las métricas y el código de las mismas y una por cada año de estudio.

Así como en el primero de ellos no nos encontramos muchos valores nulos, por el contrario, en este otro, llamado WWBIData si que nos encontramos una cantidad considerable de valores nulos, los que veremos cómo tratar más adelante para evitar distorsiones en los resultados.

Estos dos archivos se van a cargar en el área a *staging*, donde almacenamos los datos en formato original y seguidamente, al *datamart*, donde estos datos en bruto podrán ser explotados por los usuarios.

La estructura de los datos cargados en *staging* es la siguiente. Hemos creado una tabla llamada WWBI Country con todas las columnas que aparecen en el CSV. También se ha elaborado una segunda tabla donde hemos cargado los datos del segundo documento, la cual hemos nombrado WWBI Data.

El *Datamart* consiste en 4 tablas conectadas. Tres ellas son tablas de dimensiones (métricas, tiempo y país) y la otra es una tabla de hechos.

La estructura del *datamart* se puede observar en el siguiente diagrama de entidad-relación del modelo. Tambien podemos ver la cardinalidad 1 a N desde las metricas hasta la fact table.

**Figura 1: Diagrama entidad-relación del datamart**

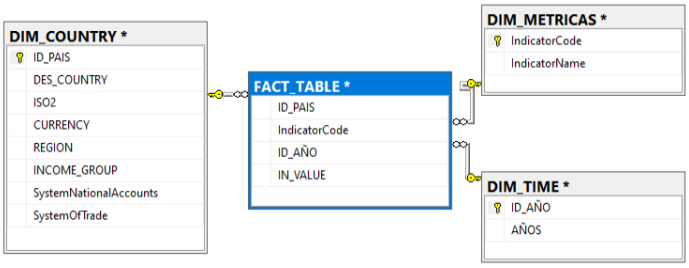



**Figura 1:** Diagrama entidad-relación del datamart

## 2. EXTRACCIÓN DE LOS CSV A UNA BBDD DE *STAGING*

Como hemos mencionado antes, el área staging es donde almacenamos temporalmente los datos entre los archivos originales, en este caso los dos archivos planos que contienen los detalles de cada país y los datos de los distintos KPIs, y el datamart. Las tablas dentro de esta área contienen datos sobre el negocio en formato original para ser procesadas posteriormente en las tareas ETL (Extraction, Transformation and Load).

Para hacer la extracción de los datos en la base de datos staging vamos a hacer uso de una herramienta de software abierto llamada Kettle que pertenece a Pentaho, también conocida por Pentaho's Data Intergaton (PDI). En concreto, utilizaremos el subprograma conocido como Spoon.

### 2.1. Creación de la base de datos staging y sus tablas

Para la creación de la base de datos y de las dos tablas que la componen hemos usado el asistente Management Studio. Hemos nombrado a la base de datos STG_Module4 y hemos creado dos tablas. Una de ellas contiene los detalles de los países, WWBI_Country y, la segunda, los datos de los distintos indicadores, WWBI_Data.

A continuación, adjuntamos los tres scripts, uno de la BBDD y los otros de las tablas.

**Figura 2: BASE DE DATOS STAGING**

```
USE [master]
GO

/****** Object:  Database [STG_Module4]    Script Date: 12/09/2022 12:57:51 ******/
CREATE DATABASE [STG_Module4]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'STG_Module4', FILENAME = N'C:\Program Files\Microsoft SQL Server\
MSSQL15.SQLEXPRESS\MSSQL\DATA\STG_Module4.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'STG_Module4_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\
MSSQL15.SQLEXPRESS\MSSQL\DATA\STG_Module4_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
 WITH CATALOG_COLLATION = DATABASE_DEFAULT
GO
```

**Figura 2:** Script de la creación de la base de datos staging

**Figura 3: TABLA DE PAISES**

```
USE [STG_Module4]
GO

/****** Object:  Table [dbo].[WWBI_Country]    Script Date: 12/09/2022 13:03:04 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[WWBI_Country](
    [CountryCode] [varchar](500) NULL,
    [ShortName] [varchar](500) NULL,
    [TableName] [varchar](500) NULL,
    [LongName] [varchar](500) NULL,
    [2-alphaCode] [varchar](500) NULL,
    [CurrencyUnit] [varchar](500) NULL,
    [Region] [varchar](500) NULL,
    [IncomeGroup] [varchar](500) NULL,
    [WB-2 _code] [varchar](500) NULL,
    [NationalAccountsBaseYear] [varchar](500) NULL,
    [NationalAccountsReferenceYear] [varchar](50) NULL,
    [SNAPriceValuation] [varchar](500) NULL,
    [LendingCategory] [varchar](500) NULL,
    [OtherGroups] [varchar](500) NULL,
    [SystemOfNationalAccounts] [varchar](500) NULL,
    [AlternativeConversionFactor] [varchar](500) NULL,
    [PPPSurveyYear] [varchar](500) NULL,
    [BalanceOfPaymentsManualInUse] [varchar](500) NULL,
    [ExternalDebtReportingStatus] [varchar](500) NULL,
    [SystemOfTrade] [varchar](500) NULL,
    [GovernmentAccountingConcept] [varchar](500) NULL,
    [IMFDataDisseminationStandard] [varchar](500) NULL,
    [LatestPopulationCensus] [varchar](500) NULL,
    [LatestHouseholdSurvey] [varchar](500) NULL,
    [SourceOfMostRecentIncomeAndExpenditureData] [varchar](500) NULL,
    [VitalRegistrationComplete] [varchar](500) NULL,
    [LatestAgriculturalCensus] [varchar](500) NULL,
    [LatestIndustrialData] [varchar](500) NULL,
    [LatestTradeData] [varchar](500) NULL
) ON [PRIMARY]
GO
```

