



UNIVERSIDAD DE MARGARITA

SUBSISTEMA DE DOCENCIA

DECANATO DE INGENIERÍA Y AFINES

COORDINACIÓN DE INVESTIGACIÓN Y PASANTÍA

**IMPLEMENTACION DE UN SERVICIO API PARA LA SINCRONIZACIÓN ENTRE
EL SISTEMA DE POS Y EL ERP DE GUUAO MARKETPLACE**

Autor: Samuel Salazar

Tutor: Ing. Valentina Martínez Hernández

El Valle del Espíritu Santo, julio de 2023

INDICE GENERAL

LISTA DE CUADROS	i
LISTA DE FIGURAS	ii
RESUMEN	iii
INTRODUCCION	iv
PARTES I	1
DESCRIPCIÓN GENERAL DEL PROBLEMA	1
1.1 Formulación del problema	1
1.2 Interrogantes.....	5
1.3 Objetivo general	5
1.4 Objetivos Específicos	5
1.5 Valor Académico de la Investigación	6
PARTES II.....	7
DESCRIPCIÓN TEÓRICA.....	7
2.1 Antecedentes	7
2.2 Bases teóricas	9
2.2.1 Sistemas de Punto de Venta (POS).....	9
2.2.2 Sistemas ERP (Enterprise Resource Planning)	12
2.2.3 Integración de sistemas.....	14
2.2.4 Servicios API (Application Programming Interface)	19
2.5 Bases legales	22
2.5.1 Constitución De La República Bolivariana De Venezuela (publicada en Gaceta Oficial Extraordinaria N° 36.860, de fecha 30 de diciembre de 1999)	22
2.5.2 Ley Orgánica De Ciencia, Tecnología E Innovación (publicada en Gaceta Oficial Extraordinaria N° 37.291, de fecha 26 de septiembre de 2001).....	23
2.5.3 Ley Contra los Delitos Informáticos (publicada en Gaceta Oficial N° 37.313, de fecha 30 de octubre de 2001)	23
2.5.4 Ley de Telecomunicaciones (publicada en Gaceta Oficial N° 36.970, de fecha 12 de junio de 2000).....	24
2.5.5 Ley de Infogobierno (publicada en Gaceta Oficial N° 40.274, de fecha 17 de octubre de 2013)	24
2.6 Definición de términos	25
PARTES III	28
DESCRIPCIÓN METODOLOGICA	28
3.1 Naturaleza de la investigación.....	28

3.2	Tipo de investigación	28
3.3	Diseño de la investigación.....	29
3.4	Población y muestra	29
3.5	Técnicas de recolección de datos	30
3.6	Técnicas de análisis de datos.....	31
PARTE IV	33
4.1	Identificación de las funcionalidades clave del sistema de POS y del ERP de la empresa GUUAO Marketplace.	33
4.2	Desarrollo de la estructura de una solución de integración óptima para la sincronización entre el sistema de POS y el ERP de la empresa GUUAO Marketplace, considerando las opciones disponibles y aplicando el enfoque seleccionado para la implementación del sistema.	
	42	
4.3	Determinación del rendimiento del servicio API en términos de mejora de la eficiencia de la empresa GUUAO Marketplace y de la capacidad de adaptación a las cambiantes demandas del mercado.	59
PARTE V	62
5.1	Importancia de la aplicación de la propuesta.	62
5.2	Viabilidad de la propuesta.....	62
5.2.1	Viabilidad técnica.	62
5.2.2	Viabilidad operativa.	63
5.2.3	Viabilidad económica.	63
5.3	Objetivos de la propuesta.	64
5.3.1	Objetivos General.	64
5.3.2	Objetivos Específicos.	64
5.4	Representación gráfica y estructurada de la propuesta	66
5.3.1	Funcionalidades y mensajes por consola de la API.....	66
5.4.2	Rutas GET de la API	76
5.4.3	Rutas POST API.....	112
5.4.4	Implementación remota de la API.....	114
REFERENCIAS BIBLIOGRAFÍCAS.....		121
ANEXOS.....		127

LISTA DE CUADROS

Cuadro 1: Entrevista sobre el sistema POS.....	36
Cuadro 2 Entrevista sobre el ERP.....	40
Cuadro 3 Funcionalidades claves del sistema POS y el ERP	42
Cuadro 4 Comparación entre diferentes lenguajes de programación usados en el desarrollo de APIs.....	43
Cuadro 5 Comparación entre las diferentes arquitecturas de APIs.....	45
Cuadro 6 Comparación entre librerías para la conexión con la Base de Datos.	48
Cuadro 7 Resumen de la información recopilada	52
Cuadro 8 Cuadro comparativo WebSockets y tareas Cron.....	58
Cuadro 9 Matriz FODA	61
Cuadro 10 Especificaciones técnicas del servidor remoto.....	63
Cuadro 11 Especificaciones operativas	63
Cuadro 12 Costos de la inversión única del servidor.....	64
Cuadro 13 Requisitos económicos.....	64

LISTA DE FIGURAS

Figura 1 Estructura de los archivos de una API REST desarrollada en TypeScript.....	46
Figura 2 Diagrama UML sobre el MVC.....	47
Figura 3 Ejemplo de conexión a una base de datos y consulta usando MSSQL.	49
Figura 4 Ejemplo de cómo hacer uso de la librería DotEnv	50
Figura 5 Estructura de carpetas de la API.....	52
Figura 6 Diagrama de clases sincronizadores.....	53
Figura 7 Estructura de los sincronizadores.	53
Figura 8 Diagrama de clases controladores	54
Figura 9 Estructura archivos de las rutas.	54
Figura 10 Diagrama JSON de las facturas	55
Figura 11 Diagrama JSON de las notas de crédito	55
Figura 12 Diagrama JSON de los pagos	55
Figura 13 Estructura JSON de los productos	56
Figura 14 Estructura JSON de las existencias	56
Figura 15 Estructura JSON de los precios	56
Figura 16 Estructura JSON de los códigos	57
Figura 17 Estructura JSON de las categorías.....	57
Figura 18 Diagrama de la sincronización entre los sistemas	59
Figura 19 Diagrama de flujo inicio de la API.....	66
Figura 20 Mensajes por consola del inicio del programa	67
Figura 21 Diagrama de flujo sincronización de la BDD del POS	68
Figura 22 Mensajes por consola inicio sincronización BDD.....	68
Figura 23 Mensajes por sincronización de categorías	69
Figura 24 Mensajes por sincronización de productos.....	69
Figura 25 Mensajes por sincronización de códigos de barra	69
Figura 26 Mensajes por sincronización de inventario en todos los almacenes.....	70
Figura 27 Diagrama de flujo sincronización documentos para B2B	71
Figura 28 Mensajes inicio de sincronización B2B	72
Figura 29 Mensajes sincronización de documentos: SIN ENVIOS	72
Figura 30 Mensajes sincronización de documentos: CON ENVIOS	72

Figura 31 Mensajes sincronización de pagos: SOLO REGISTROS	72
Figura 32 Diagrama de flujo sincronización de precios y tasa de cambio en BDD	73
Figura 33 Mensajes sincronización de precio y tasa: HORARIO NO VALIDO	74
Figura 34 Mensajes sincronización de tasa de cambio	74
Figura 35 Mensajes sincronización de precios con USD y sin USD	74
Figura 36 Ejemplo registro de logs en archivos de la API	75
Figura 37 Figuras de los json en crudo y en formato tabla de las rutas	76
Figura 38 Datos JSON en formato de tabla	77
Figura 61 Describiendo el inicio automático del servicio	115
Figura 62 Desencadenador al iniciar el servidor.....	116
Figura 63 Acción al iniciar el servidor	117
Figura 64 Ejemplo real del uso de la API, Sambil Margarita	127
Figura 65 Ejemplo de uso de la API, página local de GUUAO MP	128
Figura 66 Captura de documentación de las funciones de la API	128
Figura 67 Captura de documentación de las variables.....	129
Figura 68 Ejemplo de explicación de una función en la documentacion	129
Figura 69 readme.md de configuraciones y credenciales de la API	130

UNIVERSIDAD DE MARGARITA
SUBSISTEMA DE DOCENCIA
COORDINACIÓN DE INVESTIGACIÓN Y PASANTÍA

**IMPLEMENTACION DE UN SERVICIO API PARA LA SINCRONIZACIÓN ENTRE
EL SISTEMA DE POS Y EL ERP DE GUUAO MARKETPLACE**

Autor: Samuel Alexander Salazar Salazar

Tutor: Ing. Valentina Martínez Hernández

Julio de 2023

RESUMEN

Actualmente, muchas organizaciones se ven enfrentadas al desafío de integrar y gestionar eficazmente sus datos entre diferentes sistemas. Una vez identificada esta problemática en la empresa GUUAO Marketplace, se propuso la creación de una API capaz de integrar una base de datos del punto de venta con su sistema ERP, que además incorpora una estructura de proyecto robusta. Este avance se desarrolló bajo una metodología cuantitativa, combinando el diseño de campo, orientado como un proyecto factible. Esta iniciativa identificó las funcionalidades y los datos manejados por dichos sistemas, para aumentar la eficiencia de sus procesos comerciales, implementando una estructura desarrollada de manera modular que posee la ventaja de adaptarse a los posibles cambios del mercado.

Descriptores: desarrollo de API, sistema ERP, sistema POS, sincronización de datos, integración de sistemas, implementación de API, base de datos.

INTRODUCCION

En la actualidad, la digitalización de procesos se ha vuelto una necesidad y una tendencia creciente en diversas organizaciones. Sin embargo, estas enfrentan desafíos al tener que lidiar con la gestión eficiente de sus datos, especialmente cuando provienen de múltiples fuentes. El hecho de no contar con una solución efectiva puede resultar en errores, ineficiencias y decisiones de negocios no óptimas.

Por un lado, existen sistemas de punto de venta que recolectan información vital de las transacciones diarias y, por el otro, se encuentran los sistemas ERP que manejan una amplia gama de operaciones empresariales. Una integración efectiva de ambos puede facilitar la gestión de información, mejorar la eficiencia y la toma de decisiones. Sin embargo, también surge la necesidad de garantizar la seguridad de los datos durante este proceso de integración.

Teniendo en cuenta estas necesidades en la empresa GUUAO Marketplace, la presente propuesta se enfoca en la implementación de una API para la sincronización de su sistema POS con su sistema ERP, garantizando una transmisión de datos segura y eficiente. Para lograr tal fin, se lleva a cabo el presente trabajo con la siguiente estructura:

En la **Parte I**, se presenta la descripción general del problema, incluyendo la formulación del problema, el objetivo general y específicos, y el valor académico de la investigación.

En la **Parte II**, se presenta la descripción teórica, incluyendo los antecedentes, las bases teóricas y legales, y la definición de términos claves.

En la **Parte III**, se proporciona la descripción metodológica de la investigación, incluyendo aspectos técnicos y procedimientos utilizados.

En la **Parte IV**, se incluye el análisis y presentación de los resultados obtenidos.

En la **Parte V**, se presenta la propuesta de la implementación de la API, su importancia, viabilidad técnica, operativa y económica, los objetivos, su estructura y representación gráfica; para finalizar con las conclusiones y recomendaciones derivadas de la investigación.

Por último, se presentan las referencias bibliográficas y anexos utilizados para la investigación

PARTE I

DESCRIPCIÓN GENERAL DEL PROBLEMA

El planteamiento del problema, según Hernández, R. (2014), “establece la naturaleza del problema, su ubicación en el contexto y los objetivos que persigue”, es una etapa crucial de la investigación. Es crucial tener en cuenta los componentes de la estrategia que respaldan la 'relevancia' del problema. Al hacer esto, es posible garantizar que la investigación no se desvíe del problema real sino de su enfoque. Además, al delinear claramente el problema, se pueden identificar problemas importantes, lo que ayuda a lograr con éxito los objetivos de la investigación.

1.1 Formulación del problema

El sistema POS (Punto de Venta) es un software utilizado para manejar transacciones. Según los autores Lee S. y Han I. (2018), este "se ha convertido en una herramienta crucial para los minoristas y las empresas que buscan mejorar la eficiencia y la precisión de sus procesos de ventas" (p. 1). Con el uso de este sistema, los minoristas pueden registrar y manejar transacciones rápidamente, monitorear el inventario y proporcionar datos de ventas. Una terminal de punto de venta, una impresora de recibos y un software para el procesamiento de pagos, conforman un sistema POS típico. El hardware utilizado para escanear códigos de barras, ingresar datos de precios y manejar pagos se conoce como terminal de punto de venta. La impresora de recibos se utiliza para imprimir recibos y facturas de ventas. El software de procesamiento de pagos procesa las transacciones y registra los datos de venta. En términos prácticos, la implementación de un sistema POS puede resultar costosa y compleja, especialmente para las pequeñas empresas. Sin embargo, la adopción de esta herramienta puede ser beneficiosa para la gestión de inventario y el seguimiento de las ventas. Además, los sistemas POS modernos a menudo incluyen características adicionales, como análisis de datos y seguimiento de clientes.

Asimismo, los ERP (Enterprise Resource Planning), según los autores Kumar K. y Hillegersberg J. (2000) "Son sistemas de información de gestión que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa". De igual manera, Kumar K. y Hillegersberg J. plantean que “Proporcionan una visión integrada de los procesos de negocio y operaciones de la empresa” (p. 94). Esto incluye controlar el presupuesto, el inventario, la producción y las ventas. Las actividades comerciales de una

organización son administradas y centralizadas por un sistema ERP, que consta de una colección de programas de software. Su objetivo principal es ofrecer una imagen integrada de las operaciones corporativas y los procesos de la empresa, lo que puede impulsar la productividad y la toma de decisiones. Los módulos que componen los sistemas ERP incluyen los de finanzas, recursos humanos, fabricación, inventario y ventas.

Por otra parte, según Vargo, S. y Lusch, R. (2008), un servicio es "un proceso de co-creación de valor en el cual los clientes reciben beneficios a través de la aplicación de recursos operativos y tecnológicos de la empresa". Teniendo en cuenta lo planteado por los autores, un servicio se puede considerar como una acción que realiza una empresa para satisfacer una necesidad. La implementación de estos servicios se puede llevar a cabo en distintos sectores o ámbitos. Por ejemplo, los supermercados pueden brindar asistencia al cliente, gestión de compras en línea, entrega a domicilio y programas de fidelización para consumidores habituales.

Asimismo, este tipo de establecimientos necesitan varios servicios para funcionar bien, además de proporcionar beneficios a los clientes. Por ejemplo, pueden utilizar la logística y el transporte para recibir y distribuir los productos, así como servicios de limpieza y mantenimiento para mantener las instalaciones en buen estado, de publicidad y marketing para promocionar los productos y atraer a más clientes. También pueden requerir servicios de tecnología de la información para gestionar el inventario y las transacciones, así como servicios de asesoría legal y financiera para garantizar el cumplimiento de las regulaciones y optimizar los resultados económicos. En general, estas actividades se caracterizan por ser intangibles, heterogéneas, perecederas, producidas y consumidas simultáneamente. Además, suelen ser altamente personalizadas y están fuertemente ligadas a la experiencia del cliente. Para que estas actividades sean efectivas, se requiere de una estrecha colaboración entre el proveedor y el cliente, y una comprensión clara de las necesidades del cliente.

Considerando lo anterior, los servicios en el ámbito de la tecnología pueden ser provistos a través de una Interfaz de Programación de Aplicaciones (API), la cual, según Fielding, R. (2000), es "un conjunto de reglas que describen cómo el software de Aplicación puede comunicarse entre sí". Una API define los formatos de mensaje, los protocolos de comunicación y los estándares de intercambio de datos que se pueden usar entre aplicaciones, de esta manera, facilita la comunicación e interacción efectiva y eficiente entre numerosos programas. Se podría acceder a

un sistema de pago en línea a través de una aplicación móvil para compras en línea, por ejemplo, lo que permitiría a los usuarios realizar transacciones de forma rápida y segura. Los desarrolladores pueden crear un nuevo software que interactúe con el sistema existente e incluya nuevas funciones al proporcionar una API clara y abierta.

Por lo general, las API hacen posible que los programas se comuniquen e interactúen de manera efectiva, a través del intercambio de información entre sistemas, lo que puede aumentar significativamente la efectividad y la utilidad de los sistemas actuales.

El problema de la desconexión entre el sistema de punto de venta y el ERP es bastante importante en el sector comercial, a raíz de la posible ocurrencia de problemas que se pueden generar, como la duplicación de tareas, lo que aumenta la carga de trabajo de los empleados. Además, la demora en el procesamiento de la información puede ser un problema, especialmente cuando se trata de decisiones comerciales urgentes. Esto puede resultar en un mal juicio y la pérdida de posibilidades. El problema se enfoca en la dificultad de sincronización de los datos de ventas generados en el sistema POS y el sistema ERP de la empresa. La ausencia de un sistema adecuado puede ser un obstáculo importante para la integración eficiente, sin una API que para poder manejar la información la integración de la misma puede ser tediosa o incluso inexistente.

Existen distintos antecedentes en empresas a nivel mundial donde se ha demostrado la importancia de implementar APIs en el mundo empresarial. El uso de APIs como herramienta que permite la innovación y la expansión es evidente a partir de estos dos ejemplos: la utilización de la API de Google Maps por parte de Uber, que permite a los conductores privados buscar opciones de transporte confiables a través de su aplicación; esto ilustra cómo las API brindan facilidad de uso y, al mismo tiempo, aumentan significativamente las medidas de seguridad. Del mismo modo, gracias a su plataforma API abierta, Salesforce continúa dominando las soluciones de CRM al permitir que los desarrolladores de todo el mundo accedan a diferentes plataformas comerciales de manera eficiente; la importancia demostrada por estas instancias enfatiza por qué las empresas deben enfocarse en el uso de API para promover el crecimiento. En el clima empresarial actual, se ha demostrado que la capacidad de una empresa para brindar soluciones innovadoras mediante la fusión de múltiples servicios e integrar las soluciones actuales en los sistemas comerciales existentes es esencial para el éxito corporativo.

En el Estado Nueva Esparta conviven diversas empresas del rubro distribución de alimentos, entre ellas Grupo Leiros, que a través de su empresa GUUAO Marketplace ubicada en el Centro Comercial El Sambil, se dedica a la venta de productos de consumo diario. Considerando que el autor de la presente investigación trabaja en la empresa, se ha podido detectar que las tecnologías utilizadas en el supermercado no están integradas. Se pudo identificar que el sistema de punto de venta (POS) utilizado en el supermercado no está integrado con el sistema ERP de la empresa. Dando a lugar la falta de comunicación entre los sistemas y, por lo tanto, una falta de sincronización entre la información por los mismos. La falta de integración entre ERP y POS en GUUAO Marketplace ha resultado en una desconexión significativa en la transmisión de datos importantes entre ambos sistemas. Por ejemplo, el POS recopila información valiosa sobre las ventas diarias, mientras que el ERP maneja información relacionada con finanzas, logística y gestión de la cadena de suministro. Si estos sistemas estuvieran integrados, la información que maneja el ERP se transmitiría automáticamente al POS. Además, los datos del POS, incluidas las compras y las facturas, pueden enviarse inmediatamente al ERP, mejorando la gestión de ventas y la satisfacción del cliente. En general, la eficiencia del negocio y la satisfacción del cliente pueden verse afectadas por una falta de conexión entre el ERP y el POS.

Por lo tanto, se requiere el desarrollo de una API que permita una sincronización de datos efectiva y eficiente entre los sistemas POS y el ERP de la empresa GUUAO. La implementación de un servicio de API adecuado ayudará a resolver los problemas de integración entre los sistemas ya mejorar la eficiencia y productividad de la empresa. Esto implicará la identificación de los datos relevantes a sincronizar, la definición de un protocolo de comunicación, el diseño de una arquitectura de software adecuada y la implementación y prueba del servicio API.

1.2 Interrogantes

En correspondencia con lo expuesto en el apartado anterior, surge la siguiente interrogante: ¿Cómo será el desarrollo de una API para la sincronización entre el sistema POS y el ERP de la empresa GUUAO Marketplace? Así mismo, partiendo de dicha interrogante, se desglosan las siguientes:

- ¿Cuáles son las funcionalidades clave del sistema de POS y del ERP de GUUAO que deben ser consideradas en la integración de ambos sistemas?
- ¿Cuál es la metodología más efectiva para desarrollar e implementar una integración para la sincronización entre el sistema POS y el ERP?
- ¿Cuál es el rendimiento del servicio API en términos de mejora de la eficiencia de la empresa y de la capacidad de adaptación a las cambiantes demandas del mercado?

1.3 Objetivo general

Implementar un servicio API para la sincronización entre el sistema de POS y el ERP de la empresa GUUAO Marketplace.

1.4 Objetivos Específicos

- Identificar las funcionalidades clave del sistema de POS y del ERP de la empresa GUUAO Marketplace.
- Desarrollar la estructura de una solución de integración para la sincronización entre el sistema de POS y el ERP de la empresa GUUAO Marketplace, considerando las opciones disponibles y aplicando el enfoque seleccionado para la implementación del sistema.
- Determinar el rendimiento del servicio API en términos de mejora de la eficiencia de la empresa GUUAO Marketplace y de la capacidad de adaptación a las cambiantes demandas del mercado.

1.5 Valor Académico de la Investigación

Esta investigación aporta una solución práctica a un problema común en el campo de la integración de sistemas de información empresarial. Al proporcionar una solución específica a este problema, esta investigación no sólo mejorará la eficiencia y la competitividad de la empresa GUUAO Marketplace, sino que también tiene el potencial de tener un impacto positivo en su rentabilidad y capacidad de adaptación al mercado.

En relación con la comunidad educativa en general, esta investigación también puede servir como un recurso valioso en la enseñanza y el aprendizaje de la integración de sistemas de información en el ámbito empresarial. Los resultados y hallazgos de esta investigación pueden ser incorporados en programas de estudio y currículos académicos relacionados con sistemas de información, gestión empresarial y tecnologías de la información. De este modo, se enriquece el conocimiento y las competencias de los estudiantes, permitiéndoles comprender la importancia de la integración de sistemas y su aplicación práctica en el mundo empresarial.

Adicionalmente, la investigación tiene un impacto significativo en la comunidad educativa al fomentar la colaboración interdisciplinaria entre académicos de diversas áreas de estudio, como informática, administración, ingeniería de sistemas y economía, entre otras. Esta colaboración enriquece el enfoque de la investigación, promueve el pensamiento crítico y estimula la creatividad en la búsqueda de soluciones a problemas complejos.

