

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO
BASE DE DATOS 1



Pablo Larnaudie – 340181



Natalia Rebella – 327283

Grupo: N2A

Docente: Rafael Cohen

ATI

Fecha de entrega: 21-11-2024

Índice

1.	Introducción.....	4
2.	Modelo entidad relación (MER).....	4
2.1	Diagrama Entidad Relación	4
2.2	Restricciones no estructurales.....	5
2.3	Justificaciones y otras consideraciones.....	6
3.	Modelo relacional (MR)	6
3.1.	Restricciones.....	8
3.2.	Normalización.....	9
	Tabla: TURISTA_USUARIO.....	9
	Tabla: TURISTAS_TELEFONOS	9
	Tabla: PASAJE.....	10
	Tabla: DESTINO.....	10
	Tabla: TERMINAL.....	10
	Tabla: DEPARTAMENTO	10
	Tabla: BUS.....	10
	Tabla: ASIENTO.....	10
	Tabla: CATEGORIA	10
	Tabla: BENEFICIO	10
	Tabla: TURISTA_NO_REGISTRADO	10
	Tabla: TURISTAS_NO_REGISTRADOS_TELEFONOS	11
	Tabla: FUNCIONARIO.....	11
	Tabla: TURISTA_USUARIO.....	11
	Tabla: TURISTAS_TELEFONOS	11

Tabla: TURISTAS_DOCUMENTO	11
Tabla: PASAJE.....	11
Tabla: DESTINO.....	12
Tabla: TERMINAL.....	12
Tabla: DEPARTAMENTO	12
Tabla: BUS.....	12
Tabla: ASIENTO.....	12
Tabla: CATEGORIA	12
Tabla: BENEFICIO	12
Tabla: TURISTA_NO_REGISTRADO	12
Tabla: TURISTAS_NO_REGISTRADOS_TELEFONOS	12
Tabla: TURISTAS_NO_REGISTRADO_DOCUMENTO	12
Tabla: FUNCIONARIO.....	13
4. SQL.....	13
4.1 Data Manipulation Language (DML)	13
4.2 Consultas SQL	17
4.3.1. Consulta 1	17
4.3.2. Consulta 2	18
4.3.3. Consulta 3	19
4.3.4. Consulta 4	19
4.3.5. Consulta 5	20

1. Introducción

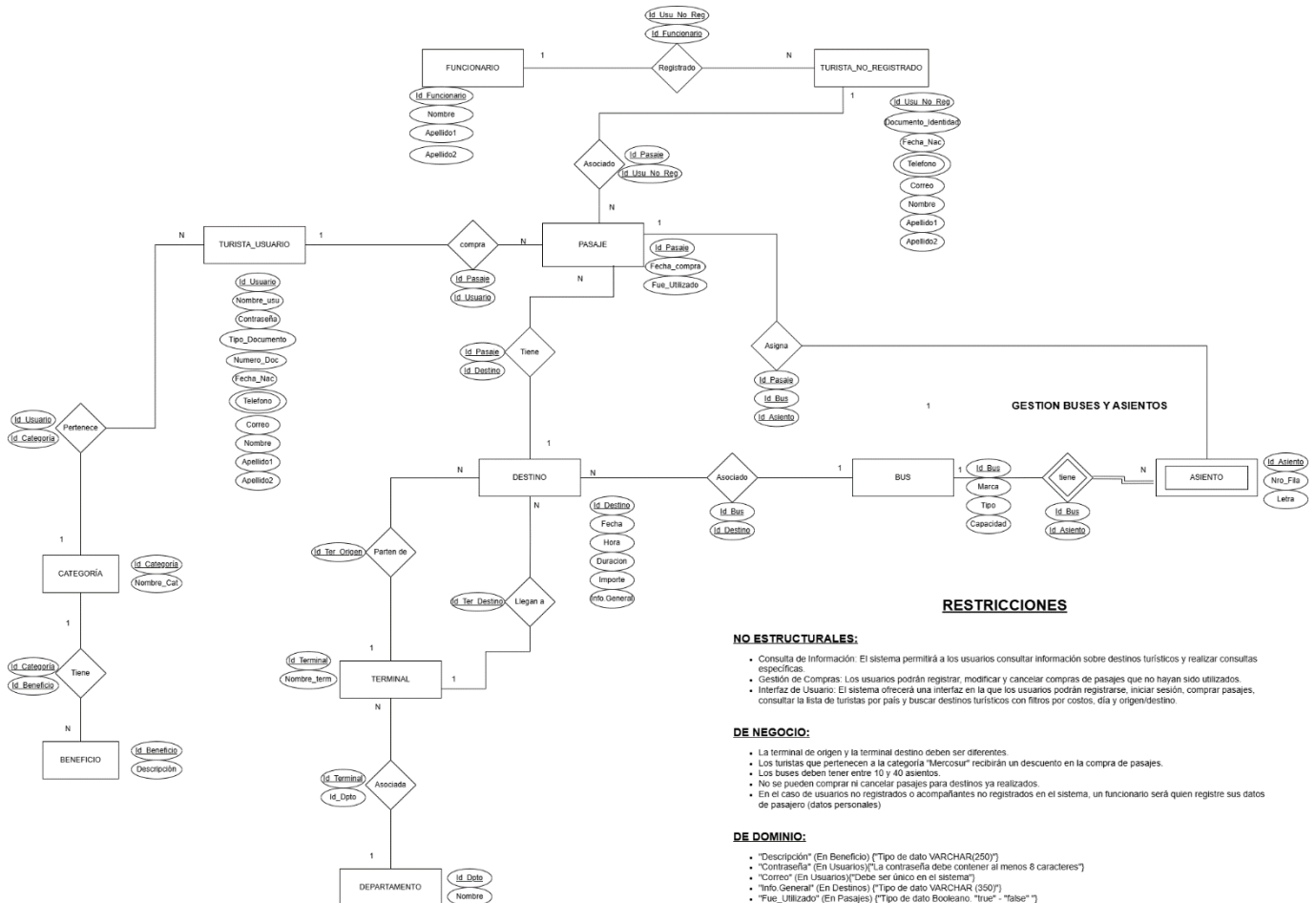
El objetivo de este trabajo es diseñar una base de datos para una empresa de turismo que ofrece tours en buses en modalidad express. La aplicación debe cubrir todos los aspectos necesarios para permitir a los turistas consultar destinos, registrar compras, y gestionar sus pasajes de forma intuitiva y segura.