**Figura 3:** Script de la creación de la creación de la tabla de países

**Figura 4: TABLA DE DATOS**

```
USE [STG_Module4]
GO

/****** Object:  Table [dbo].[WWBI_Data]    Script Date: 12/09/2022 13:05:13 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[WWBI_Data](
    [CountryName] [varchar](500) NULL,
    [CountryCode] [varchar](500) NULL,
    [IndicatorName] [varchar](500) NULL,
    [IndicatorCode] [varchar](500) NULL,
    [a2000] [varchar](500) NULL,
    [a2001] [varchar](500) NULL,
    [a2002] [varchar](500) NULL,
    [a2003] [varchar](500) NULL,
    [a2004] [varchar](500) NULL,
    [a2005] [varchar](500) NULL,
    [a2006] [varchar](500) NULL,
    [a2007] [varchar](500) NULL,
    [a2008] [varchar](500) NULL,
    [a2009] [varchar](500) NULL,
    [a2010] [varchar](500) NULL,
    [a2011] [varchar](500) NULL,
    [a2012] [varchar](500) NULL,
    [a2013] [varchar](500) NULL,
    [a2014] [varchar](500) NULL,
    [a2015] [varchar](500) NULL,
    [a2016] [varchar](500) NULL
) ON [PRIMARY]
GO
```

**Figura 4:** Script de la creación de la tabla de datos

## 2.2. Cuestiones

Para realizar la carga de datos a través de la herramienta Spoon, he decidido crear dos transformaciones ya que el archivo de datos es bastante extenso y pienso que es más eficiente hacerlo de esta manera.

En la primera transformación, nombrada como STG_Tabla_Country, he usado tres objetos, siendo estos:

- Entrada de archivo CSV (WWBICountry)
- Mapping de los datos entre el CSV y la tabla de los países
- Una salida de tabla conectada a la tabla de países en la base de datos staging

**Figura 5: CARGA DE STG_Tabla_Country**


**Figura 5:** Carga de STG_Tabla_Country

Para la segunda, guardada como STG_Tabla_Data, he usado también tres objetos, siendo estos:

- Entrada de archivo CSV (WWBIData)
- Mapping de los datos entre el CSV y la tabla de los datos
- Una salida de tabla conectada a la tabla de datos en la base de datos staging

**Figura 6: CARGA DE STG_Tabla_Data**

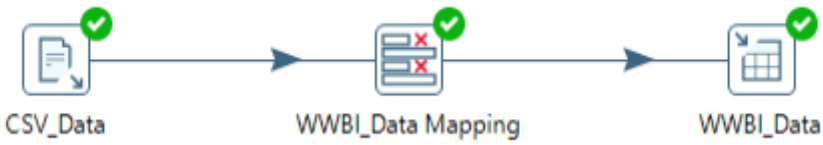



**Figura 5:** Carga de STG_Tabla_Data

En la primera transformación, se han cargado 115 filas en la tabla de país en la BD de Staging, lo que quiere decir que tenemos una muestra de 115 países diferentes.

En la segunda tabla, la de datos, se han cargado 10005 filas. Es decir, haciendo la división, por cada uno de los 115 países tenemos 87 indicadores diferentes.

Por último, relativo a la pregunta del uso del componente Start, no lo he usado ya que solo he realizado dos transformaciones. El componente Start es el primer elemento del trabajo o job que realizaré más adelante para compilar las distintas transformaciones en él.

## 3. TRANSFORMAR Y CARGAR LOS DATOS DESDE STAGING AL *DATAMART*

El data warehouse es un almacén de información para su análisis en las organizaciones, es decir, un repositorio de datos listo para ser explotado por aquellos usuarios capacitados. Son independientes de otros sistemas lo que garantiza que las consultas que se realizan sobre los datos en el data warehouse no afectan a la operativa de la organización. Pueden almacenar datos de diversos orígenes y diferentes estructuras los cuales se deben integrar para poder proveer la información necesaria para el usuario en el momento adecuado y de forma correcta.

En este apartado vamos a crear un data warehouse con 4 tablas, tres de ellas serán de dimensiones y la otra será una tabla de hechos. Las tablas de dimensiones son las que almacenan variables por las que se analizan la información, por ejemplo, el país, el año y el indicador. Por otro lado, la tabla de hechos guarda lo que ha ocurrido, es decir, almacenan métricas o valores cuantificables.

Existen dos modelos para estructurar un data warehouse; el modelo estrella, caracterizado por ser una tabla de hechos rodeada de tablas de dimensiones; y el modelo copo de nieve, surgido al encontrarnos más de una tabla de hechos. En este caso práctico vamos a hacer uso del modelo estrella, como podemos observar en la figura 1.

### 3.1. Creación del Data warehouse y sus tablas

A continuación, vamos a adjuntar los scripts de la creación de la base de datos data warehouse y los distintos elementos que la componen.

**Figura 7: CREACIÓN DE LA BASE DE DATOS DATAWAREHOUSE**

```
USE [master]
GO

/****** Object:  Database [DWH_Module4]    Script Date: 13/09/2022 12:03:05 ******/
CREATE DATABASE [DWH_Module4]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'DWH_Module4', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\
MSSQL\DATA\DWH_Module4.mdf' , SIZE = 73728KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'DWH_Module4_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\
MSSQL\DATA\DWH_Module4_log.ldf' , SIZE = 73728KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
 WITH CATALOG_COLLATION = DATABASE_DEFAULT
GO
```

**Figura 7:** Base de datos data warehouse

**Figura 8: CREACION DE LA TABLA DIMENSION MÉTRICAS**

```
USE [DWH_Module4]
GO

/****** Object:  Table [dbo].[DIM_METRICAS]    Script Date: 13/09/2022 12:04:22 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[DIM_METRICAS](
    [IndicatorCode] [varchar](500) NOT NULL,
    [IndicatorName] [varchar](500) NULL,
 CONSTRAINT [PK_DIM_METRICAS] PRIMARY KEY CLUSTERED
(
    [IndicatorCode] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**Figura 8:** Tabla de la dimensión métrica

**Figura 9: CREACION DE LA TABLA DIMENSIÓN COUNTRY**

