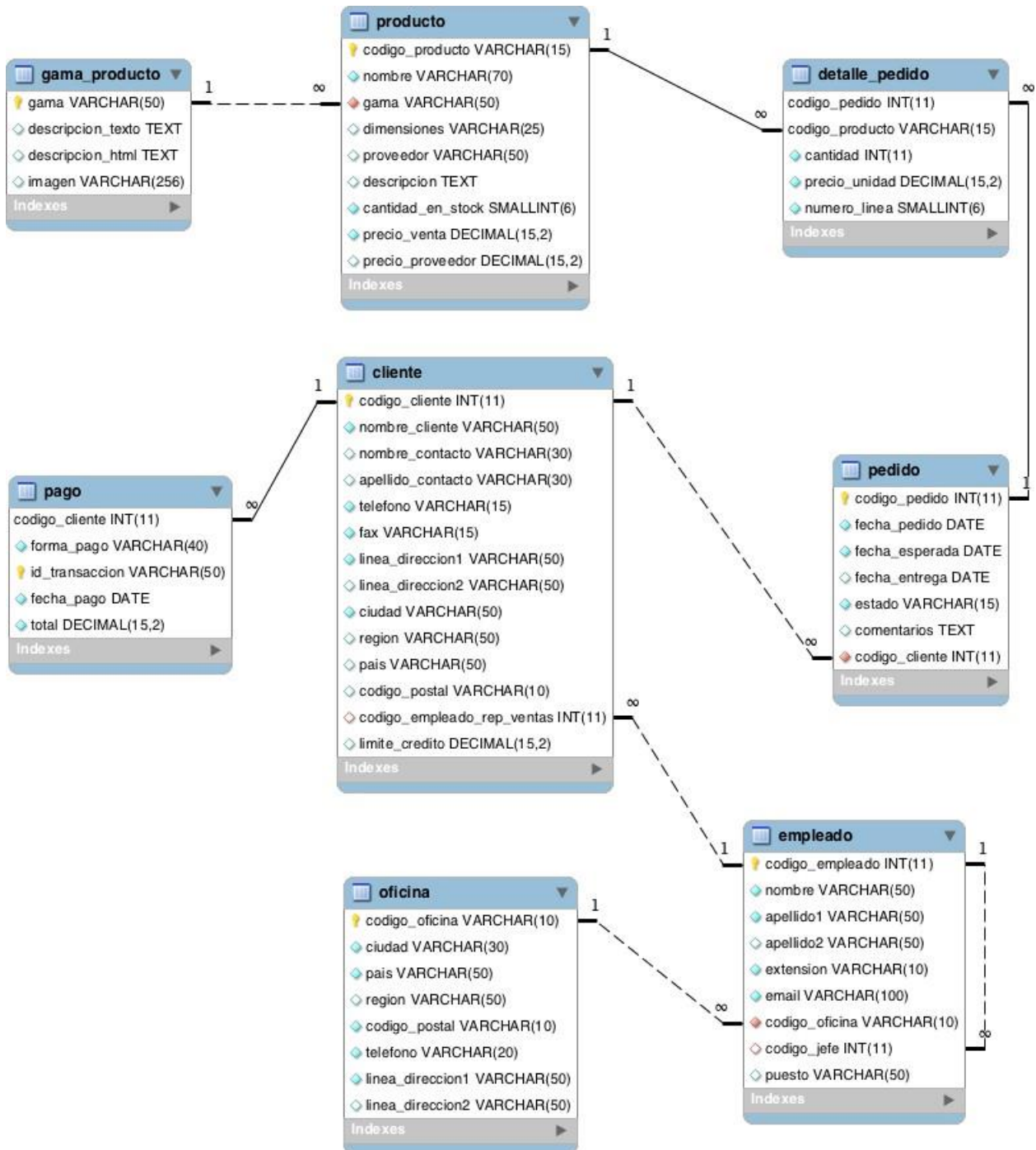


## EJERCICIO BD-MYSQL: JARDINERÍA

Dado el siguiente diagrama E-R:



Obtenemos las siguientes tablas:

```

DROP DATABASE IF EXISTS jardineria;
CREATE DATABASE jardineria CHARACTER SET utf8mb4;
USE jardineria;

CREATE TABLE oficina (
    codigo_oficina VARCHAR(10) NOT NULL,
    ciudad VARCHAR(30) NOT NULL,

```

## DESARROLLO DE INTERFACES – 2º DAM

```
pais VARCHAR(50) NOT NULL,
region VARCHAR(50) DEFAULT NULL,
codigo_postal VARCHAR(10) NOT NULL,
telefono VARCHAR(20) NOT NULL,
linea_direccion1 VARCHAR(50) NOT NULL,
linea_direccion2 VARCHAR(50) DEFAULT NULL,
PRIMARY KEY (codigo_oficina)
);

CREATE TABLE empleado (
codigo_empleado INTEGER NOT NULL,
nombre VARCHAR(50) NOT NULL,
apellido1 VARCHAR(50) NOT NULL,
apellido2 VARCHAR(50) DEFAULT NULL,
extension VARCHAR(10) NOT NULL,
email VARCHAR(100) NOT NULL,
codigo_oficina VARCHAR(10) NOT NULL,
codigo_jefe INTEGER DEFAULT NULL,
puesto VARCHAR(50) DEFAULT NULL,
PRIMARY KEY (codigo_empleado),
FOREIGN KEY (codigo_oficina) REFERENCES oficina (codigo_oficina),
FOREIGN KEY (codigo_jefe) REFERENCES empleado (codigo_empleado)
);

CREATE TABLE gama_producto (
gama VARCHAR(50) NOT NULL,
descripcion_texto TEXT,
descripcion_html TEXT,
imagen VARCHAR(256),
PRIMARY KEY (gama)
);

CREATE TABLE cliente (
codigo_cliente INTEGER NOT NULL,
nombre_cliente VARCHAR(50) NOT NULL,
nombre_contacto VARCHAR(30) DEFAULT NULL,
apellido_contacto VARCHAR(30) DEFAULT NULL,
telefono VARCHAR(15) NOT NULL,
fax VARCHAR(15) NOT NULL,
linea_direccion1 VARCHAR(50) NOT NULL,
linea_direccion2 VARCHAR(50) DEFAULT NULL,
ciudad VARCHAR(50) NOT NULL,
region VARCHAR(50) DEFAULT NULL,
pais VARCHAR(50) DEFAULT NULL,
codigo_postal VARCHAR(10) DEFAULT NULL,
codigo_empleado_rep_ventas INTEGER DEFAULT NULL,
limite_credito NUMERIC(15,2) DEFAULT NULL,
PRIMARY KEY (codigo_cliente),
```

