



**Desarrollo de un Sistema de Información Web para la Gestión y Manejo de Inventarios de la
IPSI Outtajiapulee en Uribí La Guajira**

Manual Técnico - Sistema de Gestión de Inventarios para IPSI
Trabajo de grado presentado como requisito parcial para optar al título de:
Ingeniero De Sistemas

Director (a): Omar Ernesto Torres Ladino Ph.D director(a)
Línea de Investigación: Desarrollo de Software

Leonardo Almazo Epinayu
21141715918

Universidad Antonio Nariño

Programa Ingeniería de Sistemas y Computación

Facultad de Ingeniería de Sistemas

Bogotá, Colombia

2023

Tabla de contenido

Descripción de la Aplicación	4
Objetivos del Sistema	4
Roles en el Sistema	4
Requisitos de Hardware y Software.....	5
Configuración Recomendada.....	5
Rendimiento del Sistema	5
Seguridad del Sistema.....	6
Escalabilidad del Sistema	6
Instalación y Configuración.....	6
Estructura del Sistema.....	6
Arquitectura del Sistema.....	8
Componentes Principales.....	8
Gestión de Usuarios	8
Gestión de Inventarios	8
Control de Inventarios.....	9
Descripción Detallada de Funciones.....	9
Flujo de Trabajo	10
Pasos para Tareas Comunes.....	10

Configuración del Sistema	10
Personalización Avanzada	10
Mantenimiento y Actualizaciones.....	11
Procedimientos de Mantenimiento	11
Monitoreo.....	11
Mantenimiento Regular	12
Instrucciones para Aplicar Actualizaciones y Cambios en el Sistema	12
Resolución de Problemas	13
Problemas Comunes y Soluciones	13
• Problema: Lentitud en la Carga de Páginas.....	13
Bibliografía	14

Descripción de la Aplicación

El Sistema de Gestión de Inventarios para IPSI es una aplicación web diseñada para optimizar y automatizar la gestión de inventarios en las Instituciones Prestadoras de Servicios de Salud Indígena (IPSI). La aplicación permite un seguimiento eficiente de productos médicos y farmacéuticos, control de niveles de stock y generación de informes detallados para la toma de decisiones informadas.

Objetivos del Sistema

El propósito principal del sistema es mejorar la eficiencia en la gestión de inventarios, reducir errores y proporcionar una visión integral del estado actual de los recursos en las IPSI.

Los objetivos específicos incluyen:

Facilitar el seguimiento en tiempo real de productos médicos y farmacéuticos.

Controlar los niveles de stock para evitar faltantes o excesos.

Generar informes detallados para respaldar la toma de decisiones.

Roles en el Sistema

1. Nuevo Empleado

Acceso limitado para el registro de nuevos empleados y usuarios.

2. Usuario Registrado

Acceso básico para funciones de gestión de inventarios.

3. Personal de IPSI

Acceso avanzado para la configuración de productos, clientes y proveedores.

4. Gerente

Acceso completo a todas las funcionalidades del sistema.

5. Administrador

Configuración avanzada y control total del sistema.

Requisitos de Hardware y Software

Servidor

Recomendación: Se sugiere un servidor con al menos 4 GB de RAM y un procesador de doble núcleo.

Sistema Operativo: Se prefiere Ubuntu 20.04 LTS.

Base de Datos: Motor de Base de Datos: MySQL 8.0.

Configuración Adicional: Se requiere la configuración de SQL_MODE como 'STRICT_TRANS_TABLES'.

Navegadores Compatibles: Google Chrome, Mozilla Firefox, Microsoft Edge.

Configuración Recomendada

Rendimiento del Sistema

Cache: Implementar un sistema de almacenamiento en caché utilizando Memcached o Redis.

Compresión de Recursos: Habilitar la compresión de recursos estáticos como CSS y JavaScript.

Optimización de Imágenes: Optimizar imágenes antes de cargarlas al sistema.

Seguridad del Sistema

Firewall: Configurar un firewall para limitar el tráfico no deseado.

Actualizaciones de Seguridad: Mantener actualizados todos los componentes del sistema.

Logs de Seguridad: Configurar logs de seguridad detallados.

HTTPS: Implementar el protocolo HTTPS con un certificado SSL válido.

Políticas de Contraseñas: Establecer políticas de contraseñas robustas.

Escalabilidad del Sistema

Balanceo de Carga: Configurar un sistema de balanceo de carga.

Escalabilidad Horizontal: Diseñar la arquitectura para escalar horizontalmente.

Monitorización de Rendimiento: Implementar herramientas de monitorización de rendimiento.

Instalación y Configuración del Aplicativo IPSI en PythonAnywhere

Descargar el Código Fuente del Repositorio GitHub: No necesitarás clonar el repositorio en tu máquina local. Puedes trabajar directamente en el entorno proporcionado por PythonAnywhere.

Configurar un Entorno Virtual: PythonAnywhere proporciona automáticamente un entorno virtual. No necesitas configurar uno manualmente.

Instalar Dependencias: Utiliza la consola de PythonAnywhere o un archivo requirements.txt para instalar las dependencias. Ejecuta el siguiente comando en la consola de PythonAnywhere:

```
bashCopy code
```

```
pip3 install -r /ruta/a/tu/proyecto/requirements.txt
```

Configurar Variables de Entorno: En PythonAnywhere, configura las variables de entorno directamente en el panel de control web.

Ve a la sección "Consoles" y selecciona la opción "Web".

En la consola web, establece las variables de entorno utilizando el comando export.

