

TP Integrador Gestion de Datos “UberFrba”

Estrategia

Integrantes

Fernández, Gastón

Gil, Camila

Gómez, Pablo Matías

Szyszczyk, Agustin



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

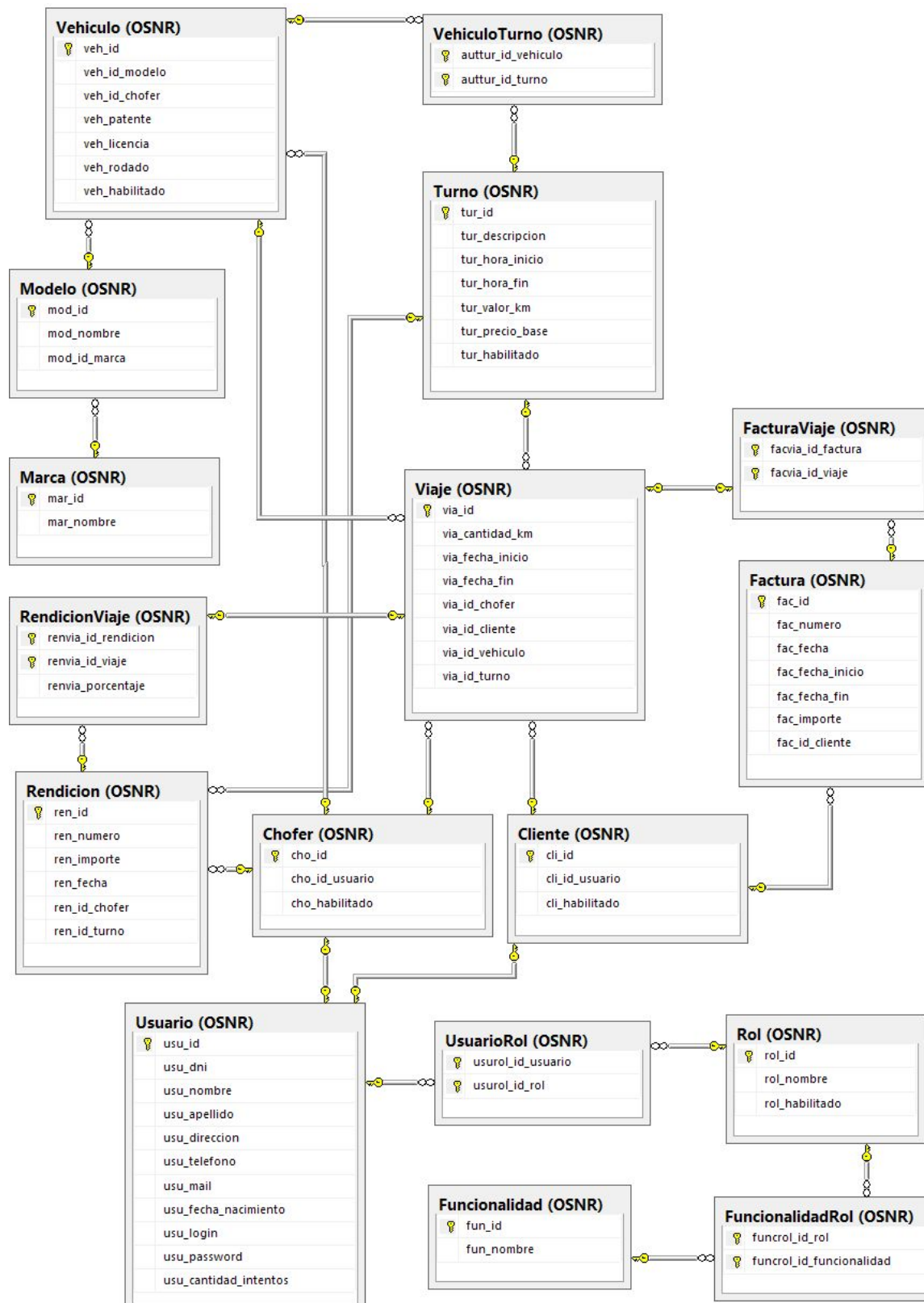


Indice

Indice	1
Diagrama Entidad Relación	2
Decisiones tomadas sobre el DER	3
Descripción de las entidades y relaciones	3
Migración de Datos	6
Stored Procedures Creados	7
Listados Estadísticos	7
Rendición de viajes	7
ABMs Chofer/Cliente/Turno	8
Facturación	8
Registro Viajes	8
Consideraciones sobre las interfaces de usuario	10
Interfaces generales	10
Facturación a Cliente	10
ABM Rol	10
ABM Cliente	11
ABM Chofer	11
ABM Turnos	11
ABM Automovil	12
Rendición de Viajes	12
Listado Estadístico	12
Archivo de configuración	13



Diagrama Entidad Relación





Decisiones tomadas sobre el DER

- Unificamos los datos de Chofer y Cliente en una única entidad llamada Usuario ya que los datos de una misma persona son comunes y esto además nos permite no duplicar datos si la persona es tanto Chofer como Cliente
- Se mantuvo de todas maneras las entidades Chofer y Cliente ya que apuntan a la posibilidad de que cada entidad en particular tenga datos específicos en un futuro.