```
USE [DWH_Module4]
GO

/****** Object:  Table [dbo].[DIM_COUNTRY]    Script Date: 13/09/2022 12:03:47 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[DIM_COUNTRY](
    [ID_PAIS] [varchar](500) NOT NULL,
    [DES_COUNTRY] [varchar](500) NULL,
    [ISO2] [varchar](500) NULL,
    [CURRENCY] [varchar](500) NULL,
    [REGION] [varchar](500) NULL,
    [INCOME_GROUP] [varchar](500) NULL,
    [SystemNationalAccounts] [varchar](500) NULL,
    [SystemOfTrade] [varchar](500) NULL,
 CONSTRAINT [PK_DIM_COUNTRY] PRIMARY KEY CLUSTERED
(
    [ID_PAIS] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**Figura 9:** Tabla de la dimensión country

**Figura 10: CREACION DE LA DIMENSION TIME**

```
USE [DWH_Module4]
GO

/****** Object:  Table [dbo].[DIM_TIME]    Script Date: 15/09/2022 16:42:37 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[DIM_TIME](
    [ID_AÑO] [int] NOT NULL,
    [AÑOS] [int] NULL,
 CONSTRAINT [PK_DIM_TIME] PRIMARY KEY CLUSTERED
(
    [ID_AÑO] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**Figura 10:** Tabla de dimensión time

**Figura 11: CREACION DE LA TABLA DE HECHOS**

```
USE [DWH_Module4]
GO

/****** Object:  Table [dbo].[FACT_TABLE]    Script Date: 13/09/2022 12:04:57 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[FACT_TABLE](
    [ID_PAIS] [varchar](500) NULL,
    [IndicatorCode] [varchar](500) NULL,
    [ID_AÑO] [int] NULL,
    [IN_VALUE] [float] NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[FACT_TABLE]  WITH CHECK ADD  CONSTRAINT [FK_FACT_TABLE_DIM_COUNTRY] FOREIGN KEY([ID_PAIS])
REFERENCES [dbo].[DIM_COUNTRY] ([ID_PAIS])
GO

ALTER TABLE [dbo].[FACT_TABLE] CHECK CONSTRAINT [FK_FACT_TABLE_DIM_COUNTRY]
GO

ALTER TABLE [dbo].[FACT_TABLE]  WITH CHECK ADD  CONSTRAINT [FK_FACT_TABLE_DIM_METRICAS] FOREIGN KEY([IndicatorCode])
REFERENCES [dbo].[DIM_METRICAS] ([IndicatorCode])
GO