Realizar Migraciones: Ejecuta el siguiente comando para aplicar las migraciones en PythonAnywhere:

```
bashCopy code
```

```
python3 manage.py migrate
```

Iniciar el Servidor de Desarrollo: Utiliza el panel de control web para iniciar el servidor de desarrollo en PythonAnywhere.

```
bashCopy code
```

```
python3 manage.py runserver
```

Acceder a la Aplicación: Una vez que el servidor se inicie, puedes acceder a tu aplicación desde la URL proporcionada por PythonAnywhere.

bashCopy code

http://tu_usuario.pythonanywhere.com

Recuerda ajustar las rutas y comandos según la estructura específica de tu aplicación en PythonAnywhere.

Estructura del Sistema

Arquitectura del Sistema

- **Modelo.** El modelo define la estructura de la base de datos, incluyendo entidades como Usuario, Producto, Cliente, Proveedor, Entidades de Salud, Entrada de Inventarios, Salida de Inventario, Inventarios, Notificaciones, etc.
- **Plantillas.** La vista o plantilla, se encarga de la presentación y lógica de interacción del usuario mediante plantillas HTML y CSS.
- **Controlador.** El controlador maneja las solicitudes del usuario y gestiona la lógica del negocio a través de las vistas de Django.

Componentes Principales

Gestión de Usuarios

Registro y autenticación de empleados y usuarios.

Gestión de Inventarios

Registro y seguimiento de productos, clientes, proveedores y entidades de salud.

Control de Inventarios

Funcionalidades para el control de niveles de stock, movimientos de inventario e informes.

Funcionalidades Principales

Descripción Detallada de Funciones

- **Registro de Empleados y Usuarios**

Proceso para registrar y autenticar nuevos empleados y usuarios.

- **Configuración de Productos**

Funciones para añadir, modificar y eliminar productos.

- **Configuración de Clientes y Proveedores**

Gestión de información de clientes y proveedores.

- **Control de Inventarios**

Registro de movimientos de inventario y ajustes de stock.

- **Generación de Informes**

Creación de informes detallados sobre el estado del inventario.

- **Notificaciones**

Configuración de notificaciones automáticas para niveles bajos de stock.

Flujo de Trabajo

Pasos para Tareas Comunes

- **Registro de un Nuevo Empleado**
 1. Acceder al sistema.
 2. Completar el formulario de registro.
 3. Verificar la cuenta mediante correo electrónico.

- **Control de Niveles de Stock**
 1. Acceder al panel de control.
 2. Revisar el estado actual del inventario.
 3. Realizar ajustes según sea necesario.

Configuración del Sistema

- **Configuración Inicial**

Instrucciones para personalizar el sistema según las necesidades específicas de la IPSI.

Personalización Avanzada

Configuración avanzada para adaptar el sistema a requisitos específicos.

Mantenimiento y Actualizaciones

Procedimientos de Mantenimiento

- **Copias de Seguridad.** Las copias de seguridad son vitales para garantizar la integridad y disponibilidad de los datos. Siga estas instrucciones para realizar copias de seguridad periódicas:
- **Frecuencia de Copias de Seguridad.** Establezca una frecuencia regular (diaria o semanal) para las copias de seguridad.
- **Tipo de Copias de Seguridad.** Implemente copias de seguridad completas y diferenciales para optimizar el espacio de almacenamiento y reducir el tiempo de recuperación.
- **Almacenamiento de Copias.** Guarde las copias de seguridad en ubicaciones externas seguras, como servicios en la nube o dispositivos físicos. Asegúrese de que el personal autorizado tenga acceso a estas copias.

Monitoreo

El monitoreo constante del rendimiento del sistema ayuda a identificar problemas antes de afectar la operación normal. Siga estas pautas:

- **Herramientas de Monitoreo.** Implemente herramientas como Django Debug Toolbar para evaluar el rendimiento de consultas a la base de datos y la carga de páginas.
- **Alertas de Monitoreo.** Configure alertas para notificar anomalías en el rendimiento. Establezca umbrales específicos para cada métrica monitoreada.

Mantenimiento Regular

El mantenimiento regular asegura la estabilidad del sistema. Realice estas tareas periódicas:

- **Limpieza de Base de Datos:**
Programe tareas de limpieza para eliminar registros obsoletos.
- **Optimización de Consultas:**
Identifique y optimice consultas SQL para mejorar la eficiencia.
- **Actualizaciones de Dependencias:**
Verifique y aplique actualizaciones de dependencias, como bibliotecas de Python.

Actualizaciones del Sistema

Instrucciones para Aplicar Actualizaciones y Cambios en el Sistema

Aplique actualizaciones de manera planificada y cuidadosa con estas instrucciones:

- **Evaluación de Actualizaciones:**
Evalúe la compatibilidad y los posibles impactos antes de aplicar actualizaciones.
- **Entorno de Pruebas:**
Realice pruebas exhaustivas en un entorno de pruebas antes de aplicar actualizaciones en producción.

- **Procedimiento de Respaldo:**

Realice copias de seguridad completas antes de aplicar actualizaciones críticas.

- **Seguimiento de Actualizaciones:**

Mantenga un registro detallado de todas las actualizaciones aplicadas.

Resolución de Problemas

Problemas Comunes y Soluciones

- **Problema:** No se Puede Iniciar Sesión en el Sistema

- **Solución:**

1. Verifique la conexión a Internet.
2. Restablezca la contraseña si es necesario.
3. Revise los logs de errores para identificar problemas de autenticación.

- **Problema:** Lentitud en la Carga de Páginas

- **Solución:**

1. Verifique el rendimiento del servidor y la base de datos.
2. Optimice consultas y recursos estáticos.
3. Considere escalar el sistema horizontalmente.

Bibliografía

- [Documentación de Django](#)
- [MySQL Documentation](#)
- [Ubuntu Documentation](#)
- [Memcached Documentation](#)
- [Redis Documentation](#)
- [Django Debug Toolbar Documentation](#)