

## Práctica III

Grupo Miércoles A

---



28/02/2024

PABLO MORENO MUÑOZ 841972@unizar.es

ANDREI DUMBRAVA LUCA 844417@unizar.es

GUILLERMO BAJO LABORDA, 842748@unizar.es

# Índice

<b>Introducción</b>	<b>4</b>
<b>Parte 1</b>	<b>5</b>
<b>Esquema conceptual y lógico de la base de datos relacional diseñada en la práctica anterior:</b>	<b>5</b>
Determinación del esquema lógico y conceptual de la base de datos Oracle a integrar	8
Mejoras sugeridas para la base de datos a integrar	11
Definición e implementación del esquema global en Oracle	12
<b>Parte 2</b>	<b>13</b>
<b>Enunciado de un problema de diseño de bases de datos</b>	<b>13</b>
Esquema conceptual para el problema enunciado	14
Esquema lógico 1 para el problema enunciado.	15
Esquema lógico 2 para el problema enunciado	16
Esquema global y su implementación en PostgreSQL,	17
<b>Actualizaciones de datos sobre el esquema global:</b>	<b>18</b>
<b>Esfuerzos invertidos:</b>	<b>19</b>

# Introducción

Esta práctica aborda el desafiante problema de integración de bases de datos para definir una base de datos federada. En este contexto, nos enfrentamos a la tarea de conectar y analizar una base de datos Oracle proporcionada, así como integrar su esquema con el desarrollado previamente en la práctica anterior. Además, exploramos la integración de dos bases de datos en PostgreSQL. Este proceso implica el uso de herramientas como DATABASE LINK de Oracle y dblink de PostgreSQL, junto con la consideración de otras opciones posibles.

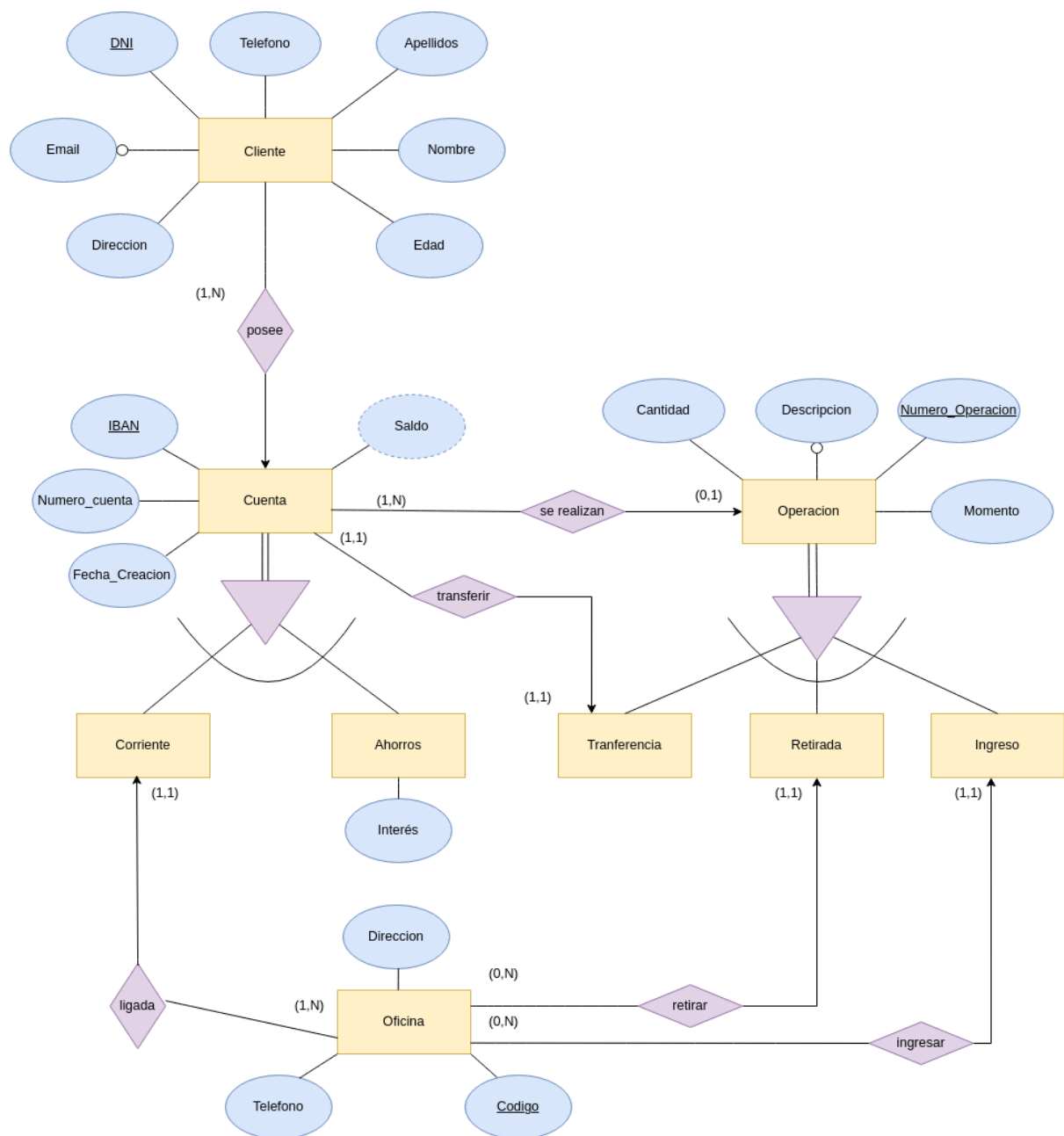
El contexto del problema se centra en la integración de un banco previamente diseñado en prácticas anteriores en otro banco más grande con su propia base de datos. El objetivo es diseñar e implementar un esquema global que permita visualizar ambas bases de datos de manera uniforme, manteniendo su autonomía. Se espera que enfrentemos desafíos como la inferencia del esquema conceptual de la base de datos proporcionada, la integración sin modificar los diseños locales y la preparación para futuras mejoras del sistema de información del banco integrado.

