

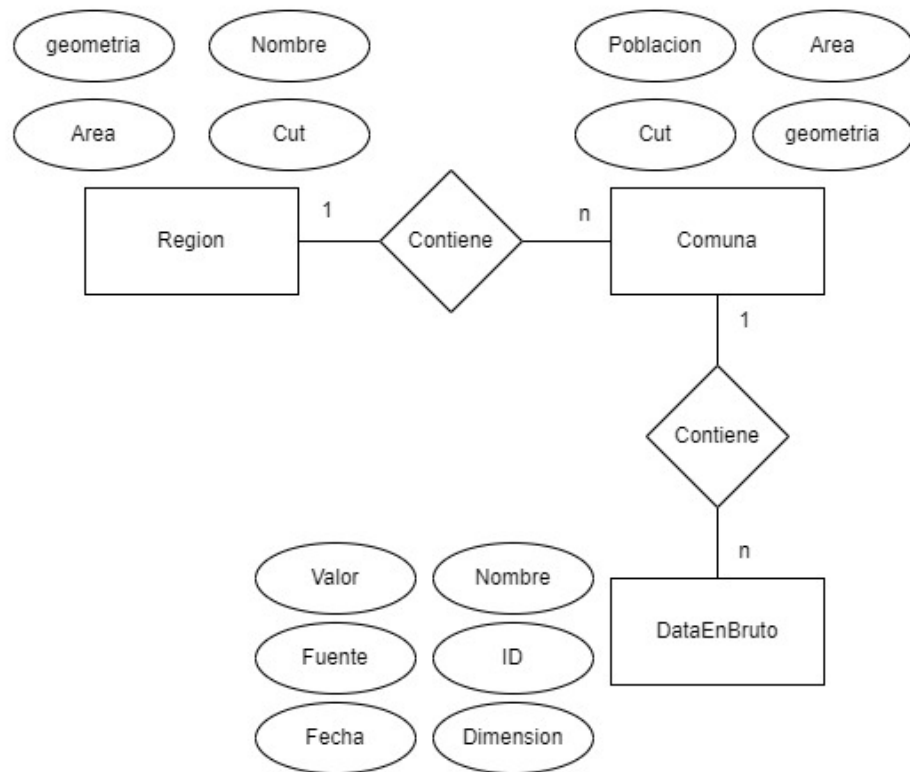


# Arquitectura

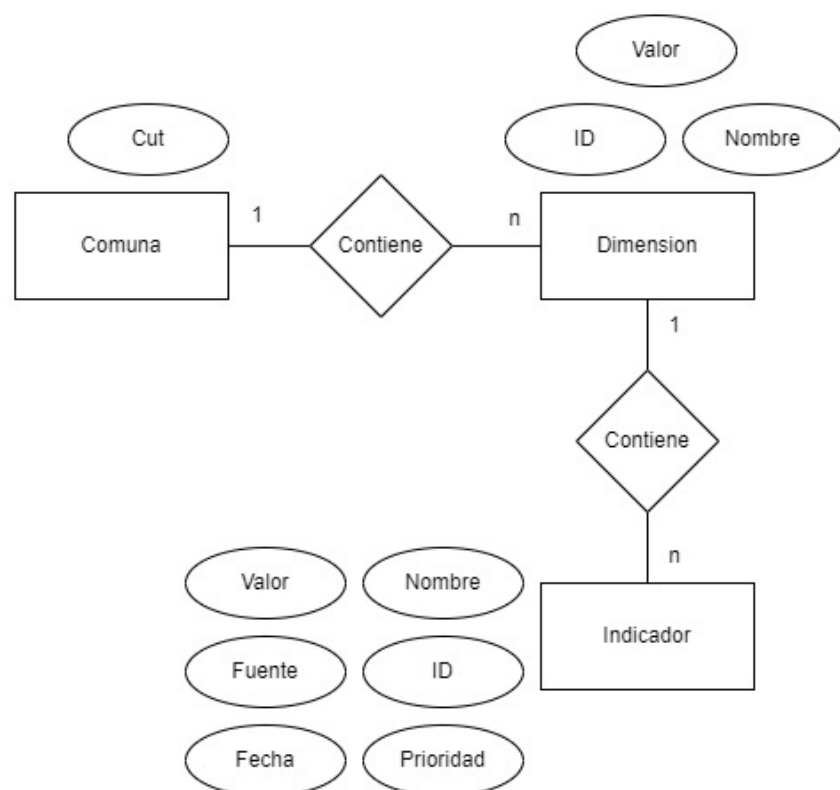
## Modelo Entidad-Relación

▼ Bases de datos 09/10

## DB Transaccional

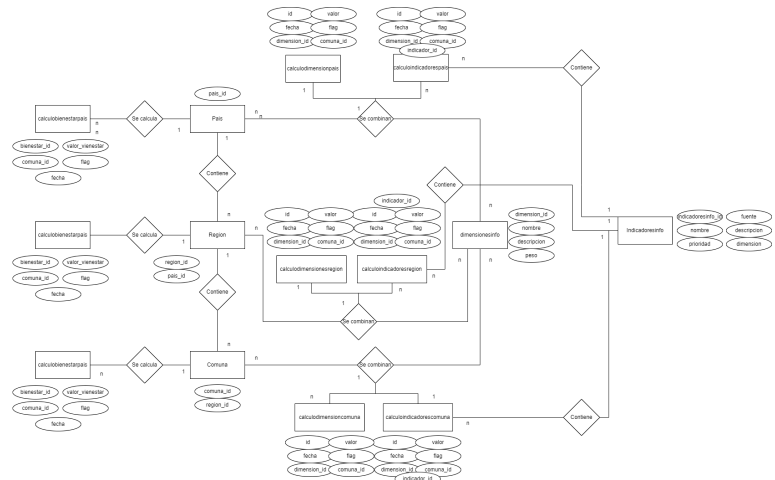
