

Gestión de Datos

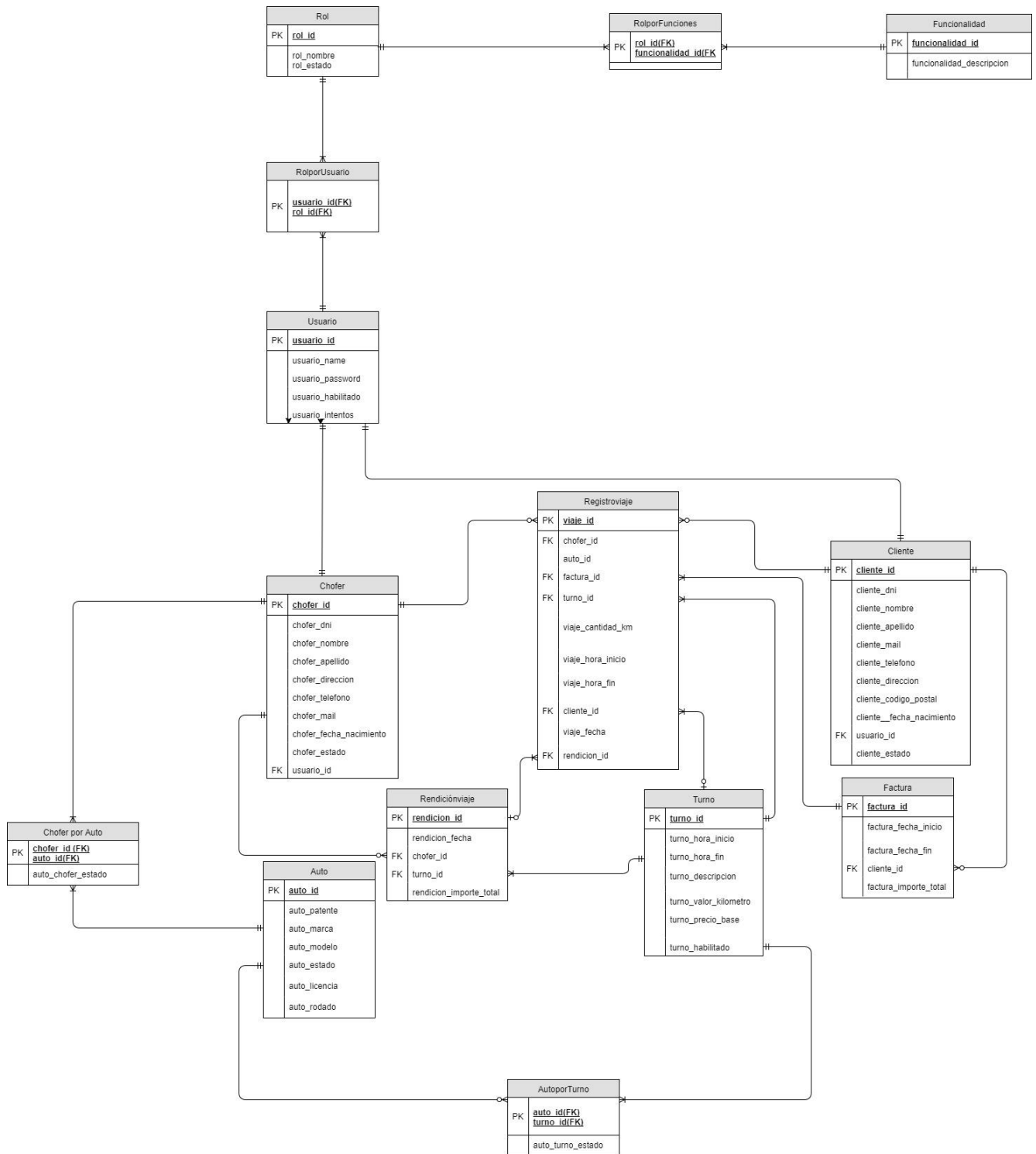
Trabajo Práctico "UberFrba" (1C2017)

Grupo N° 12 "Push_it_to_the_limit"

Integrantes:

- Amoroso, Lucas
- Ezeiza, Brian
- Marussi Lloret, Juan Martin
- Nobile, Nicolas

Diagrama de Entidad Relación:



Descripción de las entidades del DER:

● **Funcionalidad:**

En esta entidad se almacenan todas las funcionalidades que se pueden realizar dependiendo del rol del usuario.

