

## 1 ARCHIVO MAIN.PY

En este archivo .py tenemos la programación de la lógica de toda la aplicación. Aquí se gestiona lo que van a hacer los botones, la información que hay que introducir para manejar la base de datos y la generación de informes.

Lo primero que hace la aplicación es recoger la cantidad de pedidos que ya se han hecho para generar el id del próximo pedido.

Lo siguiente que se hace nada más iniciar la aplicación es generar los wizard de ayuda para las diferentes funciones. La estructura de todos ellos es:

Después de los wizards configuramos las tablas para asignar las columnas y el nombre de estas.

Seguidamente asignamos a los botones los métodos que tienen que ejecutar estos.

Por último del constructor generamos el Widget que nos permite visualizar los informes en la aplicación.

Funciones de este archivo:

```
def informeTodosPedidos(self):
```

Esta función recoge todos los pedidos que se han realizado de la base de datos y rellena un informe de esos pedidos. Recoge el id del producto, el nombre del producto, el precio y por último el coste total de esos pedidos, aunque esta opción es opcional y estará disponible en el wizard que te ayuda a generar este informe.

```
def rellenarInformePedido(self, idPedido:str):
```

Esta función genera el informe de solo un pedido, el que le indiquemos en el wizard que se muestra para generar un informe de un solo pedido.

```
def realizarPedido(self):
```

Con esta función se actualiza la cantidad de stock que hay de un producto en concreto.

```
def añadirPedido(self)
```

Esta función se ejecuta al presionar el botón de hacer pedido. Añade un pedido a la base de datos.

```
def actDimensiones(self):
```

Esta función actualiza las dimensiones finales para ponerlas todas en un solo String del tipo altura x ancho x largo.

```
def añadir(self):
```

Esta función se ejecuta al añadir un mueble. Recoge la información de los cuadros de texto donde se ponen la información de los campos del mueble. Añade este mueble generado a la base de datos.

```
def obtenerCantidades(self):
```

Esta función actualiza la tabla donde se muestran las cantidades que hay de cada producto. Se ejecuta cuando se realiza un pedido y actualiza al instante las cantidades de ese mueble.

```
def obtener(self):
```

Esta función actualiza los muebles que hay en la base de datos. Se ejecuta al añadir un mueble o puedes activarla manualmente desde el botón listar muebles.

```
def eliminar(self):
```

Esta función se ejecuta cuando desees borrar un mueble de la base de datos. Se ejecuta al presionar el botón eliminar mueble después de completar el wizard de ayuda.

```
def modificar(self):
```

Esta función modifica un mueble en la base de datos. Se ejecuta cuando presionamos el botón modificar mueble después de completar el wizard.

## 2 ARCHIVO GESTIONARMUEBLES.PY

Con este archivo se maneja la tabla de los muebles de la base de datos creada. Consta de solo dos funciones de añadir y eliminar un mueble.

```
def eliminarMueble(mueble)
```

Con esta función se elimina un mueble. Se le pasa un mueble, el que se desea eliminar, lo busca en la base de datos y al encontrarlo lo elimina.

```
def añadirMueble(mueble):
```

Con esta función se añade un mueble. Se le pasa un mueble, el que se desea añadir, y lo añade a la base de datos.

## 3 ARCHIVO GESTIONARPEDIDOS.PY

Con este archivo se maneja la tabla de los pedidos de la base de datos creada. Consta de solo dos funciones de añadir y eliminar un pedido.

```
def eliminarPedido(pedido):
```

Con esta función se elimina un pedido. Se le pasa un pedido, el que se desea eliminar, lo busca en la base de datos y al encontrarlo lo elimina.

```
def añadirPedido(pedido):
```

Con esta función se añade un pedido. Se le pasa un pedido, el que se desea añadir, y lo añade a la base de datos.

#### 4 ARCHIVO UI\_DESIGN.PY

En este archivo se encuentran todos los componentes generados desde Qt design y pasados después a un código de python.