**Tesis de Proyecto de Desarrollo de Software: Aplicación de Gestión de Inventario para el Sector Mecánico**

**Documento de Presentación del Proyecto: Aplicación de Gestión de Inventario para XYZ Mecánica**Desarrollado por el equipo: Error Not Found 404 (ENF)  
Cliente: XYZ Mecánica  
Sector: Mecánico **Miembros del equipo:**

* Líder de Proyecto: [Nombre del líder]
* Desarrollador Backend: [Nombre del desarrollador backend]
* Desarrollador Frontend: [Nombre del desarrollador frontend]
* Tester y Documentador: [Nombre del tester y documentador]

**Fecha: [Fecha de entrega]**

**Índice**

1. Introducción
2. Justificación del Proyecto
3. Objetivos
4. Alcance del Proyecto
5. Metodología
6. Etapas de Desarrollo
7. Requisitos del Sistema
8. Plan de Proyecto
9. Roles y Responsabilidades del Equipo
10. Presupuesto
11. Plazos de Entrega
12. Conclusión

**1. Introducción**

Como desarrollador del equipo Error Not Found 404 (ENF), presento la propuesta para una aplicación de gestión de inventario que cubra las necesidades de XYZ Mecánica, una empresa del sector mecánico. Actualmente, XYZ Mecánica enfrenta desafíos en la administración de su inventario, manejándolo de manera manual, lo cual resulta ineficiente y propenso a errores. Este proyecto busca resolver estos problemas mediante una aplicación que facilite la organización, consulta y actualización de las existencias de piezas y repuestos.

**1. Resumen Ejecutivo**

Este documento describe el desarrollo de una aplicación de gestión de inventario diseñada para satisfacer las necesidades de XYZ Mecánica. La propuesta busca resolver los problemas de control manual de inventario que enfrenta actualmente la empresa, mediante una solución que reduce errores humanos, optimiza el tiempo de gestión y facilita el acceso a reportes y análisis en tiempo real. La aplicación será desarrollada en Python, utilizando una base de datos en SQLite y empleando la metodología de desarrollo en cascada para asegurar un proceso estructurado y controlado. Se contempla una interfaz amigable, reportes automatizados y, opcionalmente, un portal web para el acceso remoto. Este proyecto representa una inversión en eficiencia y organización para XYZ Mecánica, y el equipo Error Not Found 404 (ENF) se compromete a entregar una solución de calidad en el tiempo y presupuesto establecidos.

**2. Justificación del Proyecto**

La gestión manual del inventario puede llevar a errores en el registro de entradas y salidas de stock, falta de visibilidad en el estado actual de las existencias, y dificultades en el control de reposiciones. Una aplicación de software específicamente diseñada para la gestión de inventario ofrecerá:

* **Optimización del tiempo** mediante una actualización automática del stock en cada entrada o salida.
* **Reducción de errores humanos** en el registro y actualización de inventario.
* **Control de niveles mínimos de inventario** mediante alertas que aseguren la disponibilidad de piezas.
* **Reportes en tiempo real** para una mejor planificación de la demanda y toma de decisiones informada.

**3. Objetivos**

**3.1 Objetivo General**

Desarrollar una aplicación de software en Python que permita gestionar el inventario de XYZ Mecánica de forma eficiente y segura, optimizando el registro de piezas y ofreciendo alertas y reportes automáticos para la toma de decisiones.

**3.2 Objetivos Específicos**

1. Diseñar una interfaz amigable que permita a los usuarios gestionar y consultar el inventario.
2. Implementar alertas automáticas para avisar cuando los niveles de stock están bajos.
3. Crear una base de datos en SQLite para almacenar información detallada de las piezas.
4. Generar reportes de inventario en Excel y PDF opcionalmente.
5. Proporcionar, opcionalmente, un portal web seguro para la consulta de inventarios en tiempo real desde cualquier dispositivo con acceso a internet.

**4. Alcance del Proyecto**

El sistema de gestión de inventario cubrirá las siguientes áreas:

* **Gestión de Entradas y Salidas**: Registro y actualización automática de la cantidad de cada pieza en inventario.
* **Control de Stock**: Monitoreo del inventario para mantener niveles óptimos.
* **Alertas de Reabastecimiento**: Notificaciones automáticas al acercarse al nivel mínimo de inventario.
* **Generación de Reportes**: Reportes detallados en Excel (y PDF opcional) para el análisis de inventarios.
* **Acceso Web Opcional**: Un portal opcional para acceso remoto, de uso exclusivo para consulta de inventarios.

