Trabajo Práctico – Gestión de Datos

**Nombre Proyecto:** Palconet.

**Integrantes:** Canaviri Rodrigo, Corti Gerónimo, Torrico Juan Manuel.

**Fecha de Entrega:** 5 de diciembre de 2018.

**Ciclo Lectivo:** 2018.

Contenido

[Decisiones de Arquitectura de la Base de Datos 6](#_Toc531890729)

[Funcionalidad 6](#_Toc531890730)

[Formato 6](#_Toc531890731)

[Descripción 6](#_Toc531890732)

[Rol 6](#_Toc531890733)

[Formato 6](#_Toc531890734)

[Descripción 6](#_Toc531890735)

[Rol Funcionalidad 6](#_Toc531890736)

[Formato 6](#_Toc531890737)

[Descripción 7](#_Toc531890738)

[Usuario 7](#_Toc531890739)

[Formato 7](#_Toc531890740)

[Descripción 7](#_Toc531890741)

[Usuario Rol 7](#_Toc531890742)

[Formato 7](#_Toc531890743)

[Descripción 8](#_Toc531890744)

[Cliente 8](#_Toc531890745)

[Formato 8](#_Toc531890746)

[Descripción 8](#_Toc531890747)

[Puntuación 9](#_Toc531890748)

[Formato 9](#_Toc531890749)

[Descripción 9](#_Toc531890750)

[Premio 9](#_Toc531890751)

[Formato 9](#_Toc531890752)

[Descripción 9](#_Toc531890753)

[Usuario Premio 10](#_Toc531890754)

[Formato 10](#_Toc531890755)

[Descripción 10](#_Toc531890756)

[Empresa 10](#_Toc531890757)

[Formato 10](#_Toc531890758)

[Descripción 11](#_Toc531890759)

[Publicación 11](#_Toc531890760)

[Formato 11](#_Toc531890761)

[Descripción 11](#_Toc531890762)

[Rubro 11](#_Toc531890763)

[Formato 11](#_Toc531890764)

[Descripción 12](#_Toc531890765)

[Estado Publicación 12](#_Toc531890766)

[Formato 12](#_Toc531890767)

[Descripción 12](#_Toc531890768)

[Grado Publicación 12](#_Toc531890769)

[Formato 12](#_Toc531890770)

[Descripción 12](#_Toc531890771)

[Ubicación 12](#_Toc531890772)

[Formato 12](#_Toc531890773)

[Descripción 13](#_Toc531890774)

[Compra 13](#_Toc531890775)

[Formato 13](#_Toc531890776)

[Descripción 13](#_Toc531890777)

[Factura 13](#_Toc531890778)

[Formato 13](#_Toc531890779)

[Descripción 14](#_Toc531890780)

[Ítem 14](#_Toc531890781)

[Formato 14](#_Toc531890782)

[Descripción 14](#_Toc531890783)

[Datos Generales de la aplicación 15](#_Toc531890784)

[Login y Seguridad 18](#_Toc531890785)

[ABM de Rol (ADMIN) 20](#_Toc531890786)

[Crear Rol 20](#_Toc531890787)

[Editar o Borrar un rol 21](#_Toc531890788)

[Registro de Usuario 23](#_Toc531890789)

[ABM Clientes (ADMIN) 25](#_Toc531890790)

[Crear Clientes 25](#_Toc531890791)

[Editar o Borrar Clientes 25](#_Toc531890792)

[ABM de Empresas (ADMIN) 28](#_Toc531890793)

[Crear Empresa 28](#_Toc531890794)

[Editar o Borrar Empresas 28](#_Toc531890795)

[ABM de Rubro (ADMIN) 31](#_Toc531890796)

[ABM Grado Publicación 32](#_Toc531890797)

[Crear Grado 32](#_Toc531890798)

[Editar o Eliminar Grado 32](#_Toc531890799)

[Generar Publicación 34](#_Toc531890800)

[Editar o Publicar una Publicación 36](#_Toc531890801)

[Comprar 39](#_Toc531890802)

[Historial Cliente 41](#_Toc531890803)

[Canje y Administración de puntos 42](#_Toc531890804)

[Criterio de puntos asociados a compras 42](#_Toc531890805)

[Canje de puntos 42](#_Toc531890806)

[Visualización de Premios 42](#_Toc531890807)

[Generar Rendiciones 44](#_Toc531890808)

[Listado Estadístico 46](#_Toc531890809)

[DER 47](#_Toc531890810)

[47](#_Toc531890811)

# Decisiones de Arquitectura de la Base de Datos

Para tener en cuente, se crearon primeros las estructuras de las tablas con las restricciones que no generaban conflictos con otras tablas (único, clave primaria, etc) y luego de creadas las tablas, se agregaron las restricciones de relación (

## Funcionalidad

### Formato

create table PLEASE\_HELP.Funcionalidad

(

Func\_Id int identity(1,1),

Func\_Desc nvarchar(50) NOT NULL,

CONSTRAINT UQ\_FUNCIONALIDAD\_DESC UNIQUE (Func\_Desc),

CONSTRAINT PK\_FUNCIONALIDAD\_ID PRIMARY KEY (Func\_Id)

)

### Descripción

En esta tabla almacenamos las funcionalidades descritas en el enunciado del Trabajo Práctico, con sus respectivos Id. Al no estar estos datos en la tabla maestra, realizamos el insertado de los mismos desde el script de migración.

El id es autoincremental, y la descripción es no nula y única en toda la tabla.

## Rol

### Formato

create table PLEASE\_HELP.Rol

(

Rol\_Id int identity(1,1),

Rol\_Nombre nvarchar(20) NOT NULL,

Rol\_Habilitado bit DEFAULT 1,

Rol\_Baja bit DEFAULT 0,

CONSTRAINT UQ\_ROL\_NOMBRE UNIQUE (Rol\_Nombre),

CONSTRAINT PK\_ROL\_ID PRIMARY KEY (Rol\_Id)

)

### Descripción

Almacenamos los roles administrativo, cliente y empresa con sus respectivos Id (clave primaria) y nombre desde el script de migración. Además se agregan dos campos adicionales para la habilitación de un rol, y la baja del mismo.

## Rol Funcionalidad

### Formato

create table PLEASE\_HELP.Rol\_Funcionalidad

(

Rol\_Id int,

Func\_Id int,

CONSTRAINT PK\_ROLXFUNC PRIMARY KEY (Rol\_Id, Func\_Id)

)

ALTER TABLE PLEASE\_HELP.Rol\_Funcionalidad

ADD CONSTRAINT FK\_ROLXFUNCIONALIDAD\_ROL FOREIGN KEY (Rol\_Id) REFERENCES PLEASE\_HELP.Rol(Rol\_Id),

CONSTRAINT FK\_ROLXFUNCIONALIDAD\_FUNC FOREIGN KEY (Func\_Id) REFERENCES PLEASE\_HELP.Funcionalidad(Func\_Id)

### Descripción

Debido a que un rol puede tener más de una funcionalidad y una funcionalidad puede pertenecer a más de un rol, decidimos crear una tabla intermedia con la correspondiente relación entre estas dos entidades. Define las funcionalidades que tiene cada rol del sistema. Claramente, las claves foráneas representan esto último mencionado.

## Usuario

### Formato

create table PLEASE\_HELP.Usuario

(

Usuario\_Id int identity(1,1),

Usuario\_Username nvarchar(50) NOT NULL,

Usuario\_Password varbinary(255) NOT NULL,

CONSTRAINT UQ\_USUARIO\_USERNAME UNIQUE (Usuario\_Username),

CONSTRAINT PK\_USUARIO\_ID PRIMARY KEY (Usuario\_Id)

)

### Descripción

Los usuarios de nuestro sistema, tanto sea cliente, empresa, admin o cualquier otro tipo de usuario que se cree a posteriori, tienen datos en común como el nombre de usuario y la contraseña. Decidimos crear la tabla Usuario para almacenar los mismos y una tabla propia para cada rol donde se reflejan sus datos particulares, excepto para el administrador que no posee datos adicionales.

