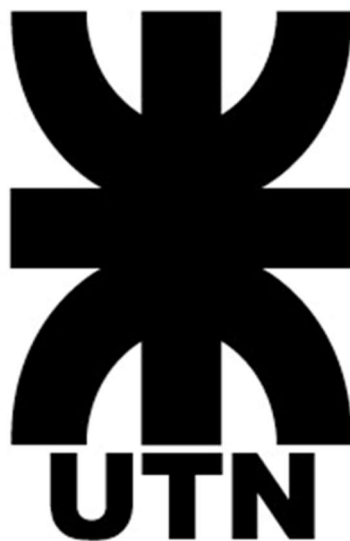


TRABAJO PRACTICO INTEGRADOR



**UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL RESISTENCIA
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN**

DISEÑO Y ADMINISTRACIÓN DE BASE DE DATOS

GRUPO. N°3

DOCENTES. GAONA GERMAN - BARRIOS SOLEDAD

ALUMNO. ALVAREZ BISORDI LUCAS MARTIN

AÑO. 2023

ÍNDICE

Consignas:

- ✓ Escenario. (pág. 3)
- ✓ Funcionalidades. (pág. 3)
- ✓ Formato. (pág. 4)

Modelos:

- ✓ Entidad Relación. (pág. 6)
- ✓ Relacional. (pág. 7)

Consultas SQL:

- ✓ Create. (pág. 8)
- ✓ Insert. (pág. 8)
- ✓ Update. (pág. 9)
- ✓ Delete. (pág. 9)
- ✓ Select. (pág. 10)

Objetos SQL:

- ✓ Tabla. (pág. 17)
- ✓ Procedimiento. (pág. 17)
- ✓ Disparador. (pág. 17)

Anexo:

- ✓ Google Drive. (pág. 18)

CONSIGNAS

Escenario:

- ✓ La empresa de transporte de pasajeros "ViajaPlus" se encuentra en proceso de modernización de su sistema de Ventas y Reservas de pasajes, con el objetivo de ofrecer un servicio más eficiente y conveniente para sus clientes.
- ✓ "ViajaPlus" opera en múltiples itinerarios que conectan diferentes ciudades. Cada itinerario tiene un punto de partida y un punto de destino, como, por ejemplo, el itinerario que va desde Resistencia a Buenos Aires. Dentro de cada itinerario, existen paradas intermedias en diversas ciudades a lo largo del recorrido, como Villa Ocampo, Reconquista, Santa Fe, entre otras.
- ✓ Los itinerarios están predefinidos con horarios de partida y llegada, ciudades de origen y destino, así como puntos intermedios. "ViajaPlus" permite la venta de pasajes para el itinerario completo o para tramos individuales entre cualquiera de sus puntos, siempre y cuando la partida sea anterior al destino dentro del itinerario.
- ✓ Un servicio se refiere a un viaje específico dentro de un itinerario. Cada servicio se caracteriza por incluir el itinerario completo (con todas sus paradas intermedias), una fecha de partida y llegada, un horario de partida y llegada, y una unidad de transporte (colectivo) asignada.
- ✓ Las unidades de transporte son vehículos colectivos con disposiciones de asientos específicas. Existen unidades de uno y dos pisos, y se clasifican en categorías, como común, semicama y coche cama. La calidad de un servicio se determina por la categoría de la unidad y el tipo de atención, ya sea común o ejecutivo. El costo del pasaje varía según la calidad del servicio y el itinerario (o tramo de este).

Funcionalidades:

- ✓ El programador de servicios, responsable de crear y mantener servicios, asigna un itinerario predefinido, una fecha de partida, una unidad de transporte y especifica la calidad del servicio para cada viaje.

