

# TRABAJO PRÁCTICO GESTIÓN DE DATOS ESTRATEGIA

**Grupo:** *SERVOMOTOR* (Numero 12)

**Integrantes:**

- Ayala**, Ignacio (156.106-6)
- Chen**, Fernando (155.592-3)
- Zeitune**, Matias (156.845-0)

AÑO: 2017

2° Cuatrimestre

UTN FRBA

## **Índice**

Pág. 3 .....	Estructuras
Pág. 3.....	Estados de habilitación/activación
Pág. 4.....	Operaciones
Pág. 5.....	Normalización
Pág. 5.....	Triggers
Pág. 5.....	Relación muchos a muchos
Pág. 7.....	Consideraciones

# ESTRATEGIA

## ***ESTRUCTURAS***

### USUARIOS Y SUCURSALES

El enunciado del trabajo nos indica que un usuario tiene asociado muchas sucursales, es por eso que en nuestro modelo de datos decidimos separar estas dos entidades.

Un usuario puede ser tanto un administrador el cual tiene permitido realizar todo tipo de funcionalidad y un cobrador el cual está restringido a ciertos roles y funcionalidades.

### FACTURAS

Un cliente tiene asociadas varias facturas y pagos , donde si tomamos en cuenta que la tabla facturas tiene un campo llamado total el cual esta de normalizado y pre calculado el cual se actualiza automáticamente mediante un trigger, como se explicara mas adelante , de manera sencilla, si este campo no estuviera en la tabla Factura , cada vez que quisiéramos consultar el total de una factura , tendríamos que calcular en el momento su valor , teniendo que joinear con todas la tabla items ,por lo tanto nos pareció mas performante dejar este dato denormalizado antes que calcularlo por cada vez que se necesite.

Tiene un campo denominado ESTADO, el cual puede ser “NO PAGA” (estado en el que se encuentra una factura que se da de alta), “PAGA” (cuando forma parte de un pago en particular), “DEVUELTA” (la factura fue devuelta estando PAGA) y “RENDIDA” (cuando forma parte de una rendición).

## ***ESTADOS DE HABILITACION/ACTIVACION***

Algunas estructuras como los clientes, facturas, empresas, usuarios y sucursales tienen un campo llamado “estado habilitación” o ”estado habilitación” que simplemente se almacena un 1 cuando la estructura se encuentra habilitada o un 0(cero) cuando no lo está. De esta forma podemos considerar mostrar o no los datos que el usuario quiera consultar (se habla más de esto en la [consideración 1](#)).

# ***OPERACIONES***

En las operaciones de este trabajo práctico encontramos:

## **Pago de facturas.**

Aquí se pueden elegir el conjunto de facturas para poder realizar un único pago restringiendo las facturas a mostrar ([ver consideración 1](#)). La tabla PAGOS se relaciona con CLIENTES porque otra persona que no sea el cliente de la factura propiamente dicha puede efectuar el pago (alguien puede pagar una boleta a nombre de otra persona), de esta manera ganamos más flexibilidad al sistema.

## **Devolución de facturas.**

En esta pantalla se pueden elegir una o muchas facturas a devolver, restringiendo las facturas a mostrar ([ver consideración 1 y 3](#)). Decidimos la relación muchos a muchos porque las facturas se pueden devolver más de una vez y por diversos motivos. En cuanto a lo último decidimos colocar 2 motivos descriptos en el enunciado del TP:

- “Error de cobro”
- “Cliente retrotajo pago”

## **Rendición de facturas a una empresa.**

Esta funcionalidad permite rendir una o varias facturas para una empresa en una fecha determinada y por única vez en un mes restringiendo las facturas a mostrar ([ver consideración 1](#)). Tuvimos una consideración importante en el modo de establecer la fecha de rendición para una empresa que se puede ver en la [consideración 3](#).

## ***NORMALIZACIÓN***

Decidimos normalizar los datos de los rubros en una tabla aparte, especificando el código del rubro y su respectivo nombre. Lo que nos llevó a esta decisión fue que sería más fácil poder listar en un componente de UI los rubros disponibles para asignar a una empresa, ya que así no se reflejarían inconsistencias en la aplicación.

El mismo razonamiento se tuvo con la tabla Medios de pago. Normalizando los medios de pago, se tienen registros con el código de ellas y sus nombres, logrando un mantenimiento de datos más sencillo en caso de cambios. A su vez, ya que el enunciado pedía que pudiesen incorporarse de manera simple otros tipos de medios de pago, al tener los datos normalizados, este requerimiento se cumple insertando otra fila a la tabla, detallando el código y el nombre.