Por ello, la tabla tiene su id autoincremental. El usuario único, y la contraseña del tipo varbinary para almacenarla encriptada.

## Usuario Rol

### Formato

create table PLEASE\_HELP.Usuario\_Rol

(

Usuario\_Id int,

Rol\_Id int,

CONSTRAINT PK\_USUARIOXROL PRIMARY KEY (Usuario\_Id, Rol\_Id)

)

ALTER TABLE PLEASE\_HELP.Usuario\_Rol

ADD CONSTRAINT FK\_USUARIOXROL\_USUARIO FOREIGN KEY (Usuario\_Id) REFERENCES PLEASE\_HELP.Usuario(Usuario\_Id),

CONSTRAINT FK\_USUARIOXROL\_ROL FOREIGN KEY (Rol\_Id) REFERENCES PLEASE\_HELP.Rol(Rol\_Id)

### Descripción

Cada usuario puede tener más de un rol asignado y cada rol puede pertenecer a más de un usuario, razón por la cual se creó esta tabla intermedia con los Id de Usuarios y Roles. Define los roles que tiene cada usuario del sistema. Las claves foráneas representan esta relación.

Cliente

### **Formato**

create table PLEASE\_HELP.Cliente

(

Cli\_Usuario int,

Cli\_Nombre nvarchar(255),

Cli\_Apellido nvarchar(255),

Cli\_Tipo\_Documento nvarchar(255),

Cli\_Nro\_Documento numeric(18,0),

Cli\_Cuil numeric(11,0),

Cli\_Email nvarchar(255),

Cli\_Telefono numeric(15,0),

Cli\_Localidad nvarchar(255),

Cli\_Direccion nvarchar(255),

Cli\_Nro\_Piso numeric(18,0),

Cli\_Depto nvarchar(255),

Cli\_Cod\_Postal nvarchar(255),

Cli\_Fecha\_Nac datetime,

Cli\_Fecha\_Creacion datetime,

Cli\_Tarjeta\_Credito nvarchar(255),

Cli\_Habilitado bit DEFAULT 1,

Cli\_Intentos\_Fallidos smallint DEFAULT 0,

Cli\_Baja bit DEFAULT 0,

Cli\_Primer\_Login bit DEFAULT 0,

CONSTRAINT PK\_CLIENTE\_USUARIO PRIMARY KEY (Cli\_Usuario),

CONSTRAINT UQ\_CLIENTE\_CUIL UNIQUE (Cli\_Cuil),

CONSTRAINT UQ\_CLIENTE\_NRO\_DOCUMENTO UNIQUE (Cli\_Nro\_Documento)

)

ALTER TABLE PLEASE\_HELP.Cliente

ADD CONSTRAINT FK\_CLIENTE\_USUARIO FOREIGN KEY (Cli\_Usuario) REFERENCES PLEASE\_HELP.Usuario(Usuario\_Id)

### Descripción

Almacena los datos de cada cliente en el sistema. Consideramos al Id del Cliente como clave primaria y clave foránea al mismo tiempo y ésta última referencia a la tabla usuario donde se encuentra su usuario y contraseña.

Además de los atributos personales del cliente incluimos Cli\_Habilitado (para indicar la inhabilitación del usuario en caso de que ingrese la erróneamente la contraseña tres veces seguidas), Cli\_Intentos\_Fallidos (para registrar la cantidad de intentos fallidos que comete el usuario), Cli\_Baja (define si el usuario fue dado de baja por el administrador) y Cli\_Primer\_Login (para saber si es la primera vez que el usuario entra al sistema para mostrarle la ventana de cambio de contraseña).

El número de cuil y número de documento son únicos ya que no puede existir dos usuarios con mismo cuil o número de documento.

Puntuación

### **Formato**

create table PLEASE\_HELP.Puntuacion

(

Puntuacion\_Id int identity(1,1),

Puntuacion\_Cantidad int DEFAULT 0,

Puntuacion\_Fecha\_Vencimiento datetime,

Puntuacion\_Cliente int,

CONSTRAINT PK\_PUNTUACION\_ID PRIMARY KEY (Puntuacion\_Id)

)

ALTER TABLE PLEASE\_HELP.Puntuacion

ADD CONSTRAINT FK\_PUNTUACION\_CLIENTE FOREIGN KEY (Puntuacion\_Cliente) REFERENCES PLEASE\_HELP.Cliente(Cli\_Usuario)

### Descripción

Esta tabla está relacionada con la tabla Cliente asumiendo que cada cliente tiene registros con cantidades de puntos y sus vencimientos, estos puntos los adquiere al realizar compras de entradas y según el precio de las mismas adquiere una determinada cantidad. Para este fin consideramos que la cantidad de puntos a adquirir es el resultado de realizar la división entera entre el precio de la compra y el valor 10, y con un vencimiento de un año a partir de la fecha de compra. Es decir que un cliente puede tener varios conjuntos de puntos que vencen en distinto día.

La tabla cuenta con el identificador autoincremental, la cantidad de puntos, la fecha de vencimiento de los mismos y el cliente el cual cuenta con esos puntos.

## Premio

### Formato

create table PLEASE\_HELP.Premio

(

Premio\_Id int identity(1,1),

Premio\_Descripcion nvarchar(50),

Premio\_Puntos int,

CONSTRAINT PK\_PREMIO\_ID PRIMARY KEY (Premio\_Id)

)

### Descripción

Los puntos obtenidos por los clientes son utilizados para canjearlos por premios, a tal fin incluimos la tabla Premio que detalla la descripción de los mismos y los puntos necesarios para adquirirlos.

Al no existir premios en la tabla maestra, insertamos registros de premios por default en el script de migración. Consideramos que el stock de los mismo es ilimitado.

## Usuario Premio

### Formato

create table PLEASE\_HELP.Usuario\_Premio

(

Cli\_Usuario int,

Premio\_Id int,

Premio\_Cantidad int DEFAULT 1,

CONSTRAINT PK\_USUARIOXPREMIO PRIMARY KEY (Cli\_Usuario, Premio\_Id)

)

ALTER TABLE PLEASE\_HELP.Usuario\_Premio

ADD CONSTRAINT FK\_USUARIOXPREMIO\_CLIENTE FOREIGN KEY (Cli\_Usuario) REFERENCES PLEASE\_HELP.Cliente(Cli\_Usuario),

CONSTRAINT FK\_USUARIOXPREMIO\_PREMIO FOREIGN KEY (Premio\_Id) REFERENCES PLEASE\_HELP.Premio(Premio\_Id)

### Descripción

Tabla intermedia de la relación muchos a muchos entre las tablas Cliente y Premio. Define los premios que obtuvieron los clientes y la cantidad correspondiente (se considera un stock **ILIMITADO**).

## Empresa

### Formato

create table PLEASE\_HELP.Empresa

(

Emp\_Usuario int,

Emp\_Razon\_Social nvarchar(255),

Emp\_Email nvarchar(50),

Emp\_Telefono numeric(15,0),

Emp\_Localidad nvarchar(255),

Emp\_Direccion nvarchar(50),

Emp\_Piso numeric(18,0),

Emp\_Depto nvarchar(50),

Emp\_Cod\_Postal nvarchar(50),

Emp\_Ciudad nvarchar(255),

Emp\_Cuit nvarchar(255),

Emp\_Habilitado bit DEFAULT 1,

Emp\_Intentos\_Fallidos smallint DEFAULT 0,

Emp\_Baja bit DEFAULT 0,

Emp\_Primer\_Login bit DEFAULT 0,

CONSTRAINT PK\_EMPRESA\_USUARIO PRIMARY KEY (Emp\_Usuario),

CONSTRAINT UQ\_EMPRESA\_CUIT UNIQUE (Emp\_Cuit),

CONSTRAINT UQ\_EMPRESA\_RAZON\_SOCIAL UNIQUE (Emp\_Razon\_Social)

)

ADD CONSTRAINT FK\_EMPRESA\_USUARIO FOREIGN KEY (Emp\_Usuario) REFERENCES PLEASE\_HELP.Usuario(Usuario\_Id)

### Descripción

Al igual que con los clientes, ésta tabla guarda los datos particulares de la empresa, además los atributos Emp\_Habilitado, Emp\_Intentos\_Fallidos y Emp\_Baja, los cuáles cumplen la misma función que para los Clientes. Estos tres últimos atributos no fueron incluidos en la tabla Usuario ya que consideramos que el usuario Administrador no puede ser inhabilitado, ni dado de baja, ni puede tener intentos fallidos.