Finalmente, es importante destacar el valor académico que tiene para los investigadores comprobar la importancia del manejo eficaz de la información en las empresas, así como generar expectativas sobre cómo debería funcionar de manera eficiente la integración de sistemas empresariales.

Por otro lado, la integración de sistemas como el ERP y el POS no solo permite una mejor gestión de la información dentro de una sucursal, sino que también posibilita su uso en otras sucursales propias o de aliados comerciales de la misma empresa, lo que podría redundar en mayores beneficios y una mejor eficiencia operativa.

PARTE II

DESCRIPCIÓN TEÓRICA

Martínez, M. (2006) sostiene que "la descripción teórica del trabajo es el conjunto de conceptos, teorías y enfoques que sirven como fundamentos y guías en una investigación". Siguiendo esta premisa, este capítulo se dedica a explicar y profundizar en dichos elementos que componen el marco teórico de este estudio. De esta manera, se busca establecer una base sólida y coherente que respalte y justifique las decisiones metodológicas implementadas en la investigación.

2.1 Antecedentes

Morales y Vásquez (2021) realizaron su trabajo de fin de grado en la Escuela Politécnica Nacional (EPN) en Quito, Ecuador, titulado: *DESARROLLO DE API REST PARA AUTOMATIZAR PROCESO DE RECEPCIÓN, SEGUIMIENTO Y CALIFICACIÓN DE PROYECTOS DE TITULACIÓN-ESFOT*, el cual se desarrolló dentro de un modelo cuantitativo, bajo el término de proyecto factible. Esta investigación tuvo por objetivo mejorar el proceso de titulación en la Escuela de Formación de Tecnólogos (ESFOT) de la Escuela Politécnica Nacional (EPN). Para ello, desarrollaron una API REST que automatiza la recepción, seguimiento y calificación de proyectos de titulación, con el fin de solucionar problemas como la centralización del proceso en las autoridades y el personal administrativo, la falta de comunicación entre tesistas y directores, y los largos tiempos de espera para la corrección y devolución de los planes. La implementación de esta API REST permitió la creación de un sistema web que ofrece beneficios como el monitoreo de tiempos, registro de acciones, envío de notificaciones, registro de calificaciones y observaciones, entre otros. Gracias a la escalabilidad y replicabilidad del sistema, la implementación de esta API REST pudo mejorar significativamente el proceso de titulación en la ESFOT.

De tal manera el trabajo de fin de grado de dichos autores es un excelente ejemplo de cómo las APIs pueden ser utilizadas para automatizar tareas complejas y mejorar procesos en distintos ámbitos. Es relevante para el uso de APIs en la carga de información, ya que demuestra cómo se puede utilizar esta tecnología para mejorar procesos que antes eran manuales y centralizados. Al permitir la automatización de las tareas, se logra una mayor eficiencia y calidad en la gestión de

proyectos de titulación. De esta manera, se evidencia el potencial de las APIs para la creación de sistemas más eficientes y productivos en distintas áreas.

Por otro lado, Martínez (2019) en su trabajo de fin de grado presentado en la Universidad de Sevilla: *DESARROLLO DE UNA API REST PARA LA GESTIÓN INTEGRAL DE INVENTARIOS Y PUNTOS DE VENTA*, Desarrollado bajo el modelo de investigación cuantitativo de tipo proyecto factible; buscó desarrollar una API REST que permitiera la administración de inventarios y la integración de sistemas de punto de venta (POS) en diferentes plataformas. Martínez estudió la arquitectura REST y las mejores prácticas para el diseño de APIs, y luego implementó una API que facilitaba la comunicación entre sistemas de gestión de inventarios y POS. Este trabajo es relevante para la investigación, ya que aborda el tema de la sincronización entre sistemas POS y otros sistemas empresariales.

De manera similar, García (2018) en su trabajo de fin de grado presentado ante la Universidad Autónoma de Madrid: *DEFINICIÓN E IMPLEMENTACIÓN DE UNA API REST PARA SISTEMAS DE RECOMENDACIÓN*; Este trabajo presentó una metodología documental, donde la información obtenida fue filtrada y analizada mediante criterios de inclusión. Tenía como objetivo implementar una API REST basada en un Sistema de Recomendación, que permitiera generar recomendaciones personalizadas para el usuario gracias a la utilización de algoritmos de filtrado colaborativo y librerías externas. Para lograrlo, el autor realizó un estudio previo sobre los Sistemas de Recomendación, técnicas y evaluaciones existentes, y aplicaciones populares. A partir de este estudio, diseñó una estructura simple basada en usuarios, ítems y eventos, y desarrolló la API REST con éxito. Además, se creó una interfaz de usuario para probarla de manera gráfica y se realizaron pruebas unitarias para garantizar su correcto funcionamiento.

En general, la implementación de una API REST para Sistemas de Recomendación es posible y efectiva para ofrecer recomendaciones personalizadas a los usuarios, y es importante conocer las distintas técnicas y evaluaciones para seleccionar la más adecuada para el problema a resolver. Los hallazgos y conclusiones del trabajo de García son de utilidad para el desarrollo de proyectos que requieran la implementación de sistemas de recomendación basados en API REST. En este sentido, la implementación exitosa de este caso de estudio, puede proporcionar una base sólida para el desarrollo de servicios similares en otros ámbitos, además de proporcionar un ejemplo de una estructura para APIs como lo es la estructura RESTful.

Para concluir, Meneses (2018) en su trabajo de grado para la Universidad Nacional Mayor de San Marcos: *DESARROLLO DE UNA API REST PARA LA INTEGRACIÓN DE CRM CON EL SISTEMA DE GESTIÓN DE COLAS EN EMPRESA DE FONDOS COLECTIVOS*, abordado mediante un modelo cuantitativo de tipo proyecto factible; tenía como meta mejorar la atención al cliente, y se logró mediante el desarrollo de una API REST con especificaciones claras e intuitivas, que permitió la actualización en línea del estado de atención del ticket atendido desde el CRM, y el desarrollo de un servicio web que permitió realizar la encuesta de atención desde una tablet conectada por USB. La implementación se consideró exitosa, ya que se logró la creación de un nuevo canal de generación y atención de tickets que permite la atención del cliente desde el CRM.

Por tal motivo el trabajo de Meneses es un ejemplo concreto de cómo una API puede ser utilizada para integrar sistemas. Es una referencia para aquellos que busquen implementar soluciones similares, y en particular, puede ser útil para el solicitante de los antecedentes al momento de considerar el uso de una API REST para mejorar la integración de sistemas en su propia empresa.

2.2 Bases teóricas

2.2.1 Sistemas de Punto de Venta (POS)

En esta investigación se destaca la importancia de los sistemas de punto de venta (POS) en la gestión de operaciones comerciales y su impacto en la eficiencia de las empresas. Según Pantano, E. y Viassone, M. (2015: p. 323), un sistema de punto de venta (POS) es "un conjunto de dispositivos electrónicos y software utilizado para registrar y completar transacciones minoristas". Esta afirmación resalta la relevancia de los sistemas de POS en el ámbito de ventas, como GUUAO Marketplace.

Por otro lado, He, Y. y Siau, K. (2017) argumentan que:

Aunque los sistemas POS son cruciales para la gestión de las operaciones comerciales, también pueden presentar desafíos en términos de seguridad y confidencialidad de los datos. Los autores sugieren que las empresas deben tener en cuenta estos riesgos al elegir e implementar un sistema de POS y tomar medidas adecuadas para proteger la información de los clientes y la empresa.

En contraposición, Sharma, G. y Roy, R. (2019) sostienen que:

Con el advenimiento de nuevas tecnologías, como la inteligencia artificial y el análisis de datos, los sistemas de POS modernos pueden ofrecer una serie de beneficios adicionales a las empresas, más allá de la simple gestión de transacciones. Estos beneficios incluyen una mayor eficiencia en la toma de decisiones, una mejor gestión de inventarios y una mayor capacidad para personalizar las ofertas a los clientes, lo que puede conducir a una mayor satisfacción del cliente y un aumento de las ventas

A partir de estas perspectivas, se infiere que, si bien los sistemas de POS son esenciales para el éxito de las empresas minoristas, es importante considerar tanto los riesgos asociados con su implementación como los beneficios adicionales que pueden proporcionar las tecnologías emergentes. Al abordar adecuadamente estos desafíos y aprovechar al máximo las oportunidades, las empresas como GUUAO Marketplace pueden mejorar significativamente su eficiencia y competitividad en el mercado.

2.2.1.1 Componentes de un sistema POS

Según Bobrowski, K. y Markewicz, I. (2018), los sistemas de POS "son sistemas informáticos que permiten registrar y gestionar las transacciones comerciales realizadas en un establecimiento". Se evidencia que los sistemas de POS han sido diseñados para apoyar las operaciones comerciales diarias, tales como la gestión de ventas y el control de inventario. Estos sistemas suelen incluir componentes como terminales de venta, lectores de código de barras, impresoras de recibos y sistemas de gestión de pagos.

Por otra parte, Roh, J., Kunnathur, A., y Tarafdar, M. (2014) señalan que "los sistemas de POS integran tanto hardware como software en sus operaciones, permitiendo llevar a cabo transacciones y gestionar inventarios". Y es que, estos elementos de software incluyen bases de datos, interfaces de usuario y registros de actividades, y ofrecen funcionalidades clave como el procesamiento de pagos y la gestión de inventarios.

Sin embargo, Nguyen, T. y Kelly, L. (2016) argumentan que:

Los sistemas de POS son esenciales para las empresas minoristas, la complejidad de los componentes y las integraciones de software

pueden generar dificultades en su implementación y mantenimiento (...) un sistema de POS adecuado requiere una cuidadosa evaluación de las necesidades específicas de la empresa y la compatibilidad con las soluciones de software y hardware existentes.

En contraste, Fransson, A., y Pascoa, R. (2018) sostienen que:

Los avances tecnológicos, como la adopción de soluciones basadas en la nube, pueden simplificar la gestión y el mantenimiento de los sistemas de POS, al tiempo que ofrecen nuevas oportunidades para mejorar la eficiencia y la rentabilidad de las empresas.

A partir de estas perspectivas, se puede afirmar que un sistema POS adecuado debe incluir tanto componentes de hardware como de software que satisfagan las necesidades específicas de la empresa y se integren adecuadamente con las soluciones existentes. Además, estos sistemas pueden incluir o hacer uso tecnologías como el manejo de la nube para gestión de almacenamiento y mantenimiento.

2.2.1.2 Tipos de sistemas de POS

Al respecto, Murdick, R., Ross, J., y Claggett, R. (2004) mencionan que:

Los sistemas de POS pueden ser clasificados en dos categorías principales: sistemas de POS tradicionales y sistemas de POS basados en la nube (...). Los sistemas POS tradicionales son aquellos que se instalan localmente en los dispositivos de la empresa y requieren de un servidor local, mientras que los sistemas de POS basados en la nube funcionan a través de internet y almacenan los datos en servidores remotos, permitiendo una mayor movilidad y accesibilidad a la información.

Sin embargo, Kim, D. y Lim, J. (2011) argumentan que, “Aunque los sistemas de POS basados en la nube ofrecen ventajas como la movilidad y el acceso remoto a la información, también pueden presentar desafíos en términos de seguridad y privacidad de los datos”.

Por otro lado, Martin, J. y Nwulu, N. (2017) sostienen que:

La adopción de sistemas de POS basados en la nube puede mejorar significativamente la eficiencia y la rentabilidad de las empresas, ya que facilita la integración con otras soluciones de software y permite la automatización de procesos. (...) los sistemas de POS basados en la nube pueden ofrecer una mayor escalabilidad y flexibilidad en comparación con los sistemas de POS tradicionales.

Teniendo en cuenta estas perspectivas, se deduce que la elección entre un sistema de POS tradicional y uno basado en la nube debe depender de las necesidades específicas de la empresa, así como de la evaluación de factores como la seguridad de los datos, la privacidad, la integración con otras soluciones de software y la escalabilidad, adicionalmente el autor de la presente investigación propone tomar en cuenta otra clasificación de POS dividida en sistemas propios o de terceros. Los sistemas propios harían referencia a aquellos sistemas cuya propiedad intelectual pertenezca a la empresa, y por ende se tenga acceso a su código fuente y los sistemas de terceros conformados por aquellos sistemas que requieran de una licencia o que requiera de un tercero para modificar sus funciones.

En el caso de GUUAO Marketplace, la empresa utiliza actualmente un sistema de POS tradicional y de terceros instalado en un servidor local ubicado en el edificio principal de la empresa GUUAO, accediendo los usuarios desde GUUAO Marketplace a través de accesos remotos al servidor local de la empresa GUUAO.

2.2.2 Sistemas ERP (Enterprise Resource Planning)

Monk, E. y Wagner, B. (2012) definen un sistema ERP como "un conjunto de aplicaciones de software que permiten a las organizaciones integrar y gestionar sus procesos empresariales de manera centralizada y estandarizada".

Por otro lado, Davenport, T. (1998) argumenta que, "Los sistemas ERP pueden proporcionar una integración eficiente y la gestión de procesos empresariales, también pueden ser costosos y difíciles de implementar".

Ciertamente, las empresas deben evaluar cuidadosamente los costos y beneficios asociados con la adopción de un sistema ERP antes de tomar una decisión.

Sin embargo, Umble, E. y Umble, M. (2003) sugieren que, "La implementación exitosa de un sistema ERP puede generar ventajas competitivas significativas para las empresas, incluida la mejora de la eficiencia operativa, la reducción de costos y la mejora de la calidad del servicio al cliente".

Según los autores, los sistemas ERP ofrecen una visión unificada y en tiempo real de la información y los recursos de la empresa, lo que permite una toma de decisiones más informada y

eficiente, estos sistemas deben de estar adaptados a los requisitos de la empresa y por la misma razón su adopción suele ser costosa. En el contexto de la investigación, el ERP de la empresa GUUAO Marketplace es utilizado por el personal de la empresa para el manejo y visualización de la información a través de una sola plataforma, dicha información incluye el manejo de productos, existencias de los mismos, precios y facturas.

2.2.2.1 Componentes y funcionalidades de los sistemas ERP

Gattiker, T. y Goodhue, D. (2005) definen un ERP como "un sistema de software integrado y modular que respalda el flujo de información entre las distintas funciones empresariales de una organización".

De manera similar, Davenport T. (1998) afirmó que "son sistemas de información que integran y automatizan muchas de las prácticas comerciales asociadas con los aspectos operativos o productivos de una empresa".

Por otro lado, Kumar, V. y Reinartz, W. (2016) sostienen que:

La selección adecuada de los componentes y funcionalidades del ERP es esencial para garantizar la alineación entre las necesidades de la empresa y las capacidades del sistema (...) para aprovechar al máximo las ventajas de un sistema ERP, es fundamental identificar y seleccionar las funcionalidades y componentes que mejor se adapten a las necesidades y objetivos específicos de la empresa.

En base a esto, se entiende que los sistemas ERP tienen como objetivo principal la integración y automatización de procesos empresariales, permitiendo una mayor eficiencia y control en la gestión de los recursos. Los componentes y funcionalidades de un sistema ERP incluyen diferentes áreas operativas y productivas de la empresa, como finanzas, recursos humanos, producción, cadena de suministro, ventas y marketing, entre otras diversas funcionalidades según la necesidades y módulos manejados por la empresa. Estos módulos trabajan de manera integrada unificados en el sistema ERP, facilitando la comunicación entre departamentos y la toma de decisiones basada en datos actualizados y precisos.

Por dicha razón, conocer de los diferentes módulos que conforman el ERP de la empresa GUUAO Marketplace en sus distintos departamentos es necesario para la integración de la información manejada en el POS de la empresa.

2.2.2.2 Tipos de sistemas ERP

Klaus, H. y Rosemann, M. (2000) divide los sistemas ERP en:

Los sistemas ERP genéricos son soluciones universales que pueden ser utilizadas por una amplia gama de empresas; los sistemas ERP verticales están diseñados para satisfacer las necesidades específicas de un sector industrial; y los sistemas ERP de nicho se centran en áreas funcionales específicas dentro de una empresa.

Por otra parte, Light, B. (2001), comenta que “Los sistemas ERP basada en módulos permiten a las empresas seleccionar e implementar soluciones específicas de acuerdo con sus necesidades funcionales y operativas particulares”.

Sin embargo, Gupta, A. y Tschirgi, M. (2003) sostienen que:

Cada tipo de sistema ERP tiene sus ventajas y limitaciones, la elección adecuada de un tipo de sistema ERP depende de factores como la estructura organizativa de la empresa, la estrategia de negocio y el nivel de adaptabilidad requerido. (...) las empresas pueden optar por combinar diferentes tipos de sistemas ERP para satisfacer sus necesidades específicas y obtener el máximo beneficio.

Lo planteado por los autores sugiere que, dependiendo de las necesidades de una empresa, se puede elegir un sistema ERP específico que se adapte mejor a sus características y requerimientos. Esto proporciona una base sólida para la identificación del tipo de sistema ERP utilizado por GUUAO Marketplace, teniendo en cuenta sus necesidades específicas y las características del sector en el que opera, se pudo identificar el sistema ERP de la GUUAO Marketplace como un sistema ERP modular.

2.2.3 Integración de sistemas

Lamb, J. (2005) describe la integración de sistemas como "el proceso de conectar y coordinar distintos sistemas de información y componentes de hardware y software para mejorar la eficiencia y efectividad en la gestión de la información y los procesos empresariales".

Sin embargo, Themistocleous, M. (2004) señala que "la integración de sistemas se realiza mediante la creación de interfaces que permiten la comunicación y el intercambio de datos entre sistemas independientes"

Por otro lado, Chung, S., y Hossain, M. A. (2009) argumentan que "la integración exitosa de sistemas requiere una comprensión clara de las necesidades de la organización y un enfoque estructurado para abordar los desafíos técnicos y organizativos". Y es que, los autores enfatizan la importancia de considerar tanto aspectos técnicos como organizativos en el proceso de integración de sistemas, lo cual incluye la identificación de requerimientos, la selección de tecnologías apropiadas y la gestión del cambio.

Según lo establecido, la integración de sistemas puede definirse como un proceso estructurado y multifacético que busca unificar y armonizar diferentes sistemas informáticos dentro de una organización, con el propósito de mejorar la eficiencia y efectividad en la gestión de los procesos de negocio.

En el contexto de la empresa GUUAO Marketplace, la integración exitosa del sistema de POS y el ERP requiere un enfoque que considere la arquitectura de ambos sistemas, las tecnologías de integración disponibles y las necesidades organizativas, a fin de lograr una comunicación efectiva y una coordinación adecuada que permita optimizar la gestión de los procesos de negocio, es fundamental seguir un enfoque estructurado que considere las necesidades específicas de la empresa y los desafíos técnicos y organizativos involucrados en la integración.

2.2.3.1 Técnicas y métodos de integración de sistemas

Allen, P. (2006) establece que:

Una visión de alto nivel de los métodos de integración se puede clasificar en cuatro categorías: transferencia de archivos, base de datos compartida, invocación de procedimientos remotos e integración orientada a servicios. La elección del enfoque de integración está impulsada por los requisitos del negocio y los recursos disponibles.

En dicho texto se establece que el método de integración debe ser moldeado según los requisitos y recursos disponibles.

En este sentido, Taylor, R. (2010), establece que "El enfoque más simple de integración de sistemas es el intercambio de archivos, en el que los sistemas intercambian datos a través de archivos". Según lo establecido por Taylor R. dicha integración se enfoca solamente en el intercambio de archivos entre sistemas.

Respecto al acceso remoto a bases de datos, Ambler, S. (2002) indica que "El acceso a bases de datos compartidas es un enfoque común para la integración de sistemas, pero puede generar problemas de seguridad y control de acceso".

En el texto, el autor plantea que dicho enfoque consiste en utilizar una misma base de datos en diferentes aplicaciones o sistemas para almacenar y recuperar información, facilitando la interoperabilidad y el intercambio de datos entre ellos. Adicionalmente sugiere que, al permitir que múltiples sistemas accedan a la misma base de datos, pueden surgir problemas de privacidad, seguridad y autorización.

De manera similar, Coulouris, G. y Dollimore, J. (2012) mencionan que la invocación de procedimientos remotos "Permite la comunicación entre sistemas a través de la llamada a procedimientos remotos, pero puede ser compleja y difícil de mantener".

Este enfoque consiste en que un sistema puede realizar llamadas a procedimientos (o funciones) que se encuentran en otro sistema como si estuvieran disponibles localmente. Sin embargo, los autores también mencionan que este enfoque puede ser complejo y difícil de mantener debido a la naturaleza distribuida de los sistemas involucrados y a las posibles diferencias en las plataformas y lenguajes de programación.

Por otro lado, Erl, T. (2005) sostiene que "La orientación a servicios es un enfoque moderno y escalable que permite la integración de sistemas a través de servicios web bien definidos". Lo planteado habla sobre la integración orientada a servicios, en este enfoque, los sistemas se comunican e interactúan mediante la exposición y consumo de servicios que siguen un conjunto de estándares y protocolos, como SOAP o REST, que facilitan la interoperabilidad.

2.2.3.2 Estándares y protocolos de integración de sistemas

Chappell, D. (2004), establece que:

(...) los estándares y protocolos son fundamentales para la integración de sistemas y para la interoperabilidad entre aplicaciones empresariales. (...) los protocolos como SOAP, REST y JSON desempeñan un papel crucial en la comunicación entre sistemas mientras que los formatos de datos como XML y JSON permiten una representación de datos eficiente y flexible.

Lo planteado por el autor mencionado destaca la importancia de los estándares, protocolos y formatos de datos en el contexto de la integración de sistemas. Estos elementos son esenciales para garantizar la comunicación efectiva y el intercambio de información entre diferentes sistemas y aplicaciones en el entorno, como en el presente caso de estudio.

En contraste, Krafzig, D., Banke, K., y Slama, D. (2004) argumentan que:

La adopción de una arquitectura de orientación a servicios (SOA) puede facilitar aún más la comunicación y la interoperabilidad entre diferentes aplicaciones empresariales. (...) SOA permite a las organizaciones diseñar, implementar y gestionar servicios web que pueden ser consumidos fácilmente por otras aplicaciones, lo que simplifica la integración de sistemas y reduce la complejidad general.