## ***TRIGGERS***

Se considero la utilización de un trigger para la actualización del total de las facturas por cada vez que se agrega un ítem, esto viene atado a la modificación de la factura. El trigger actúa sobre la inserción de la tabla items realizando las operaciones correspondientes.

También se considero un trigger para verificar que al momento de dar de baja una empresa la misma no tenga facturas rendidas si esto es así se rehabilita la empresa en cuestión y se informa debidamente al usuario de otro modo se da efectiva la baja lógica

## ***RELACIÓN MUCHOS A MUCHOS***

Existe la relación muchos a muchos entre devoluciones y facturas, debido a que estas mismas pueden llegar a ser devueltas múltiples veces por razones totalmente diferentes, es por esto que se detalla el motivo de devolución en dicha tabla que une a ambas entidades. Por otro lado una devolución puede estar formada por varias facturas según lo propuesto en el enunciado.

Otra relación muchos a muchos que existe sobre nuestro dominio es la de usuarios con sucursales, ya que en el momento del login un cobrador puede elegir la sucursal entre las que tenga habilitadas para la cual tendrán efecto sus acciones (mientras tenga permisos para esa sucursal, es decir que ese insert para ese usuario en la tabla que conecta a ambas entidades), mientras que un administrador puede tener efecto

sobre cualquier sucursal. Del otro lado, una sucursal puede cobrar facturas de muchos usuarios logeados.

## ***CONSIDERACIONES***

**(1)** . Decidimos NO mostrar aquellos datos que se encuentren inhabilitados o inactivos como ser:

- Facturas a pagar (estado “NO PAGA”) ya vencidas en la pantalla de pagos.
- Facturas ya rendidas (estado “RENDIDA”) en la pantalla de devoluciones.
- Cualquier tipo de factura que tenga una empresa inhabilitada.
- Cualquier tipo de factura que tenga un cliente inhabilitado.
- Cualquier tipo de factura que tenga una sucursal inhabilitada.

Esta decisión la tomamos por el motivo de que las consultas van a ser mas performantes al traer un menor volumen de datos y así poder evitar engorrosas validaciones complejizando la aplicación. También podemos argumentar que el cobrador va a estar siempre con una factura o un conjunto de ellas a mano, lo cual al no poder observar dicha/s factura/s puede ir a cualquier ABM buscando los datos pertinentes para poder corroborar la situación.

**(2)** . En la pantalla de RENDICIONES atendiendo al enunciado, consideramos que las rendiciones no tienen una fecha fija para cada empresa desde el momento en que se le da el alta, por ello, decidimos que al momento de ingresar una rendición se debe poner la fecha de la misma, corroborando y validando adecuadamente que a esa empresa no se la haya rendido en el mes de la fecha ingresada. Al momento de ingresar la fecha y la empresa, luego de validar lo explicado anteriormente, se mostraran las facturas PAGAS en ese mes para la empresa seleccionada.

**(3)** . Consideramos que las facturas ya RENDIDAS no pueden ser DEVUELTAS.

**(4)** . Consideramos que las funcionalidades de cada rol quedan automáticamente restringidas en el menú principal (deshabilitando cada botón según las funcionalidades existentes) para así evitar validaciones en cada pantalla enviando por constructor a cada una el rol y preguntando si puede acceder o no.

**(5)** . En algunas ABM consideramos la misma pantalla para baja lógica y modificación. Nuestro motivo por el cual se tomo esta decisión es para que el usuario cuando seleccione algún elemento de la tabla, tenga habilitado tanto modificar sus datos como dar de baja dicho elemento, reutilizando ese dataGrid o tabla.

**(6)** Consideramos que las PKs naturales no se pueden modificar en la aplicación (por ejemplo el dni del cliente , el numero de factura , el cuit de una empresa ) ya que generaría inconsistencias en el resto del modelado de la base de datos.

**(7) Usuarios:**

User	Password	Tipo
admin	w23e	Administrador
cobrador	w23e	Cobrador
usuarioX	w23e	Cobrador

**(8)** . En ABM de empresas consideramos que la CUIT ingresada por el usuario debe ser estrictamente chequeada para garantizar su validez. Por lo tanto, se realizan las siguientes verificaciones:

- La CUIT debe tener 11 dígitos (sin contar los guiones).
- El tipo de la CUIT (primeros 2 dígitos) solo puede ser 30, 33 y 34. (Estos dígitos indican que la CUIT corresponde a una empresa).
- Se realiza el [cálculo](#) para obtener el dígito verificador y lo compara con el último dígito de la CUIT ingresada.