Tanto la razón social como el cuit son datos únicos en la tabla.

## Publicación

### Formato

create table PLEASE\_HELP.Publicacion

(

Pub\_Codigo numeric(18,0) identity(1,1),

Pub\_Fecha\_Inicio datetime,

Pub\_Fecha\_Evento datetime,

Pub\_Descripcion nvarchar(255),

Pub\_Direccion nvarchar(255),

Pub\_Rubro int,

Pub\_Grado int,

Pub\_Empresa int,

Pub\_Estado int,

CONSTRAINT PK\_PUBLICACION\_CODIGO PRIMARY KEY (Pub\_Codigo)

)

ADD CONSTRAINT FK\_PUBLICACION\_GRADO FOREIGN KEY (Pub\_Grado) REFERENCES PLEASE\_HELP.Grado(Grado\_Id),

CONSTRAINT FK\_PUBLICACION\_EMPRESA FOREIGN KEY (Pub\_Empresa) REFERENCES PLEASE\_HELP.Empresa(Emp\_Usuario),

CONSTRAINT FK\_PUBLICACION\_RUBRO FOREIGN KEY (Pub\_Rubro) REFERENCES PLEASE\_HELP.Rubro(Rubro\_Id),

CONSTRAINT FK\_PUBLICACION\_ESTADO FOREIGN KEY (Pub\_Estado) REFERENCES PLEASE\_HELP.EStado(Estado\_Id)

### Descripción

La tabla publicaciones almacena datos acerca de los eventos que compraran los clientes. En la migración ya existían códigos de publicación asi que lo que se hizo es generar un índice a partir del código más grande encontrado.

Ésta entidad se relaciona con el grado o peso de la misma, con la empresa, un rubro y el estado de la misma (borrador, publicada o finalizada).

## Rubro

### Formato

create table PLEASE\_HELP.Rubro

(

Rubro\_Id int identity(1,1),

Rubro\_Descripcion nvarchar(255),

CONSTRAINT PK\_RUBRO\_ID PRIMARY KEY (Rubro\_Id),

CONSTRAINT UQ\_RUBRO\_DESCRIPCION UNIQUE (Rubro\_Descripcion)

)

### Descripción

No existían en las publicaciones de la tabla maestra, rubros asociados a ellas. Se decidió además no agregar rubros, para cumplir con el hecho de que “No se debe crear o inventar información en las tablas migradas”.

## Estado Publicación

### Formato

create table PLEASE\_HELP.Estado

(

Estado\_Id int identity(1,1),

Estado\_Descripcion nvarchar (255),

CONSTRAINT PK\_ESTADO\_ID PRIMARY KEY (Estado\_Id),

CONSTRAINT UQ\_ESTADO\_DESCRIPCION UNIQUE (Estado\_Descripcion)

)

### Descripción

Representa el estado que puede tener una publicación (borrador, publicada, finalizada). La descripción debe ser única.

## Grado Publicación

### Formato

create table PLEASE\_HELP.Grado

(

Grado\_Id int identity(1,1),

Grado\_Comision int NOT NULL,

Grado\_Descripcion nvarchar(255),

CONSTRAINT PK\_GRADO\_ID PRIMARY KEY (Grado\_Id),

CONSTRAINT UQ\_GRADO\_DESRCIPCION UNIQUE (Grado\_Descripcion)

)

### Descripción

Representa grado o peso que puede tener una publicación. Su descripción debe ser única en la tabla.

## Ubicación

### Formato

create table PLEASE\_HELP.Ubicacion

(

Ubicacion\_Publicacion numeric(18,0),

Ubicacion\_Fila varchar(3),

Ubicacion\_Asiento numeric(18,0),

Ubicacion\_Precio numeric(18,0) NOT NULL,

Ubicacion\_Tipo\_Codigo numeric(18,0),

Ubicacion\_Descripcion nvarchar(255),

Ubicacion\_Sin\_Numerar bit,

CONSTRAINT PK\_UBICACION\_FILAXASIENTOXPUBLICACION

PRIMARY KEY (Ubicacion\_Fila, Ubicacion\_Asiento, Ubicacion\_Publicacion)

)

ALTER TABLE PLEASE\_HELP.Ubicacion

ADD CONSTRAINT FK\_UBICACION\_PUBLICACION FOREIGN KEY (Ubicacion\_Publicacion) REFERENCES PLEASE\_HELP.Publicacion(Pub\_Codigo)

### Descripción

Contiene las ubicaciones disponibles para cada publicación de la aplicación. Cuenta con el número o código de publicación con la que está asociada, fila y asiento; donde estos 3 campos en conjunto conforman una clave primaria, identificando unívocamente a cada registro de la tabla. Además cuenta con información adicional como el precio, descripción de la ubicación, tipo de código y si está sin numerar o no.

## Compra

### Formato

create table PLEASE\_HELP.Compra

(

Compra\_Id int identity(1,1),

Compra\_Cliente int,

Compra\_Cantidad numeric(18,0) NOT NULL,

Compra\_Fecha datetime NOT NULL,

Compra\_Metodo\_Pago nvarchar(255) NOT NULL,

Compra\_Email nvarchar(255),

Compra\_Fecha\_Rendida datetime,

Compra\_Fila varchar(3) NOT NULL,

Compra\_Asiento numeric(18,0) NOT NULL,

Compra\_Publicacion numeric(18,0),

CONSTRAINT PK\_COMPRA\_ID PRIMARY KEY (Compra\_Id)

)

ALTER TABLE PLEASE\_HELP.Compra

ADD CONSTRAINT FK\_COMPRA\_UBICACION FOREIGN KEY (Compra\_Fila, Compra\_Asiento, Compra\_Publicacion) REFERENCES PLEASE\_HELP.Ubicacion(Ubicacion\_Fila, Ubicacion\_Asiento, Ubicacion\_Publicacion),

CONSTRAINT FK\_COMPRA\_CLIENTE FOREIGN KEY (Compra\_Cliente) REFERENCES PLEASE\_HELP.Cliente(Cli\_Usuario)

### Descripción

Contiene los datos de compra de una publicación. Dentro de sus atributos, nos encontramos con el identificador, el cliente que realizó la compra, luego información referida al cliente (método de pago, email) y e información referida a la compra (fecha de compra, fecha de rendición: nula si aún no se ha rendida, cantidad comprada: siempre va a ser 1 en nuestro modelo para no tener que hacer una relación muchos a muchos con ítems, haciendo más compleja la consulta entre tablas y más lentas por utilizar joins, una operación costosa en base de datos).

En esta tabla hay una clave foránea compuesta para relacionar ésta tabla con la ubicación perteneciente a la compra, y otra referencia para saber que cliente hizo la compra.

## Factura

### Formato

create table PLEASE\_HELP.Factura

(

Factura\_Nro numeric(18,0) identity(1,1) NOT NULL,

Factura\_Fecha datetime NOT NULL,

Factura\_Total numeric(18,2) NOT NULL,

Factura\_Empresa int,

CONSTRAINT PK\_FACTURA\_NRO PRIMARY KEY CLUSTERED (Factura\_Nro)

)

ALTER TABLE PLEASE\_HELP.Factura

ADD CONSTRAINT FK\_FACTURA\_EMPRESA FOREIGN KEY (Factura\_Empresa) REFERENCES PLEASE\_HELP.Empresa(Emp\_Usuario)

### Descripción

Al rendir las compras, se generan facturas con un número de factura (en la migración se realizó el mismo proceso que en las compras, se buscó el número de factura más grande y se inicia el contado desde ahí), la fecha de rendición, el monto total rendido y la empresa a la que se factura.