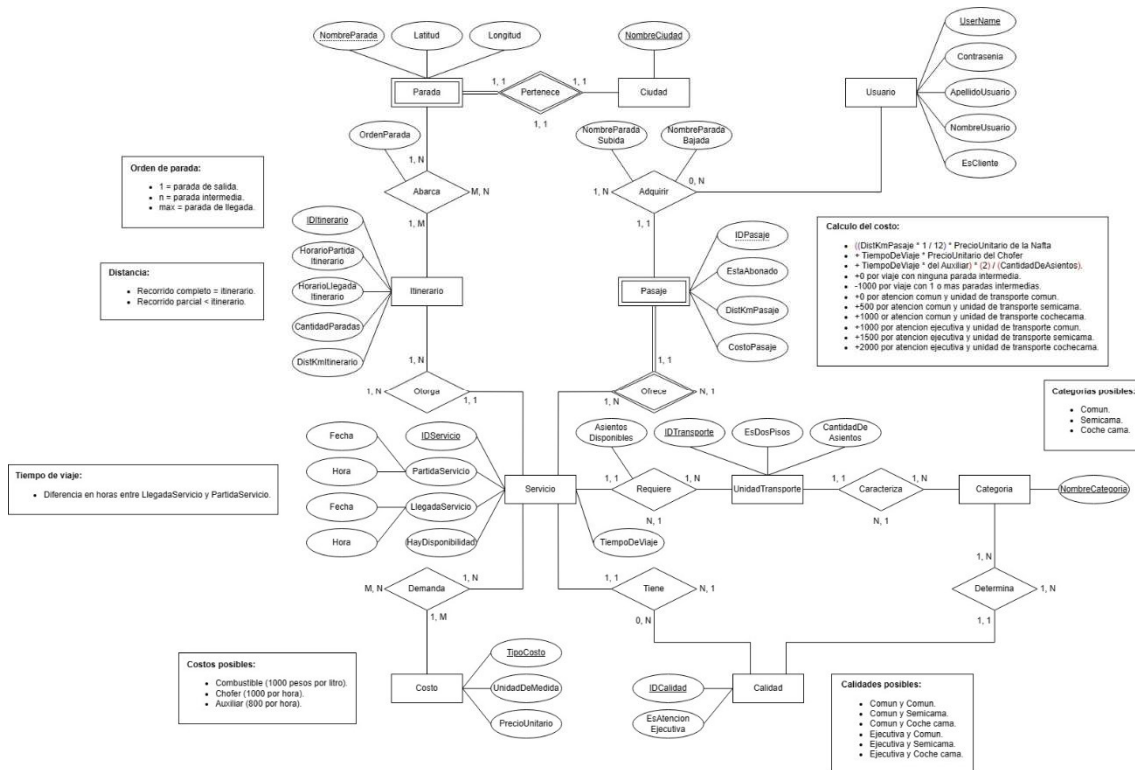
- ✓ El sistema debe permitir la organización de los servicios y la venta y reserva de pasajes. Un pasaje para un itinerario o tramo específico solo puede venderse o reservarse si hay disponibilidad.
- ✓ Las reservas caducan automáticamente treinta minutos antes del horario de partida si la venta no se ha efectuado.
- ✓ El sistema debe incluir las siguientes funciones básicas:
 1. Reserva de Pasajes: Permite a los clientes reservar pasajes para itinerarios o tramos específicos, siempre que haya disponibilidad.
 2. Venta de Pasajes: Facilita la compra de pasajes para itinerarios o tramos en función de la disponibilidad.
 3. Cancelación de Reservas a Pedido: Los clientes pueden cancelar sus reservas antes de la fecha de partida programada.
 4. Cancelación Automática de Reservas por Expiración: Las reservas se cancelan automáticamente si no se efectúa la venta dentro de los treinta minutos previos al horario de partida.
 5. Mantenimiento de Itinerarios: Permite a "ViajaPlus" administrar y actualizar los itinerarios, incluyendo horarios, ciudades y puntos intermedios.
 6. Gestión de Unidades: Facilita el mantenimiento y la gestión de las unidades de transporte, incluyendo su categoría y disponibilidad.
 7. Gestión de Servicios: Permite al programador de servicios asignar itinerarios, fechas, unidades y calidad de servicio para programar los viajes.
 8. Estadísticas de Pasajes Vendidos: Proporciona informes y estadísticas sobre la cantidad de pasajes vendidos en función de diferentes criterios, como itinerarios, fechas y categorías de unidades.
- ✓ Con este sistema mejorado, "ViajaPlus" estará en condiciones de ofrecer un servicio de transporte más eficiente y conveniente para sus clientes, al tiempo que facilita la gestión y el control de sus operaciones internas.

Formato:

- ✓ Elaborar el modelo entidad relación, agregando las entidades, vínculos y atributos que considere necesarios para satisfacer todos los requerimientos enunciados. A partir del anterior, obtenga el modelo relacional.
- ✓ Implemente el modelo relacional en un motor de bases de datos de su elección. Desarrolle una aplicación web para acceder a dicha BD, contando como mínimo con funcionalidad para que:
 - Reserva de pasajes.
 - Mantenimiento de unidades.