Además, en la segunda parte de la memoria, abordamos la integración de dos esquemas de bases de datos implementadas en PostgreSQL, lo que añade complejidad al proceso. Planteamos un problema original de diseño de bases de datos, realizamos el diseño conceptual y lógico, e implementamos ambos esquemas, considerando posibles dificultades de integración.

## Parte 1

### Esquema conceptual y lógico de la base de datos relacional diseñada en la práctica anterior:

El diseño conceptual de una base de datos es un proceso crucial donde se establecen las entidades clave y sus relaciones de manera abstracta. En este contexto, el modelo Entidad-Relación (E/R) emerge como una herramienta fundamental para representar la estructura conceptual de la base de datos. A continuación, se muestra el diseño conceptual de la base de datos diseñada en la práctica anterior:

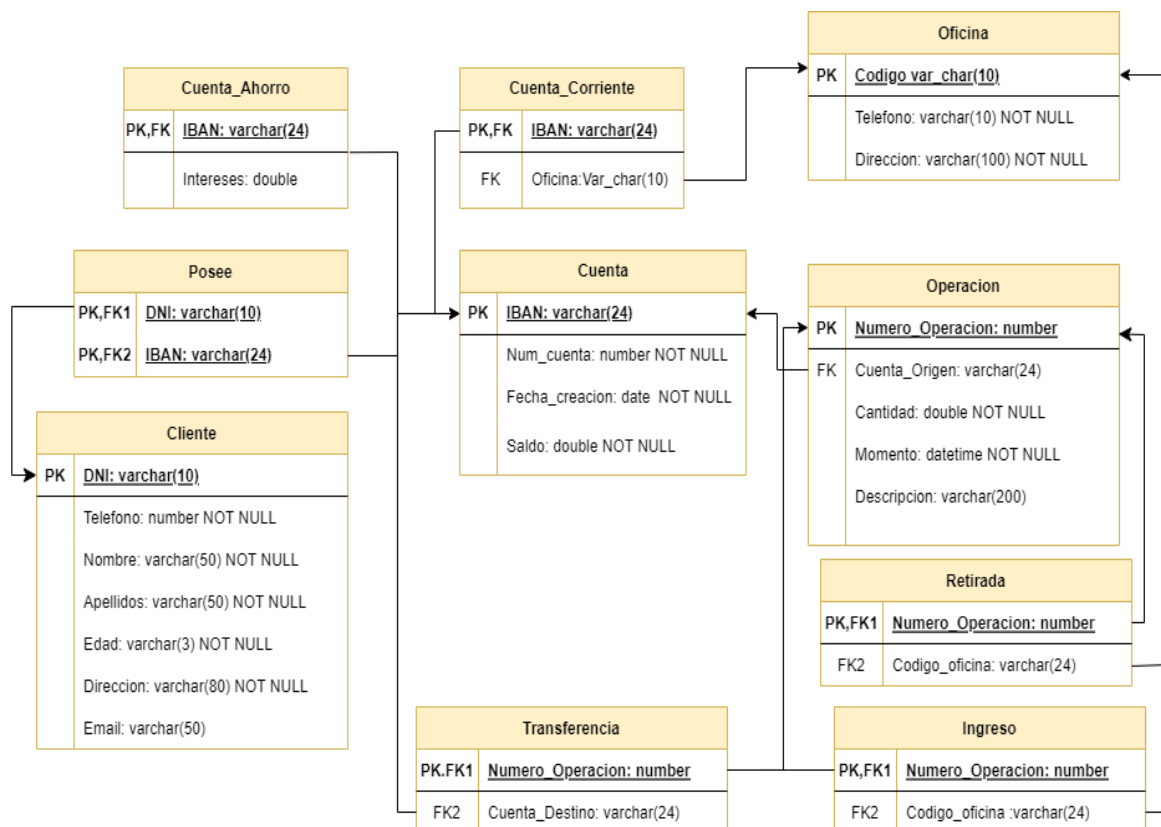


Como se puede apreciar en el esquema, aparecen un total de 9 entidades, cada una con sus respectivos atributos y relacionadas entre sí. Los atributos subrayados son únicos y representan la clave primaria de cada entidad. Aquellos ligados con un círculo son atributos opcionales. En este diseño aparecen dos especializaciones, por una parte, distinguimos dos tipos de Cuentas (Corrientes y de Ahorros). Apreciamos que la especialización es disjunta, porque una cuenta no puede pertenecer a ambas especializaciones a la vez, y de cobertura total, puesto que toda cuenta debe ser o bien de ahorros, o bien corriente. La entidad Operación también se trata de una especialización disjunta de cobertura total, en la que disponemos de las especializaciones Transferencia, Retirada e Ingreso.

Respecto a las decisiones de diseño, ha habido ciertos aspectos que han causado duda a la hora de realizar este primer diseño inicial. Por una parte, se consideró que no era necesario relacionar la entidad *Oficina* con *Retirada* e *Ingreso*, pero finalmente se hizo porque la oficina asociada a la cuenta no tiene por qué ser la sucursal desde la cual se retira/ingresa el dinero, sino que puede ser otra distinta. Por otra parte, respecto al atributo *Saldo* de la entidad *Cuenta* se ha consensuado que debe ser un atributo derivado, puesto que puede ser calculado a partir de las diferentes entradas de las tablas *Operación*, *Transferencia*, *Retirada*...

Otro aspecto que dio lugar a ciertas dudas inicialmente fue la opción de añadir a la entidad Transferencia el atributo *Cuenta\_Origen*, lo cual podría simplificar las consultas. Sin embargo, esto solo evitaría la realización de un join, y dado que significaría la existencia de redundancia en la base de datos, esta opción finalmente se descartó.