## Ítem

### Formato

create table PLEASE\_HELP.Item

(

Item\_Id int identity(1,1),

Item\_Monto numeric(18,2) NOT NULL,

Item\_Cantidad numeric(18,0) NOT NULL,

Item\_Descripcion nvarchar(60) NOT NULL,

Item\_Factura numeric(18,0),

Item\_Compra int,

CONSTRAINT PK\_ITEM\_ID PRIMARY KEY (Item\_Id)

)

ADD CONSTRAINT FK\_ITEM\_FACTURA FOREIGN KEY (Item\_Factura) REFERENCES PLEASE\_HELP.Factura(Factura\_Nro),

CONSTRAINT FK\_ITEM\_COMPRA FOREIGN KEY (Item\_Compra) REFERENCES PLEASE\_HELP.Compra(Compra\_Id)

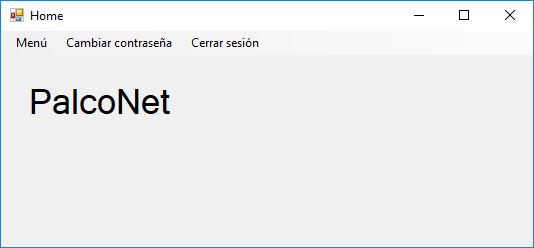
### Descripción

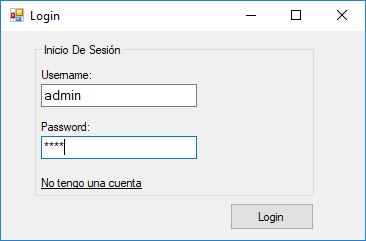
Representa los ítems de la factura. Cuenta con un identificador, un monto parcial, cantidad (siempre 1), descripción del ítem, factura a la que pertenece y número de compra.

# Datos Generales de la aplicación

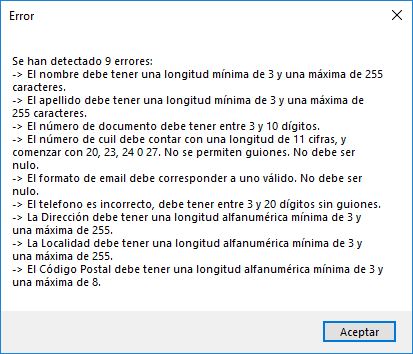
Hay ciertas cuestiones generales a saber antes de empezar a usar la aplicación, que será de utilidad para tener una mejor experiencia de usuario.

1. **La aplicación finaliza cuando se cierra la ventana Home o el inicio de sesión** desde la cruz en el extremo superior derecho.

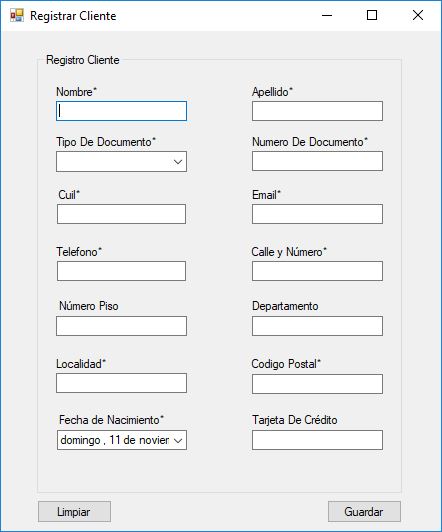




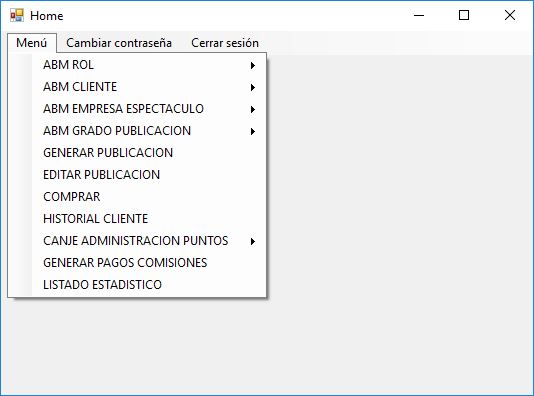
1. Cada formulario de la aplicación cuenta con ciertas restricciones con respecto al formato de los datos que ingresan al sistema. En caso de completar un formulario, confirmar la operación, y no cumplir con el formato de los datos, se mostrara un mensaje de error como la siguiente (el mensaje de error depende de cada formulario):



1. Los campos en los formularios que cuentan con un asterisco (\*) son estrictamente obligatorios, en caso de no completarlos, se arrojará un error. En cambio, los campos sin asteriscos, pueden no completarse. Por ejemplo:



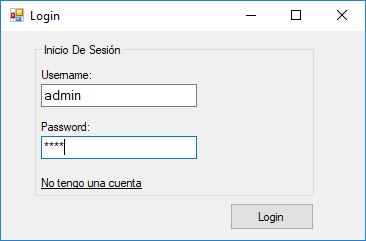
1. Todas las funcionalidades a las que el usuario tiene permitido acceder, se encuentran en las pestañas dentro de la ventana Home:



1. En los **filtros de búsqueda** de los formularios, al dejar **todos** los campos vacíos, la aplicación procederá a buscar absolutamente todos los valores posibles sin restricciones de filtrado.
2. Los listados que se utilicen para gestionar entidades, no están habilitados para crear, modificar o borrar datos de la lista. Para ello, se muestran iconos en representación a la operación que se desea realizar.
3. En la tabla maestra proporcionada por la catedra, las publicaciones no tenían un rubro asignado. Para respetar el hecho de que “no se deben crear datos en el proceso de migración”, se decidió no crear rubros.

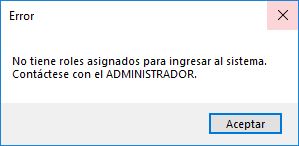
# Login y Seguridad

La plataforma, al iniciarse despliega una ventana de acceso a la misma (login), donde el usuario deberá completar obligatoriamente los campos usuario y contraseña.

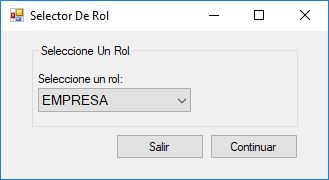


En esta ocasión pueden ocurrir 3 casos:

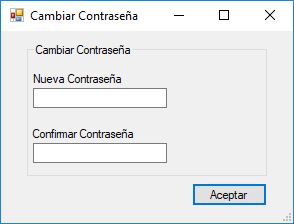
1. Que el usuario no cuente con roles en la plataforma, causado por la baja de los mismos. En esta ocasión, se le mostrará un mensaje de error diciendo que no es posible acceder al sistema si no se cuentan con roles.



1. Que el usuario tenga un solo rol. En este caso, podrá ingresar normalmente a la aplicación.
2. Que el usuario tenga más de un rol. De ser así, procederá a elegir uno de los roles asignados:



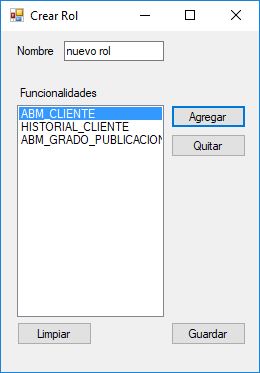
En definitiva, si cuenta con los requisitos para ingresar al sistema, y es la primera vez que accede al mismo (primer login) se le pedirá un cambio de contraseña.



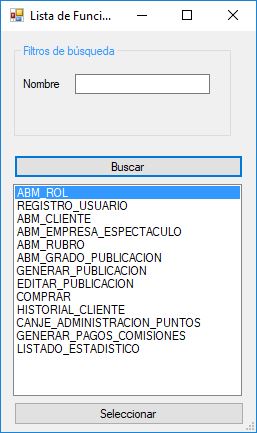
# ABM de Rol (ADMIN)

## Crear Rol

Cuando se deseen crear roles, el administrador (una vez ingresado al sistema) deberá acceder a la solapa de crear rol. Se le desplegará una ventana para completar los datos pertinentes.