MODELOS

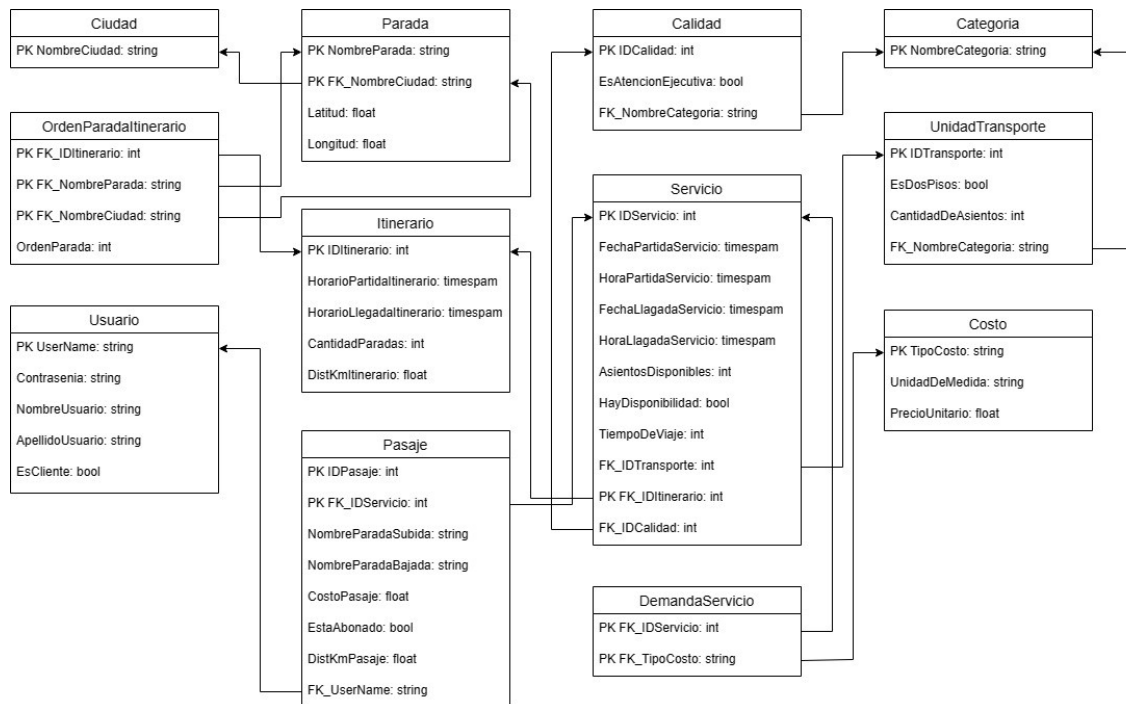
Modelo Entidad Relación:



- ✓ Los cuadros de texto ubicados alrededor del modelo hacen referencia los valores posibles dentro de las relaciones y, en algunos casos, a su interpretación.
- ✓ Ciudad de Origen y Destino no los puse como atributos, debido a que se puede acceder a ellos desde las tablas "OrdenParadaItinerario" y "Parada".
- ✓ Al atributo "CantidadParadas" lo utilizo como medida de seguridad, gracias a este atributo se puede verificar si el administrador cargo todas las paradas correspondientes.
- ✓ A los atributos "HayDisponibilidad", "AsientosDisponibles", "DistKmitinerario", "DistKmPasaje", "TiempoDeViaje" y "CostoPasaje" los tome como atributos simples que se guardan en la BD debido a que considero que son atributos esenciales de la relación, y además permiten ahorrar código a la hora de obtener los mismos utilizando Triggers para la obtención y actualización de los mismos.
- ✓ Pero también se los podría tomar como atributos derivados, calculados en el software y mostrados por pantalla. Inclusive de esta forma no haría falta el atributo "AsientosDisponibles", debido a que la disponibilidad se podría calcular

en base a la cantidad de pasajes y la cantidad de asientos del transporte asociado al servicio.

Modelo Relacional:



CONSULTAS SQL

Create:

- ✓ CREATE TABLE: Plasmado en el Script.
- ✓ CREATE PROCEDURE: Plasmado en el Script.
- ✓ CREATE TRIGGER: Plasmado en el Script.

Insert:

- ✓ FormCrearCuenta: INSERT INTO Usuario (UserName, Contraseña, NombreUsuario, ApellidoUsuario, EsCliente) VALUES ('{txtUsuario.Text}', '{txtContraseña.Text}', '{txtNombre.Text}', '{txtApellido.Text}', '1')
- ✓ FormDemanda: INSERT INTO DemandaServicio (FK_TipoCosto, FK_IDServicio) VALUES ('{comboBoxCG.Text}', '{int.Parse(comboBoxIDSG.Text)}')
- ✓ FormItinerario: INSERT INTO Itinerario (HorarioPartidaItinerario, HorarioLlegadaItinerario, CantidadParadas, DistKmlItinerario) VALUES ('{dateTimePickerSalida.Text}', '{dateTimePickerLlegada.Text}', '{txtCantidad.Text}', '{txtDistancia.Text}')
- ✓ FormOrdenParadaItinerario: INSERT INTO OrdenParadaItinerario (FK_IDItinerario, FK_NombreParada, FK_NombreCiudad, OrdenParada) VALUES ('{IDIG}', '{comboBoxIDPG.Text}', '{comboBoxIDPG.Text.Substring("Parada".Length).Trim()}', '{int.Parse(txtOrdenG.Text)}')
- ✓ FormParada: INSERT INTO Ciudad (NombreCiudad) VALUES ('{txtCiudad.Text}')
- ✓ FormaParada: INSERT INTO Parada (NombreParada, Latitud, Longitud, FK_NombreCiudad) VALUES ('{txtParada.Text}', '{txtLat.Text.Replace(',', '.')}', '{txtLon.Text.Replace(',', '.')}', '{txtCiudad.Text}')
- ✓ FormReserva: INSERT INTO Pasaje (FK_IDServicio, NombreParadaSubida, NombreParadaBajada, EstaAbonado, DistKmPasaje, FK_UserName) VALUES ('{comboBoxIDSV.Text}', '{txtParada.Text}', '{textBoxParadaF.Text}', '{0}', '{dist}', '{FormInicio.user}')
- ✓ FormServicio: INSERT INTO Servicio (FechaPartidaServicio, FechaLlegadaServicio, HoraPartidaServicio, HoraLlegadaServicio, HayDisponibilidad, FK_IDCalidad,


```
FK_IDItinerario, FK_IDTransporte) VALUES ('{dateTimePickerFechaPG.Text}',
'{dateTimePickerFechaLG.Text}', '{horaS}', '{horaL}', '{1}', '{IDCG}', '{IDIG}',
'{IDTG}')
```

