

**Estrategia de TP**

**“LOS\_PUBERTOS”**

Grupo: 49

|  |  |
| --- | --- |
| NOMBRE | LEGAJO |
| Nicolás Ledesma | 1470620 |
| Matias Farran | 1472379 |
| Diego Guida | 1466161 |
| Florencia Macarena Lopez | 1465284 |

Contenido

[Introducción 3](#_Toc496022516)

[DER 3](#_Toc496022517)

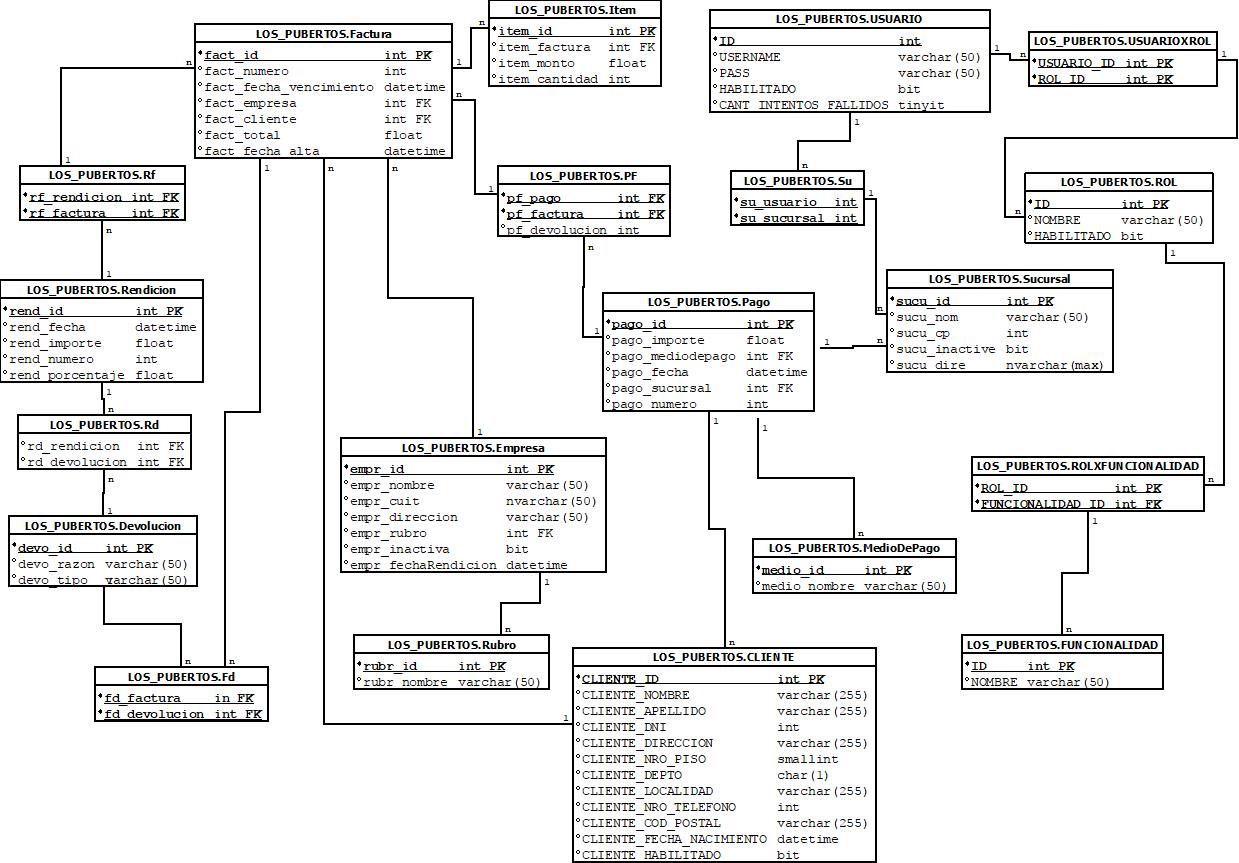
[Migración de datos 8](#_Toc496022518)

[Aplicación Desktop 9](#_Toc496022519)

## Introducción

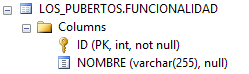
El presente documento detalla las decisiones de diseño que se tomaron durante el desarrollo del trabajo práctico.