La base de datos debe soportar funcionalidades clave, como el registro y autenticación de turistas, permitiendo a los usuarios gestionar sus reservas y realizar consultas personalizadas. Los turistas podrán filtrar destinos por diversos criterios y tendrán beneficios específicos si pertenecen a la categoría "Mercosur". Además, se deben registrar características específicas de cada tour, tales como terminales de origen y destino, información detallada del bus y de los asientos.

El diseño debe cumplir con los principios de normalización y asegurar la integridad de los datos, facilitando consultas avanzadas de gestión que podrían incluir información sobre pasajeros frecuentes, buses de gran capacidad, estadísticas de viajes, etc. Este sistema proporcionará a la empresa una herramienta sólida para el control de sus operaciones turísticas y una experiencia de usuario mejorada.

2. Modelo entidad relación (MER)

2.1 Diagrama Entidad Relación



2.2 Restricciones no estructurales

NO ESTRUCTURALES:

- Consulta de Información: El sistema permitirá a los usuarios consultar información sobre destinos turísticos y realizar consultas específicas.
- Gestión de Compras: Los usuarios podrán registrar, modificar y cancelar compras de pasajes que no hayan sido utilizados.
- Interfaz de Usuario: El sistema ofrecerá una interfaz en la que los usuarios podrán registrarse, iniciar sesión, comprar pasajes, consultar la lista de turistas por país y buscar destinos turísticos con filtros por costos, día y origen/destino.

DE NEGOCIO:

- La terminal de origen y la terminal destino deben ser diferentes.
- Los turistas que pertenecen a la categoría "Mercosur" recibirán un descuento en la compra de pasajes.
- Los buses deben tener entre 10 y 40 asientos.
- No se pueden comprar ni cancelar pasajes para destinos ya realizados.
- En el caso de usuarios no registrados o acompañantes no registrados en el sistema, un funcionario será quien registre sus datos de pasajero (datos personales)

DE DOMINIO:

- "Descripción" (En Beneficio) {"Tipo de dato VARCHAR(250)"}
- "Contraseña" (En Usuarios){ "La contraseña debe contener al menos 8 caracteres" }
- "Correo" (En Usuarios){ "Debe ser único en el sistema" }
- "Info.General" (En Destinos) {"Tipo de dato VARCHAR (350)"}
- "Fue_Utilizado" (En Pasajes) {"Tipo de dato Booleano. "true" - "false" " }

2.3 Justificaciones y otras consideraciones

Las restricciones no estructurales son aquellas que no están directamente relacionadas con la estructura o el modelo de datos del sistema, sino con el comportamiento y funcionalidades del mismo. Estas restricciones definen el comportamiento del sistema desde el punto de vista del usuario.

Por otro lado, las restricciones de negocio son reglas que reflejan cómo debe funcionar el sistema según las políticas y necesidades de la organización, asegurando que se contemplen los objetivos del negocio.

Por último, las restricciones de dominio son reglas que permiten validar el tipo de dato que recibe la base de datos en sus atributos, habilitando la posibilidad de validar un correcto y consistente ingreso de valores.

3. Modelo relacional (MR)

A continuación se detalla el Esquema Relacional inicial. En el mismo se presentan y desarrollan, primero las entidades y luego las relaciones presentes en el MER (que se convertirán en tablas). En ambos casos se incluyen sus respectivos atributos.

TURISTA_USUARIO (Id_Usuario, Nombre_usu, Contraseña, Tipo_Documento, Numero_Doc, Fecha_Nac, Telefono*, Correo, Nombre, Apellido1, Apellido2)

PASAJE (Id_Pasaje, Fecha_compra, Fue_Utilizado)

DESTINO (Id_Destino, Fecha, Hora, Duracion, Importe, Info.General)

TERMINAL (Id_Terminal, Nombre_term)

DEPARTAMENTO (Id_Dpto, Nombre)

BUS (Id_Bus, Marca, Tipo, Capacidad)

ASIENTO (Id_Asiento, Nro_Fila, Letra)

CATEGORIA (Id_Categoria, Nombre_Cat)

BENEFICIO (Id_Beneficio, Descripcion)

TURISTA_NO_REGISTRADO (Id_Usu_No_Reg, Tipo_Documento, Numero_Doc, Fecha_Nac, Telefono*, Correo, Nombre, Apellido1, Apellido2)

FUNCIONARIO (Id_Funcionario, Nombre, Apellido1, Apellido2)

COMPRA (Id_Pasaje, Id_Usuario)

TIENE (Id_Pasaje, Id_Destino)

PARTEN DE (Id_Ter_Origen, Id_Destino)

LLEGAN A (Id_Ter_Destino, Id_Destino)

ASOCIADA (Id_Terminal, Id_Dpto)

AGISNA (Id_Pasaje, Id_Bus, Id_Asiento)

ASOCIADO (Id_Destino, Id_Bus)

TIENE (Id_Bus, Id_Asiento)

PERTENECE (Id_Usuario, Id_Categoria)

TIENE (Id_Categoria, Id_Beneficio)

ASOCIADO (Id_Pasaje, Id_Usu_No_Reg)

REGISTRADO (Id_Funcionario, Id_Usu_No_Reg)

A continuación vamos a analizar si existen tablas de relación que puedan ser “absorbidas” por tablas de entidades. En cada caso vamos a analizar la cardinalidad entre las entidades participantes de cada relación. Como hemos visto en clase, si la relación es N-1, 1-N o 1-1, entonces sí es posible omitir la tabla relación. En los casos que hay N, los atributos de la relación, se colocan en la tabla de la entidad que tenga esa cardinalidad. En el caso de 1-1, se elige en qué tabla se colocarán los atributos de la relación.

