

Smart Cart - Asistente de Compra Inteligente

Sistema de Punto de Venta (POS) con Inteligencia Artificial para el Reconocimiento de Voz y Recomendaciones de productos a comprar

Propósito

Desarrollar un Software de Punto de Venta (POS) moderno que integre Inteligencia Artificial (IA) para ofrecer recomendaciones inteligentes basadas en patrones de compra y permita realizar ventas mediante reconocimiento de voz, además de las interacciones tradicionales con teclado y mouse. El sistema estará diseñado como una aplicación web y móvil, utilizando tecnologías modernas en backend, frontend y bases de datos.

Descripción del Sistema

Este proyecto les permitirá aplicar conceptos avanzados de desarrollo de sistemas de gestión empresarial, inteligencia artificial y desarrollo full-stack para crear un sistema POS innovador y funcional. Al integrar recomendaciones inteligentes y reconocimiento de voz, estarán desarrollando una solución que no solo mejora la experiencia del cliente, sino que también optimiza la eficiencia operativa de una tienda de equipos electrónicos.

El sistema POS será una solución integral para tiendas de equipos electrónicos que busca mejorar la experiencia del cliente y optimizar la gestión de ventas. Incluirá las siguientes características principales:

1. Recomendaciones Inteligentes:
 - Utiliza algoritmos de Machine Learning (ML) para analizar transacciones históricas y generar sugerencias personalizadas de productos complementarios.
 - Ejemplo: Si un cliente agrega un teléfono inteligente al carrito, el sistema recomendará auriculares inalámbricos o protectores de pantalla basados en patrones de compra previos.
2. Venta mediante Reconocimiento de Voz:
 - Permite a los clientes realizar ventas utilizando comandos de voz, como "Agregar un teléfono al carrito" o "Calcular el total".
 - Esta funcionalidad mejora la accesibilidad y reduce el tiempo de atención en tiendas concurridas.
3. Interfaz Web y Móvil :
 - Aplicación Web : Desarrollada con Angular/React para proporcionar una interfaz intuitiva y responsive para navegadores.
 - Aplicación Móvil : Desarrollada con Flutter para garantizar compatibilidad multiplataforma (iOS y Android).

4. Backend y Base de Datos :

- El backend se desarrollará en Python utilizando un framework como Flask o Django para exponer una API RESTful .
- La base de datos será PostgreSQL , que almacenará información sobre productos, transacciones, clientes y otros

Funcionalidades Principales del Sistema

1. Gestión de Ventas:

- Registrar productos en el carrito mediante teclado, mouse o reconocimiento de voz.
- Calcular totales, aplicar descuentos, realizar pagos en línea y generar recibo de venta.
- Implícitamente mientras se realiza la venta se debe aplicar técnicas de Recomendaciones Inteligentes:
 - Analizar transacciones históricas para identificar patrones frecuentes de compra utilizando **reglas de asociación** (por ejemplo, Apriori).
 - Mostrar productos sugeridos en tiempo real en la interfaz del POS.
- Además de ofertar una clásica interface mediante teclado y mouse también se debe poder ofertar la opción de realizar la venta mediante Reconocimiento de Voz:
 - Capturar comandos de voz, convertirlos en texto y procesarlos para ejecutar acciones específicas (agregar productos al carrito, calcular totales, etc.).
 - Proporcionar retroalimentación verbal o visual para confirmar las acciones realizadas.

2. Gestión de Productos e Inventario :

- Agregar, editar y eliminar productos en el catálogo.
- Monitorear niveles de inventario y recibir alertas cuando un producto esté bajo en stock. (Ejemplo enviar notificación tipo push al responsable del almacén notificando que un producto tiene bajo stock)

3. Reportes :

- Generación de informes. (Reporte por cliente en un determinado periodo, Reporte de productos mas vendido en un determinado periodos, etc,... y todos los reportes debe visualizar en pantalla, exportar pdf y Excel).