- Para la dirección del Usuario, tomamos la decisión de dejarlo expresado como un solo campo, tal como está en la tabla maestra, debido a que no existe momento en el que se decida operar sobre alguno de los campos en particular, es decir, uno en ningún momento necesita “usar” el código postal y hacer alguna operación sobre él, sino que lo que uno va a necesitar normalmente es la dirección completa, para por ejemplo, emitir una factura.

Descripción de las entidades y relaciones

- **Rol** : Contenedor de los diferentes roles del sistema (inicialmente Administrador, Cliente y Chofer).
 - Clave primaria: Rol_Id (Valor numérico autoincremental)
- **UsuarioRol** : Entidad que relaciona un rol con un usuario específico
 - Clave primaria: usurol_id_rol (Clave foránea que se relaciona con la entidad Roles) y usurol_id_usuario (Clave foránea que se relaciona con la entidad Usuarios)
- **Usuario** : En esta entidad se almacenan todos los datos de los usuarios que tienen acceso a la aplicación, tales como su nombre de usuario, su clave, los datos personales y la cantidad de intentos fallidos que tuvo al acceder a la aplicación.
 - Clave primaria: usu_id (Valor numérico autoincremental)
- **Chofer**: Entidad que relaciona un chofer con un usuario específico, a su vez, almacena si se encuentra habilitado o no.
 - Clave primaria: cho_id (Valor numérico autoincremental) y cho_id_usuario (Clave foránea que se relaciona con la entidad Usuarios)
- **Cliente**: Entidad que relaciona un cliente con un usuario específico, a su vez, almacena si se encuentra habilitado o no.



