

# Práctica final Java

## Simulador de Caja de Tienda con 15 Productos

### Descripción:

El objetivo de este ejercicio es crear un simulador de una caja de tienda utilizando una Interfaz de Usuario de Texto (TUI) compatible con `simple_screen`. El programa debe leer los productos de una tabla SQLite y permitir al usuario realizar compras interactuando únicamente a través de entradas de texto.

### Requisitos:

#### 1. Base de Datos SQLite:

- Partiendo del fichero csv que se aporta

```
1 id,nombre,precio_unitario
2 1,Manzana,0.5
3 2,Plátano,0.3
4 3,Naranja,0.7
5 4,Uvas,1.2
6 5,Lechuga,0.9
7 6,Zanahoria,0.4
8 7,Tomate,0.8
9 8,Patata,0.6
10 9,Cebolla,0.5
11 10,Pimiento,1.0
12
```

- Crearas una base de datos sqlite con la tabla **productos** definida como sigue.
  - `id` (INTEGER PRIMARY KEY)
  - `nombre` (TEXT)

- `precio_unitario` (REAL)

## 1. Interfaz de Usuario TUI:

- La pantalla de entrada debe mostrar el título del programa y una lista de los 10 productos con su código y precio unitario.
- Pedir al usuario el código del producto y el número de unidades.
- Si se introduce un código inexistente, debe mostrar un mensaje de error.
- Si se introduce el código de fin de compra, debe mostrar el total global.
- En cada interacción, mostrar al lado de cada producto el número de unidades compradas y su precio parcial.
- Acumular las unidades si se introducen más de una vez para el mismo producto.

## 2. Finalización de la Compra:

- Preguntar al usuario si desea realizar una nueva compra.
- Si el usuario responde 'S', el programa debe reiniciarse; si no, debe finalizar.

## Diseño de Pantalla:

|    |                               |           |        |          |        |
|----|-------------------------------|-----------|--------|----------|--------|
| 1  | SIMULADOR DE CAJA             |           |        |          |        |
| 2  | =====                         |           |        |          |        |
| 3  | Código                        | Producto  | Precio | Unidades | Total  |
| 4  | -----+-----+-----+-----+----- |           |        |          |        |
|    | --                            |           |        |          |        |
| 5  | 1                             | Manzana   | 0.50 € | 0        | 0.00 € |
| 6  | 2                             | Plátano   | 0.30 € | 0        | 0.00 € |
| 7  | 3                             | Naranja   | 0.70 € | 0        | 0.00 € |
| 8  | 4                             | Uvas      | 1.20 € | 0        | 0.00 € |
| 9  | 5                             | Lechuga   | 0.90 € | 0        | 0.00 € |
| 10 | 6                             | Zanahoria | 0.40 € | 0        | 0.00 € |
| 11 | 7                             | Tomate    | 0.80 € | 0        | 0.00 € |
| 12 | 8                             | Patata    | 0.60 € | 0        | 0.00 € |

|    |   |  |          |  |        |  |   |  |        |
|----|---|--|----------|--|--------|--|---|--|--------|
| 13 | 9   |  | Cebolla  |  | 0.50 € |  | 0 |  | 0.00 € |
| 14 | 10  |  | Pimiento |  | 1.00 € |  | 0 |  | 0.00 € |
| 15 | -----+-----+-----+-----+-----+-----+-----+-----+----- |  |          |  |        |  |   |  |        |
|    | --  |  |          |  |        |  |   |  |        |
| 16 | Total   |  |          |  |        |  |   |  | 0.00 € |
| 17 | Código del producto o X para terminar compra:         |  |          |  |        |  |   |  |        |
| 18 | Ingresa el número de unidades:                        |  |          |  |        |  |   |  |        |
| 19 |   |  |          |  |        |  |   |  |        |
| 20 | ¿Nueva compra? (S/N):                                 |  |          |  |        |  |   |  |        |

...

## Enunciado para el Alumno:

### 1. Configura la base de datos SQLite:

- Crea una base de datos SQLite con una tabla `productos` utilizando el fichero `create_database` proporcionado
- Importa los datos del CSV de productos en la tabla `productos`.

### 2. Desarrolla la interfaz TUI:

- Diseña una interfaz de texto que muestre la lista de productos con sus códigos y precios unitarios.
- Implementa la funcionalidad para que el usuario pueda ingresar el código del producto y el número de unidades.
- Asegúrate de verificar si el código del producto existe y manejar errores adecuadamente.
- Muestra el total global y los detalles de la compra (unidades compradas y precio parcial) en cada interacción.
- Pregunta al usuario si desea realizar una nueva compra y maneja la respuesta para reiniciar o finalizar el programa.

### 3. Funcionalidades adicionales:

- Asegúrate de que las unidades compradas se acumulen si se introduce el mismo código de producto más de una vez.
- Muestra un mensaje de error si el código del producto no existe.

- Muestra un mensaje de error si el numero de unidades no es numero o es menor de 1

## Requisitos de desarrollo

1. Utiliza el patron MVC.
2. En modelos incluye
  - los modelos necesarios, al menos **Producto** y **Tiquet**.
  - Un DAO sqlite para la consulta de todos los productos
3. En vistas crea un control para el tiquet completo (con todos los productos) y controles genericos para los distintos Inputs, recuerda validarlos. como en clase.
4. En controlador crea el controlador de la aplicacion con los métodos `__init__` donde crearas tanto los controles visuales (importandolos de vista), como los modelos/daos (imporandolos de modelos) necesarios. Y el método `__run__` con el bucle principal de aplicacion.
5. Recuerda que la aplicacion se lanzará con un fichero `main.py`.

## Entrega

Debes entregar un repositorio git en github con todo el codigo en la rama main.

Recuerda incluir

- un fichero `README.md` indicandome, como crear la base de datos con los productos.
- el fichero `requeriments.txt` con lo que debo instalar. Recuerda usar `simple_screen==0.1.16`

Ese repo debe ser publico o darme permisos de acceso de lectura para descargarlo y evaluarlo. En github soy `pigmonchu`.

El formulario de entrega de la practica es este

<https://forms.gle/19nxnp5jDiN8KQQd6> y la fecha máxima de entrega es

el 8 de septiembre a las 23:50 hora peninsular española.