Por otro lado, Hohpe, G. & Woolf, B. (2004) destacan que:

Los patrones de integración abordan problemas comunes en la integración de sistemas y proporcionan soluciones probadas y estandarizadas que pueden ser adaptadas a las necesidades específicas de cada caso. Al utilizar patrones de integración, las organizaciones pueden simplificar y estandarizar sus enfoques de integración, lo que facilita la comunicación y el intercambio de información entre sistemas y aplicaciones.

Una estrategia efectiva de integración de sistemas e interoperabilidad entre aplicaciones empresariales debe combinar el uso de estándares y protocolos adecuados, la adopción de una arquitectura orientada a servicios y la implementación de patrones de integración. Al aplicar un enfoque integrado que incorpore estos elementos, las organizaciones pueden garantizar una comunicación efectiva y un intercambio de información eficiente entre sistemas y aplicaciones, mejorando así la eficiencia operativa y la competitividad en el mercado.

2.2.3.3 Desafíos y soluciones en la integración de sistemas

Hohpe, G. y Woolf, B. (2003) afirman que "los sistemas de software a menudo tienen que trabajar juntos para lograr lo que un solo sistema no puede". Además, explican que "la integración de sistemas a menudo plantea nuevos problemas de diseño y rendimiento"

En este proceso, enfrentan diversos desafíos, como la heterogeneidad de plataformas y tecnologías, y la falta de estandarización. Según Hohpe, G. y Woolf, B. (2004), "uno de los mayores desafíos de la integración es lidiar con la diversidad de sistemas y aplicaciones, cada uno con su propio conjunto de protocolos, formatos de datos y semántica". Además, los autores enfatizan la importancia de abordar los desafíos de seguridad y rendimiento, indicando que "la integración de sistemas puede introducir riesgos de seguridad adicionales y aumentar la posibilidad de fallas en cascada".

Complementando esta perspectiva Alonso, G., Casati, F., Kuno, H., y Machiraju, V. (2004) señalan que "la integración de sistemas puede incluir también la necesidad de escalabilidad y adaptabilidad, especialmente en entornos empresariales en constante cambio."

Por otro lado, Chappell, D. (2004) destaca que, "SOA permite la creación de aplicaciones empresariales flexibles y escalables al descomponer la funcionalidad en servicios independientes que se pueden reutilizar y combinar según las necesidades del negocio".

Basado en lo planteado por los autores, es necesario reconocer la importancia de los estándares y protocolos en el proceso de integración. La adopción de patrones de integración y la implementación de arquitecturas orientadas a servicios son enfoques complementarios que pueden ayudar a abordar estos desafíos y permitir una comunicación efectiva y confiable entre sistemas heterogéneos. Adicionalmente, una combinación de patrones de integración y arquitecturas orientadas a servicios puede proporcionar una base sólida para la creación de sistemas integrados flexibles, escalables y adaptables. Y para esto los estándares y protocolos juegan un papel fundamental en el proceso, ya que permiten la compatibilidad entre diferentes sistemas y tecnologías.

Al considerar la importancia de los estándares y protocolos, se propone que el proceso de integración del sistema POS y el ERP en GUUAO Marketplace se lleve a cabo siguiendo las mejores prácticas y utilizando tecnologías estandarizadas. Esto asegurará que la API desarrollada

permita una comunicación confiable y efectiva entre los sistemas, y facilitará la adaptación y escalabilidad del sistema integrado en el futuro, a medida que cambien las necesidades del negocio y evolucionen las tecnologías disponibles.

2.2.4 Servicios API (Application Programming Interface)

Según Masse, M. (2011), una API (Interfaz de Programación de Aplicaciones) es "un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar software de aplicaciones". Las API permiten que diferentes aplicaciones se comuniquen entre sí, facilitando la interoperabilidad y el acceso a funciones y datos específicos.

Por otra parte, Masse, M. (2011) también indica que:

Las API actúan como un puente entre diferentes sistemas o aplicaciones, permitiendo el intercambio de información de manera estandarizada. (...) facilita la creación de aplicaciones más complejas y adaptadas a las necesidades del usuario, ya que un desarrollador puede aprovechar las funciones y datos ofrecidos por otras aplicaciones sin tener que recrearlos desde cero.

De igual manera, Masse, M. (2011) explica que:

Las rutas y los endpoints son elementos cruciales en las APIs. Las rutas actúan como "mapas" que guían a los métodos de peticiones hacia su destino, mientras que los endpoints funcionan como "direcciones específicas" dentro de ese mapa, asegurando que la información llegue a su destino correcto.

En el texto describe la función y el valor de las API (Application Programming Interfaces) en el desarrollo de software y la interacción entre sistemas, subraya la importancia de las API en el ámbito del desarrollo de software, destacando su papel en la facilitación de la comunicación entre sistemas.

Por otro lado, Zang, N. y Rosson, M. B. (2014) señalan que, "Las API también pueden presentar dificultades para los desarrolladores debido a la diversidad y complejidad de sus interfaces. Los autores sugieren que es importante proporcionar documentación clara y soporte adecuado para facilitar el uso efectivo de las API".

En contraposición, Vukovic, M. y Bartolini, C. (2016) argumentan que:

El crecimiento exponencial de las API y su adopción generalizada ha llevado a la aparición de nuevas metodologías y tecnologías que simplifican la creación, gestión y consumo de API, como API Gateway y API Management. Estas soluciones ayudan a abordar los desafíos de integración y permiten a las organizaciones aprovechar al máximo las ventajas de las API en el desarrollo e integración de sistemas y aplicaciones.

De acuerdo con los autores, las API son herramientas poderosas que permiten a los desarrolladores y organizaciones acceder y utilizar funcionalidades y datos de diferentes sistemas y aplicaciones de manera estandarizada y eficiente. Esto, a su vez, promueve la interoperabilidad, la colaboración y la adaptabilidad en el ámbito del desarrollo de software y la gestión de sistemas empresariales. Las API, al ser esenciales para facilitar la comunicación entre sistemas y aplicaciones, son cruciales en el proceso de integración entre el sistema de POS y el ERP de GUUAO Marketplace.

2.2.4.1 Tipos de API y arquitecturas (REST, SOAP, GraphQL)

Fielding, R. y Taylor, R. (2002) describen que "REST es un estilo arquitectónico para sistemas de red que aprovecha un conjunto de protocolos basados en la arquitectura cliente-servidor". En esta cita, los autores resaltan que REST es un enfoque popular para diseñar API debido a su simplicidad y escalabilidad, aprovechando el protocolo HTTP.

En este contexto, Fielding, R. y Taylor, R. (2002) también señalan que:

Los métodos HTTP como GET y POST son esenciales para la interacción en una arquitectura REST (...) Se puede considerar el método GET como un acto de "solicitar" información. (...) En contraposición, el método POST se asemeja a "entregar" información.

En contraste, Box, D. y Ehnebuske, D. (2000) afirman que "SOAP es un protocolo de mensajería basado en XML diseñado para la comunicación entre aplicaciones a través de Internet". Destacan que SOAP es un protocolo basado en el formato XML para la transmisión de información entre aplicaciones.

Por otra parte, Schrock, N. (2015) explica que GraphQL "permite a los clientes solicitar exactamente lo que necesitan y nada más (...) creado por Facebook para resolver muchas de las inefficiencias en las API REST". Esta cita sugiere que GraphQL es una alternativa flexible y eficiente a REST y SOAP, proporcionando a los desarrolladores un control preciso sobre los datos solicitados y devueltos en aplicaciones con requisitos de datos complejos o cambiantes.

En el caso del presente caso de estudio, la REST fue elegida como como la arquitectura utilizada para desarrollar la API, ya que permite la simplificación de la comunicación entre los sistemas a través del protocolo HTTP mediante métodos de GET y POST, el cual se puede utilizar para establecer comunicación con el ERP de la empresa, considerando que actualmente el mismo se encuentra ubicado en la Web.

2.2.4.2 Evaluación y monitoreo de las API

La evaluación y monitoreo de las API implican llevar a cabo prácticas que permiten medir y asegurar el rendimiento, la seguridad y la confiabilidad de las APIs a lo largo del tiempo. Según Fielding, R. (2000), "El monitoreo y la evaluación de las API son fundamentales para garantizar que las aplicaciones basadas en ellas funcionen correctamente y cumplan con los requisitos de los usuarios".

Este enfoque propuesto por Fielding expresa la importancia de considerar el monitoreo y la evaluación de las API como una parte integral de su ciclo de vida. Al analizar el texto, se puede comprender que mantener un alto nivel de calidad en las API es crucial para que las aplicaciones basadas en ellas puedan ofrecer un servicio eficiente y confiable a los usuarios finales.

Por otro lado, Mulesoft, empresa dedicada a ofrecer software de integración señala que: "Es fundamental garantizar que las API se integren correctamente en el ecosistema de aplicaciones y sistemas existentes. La falta de una integración adecuada puede generar problemas de compatibilidad y rendimiento, lo que afectaría negativamente la experiencia del usuario".

En contraste, Lewis, T., & Fowler, M. (2014) argumentan que:

El monitoreo y la evaluación de las API no solo deben centrarse en aspectos técnicos como la seguridad, el rendimiento y la confiabilidad, sino también en la facilidad de uso y la experiencia

del desarrollador. Estos autores sugieren que las API bien diseñadas y fáciles de usar pueden facilitar la adopción de nuevas tecnologías y servicios en las empresas, lo que podría conducir a una mayor innovación y competitividad.

Teniendo en cuenta las perspectivas de los autores, una estrategia efectiva de evaluación y monitoreo de las API debe abordar tanto los aspectos técnicos como la experiencia del desarrollador y la integración en el ecosistema de aplicaciones y sistemas existentes. Al combinar estos enfoques, las empresas pueden garantizar que sus APIs ofrecen un alto nivel de calidad y facilitan la adopción de nuevas tecnologías y servicios, lo que puede mejorar su eficiencia y competitividad en el mercado.

2.5 Bases legales

2.5.1 Constitución De La República Bolivariana De Venezuela (publicada en Gaceta Oficial Extraordinaria N° 36.860, de fecha 30 de diciembre de 1999)

Artículo 110. - El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para los mismos. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía.

La mencionada ley resalta la importancia de la ciencia, la tecnología, el conocimiento y la innovación en el desarrollo económico, social y político del país. También destaca el papel del Estado y del sector privado en el fomento y desarrollo de estas actividades, así como la garantía del cumplimiento de principios éticos y legales en las actividades de investigación científica y tecnológica.

2.5.2 Ley Orgánica De Ciencia, Tecnología E Innovación (publicada en Gaceta Oficial Extraordinaria N° 37.291, de fecha 26 de septiembre de 2001)

Artículo 19. - De la propiedad intelectual. La autoridad nacional con competencia en materia de ciencia, tecnología, innovación y sus aplicaciones, formulará las políticas y los programas donde se establecen las condiciones de la titularidad y la protección de los derechos de propiedad intelectual derivadas de la actividad científica, tecnológica y sus aplicaciones que se desarrollen con sus recursos o los de sus órganos y entes adscritos conjuntamente con el Servicio Autónomo de Propiedad Intelectual (SAPI).

Dicho artículo establece las condiciones de titularidad y protección de los derechos de propiedad intelectual relacionados con la actividad científica y tecnológica. Asegura la protección de los derechos de autor y la propiedad intelectual en el desarrollo de la presente investigación.

Artículo 21. - Invención e innovación popular. La autoridad nacional con competencia en materia de ciencia, tecnología, innovación y sus aplicaciones creará mecanismos de apoyo, promoción y difusión de invenciones e innovaciones populares, que generen bienestar a la población o logren un impacto económico o social en la Nación.

El presente artículo destaca el compromiso de la autoridad nacional en la creación de mecanismos de apoyo, promoción y difusión de invenciones e innovaciones tecnológicas, como el servicio API que propone el presente proyecto, que puede generar bienestar a la población o tener un impacto económico o social en el país.

2.5.3 Ley Contra los Delitos Informáticos (publicada en Gaceta Oficial N° 37.313, de fecha 30 de octubre de 2001)

Artículo 2. - Objeto. Esta Ley tiene por objeto la protección integral de los sistemas que utilicen tecnologías de información, así como la prevención y sanción de los delitos cometidos contra éstos o cualquiera de sus componentes, o aquellos delitos cometidos mediante el uso de dichas tecnologías, en resguardo de la seguridad y defensa de la Nación, la economía del Estado, la industria nacional, los ciudadanos y las personas jurídicas.

Este artículo establece la protección integral de los sistemas que utilizan tecnologías de información, así como la prevención y sanción de los delitos informáticos. Dado que el presente

caso de estudio involucra el desarrollo de un servicio API para la sincronización entre el sistema POS y el ERP, es relevante considerar esta ley en el marco legal del estudio para garantizar la seguridad y protección de los datos y sistemas involucrados.

2.5.4 Ley de Telecomunicaciones (publicada en Gaceta Oficial N° 36.970, de fecha 12 de junio de 2000)

Artículo 4. - De las telecomunicaciones y la informática. Las telecomunicaciones y la informática son instrumentos fundamentales para el desarrollo económico, social y político del país y, en consecuencia, para la seguridad y defensa nacional.

Este artículo enfatiza la importancia de las telecomunicaciones y la informática en el desarrollo económico, social y político del país. Dado que la presente investigación involucra la integración de un servicio API que utilizará redes de telecomunicaciones para la sincronización entre sistemas, es relevante considerar esta ley en el marco legal de la investigación.

2.5.5 Ley de Infogobierno (publicada en Gaceta Oficial N° 40.274, de fecha 17 de octubre de 2013)

Artículo 12. - De la responsabilidad en el uso de las Tecnologías de Información. Los órganos y entes del sector público y privado, así como las personas naturales y jurídicas, serán responsables por el uso indebido de las Tecnologías de Información en el ejercicio de sus competencias o actividades.

El presente artículo establece la responsabilidad de los órganos y entes del sector público y privado, así como de las personas naturales y jurídicas, por el uso indebido de las Tecnologías de Información. En el contexto de la investigación, esta ley es aplicable en cuanto a la responsabilidad que recae sobre la empresa GUUAO Marketplace y los usuarios de la API en el uso adecuado y legal de los datos en la tecnología desarrollada.

2.6 Definición de términos

API (Application Programming Interface):

"Conjunto de reglas, protocolos y herramientas para construir aplicaciones de software que permite la comunicación e interacción entre diferentes componentes de software." (Mulesoft, 2017)

Archivo:

"Conjunto de documentos, generalmente textos, que se guardan en un soporte físico o digital para su conservación y consulta." (Real Academia Española, s.f.)

Base de datos:

"Conjunto organizado de información estructurada, almacenada en formato electrónico y accesible a través de sistemas informáticos." (Real Academia Española, s.f.)

Cliente:

"En informática, se refiere al equipo o programa que solicita servicios o recursos de otro equipo o programa, conocido como servidor." (Tanenbaum, A., y Steen, M., 2007)

Endpoint:

"En el contexto de las redes de computadoras y programación, un endpoint (punto final) es un punto de conexión en una red de comunicaciones o en una interfaz de servicio. En los servicios web, los endpoints se refieren a las ubicaciones de los recursos a las que un cliente puede enviar una solicitud y desde las que puede recibir una respuesta". (Subramanian, S., 2018).

GET (Método HTTP GET):

"Un método de solicitud que recupera información de un recurso específico en un servidor" (Fielding, R. y Taylor, R., 2002).

HTTP (Hypertext Transfer Protocol):

"Protocolo de comunicación utilizado en la World Wide Web para transmitir y recibir información entre clientes y servidores web." (Mozilla Developer Network, s.f.)

Interfaz:

"Superficie o medio que permite la comunicación o interacción entre dos elementos, como sistemas informáticos o dispositivos electrónicos." (Preece, J., Rogers, Y., y Sharp, H., 2002)

JSON (JavaScript Object Notation):

"Formato de texto ligero para el intercambio de datos que es fácil de leer y escribir tanto para humanos como para máquinas." (Crockford, D., 2006)

POST (Método HTTP POST):

"Un método de solicitud diseñado para enviar datos a un servidor para crear un nuevo recurso." (Fielding, R. y Taylor, R., 2002).

Protocolo:

"Conjunto de reglas y procedimientos que permiten la comunicación entre diferentes dispositivos o sistemas informáticos." (Comer, D., 2000).

Red:

"Sistema de comunicación entre computadoras que permite compartir información y recursos a través de cables, ondas u otros medios de transmisión." (Stallings, W., 2004)

REST (Representational State Transfer):

"Un estilo arquitectónico para sistemas de red que aprovecha un conjunto de protocolos basados en la arquitectura cliente-servidor." (Fielding, R. y Taylor, R., 2002).

Ruta:

"En el contexto de programación y desarrollo web, La ruta o PATH se usa para limitar un dato específico que deseé de un servidor externo. Entonces, modificando la URL de inicio, agregando ciertos argumentos o factores de cambio según la información de la API, se obtendrán datos diferentes.". (Arias, E., 2022).

Servidor:

"En informática, se refiere al equipo o programa que proporciona servicios o recursos a otros equipos o programas, conocidos como clientes." (Tanenbaum, A., y Steen, M., 2007)

SOAP (Simple Object Access Protocol):

"Un protocolo de mensajería basado en XML diseñado para la comunicación entre aplicaciones a través de Internet." (Box, D. y Ehnebuske, D. 2000)

Software:

"Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar diversas tareas en un computador u otro dispositivo electrónico." (Pressman, R., y Maxim, B., 2018)

Soporte:

"En tecnología, se refiere al medio físico o digital que permite almacenar y conservar información, como discos duros, memorias USB, CD, DVD, entre otros." (Stallings, W., 2013).

URL (Uniform Resource Locator):

"Una referencia específica que apunta a un recurso en una red, que proporciona el medio para recuperar la representación del recurso, a través de un protocolo específico, como HTTP o FTP." (Fielding, R. y Taylor, R., 2002).

Web:

"Sistema de información global y público que se encuentra en Internet y permite el acceso a documentos y recursos multimedia a través de enlaces y navegadores." (Berners-Lee, T., y Cailliau, R., 1990)

XML (eXtensible Markup Language):

"Lenguaje de marcas que permite la representación de información estructurada en forma de texto legible por humanos y máquinas

PARTE III

DESCRIPCIÓN METODOLOGICA

Hernández, R., Fernández, C., y Baptista, P. (2010, p. 34) afirman que "el marco metodológico consiste en el conjunto de estrategias, técnicas y procedimientos que se utilizan para abordar y resolver cuestiones de investigación". Con base en esto, el presente capítulo tiene como finalidad explicar y detallar las estrategias, técnicas y procedimientos que se emplearán en el presente trabajo con el propósito de lograr los objetivos planteados en la investigación.

3.1 Naturaleza de la investigación

Creswell, J. (2009) afirma que:

La investigación cuantitativa es un enfoque para probar teorías objetivas mediante el examen de la relación entre variables. Estas variables, a su vez, pueden medirse, generalmente en datos numéricos, que luego se analizan usando procedimientos estadísticos.

En contexto de la presente investigación, se busca desarrollar una solución tecnológica que permita sincronizar eficientemente los datos entre el Sistema de POS y el ERP de GUUAO Marketplace. Para lograr este objetivo, es necesario medir y analizar variables cuantitativas relacionadas con la veracidad de la sincronización, la integridad de los datos, la consistencia y la fiabilidad del servicio API propuesto. El enfoque cuantitativo permite obtener datos numéricos precisos y realizar análisis estadísticos para evaluar el desempeño del servicio API en términos de sincronización de datos y optimizar su funcionamiento. Por tal razón, este enfoque es el más adecuado para abordar los objetivos de esta investigación.

3.2 Tipo de investigación

Creswell, J. (2009) señala que "un proyecto factible se centra en analizar y proponer soluciones viables a problemas específicos en contextos reales". Esta investigación se clasifica como un proyecto factible, ya que busca diseñar y desarrollar una estrategia de optimización para el proceso de producción en la empresa GUUAO Marketplace. El proyecto se enfoca en analizar la situación actual de la empresa y propone una solución práctica y viable para mejorar la eficiencia en la producción y la gestión de recursos.

3.3 Diseño de la investigación

Bryman, A. (2012) define la investigación de campo como aquella que:

Se basa en la recopilación de datos en el entorno natural en el que ocurren los fenómenos estudiados, permitiendo al investigador observar, analizar y comprender los comportamientos y experiencias de los participantes en su contexto real.

El estudio se centrará en la implementación de un servicio API para la sincronización del sistema POS y el ERP de la empresa GUUAO Marketplace, donde el autor de la presente investigación actualmente trabaja. Esto implica la posibilidad de realizar observaciones y análisis directos de los sistemas existentes, recolectar datos y evaluar las necesidades específicas para la sincronización y la integración de los sistemas. Adicionalmente, la interacción directa con los empleados y el análisis en su entorno real facilitará una comprensión profunda de los desafíos y oportunidades para mejorar la eficiencia y la gestión de datos en la empresa a través de la implementación del servicio API.

3.4 Población y muestra

Babbie, E. (2009), describe la población como "el conjunto de elementos o casos que el investigador quiere estudiar y que comparten ciertas características específicas".

De acuerdo a lo planteado por el autor, la población de la presente investigación estará conformada por el conjunto de empleados de la empresa GUUAO Marketplace que interactúan directamente con el manejo y desarrollo sistema de POS y el ERP. Los involucrados serían los tres (3) coordinadores, con respecto al área de soporte, redes y sistemas; dos (2) empleados responsables de la carga de datos en ambos sistemas, y por último el autor de la presente investigación.

Por otra parte, según Creswell, J. (2012), la muestra es "un subconjunto de la población al que el investigador selecciona para responder a las preguntas de la investigación".

De igual manera, Babbie, E. (2009), plantea que "En el muestreo intencional, también llamado muestreo de juicio o de propósito, el investigador selecciona los elementos de la muestra en función de su propio conocimiento de la población, sus elementos y la finalidad del estudio en sí."

En el contexto de la investigación, se tomará la muestra implementando un muestreo intencional a empleados que estén directamente involucrados en el uso de ambos sistemas y en el proceso de integración entre los mismos y que tengan conocimientos suficientes sobre los requerimientos de los sistemas. Por ende, se determina que la muestra estará conformada por los seis (6) empleados que conforman la población total de la investigación.

3.5 Técnicas de recolección de datos

Denscombe, M. (2010) sostiene que: "Las entrevistas estructuradas son similares a una encuesta en el sentido de que las preguntas son predeterminadas y están estandarizadas en términos de su contenido y orden".