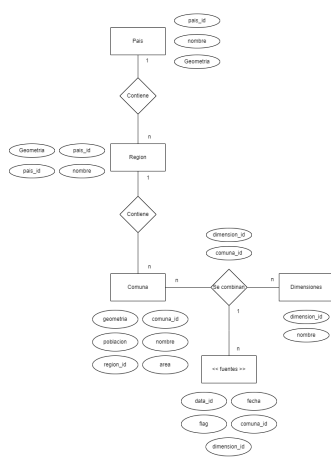


## DB Procesada



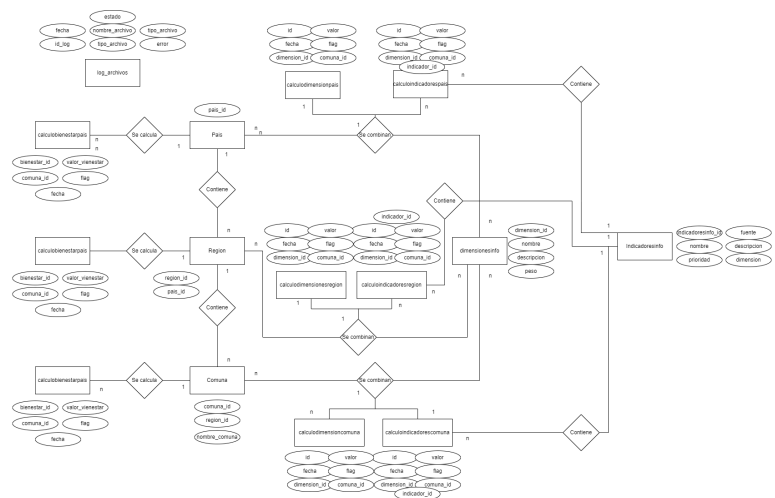
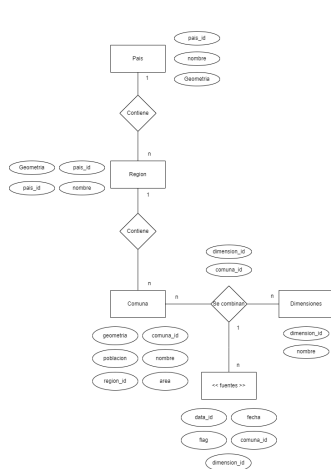
## ▼ Bases de datos 04/11