TURISTA_USUARIO (Id_Usuario, Nombre_usu, Contraseña, Tipo_Documento, Numero_Doc, Fecha_Nac, Telefono*, Correo, Nombre, Apellido1, Apellido2, **Id_Categoria (FK)**)

PASAJE (Id_Pasaje, Fecha_compra, Fue_Utilizado, **Id_Usuario (FK)**, **Id_Destino (FK)**, **Id_Usu_No_Reg (FK)**, **Id_Bus (FK)**, **Id_Asiento(FK)**)

DESTINO (Id_Destino, Fecha, Hora, Duracion, Importe, Info.General, **Id_Ter_Origen (FK)**, **Id_Ter_Destino (FK)**, **Id_Bus (FK)**)

TERMINAL (Id_Terminal, Nombre_term, **Id_Dpto (FK)**)

DEPARTAMENTO (Id_Dpto, Nombre)

BUS (Id_Bus, Marca, Tipo, Capacidad)

ASIENTO (Id_Asiento, Nro_Fila, Letra, **Id_Bus (FK)**)

CATEGORIA (Id_Categoria, Nombre_Cat)

BENEFICIO (Id_Beneficio, Descripcion, **Id_Categoria (FK)**)

TURISTA_NO_REGISTRADO (Id_Usu_No_Reg, Tipo_Documento, Numero_Doc, Fecha_Nac, Telefono*, Correo, Nombre, Apellido1, Apellido2, **Id_Funcionario (FK)**)

FUNCIONARIO (Id_Funcionario, Nombre, Apellido1, Apellido2)

3.1. Restricciones

NO ESTRUCTURALES:

- Consulta de Información: El sistema permitirá a los usuarios consultar información sobre destinos turísticos y realizar consultas específicas.
- Gestión de Compras: Los usuarios podrán registrar, modificar y cancelar compras de pasajes que no hayan sido utilizados.
- Interfaz de Usuario: El sistema ofrecerá una interfaz en la que los usuarios podrán registrarse, iniciar sesión, comprar pasajes, consultar la lista de turistas por país y buscar destinos turísticos con filtros por costos, día y origen/destino.

DE NEGOCIO:

- La terminal de origen y la terminal destino deben ser diferentes.
- Los turistas que pertenecen a la categoría "Mercosur" recibirán un descuento en la compra de pasajes.
- Los buses deben tener entre 10 y 40 asientos.
- No se pueden comprar ni cancelar pasajes para destinos ya realizados.
- En el caso de usuarios no registrados o acompañantes no registrados en el sistema, un funcionario será quien registre sus datos de pasajero (datos personales)

DE DOMINIO:

- "Descripción" (En Beneficio) {"Tipo de dato VARCHAR(250)"}
- "Contraseña" (En Usuarios) {"La contraseña debe contener al menos 8 caracteres"}
- "Correo" (En Usuarios) {"Debe ser único en el sistema"}
- "Info.General" (En Destinos) {"Tipo de dato VARCHAR (350)"}
- "Fue_Utilizado" (En Pasajes) {"Tipo de dato Booleano. "true" - "false" "}"}

3.2.Normalización

*Procederemos a realizar la **NORMALIZACIÓN** hasta 3° forma normal.*

***1° Forma Normal:** partiremos de la base de que una relación esta en 1FN cuando cada atributo tiene un valor único o también conocido como atómico. En este caso, eliminamos los teléfonos, tanto de la tabla **TURISTA_USUARIO** como **TURISTA_NO_REGISTRADO**, ya que eran atributos multivaluados (cada turista podría contener más de un número telefónico), creando dos tablas nuevas con dichos atributos, cada uno vinculado con el ID correspondiente a su tabla original.*

Tabla: TURISTA_USUARIO

Id_Us uario	Nombr e_usu	Contr aseña	Tipo_Doc umento	Numer o_Doc	Fecha _Nac	Cor reo	Nom bre	Apell ido1	Apell ido2	Id_Cat egoria (FK)
------------------------	------------------------	------------------------	----------------------------	------------------------	-----------------------	--------------------	--------------------	-----------------------	-----------------------	-----------------------------------

Tabla: TURISTAS_TELEFONOS

Id_Usuario	Teléfono
-------------------	-----------------

Tabla: PASAJE

Id_Pasaje	Fecha_compra	Fue_Utilizado	Id_Usuario (FK)	Id_Destino (FK)	Id_Usu_No_Reg (FK)	Id_Bus (FK)	Id_Asiento (FK)
------------------	---------------------	----------------------	------------------------	------------------------	---------------------------	--------------------	------------------------

Tabla: DESTINO

Id_Destino	Fecha	Hora	Duracion	Importe	Info.Genera	Id_Ter_Origen (FK)	Id_Ter_Destino (FK)	Id_Bus (FK)
-------------------	--------------	-------------	-----------------	----------------	--------------------	---------------------------	----------------------------	--------------------

Tabla: TERMINAL

Id_Terminal	Nombre_term	Id_Dpto (FK)
--------------------	--------------------	---------------------

Tabla: DEPARTAMENTO

Id_Dpto	Nombre
----------------	---------------

Tabla: BUS

Id_Bus	Marca	Tipo	Capacidad
---------------	--------------	-------------	------------------

Tabla: ASIENTO

Id_Asiento	Nro_Fila	Letra	Id_Bus (FK)
-------------------	-----------------	--------------	--------------------

Tabla: CATEGORIA

Id_Categoria	Nombre_Cat
---------------------	-------------------

Tabla: BENEFICIO

Id_Beneficio	Descripcion	Id_Categoria (FK)
---------------------	--------------------	--------------------------

Tabla: TURISTA_NO_REGISTRADO

Id_Usu_No_Reg	Tipo_Documento	Numero_Doc	Fecha_Nac	Correo	Nombre	Apellido1	Apellido2	Id_Funcionario (FK)
----------------------	-----------------------	-------------------	------------------	---------------	---------------	------------------	------------------	----------------------------

Tabla: TURISTAS_NO_REGISTRADOS_TELEFONOS

Id_Usuario	Teléfono
-------------------	-----------------

Tabla: FUNCIONARIO

Id_Funcionario	Nombre	Apellido1	Apellido2
-----------------------	---------------	------------------	------------------

2ª Forma Normal: en este caso, cada atributo que no sea clave primaria debe depender completamente de la misma, teniendo una dependencia funcional completa con ella. Se buscan eliminar dependencias parciales. En este caso, el Tipo_Documento y el Numero_Doc están relacionados entre sí y no dependen del resto de la información del usuario. Esto indica una dependencia parcial con respecto al Id_Usuario, por lo tanto, se procede a crear una nueva tabla con dichos atributos.