Se consideró la aplicación de entrevistas estructuradas debido a que estas permiten obtener información específica y precisa acerca de las funcionalidades clave del sistema de POS y del ERP directamente de los expertos o empleados de la empresa GUUAO Marketplace. Además, al ser preguntas predeterminadas y estandarizadas, se garantiza la consistencia y comparabilidad de las respuestas obtenidas.

Por otra parte, Bowen, G. (2009) explica que: "La revisión documental es un método de investigación sistemático que implica la revisión y evaluación de documentos, tanto impresos como electrónicos"

La revisión documental se consideró adecuada, ya que hace uso del análisis y la evaluación de documentos relevantes, como manuales técnicos, estudios previos y especificaciones de software, que proporcionen información sobre las diferentes opciones de integración disponibles para la sincronización entre el sistema de POS y el ERP. Adicionalmente, permite el uso de cualquier material o documento proveídos internamente, como manuales de usuario y documentación de los sistemas de la empresa GUUAO.

Para finalizar DeWalt, K., y DeWalt, B. (2011) definen la observación participante como: "Un método de investigación en el cual el investigador se involucra en la vida diaria de las personas en su entorno natural con el fin de comprender sus perspectivas y prácticas".

Se optó por la observación participante ya que, al trabajar en la empresa, el autor de la presente investigación puede involucrarse en el proceso de implementación y evaluación del servicio API,

lo que permite obtener una perspectiva interna y práctica de cómo el servicio API influye en la eficiencia y adaptabilidad de la empresa GUUAO Marketplace. Este enfoque garantiza la obtención de información contextualizada y basada en la experiencia real en el entorno laboral.

3.6 Técnicas de análisis de datos

Rubio, D., Berg, M., Tebb, S., Lee, S., y Rauch, S. (2003) señalan que: "La validación de contenido es un proceso de revisión de los ítems del instrumento y la evaluación de su relevancia y representación con respecto al dominio del constructo a medir".

La mencionada técnica de análisis de datos se seleccionó porque permite evaluar la relevancia y representación de las funcionalidades clave identificadas mediante las entrevistas estructuradas en relación con el dominio del constructo a medir (sistema de POS y ERP). Esto asegura que las funcionalidades identificadas son pertinentes y adecuadas para el contexto de la empresa GUUAO Marketplace.

Por otra parte, Saldaña, J. (2015) describe la codificación como: "Un proceso analítico que simplifica y centra los datos brutos del texto y permite su transformación en un marco conceptual"

Se hizo la selección de la mencionada técnica, ya que permite organizar y sintetizar la información obtenida de la revisión documental de manera sistemática y estructurada. Al codificar y tabular la información, se facilita la comparación y evaluación de las diferentes opciones de integración y se pueden identificar patrones y tendencias relevantes para la sincronización entre el sistema de POS y el ERP de la empresa GUUAO Marketplace.

Para finalizar, Evans, J., y Lindsay, W. (2011) explican que: "La matriz FODA es una herramienta de análisis estratégico que permite identificar las Fortalezas, Oportunidades, Debilidades y Amenazas de una empresa u organización, con el fin de tomar decisiones informadas y fundamentadas en la realidad actual y potencial de la misma."

Esta técnica de análisis de datos puede ser aplicada en el contexto de la empresa GUUAO Marketplace para evaluar la eficacia del sistema de integración entre el POS y el ERP. La matriz FODA permite una visualización organizada de los aspectos internos (fortalezas y debilidades) y externos (oportunidades y amenazas) que pueden impactar en la implementación y funcionamiento del sistema integrado. Al analizar estos factores de manera sistemática y comparativa, se facilita

la toma de decisiones y la formulación de estrategias que maximicen las fortalezas y oportunidades, y minimicen las debilidades y amenazas.

PARTE IV

ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

El siguiente apartado tiene como objetivo demostrar y presentar los resultados obtenidos durante la presente investigación, considerando que la información recopilada es de importancia para el progreso y la configuración de la propuesta en cuestión y proporciona las condiciones para arribar a las conclusiones.

4.1 Identificación de las funcionalidades clave del sistema de POS y del ERP de la empresa GUUAO Marketplace.

La consecución de este objetivo consistió en identificar las capacidades fundamentales del POS y el ERP de GUUAO Marketplace, la compresión de las capacidades y usos de ambos sistemas dentro de la empresa es esencial para llevar a cabo la integración entre ellos, por lo que es primordial conseguir toda la información necesaria para conocer a nivel técnico cómo funciona y de qué se encarga-cada uno de ellos.

Dicho objetivo fue llevado a cabo con éxito mediante la aplicación de una entrevista realizada a los coordinadores de soportes y sistemas en conjunto, la cual permitió identificar y comprobar cada una de las funcionalidades vitales para el funcionamiento del sistema POS, así como también los diferentes módulos manejados por GUUAO Marketplace en el ERP de la empresa.

Se empleó como instrumento una guía de entrevista estructurada, con preguntas sobre el sistema POS, el cual fue entregado al coordinador de soporte y otro con preguntas sobre el ERP entregado al coordinador de sistemas de la empresa, adicionalmente ambos instrumentos estaban enfocados en obtener los requerimientos necesarios para la integración de los sistemas indicados y el desarrollo de la API correspondiente.

Cuadro 1: Entrevista sobre el sistema POS.

Pregunta	Respuesta
<p>¿Podría describir brevemente las funcionalidades del sistema POS de GUUAO Marketplace?</p>	<p>La emisión de documentos fiscales, como facturas y notas de crédito. La gestión de inventario, como la carga de productos, sus existencias y las categorías a las que pertenecen los productos. La administración de precios y almacenes y, por último, el sistema también se encarga de generar reportes de caja y procesar pagos a través de la máquina fiscal.</p>
<p>¿Cómo es el proceso de gestión de inventario y precios en el sistema POS?</p>	<p>La gestión de inventario se hace principalmente a través del departamento de data, ellos utilizan el sistema de gestión empresarial SAP para llevar a cabo la gestión de los inventarios en diferentes almacenes, para GUUAO MP los inventarios se cargan a través de SAP a la base datos del punto de venta. Los precios sí se cargan directamente en B2B (el ERP de la empresa) y luego se suben a la base de datos del punto de ventas a través del sistema administrativo del mismo.</p>
<p>¿Cómo se realiza el registro de ventas y emisiones de documentos fiscales?</p>	<p>El punto de venta está unido a distintas máquinas fiscales, cada una usada en las diferentes cajas de la tienda del Sambil, la máquina se encarga de colocar los datos y seriales fiscales en un documento generado por el punto de ventas en la base de datos, el punto de venta hace todo el proceso de registro de datos como cédula, teléfonos y líneas de la</p>

	factura y la máquina fiscal coloca los datos y seriales para que sea un documento fiscal válido.
¿Cuáles son los elementos más importantes en la gestión de los datos dentro del sistema POS?	El punto de ventas cuenta con su propio sistema administrativo que viene incluido en el paquete de la licencia del software, a través de este sistema administrativo es por donde se gestionan actualmente los datos del punto de ventas, sin embargo, todos estos datos son accesibles a través de la base de datos del POS, a la cual se puede acceder a través de SQL server.
¿Existen otras funcionalidades que considera vitales para el sistema POS y por qué?	Considero que ya se cubrieron todas las partes necesarias para el funcionamiento de la tienda MP en el Sambil, emisión de documentos fiscales, gestión de inventario y precios.
En su opinión, ¿cuáles son los elementos más importantes a considerar para la integración de ambos sistemas (POS y ERP)?	En cuanto a comunicación con los otros sistemas, ya sea el ERP o SAP, solo se necesitan de los siguientes datos: Productos (códigos de barra y producto, descripciones, precios, categorías, existencias) por almacenes, ya sea GS01 (MP Sambil) o HM01 (Barra Heineken) y luego que del punto de venta se envíen los documentos fiscales a B2B. Siempre que estos datos estén en sincronía en todos los sistemas, consideraría la integración como exitosa.

<p>¿Qué desafíos podrían surgir durante la integración de estos sistemas y cómo podrían ser superados?</p>	<p>El cómo se transmitirán los datos entre ambos sistemas, los datos almacenados en el punto de venta pueden ser accedidos sin problemas desde la base de datos, sin embargo, para enviar una factura al ERP, se debe de conocer muy bien la arquitectura de la base de datos y también de la factura misma, ya que datos como las líneas y documentos se guardan en tablas diferentes.</p>
<p>¿Qué beneficios espera GUUAO Marketplace de la implementación de una API para la sincronización del sistema POS y ERP?</p>	<p>La automatización de procesos, facilidad a la hora de consultar información almacenada en el POS, evitar fallos en la sincronización de datos, más seguridad a la hora de actualizar los precios, ya que se tomarán directo desde el ERP y, por último, la posibilidad de desarrollar reportes personalizados que contengan datos que no sean generados en otros reportes ya manejados por el POS.</p>

Cuadro 1: Entrevista sobre el sistema POS.

Fuente: Elaboración propia (2023).

Las funciones críticas del sistema POS, según el Coordinador de Soporte, Lenin Acosta, incluyen la emisión de documentos fiscales, la gestión del inventario, la administración de precios y almacenes, y la generación de informes de caja y procesamiento de pagos a través de la máquina fiscal.

En cuanto al proceso de gestión de inventario y precios, se encontró que este se lleva a cabo a través del sistema de gestión empresarial SAP y B2B (ERP de la empresa). El inventario se carga a través de SAP a la base de datos del punto de venta y los precios se cargan directamente

en B2B, luego se suben a la base de datos del punto de ventas a través del sistema administrativo del mismo.

En lo que respecta al registro de ventas y la emisión de documentos fiscales, este se realiza mediante la máquina fiscal que se une al punto de venta, y luego se generan en la base de datos del punto de venta. Finalmente, la entrevista reveló que los datos de la base de datos del POS pueden ser accesibles a través de SQL server.

Las respuestas del Coordinador de Soporte, Lenin Acosta, sugieren que la clave para una integración exitosa radica en garantizar que los datos de productos, almacenes y documentos fiscales estén sincronizados en todos los sistemas. Además, aborda que los desafíos podrían surgir al transmitir datos entre ambos sistemas, pero pueden superarse con un conocimiento profundo de la arquitectura de la base de datos y de la factura.

El beneficio esperado de esta integración, tal como se destacó en la entrevista, sería la automatización de procesos, una mayor facilidad para consultar la información almacenada en el POS, la prevención de fallos en la sincronización de datos, mayor seguridad al actualizar los precios, y la posibilidad de desarrollar informes personalizados.

Estos resultados proporcionan un camino claro hacia la implementación de la API para sincronizar el sistema de POS y el ERP, destacando la importancia de un enfoque estratégico y una planificación cuidadosa.

Cuadro 2: Entrevista sobre el ERP.

Pregunta	Respuesta
¿Podría describir brevemente los módulos más relevantes del sistema ERP de GUUAO Marketplace?	Para GUUAO MP, el ERP maneja los módulos de carga de documentos fiscales, existen diversos endpoint que poseen funciones para la creación de documentos través de métodos POST, adicionalmente en el ERP se maneja la mayoría de la información que se termina manejando en el POS.

<p>¿Cómo se puede acceder a los datos almacenados en el ERP para la posible sincronización con el POS?</p>	<p>Los diferentes módulos poseen endpoints para la consulta de datos a través de peticiones por HTTP, como las listas de precios y las existencias por almacenes, en el ERP se carga información no solo de GUUAO MP, también se manejan documentos de Sellers E-commerce de GUUAO y por esto ya existen varios endpoints para consultar e insertar datos en el ERP, además que el ERP está integrado con SAP a través de una API propia, por ende también se pueden consultar datos directamente desde SAP para la sincronización hacia el POS.</p>
<p>¿Qué datos generados por el POS son necesarios en el ERP para la sincronización entre ambos sistemas?</p>	<p>En el ERP solo necesita que se generen los documentos fiscales, sin necesidad registrar los pagos, ya que, al ser facturas de la tienda física, se asume que todo se paga de manera inmediata en el punto de venta, en caso de que haya un error con los pagos, el equipo de tienda se encarga de realizar las notas de crédito.</p>
<p>¿Cómo se maneja la creación de los documentos fiscales en el ERP?</p>	<p>Los documentos de tienda se generan a través un endpoint único, ya que, al no poseer referencia de pagos, es necesario tratarlos diferentes a los documentos de Sellers del E-commerce, para la creación del documento es necesario que se le pase a través de la Request POST datos del documento como las líneas, número del documento, serial fiscal y entre otros datos generados en el POS de tienda.</p>

<p>¿Existen otros módulos o funcionalidades que considera vitales para el sistema ERP de GUUAO Marketplace y por qué?</p>	<p>Principalmente, para lo que es Marketplace, considero que ya se abarcó lo más importante, tener en cuenta que para la sincronización de ambos sistemas es necesario consultar datos no solo del ERP, sino también de la API de SAP, ya que esto permitirá una sincronización más precisa al consultar datos de la fuente principal y ya que algunos de los endpoints del ERP, como el de consulta de productos, se reemplazó por el endpoint de la API SAP.</p>
<p>En su opinión, ¿cuáles son los elementos más importantes a considerar para la integración de ambos sistemas (POS y ERP)?</p>	<p>Que los datos de los documentos enviados estén correctos y bien estructurados para luego ser enviado al ERP, además tomar en cuenta que enviar muchas request con demasiados datos a corto plazo puede ocasionar problemas de sobrecarga en el sistema ERP.</p>
<p>¿Qué desafíos podrían surgir durante la integración de estos sistemas y cómo podrían ser superados?</p>	<p>Conocer la estructura y que datos se deben de enviar a través de los endpoints, ya que se necesita seguir una estructura específica en el JSON para que el ERP pueda crear correctamente el documento y estos datos enviados en el JSON deben de ser 100% correctos, ya que luego de crearse en el ERP estos documentos viajan a SAP, en el cual los documentos errados significarían un problema a nivel de contabilidad de la empresa.</p>
<p>¿Qué beneficios espera GUUAO Marketplace de la implementación de una API para la sincronización del sistema POS y ERP?</p>	<p>La automatización de los documentos fiscales, cada mitad y a finales de mes, es necesario que el departamento de contabilidad haga reportes sobre de la tienda MP, automatizar que estos</p>

	documentos viajen al ERP y que luego el ERP los integre a SAP facilitaría este proceso en gran medida, además se tendría mejor control de cualquier error en los documentos que pueda haberse generado.
--	---

Cuadro 2 Entrevista sobre el ERP.

Fuente: Elaboración propia (2023).

Después de entrevistar a Diego Soto, Coordinador de Sistemas en GUUAO Marketplace, quedó claro cuán importante es sincronizar su sistema ERP con su POS. Se identificaron los módulos y funcionalidades clave dentro del ERP que deben tenerse en cuenta al implementar una API para la sincronización, lo que es más importante, cargar documentos fiscales y administrar información dentro de su sistema POS. Se puede acceder a los datos actualmente almacenados dentro del ERP a través de endpoints fáciles de usar, donde se inician solicitudes HTTP para consultar precios o verificar niveles de inventario.

La sincronización de estos sistemas críticos se puede lograr, en última instancia, mediante la generación de documentos fiscales dentro de la infraestructura ERP existente, que luego pasará datos relevantes del documento a través de un único endpoint para optimizar los procesos entre ambos sistemas. Si bien esto suena relativamente sencillo en papel, es importante reconocer que la sincronización precisa requiere, no solo consultar la información disponible en los registros actuales, sino también la API de SAP para mayor precisión, abordando algunos de los desafíos más sutiles, como las inconsistencias contables que rodean los registros mal interpretados o las designaciones de entradas erróneas que hacen impuestos.

El siguiente apartado, representado por el Cuadro N°3, ofrece una evaluación minuciosa de la interacción entre los sistemas ERP y POS. El análisis presenta los resultados del estudio de las funcionalidades de los sistemas para su sincronización.

Cuadro 3: Funcionalidades claves del sistema POS y el ERP para su sincronización.

	Áreas Clave	Coordinador de Soporte: Lenin Acosta	Coordinador de Sistema: Diego Soto
	Sistema de Punto de Venta (POS)		

			Sistemas de planificación de recursos empresariales (ERP)
1	Funcionalidades críticas del sistema	<ul style="list-style-type: none"> - Emisión de documentos fiscales - Gestión de inventario - Administración de precios y almacenes - Generación de reportes de caja y procesamiento de pagos 	<ul style="list-style-type: none"> - Carga de documentos fiscales - Gestión de información relevante para el POS
2	Acceso a datos	<ul style="list-style-type: none"> - A través del sistema administrativo incluido en la licencia del software del POS. - Datos accesibles a través de la base de datos del POS, a la cual se puede acceder a través de SQL Server. 	<ul style="list-style-type: none"> - A través de endpoints para consultas de datos mediante peticiones HTTP - Consulta de datos directamente desde SAP
3	Integración de sistemas	<ul style="list-style-type: none"> - Sincronización de los siguientes datos: Productos (códigos de barra y producto, descripciones, precios, categorías, existencias) por almacenes. - Envío de los documentos fiscales a B2B 	<ul style="list-style-type: none"> - Generación de documentos fiscales en el ERP - Consideración de la sobrecarga del sistema ERP por envío de muchas solicitudes a corto plazo
4	Desafíos en la integración	<ul style="list-style-type: none"> - Conocimiento detallado de la arquitectura de la base de datos y los documentos enviados al ERP. 	<ul style="list-style-type: none"> - Estructuración correcta y precisión de los datos enviados a través de los endpoints - Conocimiento detallado de la estructura de los datos y su precisión
5	Beneficios de la integración	<ul style="list-style-type: none"> - Automatización de procesos. - Facilidad en la consulta de información. 	<ul style="list-style-type: none"> - Automatización de los documentos fiscales - Mejora en el control de errores en los documentos

		<ul style="list-style-type: none"> - Mayor seguridad en la actualización de precios. - Desarrollo de reportes personalizados 	<ul style="list-style-type: none"> - Facilidad en la generación de informes
--	--	--	--

Cuadro 3 Funcionalidades claves del sistema POS y el ERP

Fuente: Elaboración propia (2023).

Las conclusiones derivadas del anterior cuadro fueron cruciales para la dirección del siguiente objetivo, considerando que han permitido identificar áreas clave que necesitan atención y nos proporcionan la información necesaria para hacer ajustes oportunos.

4.2 Desarrollo de la estructura de una solución de integración óptima para la sincronización entre el sistema de POS y el ERP de la empresa GUUAO Marketplace, considerando las opciones disponibles y aplicando el enfoque seleccionado para la implementación del sistema.

Este objetivo consiste en estructurar una solución que permita sincronizar ambos sistemas, para luego proceder a su implementación. Para lograr este fin, se hizo uso de repositorios, documentación, análisis de la base de datos del sistema POS y los endpoints del ERP y de las técnicas de análisis de datos correspondientes para desarrollar la solución final.

El proceso inicial consistió en realizar una investigación profunda de los repositorios existentes y el material de documentación pertinente al desarrollo de APIs. Este material abarcaba desde documentación oficial de los lenguajes y tecnologías elegidos, hasta guías y tutoriales en línea, blogs técnicos y foros de discusión como Stack Overflow.

Se consideraron varias alternativas al elegir la tecnología adecuada para el desarrollo de la API. Estas opciones incluyeron diferentes lenguajes de programación y tecnologías como JavaScript, Python, Ruby y TypeScript, tomando en cuenta sus principales aspectos para el desarrollo back-end de una API.

Cuadro 4: Cuadro comparativo entre diferentes lenguajes de programación usados en el desarrollo de APIs

Lenguaje	Rendimiento	Compatibilidad con otras tecnologías	Bibliotecas y frameworks	Manejo de tareas
JavaScript	Moderado	Alta (ampliamente usado en el lado del cliente y del servidor). Tipado de datos dinámico	Es uno de los lenguajes con mayor cantidad de bibliotecas y frameworks disponibles.	Un solo hilo de procesos, asíncrono, basado en eventos (varias tareas, una a la vez)
TypeScript	Moderado	Alta (compatible con todas las bibliotecas JavaScript). Tipado de datos estático	Hereda la amplia gama de bibliotecas y frameworks de JavaScript	Un solo hilo de procesos, asíncrono, basado en eventos (varias tareas, una a la vez)
Python	Puede ser lento para tareas de alta intensidad de CPU.	Alta (se utiliza en una variedad de aplicaciones, desde el desarrollo web hasta la ciencia de datos). Tipado de datos dinámico	Amplia gama de bibliotecas para el manejo de servidores	Soporte nativo para múltiples hilos, pero con ciertas limitaciones.
Ruby	Puede ser lento para tareas de alta intensidad de CPU	Moderada (ampliamente utilizado en el back-end, pero no tanto en otras áreas). Tipado de datos dinámico	Frameworks fuertes para el desarrollo back-end (Ruby on Rails)	Soporte para paralelismo, pero puede ser más complejo de manejar que en otros lenguajes

Cuadro 4 Comparación entre diferentes lenguajes de programación usados en el desarrollo de APIs

Fuente: Elaboración propia (2023).

En este contexto, se decidió utilizar TypeScript sobre los demás lenguajes, basándose en los siguientes aspectos: el tipado en TypeScript es estático, esto significa que los tipos datos se definen antes de iniciar el programa, lo cual ayuda a prevenir muchos errores comunes en JavaScript, al comprobar los tipos durante la compilación y proporcionar un mejor autocompletado y asistencia de código en los editores modernos. Adicionalmente, los endpoints para la creación de documentos fiscales del ERP reciben datos con un formato específico, para estos casos, dicho tipado permite el uso de interfaces, las cuales se utilizan para definir una estructura específica que los objetos, clases o variables deben seguir.

Adicionalmente, TypeScript es un superconjunto de JavaScript, lo que significa que el código JavaScript es también código TypeScript válido. Esto permite la capacidad de utilizar todas las bibliotecas y características de JavaScript existentes. Además, aunque TypeScript puede tener una curva de aprendizaje inicial más empinada que JavaScript debido a su sistema de tipado, este esfuerzo adicional se ve compensado con un mantenimiento más fácil y menos errores en el código en el futuro. Esto es especialmente beneficioso para proyectos más grandes y complejos como el desarrollo de una API.

De manera similar, se procedió a seleccionar la arquitectura que tendría la API. En el diseño de una API, la elección de su arquitectura es un elemento fundamental, pues esta define cómo serán las interacciones entre los componentes del sistema, cómo se organizarán los datos y cuál será el formato de las peticiones y las respuestas.