## DER

**

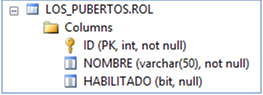
**Entidades**

### Tabla funcionalidades



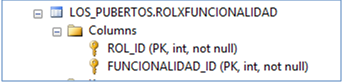
* Compone las funciones que estarán contenidas dentro de un rol en particular y permitirán al usuario que posea dicho rol, realizar diferentes acciones con estas funcionalidades.
* El nombre aparecerá en las ABM para que el usuario las reconozca.

### Tabla rol



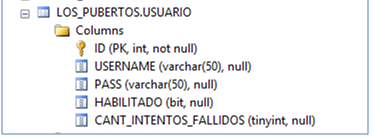
* Nombre del rol (Administrador, cobrador y el resto de los roles que sean creados en ABM Rol).
* Habilitado indica si ese rol se encuentra disponible para usar o también para cuando se desee dar de baja un rol activo.

### Tabla rolxfuncionaidad



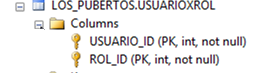
* Agrupa las funciones con los roles correspondientes. Se aplica ya que una función pueden pertenecer a varios roles. Y los roles poseen varias funciones a la vez.

### Tabla usuario



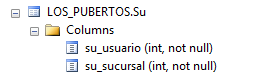
* Id identificación univoca del usuario. Para realizar búsquedas
* Username y pass se utilizaran para el logue del mismo al sistema.
* Username se usó el SHA\_256.
* Habilitado se ulitizar para registrar la baja de un usuario.
* Cant\_intentos\_fallidos este campo sirve por si el usuario excede la cantidad de intentos fallidos en el ingreso de la contraseña. Se ha considera que el usuario puede ingresar mal su contraseña hasta 3 veces, una vez completado las 3 veces el sistema dara de baja al usuario por lo que no podrá volver a ingresar. Antes de llegar a las 3 veces, en culqier momento en el que se vuelva a ingresar correctamente la clave el sistema borra la cantidad de intentos fallidos. Se toma esta decisión a la hora de programar el sistema.

### Tabla usuarioxrol



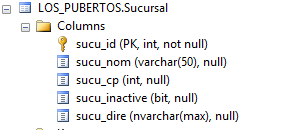
* Se asocia los usuarios a la cantidad de roles que se desea pueden ser mas de uno
* Y los roles se pueden asignar a varios usuarios.

### Tabla su



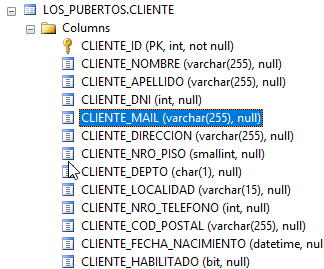
* Los usuarios pueden pertenecer a varias sucursales al mismo tiempo y seleccionaran la que quieren al momento de loguearse.
* Las sucursales tiene a su vez varios usuarios registrados.

### Tabla sucursal



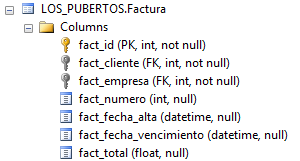
* Información perneciente a las sucursales
* Se inactivan con el campo inactive.
* No pueden tener el mismo cuit entre ellas.

### Tabla Cliente



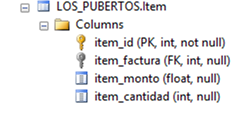
* Información de los clientes que se acerque al local a realizar pagos.
* Los mismos no pueden tener el mismo mail entre ellos. Para ello existen validaciones que no dejan poner mails repetidos.
* Todos los campos son obligatorios. A la hora de querer dar de alta un nuevo cliente el sistema te pedirá que llenes todos los campos del formulario.
* Se pueden dar de baja con el campo habilitado.

### Tabla Factura



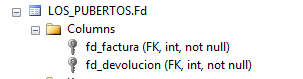
* Registros de las facturas a pagar.
* Cada factura pertenece a una empresa en particular.
* Y se le asociara un cliente.
* El numero de factura se conserva para no perder datos de la migración, para la identificación de las misma se usa el id.

### Tabla Item



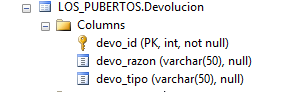
* las facturas están compuestas de varios ítem con sus propios datos

### Tabla Fd



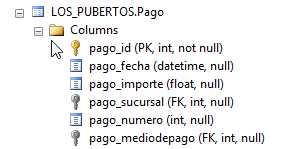
* las facturas pueden ser devueltas, de a una o varias.

### Tabla Devolución



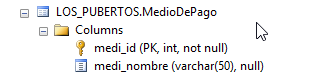
* se registra devoluciones por errores.
* Se registra la razón y el tipo de la misma.