El diseño lógico de una base de datos es la etapa en la que se traduce el modelo conceptual, basado en el modelo Entidad-Relación (E/R), a un modelo de datos específico para el sistema de gestión de bases de datos (SGBD) que se utilizará. En esta fase, se definen las tablas, los atributos, las claves primarias y foráneas, así como las restricciones de integridad necesarias para representar con precisión la estructura conceptual en un esquema que pueda ser implementado en un SGBD. Este proceso de diseño lógico es fundamental para garantizar la eficiencia, la integridad y la consistencia de los datos dentro del sistema de base de datos. A continuación, se muestra el diseño lógico de la base de datos, de la práctica anterior, así como diversas decisiones de diseño que fueron tomadas:



Como se puede observar en el diseño, se optó por el mismo método a la hora de pasar a relacional ambas especializaciones. Se creó la entidad padre con un identificador único y los atributos correspondiente, y por otra parte, las entidades hijas con ese identificador como clave foránea (que a su vez es clave primaria de la entidad hija), y atributos específicos de cada entidad hija, como bien puede ser la cuenta destino en la entidad *Transferencia* o el código de oficina en la entidad *Ingreso*.

## Determinación del esquema lógico y conceptual de la base de datos Oracle a integrar

Para determinar el esquema lógico y conceptual de la base de datos Oracle a integrar, se llevaron a cabo una serie de operaciones específicas que permitieron obtener una comprensión detallada de su estructura y diseño. Estas operaciones se realizaron mediante consultas SQL ejecutadas en el entorno proporcionado. A continuación, se detallan los pasos seguidos:

En primer lugar, se identificó el usuario de la base de datos mediante la ejecución de la siguiente consulta SQL, esencial para acceder a la información relevante:

```
Unset  
SELECT OWNER, TABLE_NAME FROM all_tables@SCHEMA2BD2 WHERE table_name='CUENTA' ;
```

Posteriormente, se procedió a investigar las tablas asociadas al usuario en cuestión. Esto implicó recuperar una lista de todas las tablas pertenecientes al usuario relevante en la base de datos Oracle, utilizando la siguiente consulta SQL:

```
Unset  
SELECT TABLE_NAME FROM all_tables@SCHEMA2BD2 WHERE OWNER='TEST' ;
```

A continuación, se realizó una descripción detallada de las tablas seleccionadas. Mediante el uso del comando DESCRIBE en SQL, se obtuvo información crucial sobre la estructura de cada tabla, incluidos los nombres de las columnas y los tipos de datos utilizados. A continuación se muestra un ejemplo de esta consulta para la tabla "DIRECCION":

```
Unset  
DESCRIBE DIRECCION;
```

Finalmente, se llevaron a cabo consultas adicionales para identificar las claves primarias y extranjeras en las tablas de interés. Esto se logró mediante consultas SQL que filtraban las restricciones de integridad de la base de datos, permitiendo así una comprensión más profunda de las relaciones entre las entidades. Aquí se muestra un ejemplo de estas consultas para identificar las claves primarias y extranjeras:

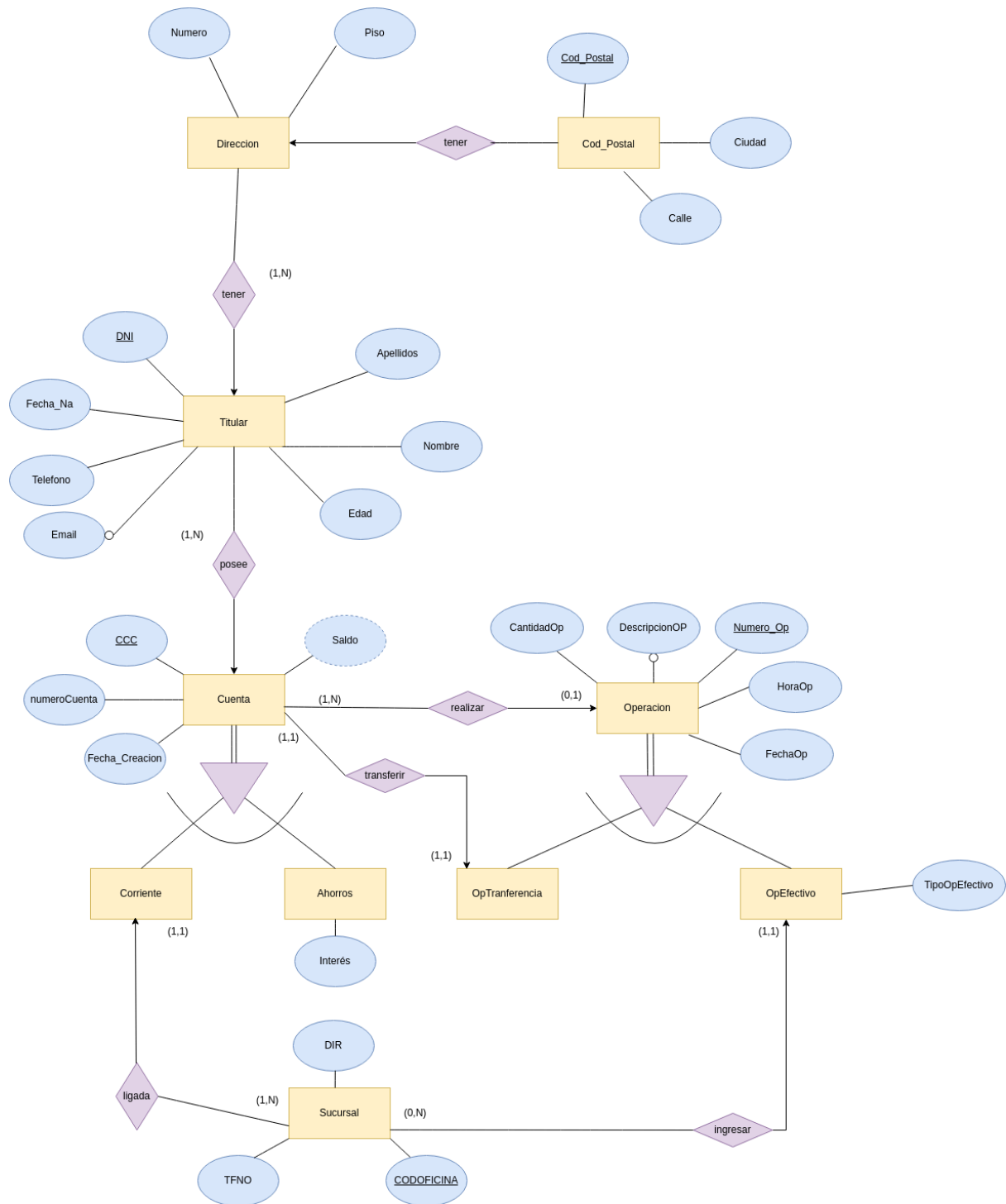
```
Unset
# Claves primarias
SELECT acc.column_name, acc.constraint_name
FROM all_cons_columns@schema2bd2 acc
JOIN    all_constraints@schema2bd2    ac    on    ac.constraint_name    =
acc.constraint_name
WHERE ac.constraint_type = 'P' and ac.owner= 'TEST';
```

```
Unset
# Claves extranjeras
SELECT acc.column_name, acc.constraint_name, ac.r_constraint_name
FROM all_cons_columns@schema2bd2 acc
JOIN    all_constraints@schema2bd2    ac    on    ac.constraint_name    =
acc.constraint_name
WHERE ac.constraint_type = 'R' and ac.owner= 'TEST';
```