**5. Metodología**

El proyecto seguirá la metodología de desarrollo en cascada, la cual establece una secuencia de fases que deben completarse en orden antes de avanzar. Este enfoque garantiza una estructura organizada y un control claro de cada etapa del desarrollo.

**6. Etapas de Desarrollo**

**6.1 Investigación y Recopilación de Información**

En esta fase, realizaremos reuniones con el equipo de XYZ Mecánica para comprender sus requerimientos específicos y documentar las funcionalidades esenciales de la aplicación.

**6.2 Análisis de Requisitos**

Con la información recopilada, identificaremos los requisitos del sistema, incluyendo funciones específicas, estructura de datos y requerimientos técnicos.

**6.3 Diseño del Sistema**

En esta fase se diseñarán:

* **Diagrama de base de datos**: Para estructurar el almacenamiento y la consulta de datos.
* **Arquitectura modular**: Para organizar las funcionalidades y crear un sistema flexible y escalable.
* **Flujo de interfaz de usuario**: Para una navegación intuitiva y simple.

**6.4 Implementación**

Desarrollaremos la aplicación en Python y sus módulos esenciales:

1. **Interfaz de Usuario**: Utilizando Tkinter para una experiencia de usuario simple e intuitiva.
2. **Base de Datos en SQLite**: Para almacenar toda la información del inventario.
3. **Módulo de Reportes**: Exportación de datos en formatos como Excel y opcionalmente en PDF, usando librerías como Pandas y FPDF.
4. **Alertas de Inventario**: Implementación de alertas automáticas para reabastecimiento.
5. **Portal Web Opcional**: Construido con Flask para permitir la consulta remota de inventarios.

**6.5 Pruebas**

Cada módulo será probado individualmente (pruebas unitarias) y en conjunto (pruebas de integración) para asegurar la funcionalidad y seguridad de los datos.

**6.6 Documentación**

Se generará una guía de usuario para XYZ Mecánica, además de la documentación técnica necesaria para el equipo de desarrollo, con detalles del sistema, códigos de referencia y manual de administración.

**6.7 Implementación Final y Capacitación**

Instalaremos la aplicación en los equipos de XYZ Mecánica y ofreceremos una capacitación al equipo de operaciones para su uso.

**7. Requisitos del Sistema**

**7.1 Requisitos de Hardware**

* Procesador Intel Core i5 o superior.
* Memoria RAM mínima de 8 GB.
* Espacio en disco mínimo de 500 MB.

**7.2 Requisitos de Software**

* Windows 10 o superior.
* Python 3.x y las librerías necesarias (Tkinter, Pandas, SQLite, Flask para la página web).
* Microsoft Excel (para reportes).

**8. Plan de Proyecto**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase** | **Tiempo Estimado** | **Fecha de Inicio** | **Fecha de Fin** |
| Investigación y Recopilación | 2 semanas | 1/11/2023 | 15/11/2023 |
| Análisis de Requisitos | 1 semana | 16/11/2023 | 22/11/2023 |
| Diseño del Sistema | 1 semana | 23/11/2023 | 30/11/2023 |
| Implementación | 4 semanas | 1/12/2023 | 31/12/2023 |
| Pruebas | 1 semana | 2/01/2024 | 8/01/2024 |
| Documentación | 1 semana | 9/01/2024 | 15/01/2024 |
| Implementación Final y Soporte | 1 semana | 16/01/2024 | 22/01/2024 |

**Duración total**: Aproximadamente 12 semanas.

**9. Roles y Responsabilidades del Equipo**

**Equipo ENF**

1. **Líder de Proyecto**: Coordina todas las fases y es el contacto directo con XYZ Mecánica.
2. **Desarrollador Backend**: Responsable del desarrollo de la base de datos y de las funcionalidades principales de la aplicación.
3. **Desarrollador Frontend**: Diseña e implementa la interfaz de usuario y la página web opcional.
4. **Tester y Documentador**: Realiza pruebas de calidad, documenta el código y genera los manuales de usuario.

**10. Presupuesto**

|  |  |
| --- | --- |
| **Concepto** | **Costo** |
| Investigación y Recopilación | 500 EUR |
| Análisis y Diseño | 600 EUR |
| Desarrollo del Sistema | 3000 EUR |
| Pruebas | 400 EUR |
| Documentación | 300 EUR |
| Implementación y Soporte | 200 EUR |
| **Total** | **5000 EUR** |

**11. Plazos de Entrega**

Se propone un calendario de entregas parciales en conjunto con reuniones de avance quincenales. El objetivo es que la aplicación esté completamente funcional y probada para el 22 de enero de 2024.