### Base de datos transaccional / Base de datos procesados



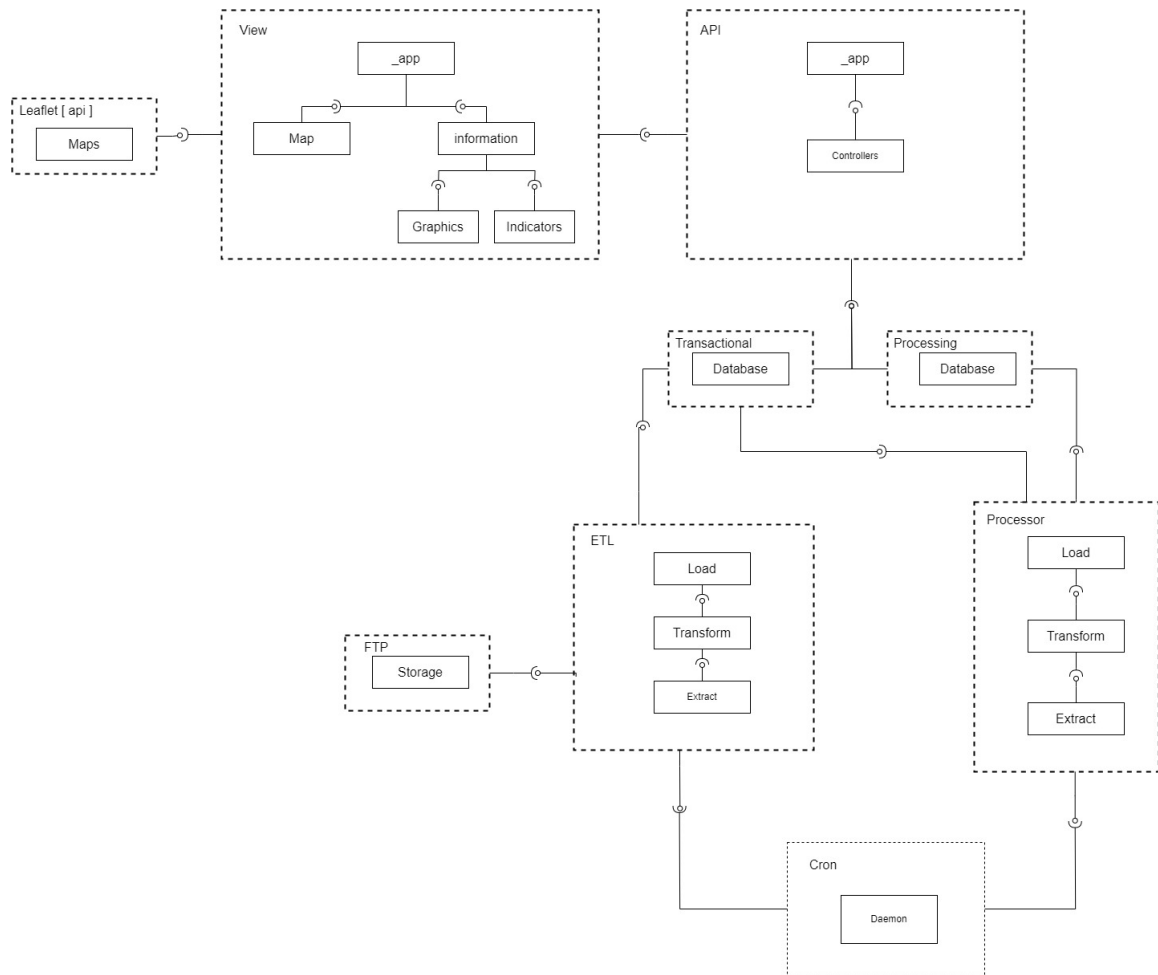
## ▼ Bases de datos 13/11

### Base de datos transaccional / Base de datos procesados



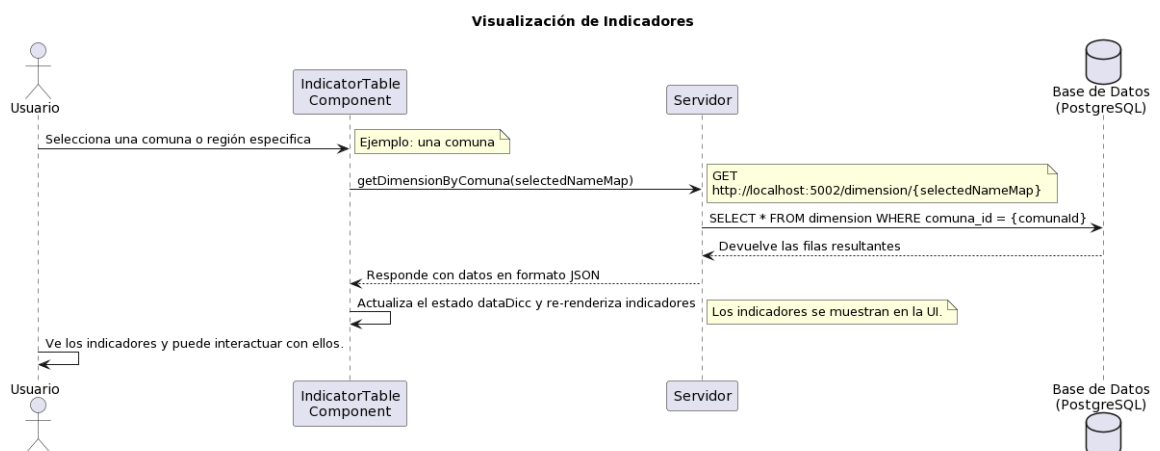
## Diagrama de componentes

### ▼ Diagrama de componentes 09/10

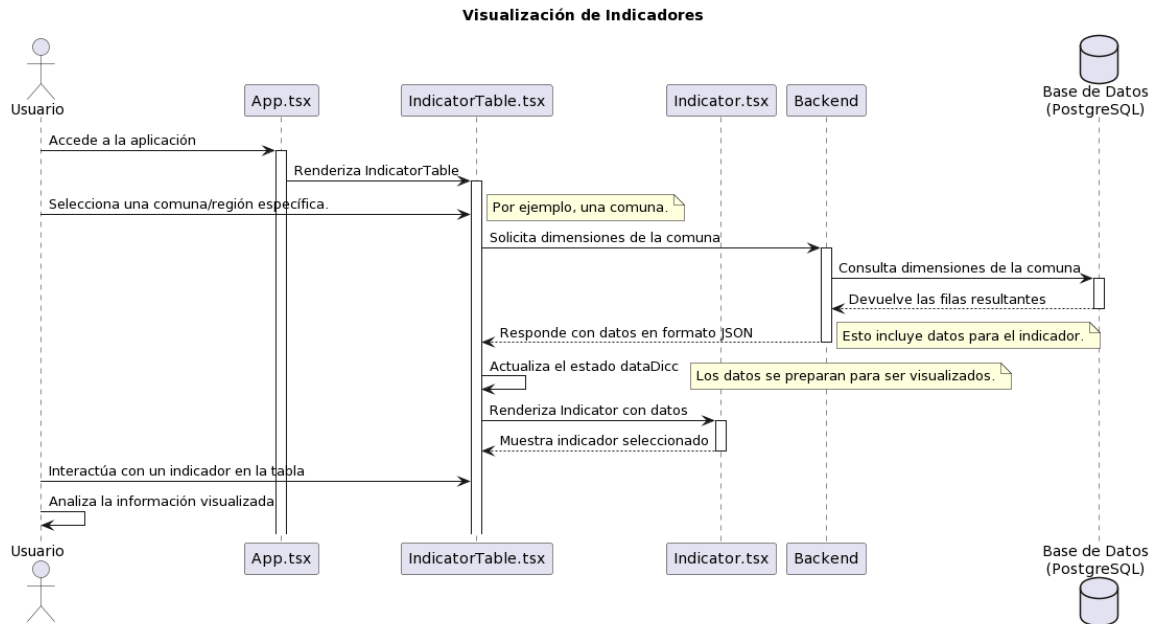


## Diagramas de secuencia

### ▼ Visualización de indicadores



### ▼ Visualización de indicadores (actualizado) pd: Backend se encuentra simplificado



@startuml