ALTER TABLE [dbo].[FACT_TABLE] CHECK CONSTRAINT [FK_FACT_TABLE_DIM_METRICAS]
GO
```

**Figura 11:** Tabla de hechos

### 3.2. Cuestiones

Para la carga de los datos he usado tres transformaciones diferentes. En la primera de ellas he cargado los datos de las tablas de dimensiones país y métricas, en la segunda, la dimensión tiempo y, en la tercera y última, la tabla de hechos.

En la primera transformación y para la tabla de la dimensión métrica he usado 3 pasos:

- Una entrada de la tabla con los datos en el área staging (STG_Tabla_Data)
- Un mapeo entre las dos tablas
- Una salida a la tabla de dimensión métrica conectada a la BBDD data warehouse

En esta misma transformación y para la tabla de la dimensión país he usado también 3 pasos:

- Una entrada de la tabla con los países en el área staging (STG_Tabla_Country)
- Un mapeo entre las dos tablas
- Una salida a la tabla de dimensión país conectada a la BBDD data warehouse

**Figura 12: CARGA DE LAS DIMENSIONES PAÍS Y MÉTRICAS**

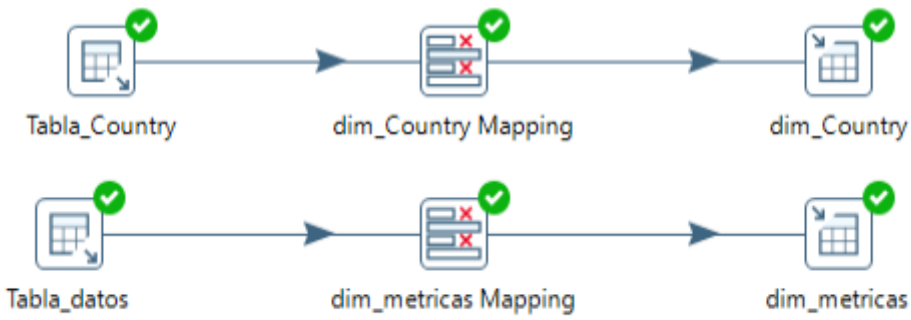



**Figura 12:** Carga de las dimensiones país y métricas

Para la trasformación de la dimensión tiempo he usado 5 objetos:

- Una entrada de los datos con los distintos años ya normalizados en una columna
- Una ordenación de los mismos ya que no estaban ordenados
- Una secuenciación de 1 a 16 para usar como clave primaria
- Un mapeo entre las tablas
- Una salida a la tabla de dimensión tiempo en el data warehouse

**Figura 13: CARGA DE LA DIMENSIÓN TIEMPO**

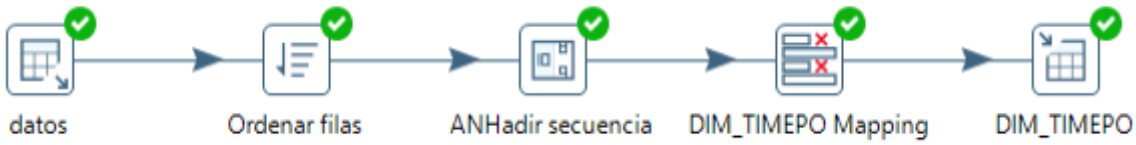



**Figura 13:** Carga de la dimensión tiempo

Se han cargado 87 filas para la tabla dimensión métricas, es decir, tenemos 87 métricas diferentes para cada país. Por la parte de la dimensión de país, han sido un total de 115, lo que nos dice que los datos se han obtenido en 115 países. En la dimensión años se han cargado 17, desde 2000 hasta 2016, es decir, un total de 17 años de recogida de datos.

Para la tabla de hechos o fact table he incluido 4 componentes: el código del país, el código del indicador, el año en el que se evalúa el indicador y el valor del indicador. En esta transformación se han cargado 170085 filas en total. Como podemos observar el número de filas se ha visto multiplicado por 17 ya que aparece una fila por cada año, país e indicador. Es decir, tenemos 115 países con 87 métricas y 17 años de registros. Si multiplicamos estos tres valores obtenemos el número de filas totales.

Para la carga y transformación de esta tabla he usado 4 pasos:

- Una entrada de la tabla de datos del área staging (STG_Tabla_Data)
- Una transformación para normalizar la fila de años
- Un mapeo entre la tabla normalizada de datos y la tabla de hechos
- Una salida a la tabla de hechos en el data warehouse

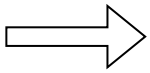
**Figura 14: CARGA DE LA TABLA DE HECHOS**


**Figura 14:** Carga de la tabla de hechos

**Nota:** La normalización de una fila quiere decir que pivotamos la fila en donde aparecen los distintos años para que se transforme en una columna. Vamos a ver un ejemplo visual de esta transformación. Empezamos teniendo los datos estructurados como en la figura 15 para acabar teniendo los datos como la figura 16.

**Figura 15 y 16: PIVOTAR TABLA**

| 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 |
|---|---|---|---|---|---|---|---|
| data | data | data | data | data | data | data | data |
| data | data | data | data | data | data | data | data |
| data | data | data | data | data | data | data | data |



| | | | |
|---|---|---|---|
| **2000** | data | data | data |
| **2001** | data | data | data |
| **2002** | data | data | data |
| **2003** | data | data | data |
| **2004** | data | data | data |
| **2005** | data | data | data |
| **2006** | data | data | data |
| **2007** | data | data | data |

**Figura 15 y 16:** Pivotar tablas

## 4. CREAR LA TAREA QUE PERMITA CARGAR TODO EL *DATAMART*

Una vez realizadas todas las transformaciones necesarias vamos a proceder a crear un trabajo o job. Un job son conjuntos de transformaciones que se procesan para realizar tareas determinadas. Su función es orquestar las distintas transformaciones, es decir, permitir la ejecución de los procesos de integración de los datos sincronizadamente.

Para ello vamos a usar una tarea que integre los datos de las distintas fuentes siguiendo un flujo de las cinco transformaciones realizadas anteriormente. Este flujo va a comenzar cargando los archivos al área staging para continuar cargando las tablas de dimensiones y terminar cargando la tabla de hechos. No se hace uso de una transformación ya que el propósito del trabajo es orquestar los datos, no transformarlos.

Los objetos usados para crear el trabajo o job son los siguientes:

- El objeto START, el cual no necesita configuración
- La transformación donde cargamos los datos de los países al área staging
- La transformación donde cargamos los datos al área staging
- La transformación donde cargamos las tablas de dimensiones en el DWH
- La transformación de la dimensión tiempo
- La transformación de la carga de la tabla de hechos

Resultando el siguiente esquema del flujo de las tareas:

**Figura 17: FLUJO DE TAREAS EN EL JOB**

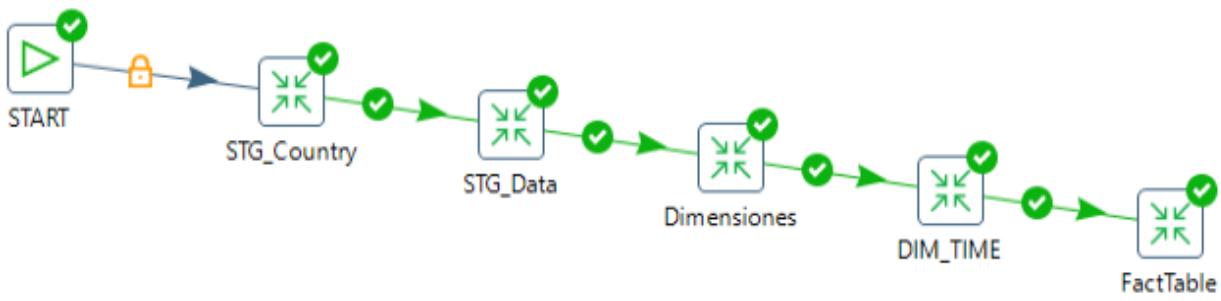



**Figura 17:** Flujo de tareas en el job

## 5. CONSULTAS SQL

Para realizar consultas en SQL se usan las sentencias referentes al lenguaje de consulta de datos o Data Query Language (DQL). Para ello vamos a hacer uso del único comando existente: SELECT.

### 5.1. ¿Cuántos países pertenecen a cada grupo de ingresos (income group)?

Para resolver esta consulta vamos a usar la tabla de dimensión país del data warehouse. Como comenté antes, vamos a usar el comando SELECT para contar el numero de países pertenecientes a cada grupo de ingresos con el comando COUNT. Para visualizar los resultados de una manera clara vamos a agrupar por el grupo de ingresos con GROUP BY. Quedando el código de la siguiente manera:

```
SELECT [INCOME_GROUP]
    ,COUNT([ID_PAIS]) Numero_de_paises
FROM [DWH_Module4].dbo.DIM_COUNTRY
GROUP BY [INCOME_GROUP]
```

Los resultados obtenidos quedan reflejados en la siguiente tabla:

**Figura 18: CONSULTA AGRUPADA POR GRUPOS DE INGRESOS**

| | INCOME_GROUP | Numero_de_paises |
|---|---|---|
| 1 | High income | 21 |
| 2 | Low income | 27 |
| 3 | Lower middle income | 37 |
| 4 | Upper middle income | 30 |

**Figura 18:** Consulta de cantidad de países agrupados por el grupo de ingresos

En la tabla observamos que el grupo mayoritario es el de ingresos medio bajos, seguido del de ingresos medio altos, continuando con los ingresos bajos y, por último, el minoritario en cuanto a cantidad de paises es el de ingresoso altos.

### 5.2. ¿Cuántas métricas existen? ¿Y que tengan valor no nulo en el año 2000?

El número de métricas es 87. Para realizar esta consulta he usado el siguiente código:

```
SELECT Count (DISTINCT [IndicatorCode]) Numero_Indicadores
FROM [DWH_Module4].dbo.DIM_METRICAS
```

Para la siguiente consulta he usado el comando SELECT para seleccionar la columna de indicadores de la base de datos del data warehouse, en concreto, la tabla de hechos. También utilicé el comando WHERE para encontrar los valores para el año 2000 y que no sean nulos. Por último, los he agrupado por el código del indicador para saber cuántos indicadores diferentes tienen, aunque sea para un solo país, un valor de la métrica para el año 2000.