Asimismo, entre las diversas arquitecturas disponibles para el desarrollo de APIs, se consideraron principalmente tres opciones: la arquitectura basada en SOAP (Simple Object Access Protocol), la arquitectura basada en GraphQL y la arquitectura REST (Representational State Transfer). A continuación, se presenta un cuadro comparativo entre estas tres alternativas, resaltando las principales características de cada una.

Cuadro 5: Cuadro comparativo entre las diferentes arquitecturas de APIs.

Arquitectura	Ventajas	Desventajas	Casos de Uso
SOAP	Protocolo estándar con soporte amplio. Ofrece alta seguridad, robustez	Más complejo y pesado. Requiere más recursos de red. Principalmente	Ideal para entornos empresariales donde se requiere un alto nivel de

	y fiabilidad. Permite la comunicación entre aplicaciones sobre cualquier protocolo de transporte (HTTP, SMTP, TCP, etc.).	limitado a XML, lo que puede ser menos eficiente y más difícil de trabajar que JSON.	seguridad y transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
GraphQL	Permite al cliente especificar exactamente lo que necesita, evitando así datos innecesarios. Facilita la consulta de datos eficiente y ofrece un rendimiento óptimo.	Más complejo de implementar y mantener. No es tan maduro o ampliamente soportado como REST o SOAP.	Ideal para aplicaciones donde los usuarios necesitan consultar datos de manera eficiente y dinámica. Muy útil para aplicaciones con datos profundamente anidados.
REST	Simple de entender y usar. Soporte amplio en muchos lenguajes de programación. Flexibilidad en los formatos de datos (JSON, XML, etc.). Adherencia a los métodos estándar de HTTP (GET, POST, PUT, DELETE, etc.).	No es tan eficiente en la consulta de datos como GraphQL. No tiene soporte nativo para las transacciones en tiempo real como WebSockets.	Ideal para aplicaciones web públicas debido a su simplicidad, escalabilidad y rendimiento. Bueno para aplicaciones móviles debido a su ligereza y al soporte de JSON.

Cuadro 5 Comparación entre las diferentes arquitecturas de APIs.

Fuente: Elaboración propia (2023).

En este contexto, la elección fue la arquitectura REST, debido a su simplicidad y al amplio soporte que tiene en diversos lenguajes de programación, incluyendo TypeScript. Además, REST ofrece una gran flexibilidad en cuanto al formato de los datos, permitiendo el uso de JSON, que es un formato de datos muy común y fácil de manejar en JavaScript y TypeScript.

Asimismo, a pesar de que GraphQL puede proporcionar una mayor eficiencia en la consulta de datos, para el caso específico de la integración entre el sistema de POS y el ERP, las características de REST se adaptan mejor a los requerimientos del proyecto, proporcionando una solución simple y eficaz.

Finalmente, una vez seleccionada la arquitectura REST para el desarrollo de la API, se decidió basar el proyecto en una API con una estructura de carpeta, la cual maneja la siguiente distribución:

Figura 1: Diagrama de estructura de los archivos de una API REST desarrollada en TypeScript

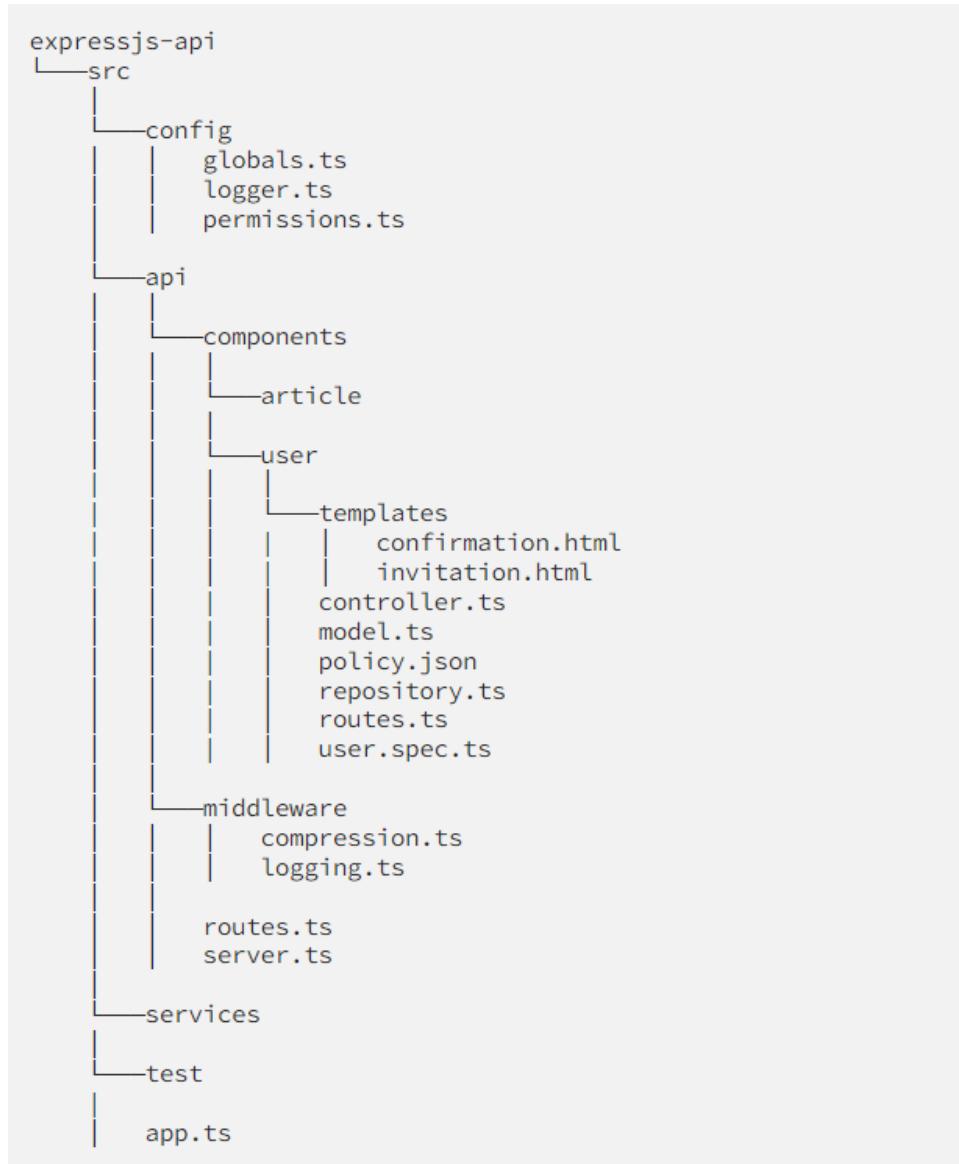


Figura 1 Diagrama de estructura de los archivos de una API REST desarrollada en TypeScript

Fuente: Wächter, L (2020).

El anterior diagrama pertenece a una API REST desarrollada con TypeScript, Node.js y Express.js, y sirvió de guía para el desarrollo de la estructura de carpetas de la presente API. Se tomaron aspectos importantes de la estructura de la API REST del diagrama anteriormente expuesto, como el uso de registradores, conocidos como loggers, para documentar y mantener información valiosa a lo largo del tiempo.

Asimismo, el uso de modelo y controladores presentes en el diagrama es una práctica utilizada en el patrón de diseño Modelo, Vista, Controlador o MVC. De acuerdo con lo establecido por la organización Mozilla (2022), en su sitio web, dicho modelo se define como:

Un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización. Esta "separación de preocupaciones" proporciona una mejor división del trabajo y una mejora de mantenimiento.

Figura 2: Diagrama UML sobre el MVC

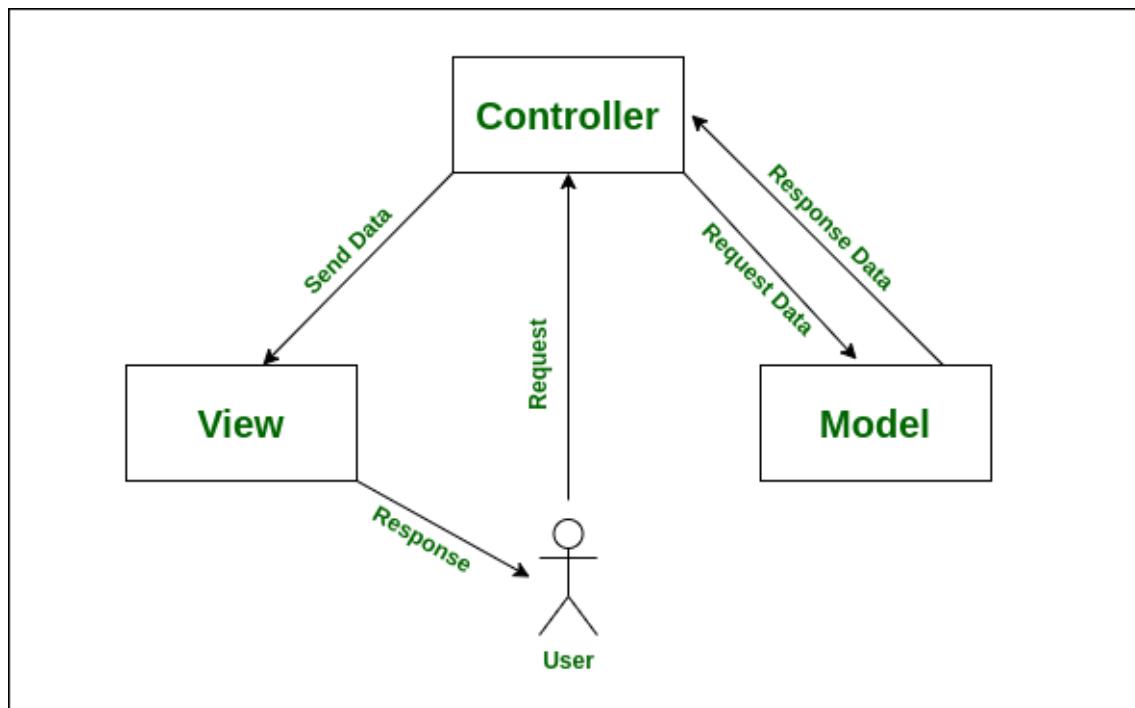


Figura 2 Diagrama UML sobre el MVC

Fuente: Anushka, V. (s.f.).

En el caso del desarrollo de la API, para la sincronización entre el POS y el ERP no se hizo uso de Vistas; únicamente se trabajó con la lógica y el patrón entre Modelo para la base de datos y Controlador para manejar las acciones del Modelo.

Asimismo, la sincronización de los sistemas requirió de establecer conexión con bases de para el manejo de la misma, en el caso de la API desarrollada la conexión fue diseñada específicamente para interactuar con la base de datos del software de punto de venta (POS). Esta base de datos, que ya está estructurada gracias a su origen de una solución de software de terceros, utiliza SQL Server como su gestor de base de datos. Por lo tanto, para el desarrollo de la API, se hizo uso de la librería de que permitiera interactuar con la base de datos del POS a través de SQL Server. Entre las opciones consideradas se encontraban Tediou, Sequelize, y la librería MSSQL de Node.js. Cada una de estas opciones tenía sus propias ventajas y desventajas, como se refleja en el siguiente cuadro comparativo:

Cuadro 6: Cuadro comparativo entre librerías para la conexión con la Base de Datos.

Opción	Facilidad de uso	Características Distinguibles
Tediou	Requiere un mayor conocimiento de SQL	Ofrece un control completo sobre cómo se estructuran y ejecutan las consultas
Sequelize	Fácil de usar con comandos de alto nivel	Permite manejar y organizar datos de manera más eficiente, facilita la actualización de la base de datos
MSSQL	Interfaz sencilla y respuestas directas	Mejora la eficiencia de la conexión a la base de datos, soporta transacciones (múltiples operaciones como un grupo) y consultas en tiempo real

Cuadro 6 Comparación entre librerías para la conexión con la Base de Datos.

Fuente: Elaboración propia (2023).

En contexto de las diversas opciones disponibles, se seleccionó la librería MSSQL como la opción óptima para la implementación de la API. Esta decisión se fundamentó, tanto en factores técnicos, como en aspectos de coherencia tecnológica y compatibilidad con el sistema de POS en uso. Esta elección se alinea con las tecnologías previamente seleccionadas y estuvo respaldada por una amplia documentación y una comunidad activa de desarrolladores, asegurando de esta manera el progreso efectivo y consistente del proyecto.

Asimismo, según NPM en la versión del software MSSQL (Versión 7.2.1), para establecer la conexión con la base de datos se requiere, obligatoriamente, de los siguientes campos: usuario, contraseña, instancia del servidor del gestor de base datos y dirección del host donde se encuentre. Adicionalmente, la documentación incluye cómo realizar consultas usando la librería MSSQL.

Figura 3: Ejemplo de conexión a una base de datos y consulta usando MSSQL.

```
const sql = require('mssql')
const sqlConfig = {
    user: process.env.DB_USER,
    password: process.env.DB_PWD,
    database: process.env.DB_NAME,
    server: 'localhost',
    pool: {
        max: 10,
        min: 0,
        idleTimeoutMillis: 30000
    },
    options: {
        encrypt: true, // for azure
        trustServerCertificate: false // change to true for local dev / self-signed c
    }
}

async () => {
    try {
        // make sure that any items are correctly URL encoded in the connection string
        await sql.connect(sqlConfig)
        const result = await sql.query`select * from mytable where id = ${value}`
        console.dir(result)
    } catch (err) {
        // ... error checks
    }
}
```

Figura 3 Ejemplo de conexión a una base de datos y consulta usando MSSQL.

Fuente: NPM (2023). MSSQL (Versión 7.2.1) [Software].

El anterior ejemplo muestra cómo establecer la conexión con la base de datos usando las credenciales ya mencionadas y se hace uso de técnicas para el manejo de las credenciales mediante la librería DotEnv, ya que, exponer las credenciales mediante el código de la API, supone un alto

riesgo de violaciones de seguridad y potenciales daños irreparables en caso de un incidente de seguridad. Por lo tanto, es primordial encontrar un método que permita almacenar esta información de forma segura y eficiente.

En este contexto, se consideró el uso de variables de entorno, una técnica comúnmente adoptada en el desarrollo de software para administrar la configuración de aplicaciones mediante el uso de archivos .env, al respecto Moreno, K. (2022), en su blog, plantea que “los archivos .env contienen credenciales en formato clave-valor para los servicios utilizados por el programa que están creando. Están destinados a almacenarse localmente y no cargarse a repositorios de códigos en línea para que todos los lean”. Estas son ideales para este propósito, ya que permiten que la configuración sea cambiada sin modificar el código y aíslan los datos confidenciales del código en sí.

Por último, el manejo de credenciales se utilizó en otros aspectos como las llaves necesarias para la comunicación con el ERP, los dominios base de las API de SAP y el ERP, ambiente de desarrollo (productivo o desarrollo) y otras configuraciones ajustables en el comportamiento de la API.

Figura 4: Ejemplo de cómo hacer uso de la librería DotEnv

```
npm install dotenv
```

Importe el módulo en la parte superior del script de inicio para su base de código:

```
require('dotenv').config()
```

Eso es todo, ahora puede acceder a los secretos en cualquier lugar de su base de código:

```
// display the value of SECRET_1 into your code
console.log(process.env.SECRET_1)
// -> 924a137562fc4833be60250e8d7c1568

// display the value of SECRET_2 into your code
console.log(process.env.SECRET_2)
// -> cb5000d27c3047e59350cc751ec3f0c6
```

Figura 4 Ejemplo de cómo hacer uso de la librería DotEnv

Fuente: Moreno, K. (2022).

Para finalizar, dicho algoritmo fue trabajado bajo un entorno de desarrollo integrado llamado “Visual Studio Code”, el cual posee múltiples ventajas debido al soporte que le da a una extensa variedad de lenguajes de programación y a sus extensiones, las cuales permiten personalizar y mejorar la funcionalidad con la que se trabaja. Además, se utilizó el controlador de versiones Git para versionar el desarrollo de la API, a través de la plataforma gitlab.

Cuadro 7: Resumen de la información recopilada

Etapa	Herramientas	Descripción
Lenguaje de Programación	TypeScript, JavaScript	TypeScript es un superconjunto de JavaScript, que fue elegido por sus diversas herramientas para el desarrollo de API mediante el uso de protocolos HTTP.
Entorno de Ejecución	Node.js, Express.js	La API se desarrolló haciendo uso de Node.js y Express.js.
Estructura del Proyecto	API REST	Basándose en los repositorios existentes y la documentación relevante, se decidió implementar una estructura de proyecto basada en una API REST.
Base de Datos	SQL Server, Librería Node.js MSSQL	La conexión con la base de datos se realiza a través de la librería Node.js MSSQL, permitiendo la interacción con la base de datos del POS en SQL Server.
Patrón de Diseño	Modelo, Controlador (Basado en MVC)	Se optó por el patrón de diseño MVC, con énfasis en el Modelo y el Controlador, ya que no se hizo uso de Vistas. Este patrón permite una mejor separación entre la lógica de negocios y su visualización.
Entorno de Desarrollo	Visual Studio Code	El desarrollo del algoritmo se realizó en Visual Studio Code, que ofrece soporte para una amplia variedad de lenguajes de programación y permite la personalización y mejora de la funcionalidad mediante extensiones.

Control de Versiones	Git, Gitlab	Para el control de versiones, se utilizó Git a través de la plataforma Gitlab.
-----------------------------	-------------	--

Cuadro 7 Resumen de la información recopilada

Fuente: Elaboración propia (2023)

El desarrollo de la API se llevó a cabo en 2 partes principales: El desarrollo de los sincronizadores para la base de datos del punto de ventas y las facturas del ERP y las rutas de consultas para el manejo de los reportes y datos, terminando con una estructura en las carpetas de proyecto de la siguiente manera:

Figura 5: Estructura de carpetas de la API.

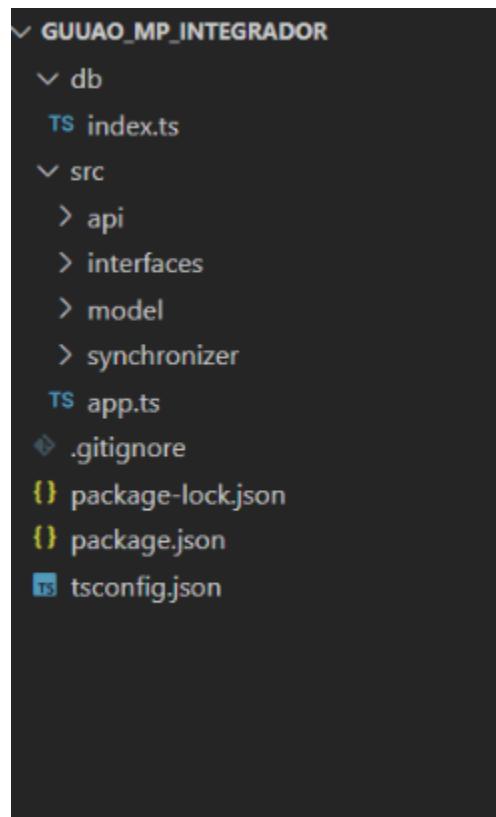


Figura 5 Estructura de carpetas de la API.

Fuente: Elaboración propia (2023).

Los mencionados sincronizadores se encargan de manejar todos los métodos necesarios para la integración de los datos en los sistemas, y poseen la siguiente estructura:

Figura 6: Diagrama de clases

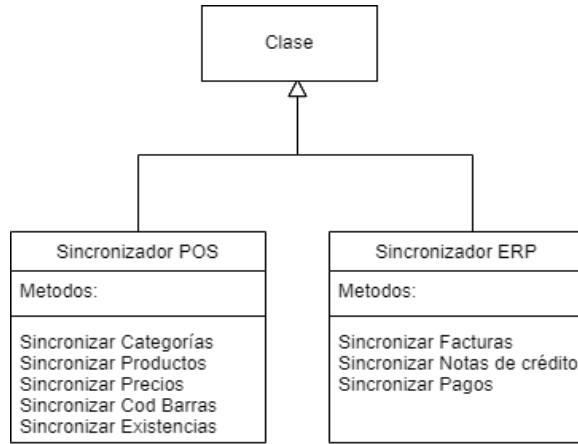


Figura 6 Diagrama de clases sincronizadores

Fuente: Elaboración propia (2023).

Figura 7: Estructura de los sincronizadores.

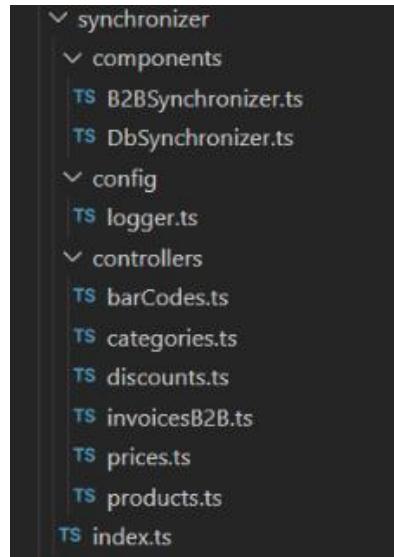


Figura 7 Estructura de los sincronizadores.

Fuente: Elaboración propia (2023).

De manera similar, los controladores de las rutas de la API fueron conformados por dos clases principales, Controlador de Productos y el Controlador de Documentos, los cuales se encargan de manejar toda la información almacenada en la base de datos del POS para enviarla como respuesta a consultas GET realizadas a las rutas, y poseen la siguiente estructura:

Figura 8: Diagrama de clases

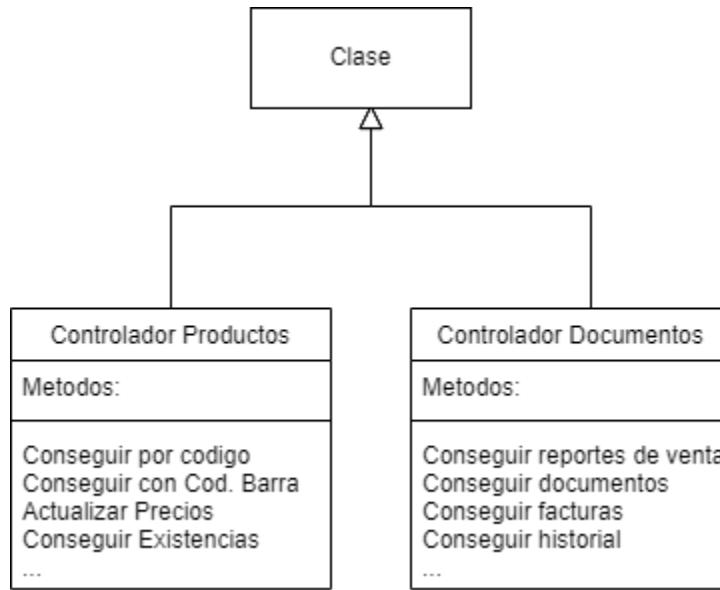


Figura 8 Diagrama de clases controladores

Fuente: Elaboración propia (2023).

