1. **Briefing**
2. **Consideraciones Técnicas**
3. **Componentes Pro**
4. **Análisis DAFO**
5. **Tecnologías usadas**
6. **Diagrama de flujo**
7. **Código fuente**
8. **Ideas para mejorar a largo plazo**
9. **Conclusión**

Briefing

Proyecto para la venta de componentes de pc y arreglar PCs

**Objetivo:**

Desarrollo de un programa de Python para la venta de componentes y pc con el cual se puedan hacer inventarios, ventas, reparaciones y agregar clientes de la mejor manera posible

**Funciones principales:**

**Facturación y contabilidad:**

Registrar los gastos y los ingresos

**Ventas:**

Posibilidad de venta de componentes de pc o PCs completos a disposición del cliente y reparar PCs

Registro de las ventas

Hacer descuentos cada X tiempo

**Clientes:**

Registro de clientes con información personal y numero de tlf historial de las compras de los clientes

**Inventario:**

Registro de los componentes que se venden y nos llegan de nuestro proveedor

Actualización de inventario

**Consideraciones Técnicas:**

Lenguaje de Programación:

Python 3.x para el desarrollo del programa.

**Documentación:**

Creación de documentación clara y detallada para facilitar la lectura, el mantenimiento y la comprensión del código fuente del programa.

**Desarrollo Iterativo:**

El desarrollo del programa se llevará a cabo en iteraciones, con revisiones periódicas para recoger

comentarios y realizar ajustes según sea necesario.

**Entregables:**

Código fuente del programa.

Documentación técnica y de usuario.

**Consideraciones Adicionales:**

El programa debe ser escalable para adaptarse a futuras expansiones o modificaciones.

Se fomenta el uso de buenas prácticas de programación y diseño.

Componentes Pro: Solución Tecnológica Tiendas de Componentes Electrónicos **🚀**

**Propuesta de Valor**

Componentes Pro es un sistema de gestión desarrollado en Python 3.12.6 que automatiza y optimiza cada aspecto de tu negocio de componentes electrónicos, resolviendo problemas comunes como:

* 🛠️ **Gestión compleja de inventarios**
* 📉 **Dificultad en el seguimiento de ventas**
* 👥 **Control ineficiente de clientes**
* 📊 **Reportes poco precisos**

**Funcionalidades Estratégicas**

1. **Gestión de Inventario Inteligente** 🗃️
   * Control en tiempo real de 8+ componentes
   * Alertas automáticas de stock bajo
   * Seguimiento detallado: precio, existencias
   * Actualización instantánea tras cada venta
2. **Gestión Integral de Clientes** 👤
   * Registro completo de datos personales
   * Historial detallado de compras
   * Perfil con información de contacto
   * Seguimiento de preferencias
3. **Sistema de Ventas Avanzado** 💳
   * Facturación automática
   * Generación de tickets de venta
   * Reducción instantánea de inventario
   * Cálculo automático de totales
4. **Reportes Gerenciales** 📈
   * Análisis de ventas mensuales
   * Ranking de productos más vendidos
   * Proyección de inventario
   * Informes personalizables

**Tecnología y Beneficios**

* **Tecnología**: Python 3.12.6, librerías Rich y Prompt Toolkit.
* **Interfaz moderna**: Consola, escalabilidad garantizada y personalización bajo demanda.
* **Beneficios**:  
  **✅** Control total del negocio  
  **✅** Reducción de errores administrativos  
  **✅** Toma de decisiones basada en datos  
  **✅** Mejora de la experiencia del cliente  
  **✅** Ahorro de tiempo en gestión

**Inversión y Contacto**

* 💰 Inversión desde 499€
* 🚀 Implementación inmediata
* 🔧 Soporte técnico incluido
* 📈 Retorno garantizado

**Llamada a la Acción**

¿Listo para llevar tu negocio al siguiente nivel? 🎯  
📞 Solicita tu demo personalizada: [Jorge.enrique@itgranada.com](mailto:Jorge.enrique@itgranada.com)