Tabla: TURISTA_USUARIO

Id_Usua rio	Nombre_ usu	Contrase ña	Fecha_ Nac	Corr eo	Nomb re	Apellid o1	Apellid o2	Id_Catego ria (FK)
------------------------	------------------------	------------------------	-----------------------	--------------------	--------------------	-----------------------	-----------------------	-------------------------------

Tabla: TURISTAS_TELEFONOS

Id_Usuario	Teléfono
-------------------	-----------------

Tabla: TURISTAS_DOCUMENTO

Id_Usuario	Tipo_Documento	Numero_Doc
-------------------	-----------------------	-------------------

Tabla: PASAJE

Id_Pas aje	Fecha_com pra	Fue_Utiliz ado	Id_Usua rio (FK)	Id_Desti no (FK)	Id_Usu_No_ Reg (FK)	Id_B us (FK)	Id_Asie nto (FK)
-----------------------	--------------------------	---------------------------	-----------------------------	-----------------------------	--------------------------------	-----------------------------	-----------------------------

Tabla: DESTINO

Id_Destino	Fecha	Hora	Duración	Importe	Info.General	Id_Ter_Origen (FK)	Id_Ter_Destino (FK)	Id_Bus (FK)
-------------------	--------------	-------------	-----------------	----------------	---------------------	---------------------------	----------------------------	--------------------

Tabla: TERMINAL

Id_Terminal	Nombre_term	Id_Dpto (FK)
--------------------	--------------------	---------------------

Tabla: DEPARTAMENTO

Id_Dpto	Nombre
----------------	---------------

Tabla: BUS

Id_Bus	Marca	Tipo	Capacidad
---------------	--------------	-------------	------------------

Tabla: ASIENTO

Id_Asiento	Nro_Fila	Letra	Id_Bus (FK)
-------------------	-----------------	--------------	--------------------

Tabla: CATEGORIA

Id_Categoria	Nombre_Cat
---------------------	-------------------

Tabla: BENEFICIO

Id_Beneficio	Descripcion	Id_Categoria (FK)
---------------------	--------------------	--------------------------

Tabla: TURISTA_NO_REGISTRADO

Id_Usu_No_Reg	Fecha_Nacimiento	Correo	Nombre	Apellido 1	Apellido 2	Id_Funcionario (FK)
----------------------	-------------------------	---------------	---------------	-------------------	-------------------	----------------------------

Tabla: TURISTAS_NO_REGISTRADOS_TELEFONOS

Id_Usuario	Teléfono
-------------------	-----------------

Tabla: TURISTAS_NO_REGISTRADO_DOCUMENTO

Id_Usu_No_Reg	Tipo_Documento	Numero_Doc
----------------------	-----------------------	-------------------

Tabla: FUNCIONARIO

Id_Funcionario	Nombre	Apellido1	Apellido2
-----------------------	---------------	------------------	------------------

3° Forma Normal: en este paso se busca eliminar las redundancias y asegurar la integridad de los datos. Se deben identificar dependencias transitivas, esto implica que ningún atributo no clave debe depender de otro atributo no clave. En el caso de las tablas presentadas hasta el momento, no se identifica ninguna dependencia transitiva, por lo que se podría concluir la 3° forma normal.

4. SQL

4.1 Data Manipulation Language (DML)

A continuación se presentan las precargas de datos que se utilizaron para realizar las consultas solicitadas.

--Inserción de turistas registrados

```
INSERT INTO TURISTA_USUARIO (id_usuario, nombre_usuario, contraseña, fecha_nac, correo, nombre, apellido1, apellido2, id_categoria)
VALUES
(1, 'juanperez', 'password123', '1985-05-12', 'juanp@gmail.com', 'Juan', 'Perez', 'Gomez', 1),
(2, 'sofia_lopez', 'SofiLopez123', '1990-08-21', 'sofi@gmail.com', 'Sofia', 'Lopez', 'Martinez', 2),
(3, 'maria_gonz', 'MariaGonz1234', '1988-03-30', 'maria@gmail.com', 'Maria', 'Gonzalez', 'Sanchez', 1),
(4, 'soyturista', 'SoyTurista2017', '1992-01-15', 'soyturista@gmail.com', 'Carlos', 'Gomez', 'Rodriguez', 1);
```

--Inserción de turistas registrados teléfonos

```
INSERT INTO TURISTA_TELEFONOS (id_usuario, telefono)
VALUES
(1, '091234567'),
(2, '092345678'),
(3, '093456789'),
(4, '094567890');
```

--Inserción de turistas registrados documento

```

INSERT INTO TURISTA_DOCUMENTO (id_usuario, tipo_documento, numero_documento)
VALUES
(1, 'CI', '45678912'),
(2, 'CI', '56789123'),
(3, 'Pasaporte', 'AB123456'),
(4, 'CI', '67891234');

```

--Inserción de pasaje

```

INSERT INTO PASAJE (id_pasaje, fecha_compra, fue_utilizado, id_usuario,
id_destino, id_usu_no_reg, id_bus, id_asiento)
VALUES
(101, '2023-09-01', 1, 1, 250, NULL, 301, 401),
(102, '2023-09-05', 1, 1, 254, NULL, 302, 402),
(103, '2023-09-10', 1, 2, 250, NULL, 303, 403),
(104, '2023-09-15', 0, 2, 248, NULL, 304, 404),
(105, '2023-09-20', 0, 3, 255, NULL, 305, 405),
(106, '2023-09-25', 1, 3, 255, NULL, 305, 406),
(107, '2023-09-01', 1, 4, 248, NULL, 301, 407),
(108, '2023-09-05', 1, 4, 249, NULL, 302, 408),
(109, '2023-09-10', 1, 4, 250, NULL, 303, 409),
(110, '2023-09-15', 1, 4, 251, NULL, 304, 410),
(111, '2023-09-20', 1, 4, 252, NULL, 305, 411),
(112, '2017-09-01', 1, 4, 253, NULL, 306, 412), -- Pasaje del 2017
(113, '2017-09-10', 1, 4, 254, NULL, 306, 413); -- Pasaje del 2017