El resultado de estos pasos proporcionó una visión completa del esquema lógico y conceptual de la base de datos Oracle, incluidas las relaciones entre las entidades y las restricciones de integridad asociadas. Este conocimiento es fundamental para el proceso de integración de bases de datos y garantiza una integración efectiva y coherente entre sistemas heterogéneos.



Gracias a la investigación detallada de la estructura de la base de datos Oracle mediante consultas SQL, hemos representado un nuevo esquema entidad-relación que garantiza la coherencia del proceso y tiene el siguiente aspecto:



## Mejoras sugeridas para la base de datos a integrar

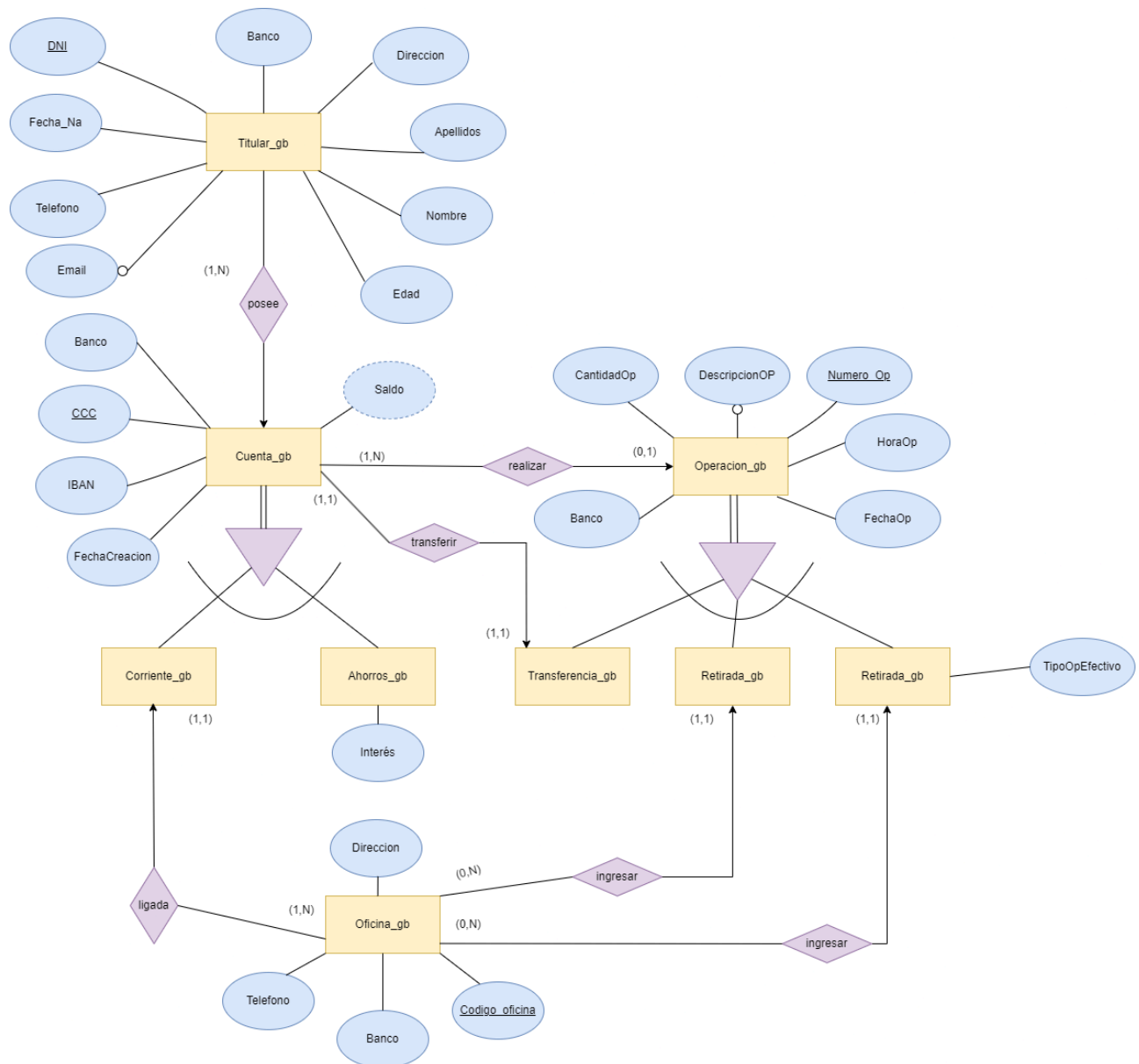
Después de un análisis de las tablas y sus relaciones en la base de datos proporcionada, se han identificado áreas donde se podrían realizar mejoras significativas para optimizar su estructura y rendimiento. A continuación, se presentan algunas sugerencias de mejoras que podrían considerarse para aumentar la eficiencia y la coherencia de la base de datos:

- Eliminación de la tabla CODENTIDADES: Tras revisar las relaciones entre las entidades, se observó que la tabla CODENTIDADES parece no estar vinculada con ninguna otra tabla en la base de datos. Por lo tanto, su eliminación podría simplificar la estructura general y reducir la complejidad innecesaria.
- Rediseño de las tablas de Cuenta: Se detectó que ambas tablas comparten la misma clave primaria que la tabla Cuenta. Esta duplicación de claves primarias podría provocar dificultades al intentar acceder a la información relacionada con las cuentas. Una alternativa más eficiente sería asignar claves primarias únicas a cada una de estas tablas y establecer relaciones mediante claves foráneas con la tabla Cuenta, mejorando así la gestión y accesibilidad de los datos.
- Normalización adicional: Aunque la base de datos parece estar normalizada en gran medida, podrían considerarse algunas medidas para una mayor normalización que puedan reducir la redundancia de datos y mejorar la eficiencia del sistema. Por ejemplo, actualmente, en la tabla "DIRECCION" se almacena información como la calle, el número, el piso y la ciudad. Sin embargo, si en el futuro se prevé la posibilidad de que un titular de cuenta tenga múltiples direcciones registradas (por ejemplo, dirección de residencia, dirección de trabajo, dirección de envío, etc.), sería beneficioso normalizar esta estructura. Se podría crear una nueva tabla llamada "DIRECCIONES", donde cada registro represente una dirección única asociada a un titular de cuenta, incluyendo los campos de calle, número, piso y ciudad, junto con una clave foránea que haga referencia al titular correspondiente. Esta normalización permitiría una gestión más flexible y eficiente de las direcciones de los titulares, así como la posibilidad de añadir nuevas direcciones en el futuro sin necesidad de modificar la estructura de la base de datos principal. Como este, hay diversos aspectos que podrían considerarse para llevar a cabo una normalización adicional según las necesidades en caso de que el problema se fuera expandiendo. Por ahora, por simplicidad no realizaremos ninguna normalización de más.
- Implementación de índices adecuados: Se podría considerar también la necesidad de índices adicionales en las tablas para mejorar el rendimiento de las consultas más frecuentes y

optimizar la velocidad de acceso a los datos. Un ejemplo de índice que podría mejorar el rendimiento de las consultas sería un índice en la columna "FECHACRE" de la tabla "CUENTA". Supongamos que fuera común realizar consultas que seleccionan cuentas basadas en su fecha de creación. Al crear un índice en esta columna, las consultas que filtran por fecha de creación se ejecutarán de manera más eficiente, ya que el motor de base de datos puede utilizar el índice para buscar rápidamente las filas que cumplen con el criterio de fecha. Esto resultaría en tiempos de respuesta más cortos para estas consultas y una mejora general en el rendimiento del sistema.

## Definición e implementación del esquema global en Oracle

Se ha creado un esquema E/R global con el que poder apoyarse para realizar la implementación, además de que sirve de ayuda para las personas que quieran consultar estas tablas sin necesidad de mirar el código.



Como bien se comenta después, se han intentado priorizar las decisiones de banquete sobre el esquema, pero con algunas excepciones. Se ha decidido mantener el IBAN en las tablas donde existiesen, para no perder información incluida en la base de datos de Banquito, además de añadir un atributo a todas las tablas que define el Banco al que pertenece. Como esta información es tan solo un nombre, se tomó la decisión de tener un atributo, pero de necesitar más información se necesitaría otra tabla, como Codentidades, en la que guardar esto.

Para implementar el esquema global en Oracle y facilitar la integración de los dos esquemas relacionales, se han creado vistas que establecen relaciones entre las tablas de ambos esquemas. Estas vistas actúan como una capa de abstracción sobre las tablas subyacentes, permitiendo consultar datos de manera unificada. Cada vista se corresponde con una tabla y relaciona los atributos de ambas tablas. En los casos donde un atributo no está presente en una de las tablas, se asigna un valor nulo en la vista correspondiente, garantizando la consistencia de los datos.

La mayoría de estas vistas tienen una estructura similar y utilizan la operación UNION entre dos o más SELECT para eliminar columnas duplicadas y fusionar datos de ambas tablas. Sin embargo, algunas vistas requieren el uso de funciones y expresiones para procesar y transformar datos de diferentes tablas. Un ejemplo de esto es la dirección en la tabla titular, donde se optó por adoptar la disposición de los datos del esquema de Banquete debido a su mayor simplicidad. Para lograr esta integración, fue necesario concatenar los atributos de la tabla dirección de la base de datos de Banquete. De esta manera se lograba eliminar del esquema global las tablas relacionadas con la dirección, aunque cabe remarcar que ahora sería necesario usar consultas con *like* para acceder a direcciones de una ciudad o una calle en concreto.

Es importante destacar que, si bien se realizaron modificaciones en ambos esquemas, se priorizó la estructura del esquema de Banquete, en exclusividad del caso de las operaciones de retirada e ingreso, que se tomó la decisión de separarla en dos tablas distintas, pues el concepto de operación encajaba mejor así, dándole la suficiente importancia a cada tipo de operación. Es cierto que esto facilita algunas consultas y todo lo contrario para otras, pero la decisión depende de las preferencias que se tengan. En los casos donde surgieron conflictos entre los esquemas, se realizaron las adaptaciones necesarias en el otro esquema para garantizar la coherencia y consistencia del esquema global. Este enfoque asegura una integración efectiva de las bases de datos, minimizando las discrepancias y facilitando la consulta unificada de datos.