## DESARROLLO DE INTERFACES – 2º DAM

```
FOREIGN KEY (codigo_empleado_rep_ventas) REFERENCES empleado (codigo_empleado)
);
```

```
CREATE TABLE pedido (
  codigo_pedido INTEGER NOT NULL,
  fecha_pedido date NOT NULL,
  fecha_esperada date NOT NULL,
  fecha_entrega date DEFAULT NULL,
  estado VARCHAR(15) NOT NULL,
  comentarios TEXT,
  codigo_cliente INTEGER NOT NULL,
  PRIMARY KEY (codigo_pedido),
  FOREIGN KEY (codigo_cliente) REFERENCES cliente (codigo_cliente)
);
```

```
CREATE TABLE producto (
  codigo_producto VARCHAR(15) NOT NULL,
  nombre VARCHAR(70) NOT NULL,
  gama VARCHAR(50) NOT NULL,
  dimensiones VARCHAR(25) NULL,
  proveedor VARCHAR(50) DEFAULT NULL,
  descripcion text NULL,
  cantidad_en_stock SMALLINT NOT NULL,
  precio_venta NUMERIC(15,2) NOT NULL,
  precio_proveedor NUMERIC(15,2) DEFAULT NULL,
  PRIMARY KEY (codigo_producto),
  FOREIGN KEY (gama) REFERENCES gama_producto (gama)
);
```

```
CREATE TABLE detalle_pedido (
  codigo_pedido INTEGER NOT NULL,
  codigo_producto VARCHAR(15) NOT NULL,
  cantidad INTEGER NOT NULL,
  precio_unidad NUMERIC(15,2) NOT NULL,
  numero_linea SMALLINT NOT NULL,
  PRIMARY KEY (codigo_pedido, codigo_producto),
  FOREIGN KEY (codigo_pedido) REFERENCES pedido (codigo_pedido),
  FOREIGN KEY (codigo_producto) REFERENCES producto (codigo_producto)
);
```

```
CREATE TABLE pago (
  codigo_cliente INTEGER NOT NULL,
  forma_pago VARCHAR(40) NOT NULL,
  id_transaccion VARCHAR(50) NOT NULL,
  fecha_pago date NOT NULL,
  total NUMERIC(15,2) NOT NULL,
  PRIMARY KEY (codigo_cliente, id_transaccion),
  FOREIGN KEY (codigo_cliente) REFERENCES cliente (codigo_cliente)
```

```
);
```

Usando esta BD, queremos elaborar una aplicación de formularios que me permita realizar las siguientes operaciones:

- **Añadir productos:** realiza un formulario que permita introducir los datos para un nuevo producto. Controla que los campos en los que se han de introducir números, informen de error si se introdujesen letras, y que en los campos que son claves externas sólo se puedan introducir valores válidos, (puedes utilizar un combo). Cuando le demos a añadir, los datos para ese producto deben quedar almacenados en la base de datos.
- **Eliminar gama:** debes realizar un formulario en el que el **nombre** de las posibles gamas se podrá seleccionar mediante un combobox. Una vez seleccionada aparecerán los datos de la misma. Al darle al botón de eliminar, se solicitará al usuario una confirmación de dicha operación y si el usuario confirma, entonces se procederá a su eliminación de la base de datos.
- **Consultar pedidos entregados:** elabora un formulario en el que podamos visualizar el resultado de la siguiente consulta: ***"Devuelve un listado de todos los pedidos que han sido entregados en un mes introducido por el usuario de cualquier año"***. Cuando seleccionamos esta opción, mostrará un formulario que nos permitirá introducir el mes en el que queremos conocer los pedidos entregados, una vez introducido habrá un botón Buscar, que me debe mostrar en un DataGridView con el resultado.
- **Consultar empleados de un jefe determinado:** tienes que elaborar un formulario que me muestre los datos de empleados, NO UTILIZAR DATAGRIDVIEW, resultado de la siguiente consulta: ***Devuelve un listado con el nombre, apellidos y email de los empleados cuyo jefe tiene un código de jefe igual a ?***. El código del jefe se solicitará al usuario.
- **Consultar clientes que no pagan:** tienes que elaborar un formulario que me muestre los datos de los clientes resultado de la siguiente consulta: ***Devuelve un listado con los clientes que han realizado algún pedido, pero no han realizado ningún pago.***
- **Consultar el precio del producto más caro y más barato:** construye un formulario que me proporcione la siguiente información: ***Calcula el precio de venta del producto más caro y más barato en una misma consulta.***

**DEBES CONTROLAR LAS EXCEPCIONES PARA EVITAR ACCESOS ERRÓNEOS A LA BASE DE DATOS.**