```
SELECT [IndicatorCode]
FROM [DWH_Module4].dbo.FACT_TABLE
WHERE ID_AÑO=2000 AND IN_VALUE is not null
GROUP BY [IndicatorCode]
```

Con esta consulta obtenemos un resultado de 85 filas. Es decir, de las 87 métricas diferentes que tenemos en un principio, para el año 2000, hay 85 que no tienen valores nulos para, al menos, uno de los países.

En el caso de que queramos saber cuantos valores en total tenemos con datos que no sean nulos, habría que eliminar el comando GROUP BY. Al realizar esta operación vemos que nos aparece una tabla con los indicadores con 989 filas, es decir, tenemos 989 valores en el 2000 que no son nulos.

Si queremos saber para cuantos países tenemos valores no nulos en cada métrica, es tan sencillo como añadir el comando COUNT con los IDs de los países, quedando el código de la siguiente forma:

```
SELECT [IndicatorCode]
      ,COUNT([ID_PAIS])
FROM DWH_Module4.dbo.FACT_TABLE
WHERE ID_AÑO=2000 AND IN_VALUE is not null
GROUP BY [IndicatorCode]
```

## 6. CREAR UN INFORME EN POWER BI ACCEDIENDO AL *DATAMART*

Los seres humanos percibimos mejor la información a través de visualizaciones ya que nos permiten tener una mejor interpretación de los mismo y de manera más rápida.

La gran cantidad de datos que se manejan hoy en día, muchas veces, son difíciles de analizar simplemente a través de tablas, por ello necesitamos de herramientas que nos ayuden a visualizar los datos para poder obtener conocimiento de los mismos.

La representación de los datos permite a los usuarios identificar rápidamente patrones clave o insights que nos ayuden a tomar decisiones clave acerca del negocio. Es a través de estas capacidades, entender y asimilar la información, lo que hace que el conocimiento que obtenemos nos ayude a alcanzar la sabiduría.

Para ello vamos a elaborar un informe con la aplicación PowerBI. Un informe o reporte es una herramienta de visualización y análisis que contienen información detallada sobre un tema. En este caso, vamos a crear un informe de análisis o consultas ad hoc para realizar un análisis sobre el conjunto de datos.

### 6.1. Indicar la estructura del modelo de datos.

Este modelo es en forma de estrella y cuenta con un total de cuatro tablas; tres de ellas son de dimensiones y la cuarta es la tabla de hechos. Podemos ver una representación del mismo realizada en PowerBI en la siguiente figura:

