Testing Implantación y Mantenimiento de Sistemas: Informe Código y Requerimientos de la Guerra de los Testers

Sistema de Farmacia

## Muñante Quispe Piero Jesus, Rivera Rivera Joaquin Saul, Torres Arenas Guerhalt Randall

*Escuela Profesional de Ingeniería de Sistemas, Universidad Católica de Santa María Arequipa Perú*

[piero.munante@ucsm.edu.pe](mailto:piero.munante@ucsm.edu.pe)

[joaquin.rivera@ucsm.edu.pe](mailto:joaquin.rivera@ucsm.edu.pe)

[guerhalt.torres@ucsm.edu.pe](mailto:guerhalt.torres@ucsm.edu.pe)

**Abstracto—**En este sistema de farmacia desarrollado es una solución integral diseñada para optimizar la gestión de farmacias mediante funcionalidades avanzadas que abarcan la administración de inventarios, ventas, clientes, proveedores y reportes financieros. Este sistema, construido con lenguaje python, garantiza una experiencia de suario eficiente y segura.

# INTRODUCCIÓN

# El proyecto denominado "Sistema de Farmacia" tiene como objetivo desarrollar una solución integral para la gestión de operaciones diarias en una farmacia. Este sistema abarca desde la facturación y gestión de inventarios hasta el control de clientes y proveedores, proporcionando una plataforma eficiente y segura para optimizar el funcionamiento de la farmacia.

# DESARROLLO

# El desarrollo del sistema de farmacia se realizó utilizando tecnologías web modernas, con una arquitectura basada en Python y Django para el backend, y HTML, CSS y JavaScript para el frontend. El proyecto está estructurado en varios módulos para facilitar la escalabilidad y mantenimiento. A continuación, se describen los componentes principales del sistema:

1. **Descripción del Sistema:**

Descripción del Sistema El sistema de farmacia está compuesto por varios módulos interconectados que permiten una gestión eficiente y efectiva de las operaciones diarias. Los módulos principales incluyen:

* + - **Módulo de Administración de Productos:** Permite agregar, actualizar y eliminar productos del inventario. Este módulo asegura que la información del inventario esté siempre actualizada.
    - **Módulo de Gestión de Ventas:** Facilita la creación y registro de ventas, generando facturas y recibos. Permite a los usuarios registrar las ventas de manera rápida y precisa.
    - **Módulo de Gestión de Clientes:** Mantiene un registro de los clientes, permitiendo gestionar información personal y el historial de compras.
    - **Módulo de Reportes:** Genera reportes detallados sobre ventas, inventarios y otros aspectos operativos. Estos reportes ayudan en la toma de decisiones estratégicas. Cada módulo está diseñado para interactuar de manera fluida con los demás, proporcionando una experiencia de usuario coherente y eficiente.

# REQUERIMIENTOS DE LA GUERRA DE TESTER

Para garantizar la calidad del sistema, se implementaron diversos tipos de pruebas, incluyendo:

* **Pruebas Unitarias:** Se desarrollaron pruebas para cada función y método individual, asegurando que cada componente del sistema funcione correctamente de manera aislada.
* **Pruebas de Integración:** Verifican que los diferentes módulos del sistema interactúan correctamente entre si. Pruebas Funcionales: Aseguran que el sistema cumple con los requisitos funcionales establecidos. Pruebas de Usabilidad: Evaluaron la facilidad de uso del sistema desde la perspectiva del usuario final.
* **Pruebas de Carga y Rendimiento**: Evaluaron cómo se comporta el sistema bajo condiciones de carga elevada para asegurar que puede manejar el tráfico y las operaciones esperadas.
* **Pruebas de Seguridad:** Verificaron la robustez del sistema frente a posibles vulnerabilidades y ataques, asegurando que la información sensible esté protegida.
* **Pruebas de Regresión:** Aseguraron que las nuevas actualizaciones o cambios no introduzcan nuevos errores ni afecten negativamente a las funcionalidades existentes.

1. **CODIGO IMPLEMENTADO**
2. **CLIENTES:**

Models.py define los modelos de datos para los clientes.

A screen shot of a computer program

Description automatically generated

Views.py define las vistas para gestionar las operaciones relacionadas con clientes.