```

```

INSERT INTO PASAJE (id_pasaje, fecha_compra, fue_utilizado, id_usuario,
id_destino, id_usu_no_reg, id_bus, id_asiento)
VALUES
(114, '2023-09-10', 1, 1, 249, NULL, 303, 403),
(115, '2023-09-12', 0, 1, 248, NULL, 304, 404),
(116, '2023-09-15', 1, 1, 250, NULL, 305, 405),
(117, '2023-09-18', 0, 1, 255, NULL, 306, 406);

```

```

INSERT INTO PASAJE (id_pasaje, fecha_compra, fue_utilizado, id_usuario,
id_destino, id_usu_no_reg, id_bus, id_asiento)
VALUES
(120, '2024-09-23', 1, 4, 253, NULL, 306, 406), -- Pasaje en septiembre de
2024
(121, '2024-09-18', 1, 4, 254, NULL, 305, 405); -- Pasaje en septiembre de
2024

```

--Inserción de destino

```

INSERT INTO DESTINO (id_destino, fecha, hora, duracion, importe, info_general,
id_ter_origen, id_ter_destino, id_bus)
VALUES
(248, '2024-10-25', '10:00:00', '02:00:00', 150.00, 'Destino a la ciudad', 101,
102, 301),
(249, '2024-10-26', '15:00:00', '03:00:00', 200.00, 'Destino a la playa', 103,
104, 302),
(250, '2024-10-27', '09:00:00', '01:30:00', 120.00, 'Destino a la montaña', 105,
106, 305),

```

```
(251, '2024-10-28', '14:00:00', '02:30:00', 180.00, 'Destino a la selva', 107,
108, 303),
(252, '2024-10-29', '16:00:00', '03:00:00', 220.00, 'Destino a la playa norte',
109, 110, 304),
(253, '2017-09-01', '10:00:00', '02:00:00', 150.00, 'Destino a la ciudad', 101,
102, 306),
(254, '2017-09-10', '12:00:00', '01:30:00', 120.00, 'Destino a la playa', 103,
104, 306),
(255, '2024-10-27', '09:00:00', '01:30:00', 120.00, 'Destino a la montaña', 105,
106, 305);
```

--Inserción de terminal

```
INSERT INTO TERMINAL (id_terminal, nombre_term, id_dpto)
VALUES
(101, 'Terminal Central', 10),
(102, 'Terminal Norte', 11),
(103, 'Terminal Sur', 12),
(104, 'Terminal Este', 13),
(105, 'Terminal Oeste', 14),
(106, 'Terminal Interurbano', 15),
(107, 'Terminal Norte 2', 10),
(108, 'Terminal Sur 2', 11),
(109, 'Terminal Este 2', 12),
(110, 'Terminal Oeste 2', 13);
```

--Inserción de departamento

```
INSERT INTO DEPARTAMENTO (id_dpto, nombre)
VALUES
(10, 'Montevideo'),
(11, 'Canelones'),
(12, 'Maldonado'),
(13, 'Colonia'),
(14, 'Paysandú'),
(15, 'Salto');
```

--Inserción de bus

```
INSERT INTO BUS (id_bus, marca, tipo, capacidad)
VALUES
(301, 'Mercedes', 'Semi-Cama', 40),
(302, 'Volvo', 'Común', 36),
(303, 'Scania', 'Cama', 38),
(304, 'Mercedes', 'Semi-Cama', 40),
(305, 'Volvo', 'Común', 39),
(306, 'Scania', 'Cama', 40);

INSERT INTO BUS(id_bus, marca, tipo, capacidad)
VALUES
(307, 'Volvo', 'Semi-Cama', 25),
(308, 'Scania', 'Común', 28);
```

--Inserción de asiento

```
INSERT INTO ASIENTO (id_asiento, fila, letra, id_bus)
VALUES
```

```
(401, 1, 'A', 301),
(402, 2, 'B', 302),
(403, 3, 'C', 303),
(404, 4, 'D', 304),
(405, 5, 'E', 305),
(406, 6, 'F', 305),
(407, 7, 'G', 301),
(408, 8, 'H', 302),
(409, 9, 'I', 303),
(410, 10, 'J', 304),
(411, 11, 'K', 305),
(412, 12, 'L', 306),
(413, 13, 'M', 306);
```

--Inserción de categoria

```
INSERT INTO CATEGORIA (id_categoria, nombre_cat)
VALUES
```

```
(1, 'Regular'),
(2, 'VIP');
```

--Inserción de beneficio

```
INSERT INTO BENEFICIO (id_beneficio, descripcion, id_categoria)
VALUES
```

```
(1, 'Descuento 10%', 1),
(2, 'Acceso a sala VIP', 2);
```

--Inserción de turista no registrado

```
INSERT INTO TURISTA_NO_REGISTRADO (id_usu_no_reg, fecha_nac, correo, nombre,
apellido1, apellido2, id_funcionario)
```

```
VALUES
(1, '1995-12-05', 'no_registrado1@gmail.com', 'Pedro', 'Fernandez', 'Garcia', 1),
(2, '1987-06-10', 'no_registrado2@gmail.com', 'Ana', 'Lopez', 'Mendez', 2);
```

--Inserción de turista no registrado telefono

```
INSERT INTO TURISTAS_NO_REGISTRADOS_TELEFONOS (id_usu_no_reg, telefono)
VALUES
```

```
(1, '099876543'),
(2, '098765432');
```

--Inserción de turista no registrado documento

```
INSERT INTO TURISTAS_NO_REGISTRADO_DOCUMENTO (id_usu_no_reg, tipo_documento,
numero_doc)
```

```
VALUES
(1, 'CI', '12345678'),
(1, 'Pasaporte', 'AB987654'),
(2, 'CI', '23456789'),
(2, 'Pasaporte', 'CD876543');
```