- ✓ FormTransporte: INSERT INTO UnidadTransporte (EsDosPisos, CantidadDeAsientos, FK_NombreCategoria) VALUES ('{comboBoxPisos.Text}', '{txtAsientos.Text}', '{IDCG}')

Update:

- ✓ FormAbonar: UPDATE Pasaje SET EstaAbonado = '{1}' WHERE IDPasaje = '{comboBoxPA.Text}'
- ✓ FormItinerario: UPDATE Itinerario SET HorarioPartidaltinerario = '{dateTimePickerSalidaM.Text}', HorarioLlegadaltinerario = '{dateTimePickerLlegadaM.Text}', CantidadParadas = '{txtCantidadM.Text}' WHERE IDItinerario = '{IDM}'
- ✓ FormItinerario: UPDATE Servicio SET HoraPartidaServicio = '{dateTimePickerSalidaM.Text}', HoraLlegadaServicio = '{dateTimePickerLlegadaM.Text}' WHERE FK_IDItinerario = '{IDM}'
- ✓ FormItinerario: UPDATE Itinerario SET DistKmltinerario = '{textBoxDistV.Text.Replace(',','.')}' WHERE IDItinerario = '{IDV}'
- ✓ FormOrdenParadaltinerario: UPDATE OrdenParadaltinerario SET OrdenParada = '{int.Parse(txtOrdenM.Text)}' WHERE FK_IDItinerario = '{IDIM}' AND FK_NombreParada = '{comboBoxIDPM.Text}'
- ✓ FormServicio: UPDATE Servicio SET FechaPartidaServicio = '{dateTimePickerFechaPM.Value.ToShortDateString()}', HoraPartidaServicio = '{dateTimePickerHoraPM.Text}', FechaLlegadaServicio = '{dateTimePickerFechaLM.Value.ToShortDateString()}', HoraLlegadaServicio = '{dateTimePickerHoraLM.Text}', HayDisponibilidad = '{disp}', FK_IDCalidad = '{IDCM}', FK_IDItinerario = '{IDIM}', FK_IDTransporte = '{IDTM}' WHERE IDServicio = '{IDSM}'
- ✓ FormTransporte: UPDATE UnidadTransporte SET EsDosPisos = '{pisos}', CantidadDeAsientos = '{txtAsientosM.Text}', FK_NombreCategoria = '{comboBoxIDCM.Text}' WHERE IDTransporte = '{IDTM}'

- ✓ FormTransporte: UPDATE Servicio SET AsientosDisponibles = AsientosDisponibles + {diferencia}, HayDisponibilidad = CASE WHEN AsientosDisponibles + {diferencia} > 0 THEN '{1}' ELSE '{0}' END WHERE FK_IDTransporte = '{IDTM}'

Delete:

- ✓ FormCancelarReserva: DELETE FROM Pasaje WHERE IDPasaje = '{comboBoxPE.Text}'
- ✓ FormDemanda: DELETE FROM DemandaServicio WHERE FK_IDServicio = '{comboBoxIDSE.Text}' AND FK_TipoCosto = '{comboBoxCE.Text}'
- ✓ FormItinerario: DELETE FROM Pasaje WHERE FK_IDServicio IN (SELECT Servicio.[IDServicio] FROM Itinerario INNER JOIN Servicio ON Servicio.[FK_IDItinerario] = Itinerario.[IDItinerario] WHERE Itinerario.[IDItinerario] = '{ID}')
- ✓ FormItinerario: DELETE FROM Itinerario WHERE IDItinerario = '{ID}'
- ✓ FormOrdenParadaItinerario: DELETE FROM OrdenParadaItinerario WHERE FK_IDItinerario = '{IDIE}' AND FK_NombreParada = '{comboBoxIDPE.Text}'
- ✓ FormParada: DELETE FROM Parada WHERE NombreParada = '{txtParada.Text}'
- ✓ FormParada: DELETE FROM Ciudad WHERE NombreCiudad = '{txtCiudad.Text}'
- ✓ FormServicio: DELETE FROM Servicio WHERE IDServicio = '{IDSE}'
- ✓ FormTransporte: DELETE FROM UnidadTransporte WHERE IDTransporte = '{IDTE}'