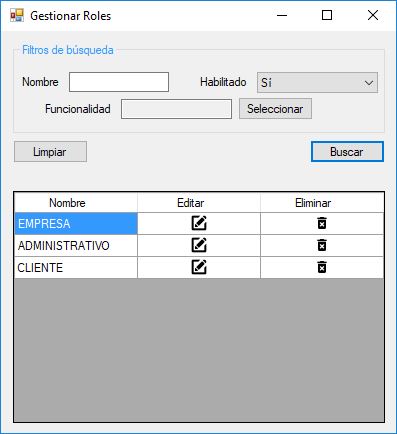
Dentro de ésta funcionalidad, deberá ingresar el nombre del rol a crear, y asignarle una o más funcionalidades. Esto último se realiza con el botón “Agregar”. En caso de equivocación al asignar funcionalidades, podrá deshacer la acción con el botón “Quitar”.



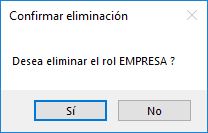
El administrador podrá escoger los roles uno por uno, para asignárselos al rol por crear.

## Editar o Borrar un rol

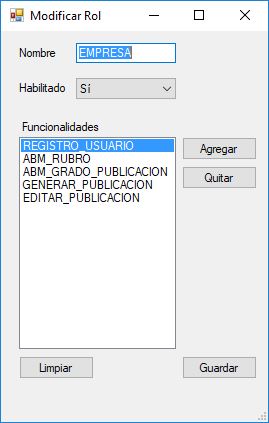
Está la posibilidad de querer editar o borrar roles ya creados, por lo que estás operaciones se pueden realizar dentro de la funcionalidad de gestionar roles.



En caso de querer borrar lógicamente un rol, se deberá presionar en el botón del tacho de basura. Confirmando la operación, se habrá dado de baja el rol.

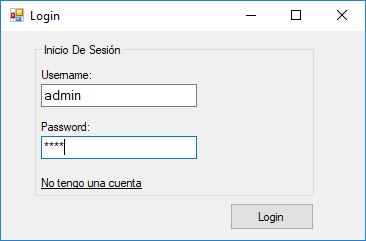


En caso de querer modificar un rol, se deberá presionar en el botón del lápiz. De esta manera, se abrirá una ventana similar a la de crear rol, explicada anteriormente, con los datos completos del rol a modificar, y procederá a editar los campos necesarios.

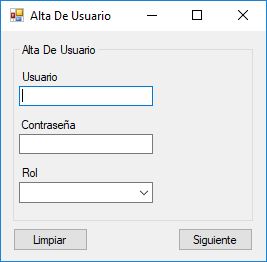


# Registro de Usuario

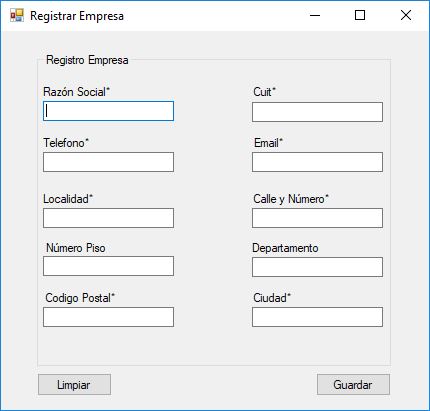
Si el usuario no cuenta con una cuenta dentro de Palconet, puede proceder a crear la misma. Dentro del login, deberá acceder al link con la descripción “No tengo una cuenta”.

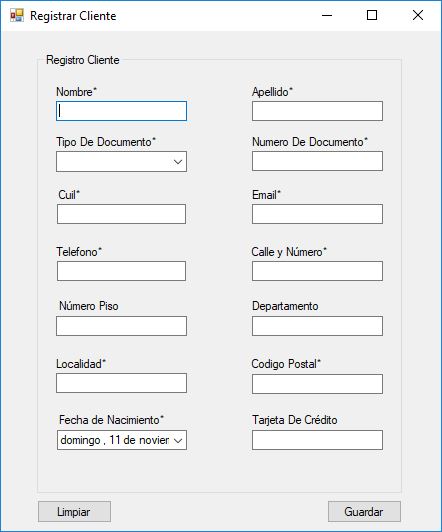


Procediendo, se tendrán que completar los datos de usuario, contraseña y el rol que desee.



Dependiendo del rol ingresado, se desplegará una ventana donde se deberá completar la información correspondiente al rol.

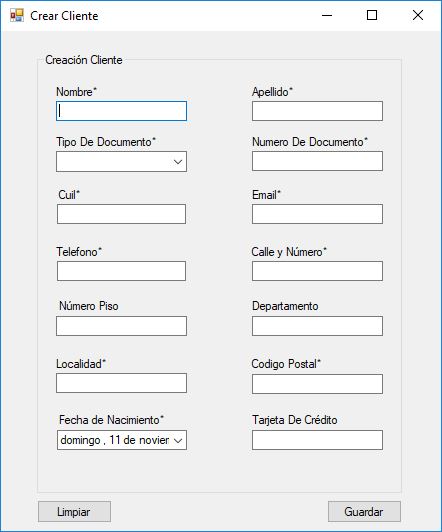




# ABM Clientes (ADMIN)

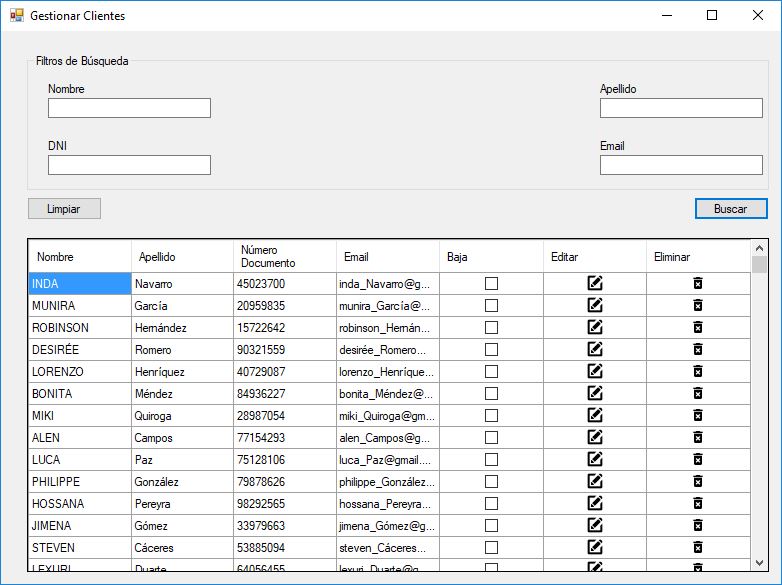
## Crear Clientes

Para crear clientes se procederá de la misma forma que en el registro de clientes.

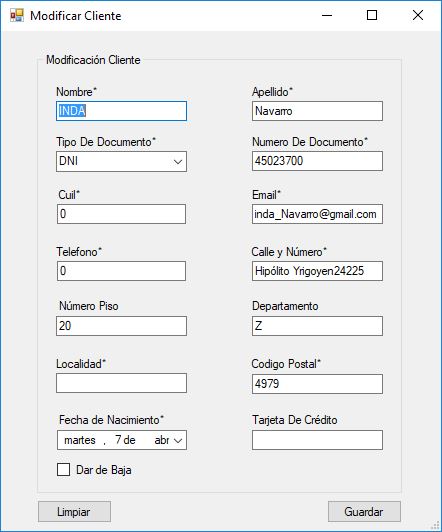


## Editar o Borrar Clientes

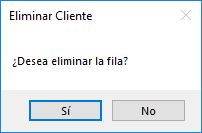
Dentro de la funcionalidad gestionar clientes, se podrán realizar operaciones de modificado y borrado de datos con respecto a clientes ya registrados en el sistema.



En caso de querer modificar, al presionar el botón de editar, se abrirá una ventana similar a la de registro de clientes, donde se procederá a completar la información. Esta ventana, a diferencia de la de registrar cliente, contiene un campo checkbox por si se quiere deshabilitar al usuario.



