TP Gestión de Datos
2C - 2017

Pago Ágil

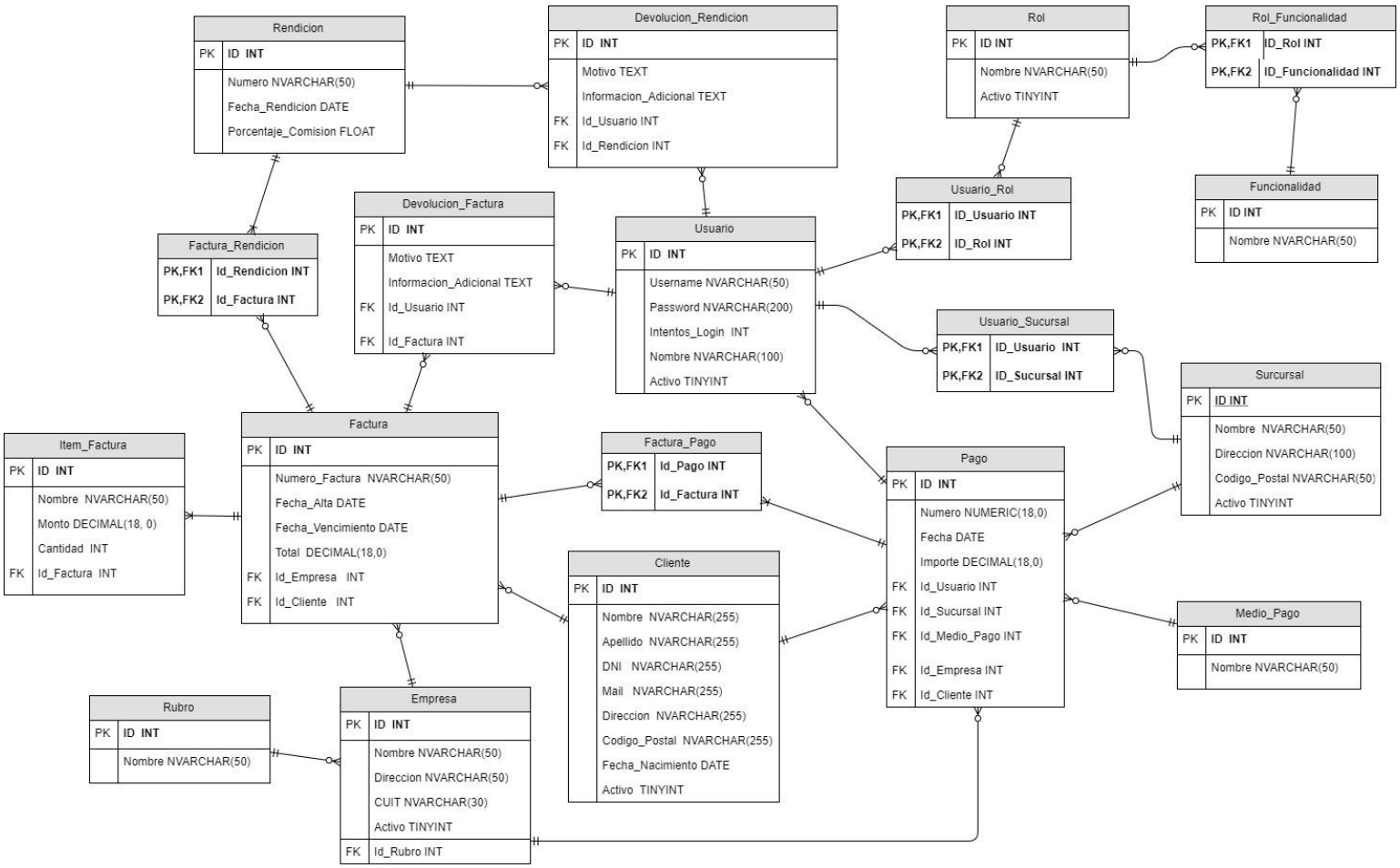
Grupo PUNTO_ZIP:

- Tomas Chejanovich (155547-9)
- Leila Feiguin (155990-4)

Índice

DER……..……3
Estructuras de datos…….……...4
Tablas……..4
Stored procedures…….……8
Functions…….14
Aplicación C#.....15
Sobre la no utilización de clases……..15
Útiles…….15
Usuarios del sistema……..16