--Inserción de funcionario

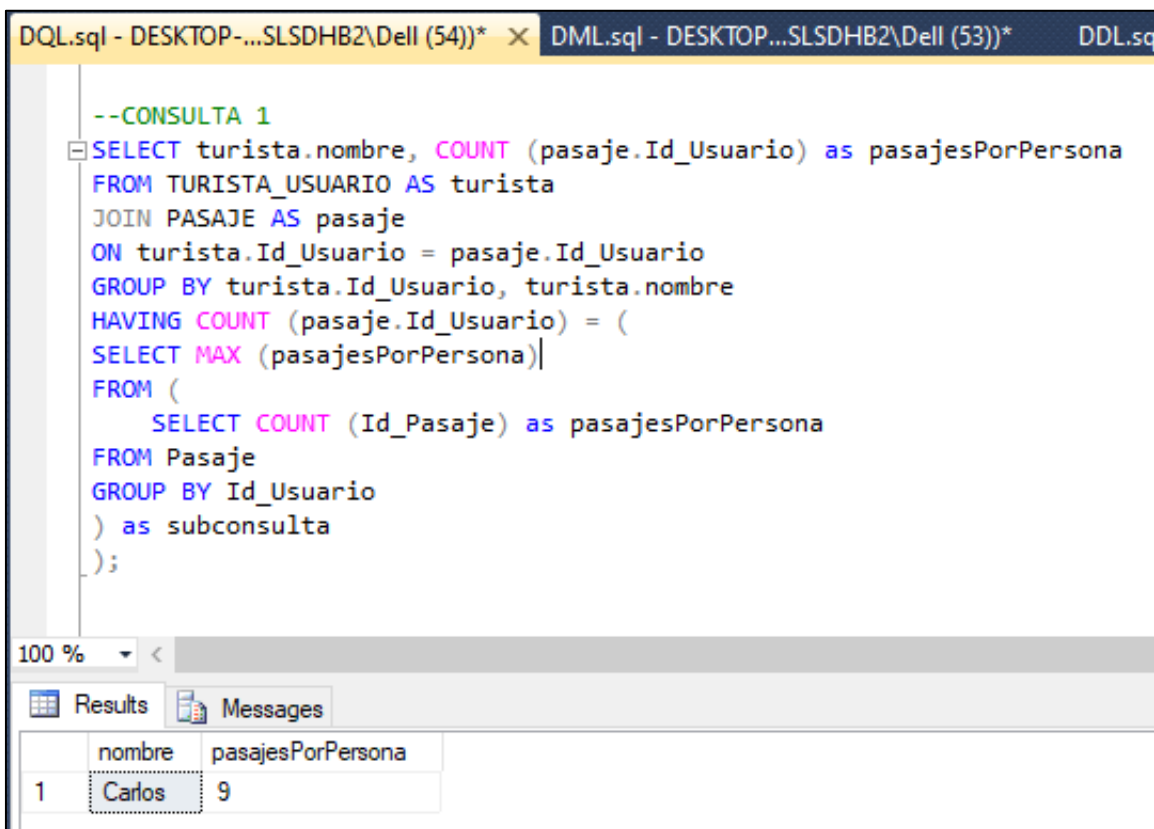
```
INSERT INTO FUNCIONARIO (id_funcionario, nombre, apellido1, apellido2)
VALUES
(1, 'Laura', 'Rodriguez', 'Perez'),
(2, 'Mario', 'Gomez', 'Silva');
```

4.2 Consultas SQL

Se presentan las consultas realizadas en SQL, en base a lo solicitado en la letra del trabajo. Así mismo, se adjunta la imagen correspondiente a los resultados obtenidos con las mismas.

4.3.1. Consulta 1

Listar el o los nombres de los pasajeros con la mayor cantidad de pasajes comprados a su nombre.

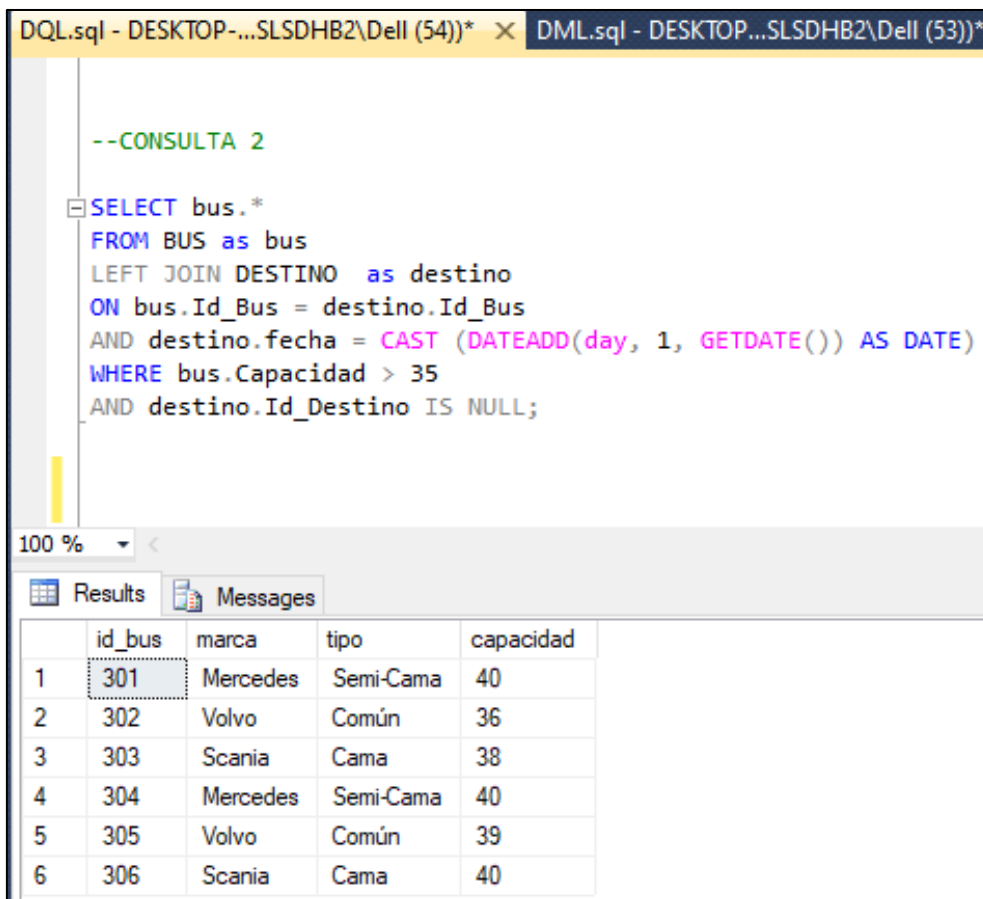


```
--CONSULTA 1
SELECT turista.nombre, COUNT (pasaje.Id_Usuario) as pasajesPorPersona
FROM TURISTA_USUARIO AS turista
JOIN PASAJE AS pasaje
ON turista.Id_Usuario = pasaje.Id_Usuario
GROUP BY turista.Id_Usuario, turista.nombre
HAVING COUNT (pasaje.Id_Usuario) = (
SELECT MAX (pasajesPorPersona)
FROM (
    SELECT COUNT (Id_Pasaje) as pasajesPorPersona
    FROM Pasaje
    GROUP BY Id_Usuario
) as subconsulta
);
```

