

# PROYECTO PL/SQL

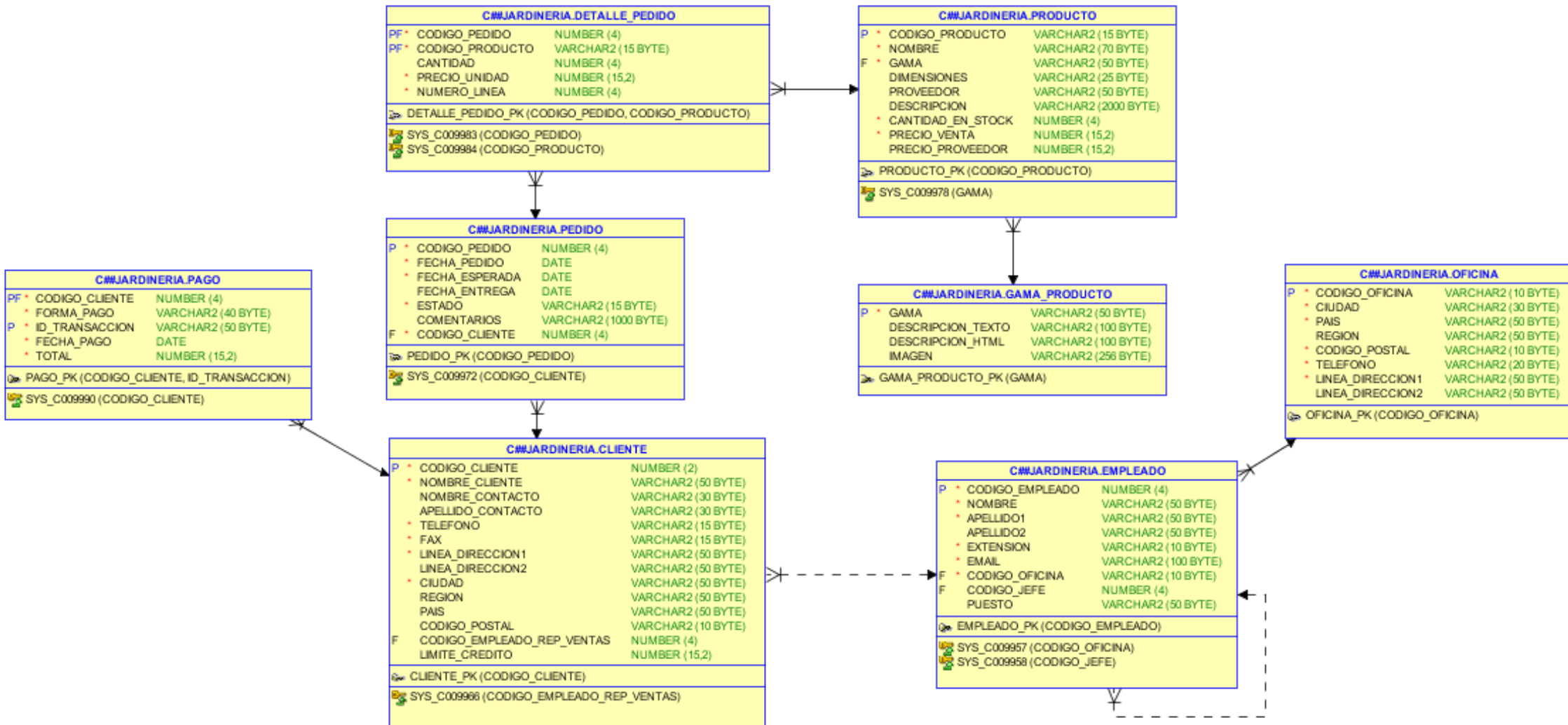


Rafael Martín Berenguer  
1ºDAM

# Índice

1. Modelo Relacional
2. Base de datos
3. Requisitos
  - 3.1. Bloque anónimo
  - 3.2. Función
  - 3.3. Procedimientos
  - 3.4. Disparadores
  - 3.5. Código de GitHub

# 1. Modelo Entidad Relación:



## 2. Base de datos

En el proyecto de PL/SQL he decidido utilizar la base de datos de jardinería.

He utilizado dicha base de datos, ya que, esta es muy completa en cuanto a las tablas que contiene y sus atributos. Es decir, la base de datos contiene una serie de tablas y variedad de atributos de todos los tipos. Esto hace más fácil la elaboración de procedimientos, por la gran cantidad de información que podemos sacar; funciones y disparadores.