DER


Rendicion
PK ID INT
Numero NVARCHAR(50)
Fecha_Rendicion DATE
Porcentaje_Comision FLOAT
Devolucion_Rendicion
PK ID INT
Motivo TEXT
Informacion_Adicional TEXT
FK Id_Usuario INT
FK Id_Rendicion INT
Rol
PK ID INT
Nombre NVARCHAR(50)
Activo TINYINT
Rol_Funcionalidad
PK,FK1 ID_Rol INT
PK,FK2 ID_Funcionalidad INT
Usuario_Rol
PK,FK1 ID_Usuario INT
PK,FK2 ID_Rol INT
Funcionalidad
PK ID INT
Nombre NVARCHAR(50)
Devolucion_Factura
PK ID INT
Motivo TEXT
Informacion_Adicional TEXT
FK Id_Usuario INT
FK Id_Factura INT
Usuario
PK ID INT
Username NVARCHAR(50)
Password NVARCHAR(200)
Intentos_Login INT
Nombre NVARCHAR(100)
Activo TINYINT
Factura_Rendicion
PK,FK1 Id_Rendicion INT
PK,FK2 Id_Factura INT
Usuario_Sucursal
PK,FK1 ID_Usuario INT
PK,FK2 ID_Sucursal INT
Surcursal
PK ID INT
Nombre NVARCHAR(50)
Direccion NVARCHAR(100)
Codigo_Postal NVARCHAR(50)
Activo TINYINT
Item_Factura
PK ID INT
Nombre NVARCHAR(50)
Monto DECIMAL(18, 0)
Cantidad INT
FK Id_Factura INT
Factura
PK ID INT
Numero_Factura NVARCHAR(50)
Fecha_Alta DATE
Fecha_Vencimiento DATE
Total DECIMAL(18,0)
FK Id_Empresa INT
FK Id_Cliente INT
Factura_Pago
PK,FK1 Id_Pago INT
PK,FK2 Id_Factura INT
Pago
PK ID INT
Numero NUMERIC(18,0)
Fecha DATE
Importe DECIMAL(18,0)
FK Id_Usuario INT
FK Id_Sucursal INT
FK Id_Medio_Pago INT
FK Id_Empresa INT
FK Id_Cliente INT
Cliente
PK ID INT
Nombre NVARCHAR(255)
Apellido NVARCHAR(255)
DNI NVARCHAR(255)
Mail NVARCHAR(255)
Direccion NVARCHAR(255)
Codigo_Postal NVARCHAR(255)
Fecha_Nacimiento DATE
Activo TINYINT
Medio_Pago
PK ID INT
Nombre NVARCHAR(50)
Rubro
PK ID INT
Nombre NVARCHAR(50)
Empresa
PK ID INT
Nombre NVARCHAR(50)
Direccion NVARCHAR(50)
CUIT NVARCHAR(30)
Activo TINYINT
FK Id_Rubro INT

Estructuras de datos

Tablas:

- **Sucursal**: representa a la sucursal.
    - id: int **(PK)**
    - nombre: nvarchar(50)
    - direccion: nvarchar(100)
    - codigo_postal: nvarchar(50)
    - activo: tinyint

- **Usuario**: representa al usuario. El número de intentos máximo del login antes de bloquear el usuario es de 3.
    - id: int **(PK)**
    - username nvarchar(50)
    - password nvarchar(200) : Se guarda encriptado mediante la función HashBytes('SHA_256', @password).
    - nombre nvarchar(50)
    - intentos_login: int
    - activo: tinyint

- **Usuario_Sucursal**: representa la relación entre usuario y sucursal mediante Foreign Keys. Se utiliza esta tabla ya que la relación Usuario-Sucursal es de muchos a muchos.
    - id_usuario: int **(PK, FK Usuario)**
    - id_sucursal: int **(PK, FK Sucursal)**

- **Rol**: respresenta al rol.
    - id: int **(PK)**
    - nombre: nvarchar(50)
    - activo: tinyint

- **Usuario_Rol**: representa la relación entre usuario y rol mediante Foreign Keys. Se utiliza esta tabla ya que la relación Usuario-Rol es de muchos a muchos.
    - id_usuario: int **(PK, FK Usuario)**
    - id_rol: int **(PK, FK Rol)**