title Visualización de Indicadores

actor Usuario as U

participant "App.tsx" as App

participant "IndicatorTable.tsx" as ITC

participant "Indicator.tsx" as I

participant Backend as B

database "Base de Datos\n(PostgreSQL)" as DB

U -> App: Accede a la aplicación  
activate App

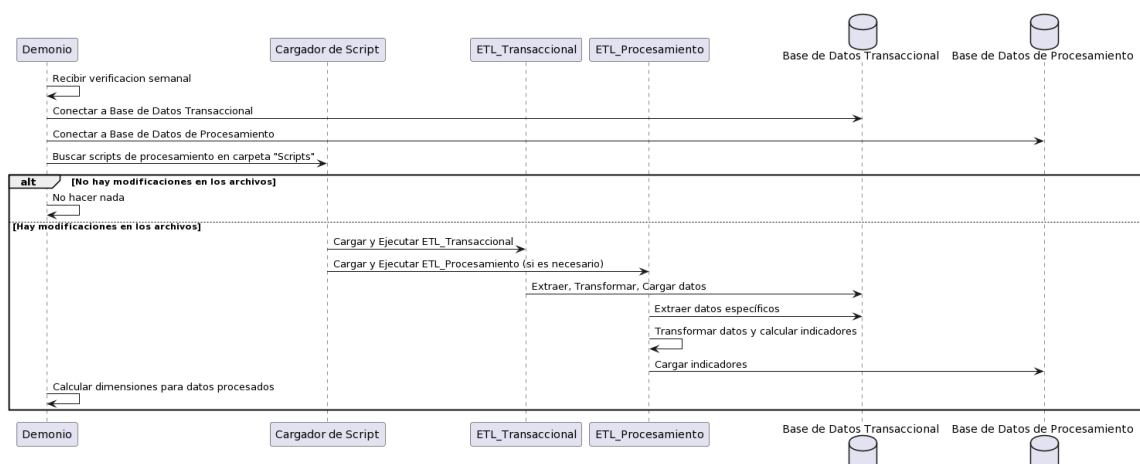
App -> ITC: Renderiza IndicatorTable  
activate ITC

U -> ITC: Selecciona una comuna/región específica.  
note right: Por ejemplo, una comuna.