Select:

- ✓ FormAbonar: SELECT IDPasaje FROM Pasaje WHERE EstaAbonado = '{0}' AND FK_UserName = '{FormInicio.user}'
- ✓ FormAbonar: SELECT CostoPasaje FROM Pasaje WHERE IDPasaje = '{comboBoxPA.Text}' AND FK_UserName = '{FormInicio.user}'
- ✓ FormCalidad: SELECT IDCalidad FROM Calidad
- ✓ FormCalidad: SELECT EsAtencionEjecutiva, FK_NombreCategoria FROM Calidad WHERE IDCalidad = '{int.Parse(comboBoxIDCV.Text)}'

- ✓ FormCancelarReserva: SELECT IDPasaje FROM Pasaje WHERE EstaAbonado = '{0}' AND FK_UserName = '{FormInicio.user}'
- ✓ FormCategoria: SELECT * FROM Categoria
- ✓ FormConsulta: SELECT Itinerario.[IDItinerario],
COALESCE(COUNT(Pasaje.[IDPasaje]), 0) Pasajes, CAST (100 *
COALESCE(COUNT(Pasaje.[IDPasaje]), 0) / (SELECT COUNT(*) FROM Pasaje) AS
NUMERIC (5, 2)) Porcentaje FROM Itinerario LEFT JOIN Servicio ON
Itinerario.[IDItinerario] = Servicio.[FK_IDItinerario] LEFT JOIN Pasaje ON
Pasaje.[FK_IDServicio] = Servicio.[IDServicio] GROUP BY Itinerario.[IDItinerario]
ORDER BY Pasajes DESC
- ✓ FormCosto: SELECT TipoCosto FROM Costo
- ✓ FormCosto: SELECT * FROM Costo WHERE TipoCosto = '{comboBoxIDCV.Text}'
- ✓ FormDemanda: SELECT IDServicio FROM Servicio
- ✓ FormDemanda: SELECT TipoCosto FROM Costo
- ✓ FormDemanda: SELECT FK_TipoCosto FROM DemandaServicio WHERE
FK_IDServicio = '{comboBoxIDSV.Text}'
- ✓ FormInicio: SELECT UserName, Contraseña, EsCliente FROM Usuario
- ✓ FormItinerario: SELECT IDItinerario FROM Itinerario
- ✓ FormItinerario: SELECT COUNT(*) Contador FROM OrdenParadaItinerario
WHERE FK_IDItinerario = '{IDV}'
- ✓ FormItinerario: SELECT CantidadParadas, HorarioPartidaItinerario,
HorarioLlegadaItinerario FROM Itinerario WHERE IDItinerario = '{IDV}'
- ✓ FormItinerario: SELECT * FROM Itinerario WHERE IDItinerario = '{IDV}'
- ✓ FormItinerario: SELECT P.[FK_NombreCiudad] FROM Parada P INNER JOIN
OrdenParadaItinerario O ON P.[NombreParada] = O.[FK_NombreParada] WHERE
O.[OrdenParada] = (SELECT MIN(OrdenParadaItinerario.[OrdenParada]) FROM
Itinerario INNER JOIN OrdenParadaItinerario ON
OrdenParadaItinerario.[FK_IDItinerario] = Itinerario.[IDItinerario] WHERE
IDItinerario = '{IDV}')) AND O.[FK_IDItinerario] = '{IDV}'
- ✓ FormItinerario: SELECT P.[FK_NombreCiudad] FROM Parada P INNER JOIN
OrdenParadaItinerario O ON P.[NombreParada] = O.[FK_NombreParada] WHERE
O.[OrdenParada] = (SELECT MAX(OrdenParadaItinerario.[OrdenParada]) FROM

- ```

Itinerario INNER JOIN OrdenParadaltinerario ON
OrdenParadaltinerario.[FK_IDItinerario] = Itinerario.[IDItinerario] WHERE
IDItinerario = '{IDV}') AND O.[FK_IDItinerario] = '{IDV}'