	nombre	pasajesPorPersona
1	Carlos	9

4.3.2. Consulta 2

Listar todos los datos de los buses con más de 35 asientos que no tengan asignado ningún destino que parta el día de mañana.



```
--CONSULTA 2

SELECT bus.*
FROM BUS as bus
LEFT JOIN DESTINO as destino
ON bus.Id_Bus = destino.Id_Bus
AND destino.fecha = CAST (DATEADD(day, 1, GETDATE()) AS DATE)
WHERE bus.Capacidad > 35
AND destino.Id_Destino IS NULL;
```

100 %

Results Messages

	id_bus	marca	tipo	capacidad
1	301	Mercedes	Semi-Cama	40
2	302	Volvo	Común	36
3	303	Scania	Cama	38
4	304	Mercedes	Semi-Cama	40
5	305	Volvo	Común	39
6	306	Scania	Cama	40

4.3.3. Consulta 3

Listar todos los datos de los pasajeros para los cuales haya registrados en el sistema más de 5 pasajes comprados.

DQL.sql - DESKTOP-...SLSDHB2\Dell (54))* X DML.sql - DESKTOP-...SLSDHB2\Dell (53))* DDL.sql - DESKTOP-...SLSDHB2\Dell (52))

```
--CONSULTA 3
SELECT turista.*, COUNT (pasaje.id_pasaje) AS cantidadPasajes
FROM TURISTA_USUARIO as turista
LEFT JOIN PASAJE as pasaje
ON turista.id_usuario = pasaje.id_usuario
GROUP BY turista.id_usuario, turista.nombre_usuario, turista.contraseña, turista.fecha_nac, turista.correo, turista.nombre, turista.apellido1, turista.apellido2, turista.id_categoria
HAVING COUNT (pasaje.id_pasaje) > 5;
```

100 % < >

Results Messages

	id_usuario	nombre_usuario	contraseña	fecha_nac	correo	nombre	apellido1	apellido2	id_categoria	cantidadPasajes
1	1	juanperez	password123	1985-05-12	juanp@gmail.com	Juan	Perez	Gomez	1	6
2	4	soyturista	SoyTurista2017	1992-01-15	soyturista@gmail.com	Carlos	Gomez	Rodriguez	1	9

4.3.4. Consulta 4

Listar idpasajero, nombre, apellidos y asiento (idasiento y fila) que correspondan a pasajes comprados para el destino cuyo idviaje es 255.

DQL.sql - DESKTOP-...SLSDHB2\Dell (54))* X DML.sql - DESKTOP-...SLSDHB2\Dell (53))* DDL.sql - DESKTOP-...SLSDHB2\Dell (52))

```
--CONSULTA 4
SELECT turista.Id_Usuario, turista.Nombre, turista.Apellido1, turista.Apellido2, pasaje.Id_Asiento, asiento.fila
FROM TURISTA_USUARIO as turista
JOIN PASAJE as pasaje
ON turista.Id_Usuario = pasaje.Id_Usuario
JOIN ASIENTO as asiento
ON pasaje.Id_Asiento = asiento.Id_Asiento
WHERE pasaje.Id_Destino = 255;
```