- Clave primaria: cli_Id (Valor numérico autoincremental) y cli_id_usuario (Clave foránea que se relaciona con la entidad Usuarios)
- **Funcionalidad**: En dicha entidad se almacena la totalidad de funcionalidades del sistema que serán asignadas a los diferentes roles.
 - Clave primaria: fun_Id (Valor numérico autoincremental)
- **FuncionalidadRol** : Entidad que vincula una funcionalidad con un rol específico
 - Clave primaria: funcrol_id_rol (Clave foránea que se relaciona con la entidad Roles) y Funcrol_id_funcionalidad (Clave foránea que se relaciona con la entidad Funcionalidades)
- **Turno**: En esta entidad se almacenan todos los datos de un turno, tales como su descripción, su hora de inicio, su hora de fin, el valor por km recorrido, su precio base y si se encuentra habilitado o no.
 - Clave primaria: tur_Id (Valor numérico autoincremental)
- **Vehiculo**: En esta entidad se almacenan todos los datos de un vehículo, tales como modelo, su patente, su licencia, su rodado y si se encuentra habilitado o no.
 - Clave primaria: tur_Id (Valor numérico autoincremental) y veh_id_chofer (Clave foránea que se relaciona con la entidad Chofer), veh_id_chofer (Clave foránea que se relaciona con la entidad Chofer) y veh_id_modelo (Clave foránea que se relaciona con la entidad Modelo)
- **VehiculoTurno**: Entidad que vincula un vehículo con un turno específico.
 - Clave primaria: auttur_id_vehiculo (Clave foránea que se relaciona con la entidad Vehiculo) y auttur_id_turno (Clave foránea que se relaciona con la entidad Turno).
- **Modelo**: En esta entidad se almacena el nombre del modelo de los vehículos.
 - Clave primaria: mod_Id (Valor numérico autoincremental) y mod_id_marca (Clave foránea que se relaciona con la entidad Marca).
- **Marca**: En esta entidad se almacena el nombre de la marca de los modelos de los vehículos.
 - Clave primaria: mar_Id (Valor numérico autoincremental).
- **Viaje**: Entidad que almacena los viajes realizados incluyendo como datos su fecha de inicio, su fecha de fin y los km recorridos.



- Clave primaria: via_id (Valor numérico autoincremental), via_id_chofer(Clave foránea que se relaciona con la entidad Chofer), via_id_cliente (Clave foránea que se relaciona con la entidad Cliente), via_id_vehiculo (Clave foránea que se relaciona con la entidad Vehiculo) y via_id_turno (Clave foránea que se relaciona con la entidad Turno).

- **Factura**: Entidad que almacena los viajes realizados incluyendo como datos su fecha de inicio, su fecha de fin, número de factura y el importe.
 - Clave primaria: fac_id (Valor numérico autoincremental) y fac_id_cliente(Clave foránea que se relaciona con la entidad Cliente).

- **FacturaViaje**: Entidad que vincula una factura con un viaje específico.
 - Clave primaria: facvia_id_factura (Clave foránea que se relaciona con la entidad Factura) y facvia_id_viaje (Clave foránea que se relaciona con la entidad Viaje).

- **Rendición**: Entidad que almacena los datos para la rendición al chofer tales como el importe, el número de rendición y la fecha.
 - Clave primaria: ren_id (Valor numérico autoincremental), ren_id_turno (Clave foránea que se relaciona con la entidad Turno) y ren_id_chofer (Clave foránea que se relaciona con la entidad Chofer).

- **RendicionViaje**: Entidad que vincula un viaje con una rendición especificando su porcentaje.
 - Clave primaria: renvia_id_rendicion (Clave foránea que se relaciona con la entidad Rendicion) y renvia_id_viaje (Clave foránea que se relaciona con la entidad Viaje).



Migración de Datos

Para la migración de datos, se tomaron ciertas decisiones para que se adapte a nuestro modelo de datos armado:

- La creación de Roles, y Funcionalidades de cada rol está explícitamente hecha en la migración, esto significa que al momento de crear las correspondientes tablas, se crean a mano (lo hace el script) las roles y sus funcionalidades asociadas para que el sistema funcione correctamente.
- En todos los casos de entidades que tienen un flag de habilitado/deshabilitado, se tomo la decisión de crearlos como habilitados por default, ya que esta información no está en la tabla maestra, y no tendría sentido crearlos deshabilitados ya que el sistema sería inusable.
- Para lo intentos de login tomamos la decisión, todos los usuarios son creados con 0 intentos de login al momento de la migración
- Para el login de usuarios, se tomó como decisión usar datos del mismo usuario para la generación de sus datos de acceso, siendo estos, tanto para choferes como para clientes, los siguientes:
 - Usuario: telefono
 - Password: telefono

Ya que el teléfono es único para cada usuario y a su vez, el usuario requería ser único.