Tecnologías y Herramientas

1. Proceso de desarrollo de software basado en PUDS, Modelos basado en UML 2.5+ Usando herramientas CASE.
2. Backend :
 - Lenguaje: Python .
 - Framework: Flask o Django .
 - API: RESTful para comunicación entre el backend y los frontends.
 - Machine Learning: Scikit-learn o Mlxtend para implementar reglas de asociación.(Ver otras alternativas dentro de ML)
 - Reconocimiento de Voz: Google Cloud Speech-to-Text o Vosk para procesar comandos de voz. (O ver otras alternativas)
3. Frontend Web :
 - Framework: React/Angular .
4. Frontend Móvil :
 - Framework: Flutter .
5. Base de Datos :
 - Motor: PostgreSQL .
6. Herramientas Adicionales :
 - Control de versiones: Git/GitHub .
 - Servicios en la nube AWS , Google Cloud o Azure para alojar la aplicación o similar.

Entregables Esperados

1. Publicar código en GitHub y entregar URL + Código QR para acceder a su repositorio
2. Documentación:
 - Documentación técnica del sistema
3. Demostración y defensa:
 - Una demostración funcional del sistema, mostrando todas las características principales.

DOCUMENTO PRIMER PARCIAL

1) Perfil

1.1 Introducción

1.2 Objetivo General

1.3 Objetivos Específicos

1.4 Descripción del problema

1.4 Alcance

la documentación estará organizada en dos partes

Parte I – Fundamentación Teórica (revisar en libros o sitios en internet especializados en la temática)

- a) E-commerce: además de profundizar los conceptos generales y características de los e-commerce, experimentar de acuerdo a lo siguiente:
 - a. Como usuario: Conocer detalladamente como funciona:
 - i. Amazon
 - ii. Alibaba
 - iii. Shopify
 - b. Como desarrollador: Conocer para que sirve, que beneficios y como usar para crear sitios web para tiendas online, los siguientes open source para e-commerce:
 - i. Magento
 - ii. PrestaShop
 - iii. WooCommerce
- b) Pasarelas de pago
 - a. Describir cómo funcionan las distintas formas de pago online (tarjetas de débito, tarjetas de crédito, QR, Transferencias)
 - b. Indagar sobre **LIBÉLULA** una pasarela que se usa con frecuencia en nuestro medio
 - c. Indagar sobre **STRIPE** una opción de pasarela de pago internacional
- c) Deliverys
 - a. Describir cómo funcionan los deliverys (un ejemplo: Pedidos Ya)
 - b. Como calculan los pagos para una entrega (distancia, peso, frecuencia, tamaño de la entrega etc.)
- d) PUDS
- e) UML

Parte II – Proceso de desarrollo

- a) Seguir los pasos descritos por el PUDS para el desarrollo de la Aplicación
- b) Generar los modelos pertinentes utilizando UML

2.1) Flujo de Trabajo: Captura de Requisitos

2.2) Flujo de Trabajo: Análisis

2.3) Flujo de Trabajo: Diseño

2.4) Flujo de Trabajo: Implementación

2.5) Flujo de Trabajo: Pruebas

CONCLUSION

RECOMENDACIÓN

BIBLIOGRAFIA

TIEMPO PRIMER PARCIAL: JUEVES 27/03/2025 - MARTES 22/04/2025
(PARA SER HABILITADO A LA DEFENSA DEL EXAMEN ES OBLIGATORIO
ENVIAR LAS 3 PRESENTACIONES)

CRONOGRAMA PARA ENVIAR DOCUMENTO A LA PLATAFORMA

PRESENTACION # 1 DOMINGO 06/04/2025 PARTE I Y AVANCE PARTE 2

PRESENTACION # 2 DOMINGO 13/04/2025 TODOS FT AVANCE

PRESENTACION FINAL DOMINGO 20/04/2025 SOFTWARE COMPLETO

DEFENSA FINAL DE APP WEB/MOVIL: MARTES 22/04/2024