- **Funcionalidad**: representa a los módulos del sistema. Los registros de esta tabla se cargan en el script de creación y no se puede agregar, modificar o eliminarlos.
    - id: int **(PK)**
    - nombre: nvarchar(50)

- **Rol_Funcionalidad**: representa la relación entre rol y funcionalidad mediante Foreign Keys. Esta tabla establece a que módulos del sistema tendrá acceso cada rol.
    - id_rol: int **(PK, FK Rol)**
    - id_funcionalidad: int **(PK, FK Funcionalidad)**

- **Factura:** representa a la factura. Se decidió hacer una redundancia en cuanto al total de la misma (ya que es calculable mediante la suma de los items de la misma) pero para mejorar la performance y no tener la necesidad de iterar por todos los items de la factura cada vez que se quiere obtener el total, se decidió guardarlo e ir actualizándolo cada vez que se agreguen nuevos items.
    - id: int **(PK)**
    - numero_factura: nvarchar(50)
    - fecha_alta: date
    - fecha_vencimiento: date
    - total: decimal(18, 0)
    - id_empresa: int (FK Empresa)
    - id_cliente: int (FK Cliente)

- **Factura_Pago:** representa la relación entre factura y pago. Esta tabla no es estrictamente necesaria, ya que podria reemplazarse utilizando un id_pago en la table Factura. Sin embargo, creemos que mejor tener una entidad intermedia entre ambos. Esto hace el sistema mas escalable.
    - id_pago **(PK,FK Pago)**
    - id_factura **(PK,FK Factura)**

- **Factura_Rendicion:** representa la relación entre rendición y factura mediante Foreign Keys. Si bien cada factura puede ser rendida solo una vez, fue necesario establecer una relación de muchos a muchos para poder contemplar las devoluciones de las rendiciones, ya que en este caso la factura terminara quedando asociada a más de una rendición, pero solo una de estas será “valida”.
    - id_rendicion **(PK,FK Rendición)**
    - id_factura **(PK,FK Factura)**

- **Rendicion**: representa a la rendición. El dato más “relevante” es el porcentaje de la comision, ya que al combinarlo con el monto de la factura (que se obtiene a través de la tabla Factura_Rendicion) se puede obtener el monto de la rendición para cada factura y el total de la misma.
    - id: int **(PK),**
    - numero: nvarchar(50)
    - fecha_rendicion: date
    - porcentaje_comision: float

- **Devolucion_Rendicion**: representa a la devolución de la rendición. Se relaciona con Usuario y Rendición mediante Foreign Keys.
    - id: int **(PK)**
    - motivo: text
    - informacion_adicional: text
    - id_usuario: int (FK Usuario)
    - id_rendicion: int (FK Rendicion)

- **Cliente**: representa al cliente.
    - id: int **(PK)**
    - nombre:nvarchar(50)
    - apellido: nvarchar(50)
    - dni: nvarchar(50): no puede repetirse entre cliente.
    - mail: nvarchar(50)
    - direccion: nvarchar(50)
    - codigo_postal: nvarchar(50)
    - fecha_nacimiento: date
    - activo: tinyint

- **Item_Factura**: representa al item de la factura. Se relaciona con Factura mediante Foreign Key.
    - id: int **(PK)**
    - nombre: nvarchar(50)
    - monto: double
    - cantidad: int
    - id_factura: int (FK Factura),

- **Devolucion_Factura**: En cuanto a estructura es igual a Devolucion_Rendicion, pero está asociada a una factura en vez de a una rendición.
    - id: int **(PK)**
    - motivo: text,
    - informacion_adicional: text
    - id_usuario: int (FK Usuario)
    - id_factura: int (FK Factura)

- **Rubro**: representa al rubro. Se cargan en la creación inicial y no se pueden agregar, modificar o eliminar.
    - id: int **(PK),**
    - nombre: nvarchar(50)

- **Empresa**: representa a la empresa y se relaciona con Rubro mediante Foreign Key.
    - id: int **(PK)**
    - nombre: nvarchar(50)
    - direccion: nvarchar(50)
    - cuit: nvarchar(30)
    - activo: boolean
    - id_rubro: int (FK Rubro)

- **Medio_Pago**: representa al medio de pago. Se cargan en la creación inicial y no se pueden agregar, modificar o eliminar.
    - id: int **(PK)**
    - nombre: nvarchar(50),

- **Pago**: representa al pago. Se relaciona con Usuario, Sucursal, Medio_Pago, Cliente y Factura mediante Foreign Keys.
    - id: int **(PK)**
    - numero: numeric, (18,0)
    - fecha: date
    - importe: decimal (18.0)
    - id_usuario: int (FK)
    - id_sucursal: int (FK)
    - id_medio_pago: int (FK Medio de Pago)
    - id_cliente: int (FK Cliente)
    - id_empresa: int (FK Empresa)

Store Procedures:

- **SP_Add_Item_Factura**: agrega un ítem a una factura.
    - Parámetros: número de factura, nombre del ítem, monto del mismo y cantidad.