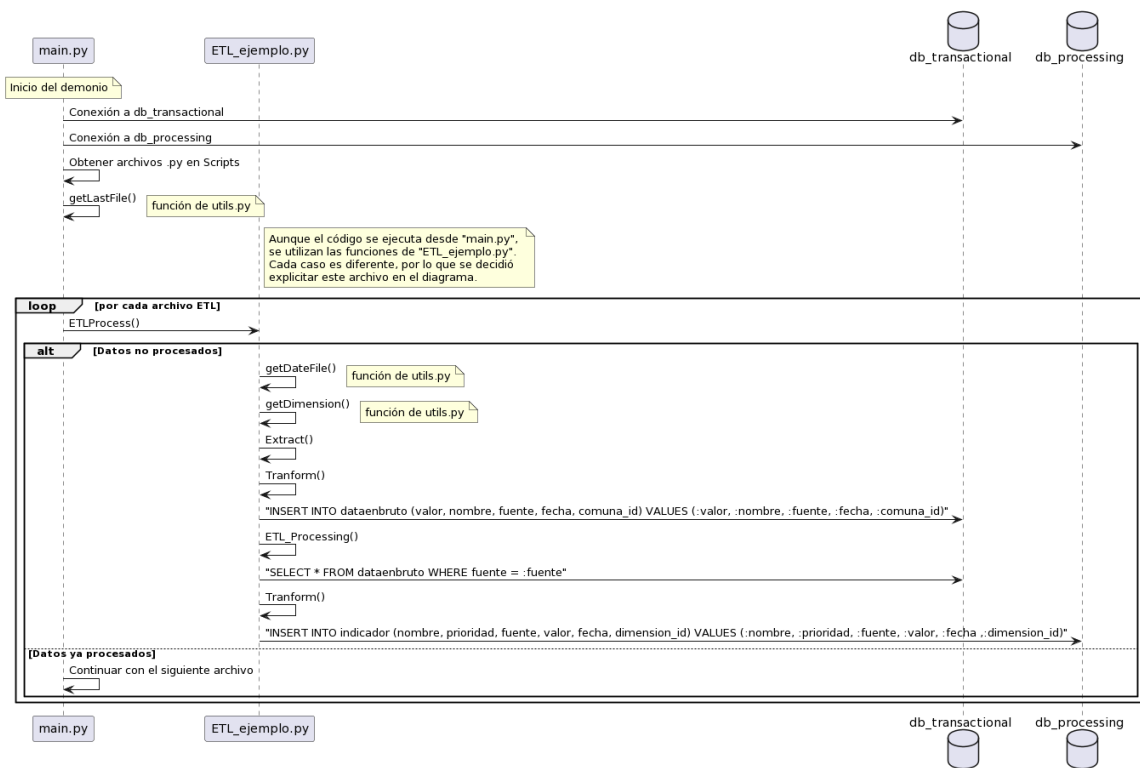
ITC -> B: Solicita dimensiones de la comuna  
activate B

**B -> DB:** Consulta dimensiones de la comuna  
 activate **DB**  
**DB --> B:** Devuelve las filas resultantes  
 deactivate **DB**  
  
**B --> ITC:** Responde con datos en formato **JSON**  
 note right: Esto incluye datos para el indicador.  
 deactivate **B**  
  
**ITC -> ITC:** Actualiza el estado dataDicc  
 note right: Los datos se preparan para ser visualizados.  
  
**ITC -> I:** Renderiza Indicator con datos  
 activate **I**  
**I --> ITC:** Muestra indicador seleccionado  
 deactivate **I**  
  
**U -> ITC:** Interactúa con un indicador en la tabla  
  
**U -> U:** Analiza la información visualizada  
  
 @endum1

## ▼ Demonio



## ▼ Demonio (actualizado)



@startuml

participant "[main.py](http://main.py/)" as Main

participant "ETL\_ejemplo.py" as ETLEjemplo

database "db\_transactional" as DBTrans

database "db\_processing" as DBProc

note over Main: Inicio del demonio

Main -> DBTrans: Conexión a db\_transactional

Main -> DBProc: Conexión a db\_processing

Main -> Main: Obtener archivos .py en Scripts

Main -> Main: getLastFile()

note right: función de [utils.py](http://utils.py/)

note right ETLEjemplo

Aunque el código se ejecuta desde "[main.py](http://main.py/)",

se utilizan las funciones de "ETL\_ejemplo.py".