Esta entidad posee los siguientes campos:

- ❖ funcionalidad_id : INT – identity(1,1) – Primary Key
- ❖ funcionalidad_descripcion: varchar(100) - NOT NULL

Funcionalidades disponibles:

- ABM de Rol
- ABM de Clientes
- ABM de Automóviles
- ABM de Turnos
- ABM de Choferes
- Registro de Viajes
- Rendición de Viajes
- Facturación de Clientes
- Listado Estadístico

Consideraciones:

1. La funcionalidad de registro de usuario no se verá almacenada en esta entidad, ya que fue agregada en el formulario de login de los usuarios.

● **Rol:**

En esta funcionalidad se almacenarán los distintos roles utilizados por los usuarios del sistema y si los mismos están habilitados o no para su uso.

Cuenta con los siguientes campos:

- ❖ rol_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ rol_nombre: VARCHAR(100) NOT NULL
- ❖ rol_estado: BIT NOT NULL DEFAULT 1

Roles del Sistema:

- Administrativo
- Cliente
- Chofer

Consideraciones:

1. El campo “**rol_estado**” se utiliza para comprobar si el rol está habilitado para ser asignado a un usuario (Al ser de tipo bit solo acepta los valores 1 o 0).

● Rol por Funciones:

Funciona como intermediario entre las entidades Funcionalidad y Rol debido a la relación de muchos a muchos entre estas tablas.

Como caso ejemplo, los roles cliente y chofer poseen el “Listado Estadístico” y el rol Administrativo tiene varios roles.

Esta entidad posee los siguientes campos:

- ❖ rol_id: INTEGER – NOT NULL
- ❖ funcionalidad_id: INTEGER – NOT NULL
- ❖ Los dos campos forman la PK

Consideraciones:

1. Esta tabla cuenta con los campos “rol_id” y “funcionalidad_id” que serán foreign key de las tablas “Rol” y “Funcionalidad”.

● Usuario:

Contiene a todos los usuarios que pueden utilizar el sistema.

Desde el script inicial se incorpora un usuario Administrador, cuyo nombre es “admin” y su contraseña es “w23e”.

Los demás usuarios tendrán un usuario_id con el cual se los identificará, además de nombre de usuario y contraseña (ambos coincidirán con el documento de quien se registre) la cual será encriptada con el algoritmo SHA256 y almacenada en la base de datos.

Usaremos también el campo “usuario_intentos” para registrar la cantidad de intentos fallidos con los que se intentó ingresar con ese nombre de usuario. Si algún usuario excediera los 3 intentos, se procederá a deshabilitarlo modificando el campo “usuario_habilitado”.

Cuenta con los siguientes campos:

- ❖ usuario_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ usuario_name: VARCHAR(255) UNIQUE NOT NULL
- ❖ usuario_password: NVARCHAR(255) NOT NULL
- ❖ usuario_habilitado: [BIT] NOT NULL DEFAULT 1
- ❖ usuario_intentos: [TINYINT] DEFAULT 0

● Rol por Usuario:

Funcionará intermediario entre las entidades Usuario y Rol dado la relación de muchos a muchos entre esas tablas. Debido a que un Usuario puede tener más de 1 rol a la vez y un Rol puede estar presente en más de un usuario a la vez.

Cuenta con los siguientes campos:

- ❖ usuario_id: INTEGER
- ❖ rol_id: INTEGER
- ❖ Ambos campos forman la PK

Esta tabla cuenta con los campos “rol_id” y “usuario_id” que son foreign key de las tablas “Rol” y “Usuario” y ambas forman una primary key compuesta.

• Cliente:

Contendrá a todos los clientes del sistema. Tendrá un ID para identificar a cada cliente además de una referencia al ID del usuario correspondiente.