Figura 9: Estructura archivos de las rutas.

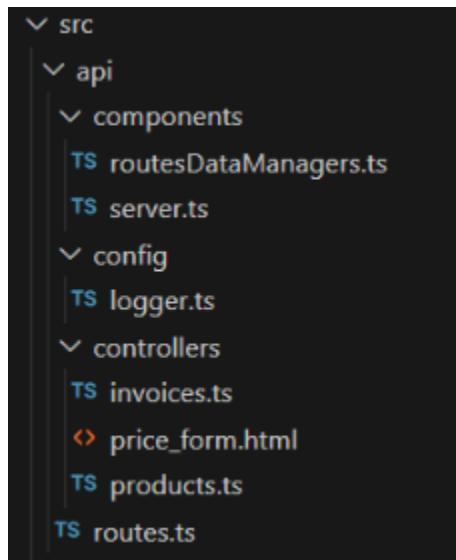


Figura 9 Estructura archivos de las rutas.

Fuente: Elaboración propia (2023)

Para llevar a cabo la sincronización de los documentos fiscales al ERP se hizo uso de la clase Sincronizador ERP, la cual se encarga de enviar todos los datos en forma de JSON mediante métodos HTTP POST a los endpoint para su creación. Asimismo, se desarrollaron interfaces de datos para las estructuras requeridas en el JSON, las cuales incluyen las siguientes estructuras:

Las facturas enviadas al endpoint: api/v1/oasis/invoice/create

Figura 10: Diagrama JSON de las facturas:

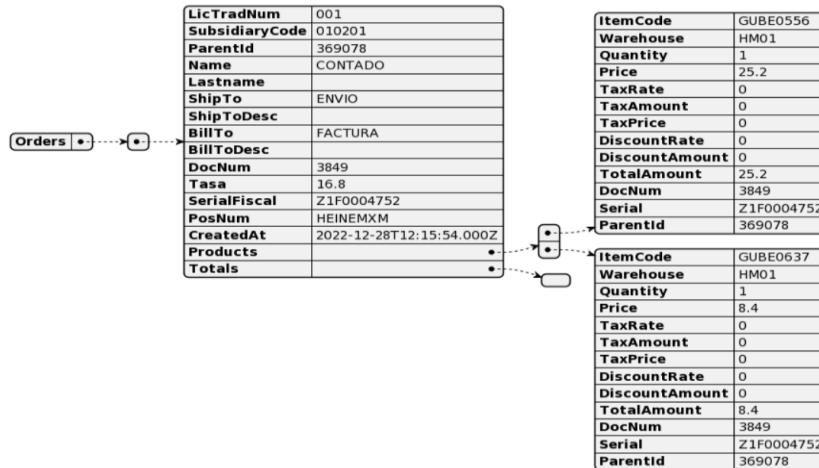


Figura 10 Diagrama JSON de las facturas

Fuente: Elaboración propia (2023).

Las notas de créditos al endpoint: api/v1/oasis/creditmemo/create_partial

Figura 11: Diagrama JSON de las notas de crédito:

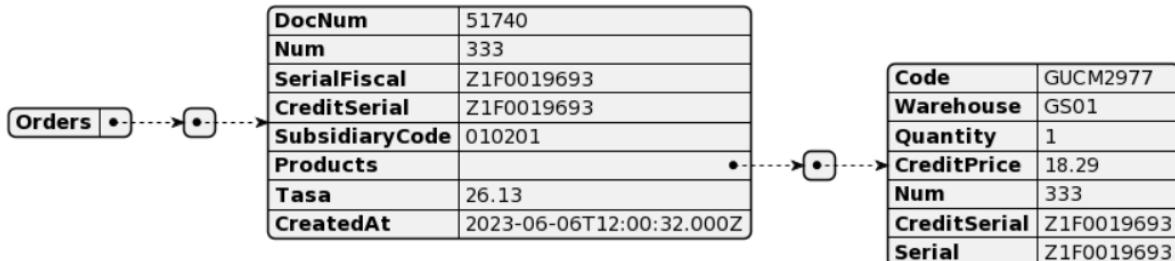


Figura 11 Diagrama JSON de las notas de crédito

Fuente: Elaboración propia (2023).

Los pagos al endpoint: api/v2/oasis/pay/add

Figura 12: Diagrama JSON de los pagos:

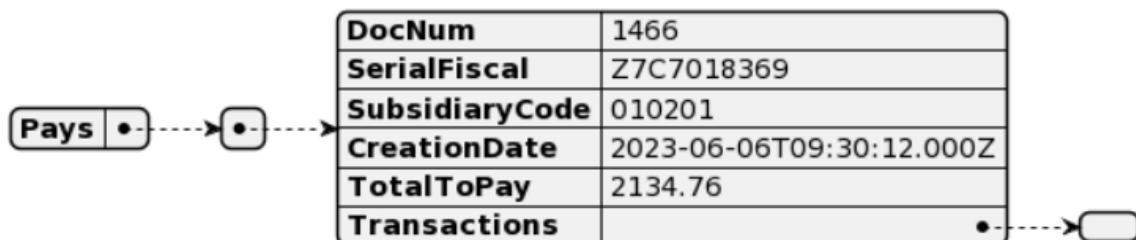


Figura 12 Diagrama JSON de los pagos

Fuente: Elaboración propia (2023).

De igual manera, para la sincronización de la base de datos del punto de venta se requirió de datos consultados mediante métodos HTTP GET a los endpoints en el ERP y la API SAP, como fue mencionado en las entrevistas. Asimismo, se desarrolló la estructura de las interfaces de datos necesarias para la sincronización con la base de datos del punto de ventas:

Los productos mediante endpoint: /oasis/Productos

Figura 13: Estructura JSON de los productos

cod_interno	GUBE1296
txt_descripcion_larga	WHISKY JOHNNIE WALKER BLACK LABEL MINI ICON 3.0 700ML
txt_descripcion_corta	WHISKY JOHNNIE WALKER BLACK LABEL MINI ICON 3.0 700ML
cod_departamento	GUBE
cod_familia	101
Ind_pesado	0
cod_impuesto	1
Clase	0
tipo_producto	1
cod_marca	1
txt_marca	DIAGEO

Figura 13 Estructura JSON de los productos

Figura 14: Estructura JSON de existencias

Las existencias mediante endpoint /api/v1/oasis/existencias/GS01

ItemCode	GUTH2399
WhsCode	GS01
cod_familia	114
cod_departamento	GUTH
OnHand	0
Compromised	0
Available	0

Figura 14 Estructura JSON de las existencias

Fuente: Elaboración propia (2023).

Los precios mediante el endpoint /api/v1/oasis/saint/precios/0/010201-1

Figura 15: Estructura JSON de un precio

cod	GUCB0096
p1	5371.38
p2	0
p3	4630.5
pUsd	175

Figura 15 Estructura JSON de los precios

Fuente: Elaboración propia (2023).

Los códigos mediante el endpoint /oasis/Codigos

Figura 16: Estructura JSON de los codigos

ItemCode	GUAU0001
BcdCode	5055148211813
cod_familia	104
cod_departamento	GUAU

Figura 16 Estructura JSON de los codigos

Fuente: Elaboración propia (2023).

Por último, mediante el endpoint /oasis/Instancias

Figura 17: Estructura JSON de las categorías

cod	GUAM00000
code	GUAM102
name	Consumo Masivo

Figura 17 Estructura JSON de las categorías

Fuente: Elaboración propia (2023).

Asimismo, una vez identificadas las estructuras de los datos, tanto para el ERP como para la base de datos del POS, se seleccionó un método para la transmisión de datos y sincronización entre sistemas de manera automática. Las opciones consideradas fueron WebSockets y la programación de tareas recurrentes con Cron. Las siguientes son los aspectos más importantes de ambas opciones:

Cuadro 8: Cuadro comparativo WebSockets y tareas Cron.

Aspecto	WebSockets	Cron
Comunicación	Bidireccional en tiempo real.	Unidireccional, en intervalos de tiempo predefinidos.

Complejidad	Mayor, requiere de una implementación más compleja.	Menor, implementación sencilla y directa.
Recursos	Mayor uso de recursos, ya que mantiene una conexión abierta.	Menor uso de recursos, sólo se usa cuando se ejecuta la tarea.
Escalabilidad	Escalabilidad limitada, ya que cada conexión consume recursos.	Mayor escalabilidad, debido a su naturaleza "sin estado".
Uso	Ideal para actualizaciones en tiempo real y aplicaciones de chat.	Ideal para tareas recurrentes y operaciones de mantenimiento.

Cuadro 8 Cuadro comparativo WebSockets y tareas Cron.

Fuente: Elaboración propia (2023).

Con base a estas características, se decidió optar por la opción Cron. Aunque los WebSockets proporcionan comunicación en tiempo real, su implementación y gestión de recursos son más complejas y menos escalables en comparación con Cron. Además, para la finalidad de este proyecto, la naturaleza de intervalos de tiempo predefinidos de Cron es más apropiada, ya que se necesita sincronizar los sistemas de manera regular y no en tiempo real, los métodos sincronizadores se pueden ejecutar en intervalos desde tres (3) a quince (15) minutos según el tipo de método sincronizador.

Con base a la metodología de sincronización, se estructuró la imagen del funcionamiento del servicio a niveles generales, lo cual puede ser representado de la siguiente manera:

Figura 18: Diagrama de la sincronización entre los sistemas

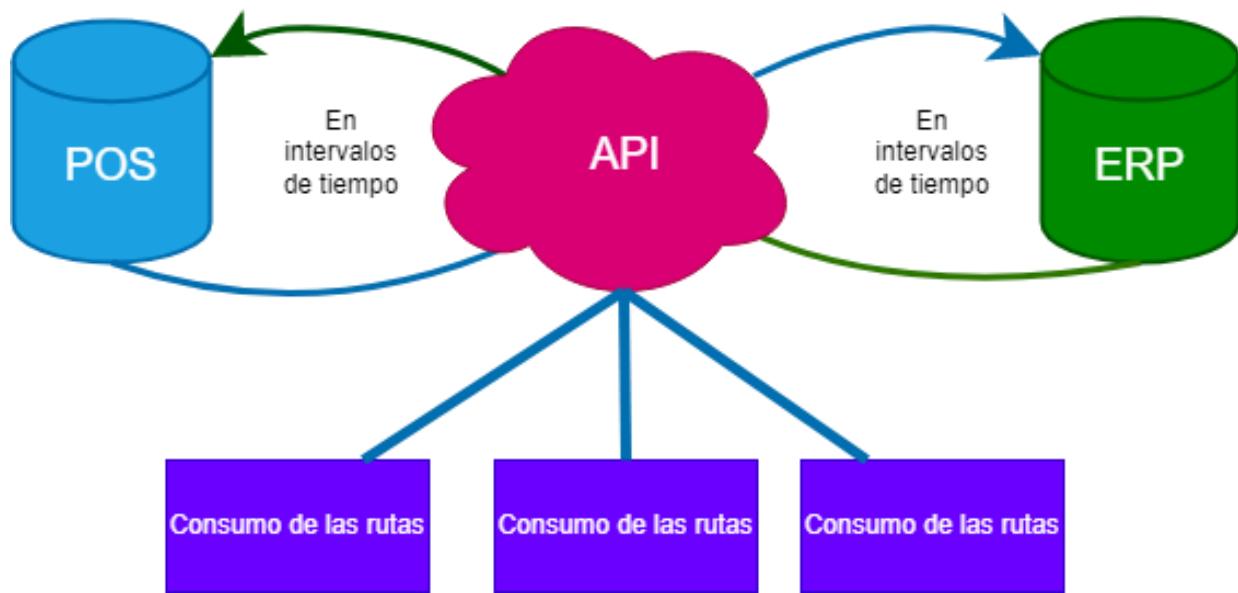


Figura 18 Diagrama de la sincronización entre los sistemas

Fuente: Elaboración propia (2023).

Asimismo, se desarrolló la estructura necesaria del proyecto, con buenas prácticas y el uso de metodologías profesionales como el manejo de credenciales mediante DotEnv y manejo de versiones con Git. Esto permite una estructura sólida en el proyecto, que facilita la colaboración de individuos externos que deseen contribuir o consumir la API, así como la incorporación de nuevos empleados, ya sean operadores o desarrolladores de la API., también nuevos empleados como operadores o desarrolladores de la API. Por lo mismo se desarrolló una documentación de la API que puede ser usada como manual para su consumo, además de un registro de logs que se guardan hasta una determinada cantidad de días establecida en la configuración .env de la API, de manera que facilite el debug de la misma.

4.3 Determinación del rendimiento del servicio API en términos de mejora de la eficiencia de la empresa GUUAO Marketplace y de la capacidad de adaptación a las cambiantes demandas del mercado.

En el curso de la implementación de la API en GUUAO Marketplace, se emplearon diversas herramientas y técnicas de evaluación para analizar su efectividad. Se realizaron meticolosas listas

de verificación, seguimiento en tiempo real y revisiones continuas de los procedimientos. Durante este proceso, se constató que GUUAO Marketplace tiene su propia página local, que se ha beneficiado enormemente de la integración con la API. Este vínculo ha generado una sinergia entre los sistemas, impulsando la eficiencia operacional y facilitando la consulta de información de la tienda por otros sistemas como el ERP.

Adicionalmente, se observó un aceleramiento significativo en la gestión de inventario y precios, gracias a la sincronización de los sistemas habilitada por la API. Esta mejora ha permitido a la empresa responder con prontitud a las variaciones del mercado, manteniendo su operatividad interna optimizada. Asimismo, es importante resaltar que la implementación de la API ha demostrado ser flexible para la integración con diversos sistemas y adaptable a sus posibles cambios. Esta característica pone de manifiesto la versatilidad de la API, así como su capacidad para escalar y adaptarse a las necesidades futuras de la empresa.

Al implementarse en el servidor, la API funciona de manera continua. Esto garantiza una operatividad constante, facilitando la gestión de operaciones y manteniendo la continuidad del negocio, por otro lado, esto significa una dependencia en el funcionamiento correcto del servidor.

De dicha manera, se determinó si la implementación ha llevado a una mejora en la eficiencia operativa, así como a una mayor adaptabilidad a las demandas cambiantes del mercado. Por lo tanto, es crucial evaluar su desempeño frente a la estrategia comercial y así poder identificar áreas que requieren mejoras u optimización para una ejecución perfecta.

Cuadro 9: Matriz FODA

Aspecto	Descripción
Fortalezas	<ul style="list-style-type: none">- Mayor eficiencia en la sincronización de los sistemas de POS y ERP.- Mejora en la precisión de los datos compartidos entre los sistemas.- Automatización de procesos- Independiente una vez implementado
Oportunidades	<ul style="list-style-type: none">- Capacidad de adaptación a las cambiantes demandas del mercado gracias a la flexibilidad modular del sistema.

	<ul style="list-style-type: none"> - Posibilidad de expansión a otras áreas del negocio que pueden beneficiarse de la sincronización. - Integración con otras tecnologías y plataformas como parte de un sistema más grande. - Posibilidad de obtener reportes personalizados de información directa del POS mediante la API.
Debilidades	<ul style="list-style-type: none"> - Dependencia de los servidores para el funcionamiento de la API. - Posibles dificultades en la adaptación del personal a los cambios en los sistemas. - Necesidad de mantenimiento y actualización continua para garantizar el correcto funcionamiento.
Amenazas	<ul style="list-style-type: none"> - Cambios en las normativas o en el entorno tecnológico que puedan afectar al funcionamiento de la API. - Posibles problemas de seguridad relacionados con la conexión y la sincronización de los sistemas. - Falta de personas capacitadas para el manejo del sistema.

Cuadro 9 Matriz FODA

Fuente: Elaboración propia (2023).

La implementación de la API en GUUAO Marketplace ha demostrado ser una medida estratégica que ha aumentado la eficiencia y la capacidad de adaptación a las cambiantes demandas del mercado. Aunque existen desafíos, principalmente relacionados con la dependencia de los servidores y la necesidad de un mantenimiento continuo, los beneficios evidentes superan a las debilidades identificadas.

PARTE V

LA PROPUESTA

5.1 Importancia de la aplicación de la propuesta.

La propuesta de integración entre el POS y el ERP de GUUAO Marketplace, es un motor potente para su escalabilidad y eficiencia operativa. Facilita la gestión de enormes volúmenes de datos, unifica la información en el sistema estándar del ERP y abre una puerta para futuras integraciones y expansiones a otras sucursales que usen el sistema de POS. Es, sin duda, un paso significativo para robustecer y optimizar el manejo de información, permitiendo a la empresa operar a mayor escala y adaptarse eficazmente a nuevas demandas y oportunidades.

5.2 Viabilidad de la propuesta.

A continuación, se presenta la factibilidad para la empresa, en los aspectos técnicos de equipos necesarios, operativos de personal capacitado y, finalmente, la viabilidad económica de la propuesta.

5.2.1 Viabilidad técnica.

La infraestructura de TI existente admite fácilmente la implementación, ya que GUUAO Marketplace posee un servidor remoto que cumple con los requisitos técnicos necesarios para hacer posible la implementación, manteniendo las complejidades técnicas al mínimo. Además, la infraestructura es lo suficientemente sólida para soportar la metodología modular y escalable de la solución y a futuros cambios en las necesidades de la empresa.

Cuadro 10: Especificaciones técnicas del servidor remoto

Especificaciones técnicas	
Memoria RAM	Memoria Ram Ddr3 32gb 1600mhz x 4 (128gb)
Procesador	Intel Xeon X5560 @ 2.80GHz 2.80GHz x 2
Almacenamiento	2TB Total en un rack de 5 discos duros HDD de 400gb

Servicios	Servicio de internet con IP publica
-----------	-------------------------------------

Cuadro 10 Especificaciones técnicas del servidor remoto

Fuente: Elaboración propia (2023).

5.2.2 Viabilidad operativa.

En términos de viabilidad operativa, la propuesta presentada resulta altamente factible y eficiente, al ser un servicio que opera mediante un servidor, por lo cual, una vez desarrollada e implementada, podría funcionar indefinidamente. Asimismo, el mantenimiento y operación de la solución planteada solo requiere de un individuo con habilidades en desarrollo, encargado de administrar, supervisar y, de ser necesario, actualizar o modificar la API. Adicionalmente, el manejo del servidor, que incluye su mantenimiento, puede ser llevado a cabo por un único operador.

Cuadro 11: Especificaciones operativas

Especificaciones operativas
Personal a cargo del desarrollo y mantenimiento de la API (Desarrollador junior)
Personal a cargo del mantenimiento del servidor

Cuadro 11 Especificaciones operativas

Fuente: Elaboración propia (2023).

5.2.3 Viabilidad económica.

Desarrollar esta API es una inversión inteligente que generará retornos sustanciales con el tiempo, desde una perspectiva financiera. El análisis de costo-beneficio demuestra que la justificación de los gastos asociados con su implementación supera cualquier impacto negativo que pueda producir. Los recursos involucrados serían el tiempo de desarrollo, los de servicios de internet y mantenimiento del servidor y la única inversión única realizada en el servidor.

Cuadro 12: Inversión única

Descripción	Costo \$
Memoria RAM (4x DDR3 32GB 1600MHz)	400

Procesador (2x Intel Xeon X5560 @ 2.80GHz 2.80GHz)	300
Almacenamiento (5x 400GB HDD)	500
TOTAL	1200

Cuadro 12 Costos de la inversión única del servidor

Fuente: Elaboración propia (2023).

Cuadro 12: Requisitos económicos

Descripción	Periodo	Costo \$
Servicio internet con ip pública	Mensual	210
Personal a cargo de mantenimiento del servidor	Mensual	150
Personal a cargo del desarrollo y mantenimiento de la API (Desarrollador junior)	Mensual	150
TOTAL	MENSUAL	510
	ANUAL	6.120

Tabla 13 Requisitos económicos

Fuente: Elaboración propia (2023).

5.3 Objetivos de la propuesta.

5.3.1 Objetivos General.

Implementar un servicio API para la sincronización del POS y el ERP de la empresa GUUAO Marketplace

5.3.2 Objetivos Específicos.

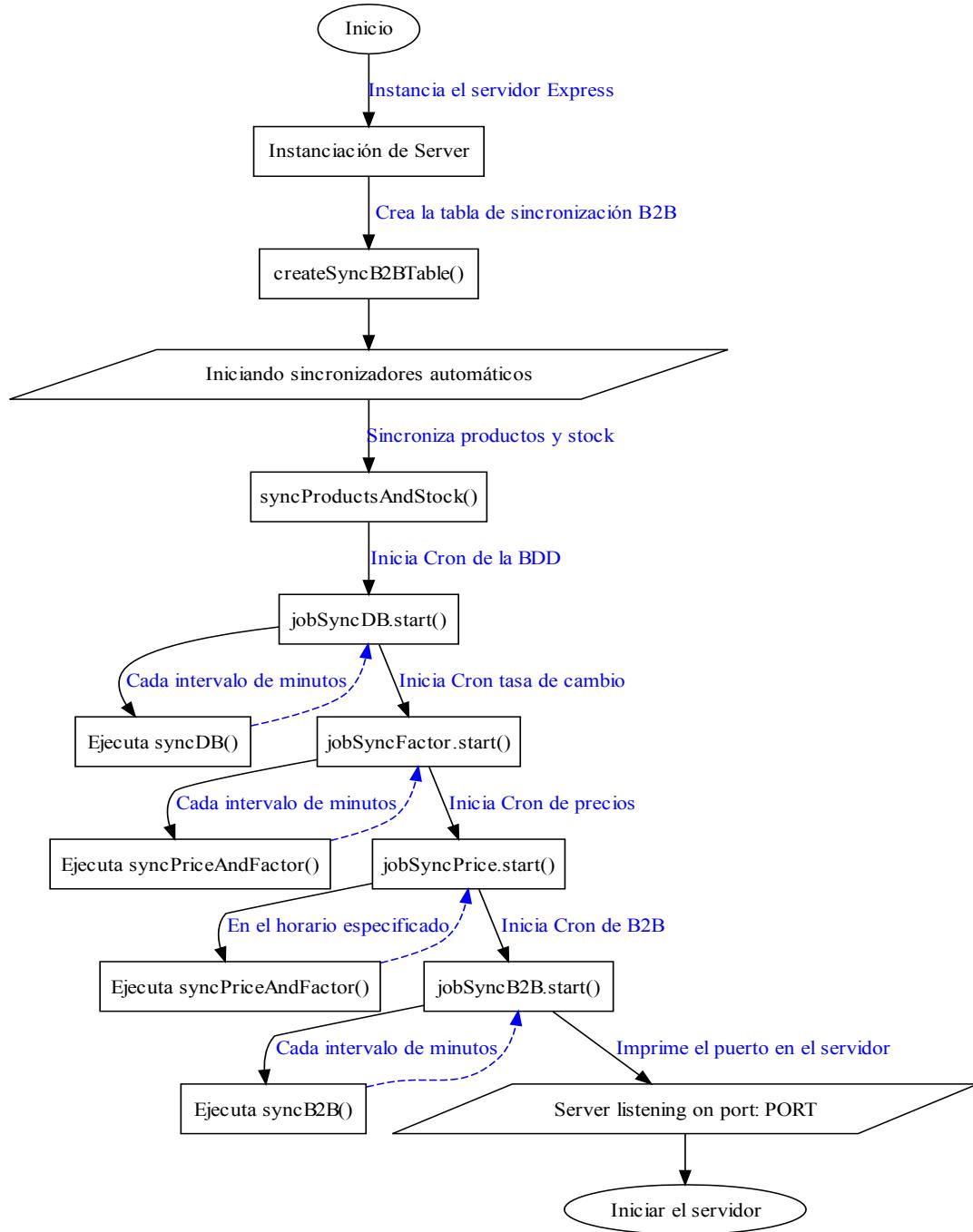
- Realizar el desarrollo de los sincronizadores automáticos para el ERP y el POS.