- Se tomó como decisión agregar ciertas constraints para impedir la creación de entidades inconsistentes, basadas en la lógica de negocio pedida. Entre ellas:
 - Constraint UNIQUE en Rendición para los campos fecha, chofer y turno, ya que no se puede crear una rendición dos veces para el mismo chofer, en el mismo turno y fecha.
 - En la entidad RendicionViaje se puso el campo viaje como UNIQUE ya que un viaje puede ser rendido una única vez.
 - En la entidad Usuario, el campo teléfono es UNIQUE, ya que esta informacion es dada en el enunciado. Además, el campo login también es UNIQUE, ya que sino habria problemas al momento de loguearse.
- Para los números de rendición, se tomo la decisión de agregar una secuencia que los genere. En el momento de la migración, se toma el valor máximo de la migración, y luego se crea una secuencia para que luego en las próximas creaciones de rendiciones, este número sea autogenerado. La rendición se llama "SecuenciaRendicionNumero".
- Para validar la superposición de horarios de los turnos, tomamos la decisión de crear una función en la base de datos que ejecute la validación, y una constraint asociada a esta, para que la ejecución de la función se realice en cualquier momento y de esta manera no poder guardar datos inválidos. La función que lo chequea se llama CheckTurnoOverlap.



- Para validar que los viajes tengan el atributo kilómetros y este sea mayor a 0 obligatoriamente se agrego una constraint específica sobre esta columna que básicamente chequea que sea mayor a 0. Esto nos permite asegurarnos la consistencia de datos.

Stored Procedures Creados

Listados Estadísticos

Tomamos la decisión de utilizar 4 stored procedures para la generación de listados estadísticos. Por cada listado estadístico pedido, se implementó un stored procedure que recibe por parámetro el rango de fecha que se desea consultar. De esta manera, la implementación del código quedó abstraída del manejo de datos para la generación de los reportes. Los nombres son:

- TOP5ChoferesConMayorRecaudacion
- TOP5ChoferesConViajeMasLargo
- TOP5ClientesConMayorConsumo
- TOP5ClientesConMayorCantidadDeMismoAutomovil

Rendición de viajes

Para la rendición de viajes a choferes se tomo la decisión de implementar un Stored Procedure que maneje toda la lógica de los viajes que componen la rendición. De esta manera, el código de la aplicación solo se encarga de pasarle por parámetros, las fecha del día a rendir, el turno correspondiente, el chofer, y el porcentaje a aplicar sobre el valor del viaje. Una vez que el SP guarda todos los datos de la rendición (incluidos los viajes que lo componen), se devuelve la lista de Viajes que fueron incluidos en dicha rendición, de manera que pueda ser mostrado en pantalla para que el usuario sepa de donde proviene el importe total generado. Cabe destacar que el SP internamente ejecuta la inserción de Viajes (Tabla intermedia ViajesRendicion) y Rendición bajo una misma transacción, de manera que si este falla, no queden datos inconsistentes.



ABMs Chofer/Cliente/Turno

Tomamos la decisión de utilizar 4 stored procedures para dichos ABMs. El primer SP que se utiliza es el que busca todos los datos para llenar las tablas de los mismos. Luego existen otros dos que proceden a habilitar o deshabilitar un turno/chofer/cliente. Por último creamos un SP para la creación y modificación del cliente todo en uno, si se le pasa un Id ya existente modifica sus datos si no, lo crea.

Los nombres son:

- Buscar
- Deshabilitar
- Habilitar
- ModificarOCrear

Facturación

Para la facturación a clientes se tomo la decisión de implementar un Stored Procedure que maneje toda la lógica de los viajes que integran la facturación. De esta manera, el código de la aplicación solo se encarga de pasarle por parámetros, las fechas de la factura y el idCliente a facturar, y, luego de que el SP es ejecutado correctamente, este mismo devuelve la lista de Viajes que fueron incluidos en dicha factura, de manera que pueda ser mostrado en pantalla para que el usuario sepa de donde proviene el importe total generado.