Dafo

**Fortalezas**

1. **Interfaz Visual Atractiva**
   * Uso efectivo de la biblioteca Rich para crear una interfaz colorida y visual
   * Menús bien estructurados y fáciles de navegar
2. **Funcionalidad Completa**
   * Sistema integral que cubre las principales necesidades de una tienda
   * Gestión completa de clientes, productos, ventas y facturación
3. **Estructura de Datos Organizada**
   * Uso eficiente de diccionarios para almacenar la información
   * Buena organización de los datos de productos por categorías
4. **Feedback al Usuario**
   * Mensajes claros de éxito y error
   * Barras de progreso para operaciones
   * Confirmaciones visuales de las acciones realizadas

**Debilidades**

1. **Persistencia de Datos**
   * No hay sistema de almacenamiento permanente (base de datos o archivos)
   * Los datos se pierden al cerrar el programa
   * No hay sistema de backup
2. **Validación de Datos**
   * Algunas validaciones podrían ser más robustas
   * Falta manejo de algunos casos de error
3. **Arquitectura del Código**
   * Código muy largo en un solo archivo
   * Falta de modularización y separación de responsabilidades
   * Ausencia de clases y objetos para mejor organización
4. **Seguridad**
   * No hay sistema de autenticación de usuarios
   * Falta de encriptación de datos sensibles
   * Sin control de acceso a diferentes funcionalidades

**Oportunidades**

1. **Mejoras Técnicas**
   * Implementación de una base de datos SQL
   * Desarrollo de una versión web del sistema
   * Integración con APIs de pagos y envíos
2. **Expansión de Funcionalidades**
   * Añadir sistema de inventario más completo
   * Implementar sistema de reportes y análisis
   * Incluir gestión de proveedores
3. **Mejoras de Negocio**
   * Incorporar sistema de fidelización de clientes
   * Añadir módulo de marketing y promociones
   * Implementar análisis de ventas y tendencias
4. **Integración y Escalabilidad**
   * Posibilidad de integración con otros sistemas
   * Exportación de datos a formatos estándar
   * Desarrollo de versión multi-tienda

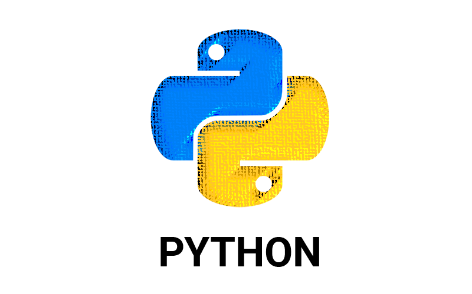
**Amenazas**

1. **Competencia**
   * Existencia de soluciones comerciales más completas
   * Sistemas ERP con más funcionalidades
   * Soluciones cloud más escalables
2. **Técnicas**
   * Dependencia de bibliotecas externas
   * Posibles problemas de rendimiento con grandes volúmenes de datos
   * Limitaciones de la interfaz de consola
3. **Operativas**
   * Necesidad de formación para usuarios
   * Resistencia al cambio por parte de usuarios
   * Tiempo de adaptación al sistema
4. **Mantenimiento**
   * Necesidad de actualizaciones regulares
   * Gestión de bugs y problemas
   * Costos de mantenimiento y mejoras

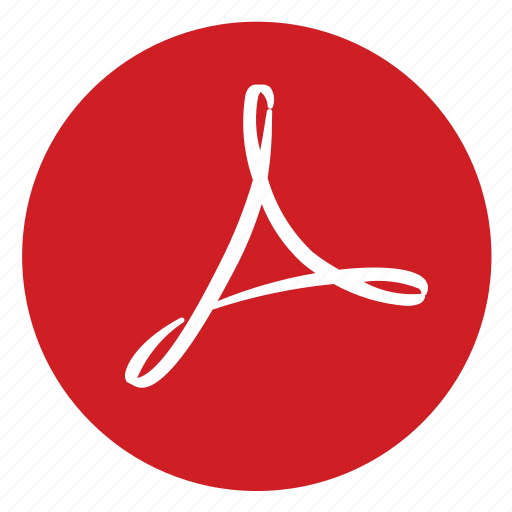
Tecnologías usadas

Para desarrollar el programa hemos utilizado visual studio code y un portátil.

Hemos utilizado Python como lenguaje de programación ya que es muy robusto y fácil de usar junto a los módulos de Rich y prompt\_toolkit para darle más interactividad y los incorporados en Python por defecto.