- Realizar el desarrollo de las rutas de la API.
- Implementar el servicio API en los servidores de IT de la empresa, encendiendo el servicio a través de accesos remotos al servidor y configurando su inicio automático.

5.4 Representación gráfica y estructurada de la propuesta

5.3.1 Funcionalidades y mensajes por consola de la API

Figura 19: Diagrama de flujo inicio de la API



Fuente: Elaboración propia (2023).

Ilustración 19 Diagrama de flujo inicio de la API

La anterior figura pertenece al diagrama de flujo del inicio del servicio API donde se puede apreciar los mensajes iniciales del mismo, las primeras instanciaciones de inventario y el inicio de las tareas Cron que se ejecutarán los métodos sincronizadores en intervalos definidos por la configuración del programa.

Figura 20: Mensajes por consola del inicio del programa

```
2023-06-20 14:04:27 info: [Main] - Servicio IntegratorSaint iniciado
2023-06-20 14:04:27 info: [Main] - Sincronizando productos...
2023-06-20 14:04:27 info: [PRODUCTS][getProducts] - Getting & login products from DB
2023-06-20 14:04:29 info: [PRODUCTS][getProducts] - getting products data from: http://10.10.2.201/oasis/Productos?api_key=
2023-06-20 14:04:29 info: [PRODUCTS][getProducts] - Comparing filtered products with DB for sync & login result
2023-06-20 14:04:30 info: [PRODUCTS][syncProducts] - Syncing with DB....
2023-06-20 14:04:31 info: [PRODUCTS][syncProducts] - Sync with DB sucessful

2023-06-20 14:04:31 info: [PRODUCTS][getStock] - getting stock data from warehouses: GS01,HM01
2023-06-20 14:04:31 info: [PRODUCTS][getStock] - Getting & login stock of products from DB of warehouse GS01
2023-06-20 14:04:32 info: [PRODUCTS][getStock] - getting stock from: https://app-next.grupoleiros.com/api/v1/oasis/existencias/GS01.json?api_key=
2023-06-20 14:04:41 info: [PRODUCTS][getStock] - Matching on hand property of products with available property
2023-06-20 14:04:41 info: [PRODUCTS][getStock] - Comparing products stock with DB and login results
2023-06-20 14:04:46 warn: [PRODUCTS][getStock] - No stock of GS01 needs to be updated

2023-06-20 14:04:46 info: [PRODUCTS][getStock] - Getting & login stock of products from DB of warehouse HM01
2023-06-20 14:04:46 info: [PRODUCTS][getStock] - getting stock from: https://app-next.grupoleiros.com/api/v1/oasis/existencias/HM01.json?api_key=
2023-06-20 14:04:50 info: [PRODUCTS][getStock] - Matching on hand property of products with available property
2023-06-20 14:04:50 info: [PRODUCTS][getStock] - Comparing products stock with DB and login results
2023-06-20 14:04:54 warn: [PRODUCTS][getStock] - No stock of HM01 needs to be updated

2023-06-20 14:04:54 info: [PRODUCTS][syncStock] - Stock sync of all warehouses done

2023-06-20 14:04:54 info: [Main] - Iniciando sincronizadores automaticos
2023-06-20 14:04:55 info: Server listening on port: 5555
```

Figura 20 Mensajes por consola del inicio del programa

Fuente: Elaboración propia (2023).

La figura pertenece a los mensajes mostrados por consola una vez se inicia el servicio API, en ella se aprecia los mensajes de la sincronización inicial de productos y existencias en los almacenes activos actualmente.

Figura 21: Diagrama de flujo sincronización de la BDD del POS

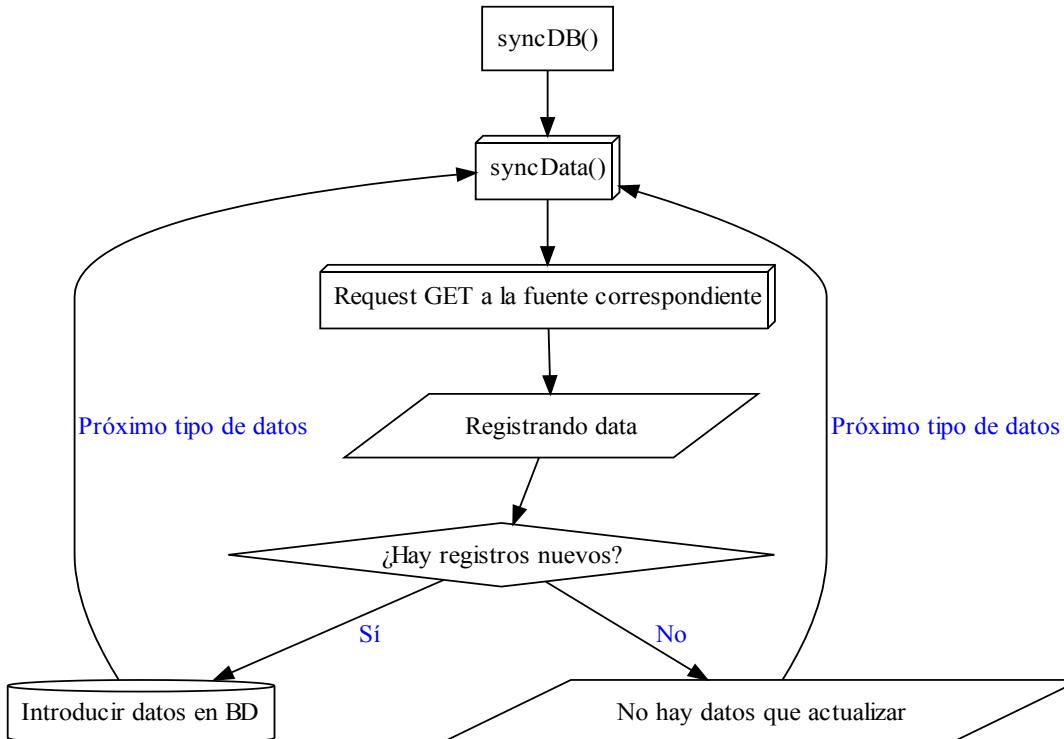


Ilustración 21 Diagrama de flujo sincronización de la BDD del POS

Fuente: Elaboración propia (2023).

La figura pertenece a un diagrama de flujo donde se puede apreciar la lógica de sincronización para los datos introducidos en la base del POS, donde se reciben mediante un GET a las fuentes de la información, se comparan con los datos en la base de datos y se introducen o actualizan los datos según los datos comparados.

Figura 22: Mensajes por consola inicio sincronización BDD

```

2023-06-20 14:13:00 info: [jobSyncDB] - INIT DB SYNC
2023-06-20 14:13:00 info: [jobSyncDB] - Pinging google..
2023-06-20 14:13:10 info: [jobSyncDB] - Response ping google: 200
2023-06-20 14:13:10 info: [jobSyncDB] - SYNC ACTIVE: true
2023-06-20 14:13:10 info: [jobSyncDB] - Executing DB synchronization...

```

Figura 22 Mensajes por consola inicio sincronización BDD

Fuente: Elaboración propia (2023).

Mensajes que marcan el inicio de la sincronización del POS, en ella se aprecia el cómo la API hace un ping a Google con la finalidad de comprobar la estabilidad de la conexión a internet para ejecutar correctamente los métodos sincronizadores.

Figura 23: Mensajes por sincronización de categorías

```
2023-06-20 14:13:10 info: [CATEGORIES][getCategories] - Getting & loggin categories from: http://10.10.2.201/oasis/Instancias?
2023-06-20 14:13:10 info: [CATEGORIES][getCategories] - Filtering & loggin unique categories
2023-06-20 14:13:10 info: [CATEGORIES][syncCategories] - Syncing with DB....
2023-06-20 14:13:10 info: [CATEGORIES][syncCategories] - Sync with DB sucessful
```

Figura 23 Mensajes por sincronización de categorías

Fuente: Elaboración propia (2023).

Figura 24: Mensajes por sincronización de productos

```
2023-06-20 14:13:10 info: [PRODUCTS][getProducts] - Getting & loggin products from DB
2023-06-20 14:13:11 info: [PRODUCTS][getProducts] - getting products data from: http://10.10.2.201/oasis/Productos??
2023-06-20 14:13:12 info: [PRODUCTS][getProducts] - Comparing filtered products with DB for sync & loggin result
2023-06-20 14:13:12 info: [PRODUCTS][syncProducts] - Syncing with DB....
2023-06-20 14:13:12 info: [PRODUCTS][syncProducts] - Sync with DB sucessful
```

Figura 24 Mensajes por sincronización de productos

Fuente: Elaboración propia (2023).

Figura 25: Mensajes por sincronización de códigos de barra

```
2023-06-20 14:13:12 info: [BAR-CODES][getBarCodes] - getting and loggin Bar Codes in DB
2023-06-20 14:13:13 info: [BAR-CODES][getBarCodes] - getting & loggin data from: http://10.10.2.201/oasis/Codigos??
2023-06-20 14:13:13 info: [BAR-CODES][getBarCodes] - Comparing bar codes received with DB and loggin results
2023-06-20 14:13:16 info: [BAR-CODES][syncBarCodes] - Syncing with DB....
2023-06-20 14:13:16 info: [BAR-CODES][syncBarCodes] - Sync with DB sucessful
```

Figura 25 Mensajes por sincronización de códigos de barra

Fuente: Elaboración propia (2023).

Mensajes por consola pertenecientes a la sincronización de categorías, productos y códigos de barra respectivamente, todos los datos que son recibidos e ingresados se guardan en los logs.

Figura 26: Mensajes por sincronización de inventario en todos los almacenes

```
2023-06-20 14:48:07 info: [PRODUCTS][getStock] - getting stock data from warehouses: GS01,HM01
2023-06-20 14:48:07 info: [PRODUCTS][getStock] - Getting & loggin stock of products from DB of warehouse GS01
2023-06-20 14:48:08 info: [PRODUCTS][getStock] - getting stock from: https://go.grupoleiros.com/api/v1/oasis/existencias/GS01.
2023-06-20 14:48:10 info: [PRODUCTS][getStock] - Matching on hand property of products with available property
2023-06-20 14:48:10 info: [PRODUCTS][getStock] - Comparing products stock with DB and login results
2023-06-20 14:48:14 info: [PRODUCTS][getStock] - Getting & loggin stock of products from DB of warehouse HM01
2023-06-20 14:48:15 info: [PRODUCTS][getStock] - getting stock from: https://go.grupoleiros.com/api/v1/oasis/existencias/HM01.
2023-06-20 14:48:36 info: [PRODUCTS][getStock] - Matching on hand property of products with available property
2023-06-20 14:48:36 info: [PRODUCTS][getStock] - Comparing products stock with DB and login results
2023-06-20 14:48:41 warn: [PRODUCTS][getStock] - No stock of HM01 needs to be updated

2023-06-20 14:48:41 info: [PRODUCTS][syncStock] - Syncing 4 products for GS01
2023-06-20 14:48:41 info: [PRODUCTS][syncStock] - Load of products per cycle: 1000
2023-06-20 14:48:41 info: [PRODUCTS][syncStock] - Syncing stock with DB 1 of 1 for GS01...
2023-06-20 14:48:41 info: [PRODUCTS][syncStock] - Stock sync of all warehouses done
```

Figura 26 Mensajes por sincronización de inventario en todos los almacenes

Fuente: Elaboración propia (2023).

La presente figura muestra los mensajes pertenecientes a la sincronización de existencias en todos los almacenes presentes en la configuración del programa, en la misma se apreció cómo la carga de las existencias se hace en ciclos de mil (1000) productos por ciclos, a petición del cliente con la finalidad de evitar cargas pesadas como la creación de un nuevo almacén. Adicionalmente, la carga por ciclo es configurable mediante las variables de entorno del DotEnv.

Figura 27: Diagrama de flujo sincronización documentos para B2B

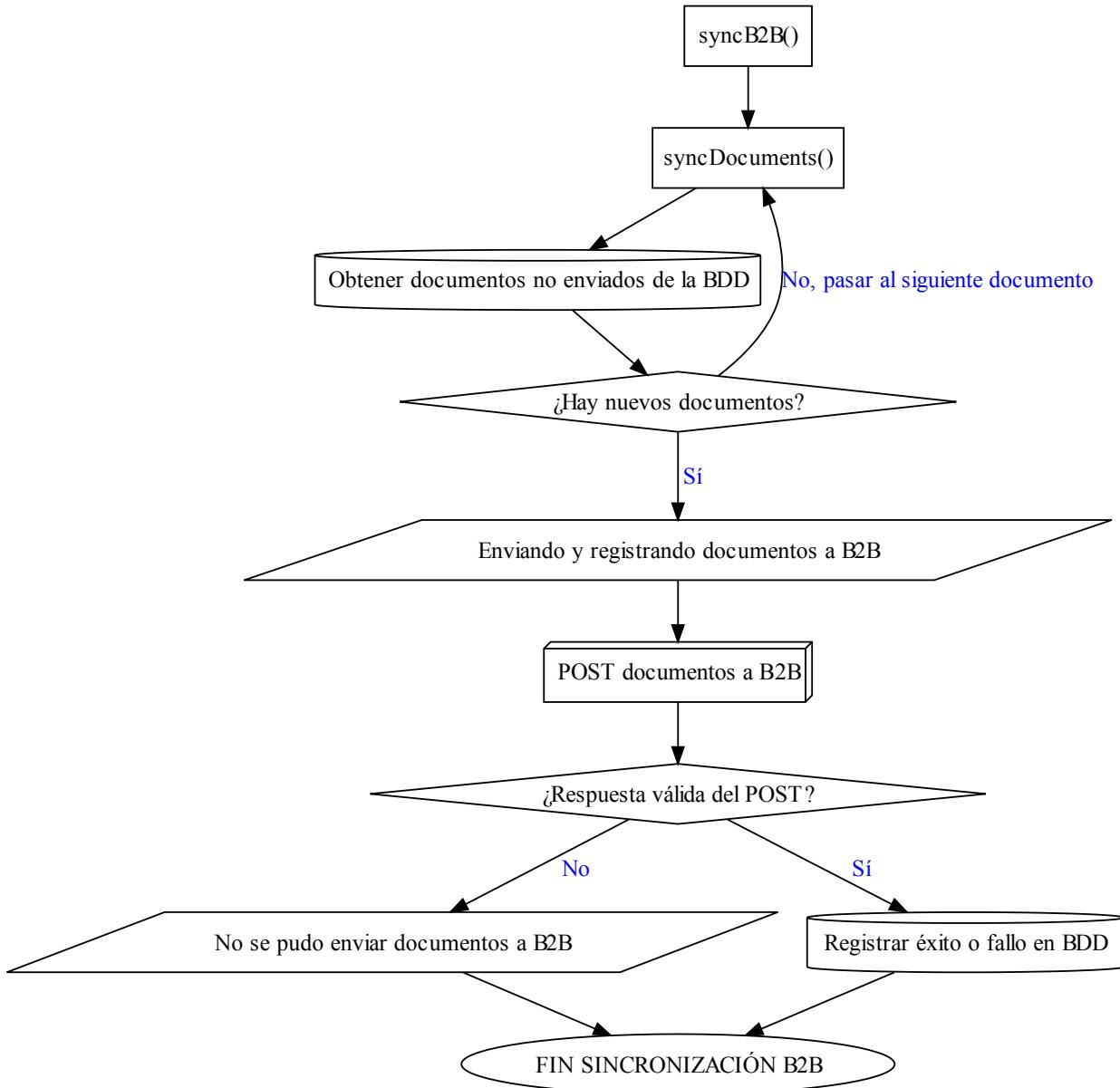


Figura 27 Diagrama de flujo sincronización documentos para B2B

Fuente: Elaboración propia (2023).

La presente figura pertenece al diagrama de flujo donde se plantea la lógica para la sincronización de documentos en B2B, el ERP de la empresa, el mismo muestra la búsqueda de documentos sin enviar en la base de datos del POS y luego lo manda a su respectivo endpoint. En caso de no conseguir documentos, pasa al siguiente tipo de documento, finalmente, al hacer el envío, se registra en la base de datos si la respuesta del ERP fue un éxito o un fallo.

Figura 28: Mensajes inicio de sincronización B2B

```
2023-06-20 14:16:00 info: [jobSyncB2B] - INIT B2B SYNC
2023-06-20 14:16:00 info: [jobSyncB2B] - Pinging google..
2023-06-20 14:16:10 info: [jobSyncB2B] - Response ping google: 200
2023-06-20 14:16:10 info: [jobSyncB2B] - SYNC ACTIVE: true
2023-06-20 14:16:10 info: [jobSyncB2B] - Executing B2B synchronization...
```

Figura 28 Mensajes inicio de sincronización B2B

Fuente: Elaboración propia (2023).

Figura 29: Mensajes sincronización de documentos: SIN ENVIOS

```
2023-06-20 14:16:10 info: [INVOICE-ORDERS][getOrdersForB2B]- Getting & loggin orders stored in DB
2023-06-20 14:16:10 warn: No orders received from DB for sync
2023-06-20 14:16:10 info: [INVOICE-CREDIT][getCreditMemoForB2B]- Getting & loggin credit memos stored in DB
2023-06-20 14:16:10 warn: No credit memos received from DB for sync
2023-06-20 14:16:10 info: [INVOICE-PAYMENT][getBanks]- Getting & loggin banks of B2B from: https://app-next.
2023-06-20 14:16:11 info: [INVOICE-PAYMENT][getPaymentsForB2B]- Getting payments stored in DB
2023-06-20 14:16:12 warn: No payments received from DB for sync
2023-06-20 14:16:12 info: [jobSyncB2B] - B2B synchronization executed successfully
```

Figura 29 Mensajes sincronización de documentos: SIN ENVIOS

Fuente: Elaboración propia (2023).

Figura 30: Mensajes sincronización de documentos: CON ENVIOS

```
2023-06-20 14:48:00 info: [INVOICE-ORDERS][getOrdersForB2B]- Getting & loggin orders stored in DB
2023-06-20 14:48:00 info: [INVOICE-ORDERS][getOrdersForB2B]- Grouping orders and products
2023-06-20 14:48:00 info: [INVOICE-ORDERS][syncOrdersForB2B]- Sending orders to https://go.grupoleiros.com/api/v1/oasis/invoice/create.json?as=1
2023-06-20 14:48:04 info: [INVOICE-ORDERS][syncOrdersForB2B]- Server response: 201
2023-06-20 14:48:04 info: [INVOICE-ORDERS][syncOrdersForB2B]- Some orders were added successfully, loggin orders added to B2B successfully
2023-06-20 14:48:07 info: [INVOICE-ORDERS][syncOrdersForB2B]- Syncing stock after sending orders
```

Figura 30 Mensajes sincronización de documentos: CON ENVIOS

Fuente: Elaboración propia (2023).

Figura 31: Mensajes sincronización de pagos: SOLO REGISTROS

```
2023-06-20 14:48:47 info: [INVOICE-PAYMENT][getBanks]- Getting & loggin banks of B2B from: https://go.grupoleiros.com/api/v1/oasis/banks.json
2023-06-20 14:48:47 info: [INVOICE-PAYMENT][getPaymentsForB2B]- Getting payments stored in DB
2023-06-20 14:48:54 info: [INVOICE-PAYMENT][getPaymentsForB2B]- Grouping transactions of payments & loggin results...
2023-06-20 14:48:54 info: [INVOICE-PAYMENT][syncPaymentsForB2B]- Sending invoices to: https://go.grupoleiros.com/api/v2/oasis/pay/add.json?as=1
2023-06-20 14:48:54 info: [INVOICE-PAYMENT][syncPaymentsForB2B]- Server response: 201
2023-06-20 14:48:54 warn: [INVOICE-PAYMENT][syncPaymentsForB2B]- Some payments couldn't be added,loggin all payments that couldn't be added
2023-06-20 14:48:57 warn: [INVOICE-PAYMENT][syncPaymentsForB2B]- No new payment was added
2023-06-20 14:48:57 info: [jobSyncB2B] - B2B synchronization executed successfully
```

Figura 31 Mensajes sincronización de pagos: SOLO REGISTROS

Fuente: Elaboración propia (2023).

Las cuatro figuras anteriores forman parte de los métodos sincronizadores para los documentos en el ERP, en ellas se aprecian casos cuando no hay documentos que enviar, cuando

son enviados satisfactoritamente, se registran como “success” y no vuelven a ser enviados. Adicionalmente, los pagos solo se registran, ya que, a petición del cliente, en este caso, los documentos de GUUAO Marketplace no necesitan una referencia de pago en el ERP.

