**ESTRATEGIA GRUPO PERO\_COMPILA**

En principio creamos **tres tipos de usuarios**, **admin** (requerido por la cátedra), **cobrador** psw: cobrador y **admingral** psw: admingral , siendo éste último un usuario con ambos roles para que se pueda probar la funcionalidad de que un usuario posea más de un rol

Cambiamos el DER , modificando la tabla Cliente en la cual pusimos como clave compuesta al mail junto con el dni para que puedan existir distintas personas (dni) pero el mail será el mismo, identificando de manera únivoca al registro, preferimos este modo de uso debido a que si no teníamos que realizar cambios sobre los datos otorgados por la cátedra.

En la entidad empresa, como bien pusimos en el DER, para la no existencia de empresas con un mismo cuil pusimos ese campo en UNIQUE

En la entidad sucursal, no habíamos contemplado que: “Bajo ninguna circunstancia puede haber más de una sucursal por código postal existente.”, por ende, decidimos agregar como UNIQUE el código postal.

Para el registro de pagos, nuestra forma de corroborar que la empresa está activa es filtrar directamente las activas, sin darle la opción al usuario de elegir una empresa no activa

Luego… en el registro de pagos, el usuario puede filtrar por alguno o todos los campos que crea conveniente para buscar la factura a cobrar. Tenemos en cuenta que… Cuando se acerca un cliente a pagar una o varias facturas, solo se le cobrarán facturas que estén hechas con su DNI, esto lo decidimos más que todo para el listado estadístico en la parte que se solicita en el enunciado:

“Clientes con más pagos”

“Clientes con mayor porcentaje de facturas pagadas (clientes cumplidores).”

Lo consideramos así porque si no a la hora de realizar el pago eligiendo cualquier medio de pago, automáticamente cargamos el DNI proveniente del cliente que se acerca a pagar, siendo requerido desde un principio al registrar el pago y de forma obligatoria, esto nos facilita a la hora de ver los clientes con más pagos, sino otro cliente puede venir a pagar facturas de otros clientes

Por otro lado, cuando registramos un pago lo hacemos sobre el total (ya sea migrado o calculado en el alta como la suma de los ítems) sin tener en cuenta si ese total está bien calculado, esto lo decidimos así porque a la hora de crear las Devoluciones dice explícitamente en el enunciado:

“La razón por la cual se desarrolla esta funcionalidad puede ser por errores de cobro o simplemente que el cliente decida retrotraer el pago efectuado.”

Por ende, decidimos que la devolución debe darse luego de que esté mal efectuado un pago.

En la entidad Factura agregamos el campo enviadoAPago para el momento en el que se realice una devolución vuelva a estar activa para pagarse.

Para los medios de pago, usamos los existentes provistos por la tabla maestra y por cada clase pusimos algunos datos relevantes, pero nosotros asumimos que una persona puede ir a pagar una o varias facturas de diferentes personas y a su vez pagar sin necesidad que sea propietario de esa o esas facturas.

Agregue estos procedure:

create procedure [pero\_compila].[sp\_get\_empresas]

as

begin

select \* from pero\_compila.Empresa where empresa\_estado=1

end

create procedure pero\_compila.sp\_get\_clientes

as

begin

select \* from pero\_compila.Cliente where cliente\_estado=1

end

IF EXISTS (SELECT name FROM sysobjects WHERE name='sp\_alta\_factura')

DROP PROCEDURE [pero\_compila].sp\_alta\_factura

GO

create procedure [pero\_compila].sp\_alta\_factura(

@clienteID int,

@empresaID int,

@codFactura int,

@total int,

@fecha\_alta datetime,

@fecha\_vencimiento datetime,

@habilitado numeric(1,0))

as

begin

insert into pero\_compila.Factura (factura\_cliente,factura\_empresa,factura\_cod\_factura,factura\_total,factura\_fecha\_alta,factura\_fecha\_vencimiento)

values (@clienteID,@empresaID,@codFactura,@total,@fecha\_alta,@fecha\_vencimiento,@habilitado)

end

GO

Fijarse si la de arriba esta bien ¿?

Créate procedure [pero\_compila].[sp\_alta\_factura]

(@cliente\_dni numeric(18,0),@cliente\_mail nvarchar(255),

@empresaId int,

@cod\_factura nvarchar(255),

@fecha\_alta datetime,@fecha\_vencimiento datetime,

@total decimal(18,2))

as

begin

insert into pero\_compila.Factura ([factura\_cliente\_dni],[factura\_cliente\_mail],[factura\_empresa],[factura\_cod\_factura],

[factura\_fecha\_alta],[factura\_fecha\_vencimiento],[factura\_total])

values(@cliente\_dni,@cliente\_mail,@empresaId,@cod\_factura,

@fecha\_alta,@fecha\_vencimiento,@total)

end