**Figura 19: ESTRUCTURA DEL MODELO DE DATOS**

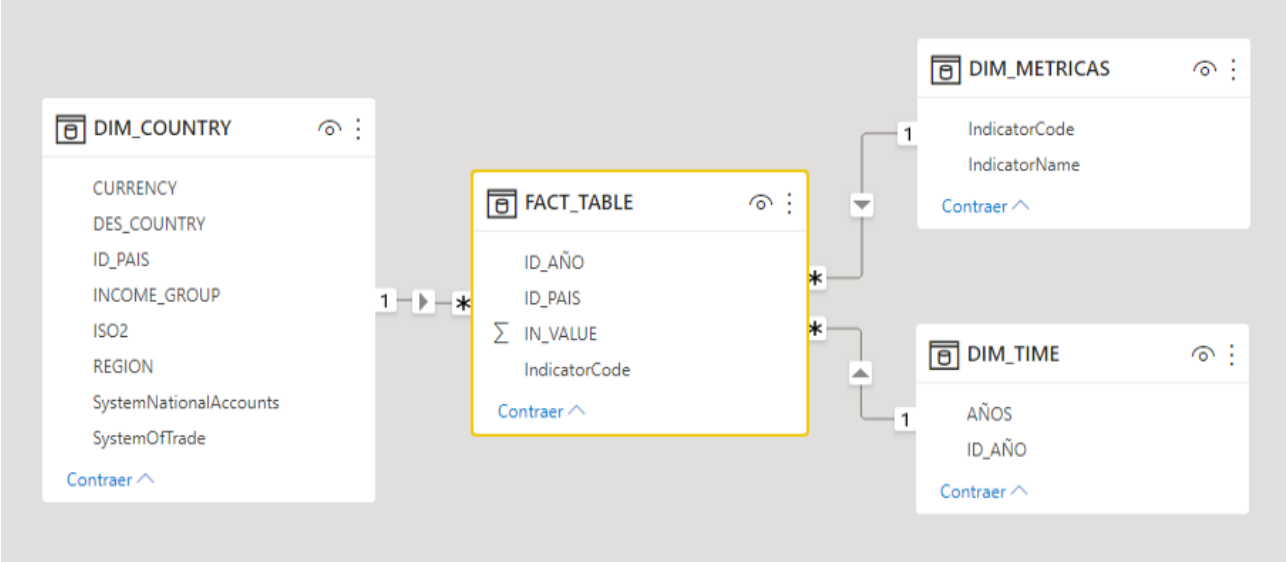



**Figura 19:** Estructura del modelo de datos

A continuación, vamos a crear una serie de visualizaciones sobre estos datos.

## 6.2. Visualizaciones

Para realizar una visualización en PowerBI existen innumerables tipos de gráficos. Dependiendo de lo que queramos representar debemos decidir cual es la mejor manera de proyectar la información y que permita transmitir el mensaje fácilmente.

Para elegir el grafico adecuado hay que tener en cuenta tres factores:

- El numero de variables que participan
- Cuantos datos vamos a mostrar
- El periodo que representan esos datos

A continuación, vamos a representar 4 visualizaciones.

### *6.2.1. Evolución en el tiempo del "Empleo del sector público como parte del empleo remunerado" y el "Empleo del sector público como parte del empleo formal" para Argentina.*

Para este caso, voy a usar el grafico de líneas. Estos gráficos son usados para mostrar valores cuantitativos asociados a la dimensión tiempo. Son útiles para mostrar tendencias en ciertos periodos, es decir, para identificar las tendencias y analizar cómo cambian los datos según la variable tiempo. En ellos, se pueden analizar más de una serie a la vez lo que nos permite compáralas entre ellas.

A continuación, se muestra el gráfico obtenido seguido de la explicación de como se ha realizado, incluyendo filtros usados, variables mostradas en los ejes y una explicación simple de la leyenda.

**Gráfico 1: EVOLUCIÓN EN EL TIEMPO PARA ARGENTINA DEL EMPLEO PUBLICO**

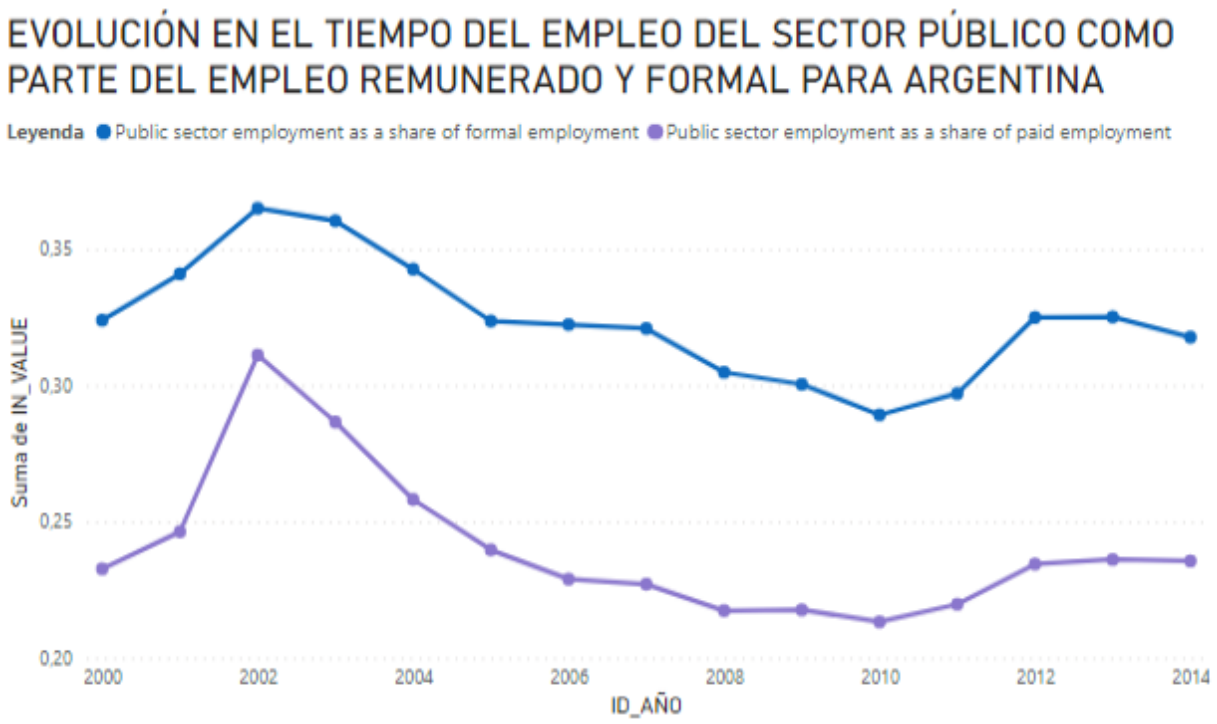



**Gráfico 1:** Evolución en el tiempo para argentina del empleo publico

En primer lugar, he seleccionado de la tabla de hechos el campo ID_AÑO para el eje X, lo que nos permitirá ir viendo como evolucionan los datos del eje Y a lo largo del tiempo. Para este segundo eje, el Y, he introducido la suma de los valores IN_VALUE de la tabla de hechos.

Seguidamente, he usado dos filtros para seleccionar un conjunto de datos concretos. Para el primero he usado el ID_PAIS para seleccionar el país Argentina (ARG) y, en el segundo, el IndicatorName para mostrar solo las métricas de “Public sector employment as a share of paid employment” y “Public sector employment as a share of formal employment”, los cuales podemos ver en la leyenda, siendo el primero de ellos la línea de color morado y la segunda métrica, la de color azul.