A screen shot of a computer

Description automatically generated

Urls.py define las rutas de URL para la vista del clientes.

A screen shot of a computer

Description automatically generated

Admin.py registra el modelo Cliente en el administrador de Django.

A screen shot of a computer

Description automatically generated

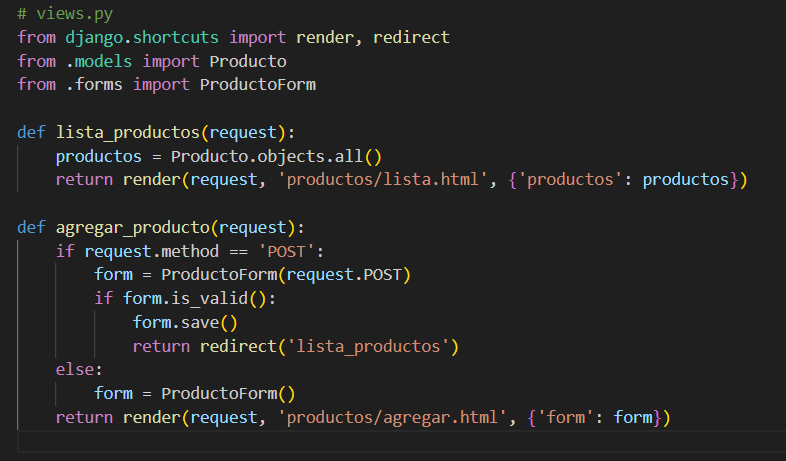
1. **PRODUCTO:**

Models.py define los modelos de datos de los productos.

A screen shot of a computer code

Description automatically generated

Views.py Define las vistas para gestionar las operaciones relacionadas con productos.



A computer screen with text

Description automatically generatedUrls.py define las rutas de URL para la vista del producto

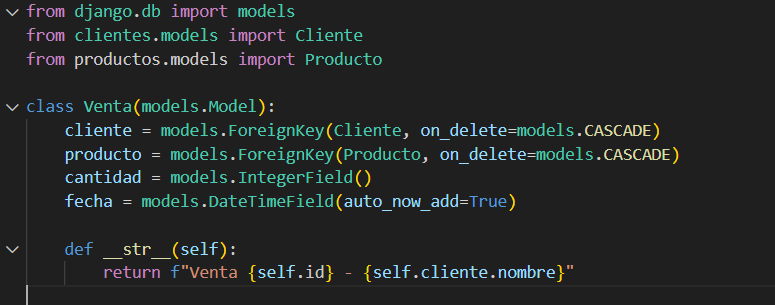
Admin.py registra el modelo Cliente en el administrador de Django.

A screen shot of a computer

Description automatically generated

1. **VENTAS:**

Models.py define los modelos de datos de las ventas.



Views.py Define las vistas para gestionar las operaciones relacionadas con ventas.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen shot of text

Description automatically generated

Urls.py define las rutas de URL para la vista de ventas.

A screen shot of a computer code

Description automatically generated

Admin.py registra el modelo Cliente en el administrador de Django.

A screen shot of a computer program

Description automatically generated

# SISTEMA DE FARMACIA

# Primero se va la carpeta de farmacias y se entra al cmd

# A screenshot of a computer Description automatically generated

# Activación del entorno virtual: Asegura que todas las dependencias del proyecto estén aisladas y correctamente configuradas.

# 

# Cambio de directorio: Navega al directorio donde se encuentra el proyecto Django.

# 

# Inicio del servidor de desarrollo: Inicia el servidor web de desarrollo de Django para probar y desarrollar la aplicación localmente.

# 

# En este punto se ejecuta el sistema de farmacia en Google poniendo localhost:8000

# A screenshot of a computer Description automatically generated

# Título de la Sección: En la parte superior de la pantalla se muestra el título "Productos", indicando que esta sección está destinada a la gestión de los productos de la farmacia.

# Botón de Agregar Productos: Un botón verde con el texto "Agregar Productos" y un ícono de signo más (+). Este botón permite al usuario añadir nuevos productos al inventario.