### Tabla Pago



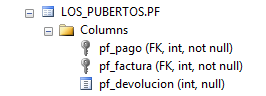
* Registros del pago.
* Se registra el numero pero no se tiene el cuenta para su identificación. Para eso se utiliza el campo id
* Los pagos pueden tener diferentes medios.
* Y estar compuestos por varias facturas.

### Tabla Medio de pago



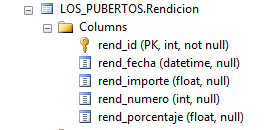
* Se registra los diferentes medios de pagos que existen.

### Tabla PF



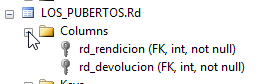
* Pagos facturas.
* Se utiliza para asociar los pagos a varias facturas.

### Tabla Rendición



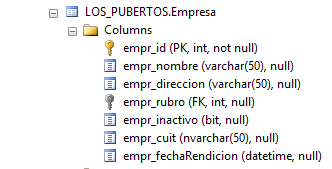
* Se registra la rendición de varias facturas de una empresa particular.
* El porcentaje representa lo que la empresa de pago se retiene del total.
* Se toma como decisión que ese porcentaje se estipula en el momento de efectuar las rendiciones. Por lo que el mismo puede variar en el tiempo.
* Se toma como decisión que las rendición solo estarán confeccionadas por facturas que correspondan al mimos mes en el que se ejecuta la rendición.

### Tabla Rd



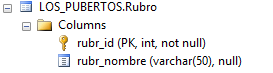
* Rendición devolución
* Se registra devoluciones de rendición

### Tabla Empresa



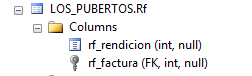
* Datos de las empresas asociadas al sistema de pago.
* El cuit no se pueden repetir. Por lo que cuando se efectúa una alta de empresa se controla que no se repita el cuit.
* Se pueden dar de baja con el campo inactivo
* Existe una validación que no permita dar de bala aquellas empresas que tienen rendiciones pendientes.
* Se toma como decisión que para el desarrollo del sistema las empresas acuerdan la fecha de rendición en el momento de darse de alta en el sistema. Esta fecha podrán ser elegidas por las empresas por lo que cada una de ella puede elegir la que más le convengan. No es necesario que sean todas el mismo día.

### Tabla Rubro



* Cada empresa pertenece a un rubro específico.

### Tabla Rf



* Factura rendición
* Las rendiciones están compuestas por varias facturas

## Migración de datos

La migración de datos se realizó haciendo inserts desde la tabla maestra. El script fue probado en una notebook hp pavilon windows 10 procesador AMD A10 RAM 12, tardó 0:20 segundos aproximadamente.

Se crean primero el DER correspondiente al sistema.

Lo primero que se realiza en el script es la verificación de que si existen los objetos se deben borrar.

Se pasa a crear las tablas con los campos que se necesitan para las nuevas funcionalidad y los campos de la tabla maestra.

Se crean las pk y fk de las todas las tablas.

Se pasan los datos de la tabla maestrar realizado insert. Los mismo estas compuestos por select y subselect.

Para el resto de las tablas se fueron haciendo insert usando joins entre la base maestra y los datos de las tablas que ya fuimos migrando satisfactoriamente.

**Validación de datos:**

Para agregar rendiciones y facturas a la base de datos se utilizó StoreProcedures.

Con datos tipos tabla para los ítem de los mismos.

Para el inicio de sesión se utiliza un StoreProcedure con los diferentes resultados:

* No existe el usuario
* Existe el usuario, no puede iniciar sesión por la cantidad de intentos fallidos
* El usuario pudo iniciar sesión
* El usuario ingreso mal la password, incremente la cantidad de intentos fallidos

Se utilizan trigger para el chequeo de datos:

* Cuit de empresas repetidas: no permite insertar empresas que posea un cuit de una empresa que ya se encuentre en el sistema. Esta validación también aplica a la hora de querer modificar una empresa.
* Inactivar una empresa: para poder dar de baja una empresa la misma no puede poseer facturas pendientes de rendición. Por lo que al momento de dar la de baja o querer modificar ese campo el sistema corrobora esa condición para permitirlo.
* Código postal de sucursales:no se permite que dos sucursales tengan el mismo código postal por lo que al momento de ingresar una nueva sucursal o bien de querer modificar una existente se controla que no se ingrese un cp igual a otra sucursal.
* Control del monto de pago: las facturas como los pagos no pueden poseer un monton menor o igual a cero.