```
- ✓ FormItinerario: SELECT Parada.[NombreParada], Parada.[Latitud],  
Parada.[Longitud], OrdenParadaltinerario.[OrdenParada] FROM Parada INNER  
JOIN OrdenParadaltinerario ON Parada.[NombreParada] =  
OrdenParadaltinerario.[FK\_NombreParada] WHERE  
OrdenParadaltinerario.[FK\_IDItinerario] = '{IDV}' ORDER BY  
OrdenParadaltinerario.[OrdenParada]
  - ✓ FormMisPasajes: SELECT Pasaje.[IDPasaje] ID, Pasaje.[NombreParadaSubida]  
Inicio, Pasaje.[NombreParadaBajada] Fin, Pasaje.[EstaAbonado],  
Pasaje.[DistKmPasaje] DistKm, Pasaje.[CostoPasaje] Costo, Servicio.[IDServicio],  
Servicio.[FechaPartidaServicio] FechaPartida, Servicio.[FechaLlegadaServicio]  
FechaLlegada, Servicio.[HoraPartidaServicio] HoraPartida,  
Servicio.[HoraLlegadaServicio] HoraLlegada, Servicio.[TiempoDeViaje],  
Calidad.[EsAtencionEjecutiva], Categoria.[NombreCategoria] Categoria FROM  
Pasaje INNER JOIN Servicio ON Servicio.[IDServicio] = Pasaje.[FK\_IDServicio]  
INNER JOIN Calidad ON Calidad.[IDCalidad] = Servicio.[FK\_IDCalidad] INNER JOIN  
Categoria ON Categoria.[NombreCategoria] = Calidad.[FK\_NombreCategoria]  
WHERE Pasaje.[FK\_UserName] = '{FormInicio.user}'
  - ✓ FormOrdenParadaltinerario: SELECT \* FROM OrdenParadaltinerario WHERE  
FK\_IDItinerario = {id} ORDER BY OrdenParada
  - ✓ FormOrdenParadaltinerario: SELECT IDItinerario FROM Itinerario
  - ✓ FormOrdenParadaltinerario: SELECT NombreParada FROM Parada
  - ✓ FormOrdenParadaltinerario: SELECT COUNT(\*) Contador FROM  
OrdenParadaltinerario WHERE FK\_IDItinerario = '{IDIG}'
  - ✓ FormOrdenParadaltinerario: SELECT CantidadParadas FROM Itinerario WHERE  
IDItinerario = '{IDIG}'
  - ✓ FormParada: SELECT \* FROM Parada
  - ✓ FormParada: SELECT COALESCE(COUNT(NombreCiudad), 0) Cantidad FROM  
Ciudad WHERE NombreCiudad = '{txtCiudad.Text}'

- ✓ FormParada: SELECT COALESCE(COUNT(NombreCiudad), 0) Cantidad FROM Ciudad WHERE NombreCiudad = '{txtCiudad.Text}'
- ✓ FormParada: SELECT \* FROM Parada WHERE NombreParada = '{nombreParada}'
- ✓ FormPasajePorCategoria: SELECT Categoria.[NombreCategoria], COALESCE(COUNT(Pasaje.[IDPasaje]), 0) Pasajes FROM Categoria LEFT JOIN UnidadTransporte ON Categoria.[NombreCategoria] = UnidadTransporte.[FK\_NombreCategoria] LEFT JOIN Servicio ON UnidadTransporte.[IDTransporte] = Servicio.[FK\_IDTransporte] LEFT JOIN Pasaje ON Pasaje.[FK\_IDServicio] = Servicio.[IDServicio] GROUP BY Categoria.[NombreCategoria] ORDER BY Pasajes DESC
- ✓ FormPasajePorFecha: SELECT Servicio.[IDServicio], Servicio.[FechaPartidaServicio], COALESCE(COUNT(Pasaje.[IDPasaje]), 0) Pasajes FROM Servicio LEFT JOIN Pasaje ON Pasaje.[FK\_IDServicio] = Servicio.[IDServicio] GROUP BY Servicio.[IDServicio], Servicio.[FechaPartidaServicio] ORDER BY Pasajes DESC
- ✓ FormPasajePorItinerario: SELECT Itinerario.[IDItinerario], COALESCE(COUNT(Pasaje.[IDPasaje]), 0) Pasajes FROM Itinerario LEFT JOIN Servicio ON Itinerario.[IDItinerario] = Servicio.[FK\_IDItinerario] LEFT JOIN Pasaje ON Pasaje.[FK\_IDServicio] = Servicio.[IDServicio] GROUP BY Itinerario.[IDItinerario] ORDER BY Pasajes DESC
- ✓ FormReservar: SELECT \* FROM Ciudad
- ✓ FormReservar: SELECT COUNT(\*) Contador FROM OrdenParadaltinerario WHERE FK\_IDItinerario = '{IDV}'
- ✓ FormReservar: SELECT CantidadParadas, HorarioPartidaltinerario, HorarioLlegadaltinerario FROM Itinerario WHERE IDItinerario = '{IDV}'
- ✓ FormReservar: SELECT \* FROM Itinerario WHERE IDItinerario = '{IDV}'
- ✓ FormReservar: SELECT Parada.[NombreParada], Parada.[Latitud], Parada.[Longitud], OrdenParadaltinerario.[OrdenParada] FROM Parada INNER JOIN OrdenParadaltinerario ON Parada.[NombreParada] = OrdenParadaltinerario.[FK\_NombreParada] WHERE OrdenParadaltinerario.[FK\_IDItinerario] = '{IDV}' ORDER BY OrdenParadaltinerario.[OrdenParada]

