

# Aplicación para una tienda especializada en vender productos estéticos.

## ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

Todos los módulos creados estarán interconectados, por compartir variables e información entre ellos.

### REQUISITOS RESPECTO A PRODUCTOS:

#### FUNCIONALES :

- Controlar los precios de los productos y ofrecer la posibilidad de operar con ellos.

Crearemos un modulo de edición de productos (para jefes y encargados de tienda)

- Proporcionar facturas de las ventas.

Crearemos un módulo de facturación que capture datos de ventas y genere facturas automáticamente (este nos permitirá además atribuirle a cada vendedor lo que vende y así tener datos de ventas de cada empleado)

- Controlar la cantidad de productos en almacén.

Implementaremos también un modulo que con cada venta conecte con los datos del almacén y los actualice

- interesa almacenar de los productos : código, marca, nombre comercial, precio, cantidad.

Base de datos de trabajadores con su código personal y productos conectada al modulo de ventas y facturación (en esta podemos tener la restricción para que solo pueda operar una persona cada vez)

#### NO FUNCIONALES :

- Operar con lector de código de barras y tarjetas de crédito.

Desarrollo de interfaces simples del programa para conexión de dispositivos con tecnologías de pago seguro y eficaz a través de los módulos de facturación y almacenamiento

- El tiempo de respuesta de la aplicación ha de ser lo menor posible.

Que las peticiones al sistema (Base de datos) sean de manera secuencial, que también puede ayudar a que cada petición sea exclusiva.

- No se podrán procesar dos peticiones a la vez, aunque haya varios equipos funcionando simultáneamente.

Requisito vetado desde la base de datos mientras opere una orden no puede finalizar otra

### REQUISITOS RESPECTO A TRABAJADORES:

## FUNCIONALES :

-La empresa también quiere almacenar información de sus trabajadores:

DNI, nombre Y apellidos, número de la Seguridad Social, fecha de nacimiento, teléfono y localidad.

Esto estará incluido en los datos del trabajador de la base de datos (acceso limitado únicamente a jefes)

-Llevar la cuenta de lo que vende cada trabajador.

Requisito cubierto en el modulo de facturas conectado a la base de datos en el apartado código de vendedor donde se le va anotando cada venta

## MODELO DE CICLO DE VIDA ELEGIDO - Modelo en Cascada con Realimentación

Este modelo es bueno para un proyecto como el que nos han encargado puesto que es rígido con pocos cambios, requisitos claros y muy definidos y no parece que vaya a haber mas evolución que cambios de datos personales o de productos o almacenaje, no en estructura.

### **Características:**

Es uno de los modelos más utilizados.

Se puede regresar a etapas anteriores para hacer modificaciones o depurar los errores.

Ideal para proyectos rígidos y con requisitos claros como el que nos abarca.

La parte mala es que los errores al finalizar el proyecto pueden provocar que haya que comenzar desde cero.

## DOCUMENTO DE DISEÑO DE ARQUITECTURA

**Lenguaje de Programación: Python**

**Herramientas: Django para el desarrollo web, SQLite para la base de datos, Git para control de versiones.**

**Justificación:** Python es versátil y fácil de aprender, Django facilita el desarrollo rápido y seguro de aplicaciones web, SQLite es ligero y adecuado para este proyecto, y Git permite un control eficiente de versiones.

La aplicación tendrá un modo de acceso para jefes con control total de datos de empleados y productos.

Los empleados desde su acceso únicamente anotaran ventas y entrada de mercancías con su código, esto generara factura automática y a través de la base de datos con archivos secuenciales (para accesos individuales) hará gestión de forma automática del inventario en almacén.

## Restantes Fases del Ciclo de Vida:

### 1. Pruebas:

- Objetivo: Garantizar la calidad y funcionamiento adecuado de la aplicación.
- Estrategia: Pruebas unitarias, de integración y de sistema. Automatización de pruebas para maximizar cobertura.

## **2. Implementación del Sistema:**

- Objetivo: Desplegar la aplicación en el entorno de producción.
- Estrategia: Utilizar servicios de alojamiento web y asegurar la configuración óptima del entorno.

## **3. Mantenimiento y Actualizaciones:**

- Objetivo: Corregir errores, realizar mejoras y adaptarse a cambios futuros.
- Estrategia: Establecer un plan de mantenimiento regular y la comunicación con el cliente

Como conclusión final aunque el Scrum parecía una buena opción por el trato directo con el cliente, ágil en el desarrollo, tratándose de un proyecto sin muchos requerimientos de online, y muy definido en cuanto al resto además de conciso, hace que no considere necesario un contacto excesivo con el cliente ahorrando tiempo y molestias, por ello finalmente me decante por el modelo en cascada con reorientación.