# Opciones de Exportación: Dos botones situados debajo del botón de agregar productos, etiquetados como "CSV" e "Imprimir". Estos permiten exportar la lista de productos en formato CSV o imprimirla directamente.

# Tabla de Productos: La mayor parte de la interfaz está ocupada por una tabla que lista los productos actualmente en el inventario.

# ID: Un identificador único para cada producto.

# Descripción: El nombre o descripción del producto.

# Costo $: El costo del producto.

# Precio $: El precio de venta del producto.

# Cant.: La cantidad disponible en inventario.

# Imagen: Una imagen asociada con el producto.

# Opciones: Acciones que se pueden realizar sobre cada producto, representadas por íconos. Las opciones disponibles parecen ser editar (ícono de lápiz) y eliminar (ícono de papelera).

# A screenshot of a computer Description automatically generated

# Título de la Sección:

# En la parte superior de la pantalla se muestra el título "Clientes", indicando que esta sección está destinada a la gestión de los clientes de la farmacia.

# Botón de Agregar Cliente:

# Un botón verde con el texto "Agregar Cliente" y un ícono de signo más (+). Este botón permite al usuario añadir nuevos clientes al sistema.

# Opciones de Exportación:

# Dos botones situados debajo del botón de agregar cliente, etiquetados como "CSV" e "Imprimir". Estos permiten exportar la lista de clientes en formato CSV o imprimirla directamente.

# Tabla de Clientes:

# La mayor parte de la interfaz está ocupada por una tabla que lista los clientes actualmente registrados en el sistema. Las columnas de la tabla incluyen:

# # ID: Un identificador único para cada cliente.

# Código: Un código de identificación del cliente.

# Nombre: El nombre del cliente.

# Teléfono: El número de teléfono del cliente.

# Opciones: Acciones que se pueden realizar sobre cada cliente, representadas por íconos. Las opciones disponibles parecen ser editar (ícono de lápiz) y eliminar (ícono de papelera).

# Barra de Búsqueda:

# A la derecha de la tabla, hay una barra de búsqueda que permite al usuario filtrar los clientes mostrados en la tabla mediante palabras clave.

# Paginación:

# En la parte inferior de la tabla, se muestra la paginación, indicando que actualmente se están mostrando de 1 a 2 de 2 registros, con controles para navegar entre diferentes páginas de registros.

# A screenshot of a computer Description automatically generated

# CONCLUSION

# El desarrollo de este sistema de farmacia ha demostrado ser una solución eficaz para la gestión de inventarios y ventas en un entorno farmacéutico. Las pruebas exhaustivas realizadas aseguran que el sistema es robusto, confiable y fácil de usar. La modularidad del código permite futuras expansiones y adaptaciones a nuevas necesidades. La implementación de un riguroso proceso de pruebas ha sido crucial para detectar y corregir errores tempranamente, garantizando un alto nivel de calidad en el producto final. Además, la capacidad del sistema para generar reportes detallados proporciona una herramienta valiosa para la toma de decisiones estratégicas en la gestión de la farmacia. La integración de funcionalidades como la administración de clientes, productos y ventas asegura que todas las operaciones críticas se realicen de manera eficiente y precisa.

# REFERENCIAS

1. Documentación de Django, "Tutorial Django desde cero y en español," Mi Diario Python. Disponible en: [http://pythondiario.com/2023/tutorial-django-desde-cero.html.](http://pythondiario.com/2023/tutorial-django-desde-cero.html)
2. E. García, "Cómo configurar Django con Postgres, Nginx y Gunicorn en Ubuntu 20.04," DigitalOcean. Disponible en: https:/[/www.digitaloc](http://www.digitalocean.com/community/tutorials/como-configurar-)e[an.com/community/tutorials/como-configurar-](http://www.digitalocean.com/community/tutorials/como-configurar-) django-con-postgres-nginx-y-gunicorn-en-ubuntu-20-04-es.
3. Documentación de Django, "Documentación oficial de Django en español," Django Project. Disponible en: <https://docs.djangoproject.com/es/3.2/>.
4. AdminLTE Documentation. (2024). Retrieved from <https://adminlte.io/docs/>
5. Bootstrap Documentation. (2024). Retrieved from <https://getbootstrap.com/docs/>