- ✓ FormReservar:               SELECT                       Servicio.[FechaLlegadaServicio],  
Servicio.[FechaPartidaServicio],   Servicio.[TiempoDeViaje],   CASE   WHEN  
Calidad.[EsAtencionEjecutiva] = 0 THEN 'Comun' ELSE 'Ejecutiva' END Atencion,  
Categoria.[NombreCategoria]   FROM   Servicio   INNER   JOIN   Calidad   ON  
Servicio.[FK\_IDCalidad] = Calidad.[IDCalidad]   INNER   JOIN   Categoria   ON  
Categoria.[NombreCategoria]   =   Calidad.[FK\_NombreCategoria]   WHERE  
Servicio.[IDServicio] = '{comboBoxIDSV.Text}'
- ✓ FormReservar: SELECT AsientosDisponibles FROM Servicio WHERE IDServicio =  
'{comboBoxIDSV.Text}'
- ✓ FormReservar:   SELECT   \*   FROM   Parada   WHERE   NombreParada   =     
'{nombreParada}'
- ✓ FormReservar:   SELECT   \*   FROM   Parada   WHERE   NombreParada   =     
'{nombreParada}'
- ✓ FormReservar:   SELECT   I.[IDItinerario]   FROM   Itinerario   I   INNER   JOIN  
OrdenParadaltinerario   O   ON   O.[FK\_IDItinerario] = I.[IDItinerario]   INNER   JOIN  
Parada   P   ON   P.[NombreParada]   =   O.[FK\_NombreParada]   WHERE  
O.[OrdenParada] = (SELECT MAX(OrdenParadaltinerario.[OrdenParada]) FROM  
OrdenParadaltinerario   WHERE   OrdenParadaltinerario.[FK\_IDItinerario]   =     
I.[IDItinerario]) AND P.[FK\_NombreCiudad] = '{comboBoxDestino.Text}'
- ✓ FormReservar:   SELECT   OrdenParada   FROM   OrdenParadaltinerario   WHERE  
FK\_NombreParada = '{txtParada.Text}' AND FK\_IDItinerario = '{IDV}'
- ✓ FormReservar:   SELECT   OrdenParada   FROM   OrdenParadaltinerario   WHERE  
FK\_NombreParada = '{textBoxParadaF.Text}' AND FK\_IDItinerario = '{IDV}'
- ✓ FormReservar:   SELECT   COUNT(\*)   ParadasParciales   FROM   OrdenParadaltinerario  
WHERE   OrdenParada >= '{fini}' AND   OrdenParada <= '{ffin}'
- ✓ FormReservar:   SELECT   Parada.[NombreParada],   Parada.[Latitud],  
Parada.[Longitud],   OrdenParadaltinerario.[OrdenParada]   FROM   Parada   INNER  
JOIN   OrdenParadaltinerario   ON   Parada.[NombreParada]   =     
OrdenParadaltinerario.[FK\_NombreParada]                                       WHERE  
OrdenParadaltinerario.[FK\_IDItinerario]                       =               '{IDV}'               AND  
OrdenParadaltinerario.[OrdenParada]                       >=               '{fini}'               AND