**Datos Agregados para el funcionamiento del Sistema:**

Para la probar del sistema se agrega dos usuarios user1 como un cobrador y admin como administrador. A los mismos se le agrega una sucursal específica para que puedan ingresar al sistema.

También se agregan las funcionalidades y se las asigna a los roles correspondientes.

INSERT INTO LOS\_PUBERTOS.ROL (NOMBRE,HABILITADO)

VALUES ('Administrador',1),

('Cobrador',1)

GO

INSERT INTO LOS\_PUBERTOS.USUARIO(ID,USERNAME,PASS,HABILITADO,CANT\_INTENTOS\_FALLIDOS)

VALUES (1,'USER1','USER1',1,0),

(3,'admin','w23e',1,0)

INSERT INTO LOS\_PUBERTOS.Su(su\_sucursal,su\_usuario)

VALUES (1,1),

(1,3)

GO

INSERT INTO LOS\_PUBERTOS.USUARIOXROL (USUARIO\_ID,ROL\_ID)

VALUES (1,2),

(3,1),

(3,2)

GO

INSERT INTO LOS\_PUBERTOS.FUNCIONALIDAD(ID,NOMBRE)

VALUES (1,'ABM de Rol'),

(2,'Login y Seguridad'),

(3,'Registro de Usuario'),

(4,'ABM de Cliente'),

(5,'ABM de Empresa'),

(6,'ABM de Sucursal'),

(7,'ABM Facturas'),

(8,'Registro de Pago de Facturas'),

(9,'Rendicion de Facturas cobradas'),

(10,'Listado Estadistico'),

(11,'Devoluciones'),

(12,'Devolucion Factura'),

(13,'Listado Estadistico')

GO

INSERT INTO LOS\_PUBERTOS.ROLXFUNCIONALIDAD(ROL\_ID,FUNCIONALIDAD\_ID)

VALUES (1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(1,8),(1,9),(1,10),(1,11),(1,12),(1,13),

(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(2,7),(2,12),(3,1),(3,2),(3,3),(3,4),(3,5),(3,6),(3,7),(3,8),(3,9),(3,10),(3,11),(3,12),(3,13)

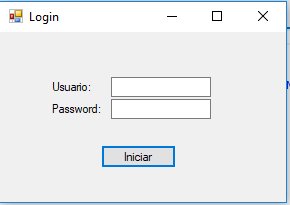
## Aplicación Desktop

Para el desarrollo de la aplicación se utiliza como diseño:

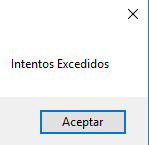
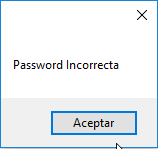
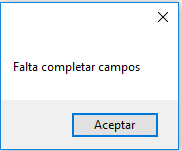
* DAO: Data Access Objects que encapsular la lógica la comunicación con la Base de Datos para realizar determinadas consultas, recupera, guardar y actualizar datos o correr procedimientos.
* Se podrá modificar la fecha del sistema en el archivo app.config.
* DTO Objeto de transferencia de datos, se utiliza para transferir los datos entre clases y módulos. DTO solo contiene campos privados para sus datos, getters, setters y constructores.
* Business: es un objeto inteligente que contiene datos y métodos que realizan operaciones.
* Entities

**Módulos Funcionales**

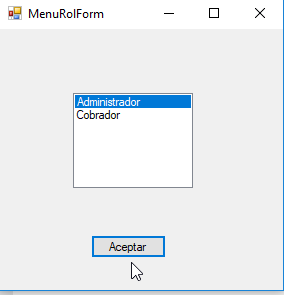
### Pantalla Login



* Pantalla para el ingreso al sistema.
* Actualmente existen 3 usuarios:
  + User1 contraseña user1
  + admin contraseña w23e
* existen pantalla para advertir al usuario la falta de datos en los campos y por ingresos de datos incorrectos.
* Si la contraseña se ingresa mal 3 veces sale la pantalla de usuario bloqueado