Cuenta con los siguientes campos:

- ❖ cliente_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ cliente_nombre: VARCHAR(255) NOT NULL
- ❖ cliente_apellido: VARCHAR(255) NOT NULL
- ❖ cliente_mail: VARCHAR(255)
- ❖ cliente_telefono: NUMERIC(18,0) UNIQUE NOT NULL
- ❖ cliente_direccion: VARCHAR(255) NOT NULL
- ❖ cliente_codigo_postal: INT
- ❖ cliente_fecha_nacimiento: DATETIME NOT NULL
- ❖ cliente_dni: NUMERIC(18,0) UNIQUE NOT NULL
- ❖ usuario_id: INT NOT NULL
- ❖ cliente_estado: BIT NOT NULL DEFAULT 1

Consideraciones:

1. El campo “**usuario_id**” es foreign key de la tabla “**Usuario**” ya que un cliente es un usuario.
2. Debido a que los números de teléfono son únicos para cada cliente (campo con atributo UNIQUE), no pueden existir clientes con el mismo número de teléfono.
Tenemos la misma situación con el DNI.
3. El campo “**cliente_estado**” se utiliza para comprobar si el cliente está habilitado, pudiendo realizar viajes, o si el mismo esta deshabilitado (Al ser de tipo bit solo acepta los valores 1 o 0).

• Chofer:

Contendrá todos los choferes del sistema.

Posee los distintos campos:

- ❖ chofer_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ chofer_dni: NUMERIC(18,0) UNIQUE NOT NULL
- ❖ chofer_nombre: [VARCHAR](255) NOT NULL
- ❖ chofer_apellido: [VARCHAR](255) NOT NULL
- ❖ chofer_direccion: [VARCHAR](255)
- ❖ chofer_telefono: NUMERIC(18,0) UNIQUE
- ❖ chofer_mail: VARCHAR(50)
- ❖ chofer_fecha_Nacimiento: DATETIME NOT NULL
- ❖ chofer_estado: [bit] NOT NULL DEFAULT 1
- ❖ usuario_id: INT

Consideraciones:

1. El campo “**usuario_id**” es foreign key de la tabla “**Usuario**” ya que un chofer es un usuario.
2. El campo “**chofer_estado**” se utiliza para comprobar si el chofer está habilitado, pudiendo realizar viajes y rendiciones , o si el mismo esta deshabilitado (Al ser de tipo bit solo acepta los valores 1 o 0).
3. Al igual que “**Cliente**” los números de teléfono y los DNI son únicos para cada chofer.

• Auto:

Contendrá todos los autos del sistema que serán utilizados en los viajes.

Cuenta con los siguientes campos:

- ❖ auto_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ auto_patente: VARCHAR(10) UNIQUE NOT NULL
- ❖ auto_marca: VARCHAR(255) NOT NULL
- ❖ auto_modelo: VARCHAR(255) NOT NULL
- ❖ auto_estado: BIT not null DEFAULT 1
- ❖ auto_licencia: VARCHAR(255)
- ❖ auto_rodado: VARCHAR(10)

Consideraciones:

1. Debido a que en el sistema no pueden existir autos con la misma patente (autos mellizos) se le asigna el atributo UNIQUE a “**auto_patente**”.

• Chofer por Auto:

Funcionará intermediario entre las entidades Chofer y Auto por la relación de muchos a muchos entre esas tablas. Debido a que un Chofer puede haber usado más de 1 Auto para un viaje y un Auto puede haber sido utilizado por más de un Chofer.

Cuenta con los siguientes campos:

- ❖ chofer_id; INT
- ❖ auto_id: INT
- ❖ auto_chofer_estado: BIT not null DEFAULT 1

Consideraciones:

1. Esta entidad cuenta con los campos “**chofer_id**” y “**auto_id**” que son foreign key de las tablas “**Chofer**” y “**Auto**” y ambas forman una primary key compuesta.
2. “**auto_chofer_estado**” es un campo utilizado para verificar que el auto tenga asignado un chofer (Al ser de tipo bit solo acepta los valores 1 o 0).

• Turno:

Contendrá todos los turnos del sistema en los que se realizarán los viajes.

Cuenta con los siguientes campos:

- ❖ turno_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ turno_hora_inicio: NUMERIC(18,0) NOT NULL
- ❖ turno_hora_fin: NUMERIC(18,0) NOT NULL
- ❖ turno_descripcion: VARCHAR(255)
- ❖ turno_valor_kilometro: NUMERIC(18,2) NOT NULL
- ❖ turno_precio_base: NUMERIC(18,2) NOT NULL
- ❖ turno_habilitado: BIT NOT NULL DEFAULT 1

Consideraciones:

1. Utilizamos un campo “**turno_habilitado**” debido a que si el sistema posee turnos que se superponen solo 1 de ellos va a estar habilitado en un instante dado.

