Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2024

IMPLEMENTACIÓN DE UNA API PARA GESTIÓN DE TRANSACCIONES Y SERVICIOS DE BANCA ONLINE EN INDUSTRIA TÍPICA GUATEMALTECA, S.A. (ITGSA)

202100106 – Keneth Willard Lopez Ovalle

Resumen

Este ensayo explora la implementación de una solución integral basada en una API para servicios web, utilizando el protocolo HTTP y el paradigma de programación orientada a objetos (POO), en el contexto de una tienda virtual para Industria Típica Guatemalteca, S.A. (ITGSA). La relevancia del tema radica en la adaptación tecnológica que enfrentan las empresas en Guatemala y globalmente, adaptándose a las nuevas exigencias del mercado digital y la banca en línea. Se examinan las técnicas y tecnologías aplicadas, como el uso de bases de datos y XML para la gestión de datos, y las expresiones regulares para el procesamiento de la información. El impacto del proyecto es multifacético, afectando áreas técnicas, económicas y sociales, facilitando a los clientes un acceso más eficiente a los productos y servicios, mientras se mejora la gestión de transacciones financieras. Las conclusiones destacan la efectividad de la solución desarrollada en mejorar la interfaz de usuario y la eficiencia operativa, estableciendo un modelo para futuras implementaciones en el sector.

Palabras clave

API, programación orientada a objetos, banca en línea, gestión de datos, tienda virtual

Abstract

This essay examines the implementation of an integrated solution based on an API for web services, utilizing the HTTP protocol and object-oriented programming (OOP) paradigm, within the context of an online store for Industria Típica Guatemalteca, S.A. (ITGSA). The relevance of the topic lies in the technological adaptation faced by companies in Guatemala and globally, adjusting to the new demands of the digital market and online banking. The techniques and technologies applied, such as the use of databases and XML for data management, and regular expressions for information processing, are explored. The project's impact is multifaceted, affecting technical, economic, and social areas, facilitating more efficient access for customers to products and services while improving financial transaction management. The conclusions highlight the effectiveness of the developed solution in enhancing user interface and operational efficiency, establishing a model for future implementations in the industry.

Keywords

API, object-oriented programming, online banking, data management, online store

Introducción

En la era digital actual, las empresas buscan continuamente formas de integrar tecnologías avanzadas para mejorar sus operaciones y la experiencia del cliente. Industria Típica Guatemalteca, S.A. (ITGSA), una prominente empresa local, no es la excepción, abordando este desafío a través de la creación de una tienda virtual y servicios de banca en línea. Este ensayo explora la implementación de una API que utiliza el protocolo HTTP y la programación orientada a objetos para gestionar transacciones en una plataforma de comercio electrónico. La relevancia de este proyecto radica en su potencial para transformar el acceso a productos y servicios, optimizando la eficiencia y satisfacción del cliente. ¿Cómo pueden las tecnologías d e API mejorar las operaciones comerciales y la interacción con el cliente en un entorno digital? Este ensayo busca responder esta pregunta, proporcionando un análisis detallado de las soluciones tecnológicas adoptadas y su impacto en el comercio electrónico.

Desarrollo del tema

a) Tecnologías y Herramientas Utilizadas

El proyecto de ITGSA implementa una API REST utilizando Python con el framework Flask, debido a su capacidad para manejar solicitudes HTTP de manera eficiente y su extensa biblioteca de módulos para el desarrollo web. Se eligió Django para el frontend por su robustez y su patrón de diseño MVT (Modelo-Vista-Template), que facilita la gestión de la interfaz de usuario y la interacción con el backend.

b) Diseño e Implementación de la API

- Diseño e Implementación de la API

El diseño de la API en views.py para el backend de ITGSA ilustra cómo Flask y Python pueden utilizarse para manejar las interacciones de datos de forma eficiente y segura. Se detallan aquí las operaciones clave definidas en el módulo.

- Configuración y Carga de Datos:

La ruta /grabarConfiguracion en la API recibe configuraciones en formato XML, las procesa y actualiza la base de datos. Utiliza métodos POST para recibir los datos, que luego son manejados por la clase GestorTransacciones para crear o actualizar registros de clientes y bancos. La respuesta se genera en formato XML, indicando cuántos registros fueron creados o actualizados, lo que facilita la trazabilidad y la verificación por parte de los usuarios.

- Manejo de Transacciones:

Similar a la configuración, la ruta /grabarTransaccion maneja las transacciones financieras de los clientes. La API acepta detalles de facturas y pagos, verificando cada transacción para evitar duplicados y errores. Los resultados se devuelven también en XML, proporcionando detalles claros sobre nuevas facturas, duplicados y errores, lo cual es crucial para el mantenimiento de registros precisos y la transparencia operacional.