### *6.2.2. Evaluar la edad media de los empleados del sector privado y público por región.*

Los gráficos de barras son usados para comparar variables discretas o numéricas entre un conjunto de categorías, en este caso van a ser las regiones.

Hay varios tipos de gráficos de barras como, por ejemplo, los agrupados, los apilados, entre otros. En este caso vamos a usar un grafico de barras agrupado donde podremos ver el sector privado y publico por separado para la misma región, lo que nos permitirá evaluarlo más fácilmente y compararlos.

**Nota:** se puede añadir una jerarquía si se quiere ver más específicamente añadiendo el campo país, lo que nos permite ver en mayor detalle las diferencias por regiones.

**Gráfico 2: EDAD MEDIA DE LOS EMPLEADOS DEL SECTOR PÚBLICO Y PRIVADO POR REGIONES**

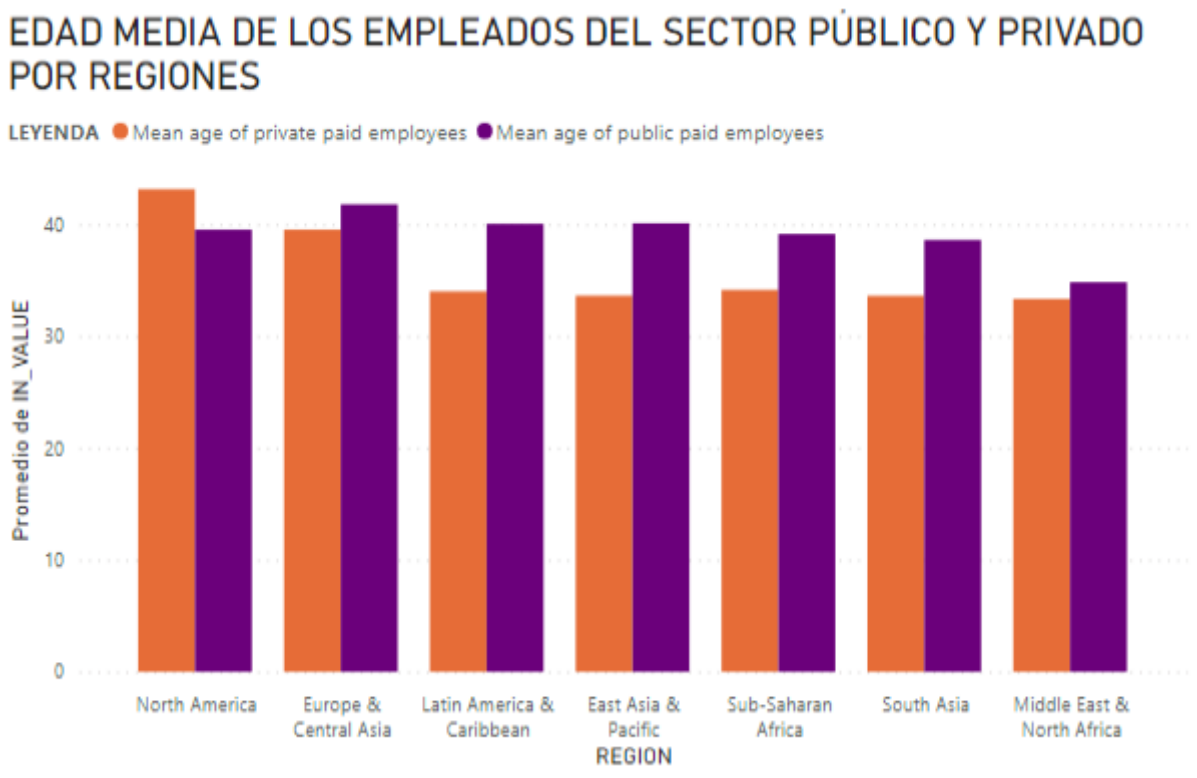



**Gráfico 2:** Edad media de los empleados del sector público y privado por regiones

Para la realización del gráfico, he usado las regiones de la tabla de dimensión país en el eje X y en el eje Y, he representado el valor promedio de las medias de las edades ya que hay muchos valores nulos para algunas de las regiones y así solo tiene en cuenta los valores que no son nulos.

El filtro usado para este objeto visual es el campo de IndicatorName seleccionando las categorías de “Mean age of private paid employees” y “Mean age of public paid employees”. Estas dos categorías son las que forman la leyenda del gráfico; en color naranja podemos ver los empleados del sector privado y en morado, los del sector público.

### *6.2.3. Realizar una gráfica del promedio del peso relativo de los cargos técnicos en los sectores privados y públicos a lo largo del tiempo.*

Para hacer la representación vamos a hacer uso el diagrama de áreas, el cual nos permite mostrar la evolución de una variable cuantitativa en el tiempo. Es útil en los casos donde se quiere representar la evolución del peso relativo de variables cuantitativas, permitiéndonos observar el volumen total de las métricas y el individual de cada una de ellas.

**Gráfico 3: EVOLUCIÓN DEL PESO RELATIVO DE LOS CARGOS TÉCNICOS EN EL SECTOR PÚBLICO Y PRIVADO EN EL TIEMPO**

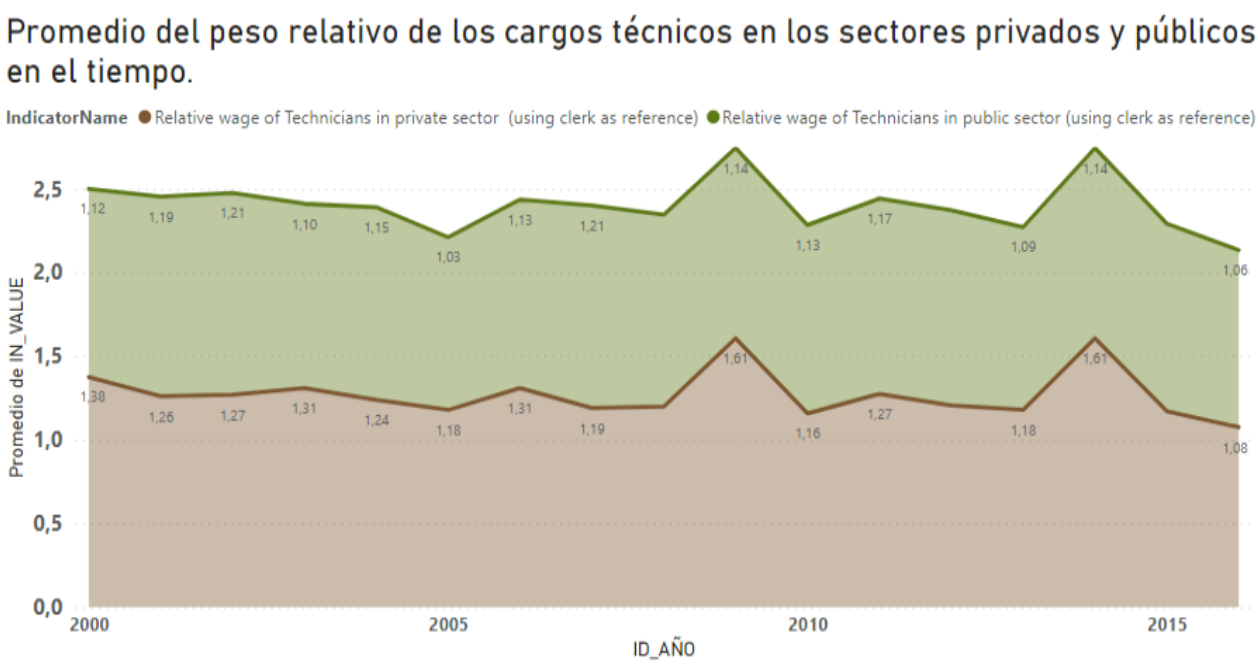



**Gráfico 3:** Evolución del peso relativo de los cargos técnicos en el sector público y privado en el tiempo

Como observamos en el gráfico, en el eje X he usado la columna de ID_AÑO de la tabla de hechos y, de la misma tabla, el promedio de la columna de IN_VALUE para el eje Y.