- OrdenParadaltinerario.[OrdenParada] <= '{ffin}' ORDER BY  
OrdenParadaltinerario.[OrdenParada]
- ✓ FormReservar: SELECT IDServicio FROM Servicio WHERE FK\_IDItinerario = '{IDV}'  
AND HayDisponibilidad = '{1}'
  - ✓ FormServicio: SELECT IDServicio FROM Servicio
  - ✓ FormServicio: SELECT IDItinerario FROM Itinerario
  - ✓ FormServicio: SELECT IDTransporte FROM UnidadTransporte
  - ✓ FormServicio: SELECT IDCalidad FROM Calidad
  - ✓ FormServicio: SELECT HorarioPartidaltinerario, HorarioLlegadaltinerario FROM  
Itinerario WHERE IDItinerario = '{IDIG}'
  - ✓ FormServicio: SELECT \* FROM Servicio WHERE IDServicio = '{IDSV}'
  - ✓ FormServicioPorItinerario: SELECT Itinerario.[IDItinerario],  
Itinerario.[CantidadParadas], Itinerario.[DistKmlItinerario] DistKm, ( SELECT  
P.[FK\_NombreCiudad] FROM Parada P INNER JOIN OrdenParadaltinerario O ON  
P.[NombreParada] = O.[FK\_NombreParada] WHERE O.[OrdenParada] = ( SELECT  
MIN(OrdenParadaltinerario.[OrdenParada]) FROM OrdenParadaltinerario  
WHERE OrdenParadaltinerario.[FK\_IDItinerario] = Itinerario.[IDItinerario]) AND  
O.[FK\_IDItinerario] = Itinerario.[IDItinerario]) Origen, ( SELECT  
P.[FK\_NombreCiudad] FROM Parada P INNER JOIN OrdenParadaltinerario O ON  
P.[NombreParada] = O.[FK\_NombreParada] WHERE O.[OrdenParada] = ( SELECT  
MAX(OrdenParadaltinerario.[OrdenParada]) FROM OrdenParadaltinerario  
WHERE OrdenParadaltinerario.[FK\_IDItinerario] = Itinerario.[IDItinerario]) AND  
O.[FK\_IDItinerario] = Itinerario.[IDItinerario]) Destino, Servicio.[IDServicio],  
Servicio.[FechaPartidaServicio] FechaPartida, Servicio.[FechaLlegadaServicio]  
FechaLlegada, Servicio.[HoraPartidaServicio] HoraPartida,  
Servicio.[HoraLlegadaServicio] HoraLlegada, Servicio.[TiempoDeViaje], CASE  
WHEN Calidad.[EsAtencionEjecutiva] = 1 THEN 'Ejecutiva' WHEN  
Calidad.[EsAtencionEjecutiva] = 0 THEN 'Comun' ELSE " END Atencion,  
Categoria.[NombreCategoria] Categoria, COALESCE(COUNT(Pasaje.[IDPasaje]),  
0) Pasajes FROM Itinerario FULL JOIN Servicio ON Servicio.[FK\_IDItinerario] =  
Itinerario.[IDItinerario] LEFT JOIN Calidad ON Calidad.[IDCalidad] =  
Servicio.[FK\_IDCalidad] LEFT JOIN Categoria ON Categoria.[NombreCategoria] =

```
Calidad.[FK_NombreCategoria] LEFT JOIN Pasaje ON Pasaje.[FK_IDServicio] =
Servicio.[IDServicio] GROUP BY Itinerario.[IDItinerario],
Itinerario.[CantidadParadas], Itinerario.[DistKmlItinerario], Servicio.[IDServicio],
Servicio.[FechaPartidaServicio], Servicio.[FechaLlegadaServicio],
Servicio.[HoraPartidaServicio], Servicio.[HoraLlegadaServicio],
Servicio.[TiempoDeViaje], CASE WHEN Calidad.[EsAtencionEjecutiva] = 1 THEN
'Ejecutiva' WHEN Calidad.[EsAtencionEjecutiva] = 0 THEN 'Comun' ELSE '' END,
Categoria.[NombreCategoria] ORDER BY Itinerario.[IDItinerario]
```

- ✓ FormTransporte: SELECT IDTransporte FROM UnidadTransporte
- ✓ FormTransporte: SELECT NombreCategoria FROM Categoria
- ✓ FormTransporte: SELECT CantidadDeAsientos FROM UnidadTransporte WHERE  
IDTransporte = '{IDTM}'
- ✓ FormTransporte: SELECT \* FROM UnidadTransporte WHERE IDTransporte =  
'{IDTV}'



## OBJETOS SQL

### Tabla:

- ✓ Usuario
- ✓ Ciudad
- ✓ Parada
- ✓ Itinerario
- ✓ OrdenParadaItinerario
- ✓ Categoria
- ✓ UnidadTransporte
- ✓ Calidad
- ✓ Costo
- ✓ Servicio
- ✓ DemandaServicio
- ✓ Pasaje

### Procedimiento:

- ✓ EliminarPasajesNoAbonados

### Disparador:

- ✓ DML After Insert Pasaje: ActualizarCostoPasaje
- ✓ DML After Delete Pasaje: SumarAsientoPasaje
- ✓ DML After Insert Pasaje: DescuentoAsientoPasaje
- ✓ DML After Insert Servicio: ActualizarAsientosDisponibles
- ✓ DML Instead Of Delete Parada: InsteadOfDeleteParada
- ✓ DML Instead Of Delete Itinerario: InsteadOfDeleteItinerario
- ✓ DML Instead Of Delete Servicio: InsteadOfDeleteServicio
- ✓ DML Instead Of Delete Transporte: InsteadOfDeleteUnidadTransporte

## ANEXO

### Link de Google Drive:

- ✓ [https://drive.google.com/drive/folders/1\\_S3nfz1hZg\\_a4FZgatZHOvUgaaW5ItCw?usp=sharing](https://drive.google.com/drive/folders/1_S3nfz1hZg_a4FZgatZHOvUgaaW5ItCw?usp=sharing)