**12. Conclusión**

Esta aplicación resolverá los problemas de gestión de inventario de XYZ Mecánica, proporcionando una solución confiable y fácil de usar que ahorrará tiempo y mejorará la precisión de las operaciones de la empresa. La metodología en cascada garantiza un desarrollo organizado, y el equipo de Error Not Found 404 (ENF) se compromete a ofrecer un producto de alta calidad dentro de los plazos y presupuesto acordados.

Ideas a implementar  
A qué sector y cliente va dirigido:

El proyecto que describes está orientado a mejorar la eficiencia en la gestión de inventarios dentro del sector mecánico, específicamente para una empresa llamada XYZ Mecánica. Sin embargo, el enfoque y las características del sistema pueden aplicarse en varios sectores y tipos de clientes que dependen del control detallado de inventarios, como:

### 1. **Sector de Reparación y Mantenimiento (Automotriz, Electrónica, etc.)**

* **Tipo de Cliente**: Talleres mecánicos, tiendas de autopartes, centros de servicio electrónico.
* **Razón de Ajuste**: Similar a XYZ Mecánica, estos negocios tienen un flujo constante de piezas y repuestos, y la gestión manual de inventarios a menudo es ineficiente y propensa a errores. Un sistema como este permitiría un control más riguroso de las entradas y salidas de componentes, optimizando la disponibilidad de piezas y reduciendo el tiempo de búsqueda en almacén.

### 2. **Sector de Manufactura y Producción**

* **Tipo de Cliente**: Fábricas de ensamblaje, empresas de producción a pequeña o mediana escala, fabricantes de maquinaria.
* **Razón de Ajuste**: La manufactura requiere control preciso de materiales y componentes para evitar retrasos en la producción. Un sistema de inventario como el que describes ayudaría a estos negocios a gestionar su stock en tiempo real, previendo faltantes y facilitando el reabastecimiento eficiente.

### 3. **Sector Comercial (Distribuidores y Mayoristas)**

* **Tipo de Cliente**: Distribuidores de suministros industriales, ferreterías, almacenes de productos al por mayor.
* **Razón de Ajuste**: Este tipo de empresas maneja altos volúmenes de productos que necesitan control constante para asegurar disponibilidad. Las alertas de inventario mínimo y los reportes en tiempo real serían sumamente beneficiosos para la planificación de compras y la optimización del inventario.

### 4. **Sector de Salud (Clínicas y Hospitales)**

* **Tipo de Cliente**: Farmacias hospitalarias, almacenes de suministros médicos.
* **Razón de Ajuste**: La gestión de inventarios en hospitales y clínicas es crítica, especialmente para el reabastecimiento de insumos y medicamentos. Las alertas automáticas sobre niveles bajos de stock y la capacidad de generar reportes detallados ayudarían a estos clientes a evitar situaciones de desabastecimiento.

### 5. **Sector de Servicios (Logística y Almacenamiento)**

* **Tipo de Cliente**: Empresas de logística, centros de distribución, servicios de almacenamiento.
* **Razón de Ajuste**: Para las empresas que gestionan productos de diferentes clientes o que manejan grandes cantidades de stock, un sistema de inventario automatizado asegura que los productos se rastreen y gestionen eficientemente. Las capacidades de generación de reportes y de consulta remota facilitarían la toma de decisiones rápidas y basadas en datos.

En conclusión, este tipo de sistema puede enfrentarse a una amplia gama de sectores donde el control de inventarios y la reducción de errores humanos son esenciales. Adaptarlo a cada sector específico solo requeriría ajustes en la interfaz de usuario o en ciertas funcionalidades, dependiendo de las necesidades del cliente y la naturaleza de los productos gestionados.

eguridad en los datos y por qué he elegido python y explicaciones técnicas, lenguaje de programación y lenguajes empleados:

**Seguridad en los Datos**

La seguridad de los datos es crucial para cualquier sistema de gestión de inventarios, especialmente en empresas como XYZ Mecánica que dependen de la precisión y disponibilidad de la información para su operación diaria. Para garantizar la seguridad, se han implementado varias prácticas y herramientas:

1. **Autenticación y Autorización**:
   * Solo el personal autorizado tendrá acceso a la aplicación. Mediante un sistema de autenticación, aseguramos que solo los usuarios registrados puedan ingresar.
   * Se implementará un sistema de control de roles, permitiendo que ciertos usuarios tengan permisos limitados, evitando así modificaciones no autorizadas en el inventario.
2. **Cifrado de Datos**:
   * Aunque SQLite no tiene cifrado nativo, Python permite el uso de librerías como **PyCrypto** o **SQLCipher** para cifrar la base de datos y protegerla contra accesos no autorizados.
3. **Backups Automáticos**:
   * La aplicación realizará copias de seguridad automáticas de la base de datos a intervalos regulares, almacenándolas en un servidor seguro o en la nube. En caso de pérdida de datos o fallos, se podrá restaurar la información fácilmente.
4. **Protección contra Inyecciones SQL**:
   * Usaremos consultas parametrizadas en las interacciones con la base de datos para prevenir ataques de inyección SQL. Esto asegura que los datos ingresados por el usuario sean tratados de manera segura, reduciendo riesgos.
5. **Manejo Seguro de Errores**:
   * La aplicación tendrá un sistema de manejo de errores que registra fallos sin mostrar detalles técnicos al usuario final, protegiendo así información sensible sobre la estructura del sistema.

**Elección de Python**

**Python** fue seleccionado como el lenguaje de programación principal del proyecto debido a las siguientes razones:

1. **Simplicidad y Facilidad de Uso**:
   * Python es conocido por su sintaxis clara y su estructura fácil de entender, lo cual acelera el desarrollo y facilita el mantenimiento del código. Esto es beneficioso tanto para el equipo de desarrollo como para futuros desarrolladores que trabajen en el proyecto.
2. **Amplia Comunidad y Librerías Especializadas**:
   * Python cuenta con una gran cantidad de librerías que permiten desarrollar rápidamente las funcionalidades requeridas. Para este proyecto, se emplearán:
     + **Tkinter** para la interfaz gráfica de usuario (GUI).
     + **SQLite3** para la gestión de la base de datos de manera integrada y eficiente.
     + **Pandas** para la manipulación de datos y generación de reportes.
     + **FPDF** para la creación de reportes en formato PDF, ofreciendo opciones de personalización y exportación.
3. **Compatibilidad con SQLite**:
   * Python se integra perfectamente con SQLite, una base de datos ligera y sencilla que es ideal para aplicaciones de tamaño mediano como esta. SQLite es fácil de desplegar y gestionar, y no requiere configuraciones de servidor complicadas.
4. **Facilidad para Integración Web**:
   * Aunque la aplicación principal es de escritorio, se consideró un portal web opcional para consultas remotas. Python facilita esta integración gracias a frameworks como **Flask**, que permite crear aplicaciones web simples y eficientes. Esto permitirá a XYZ Mecánica acceder al inventario desde cualquier dispositivo con acceso a Internet.
5. **Soporte para Pruebas**:
   * Python ofrece herramientas de pruebas como **unittest** o **pytest**, que permiten realizar pruebas unitarias e integrales, asegurando que cada módulo funcione correctamente antes de su implementación final.

**Explicación Técnica: Lenguajes y Librerías Empleados**

1. **Lenguaje Principal: Python**
   * Python es el núcleo del desarrollo, usado tanto para la lógica de negocio como para la interfaz gráfica. Su enfoque en la simplicidad y su rica colección de librerías hacen que sea una excelente opción para aplicaciones de gestión.
2. **Librerías Empleadas en el Proyecto**:
   * **Tkinter**: Usado para desarrollar la interfaz gráfica. Tkinter permite construir aplicaciones de escritorio en Python de manera sencilla, con controles como botones, cuadros de texto y menús.
   * **SQLite**: Una base de datos ligera y rápida que es fácil de integrar con Python. Con SQLite, los datos de inventario se almacenan de manera estructurada y accesible para la aplicación.
   * **Pandas**: Librería poderosa para el manejo de datos. Nos permite organizar, filtrar y manipular los datos de inventario, generando reportes en Excel.
   * **FPDF**: Utilizada para exportar datos en formato PDF. FPDF permite personalizar el diseño de los reportes, facilitando la creación de documentos profesionales y estructurados.
   * **Flask (opcional)**: En caso de integrar el portal web, Flask servirá como framework backend para gestionar las solicitudes y proporcionar acceso remoto seguro a los datos de inventario.
3. **SQL y Python-SQLite**:
   * El uso de SQLite con Python permite ejecutar consultas SQL para la gestión de inventarios. Las consultas SQL son parametrizadas para evitar inyecciones y garantizar la integridad de los datos.

**Conclusión**

Elegir Python para el desarrollo de esta aplicación asegura una solución eficaz y flexible para XYZ Mecánica, permitiendo cumplir con todos los requisitos del proyecto, desde la interfaz gráfica hasta la seguridad de los datos. Python, con sus librerías y frameworks, permite al equipo de Error Not Found 404 (ENF) desarrollar un sistema robusto, escalable y fácil de mantener, brindando una solución de alta calidad al cliente.

4o