### Pantalla rol



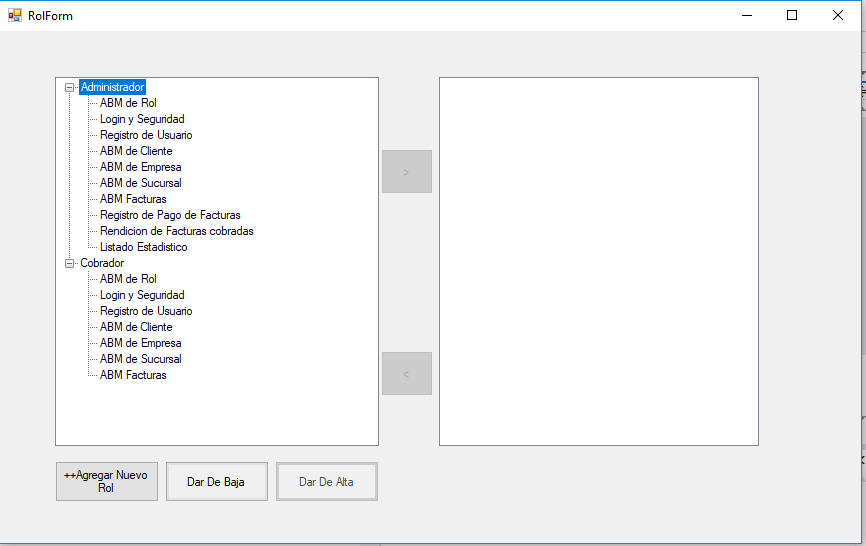
* En esta pantalla se mostraran los roles asignados a el usuario que ingreso.
* En este casa se encuentre 2 roles en el sistema: Administrador y cobrador
* Puede agregarse más roles desde ABM rol

### Pantalla de funcionalidades



* En esta pantalla se listan las funciones que están asignadas a roles.
* En este caso se muestra las funciones de un administrador.

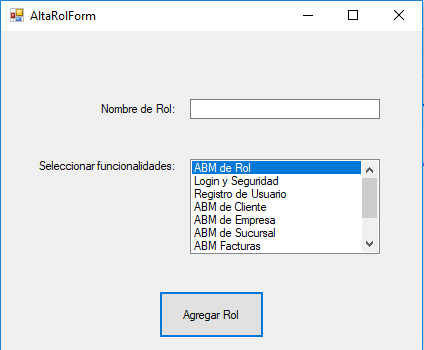
### ABM ROL



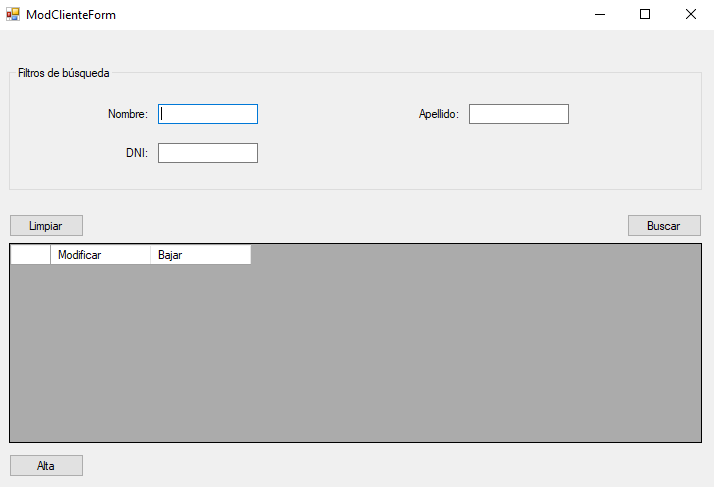
* Esta es la pantalla en la cual podemos CREAR, MODIFICAR, INACTIVAR un rol.
* Se muestra el nombre del rol con su lista de funciones debajo.
* Los roles están compuesto por el nombre y sus funciones.

Alta de un rol

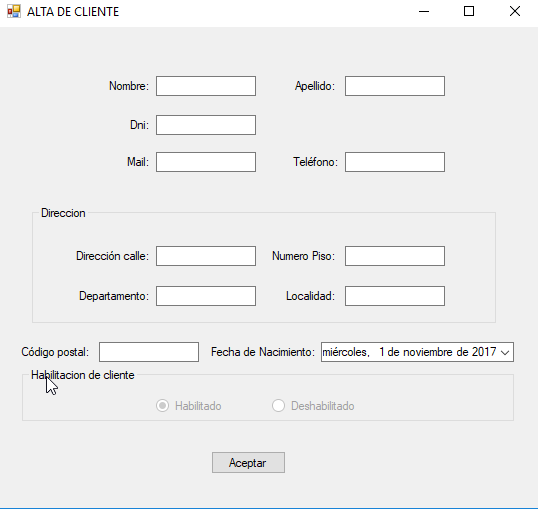
* todos los campos son obligatorios.