• Auto por Turno:

Funcionará intermediario entre las entidades Auto y Turno por la relación de muchos a muchos entre esas tablas. Debido a que un Auto puede haber usado más de 1 Turno para un viaje y en un Turno puede haber sido utilizado más de un Auto.

Cuenta con los siguientes campos:

- ❖ auto_id: int
- ❖ turno_id: INTEGER
- ❖ auto_turno_estado: BIT not null DEFAULT 1

Consideraciones:

1. Esta entidad cuenta con los campos “**auto_id**” y “**turno_id**” que son foreign key de las tablas “**Auto**” y “**Turno**” y ambas forman una primary key compuesta.
2. “**auto_turno_estado**” es un campo utilizado para verificar que el auto tenga asignado un turno (Al ser de tipo bit solo acepta los valores 1 o 0).

• Factura:

Esta tabla contiene todos los datos pertinentes a la factura emitida a un cliente.

Cuenta con los campos:

- ❖ factura_id: numeric(18,0) IDENTITY(1,1) PRIMARY KEY
- ❖ factura_fecha_inicio: DATETIME
- ❖ factura_fecha_fin: DATETIME
- ❖ cliente_id: INT NOT NULL
- ❖ factura_importe_total: NUMERIC(18,2)

Consideraciones:

1. El campo “**cliente_id**” es foreign key que relaciona esta tabla con la tabla “**Cliente**”

• Rendición Viaje:

Esta tabla contiene todos los datos pertinentes a la rendición emitida a un chofer.

Cuenta con los campos:

- ❖ rendicion_id: numeric(18,0) IDENTITY(1,1) PRIMARY KEY
- ❖ rendicion_fecha: DATETIME NOT NULL
- ❖ chofer_id: int NOT NULL
- ❖ turno_id: int NOT NULL
- ❖ rendicion_importe_total: NUMERIC(18,2) NOT NULL

Consideraciones:

1. El campo “**chofer_id**” es foreign key ya que relaciona con la entidad “**Chofer**” ya que se le hace la rendición a un chofer determinado y “**turno_id**” también es foreign key ya que relaciona con la entidad “**Turno**” dado que se le hace la rendición por el turno.

• Registro Viaje:

Contiene toda información sobre los viajes que se realizaron entre clientes y choferes en un turno determinado.

Cuenta con los siguientes campos:

- ❖ viaje_id: INT IDENTITY(1,1) PRIMARY KEY
- ❖ chofer_id: INT NOT NULL
- ❖ auto_id: INT NOT NULL,
- ❖ factura_id: NUMERIC(18,0)
- ❖ turno_id: INT NOT NULL
- ❖ viaje_cantidad_km: NUMERIC(18,0) NOT NULL
- ❖ rendicion_id: NUMERIC(18,0)
- ❖ viaje_fecha: DATETIME NOT NULL
- ❖ viaje_hora_inicio: TIME(0)
- ❖ viaje_hora_fin: TIME(0)
- ❖ cliente_id: INT NOT NULL

Consideraciones:

1. El campo “**chofer_id**” es foreign key, relaciona con la entidad “**Chofer**” debido a que un chofer determinado realiza el viaje.
 El campo “**factura_id**” es foreign key, relaciona con la entidad “**Facutra**” ya que al cliente se le va a facturar este viaje.
 El campo “**turno_id**” es foreign key, relaciona con la entidad “**Turno**” debido que el viaje se realiza en un turno determinado.
 El campo “**rendicion_id**” es foreign key, relaciona con la entidad “**Rendicion**” porque al chofer se le hace la rendición por el viaje realizado.
 El campo “**cliente_id**” es foreign key, relaciona con la entidad “**Cliente**” debido a que un cliente determinado realiza el viaje.
 El campo “**auto_id**” es foreign key, relaciona con la entidad “**Auto**” debido a que un auto determinado realiza el viaje.
2. “**viaje_hora_fin**” acepta valores de tipo NULL ya que en la tabla maestra esta columna está vacía.