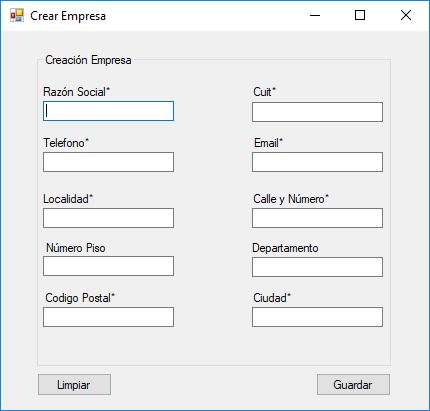
En caso de querer eliminar un usuario, se deberá aceptar el mensaje de confirmación.



# ABM de Empresas (ADMIN)

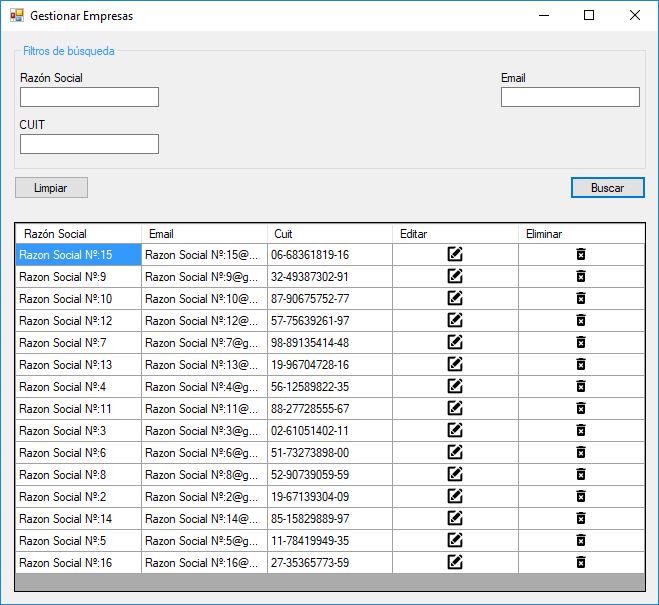
## Crear Empresa

Para crear una empresa, se procede de la misma forma que cuando se registra una empresa en la aplicación.

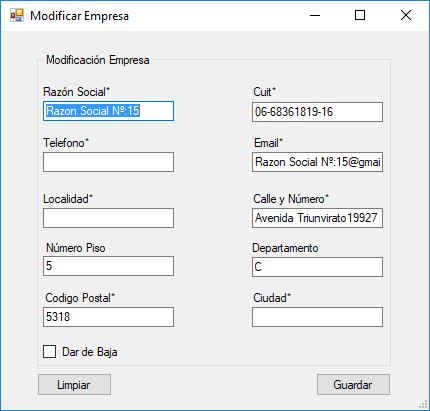


## Editar o Borrar Empresas

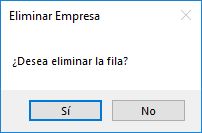
Dentro de la funcionalidad gestionar empresas, se podrán realizar operaciones de modificado y borrado de datos con respecto a empresas ya registrados en el sistema.



En caso de querer modificar, al presionar el botón de editar, se abrirá una ventana similar a la de registro de empresas, donde se procederá a completar la información. Esta ventana, a diferencia de la de registrar empresa, contiene un campo checkbox por si se quiere deshabilitar al usuario.



En caso de querer eliminar un usuario, se deberá aceptar el mensaje de confirmación.



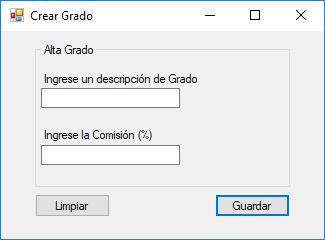
# ABM de Rubro (ADMIN)

Como expresa claramente el enunciado, no era una funcionalidad necesaria del sistema.

# ABM Grado Publicación

## Crear Grado

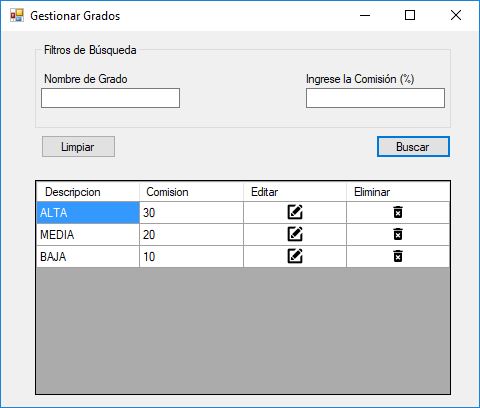
Si se quiere crear un grado de Publicación, se debe abrir la ventana de creación en la solapa del menú, y completar la información pertinente.



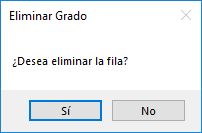
La comisión perteneciente al grado creado, representa un valor porcentual y tiene que estar entre 0 y 100, siendo su valor entero.

## Editar o Eliminar Grado

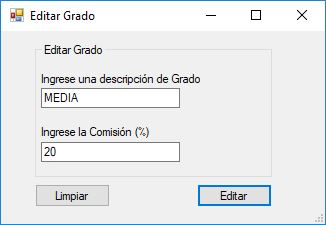
Análogamente a los listados vistos anteriormente, se presenta una ventana donde se pueden gestionar los grados de publicación.



Al eliminar un grado, se desplegará un mensaje de confirmación que en caso de aceptar, se completará la operación.

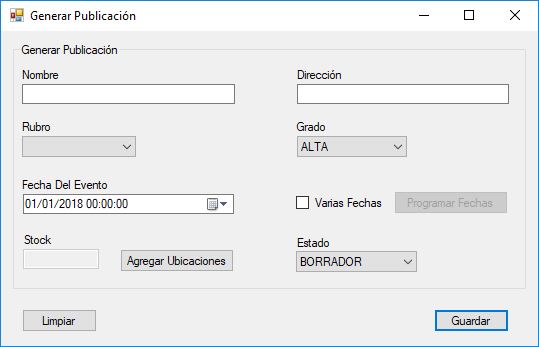


A su vez, el proceso de edición, es similar al de alta.

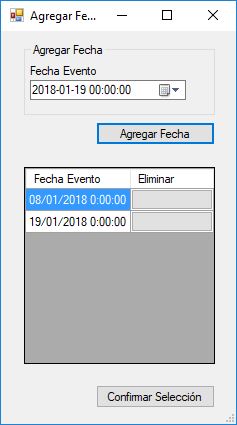


# Generar Publicación

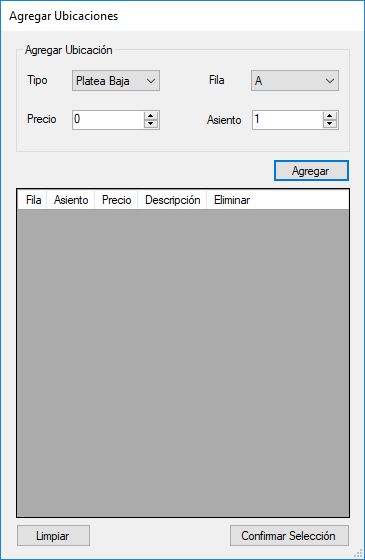
Ésta funcionalidad permite crear publicaciones en estado BORRADOR o PUBLICADA. Se deberán completar todos los datos para obtener una operación satisfactoria.



Para cargar varias fechas de evento, ya que puede repetirse el mismo evento en diferentes días a lo largo del tiempo, se agrega un checkbox que en caso de activarlo, permite esta funcionalidad. La selección de esta opción, y luego presionando el botón “Programar Fechas” abrirá una ventana donde es posible configurar las mismas.