Para que muestre los valores de las dos métricas que nos interesan he filtrado el gráfico con la columna de IndicatorName de la tabla de dimensión métricas seleccionando las filas de “Relative wage of technicians in prívate sector (using clerk as reference)” y “Relative wage of technicians in public sector (using clerk as reference)”. Para la leyenda he usado la misma columna que el en filtro, siendo la primera de las métricas la representada en color marrón y la segunda, en color verde.

Esta representación muestra el volumen de cada una de las métricas, así como el volumen total de ambas. La media del volumen total de la métrica del salario relativo de los técnicos en el sector público tomando como base el salario de un oficinista (área verde) se ve cuando restamos ambas áreas. Es decir, para el año 2000 habría que restar 2,5 (veces que ambos salarios, público y privado, superan al de un oficinista (se toma valor 1 para el salario del oficinista)) y 1,38 para obtener el dato de 1,12. Para facilitar la lectura de los datos he decidido etiquetar los valores para cada año.

*6.2.4. Obtener el promedio del peso por región del gasto en empleados públicos respecto al GDP y el gasto público.*

En este caso he usado el gráfico de columnas agrupadas el cual nos permite analizar dos variables cuantitativas diferentes con lo que se puede ver la relación de las dos métricas respecto a las distintas categorías seleccionadas.

**Gráfico 4: PROMEDIO DEL PESO POR REGIÓN DEL GASTO EN EMPLEADOS PÚBLICOS RESPECTO AL GDP Y EL GASTO PÚBLICO**

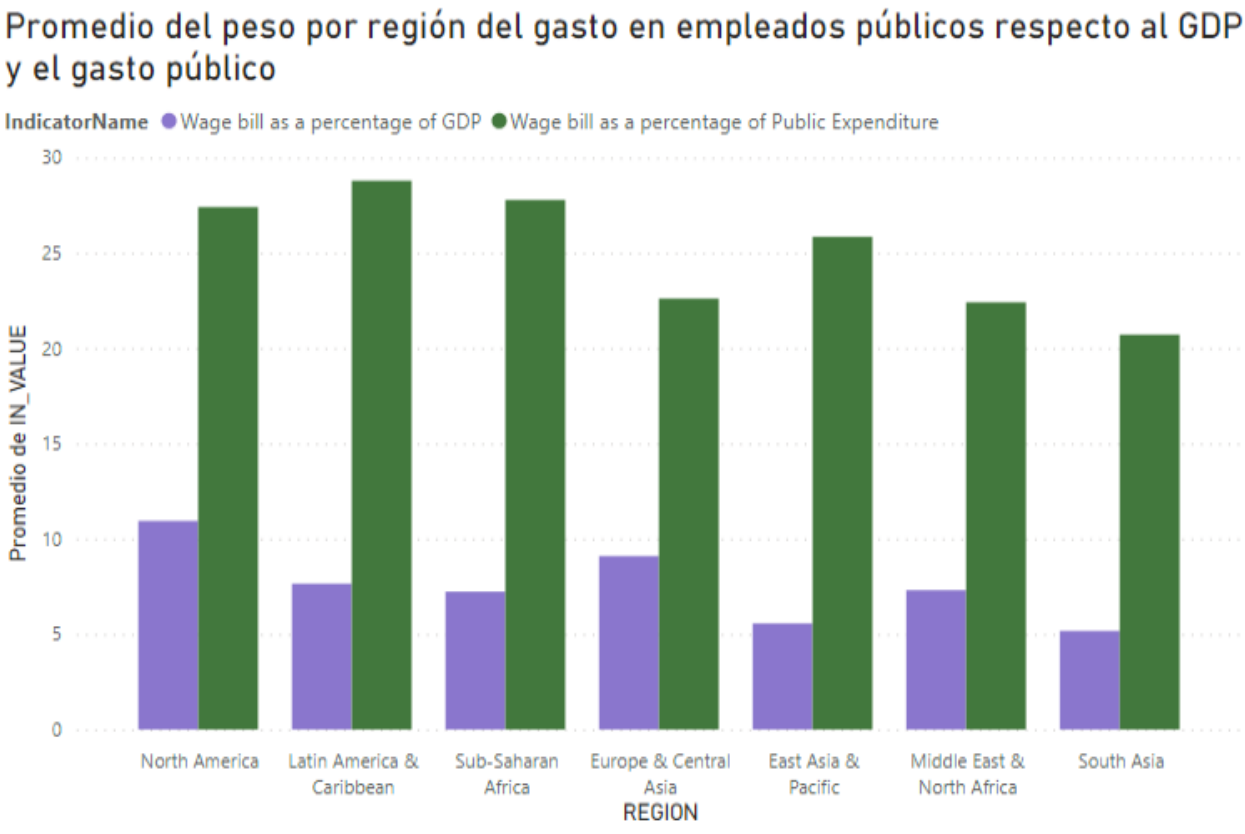



**Gráfico 4:** Promedio del peso por región del gasto en empleados públicos respecto al GDP y el gasto público

Este gráfico representa, en el eje X, las distintas regiones y, en el eje Y, el promedio de los valores de las dos métricas. Para el eje de ordenadas he usado la tabla de dimensión país,

en concreto, la columna de región y, para el eje de abscisas, el promedio de la columna IN_VALUE de la tabla de hechos.

Para filtrar los datos he usado la columna de IndicatorName seleccionando los campos “Wage bill as a percentage of GDP” y “Wage bill as a percentage of Public Expenditure”. Estos mismos son los usado en la leyenda del gráfico, siendo el primer indicador el de color morado y el segundo, el de color verde.