- **SP_Baja_Cliente_By_Id**: elimina un cliente por ID.
    - Parámetros: ID cliente.

- **SP_Baja_Empresa_By_Id**: elimina una empresa por ID.
    - Parámetros: ID empresa.

- **SP_Baja_Sucursal_By_Id**: elimina una sucursal por ID
    - Parámetros: ID sucursal.

- **SP_Creacion_Inicial**: Procedure que se utiliza en el script de creación inicial. Ejecuta las migraciones de la tabla maestra y crea los roles y usuarios iniciales.
    - Parámetros: SIN PARAMETROS.

- **SP_Creacion_Cliente**: da de alta un cliente.
    - Parámetros: nombre, apellido, DNI, email, dirección, código postal y fecha de nacimiento.

- **SP_Create_Devolucion_Factura**: crea una devolución de una factura.
    - Parámetros: ID usuario, numero factura, motivo e información adicional.

- **SP_Create_Devolucion_Rendicioon**: crea una devolución de una rendición.
    - Parámetros: ID usuario, numero rendición, motivo e información adicional.

- **SP_Create_Empresa**: da de alta una empresa.
    - Parámetros: nombre empresa, dirección, CUIT e ID del rubro.

- **SP_Create_Factura**: da de alta una factura.
    - Parámetros: ID cliente, ID empresa, numero factura, fecha de vencimiento y fecha de alta.

- **SP_Create_Pago**: registra un pago por una o mas facturas, siendo el monto del pago la suma del monto de las facturas.
    - Parámetros: números de facturas concatendados (EJ: 1,2,3), ID usuario, ID sucursal, ID medio de pago, ID cliente, ID empresa.

- **SP_Create_Rol**: da de alta un rol asignandole funcionalidades.
    - Parámetros: nombre rol, activo, IDs de las funcionalidades concatenados.

- **SP_Create_Sucursal**: da de alta una sucursal validando que no haya una ya con el mismo codigo postal.
    - Parámetros: nombre sucursal, dirección y código postal.

- **SP_Delete_Rol**: elimina un rol por id.
    - Parámetros: ID rol.

- **SP_Eliminar_Factura**: da de baja una factura por numero factura.
    - Parámetros: número de factura.

- **SP_Estadisticas_Clientes_Con_Mas_Pagos_Listado_Detallado**: muestra un listado detallado de los 5 clientes con más pagos. Muestra cantidad de pagos, monto total de los mismos, nombre del cliente, DNI, dirección, fecha de nacimiento, mail y código postal del mismo.
    - Parámetros: año y numero de trimestre.

- **SP_Estadisticas_Clientes_Con_Mas_Pagos_Listado_Simple**: muestra un listado simple de los 5 clientes con más pagos, mostrando únicamente la cantidad de pagos y el nombre del cliente.
    - Parámetros: año y numero de trimestre.

- **SP_Estadisticas_Clientes_Cumplidores_Listado_Detallado**: muestra un listado detallado de los 5 clientes más cumplidores (que más facturas pendientes pagaron). Muestra el porcentaje de facturas pagadas, la cantidad de facturas pagadas, la cantidad de facturas totales y los datos del cliente.
    - Parámetros: año y numero de trimestre.

- **SP_Estadisticas_Clientes_Cumplidores_Listado_Simple:** muestra un listado simple de los 5 clientes más cumplidores. Muestra únicamente el porcentaje de facturas pagadas y el nombre del cliente.
    - Parámetros: año y numero de trimestre.
- **SP_Estadisticas_Empresas_Mayor_Monto_Rendido_Listado_Detallado:** muestra un listado detallado de las 5 empresas más con mayor monto rendido en el periodo. Muestra el monto rendido, la cantidad de facturas rendidas y los datos de la empresa.
    - Parámetros: año y numero de trimestre.