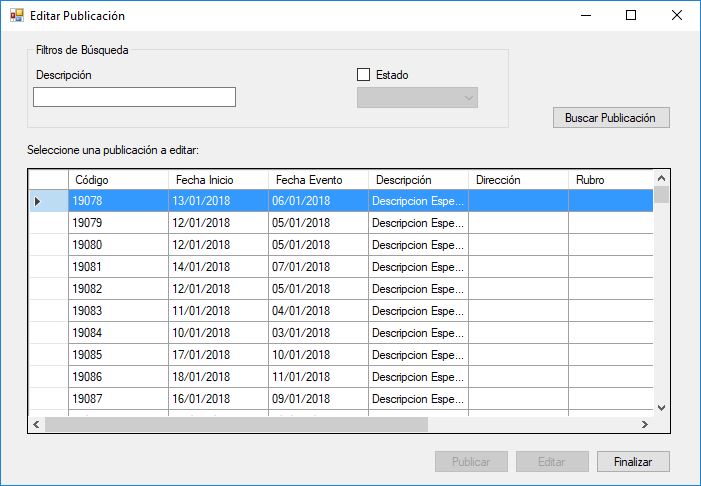
A su vez, se deberán cargar ubicaciones disponibles para el evento, presionando el botón Agregar Ubicaciones. Esto desplegará una ventana para cumplir con este requerimiento.



Por último, como bien se dijo, se podrá configurar el estado de la publicación a BORRADOR o PUBLICADA dependiendo del deseo del usuario.

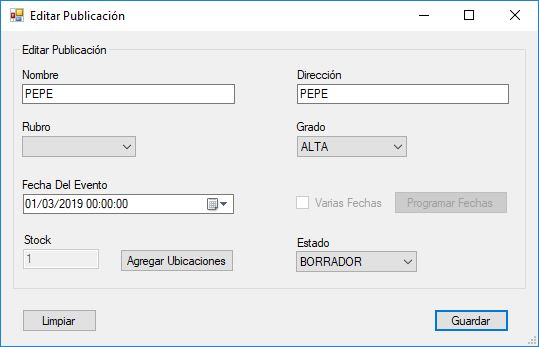
## Editar o Publicar una Publicación

Similar a los casos anteriores de gestión de entidades, se lista las publicaciones pertenecientes a la empresa en sesión.



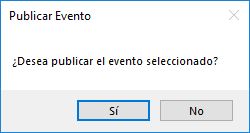
En ésta nueva metodología, se procede a elegir una publicación desde la primera columna de la izquierda, habilitándose así los botones Publicar y Editar.

En caso de Editar se abre la siguiente ventana:



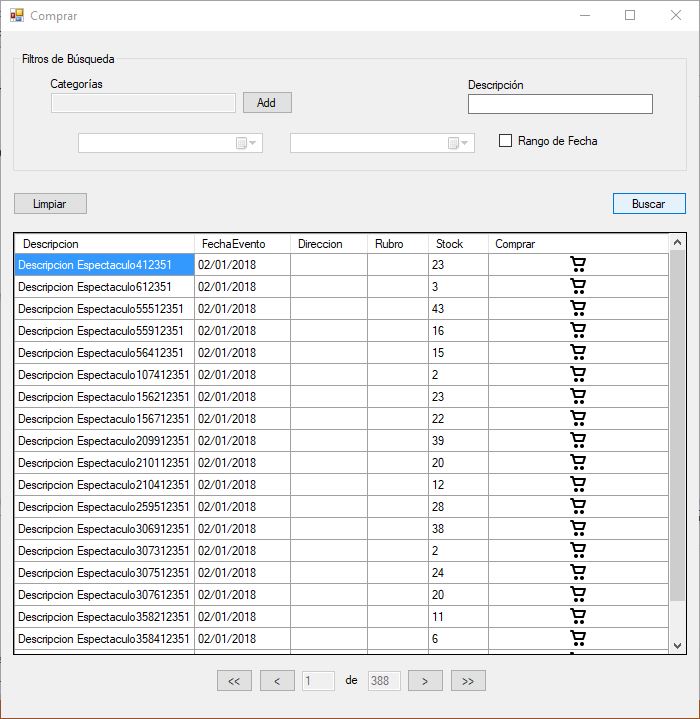
De la misma forma que el alta de publicaciones, se edita el contenido de la publicación existente.

En caso de presionar el botón Publicar, y confirmando el mensaje, automáticamente se publica la publicación en el sitio para que los clientes puedan comprar entradas.

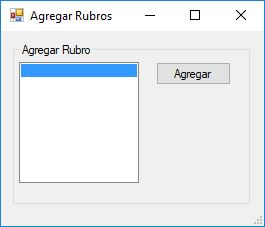


# Comprar

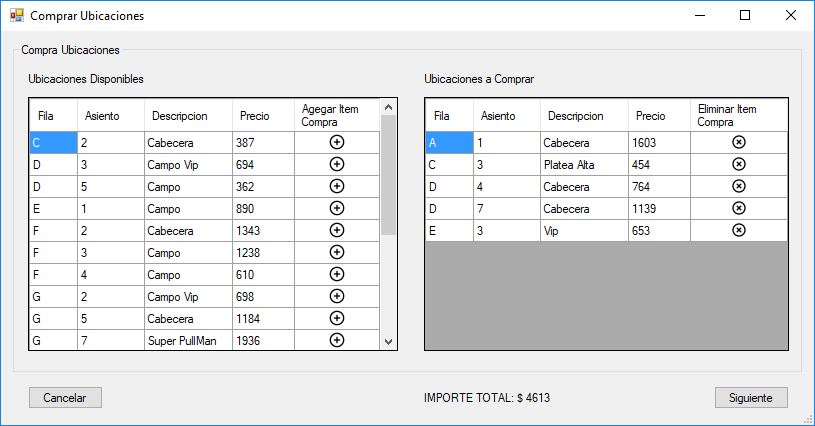
Al querer comprar una entrada, el sistema desplegará las publicaciones en estado PUBLICADA. De esta forma, el cliente junto a los filtros que se le proporciona, elegirá la publicación que desee. Presionando en el carrito, empezará el proceso de compra.



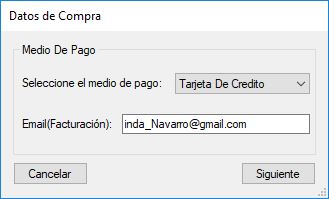
Un detalle importante es que en los filtros, podrá elegir filtrar por rubros, en lo que se le desplegará la siguiente ventana.



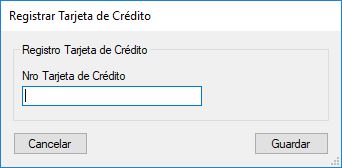
Siguiendo con la compra, en el listado izquierdo se seleccionan las ubicaciones disponibles, haciendo click en el más. En caso de arrepentirse con alguna compra, hay posibilidad de eliminar el ítem del carrito, haciendo click en la cruz, en el listado derecho que representa las ubicaciones a comprar. El monto total de la parte inferior central de la ventana.



Al darle siguiente, es necesario completar detalles adicionales de la compra, como el método de pago y el email de comunicación. Por defecto, el email se autocompleta con el que ya tiene registrado el cliente en su cuenta. En caso de querer modificarlo, podrá hacerlo sin inconvenientes.

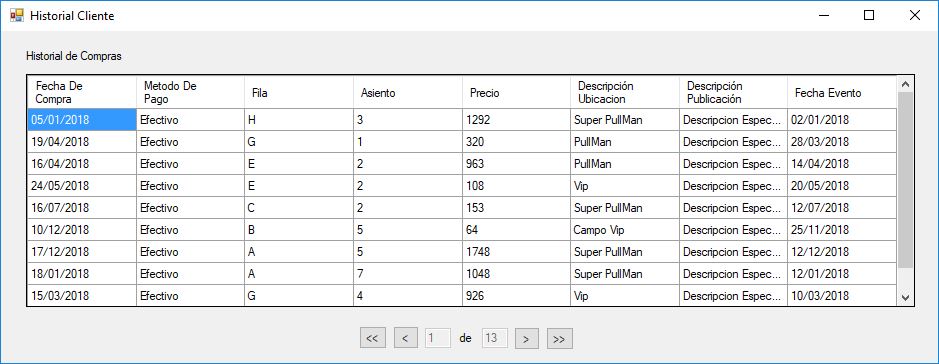


Finalmente se completa el número de tarjeta de crédito, para el pago online, en caso de no haber registrado una en el momento de creación de la cuenta. En este caso, no se admiten valores vacíos.