Cada caso es diferente, por lo que se decidió

explicitar este archivo en el diagrama.  
end note

loop por cada archivo ETL

Main -> ETL Ejemplo: ETLProcess()

alt Datos no procesados

ETL Ejemplo -> ETL Ejemplo: getDateFile()

note right: función de [utils.py](http://utils.py/)

ETL Ejemplo -> ETL Ejemplo: getDimension()

note right: función de [utils.py](http://utils.py/)

ETL Ejemplo -> ETL Ejemplo: Extract()

ETL Ejemplo -> ETL Ejemplo: Transform()

ETL Ejemplo -> DBTrans: "INSERT INTO dataenbruto (valor, nombre, fuente, fecha, comuna\_id) VALUES (:valor, :nombre, :fuente, :fecha, :comuna\_id)"

ETL Ejemplo -> ETL Ejemplo: ETL\_Processing()

ETL Ejemplo -> DBTrans: "SELECT \* FROM dataenbruto WHERE fuente = :fuente"

ETL Ejemplo -> ETL Ejemplo: Transform()

ETL Ejemplo -> DBProc: "INSERT INTO indicador (nombre, prioridad, fuente, valor, fecha, dimension\_id) VALUES (:nombre, :prioridad, :fuente, :valor, :fecha, :dimension\_id)"

else Datos ya procesados

Main -> Main: Continuar con el siguiente archivo

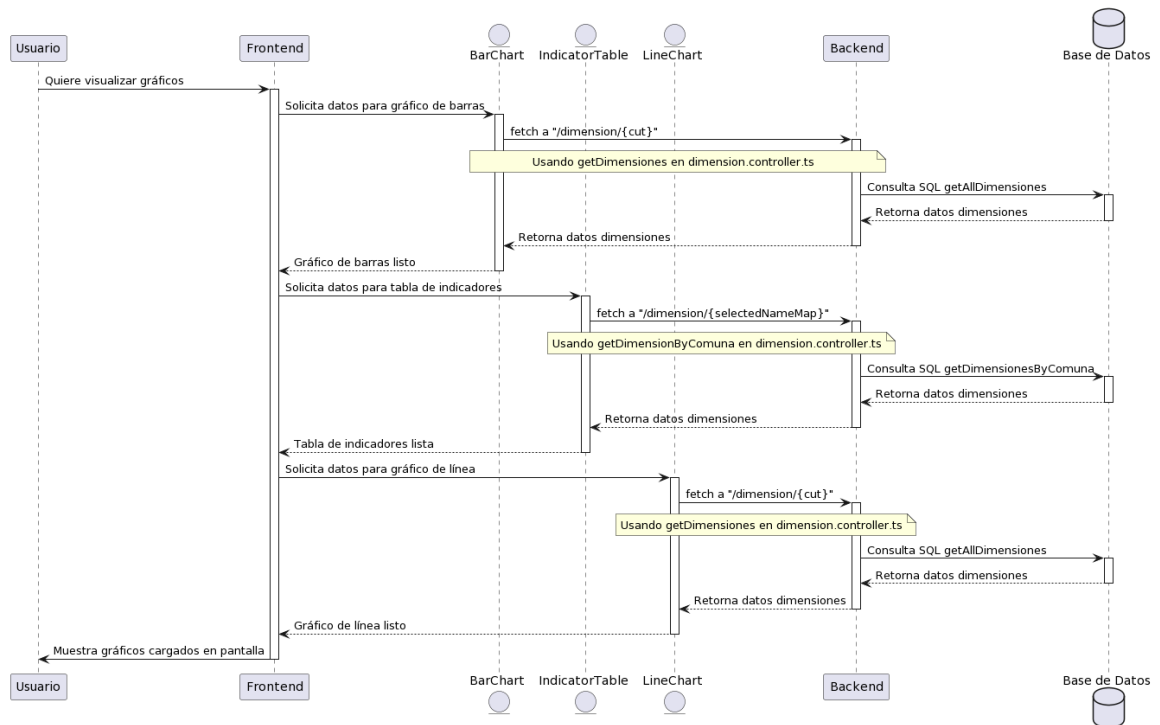
end

end

@endum1

## ▼ Visualización de gráficos.





@startuml

title "Visualizar indicadores de bienestar"

actor Usuario

boundary Frontend

control "BarChart"

control "LineChart"

control "IndicatorTable"

boundary Backend

database "Base de Datos"