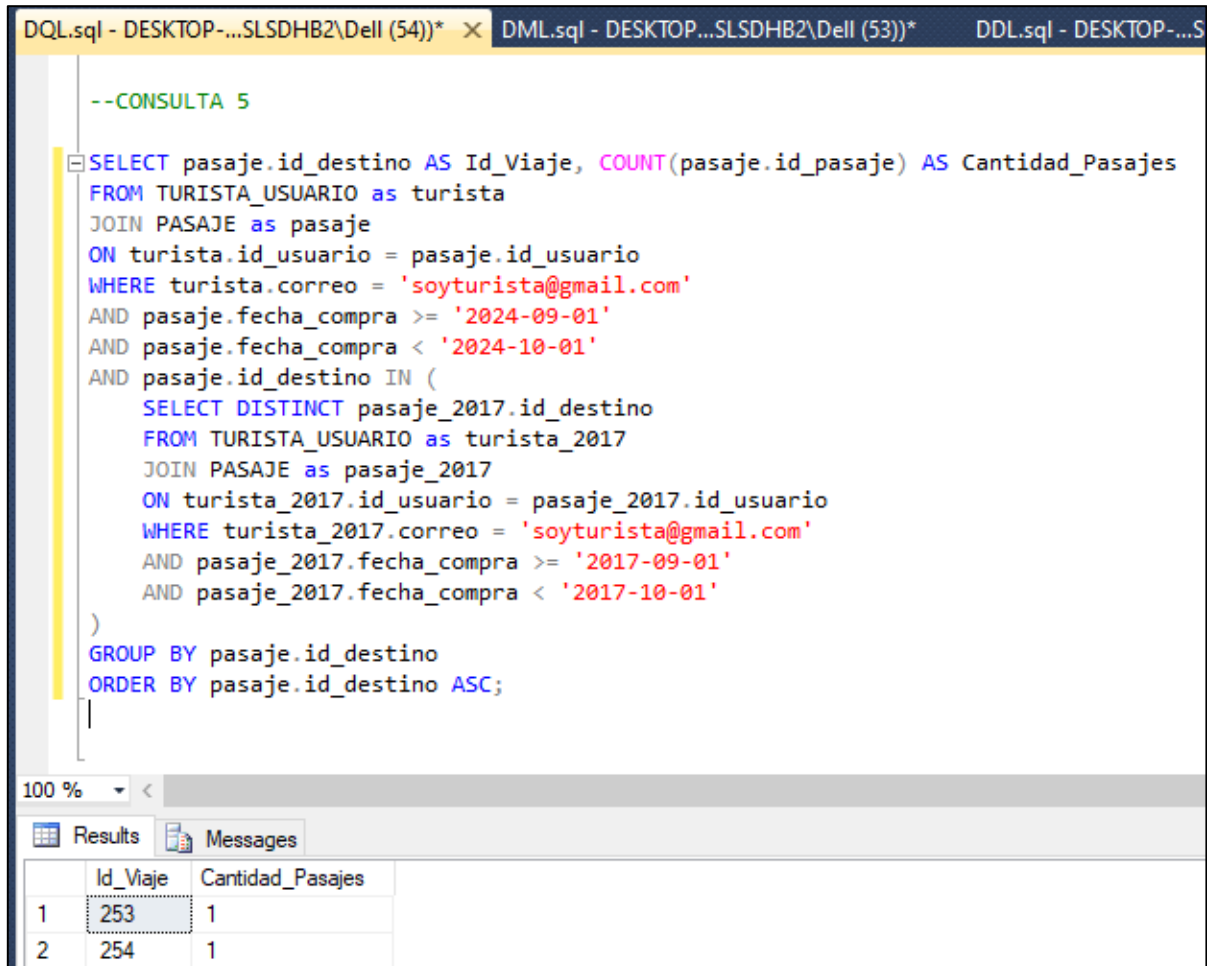
100 % < >

Results Messages

	Id_Usuario	Nombre	Apellido1	Apellido2	Id_Asiento	fila
1	3	Maria	Gonzalez	Sanchez	405	5
2	3	Maria	Gonzalez	Sanchez	406	6
3	1	Juan	Perez	Gomez	406	6

4.3.5. Consulta 5

Listar todos los idviaje y cantidad de pasajes comprados durante el mes de Setiembre de este año para c/u de los destinos del pasajero cuyo correo es soyturista@gmail.com comprados en Setiembre del 2017. La lista debe estar ordenada por idviaje ascendente.



The screenshot shows a SQL IDE with three tabs: DQL.sql, DML.sql, and DDL.sql. The DQL.sql tab is active, displaying a SQL query labeled "--CONSULTA 5". The query is a SELECT statement that joins the TURISTA_USUARIO and PASAJE tables. It filters for users with the email 'soyturista@gmail.com' and for travel dates between September 1st and October 1st of the current year (2024). It also includes a subquery to filter for destinations that were traveled to in September of 2017. The results are grouped by destination and ordered by destination ID in ascending order.

```
--CONSULTA 5
SELECT pasaje.id_destino AS Id_Viaje, COUNT(pasaje.id_pasaje) AS Cantidad_Pasajes
FROM TURISTA_USUARIO as turista
JOIN PASAJE as pasaje
ON turista.id_usuario = pasaje.id_usuario
WHERE turista.correo = 'soyturista@gmail.com'
AND pasaje.fecha_compra >= '2024-09-01'
AND pasaje.fecha_compra < '2024-10-01'
AND pasaje.id_destino IN (
    SELECT DISTINCT pasaje_2017.id_destino
    FROM TURISTA_USUARIO as turista_2017
    JOIN PASAJE as pasaje_2017
    ON turista_2017.id_usuario = pasaje_2017.id_usuario
    WHERE turista_2017.correo = 'soyturista@gmail.com'
    AND pasaje_2017.fecha_compra >= '2017-09-01'
    AND pasaje_2017.fecha_compra < '2017-10-01'
)
GROUP BY pasaje.id_destino
ORDER BY pasaje.id_destino ASC;
```

Below the query editor, the 'Results' tab is active, showing a table with two columns: 'Id_Viaje' and 'Cantidad_Pasajes'. The table contains two rows of data.

	Id_Viaje	Cantidad_Pasajes
1	253	1
2	254	1

Si separamos las letras en año 2024 y 2017, nos quedaría de la siguiente manera:

- Listar los idviaje y cantidad de pasajes comprados durante el mes de Setiembre de este año para c/u de los destinos del pasajero cuyo correo es soyturista@gmail.com - Lista ordenada por idviaje ascendente

```
--CONSULTA 5 B- Listar los idviaje y cantidad de pasajes comprados durante el mes de
```

```

SELECT pasaje.id_destino AS Id_Viaje, COUNT(pasaje.id_pasaje) AS Cantidad_Pasajes
FROM TURISTA_USUARIO AS turista
JOIN PASAJE AS pasaje
ON turista.id_usuario = pasaje.id_usuario
WHERE turista.correo = 'soyturista@gmail.com'
AND pasaje.fecha_compra >= '2024-09-01'
AND pasaje.fecha_compra < '2024-10-01'
GROUP BY pasaje.id_destino
ORDER BY pasaje.id_destino ASC;

```

	Id_Viaje	Cantidad_Pasajes
1	253	1
2	254	1

- Listar los idviaje y cantidad de pasajes comprados durante el mes de Setiembre Del 2017 para c/u de los destinos del pasajero cuyo correo es soyturista@gmail.com - Lista ordenada por idviaje ascendente

```
--CONSULTA 5 C- Listar los idviaje y cantidad de pasajes comprados durante el mes de
```

```

SELECT pasaje.id_destino AS Id_Viaje, COUNT(pasaje.id_pasaje) AS Cantidad_Pasajes
FROM TURISTA_USUARIO AS turista
JOIN PASAJE AS pasaje
ON turista.id_usuario = pasaje.id_usuario
WHERE turista.correo = 'soyturista@gmail.com'
AND pasaje.fecha_compra >= '2017-09-01'
AND pasaje.fecha_compra < '2017-10-01'
GROUP BY pasaje.id_destino
ORDER BY pasaje.id_destino ASC;

```

	Id_Viaje	Cantidad_Pasajes
1	253	1
2	254	1