### ABM CLIENTE



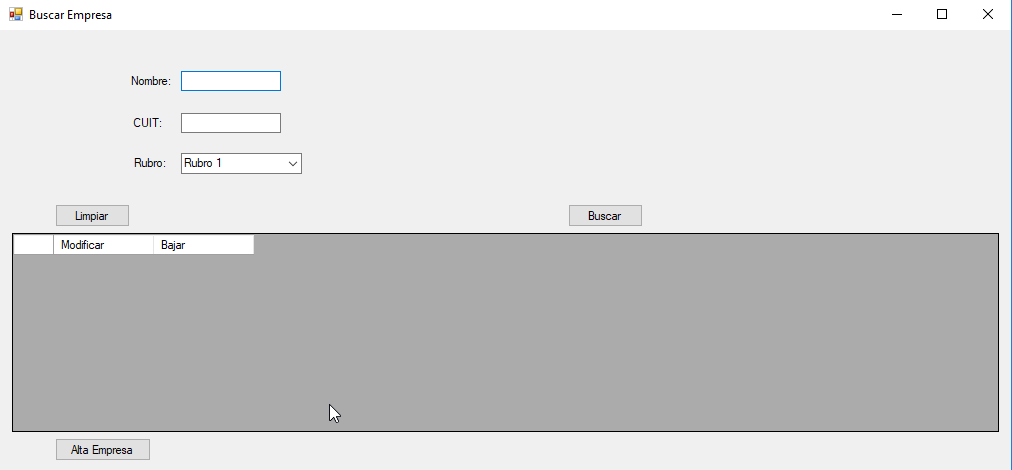
* Se presenta una pantalla donde se puede buscar un cliente a través de los filtros de DNI, nombre y apellido.
* Desde dicha búsqueda se puede realizar cambios en los datos del cliente o dar de baja al mismo.
* Se puede dar de alta un nuevo cliente en la sección de alta.

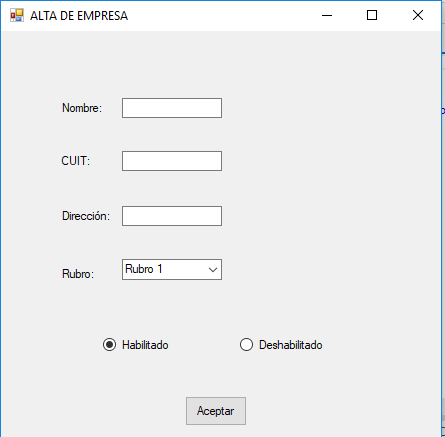


* Todos los datos son obligatorios.
* El mail es un dato que no debe repetirse.

### ABM EMPRESA

* Se utiliza esta pantalla para poder buscar una empresa particular, filtrando por nombre, cuit y rubro.
* Desde dicha búsqueda se puede seleccionar a la empresa para poder realizar cambios en sus datos o bien darla de baja.
* Existe también la función de crear una nueva empresa con alta de empresa.

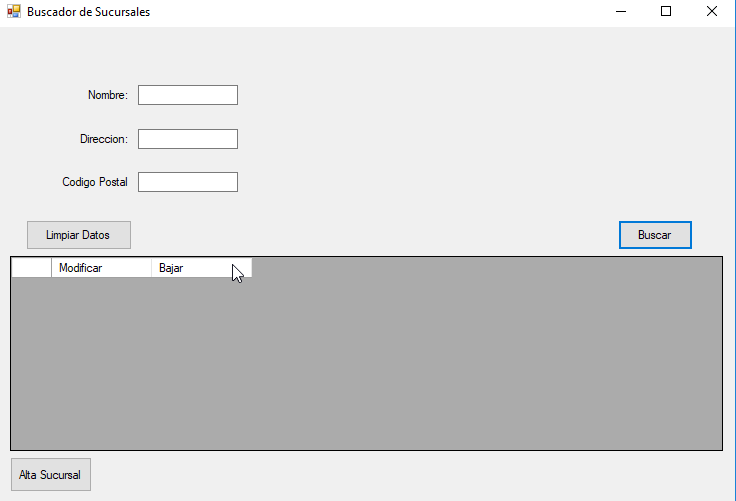




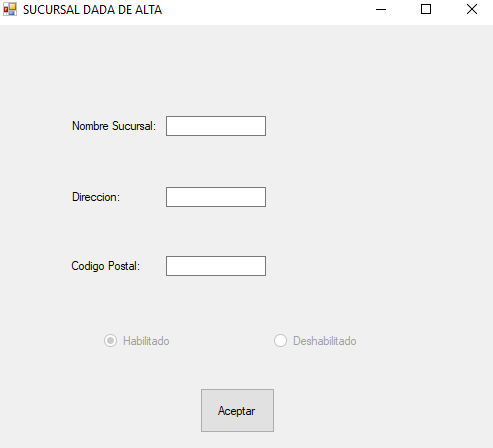
* Todos los datos son obligatorios y no puede haber dos empresas con el mismo cuit.