Usuario -> Frontend : Solicita visualizar indicadores

Frontend -> "BarChart" : Iniciar

"BarChart" -> "Backend" : Solicitar indicador

"LineChart" -> Backend : `getIndicators()`

Backend -> "Base de Datos" : Buscar indicadores

"Base de Datos" -> Backend : Retorna indicadores

Backend -> "BarChart" : Retorna indicadores

"LineChart" -> "BarChart" : Mostrar indicador

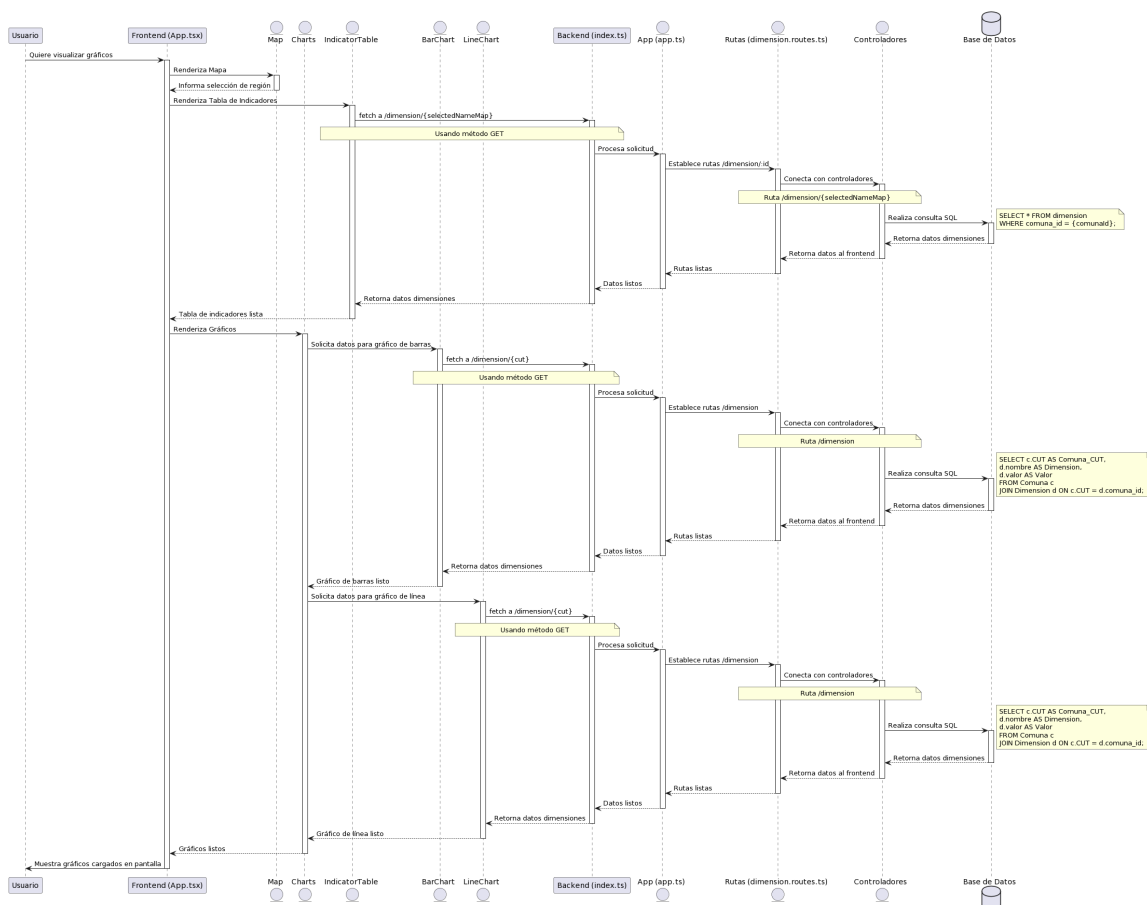
"BarChart" -> "IndicatorTable" : Solicitar tabla de indica

"IndicatorTable" -> Backend : `getIndicatorById()`

note right: `http://localhost:5002/dimension/${selectedName}`  
 Backend -> "Base de Datos" : Buscar indicador por ID  
 "Base de Datos" -> Backend : Retorna indicador específico  
 Backend -> "IndicatorTable" : Retorna indicador específico  
 "IndicatorTable" -> "BarChart" : Mostrar tabla de indicado  
 "BarChart" -> Frontend : Mostrar gráficos y tabla  
 Frontend -> Usuario : Muestra indicadores de bienestar

@endum1

## ▼ Visualización de gráficos con full detalles de flujo (corrección del anterior)



@startuml

participant "Usuario" as User

participant "Frontend (App.tsx)" as FrontApp

entity "Map" as MapComp

entity "Charts" as ChartsComp

entity "IndicatorTable" as IT

```

entity "BarChart" as BC
entity "LineChart" as LC
participant "Backend (index.ts)" as BackIndex
entity "App (app.ts)" as AppClass
entity "Rutas (dimension.routes.ts)" as Routes
entity "Controladores" as Controllers
database "Base de Datos" as DB

User -> FrontApp : Quiere visualizar gráficos
activate FrontApp

FrontApp -> MapComp : Renderiza Mapa
activate MapComp
MapComp --> FrontApp : Informa selección de región
deactivate MapComp

FrontApp -> IT : Renderiza Tabla de Indicadores
activate IT
IT -> BackIndex : fetch a /dimension/{selectedNameMap}
note over IT, BackIndex : Usando método GET
activate BackIndex
BackIndex -> AppClass : Procesa solicitud
activate AppClass
AppClass -> Routes : Establece rutas /dimension/:id
activate Routes
Routes -> Controllers : Conecta con controladores
note over Routes, Controllers : Ruta /dimension/{selectedNameMap}
activate Controllers
Controllers -> DB : Realiza consulta SQL
note right : SELECT * FROM dimension \nWHERE comuna_id = {selectedNameMap}
activate DB
DB --> Controllers : Retorna datos dimensiones
deactivate DB
Controllers --> Routes : Retorna datos al frontend
deactivate Controllers
Routes --> AppClass : Rutas listas
deactivate Routes
AppClass --> BackIndex : Datos listos