## Parte 2

### Enunciado de un problema de diseño de bases de datos

La Scoala Romaneasca, una universidad de renombre en Rumanía, está en proceso de modernizar su sistema de gestión para mejorar la eficiencia en la administración de sus recursos y optimizar la experiencia de sus miembros.

La universidad se organiza en varios departamentos académicos, de los cuales se quiere guardar el nombre, la descripción y la ubicación, cada uno ofreciendo una amplia variedad de asignaturas en distintas áreas de estudio. Las asignaturas tienen un código de identificación único, nombre, un nivel de dificultad del 0 al 10, un número de créditos asignados y un total de horas estimadas a dedicar. Estas asignaturas se distribuyen de manera cuidadosa para satisfacer las necesidades educativas y profesionales de los estudiantes. Cada asignatura puede pertenecer a un único departamento.

Además de los departamentos académicos, la universidad cuenta con otros servicios y programas. Tales pueden ser el programa de tutoría académica por parte del profesorado, el programa de deportes en el rector por parte del estudiantado, etc... Respecto a los servicios, también son ofrecidos por ambas partes. Los profesores ofrecen servicios como el asesoramiento académico personalizado, mientras que el estudiantado realizan servicios como la bienvenida a estudiantes de erasmus, el control de bienestar en bibliotecas, y más.

La comunidad universitaria está compuesta por dos tipos de miembros, profesores y estudiantes. De ambos se quiere guardar el DNI, nombre apellidos y fecha de nacimiento. De los profesores interesa conocer sus años de experiencia y de los estudiantes la nota media.

El desafío radica en diseñar e implementar un sistema de base de datos que permita gestionar eficazmente la información relacionada con los departamentos, los servicios, los programas y las personas involucradas en el entorno universitario. Esto implica establecer relaciones claras entre las diferentes entidades, garantizando una representación precisa de la estructura organizativa y los procesos académicos y administrativos de la universidad.

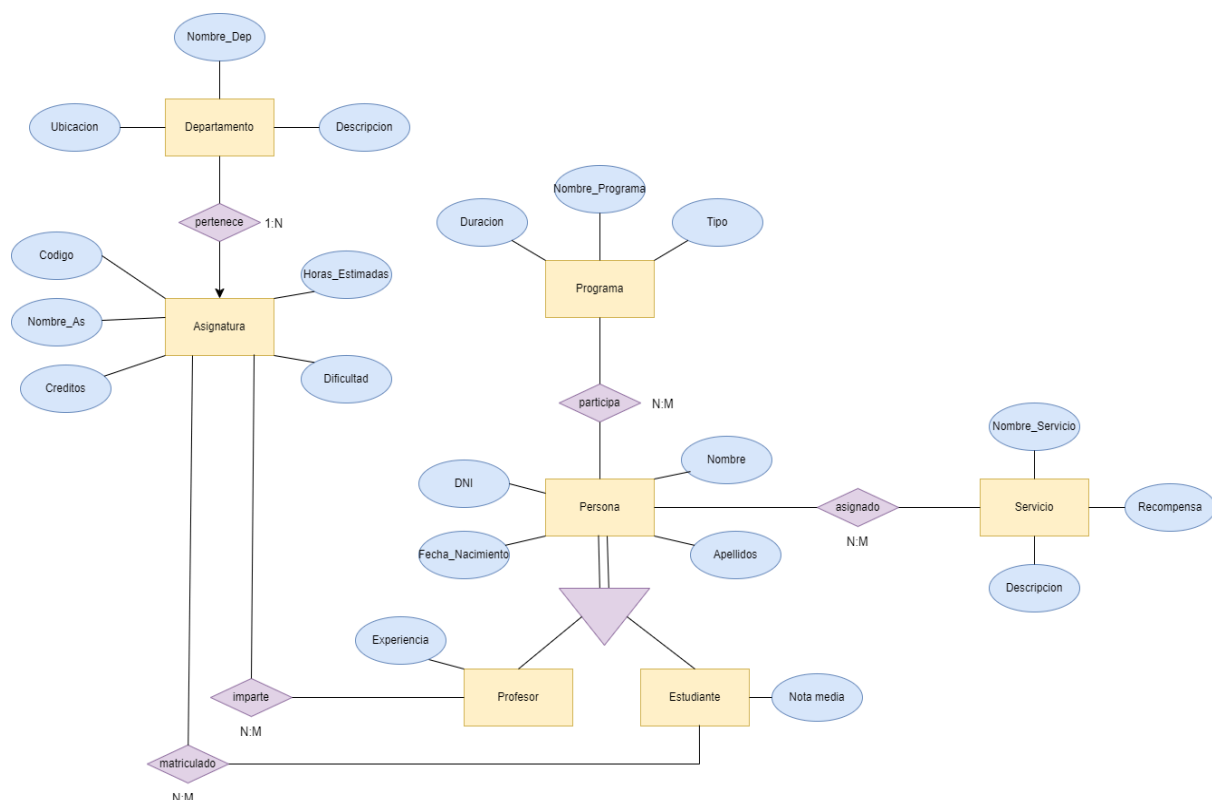
Puede haber diversas relaciones entre las entidades, abarcando desde la asignación de profesores a asignaturas específicas hasta el seguimiento de la matriculación de estudiantes en cursos particulares. Se requerirá un diseño cuidadoso y una implementación robusta para asegurar que el sistema de base de datos sea capaz de manejar todas estas relaciones de manera eficiente.

## Esquema conceptual para el problema enunciado

En el proceso de modernización del sistema de gestión de la Universidad, se ha llevado a cabo un análisis de los requerimientos y necesidades de información de la institución. Como parte de este análisis, se ha diseñado un modelo de base de datos que captura la estructura organizativa y los diversos procesos de la universidad.

En el siguiente diagrama entidad-relación se presenta una vista visual del diseño del modelo de base de datos. El diagrama muestra las entidades principales del sistema, como Departamento, Asignatura, Persona, Estudiante y Profesor, así como las relaciones entre estas entidades. Además, se incluyen entidades adicionales como Servicio y Programa, que son parte del entorno universitario y están relacionadas con las personas y las asignaturas.