### ABM SUCURSAL

* Filtramos las sucursales por los campos Nombre, dirección y código postal.
* Una vez filtradas se pueden modificar o dar de baja.

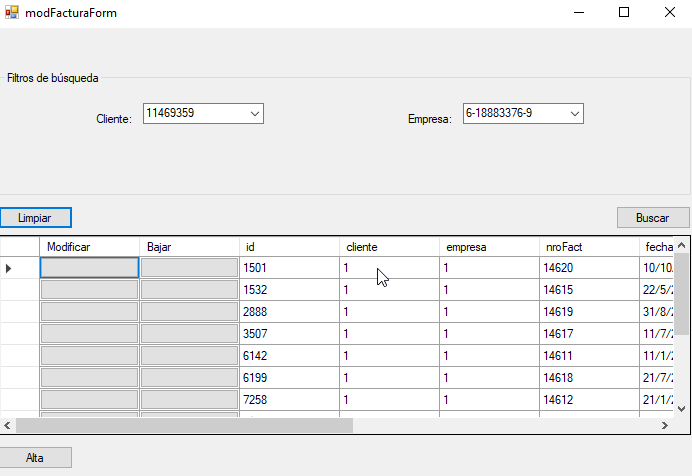


* Alta de una sucursal
* Nombre de la sucursal, dirección y código postal. El último campo no puede repetirse

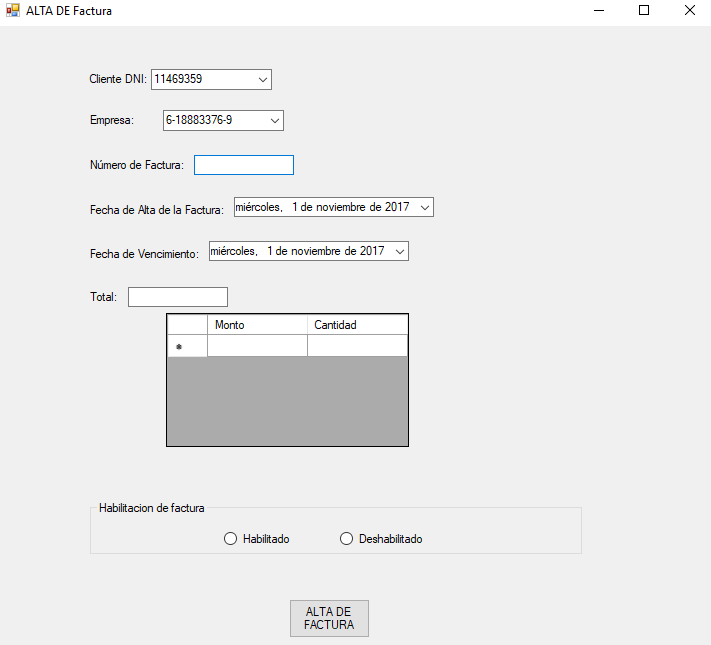


**ABM FACTURA**

* Filtramos las facturas a través de los campos cliente y empresa.
* Se muestra un listado con las facturas que cumple con el requisito.
* Se puede modificar los campos o dar de baja la factura utilizando los botones de la parte izquierda de la factura.