```

```
deactivate AppClass
BackIndex --> IT : Retorna datos dimensiones
deactivate BackIndex
IT --> FrontApp : Tabla de indicadores lista
deactivate IT

FrontApp -> ChartsComp : Renderiza Gráficos
activate ChartsComp

ChartsComp -> BC : Solicita datos para gráfico de barras
activate BC
BC -> BackIndex : fetch a /dimension/{cut}
note over BC, BackIndex : Usando método GET
activate BackIndex
BackIndex -> AppClass : Procesa solicitud
activate AppClass
AppClass -> Routes : Establece rutas /dimension
activate Routes
Routes -> Controllers : Conecta con controladores
note over Routes, Controllers : Ruta /dimension
activate Controllers
Controllers -> DB : Realiza consulta SQL
note right : SELECT c.CUT AS Comuna_CUT, \nd.nombre AS Dim
activate DB
DB --> Controllers : Retorna datos dimensiones
deactivate DB
Controllers --> Routes : Retorna datos al frontend
deactivate Controllers
Routes --> AppClass : Rutas listas
deactivate Routes
AppClass --> BackIndex : Datos listos
deactivate AppClass
BackIndex --> BC : Retorna datos dimensiones
deactivate BackIndex
BC --> ChartsComp : Gráfico de barras listo
deactivate BC

ChartsComp -> LC : Solicita datos para gráfico de línea
```

```

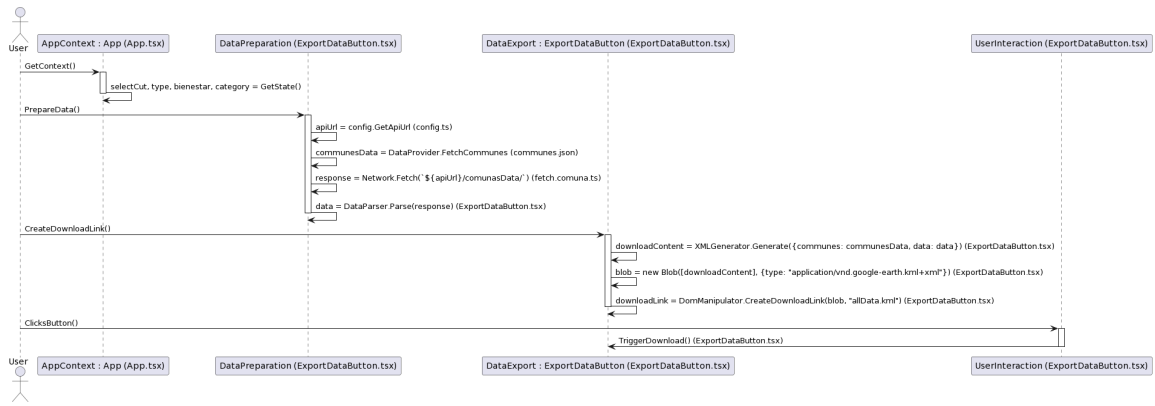
activate LC
LC -> BackIndex : fetch a /dimension/{cut}
note over LC, BackIndex : Usando método GET
activate BackIndex
BackIndex -> AppClass : Procesa solicitud
activate AppClass
AppClass -> Routes : Establece rutas /dimension
activate Routes
Routes -> Controllers : Conecta con controladores
note over Routes, Controllers : Ruta /dimension
activate Controllers
Controllers -> DB : Realiza consulta SQL
note right : SELECT c.CUT AS Comuna_CUT, \nd.nombre AS Dim
activate DB
DB --> Controllers : Retorna datos dimensiones
deactivate DB
Controllers --> Routes : Retorna datos al frontend
deactivate Controllers
Routes --> AppClass : Rutas listas
deactivate Routes
AppClass --> BackIndex : Datos listos
deactivate AppClass
BackIndex --> LC : Retorna datos dimensiones
deactivate BackIndex
LC --> ChartsComp : Gráfico de línea listo
deactivate LC

ChartsComp --> FrontApp : Gráficos listos
deactivate ChartsComp

FrontApp -> User : Muestra gráficos cargados en pantalla
deactivate FrontApp
@enduml

```

▼ Boton de exportar



## Diagrama de Despliegue

### ▼ Diagrama de despliegue 06/11