Triggers utilizado:

insertar_turno:

Este trigger se dispara después de insertar un turno y tiene varias funcionalidades con respecto a verificar los horarios del turno ingresado

1. Sus horarios no se superpongan con la de otro turno activo, de superponerse los horarios se realiza un rollback dejando todo como estaba.
2. No tenga exactamente los mismos datos que un turno ya presente en el sistema, ya sea un turno habilitado o deshabilitado.
3. Sus horarios sean valores coherentes.

Si no se pudiera agregar el turno el trigger lanzara una excepción.

APLICACIÓN DESKTOP

Consideración General:

Todas las fechas que se pueden ingresar están limitadas a la fecha establecida en el appConfig, la cual es el 30/06/2017. Para que se pueda probar el sistema hasta un mes completo.

Consideraciones Login:

En la pantalla del login se contempla la funcionalidad de registrar un usuario ya sea un chofer o un cliente, una vez registrado ya puede ingresar a la aplicación con el usuario creado.

1. Los usuarios migrados tanto sean clientes o choferes como los agregados desde la ABM Cliente o ABM Chofer tendrán :
 - **Username : DNI**
 - **Password : DNI**
2. Los usuarios registrados desde la pantalla de login (botón Registrarse) tendrán como username y password los ingresados por pantalla.
3. EL admin se registra con usuario admin y contraseña w23e.

Consideraciones ABMs :

- **ABM ROL**

1: Editar rol

En esta opción aparece un dataGridView con todos los roles existentes, habilitados y deshabilitados. Cuando se aprieta el botón "editar" se permite modificar los datos y solamente en el caso que el rol elegido estuviese deshabilitado se mostrara en la misma pantalla de modificar datos un Checkbox "Habilitado" para poder volver a habilitarlo.

2: Eliminar Rol

En el combobox Rol solo se muestran los roles habilitados es decir que tiene el campo rol_estado de la tabla rol en 1.

Esta opción es la única forma de deshabilitar un rol.

3: Asignar Roles

En esta pantalla se muestra un combobox usuario, que se carga automáticamente con el nombre de los usuarios que estén habilitados, esto significa que tienen el campo usuario_habilitado de la tabla Usuario en 1.

En el checkBoxList que se muestra se cargan automáticamente todos los roles que estén habilitados, es decir que tengan el campo rol_estado de la tabla Rol en 1.

De los roles que se muestran en el checkBoxlist, cuando se selecciona un usuario, se tildan los que este posee.

- **ABM CLIENTE**

1: Agregar Cliente

Al agregar un nuevo cliente también se le crea un usuario al cual se le establece como usuario y

contraseña el DNI.

No se permite crear un cliente con un DNI ya existente.

2: Editar Cliente

En esta opción se muestra un dataGridView con los datos del cliente .En dicha tabla aparece un checkbox que representa el estado del cliente, aparecerá desmarcado o no dependiendo del campo cliente_estado en la tabla cliente.

Un cliente puede deshabilitarse de dos maneras, destildando el checkbox que aparece en la pantalla que se despliega al tocar el botón modificar de la tabla y apretando guardar o con el botón eliminar de la tabla .En ambos casos se establece el campo cliente_estado de la tabla Cliente en 0.

Para poder habitar un cliente se tiene que hacer click en el botón modificar y tildar el checkbox "Habilitado" de la pantalla Editar Clientes , al tocar en guardar se establecerá en 1 el campo cliente_estado de la tabla Cliente.

Si cuando se modifica el DNI de un cliente este tenía como usuario y contraseña su DNI ,se le actualiza su usuario y contraseña al nuevo DNI ingresado.

• **ABM AUTOMOVÍL**

Cuestiones a tener en cuenta:

- Un auto puede tener varios turnos, habilitados o deshabilitados.
- Un auto puede tener varios choferes.
- Un chofer no puede tener más de un auto activo (puede tener más pero el resto tienen que estar deshabilitados para ese chofer).
- Rodado y licencia se migran pero no se ponen en la pantalla de editar o agregar auto por que el enunciado no lo pide.

1: Agregar Auto

En esta opción se muestra una pantalla para agregar un nuevo auto , en la cual se aparece un combobox marca que se carga automaticamente con una lista acotada , un combobox turno que se carga automaticamente con todos los turnos existentes (Habilitados o no),y un combobox chofer que se carga automaticamente con los choferes existentes que no posean un auto activo.