- Limpieza de Datos:

La función limpiar_datos() es un excelente ejemplo de cómo las APIs pueden facilitar la gestión de datos. A través de un simple comando POST, todos los datos pueden ser reseteados a su estado inicial, una funcionalidad crítica durante las fases de prueba o ante la necesidad de reiniciar el sistema.

- Consulta de Estado de Cuenta:

Esta funcionalidad demuestra la capacidad de la API para proporcionar información financiera detallada a solicitud del usuario. Al ingresar un NIT específico, la API devuelve un estado detallado de cuenta en XML, que incluye todas las transacciones relacionadas, permitiendo a los usuarios un seguimiento eficiente de su actividad financiera.

- Consultas de Ingresos:

Finalmente, consultar_ingresos() ilustra cómo la API puede utilizarse para generar informes financieros basados en datos recopilados, un aspecto vital para la toma de decisiones en cualquier empresa. Este

método valida y procesa solicitudes de ingresos por fecha y genera una respuesta JSON que resume los ingresos por banco, ofreciendo una interfaz amigable y fácilmente integrable con otros sistemas o interfaces de usuario.

c) Interacción entre el Backend y el Frontend

Este segmento detalla la interacción entre el frontend desarrollado en Django y el backend manejado por Flask, destacando cómo las tecnologías trabajan conjuntamente para procesar y gestionar datos de la tienda virtual y la banca online de ITGSA.

Integración y Comunicación:
La comunicación entre Django y Flask se realiza mediante solicitudes HTTP, utilizando la biblioteca requests de Python. Esto permite al frontend enviar y recibir datos al backend de manera eficiente. Las URL para las API de Flask están definidas en la variable FLASK_API_URL, centralizando la configuración de conexión y facilitando cambios en el futuro sin alterar el código en múltiples lugares.

- Funciones del Frontend:

- Grabar Configuración:

La función grabarConfiguracion maneja la carga de archivos XML de configuración. Una vez que el archivo es cargado a través del formulario, se lee y se envía como una solicitud POST al endpoint /grabarConfiguracion del backend. El backend procesa esta información, actualiza la base de datos y devuelve un XML con el resultado, que incluye el número de clientes y bancos creados o actualizados. El frontend recibe esta respuesta, la formatea para mejorar la legibilidad y la presenta al usuario.

- Grabar Transacción:

Similarmente, grabarTransaccion gestiona la carga de transacciones de facturación y pagos. Tras recibir el archivo XML, el frontend lo envía al backend y procesa la respuesta, que informa sobre nuevas transacciones registradas o errores encontrados.
Limpiar Datos:
Esta función envía una solicitud para limpiar todos los datos almacenados, útil durante las pruebas o reseteos del sistema. La respuesta del backend confirma la acción ejecutada.

- Devolver Estado de Cuenta:

Permite a los usuarios consultar el estado de cuenta de un cliente específico, mostrando detalles como saldo y transacciones. El frontend envía el NIT del cliente como parámetro y presenta la información devuelta en un formato legible para el usuario.
Consultar Ingresos:
Ofrece un resumen de los ingresos generados, filtrados por fecha. Los datos recibidos en formato JSON son convertidos y presentados en el frontend, permitiendo visualizaciones claras y detalladas de los ingresos.

- Manejo de Errores y Validaciones:

Cada función incluye manejo de errores para asegurar que los usuarios reciban feedback adecuado en caso de problemas, como fallos de conexión con el backend o datos inválidos. Esto asegura una experiencia de usuario fluida y reduce la posibilidad de errores durante la operación.

d) Impacto y Evaluación del Proyecto

La implementación de la API ha permitido a ITGSA mejorar significativamente la eficiencia de sus operaciones comerciales. Los clientes ahora pueden acceder a servicios y realizar transacciones de manera más rápida y segura. Los desafíos incluyeron asegurar la integridad de los datos en transacciones concurrentes y optimizar el tiempo de respuesta de la API. Estos se abordaron mediante el uso de técnicas de programación avanzadas y la optimización del código y las consultas a la base de datos.

Conclusiones

El desarrollo de la API y la integración entre el backend de Flask y el frontend de Django para ITGSA han demostrado ser un avance significativo

en la forma en que la empresa gestiona sus transacciones comerciales y la interacción con sus clientes. Esta solución tecnológica no solo ha optimizado los procesos internos, sino que también ha mejorado la experiencia del cliente al proporcionar una plataforma más accesible y eficiente.