El diseño del modelo de base de datos se ha realizado teniendo en cuenta la flexibilidad y la escalabilidad del sistema, así como la capacidad de adaptarse a futuros cambios y actualizaciones en la universidad. Este diseño proporciona una sólida base para la implementación de un sistema de gestión integral que satisfaga las necesidades actuales y futuras de la Universidad.

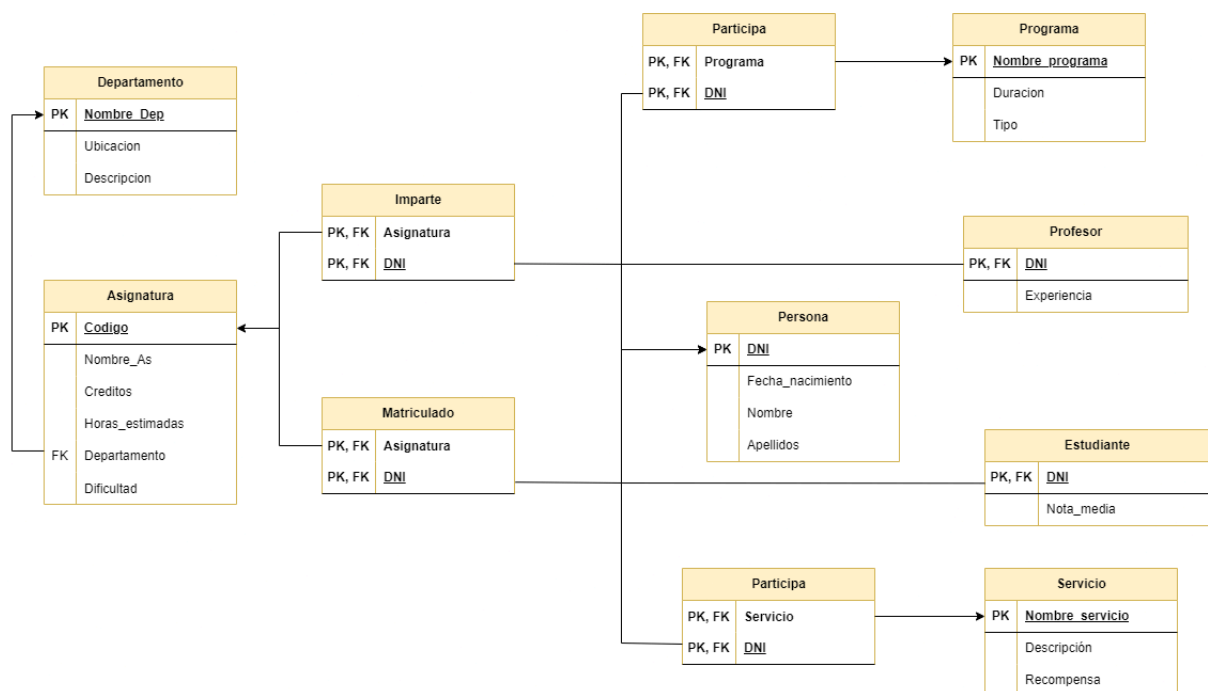


## Esquema lógico 1 para el problema enunciado.

Para poder guardar la base de datos correctamente se ha implementado un esquema relacional que define cómo se guardan las relaciones entre cada entidad. Este primer diseño se ha realizado siguiendo el estándar, procurando ejecutar una base de datos correcta y sin errores de diseño, el segundo diseño será en el que se realizarán cambios no tan correctos.

La especialización realizada por la entidad Persona, se ha empleado la creación de todas las entidades y la relación entre ellas mediante claves extranjeras. Existen otro tipo de implementaciones que se plantearán para el siguiente diseño lógico. Todas las relaciones N:M se han traducido creando una tabla intermedia con las claves primarias de las dos tablas de la relación.

Cabe destacar que se han nombrado los atributos de manera que se puedan entender y que sigan un patrón parecido, para que luego en el segundo diseño lógico se puedan nombrar de manera distinta y aumentar la heterogeneidad y plantear dificultades en la implementación del esquema global.





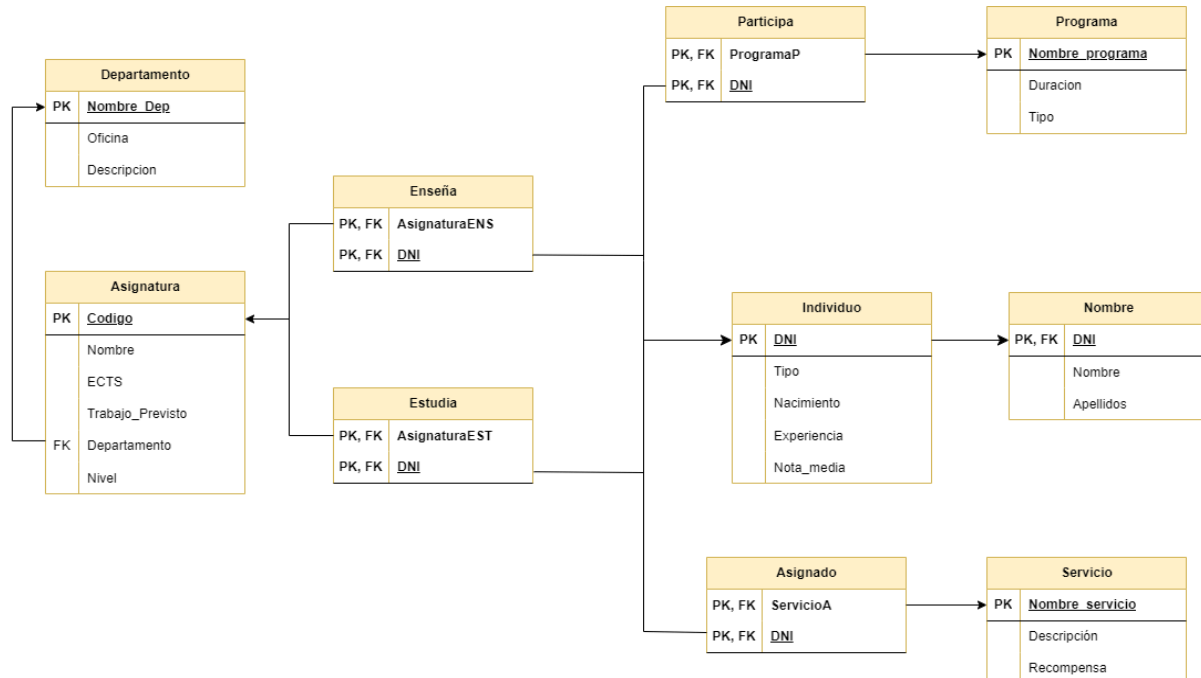
## Esquema lógico 2 para el problema enunciado