Registro Viajes

Para la parte de rendición de viajes, se permite únicamente cargar por parte de operadores o administradores viajes relacionados a autos y turnos asignados a los choferes, ya que al realizar la migración entendemos que no va a quedar ningún viaje anterior a cargar.

A su vez no dejamos cargar viajes sobre turnos inhabilitados ya que de estarlo no será posible registrar viaje alguno.

En cuanto a la estrategia utilizada y debido a que el chofer ya tiene asignado un turno y un vehículo decidimos que al mostrar los datos del chofer mostramos todos los asignados al



mismo, luego permitimos cargar los km con sus validaciones, sus fechas y por ultimo los clientes.

Para poder llevar a cabo esta estrategia utilizamos 3 Stored Procedure dos similares para la validación de viajes en fechas repetidas al cual le llegan los registros de fechas y el id de cliente o chofer según corresponda y otro para realizar el registro de viaje al cual le llegan directamente los parámetros necesarios ya validados con anterioridad.



Consideraciones sobre las interfaces de usuario

Interfaces generales

- **InicioSesion:** Formulario para iniciar sesión en el sistema. El usuario deberá ingresar sus datos de acceso para continuar. El sistema validará que los mismos sean correctos y en el caso de que la clave no lo sea, se sumará un intento fallido. Si los intentos superan la cantidad máxima establecida en el archivo de configuración, no se podrá volver a acceder a la cuenta hasta que la misma sea desbloqueada.
- **EleccionRol :** Interfaz que le permite seleccionar al usuario logueado el rol por el cual desea operar en el sistema. Solo se listan los roles que el usuario tiene asignado y que se encuentran activos.
- **FormPrincipal :** Formulario principal del sistema donde se cargan de forma dinámica las funcionalidades que un usuario tiene disponibles de acuerdo a su rol.

Facturación a Cliente

- **RendicionViajes:** En esta pantalla, pedimos al usuario que ingrese por pantalla un cliente, una fecha de inicio y una de fin para proceder a generar la facturación del periodo. Cabe destacar que una vez que el usuario genera la factura, se muestran los viajes que la incluyen, y queda la factura guardada en la base de datos, por lo que no se podrá volver a generar una nueva factura con los mismos parámetros, ya que un viaje no puede ser rendido dos veces. Otro punto importante es que la fecha de fin es completamente inclusiva sobre el día, es decir, incluye todos los viajes que tienen como fecha ese mismo día, sin importar la hora. Además de esto, se muestra el importe total que corresponde a la factura generada, así como también el número generado de la misma.

ABM Rol

- **AbmRol:** Interfaz a la cual solo puede ingresar el administrador. Posee tres botones que redireccionan a los formularios para crear, modificar y deshabilitar un rol (este es el único que no redirecciona a otro formulario, deshabilita el rol seleccionado en esta interfaz).
- **EditarRol :** Interfaz en la cual se puede crear un nuevo rol o modificar un rol existente, definiendo su nombre y sus funcionalidades. No se permite crear un rol sin funcionalidades, sin nombre o con un nombre que se encuentre en uso. Se puede habilitar en el caso de que se encuentre inhabilitado.



ABM Cliente

- **AbmCliente:** Pantalla de búsqueda de clientes en base a su nombre, apellido y/o documento. Todos estos campos son opcionales y permiten realizar la búsqueda con parte de los datos, a excepción del dni que debe ser ingresado exacto. Al pulsar buscar se listan sobre la grilla los clientes que coinciden con los criterios establecidos y se pueden utilizar los botones de Editar y Habilitar/Deshabilitar al seleccionar una de las filas. El botón de añadir y el de editar abren EditarCliente y el de Habilitar/Deshabilitar realiza una baja/alta lógica del cliente. La grilla muestra desde un principio todos los datos que se encuentran cargados en la base.
- **EditarCliente:** Interfaz en la que se permite editar o añadir a un cliente. Todos los campos deben ser completados a excepción del Email que es un campo opcional. El único dato que no puede repetirse entre clientes es el teléfono.