Aportes Principales:

Automatización y Eficiencia: La automatización de las transacciones y la gestión de datos mediante la API han reducido significativamente los errores humanos y han incrementado la eficiencia operacional.
Accesibilidad y Transparencia: La implementación del sistema ha permitido a los clientes acceder a su información financiera de manera más transparente y en tiempo real, lo cual ha mejorado la confianza y la satisfacción del cliente.
Flexibilidad y Escalabilidad: La arquitectura utilizada proporciona la flexibilidad necesaria para futuras expansiones y adaptaciones sin grandes alteraciones del sistema existente, asegurando la sostenibilidad del proyecto a largo plazo.
Recomendaciones para Futuras Investigaciones:

Seguridad de Datos: A medida que el sistema se expande y maneja más datos sensibles, sería prudente realizar una evaluación más profunda de las medidas de seguridad implementadas y explorar nuevas tecnologías de seguridad para proteger mejor la información del cliente.
Análisis de Big Data: Implementar herramientas de análisis de Big Data podría proporcionar insights más profundos sobre el comportamiento del cliente y las tendencias de mercado, lo cual podría traducirse en mejoras en la toma de decisiones y estrategias comerciales.
Integración de IA: Explorar la integración de capacidades de inteligencia artificial para automatizar aún más la atención al cliente y las respuestas a las transacciones, personalizando la experiencia del usuario y mejorando la eficiencia operativa.

Reflexión y Debate:
¿Cuáles son los desafíos éticos y prácticos asociados con la recopilación y el manejo de grandes volúmenes de datos financieros de los clientes?
¿Cómo pueden las empresas garantizar que la digitalización de servicios no excluya a segmentos de la población que quizás no tengan fácil acceso a la tecnología?
En conclusión, este proyecto no solo ha demostrado el valor de la tecnología en la transformación digital de las empresas tradicionales, sino que también ha puesto de manifiesto la necesidad de abordar continuamente cuestiones de seguridad, privacidad y equidad a medida que avanzamos hacia un futuro más digitalizado.

Referencias bibliográficas

Django documentation | Django documentation. (n.d.). Django Project. https://docs.djangoproject.com/en/5.0/

Guzman, H. C. (n.d.). Uniendo el Frontend con el Backend | Curso de Django | Hektor Profe. https://docs.hektorprofe.net/django/web-personal/uniendo-frontend-backend/

Welcome to Flask — Flask Documentation (3.0.X). (n.d.). https://flask.palletsprojects.com/en/3.0.x/

xml.dom.minidom — Minimal DOM implementation. (n.d.). Python Documentation.

https://docs.python.org/3/library/xml.dom.minidom.html

Anexos

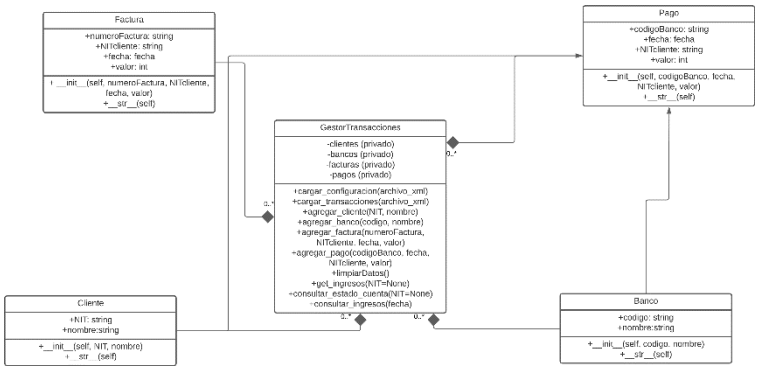

Figura 1: Diagrama de clases del proyecto.
Fuente: Elaboracion propia.

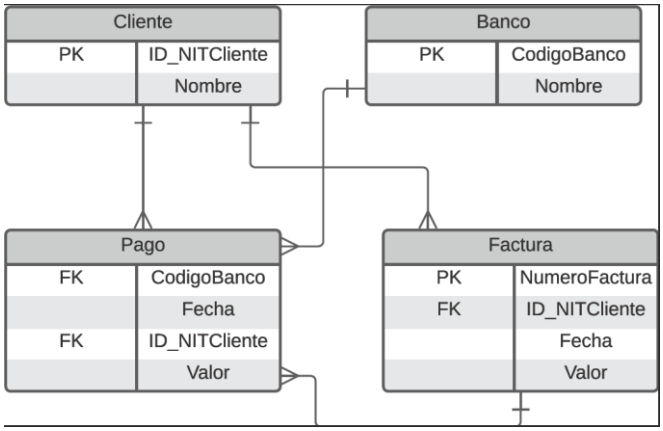

Figura 1: Diagrama de clases del proyecto.
Fuente: Elaboracion propia.