



# TP2C2017 K3571 MIRRORING\_GUYS

## Gestión de Datos

Preparado para:

Gestión de Datos

Preparado por:

Alumno	Legajo
Kahrs, Luis	127302-4
Valentin, Victor	147079-6
Lavagnino, Facundo	144532-7



## Contenido

Estrategía.....	3
DER .....	4
STORED PROCEDURES .....	5



## Estrategía

Para poder realizar el TP, la primera decisión que tomamos es la de separar la estructura del código en dos proyectos:

- **PagoAgilFrba.Core**

Consiste en el core de la aplicación. Aquí vamos a poder encontrar algunas sub carpetas que contienen:

- Database
  - La conexión a la base de datos,
  - El manejo de transacciones,
  - El manejo de sentencias.
- Entidades
  - Las clases correspondientes a los objetos que vamos a manejar:
    - Cliente,
    - Direccion,
    - Empresa,
    - EntidadBase (de la que heredan todas, contiene los métodos base como guardar, borrar, etc)
    - Factura,
    - ItemFactura,
    - Listado,
    - Rendicion,
    - Rol,
    - Rubro,
    - Sucursal,
    - Usuario.
- Errores
  - Contiene la clase que maneja los errores en la aplicación
- Helper
  - Contiene el helper para poder encriptar / desencriptar (Encriptacion.cs)

- **PagoAgilFrba.WinForms**

- Formularios
  - Vamos a poder encontrar una sub carpeta para cada formulario de la aplicación.
  - Los formularios tienen la capa de presentación y se conectan al proyecto .Core para realizar las transacciones
- Helper
  - Contiene los métodos comunes para poder reutilizar fácilmente.
- Iconos
  - Contiene las imágenes de los iconos de cada uno de los formularios.

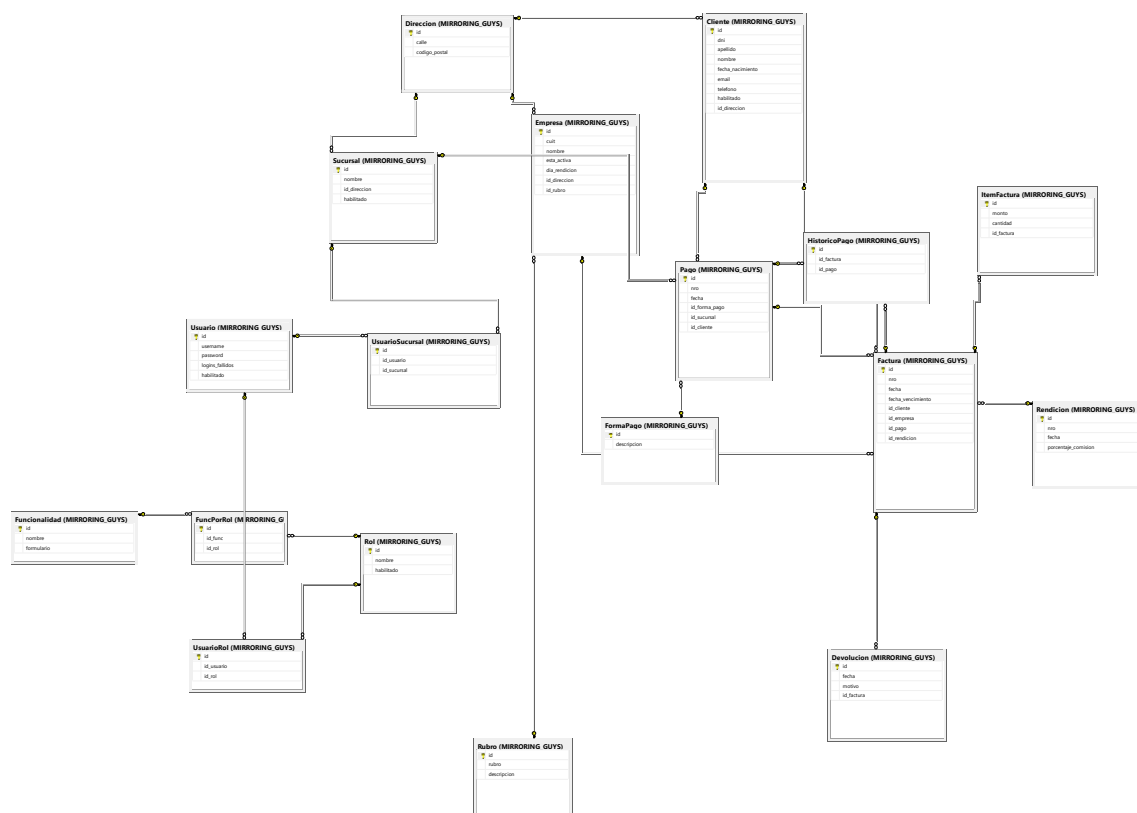


Para mantener la sencillez de la aplicación decidimos utilizar la menor cantidad de objetos de DB posibles, utilizando directamente consultas en el código fuente y algunos stored procedures.

Para mantener la integridad de datos implementamos primary keys y foreign keys que vamos a describir más adelante.

Con respecto a las validaciones de unicidad, las terminamos haciendo por código, ya que encontramos valores en la tabla maestra que no cumplían con la condición, imposibilitando el uso de constraints. Por ejemplo, hay varios clientes con el mismo mail.

## DER





## STORED PROCEDURES

### [Usuario\_Login]

Se encarga de manejar el login del usuario, controlar que exista y que no este bloqueado. También es el encargado de manejar los reintentos.

Retorna un error de DB por cada situación no valida y si no un result set con toda la información del usuario.