- **SP_Estadisticas_Empresas_Mayor_Monto_Rendido_Listado_Simple:** muestra un listado simple de las 5 empresas más con mayor monto rendido en el periodo. Muestra unicamente el monto rendido y el nombre de la empresa.
    - Parámetros: año y numero de trimestre.

- **SP_Estadisticas_Porcentaje_Facturas_Cobradas_x_Empresa_Listado_Detallado:** muestra un listado detallado de las 5 empresas más con mayor porcentaje de facturas cobradas. Muestra los datos de la empresa, el porcentaje, la cantidad de facturas cobradas y la cantidad total.
    - Parámetros: año y numero de trimestre.

- **SP_Estadisticas_Porcentaje_Facturas_Cobradas_x_Empresa_Listado_Simple:** muestra un listado simple de las 5 empresas más con mayor porcentaje de facturas cobradas. Muestra unicamente el porcentaje y el nombre de la empresa.
    - Parámetros: año y numero de trimestre.

- **SP_Get_Cliente_By_Id**: devuelve un cliente por ID.
    - Parámetros: ID cliente.

- **SP_Get_Clientes**: devuelve el listado de todos los clientes.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Detalle_Factura**: devuelve los ítems asociados a una factura.
    - Parámetros: número de factura.

- **SP_Get_Detalle_Rendicion**: devuelve las facturas asociadas a una rendición.
    - Parámetros: número de rendición.

- **SP_Get_Devolucion_x_Numero_Factura**: devuelve la devolución de una factura (si es que la tiene).
    - Parámetros: número de factura.

- **SP_Get_Devolucion_x_Numero_Rendicion**: devuelve la devolución de una rendición (si es que la tiene).
    - Parámetros: número de rendición,.

- **SP_Get_Empresas:** devuelve un listado de todas las empresas.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Empresas_x_Campos**: devuelve un listado de empresas filtrando por nombre, CUIT y rubro.
    - Parámetros: nombre, CUIT e ID rubro.

- **SP_Get_Facturas**: devuelve un listado de todas las facturas..
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Facturas_Pago**: devuelve un listado de las facturas asociadas a un pago.
    - Parámetros: numero pago.

- **SP_Get_Funcionalidades**: devuelve un listado de todas las funcionalidades del sistema.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Funcionalidades_Rol**: devuelve un listado de las funcionalidades de un determinado rol.
    - Parámetros: ID rol.

- **SP_Get_Medios_Pago**: devuelve todos los medios de pago.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Nombres_Clientes**: devuelve los nombres de todos los clientes.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Pagos**: devuelve un listado de todos los pagos del sistema.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Rendiciones**: devuelve un listado de todas las rendiciones,
    - Parámetros: SIN PARAMETROS

- **SP_Get_Roles**: devuelve un listado de todos los roles..
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Roles_Usuario**: devuelve todos los roles asociados a un usuario.
    - Parámetros: ID usuario.

- **SP_Get_Rubro_By_Nombre**: devulve el id de un rubro
    - Parámetros: nombre rubro.

- **SP_Get_Rubros**: devuelve un listado de todos los rubros.
    - Parámetros: SIN PARAMETROS.

- **SP_Get_Sucursal_By_Id**: devuelve una sucursal.
    - Parámetros: ID sucursal.

- **SP_Get_Sucursales**: devuelve un listado de todas las sucursales.
    - Parámetros: SIN PARAMETROS.
- **SP_Get_Sucursales_Usuario:** devuelve todas las sucursales asociadas a un usuario.
    - Parámetros: ID usuario.

  SP_Get_Sucursales_x_Campos: filtra sucursales por nombre, dirección y código postal.
    - Parámetros: nombre, dirección y código postal.

- **SP_Login**: valida el usuario y el password al momento de login.
    I. En caso de que el usuario no exista, devuelve “Acceso denegado”
    II. En caso de que el usuario exista pero el password sea incorrecto, registra un intento de login. A los 3 intentos fallidos deshabilita el usuario.
    III. En caso de login existoso, devuelve el ID y nombre del usuario junto con el ID del rol y de la sucursal a asociados. En caso de que de alguno de estos dos, haya más de uno (más de un rol por ejemplo), devuelve -1. Pero el caso donde no hay ninguna sucursal asociada (usuarios no cobradores por ejemplo) devuelve cero.
    - Parámetros: número de factura, nombre del item, monto del mismo y cantidad.