ABM Chofer

- **AbmChofer:** Pantalla de búsqueda de choferes en base a su nombre, apellido y/o documento. Todos estos campos son opcionales y permiten realizar la búsqueda con parte de los datos, a excepción del dni que debe ser ingresado exacto. Al pulsar buscar se listan sobre la grilla los choferes que coinciden con los criterios establecidos y se pueden utilizar los botones de Editar y Habilitar/Deshabilitar al seleccionar una de las filas. El botón de añadir y el de editar abren FormEditarChofer y el de Habilitar/Deshabilitar realiza una baja/alta lógica del chofer. La grilla muestra desde un principio todos los datos que se encuentran cargados en la base.
- **FormEditarChofer:** Interfaz en la que se permite editar o añadir a un chofer. Todos los campos deben ser completados. El único dato que no puede repetirse entre choferes es el teléfono.

ABM Turnos

- **AbmTurno:** Pantalla de búsqueda de turnos en base a su descripción. Es un campo opcional y permite realizar la búsqueda con parte de los datos. Al pulsar buscar se listan sobre la grilla los turnos que coinciden con el criterio establecido y se pueden utilizar los botones de Editar y Habilitar/Deshabilitar al seleccionar una de las filas. El botón de añadir y el de editar abren FormEditarTurno y el de Habilitar/Deshabilitar realiza una baja/alta lógica del turno. La grilla muestra desde un principio todos los datos que se encuentran cargados en la base.
- **FormEditarTurno:** Interfaz en la que se permite editar o añadir a un turno. Todos los campos deben ser completados.



ABM Automovil

- **AbmAutomovil:** Pantalla de búsqueda en base a Marca, Modelo, Patente y Chofer. Ninguno de los campos es obligatorio para la búsqueda. Luego de pulsar el botón “Buscar”, se listan todos los automóviles que cumplan con lo filtrado. Las opciones válidas sobre los buscados, son “Borrar”, realizando un borrado lógico, inhabilitando el vehículo, o “Modificar”, permitiendo que el vehículo seleccionado modifique alguno, o todos sus campos. También se encuentra visible el botón “Alta Nuevo Vehículo”, que permite agregar un nuevo vehículo al sistema.
- **AltaModVehiculo:** Interfaz que permite el alta de un nuevo vehículo, o la modificación de un vehículo seleccionado en la pantalla anterior. Los campos Marca, Modelo, Patente, Turno y Chofer son obligatorios, mientras que Licencia, Rodado y el estado del automóvil (Activo/inactivo) no lo son.

Rendición de Viajes

- **RendicionViajes:** En esta pantalla, pedimos al usuario que ingrese por pantalla un chofer, una fecha y un turno para proceder a generar la rendición de pago al chofer. En pantalla además se visualiza el porcentaje utilizado que sale del archivo de configuración. Cabe destacar que una vez que el usuario genera la rendición, se muestran los viajes que la incluyen, y queda la rendición guardada en la base de datos, por lo que no se podrá volver a generar una nueva rendición con los mismos parámetros, ya que un viaje no puede ser rendido dos veces. Además de esto, se muestra el importe total que corresponde a la rendicion generada, asi como tambien el numero de la misma

Listado Estadistico

- **ListadoEstadistico:** En esta pantalla desarrollamos los listados que muestran los primeros 5 registros de diferentes estadísticas. Para la generación del listado, se pide al usuario la introducción del año a utilizar y el trimestre del mismo, y luego utilizamos un combo que esta directamente asociado a un SP específico y que recibe como parámetro el trimestre a consultar. De esta manera la consulta para mostrar los primeros 5 es genérica y puede ser fácilmente reutilizable si se desea agregar un nuevo listado estadístico.



Archivo de configuración

La aplicación cuenta con un archivo de configuración llamado App.config, en el cual se permite diferentes parámetros a tener en cuenta:

- Fecha y hora del sistema (en formato MM/DD/YYYY HH:MM:SS)
- El porcentaje de rendición.
- ConnectionString de la base de datos.