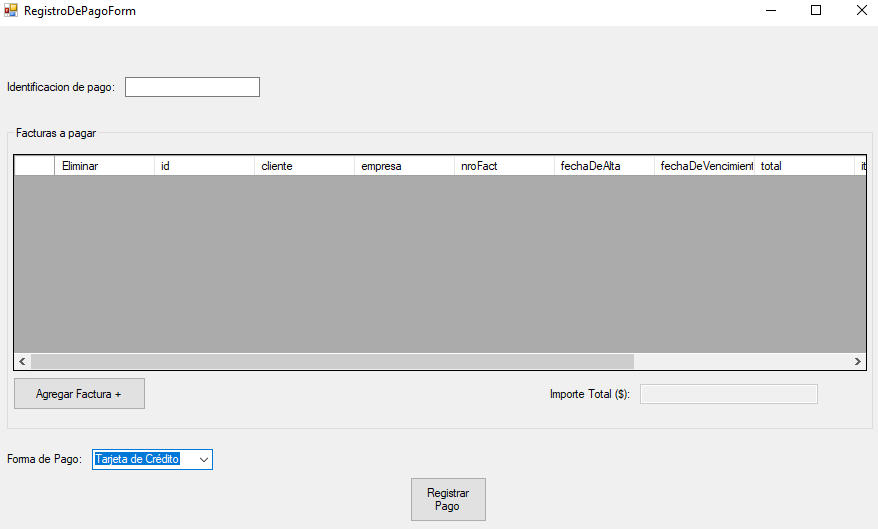


* Alta de una nueva factura
* Se muestra un formulario con todos los datos que debe poseer una factura.
* Una lista de ítem que compone a dicha factura

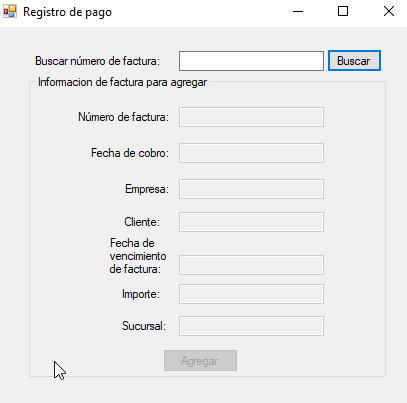


**ABM REGISTRO DE PAGO**

* El registro de pago de facturas pertenecientes al sistema.
* La primera vista del abm muestra una búsqueda de los pagos existentes.
* Para poder inhabilitar y modificar los mismo.
* Lo primero que se debe hacer es agregar las facturas que contendrán el pago con el botón “agregar factura”
* Una vez hecho de procede a registrar el pago con los datos correspondientes.

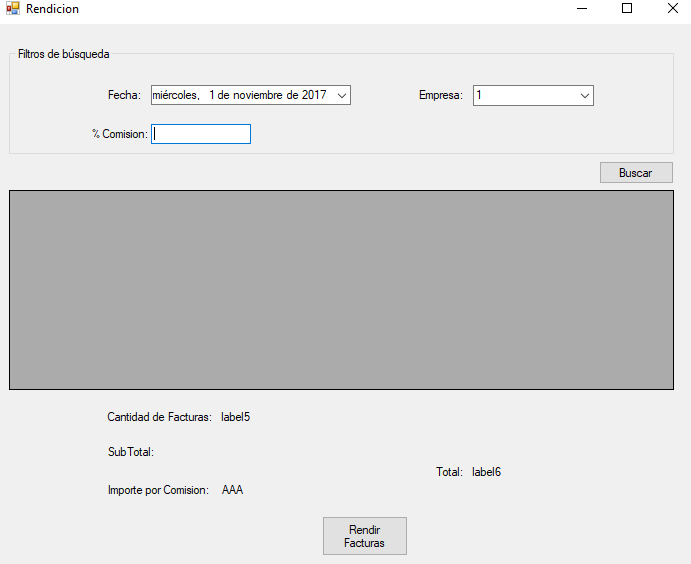


* Vista de la pantalla para buscar las facturas correspondientes al pago



**ABM RENDICION (registro de facturas cobradas)**

* Esta pantalla facilita la rendición de facturas que pertenezcan a una empresa en particular.
* También posee funciones de listado
* Y funciones para agregar nuevas rendiciones.



**Listado estadístico**

* Listado de los 5 clientes con más pagos, filtrados por la fecha y el trimestre al que pertenece. Cuenta la cantidad de pagos que se han registrado en nombre del cliente.LA FECHA SOLO USE EL AÑO
* Listados de 5 empresa con mayor monto rendido, según un años especifico y un trimestres. Se mide a través de la suma de los importes de las rendiciones hechas para esas empresas.
* Listado de las 5 empresas con mayor porcentaje de facturas cobradas en un año y trimestre especifico. Se considera que el hecho de que una factura es cobra quiere decir que se encuentran en la tabla rendición.
* Listado de los 5 clientes con el mayor porcentaje de facturas pagadas en un año y trimestre especifico. Las facturas pagadas con las que se encuentran en la tabla de pago, cuyo pago se encuentra asociado al cliente en cuestión.