2: Editar Automóvil

-A) Si al editar un automóvil en el campo chofer se ingresa un DNI, si el DNI ingresado es el mismo que el anterior, se habilita el chofer para ese auto en caso de que estuviese deshabilitado el chofer para ese auto(es decir que si la fila auto_id, chofer_id, chofer_auto_estado tuviese el ultimo campo en cero, lo pone en uno)

-B) Si el DNI ingresado es distinto al que estaba, se deshabilita el chofer anterior para este auto (poniendo "auto_id, chofer_id, chofer_auto_estado=0 ", en cero el ultimo campo en la tabla ChoferporAuto) y además se ingresa en la tabla choferporAuto el nuevo chofer para ese auto.

-C) Tener en cuenta que cada vez que se intenta agregar un chofer a un auto, tanto sea habilitando uno que ya existía o asignando uno nuevo, se valida que el chofer ya no tenga un coche activo (es decir que el auto este activo en la tabla Auto y en choferporAuto)

- D) Cuando se le intenta agregar un auto a un chofer que ya tiene uno activo muestra un mensaje de error y no deja guardar cualquier cambio al auto.
- E) Es importante entender que ,por ejemplo, cuando solamente le modifiko la marca a un auto, al hacer click en guardar , se va a intentar habilitar ese chofer a ese auto (no importa si no lo modifique) y por ahí no me deja porque el chofer ya tiene otro coche activo.
- F) Para modificar un turno se puede hacer desde el botón que aparece en esta pantalla que dice “modificar Turnos Actuales” que despliega otra pantalla donde se puede desactivar los turnos de ese auto ,es decir le pone en cero la columna turno_auto_estado de la tabla turnoPorAuto.
- G) Si no se quiere agregar ningún turno nuevo al editar un auto, en el combobox turno de la pantalla "Editar Automóvil" se debe seleccionar la opción de “ninguno”.
- H) Si al modificar un auto se tilda el checkbox de "Deshabilitar Chofer" se deshabilitara el chofer para ese auto, se pone en cero auto_chofer_estado de la tabla choferPorAuto.

3: FiltroAutomovil

- A) El checkbox que se muestra en la tabla que aparece al tocar el botón “Editar Automóvil” en el menú del ABM auto, se pone en true(aparece el tilde) si el auto está habilitado (es decir auto_estado de tabla Auto en 1) y además si el chofer tiene habilitado ese auto (es decir auto_chofer_estado en 1 para ese auto y ese chofer)
- B)Lo mismo ocurre para los turnos que tiene asociado este auto, es decir en la tabla de los turnos de un auto el checkbox que aparece (tanto desde el botón ver turnos en la grilla de auto y en la que aparece al tocar el botón modificar turnos actuales en la pantalla de modificar automóvil) se pone en True si está el turno habilitado (habilitado en la tabla turno) y además si el auto tiene habilitado ese turno (es decir turno_auto_estado en 1 para ese turno y ese auto en la tabla turnoPorAuto)

• **ABM TURNO**

1: Agregar Turno

El turno nuevo puede ser ingresado habilitado o deshabilitado, dependiendo de si se tilda o no el checkbox "Habilitado" que aparece en la pantalla.

Si al ingresar un turno su hora inicio y su hora fin ya existen en un turno registrado en la base de datos, se muestra un mensaje de error y no se deja guardar el turno.

2: Editar turno

Si se quiere habilitar o deshabilitar un turno , se lo podrá hacer tildando o no el checkbox que aparece en la pantalla que abre el botón modificar de la grilla de turnos.

Un Turno puede deshabilitarse de dos maneras, destildando el checkbox que aparece en la pantalla que se despliega al tocar el botón modificar de la tabla y apretando guardar o con el botón eliminar de la tabla .En ambos casos se establece el campo turno habilitado de la tabla Turno en 0.

• **ABM CHOFER**

1: Agregar Chofer

Al agregar un nuevo Chofer también se lo registra en la tabla "Usuario", al cual se le establece como usuario y contraseña el DNI.

No se permite crear un Chofer con un DNI ya existente.

2: Editar Chofer

En esta opción se muestra un dataGridView con los datos del Chofer. En dicha tabla aparece un checkbox que representa el estado del Chofer, aparecerá desmarcado o no dependiendo del campo chofer estado en la tabla Chofer.