- **SP_Migrar_Clientes**: migrar los clientes de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Datos**: todos los datos de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Empresas**: migrar las empresas de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Facturas**: migrar las facturas de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Items_Factura**: migrar los items de las facturas de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Medios_Pago**: migrar los medios de pago de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Pagos**: migrar los pagos de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Rendiciones**: migrar las rendiciones de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Rubros**: migrar los rubros de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Migrar_Sucursales**: migrar las sucursales de la tabla maestra.
    - Parámetros: SIN PARAMETROS.

- **SP_Rendir_Facturas**: rinde facturas especificadas.
    - Parámetros: porcentaje comision y numeros de factura.

- **SP_Set_Funcionalidades_Rol**: agrega funcionalidades a un rol.
    - Parámetros: ID rol e IDs funcionalidades

- **SP_Update_Cliente**: actualiza un cliente
    - Parámetros: todos los datos del cliente

- **SP_Update_Empresa**: actualiza una empresa
    - Parámetros: todos los datos de la empresa
    .
- **SP_Update_Rol**: actualiza un rol
    - Parámetros: ID rol, nombre, activo y listado de funcionalidades.

- **SP_Update_Sucursal**: actualiza una sucursal
    - Parámetros: todos los datos de la sucursal.

- **SP_Validar_Codigo_Postal_Sucursal**: verifica si existe una sucursal con un determinado codigo postal.
    - Parámetros: codigo postal

- **SP_Validar_Cuit_Empresa**: valida si existe una empresa distinta con el mismo CUIT.
    - Parámetros: CUIT e ID empresa.

- **SP_Validar_Funcionalidad_Rol**: valida si un rol tiene permiso para una determinada funcionlidad.
    - Parámetros: ID rol y nombre funcionalidad

- **SP_Validar_Mail_Cliente**: valida si existe otro cliente con el mismo mail.
    - Parámetros: ID cliente y mail

Functions:

- **Split**: devuelve un listado strings de acuerdo a un string base y un separador.
    - Parámetros: string base y separador.

- **Cantidad_Facturas_Cobradas_x_Empresa**: devuelve la cantidad de facturas cobradas por una empresa en un determinado periodo.
    - Parámetros: ID empresa, año y numero trimestre.

- **Meses_x_Trimestre**: devuelve los numeros de meses pertenicientes a un determinado numero de trimestre.
    - Parámetros: numero trimestre.

- **Validar_Fecha_En_Periodo**: verifica si una fecha se encuentra en un determinado periodo.
    - Parámetros: fecha, año y numero trimestre.

Aplicación C#

- **Sobre la no utilización de clases:**
  Un posible enfoque para la aplicación de escritorio podría haber sido orientado a objetos, y así mappear las entidades de la BD con clases de C#. Sin embargo, esto no nos pareció necesario por las siguientes razones:
  - La no utilización de clases permite mayor versatilidad a la hora mostrar un listado de información. Por ejemplo, si directamente insertamos el resultado de un procedure en una DataGridView, si quisiéramos agregar o quitar columnas a ese listado, bastaría con modificar el procedure. En cambio, si convirtiéramos los resultados de los procedures en clases, este cambio no resultaría tan fácil y rápido
  - Varias de las pantallas son estáticas, es decir, no tienen mayor complejidad que la carga inicial.
  - La utilización de clases suele fácil de implementar cuando se utiliza algún ORM, este no es el caso

- **Útiles:**
  Al margen de los ABMs propiamente dichos, se creo el namespace ‘Utiles’ con el objetivo de simplificar el código y alojar allí funciones o variables de uso general. El namespace está compuesto por 2 clases:
  - Útiles: clase estática con funciones de uso general
  - Global: clase estática

Usuarios del sistema

El sistema cuenta con los siguientes usuarios:

1. **Username**: admin
   Password: w23e
   Rol: Administrador
2. **Username**: tomas
   Password: 1234
   Rol: Cobrador
   Sucursal asociada: Sucursal central
3. **Username**: leila
   Password: 7777
   Rol: Cobrador
   Sucursal asociada: Sucursal central y Sucursal n°2000
4. **Username**: jefe
   Password: jajaja
   Roles: Administrador y cobrador
   Sucursal asociada: Todas