### 3. Requisitos:

#### Bloque anónimo:

- El bloque anónimo que incluyo en el proyecto es como el programa principal desde el que llamamos a los procedimientos y funciones. He creado un menú al cual según la elección del menú, te dará información sobre la opción elegida. Es decir, si la variable menú la ponemos a 1, se ejecuta la primera función, si le damos el valor 2, se ejecuta la segunda función y si damos el valor 3 o 4 se ejecuta uno de los dos procedimientos.

#### Funciones:

- La primera función del proyecto, va a ser la encargada de indicarnos la promoción de un cliente según la cantidad total que haya gastado con los pagos realizados. Es decir, si nuestro cliente indicado ha realizado un total de pagos de más de 80000 € el cliente tendrá una promoción de un 40% en la compra de una televisión Samsung de 55 pulgadas y dependiendo del gasto del cliente tendrá una serie de promociones. La función devolverá una cadena de texto con la promoción que le corresponde al cliente pasado por parámetro.
- La segunda función del proyecto, es la encargada de devolver si el producto que le pasamos como parámetro es necesario reponerlo según el número de stock que queden. Es decir, si un producto tiene un número de stock menor o igual a 50 debe devolver una cadena indicando que hay que reponer más productos, si es al contrario devolverá una cadena como que no es necesario reponer el producto porque hay todavía suficientes.

### Procedimientos:

- El primer procedimiento se va a encargar de mostrarnos información de todos los clientes que hayan realizado algún pedido. Además, nos mostrará el número de pedidos que ha hecho dicho cliente e irá mostrando información de cada uno de los pedidos de este cliente; en el caso de que ese pedido no tenga fecha de entrega se nos mostrará un mensaje como que ese pedido no se ha entregado todavía. En cada cliente mostrará, además de todo lo dicho anteriormente la cantidad total que ha gastado dicho cliente en todos sus pedidos y por último como información general se muestra la cantidad total de clientes que han realizado pedidos, la cantidad total de pedidos entre todos los clientes y la cantidad total de dinero registrado entre todos los pedidos.
- Al segundo procedimiento le pasaremos por parámetros dos fechas, una fecha inicial y otra final. El procedimiento debe mostrar el dinero que se ha obtenido entre todos los pedidos realizados en el periodo de la dos fechas pasadas por parámetros. Además de información de los clientes que han realizado pedidos, de los empleados de dicho cliente y de la oficina en la que trabaja dicho empleado. Al final, se mostrará el total de dinero gastado en el periodo de tiempo indicado.

## Disparadores:

- Crearemos un disparador que inserte en la tabla auditoria cliente la información de cada uno de los clientes que se vayan insertando, actualizando o eliminando de nuestra base de datos, en la tabla auditoria se introducirá el usuario que ha aplicado la modificación, la fecha y hora en la que se ha producido, información de la operación que ha realizado (en formato de número en la que 1 es insertar, 2 es actualizar y 3 borrar), los registros antiguos del cliente y los registros nuevos del cliente.
- Crearemos un disparador que inserte en la tabla auditoria pedido la información de cada uno de los pedidos que se vayan insertando, actualizando o eliminando de nuestra base de datos, en la tabla auditoria se introducirá el usuario que ha aplicado la modificación, la fecha y hora en la que se ha producido, información de la operación que ha realizado(en formato de número en la que 1 es insertar, 2 es actualizar y 3 borrar), los registros antiguos del pedido y los registros nuevos del pedido..
- Crearemos un disparador en el que si se actualiza un detalle pedido, se actualice la cantidad en stock del producto comprado. Hay que tener en cuenta que si en la actualización del detalle pedido la cantidad del producto aumenta se le debe restar a la cantidad en stock del detalle pedido, pero, si disminuye la cantidad en detalle pedidos, la cantidad en stock aumenta. Además, en el caso de que la cantidad del detalle pedido sea mayor que la cantidad en stock del producto nos saltará un error indicando que dicha actualización no se puede realizar.

NOTA: Las tablas auditoria cliente y auditoria pedido, son creadas en el proyecto. La tabla consta de 5 atributos: una cadena en la guardamos el usuario, una cadena en la que guardamos la fecha y hora, un atributo de tipo número en el que guardamos la operación, una cadena con el registro antiguo y una cadena con el nuevo registro. En los registros antiguos y nuevos cada atributo del cliente y de los pedidos en la cadena queda separados por “#”.

Al proyecto se le ha creado a cada apartado su gestión de excepciones correspondiente.

## 4. Código de GitHub

Código: <https://github.com/rafita12/Proyecto-PL-SQL.git>