Un chofer puede deshabilitarse de dos maneras, destildando el checkbox que aparece en la pantalla que se despliega al tocar el botón modificar de la tabla y apretando guardar o con el botón eliminar de la tabla. En ambos casos se establece el campo chofer_estado de la tabla Chofer en 0.

Para poder habilitar un Chofer se tiene que hacer click en el botón modificar y tildar el checkbox "Habilitado" de la pantalla Editar Chofer, al tocar en guardar se establecerá en 1 el campo chofer_estado de la tabla Chofer.

Si cuando se modifica el DNI de un Chofer este tenía como usuario y contraseña su DNI, se le actualiza su usuario y contraseña al nuevo DNI ingresado.

- ## **REGISTRO DE VIAJE**

En esta pantalla se muestra un combobox chofer el cual se carga automáticamente con todos los choferes habilitados (chofer_estado =1 en tabla Chofer) que tengan un auto habilitado, un textbox patente que se carga automáticamente al seleccionar un chofer (corresponde a la patente del auto habilitado que tiene ese chofer), un combobox turno que se carga automáticamente al elegir el chofer con los turnos que posee el auto asignado al chofer elegido y un combobox cliente en el cual se cargan automáticamente los clientes habilitados (cliente_estado=1 en tabla Cliente)

Al momento de ingresar los datos, habrá que hacerlo de forma ordenada, ya que la elección del DNI del chofer, traerá la patente del auto habilitado para ese chofer y eso cargará el combobox de los turnos que estén habilitados para ese auto.

Al registrar un viaje en la tabla RegistroViaje los campos factura_id Y rendicion_id se ingresaran en NULL, y se cargaran cuando se facture o se registre respectivamente dicho viaje.

- ## **RENDICION DE VIAJES**

En esta pantalla se muestra una combobox chofer el cual se carga automáticamente con todos los choferes que estén habilitados (chofer_estado=1 en tabla Chofer), un combobox turno el cual se carga automáticamente con todos los turnos habilitados y deshabilitados, ya que si un chofer hizo un viaje cuando el turno estaba habilitado y después no puede rendirlo.

El textbox Importe se cargará automáticamente al seleccionar un chofer y un turno con el 30 % del total de los viajes de ese chofer y turno seleccionado.

Es necesario cargar los datos de forma ordenada, es decir que para que se carguen el importe al seleccionar el turno previamente se tuvieron que seleccionar el chofer y la fecha.

Para registrar la rendición se deberá clicar en el botón generar, al hacerlo se actualizará el campo `rendicion_id` en tabla `Registro_Viaje` de los viajes correspondientes.

- **FACTURACION CLIENTE**

En esta pantalla se muestra un combobox cliente que se carga automáticamente con todos los clientes habilitados (`cliente_estado=1` en tabla `Cliente`), los textbox "Total Facturado" y "Cantidad Viajes" se cargan automáticamente al seleccionar la fecha fin.

Es necesario cargar los datos de forma ordenada, es decir que para que se carguen el total y la cantidad de viajes al seleccionar la fecha fin previamente se tuvieron que seleccionar el cliente y la fecha inicio.

Para facturar se deberá clicar en el botón "Crear Factura", al hacerlo se actualizará el campo `factura_id` en la tabla `Registro_Viaje` de los viajes correspondientes.

Base de datos

Para comunicarnos con la base de datos, nos proveemos de algunas clases de utilidades, todas ubicadas en el directorio *Data Provider*, entre ellas:

- ***ConstructorQuery***

Esta clase nos brinda una interfaz para construir queries fácilmente.

- ***QueryHelper***

Esta clase nos brinda una interfaz para ejecutar queries fácilmente y obtener resultados.

- ***AdministradorDeConexion***

Esta clase es la más cercana a la base de datos, nos crea la conexión con la DB y nos la abre para poder operar, así también como cerrar cuando ya hayamos terminado.

- ***DBMapper***

Permite ejecutar consultas a la base de datos de una forma más simple y sencilla. Le pasamos por parámetro la consulta y una lista con los parámetros que esta consulta necesite, y nos crea un comando listo para ejecutar. Nos abstrae de tener que tener en cuenta la conexión a la base de datos cada vez que mandamos una consulta. Además, sirve como interfaz en la creación, obtención y modificación de objetos con la DB.