# Historial Cliente

En esta sección, el cliente podrá visualizar las compras que realizo en la aplicación con completo detalle. Los datos están paginados, y pueden pasarse de una a la vez, e ir de la primera a la última hoja.



# Canje y Administración de puntos

## Criterio de puntos asociados a compras

Con respecto a los puntos por compra, se suma 1 punto cada 10 pesos vendidos por compra. En caso de que el importe final de la compra no sea múltiplo de 10, no se le sumará el proporcional (se realiza la división entera del importe por 10). Ejemplos:

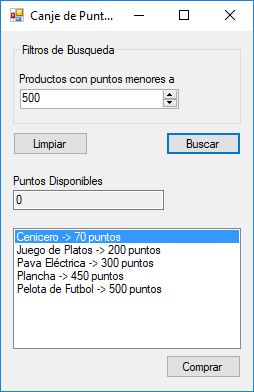
Importe = $100 -> Puntos Ganados = 10

Importe = $ 456 -> Puntos Ganados = 45

Por otro lado, cada punto por compra ganado tiene un vencimiento. El vencimiento es de 12 meses a partir del día que se realiza la compra. Por lo que puedo tener varios conjuntos de puntos con vencimientos distintos según el día que haya comprado.

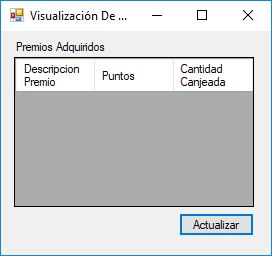
## Canje de puntos

En la ventana de canje de puntos, puedo buscar premios según un puntaje e intercambiarlos por los puntos adquiridos. Ésta sección muestra el filtro de búsqueda correspondiente a la cantidad de puntos del premio a canjear; los puntos disponibles (según todos los puntos NO vencidos que posea el usuario) y una lista con los resultados de la búsqueda.



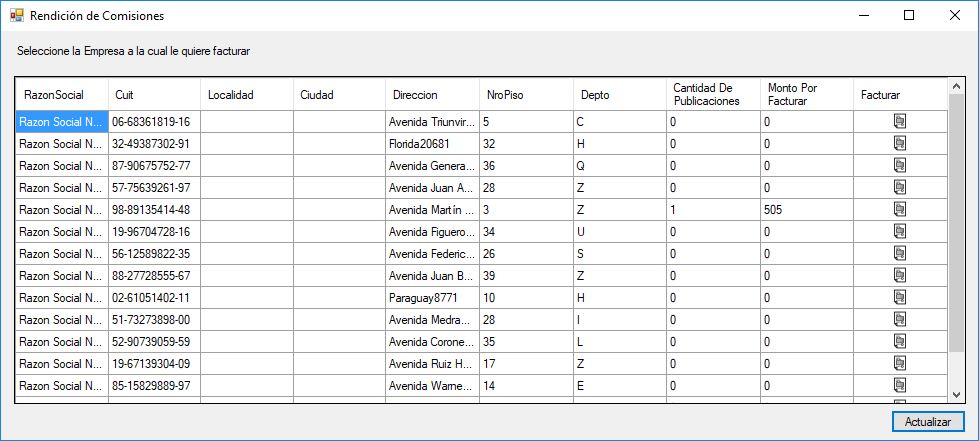
## Visualización de Premios

Por otro lado, en la sección de Visualización de premios, se nos presenta una lista con los premios adquiridos por la plataforma, indicando descripción del premio, puntos que se requiere para el canje de ese premio, y la cantidad de premios de ese tipo adquiridos hasta el momento.

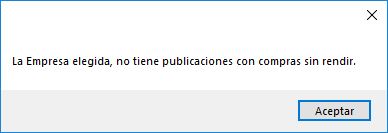


# Generar Rendiciones

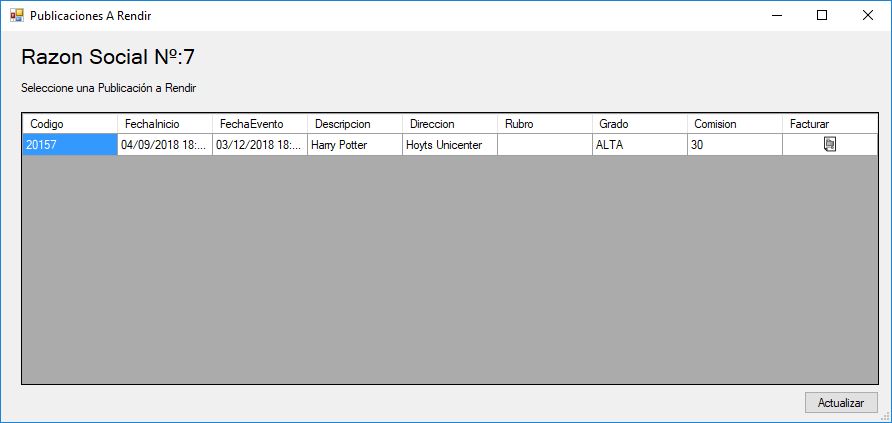
El listado presentado muestra **TODAS**las empresas de la plataforma, con parte de su información y la cantidad de publicaciones **por rendir,** es decir que tienen compras no rendidas; el monto total por facturar (la sumatoria de todos los importes de todas las compras de todas las publicaciones sin todavía haberse rendido) y la opción de facturar.



En caso de seleccionar una empresa sin compras por rendir, aparecerá un mensaje de error, informando que no es posible realizar la operación.



Al continuar con el proceso de rendición de comisiones, después de haber elegido una empresa con compras por rendir, el sistema generará una lista con las publicaciones de esa empresa que cuenten con compras sin rendir.



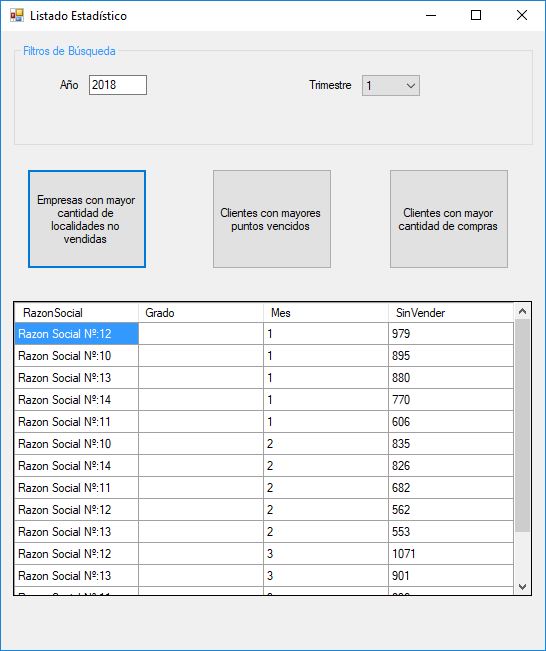
Dentro de la lista, se mostrará información correspondiente a la publicación, incluyendo el número de identificación de la publicación. Al presionar Facturar, finalmente llegaremos a la última instancia donde veremos las compras de dicha publicación, de la empresa elegida previamente, sin rendir.



Aquí se seleccionara la cantidad de compras a rendir, no pudiéndose ingresar un valor mayor a la cantidad de compras mostradas. Vale la pena aclarar que la rendición se hará desde las fecha más antiguas hasta las más actuales en el tiempo.

# Listado Estadístico

Finalmente se presenta la última sección de la aplicación que representa una estadística de ciertos aspectos del negocio.



Se selecciona un año y trimestre, y se presiona una de las tres opciones para mostrar las empresas con mayor cantidad de localidades no vendidas, los clientes con mayores puntos vencidos y/o los clientes con mayor cantidad de compras.

# DER

# 