Para la creación de la documentación hemos usado Word para la redacción de la documentación y acrobat para la creación de pdf

 +

Para la creación de diagramas de flujo hemos usado Clic Chart

Para la consulta de dudas y ayudarme en la creación hemos usado la IA de Claude

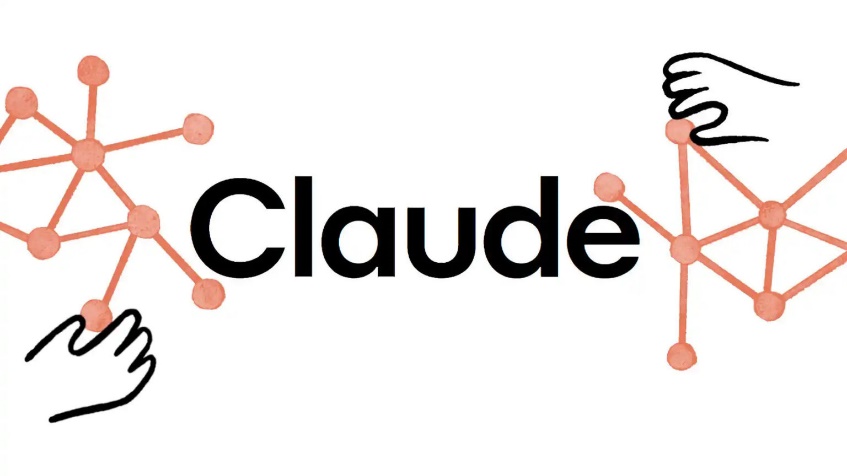


Diagrama de flujo

Diagrama de flujo lo puedes ver en el siguiente enlace: [Diagrama.png](Complementos/PRO-Jorge_Enrique_Fernandez-Diagrama.png)

Código fuente

Código fuente del proyecto en el enlace siguiente: [Tienda.py](Complementos/Pro-Jorge_Enrique_Fernandez-Tienda.py)

**Ideas para mejorar a largo plazo**

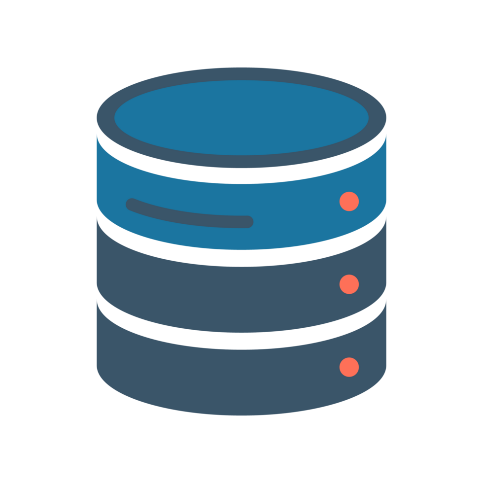
1. **Interfaz Web**

* Añadir un sistema de autenticación web para usuarios
* Implementar un diseño con CSS y HTML



1. **Base de Datos**

* Migrar los diccionarios actuales a una base de datos SQL
* Crear relaciones entre las tablas (clientes, productos, ventas)
* Implementar un sistema de respaldo automático de datos



1. **Nuevas Funcionalidades**

* Sistema de búsqueda avanzada de productos y clientes
* Filtros por categorías y precios
* Generación de reportes de ventas en PDF
* Sistema de notificaciones cuando el stock esté bajo
* Historial de cambios de precios
* Sistema de descuentos y promociones

1. **Experiencia de Usuario**

* Añadir gráficos de ventas y estadísticas
* Implementar un carrito de compras
* Sistema de valoraciones de productos
* Historial detallado de compras por cliente



1. **Seguridad**

* Implementar validación de datos más robusta
* Sistema de roles (administrador, vendedor, cliente)
* Sistema de recuperación de contraseñas



Conclusión

El proyecto es un sistema de gestión para una tienda de componentes informáticos desarrollado en Python. Cuenta con cuatro funcionalidades principales: gestión de clientes, control de inventario, registro de ventas y facturación. Destaca por su interfaz visual mejorada usando la biblioteca Rich y un sistema de menús intuitivo, siendo una base sólida para futuras mejoras.