En comparación con el modelo lógico anterior, se han realizado cambios significativos en la forma en que se representa la especialización de la entidad "Persona", así como ajustes en los nombres de algunas entidades y atributos para mejorar la claridad y coherencia del modelo.

Una de las modificaciones más importantes se refiere a la especialización de la entidad "Persona". En el modelo lógico anterior, se utilizaba una especialización con dos subentidades separadas para representar a los profesores y estudiantes. Sin embargo, en el nuevo modelo lógico, se ha optado por una estructura más simplificada donde la especialización se representa mediante un único atributo "Tipo" en la entidad "Persona". Este atributo indica si la persona es un profesor o un estudiante, permitiendo una gestión más eficiente de la información sin la necesidad de subentidades separadas.

Además, se han realizado ajustes en los nombres de algunas entidades y atributos para mejorar la consistencia y comprensión del modelo. Estos cambios han sido guiados por las mejores prácticas de diseño de bases de datos y la retroalimentación recibida durante el proceso de desarrollo.

En el siguiente diagrama se presenta el nuevo modelo lógico de la base de datos, que refleja estas mejoras y cambios en la estructura y representación de la información. Este modelo proporciona una base sólida y eficiente para la implementación del sistema de gestión de la Universidad, que satisfaga las necesidades de la institución y facilite su crecimiento y evolución futura.



## Esquema global y su implementación en PostgreSQL

Se ha llevado a cabo la implementación de un esquema global que junto dos esquemas previamente existentes mediante la creación de vistas en Postgres, con la introducción de un nuevo esquema denominado *vista\_combinada*. Este proceso ha sido crucial para facilitar la consulta unificada de datos, ya que las vistas actúan como una capa de abstracción, permitiendo la combinación de información proveniente de diversas tablas en una sola tabla virtual, gracias a la operación UNION entre SELECTs.

Uno de los aspectos esenciales en este proceso ha sido la priorización de la estructura de uno de los esquemas preexistentes sobre el otro, con el objetivo de simplificar la integración. Para ello, se adaptó el esquema menos complejo para que fuese compatible con el esquema prioritario seleccionado. Este enfoque estratégico garantizó una transición fluida y eficiente hacia el nuevo esquema global.

Posteriormente, se procedió a poblar los esquemas con datos generados automáticamente, aunque solo se incluyó una parte del código de población en el archivo de pruebas. Dado que el propósito fundamental de esta práctica era verificar el correcto funcionamiento del esquema global, se diseñaron consultas simples y directas para facilitar la evaluación de su rendimiento. Este enfoque permitió una validación rápida y eficaz del esquema global, asegurando su coherencia y consistencia.

Es importante destacar que, a pesar de la simplicidad de las consultas realizadas en las pruebas, se logró demostrar la capacidad del esquema global para integrar datos de manera efectiva y precisa.

## Actualizaciones de datos sobre el esquema global:

Las actualizaciones sobre el esquema global pueden afectar al rendimiento del sistema, especialmente en entornos de alta concurrencia. Es importante realizar pruebas exhaustivas para evaluar el rendimiento y optimizar el procesamiento de las operaciones de actualización. Es necesario implementar estrategias de control de transacciones para mantener la consistencia de los datos a través de las diferentes bases de datos involucradas.

Para evaluar la viabilidad y conveniencia de permitir actualizaciones sobre el esquema global, se realizarán pruebas y demostraciones en un entorno controlado. Se simularán diferentes escenarios de actualización, como inserción, modificación y eliminación de datos, y se analizarán los resultados obtenidos. En conclusión, la posibilidad de permitir actualizaciones sobre el esquema global ofrece beneficios potenciales en términos de gestión y eficiencia de datos. Sin embargo, es crucial abordar adecuadamente las implicaciones relacionadas con la integridad, seguridad, rendimiento y transacciones distribuidas. Mediante pruebas y evaluaciones rigurosas, se determinará si la implementación de esta funcionalidad es factible y beneficiosa para el sistema en su conjunto. Además, se debe tener en cuenta que las actualizaciones deben realizarse con precaución y siempre considerando el impacto en la integridad y consistencia de los datos en todo el entorno de bases de datos federadas.

## Esfuerzos invertidos:

	Miembros del grupo		
Tarea	Guillermo Bajo	Andrei Dumbrava	Pablo Moreno
Parte 1. Banquete	2h	4h	2h
Parte 1. Esquema global	30min	1h 30 min	2h
Parte 1. Implementación vistas	0min	2h 30 min	0min
Parte 1. Pruebas	0min	1hmin	1h
Parte 2. Enunciado problema	1h	20 min	0min
Parte 2. Esquema E/R	2h	10 min	0min
Parte 2. Diseño e implementación 1	1h 30min	30 min	2h
Parte 2. Diseño e implementación 2	1h	30 min	2h
Parte 2. Pruebas	0min	0 min	2h
Memoria	4h	2h	2h
<b>Total</b>	<b>12h</b>	<b>12h 30 min</b>	<b>13h</b>