Figura 32: Diagrama de flujo sincronización de precios y tasa de cambio en BDD

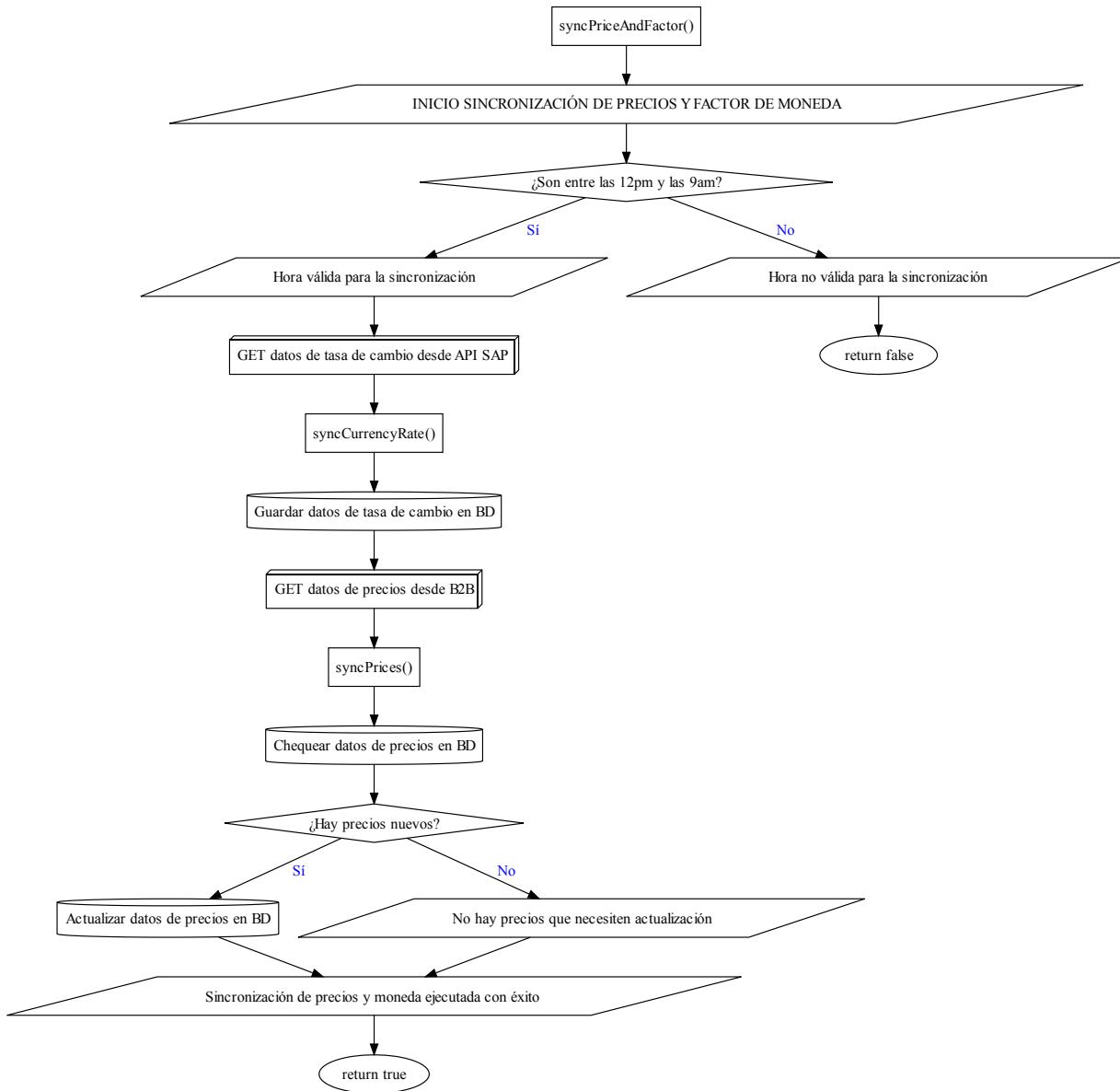


Figura 32 Diagrama de flujo sincronización de precios y tasa de cambio en BDD

Fuente: Elaboración propia (2023).

La presente figura pertenece al diagrama de flujo referente a la sincronización de precios en el POS en ella se aprecia como la sincronización solo es permitida en un horario establecido ya que, por peticiones del cliente, las actualizaciones automáticas de los precios solo son permitidas antes de que abra la tienda de GUUAO MarketPlace en el Sambil, Margarita.

Figura 33: Mensajes sincronización de precio y tasa: HORARIO NO VALIDO

```
2023-06-20 14:16:00 info: [jobSyncPrice] - INIT PRICES AND CURRENCY FACTOR SYNC
2023-06-20 14:16:00 warn: [PRICES][syncPrices&syncCurrencyRate] - Time not valid for price & currency update: 14:16
2023-06-20 14:16:00 info: [jobSyncPrice] - Time for the sync not valid, skipping synchronization
```

Figura 33 Mensajes sincronización de precio y tasa: HORARIO NO VALIDO

Fuente: Elaboración propia (2023).

Figura 34: Mensajes sincronización de tasa de cambio

```
2023-06-20 05:00:03 info: [PRICES][getCurrencyRate] - rate of: 2023-06-20
2023-06-20 05:00:03 info: [PRICES][getCurrencyRate] - getting & loggin rate from: http://190.114.244.120/rate/
2023-06-20 05:00:03 info: [PRICES][syncPrices] - Syncing factor with DB...
2023-06-20 05:00:03 info: [PRICES][syncPrices] - Sync with DB sucessful
```

Figura 34 Mensajes sincronización de tasa de cambio

Fuente: Elaboración propia (2023).

Figura 30: Mensajes sincronización de precios con USD y sin USD

```
2023-06-20 05:08:14 info: [PRICES][syncPrices] - Sending price list 1 for DB for sync
2023-06-20 05:08:14 info: [PRICES][syncPrices] - Syncing usd prices with DB...
2023-06-20 05:08:14 info: [PRICES][syncPrices] - Sync with DB sucessful
2023-06-20 05:08:14 info: [PRICES][syncPrices] - Syncing no usd prices with DB...
2023-06-20 05:08:14 info: [PRICES][syncPrices] - Sync with DB sucessful
2023-06-20 05:08:14 info: [PRICES][getPrices] - getting prices data from: https://go.grupoleiros.com/
2023-06-20 05:08:14 warn: [PRICES][getPrices] - No price needs to be updated
2023-06-20 05:08:14 info: [jobSyncPrice] - Sync of prices and currency executed successfully
```

Figura 35 Mensajes sincronización de precios con USD y sin USD

Fuente: Elaboración propia (2023).

Las anteriores figuras pertenecen a los mensajes mostrados por consola una vez se ejecuta el método sincronizador de precios y tasa de cambio, en las mismas se aprecian los casos fuera del horario válido, adicionalmente, la sincronización de precio muestra dos tipos de mensajes, para los productos que poseen referente en divisas y para los que no.

Figura 36: Ejemplo registro de logs en archivos de la API

02Payments-2023-03-09.log.gz	972	2023-06-21 16:24:00 Info: [jobSyncPrice] - Plinging B2B sync...
02Payments-2023-03-10.log.gz	973	2023-06-21 16:24:00 Info: [jobSyncPrice] - INIT PRICES AND CURRENCY FACTOR SYNC
02Payments-2023-03-11.log.gz	974	2023-06-21 16:24:00 Info: [jobSyncPrice] - Time for the sync not valid, skipping synchronization
02Payments-2023-03-12.log.gz	975	2023-06-21 16:24:00 Info: [jobSyncB2B] - Response ping google: 200
02Payments-2023-03-13.log.gz	976	2023-06-21 16:24:00 Info: [jobSyncB2B] - SYNC ACTIVE: true
02Payments-2023-03-14.log.gz	977	2023-06-21 16:24:00 Info: [jobSyncB2B] - Executing B2B synchronization...
02Payments-2023-03-15.log.gz	978	2023-06-21 16:24:12 Info: [jobSyncB2B] - B2B synchronization executed successfully
02Payments-2023-06-09.log	979	
02Payments-2023-06-13.log	980	2023-06-21 16:26:00 Info: [jobSyncDiscounts] - INIT DISCOUNT SYNC
02Payments-2023-06-19.log.gz	981	2023-06-21 16:26:00 Info: [jobSyncDB] - INIT DB SYNC
02Payments-2023-06-20.log	982	2023-06-21 16:26:00 Info: [jobSyncDB] - Plinging google...
02Payments-2023-06-21.log	983	2023-06-21 16:26:00 Info: [jobSyncDB] - Response ping google: 200
Cron&SyncStatus-2023-03-09L	984	2023-06-21 16:26:00 Info: [jobSyncDB] - SYNC ACTIVE: true
Cron&SyncStatus-2023-03-10L	985	2023-06-21 16:26:00 Info: [jobSyncDB] - Executing DB synchronization...
Cron&SyncStatus-2023-03-11L	986	2023-06-21 16:26:00 Info: [jobSyncDB] - DB synchronization executed successfully
Cron&SyncStatus-2023-03-12L	987	2023-06-21 16:26:17 Info: [jobSyncDB] - DB synchronization executed successfully
Cron&SyncStatus-2023-03-13L	988	
Cron&SyncStatus-2023-03-14L	989	2023-06-21 16:32:00 Info: [jobSyncB2B] - INIT B2B SYNC
Cron&SyncStatus-2023-03-15L	990	2023-06-21 16:32:00 Info: [jobSyncB2B] - Plinging google...
Cron&SyncStatus-2023-06-09.log	991	2023-06-21 16:32:00 Info: [jobSyncB2B] - INIT PRICES AND CURRENCY FACTOR SYNC
Cron&SyncStatus-2023-06-13.log	992	2023-06-21 16:32:00 Info: [jobSyncB2B] - Time for the sync not valid, skipping synchronization
Cron&SyncStatus-2023-06-19L	993	
Cron&SyncStatus-2023-06-20.log	994	2023-06-21 16:32:00 Info: [jobSyncB2B] - Response ping google: 200
Cron&SyncStatus-2023-06-21.log	995	2023-06-21 16:32:00 Info: [jobSyncB2B] - SYNC ACTIVE: true
Cron&SyncStatus-2023-06-21.log	996	2023-06-21 16:32:00 Info: [jobSyncB2B] - Executing B2B synchronization...
Cron&SyncStatus-2023-06-21.log	997	2023-06-21 16:32:02 Info: [jobSyncB2B] - B2B synchronization executed successfully
Cron&SyncStatus-2023-06-21.log	998	

Figura 36 Ejemplo registro de logs en archivos de la API

Fuente: Elaboración propia (2023).

La presente figura pertenece a captura de pantallas de los registros o logs generados por la API, estos fueron desarrollados con la finalidad de poseer un registro a largo plazo de todas las acciones importantes de la API, con el fin de facilitar la solución de posibles errores en producción, algunos de estos logs incluyen los datos obtenidos del ERP y los datos introducidos en el POS.

5.4.2 Rutas GET de la API

Figura 37.1: datos JSON de la ruta /products/:company_db

```
{  
    "CodProd": "GUCM0667",  
    "CodAlte": "GUCM0667",  
    "Descrip": "HARINA PAN MAIZ 1KG",  
    "Existen": 19,  
    "CodUbic": "GS01",  
    "Marca": "ALIMENTOS POLAR",  
    "T_Precio3": "Bs.49,05 ",  
    "T_REF1": "1,80 ",  
    "%Descto": 0,  
    "Precio3": "Bs.49,05 ",  
    "REF1": "1,80 ",  
    "img": "http [REDACTED] /images/placeholder.jpg",  
},  
,  
{  
    "CodProd": "GUCM0647",  
    "CodAlte": "7591002001803",  
    "Descrip": "HARINA PAN MAIZ AMARILLA 1KG",  
    "Existen": 33,  
    "CodUbic": "GS01",  
    "Marca": "ALIMENTOS POLAR",  
    "T_Precio3": "Bs.49,05 ",  
    "T_REF1": "1,80 ",  
    "%Descto": 0,  
    "Precio3": "Bs.49,05 ",  
    "REF1": "1,80 ",  
    "img": "http [REDACTED] /images/placeholder.jpg",  
},  
,  
{  
    "CodProd": "GUCM0648",  
    "CodAlte": "7591002200046",  
    "Descrip": "HARINA PAN MAIZ BLANCO Y ARROZ 1KG",  
    "Existen": 32,  
    "CodUbic": "GS01",  
    "Marca": "ALIMENTOS POLAR",  
    "T_Precio3": "Bs.43,60 ",  
    "T_REF1": "1,60 ",  
    "%Descto": 0,  
    "Precio3": "Bs.43,60 ",  
    "REF1": "1,60 ",  
    "img": "http [REDACTED] /images/placeholder.jpg",  
}
```

Figura 37 Figuras de los json en crudo y en formato tabla de las rutas

Fuente: Elaboración propia (2023).

La presente figura muestra los datos en crudo del JSON obtenido como respuesta a una solicitud GET a la ruta de productos de la API en un ambiente de pruebas, dicha ruta recibe como parámetro un “**:company_db**” y como argumento “**search**”, los cual corresponde al nombre de la

base de datos del Punto de Ventas y la búsqueda a realizar en la ruta. El ejemplo de la presente figura fueron los resultados usando **GUUAO** y **HARINA**, respectivamente.

Figura 38: Datos JSON en formato de tabla

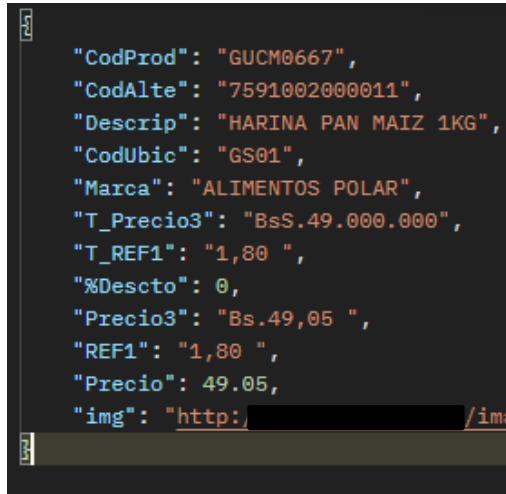
S.no	CodProd	CodAlte	Descrip	Existen	CodUbic	Marca	T_Precio3	T_REF1	%DescTo	Precio3	REF1	Img
1	GUTH0407	883049478487	DOSCIFICADOR DE HARINA KITCHENAID - KSMS			KITCHENAID	Bs.3.597,00	132,00		Bs.3.597,00	132,00	http://host/images/placeholder.png
2	GUCM2360	096619338696	HARINA DE ALMENDRA KIRKLAND SIGNATURE 13	8	GS01	PROTECOM	Bs.708,50	26,00		Bs.708,50	26,00	http://host/images/placeholder.png
3	GUCM0667	7591002000011	HARINA PAN MAIZ 1KG	19	GS01	ALIMENTOS POLAR	Bs.49,05	1,80		Bs.49,05	1,80	http://host/images/placeholder.png
4	GUCM0667	GUCM0667	HARINA PAN MAIZ 1KG	19	GS01	ALIMENTOS POLAR	Bs.49,05	1,80		Bs.49,05	1,80	http://host/images/placeholder.png
5	GUCM0647	7591002001803	HARINA PAN MAIZ AMARILLA 1KG	33	GS01	ALIMENTOS POLAR	Bs.49,05	1,80		Bs.49,05	1,80	http://host/images/placeholder.png

Ilustración 38 Datos JSON en formato de tabla

Fuente: Elaboración propia (2023).

La presente figura muestra los datos obtenidos de la ruta de productos en un formato de tabla para facilitar su lectura a personas no familiarizadas con el formato JSON y dar a entender la posibilidad de obtener varias instancias de los datos, el ejemplo se hizo seleccionando los cinco (5) primeros resultados obtenidos de la ruta, sin embargo, la cantidad será igual a cantidad de resultados obtenidos.

Figura 39.1: datos JSON de la ruta /bar/:company_db



```
"CodProd": "GUCM0667",
"CodAlte": "7591002000011",
"Descrip": "HARINA PAN MAIZ 1KG",
"CodUbic": "GS01",
"Marca": "ALIMENTOS POLAR",
"T_Precio3": "Bs.49.000.000",
"T_REF1": "1,80",
"%Descto": 0,
"Precio3": "Bs.49,05",
"REF1": "1,80",
"Precio": 49.05,
"img": "http://host/images/placeholder.png"
```

Fuente: Elaboración propia (2023).

La presente figura muestra los resultados obtenidos de la ruta de búsqueda de productos mediante el código de barra. Adicionalmente, la ruta recibe el parámetro “**bar**”, el cual será igual al código de barra del producto del cual se quiere obtener la información.

Figura 39.2: Datos JSON en formato de tabla

CodProd	GUCM0667
CodAlte	7591002000011
Descrip	HARINA PAN MAIZ 1KG
CodUbic	GS01
Marca	ALIMENTOS POLAR
T_Precio3	Bs.49.000.000
T_REF1	1,80
%Descto	0
Precio3	Bs.49,05
REF1	1,80
Precio	49.05
img	http://host/images/placeholder.png

Fuente: Elaboración propia (2023).

Figura 40.1: datos JSON de la ruta /documents/:company_db

```
[  
  {  
    "Fecha": "2022-05-20T00:00:00.000Z",  
    "Name": "CESAR ACERO",  
    "Total": 970.2,  
    "Num": "00011431",  
    "SerialIF": "1FC2309779",  
    "CodEsta": "SERVIDOR"  
  },  
  {  
    "Fecha": "2022-05-20T00:00:00.000Z",  
    "Name": "SHADY",  
    "Total": 137.2,  
    "Num": "00011432",  
    "SerialIF": "1FC2309779",  
    "CodEsta": "SERVIDOR"  
  },  
  {  
    "Fecha": "2022-05-20T00:00:00.000Z",  
    "Name": "JOSE SANCHEZ",  
    "Total": 2009,  
    "Num": "00011433",  
    "SerialIF": "1FC2309779",  
    "CodEsta": "CAJAB"  
  },  
  {  
    "Fecha": "2022-05-20T00:00:00.000Z",  
    "Name": "WILHEM SALDEÑO",  
    "Total": 80.85,  
    "Num": "00011434",  
    "SerialIF": "1FC2309779",  
    "CodEsta": "SERVIDOR"  
  },  
  {  
    "Fecha": "2022-05-20T00:00:00.000Z",  
    "Name": "RICARDO CAMPELLO",  
    "Total": 10848.6,  
    "Num": "00011435",  
    "SerialIF": "1FC2309779",  
    "CodEsta": "CAJAB"  
  },  
]
```

Fuente: Elaboración propia (2023).

La figura pertenece al reporte de las facturas emitidas, el mismo puede recibir los parámetros “datefrom” y “datefrom” para seleccionar un rango específico de la fecha de los documentos.

Figura 40.2: Datos JSON en formato de tabla

S.no	Fecha	Name	Total	Num	SerialIF	CodEsta
1	2022-05-20T00:00:00.000Z	CESAR ACERO	970.2	00011431	1FC2309779	SERVIDOR
2	2022-05-20T00:00:00.000Z	SHADY	137.2	00011432	1FC2309779	SERVIDOR
3	2022-05-20T00:00:00.000Z	JOSE SANCHEZ	2009	00011433	1FC2309779	CAJA8
4	2022-05-20T00:00:00.000Z	WILHEM SALDEÑO	80.85	00011434	1FC2309779	SERVIDOR
5	2022-05-20T00:00:00.000Z	RICARDO CAMPOLLO	10848.6	00011435	1FC2309779	CAJA8
6	2022-05-20T00:00:00.000Z	BARBARA ANGULO	44.1	00011436	1FC2309779	CAJA8
7	2022-05-20T00:00:00.000Z	LIBRERIA AVANCE, C.A	132.15	00011437	1FC2309779	CAJA8
8	2022-05-20T00:00:00.000Z	JOSE ANDARA	102.9	00011438	1FC2309779	CAJA8
9	2022-05-20T00:00:00.000Z	RAMON PAZ	57.92	00011439	1FC2309779	CAJA8
10	2022-05-20T00:00:00.000Z	ALVARO LIZARDI	11.04	00011440	1FC2309779	CAJA8

Fuente: Elaboración propia (2023).

La figura muestra los diez (10) primeros resultados del uso de la ruta en un ambiente de pruebas en un formato de tabla para su mejor visualización.

Figura 41.1: datos JSON de la ruta /existencias/:company_db/:whs

```
[  
    {  
        "ItemCode": "GUTH2333",  
        "OnHand": 7,  
        "Compromised": 0,  
        "Available": 7  
    },  
    {  
        "ItemCode": "GUTH2334",  
        "OnHand": 8,  
        "Compromised": 0,  
        "Available": 8  
    },  
    {  
        "ItemCode": "GUTH2335",  
        "OnHand": 2,  
        "Compromised": 0,  
        "Available": 2  
    },  
    {  
        "ItemCode": "GUTH2336",  
        "OnHand": 3,  
        "Compromised": 0,  
        "Available": 3  
    },  
    {  
        "ItemCode": "GUTH2337",  
        "OnHand": 1,  
        "Compromised": 0,  
        "Available": 1  
    },  
    {  
        "ItemCode": "GUTH2338",  
        "OnHand": 12,  
        "Compromised": 0,  
        "Available": 12  
    },  
    {  
        "ItemCode": "GUTH2339",  
        "OnHand": 0,  
        "Compromised": 0,  
        "Available": 0  
    }]
```

Fuente: Elaboración propia (2023).

La figura muestra la ruta utilizada para la consulta de existencias dada un almacén “:whs:”.

Figura 41.2: Datos JSON en formato de tabla

S.no	ItemCode	OnHand	Compromised	Available
1	GUTH2334	8		8
2	GUTH2335	2		2
3	GUTH2336	3		3
4	GUTH2337	1		1
5	GUTH2338	12		12
6	GUTH2339			

Fuente: Elaboración propia (2023).

Las siguientes rutas pertenecen a reportes solicitados por el cliente y la información detallada sobre las variables de ruta usadas puede ser consultada en la documentación de la API, ubicada en las referencias de la presente investigación.

Figura 43.1: Datos JSON de la ruta /history/sales/:company_db/:fechai/:fechae

```
{  
    "CodUbic": "GS01",  
    "CodItem": "GUCP1227",  
    "Articulo": "CREMA DENTAL COLGATE MENTA 75ML",  
    "Disponible": 44,  
    "PrecioUndDiv": 1.38,  
    "InvDIV": 60.72,  
    "TotalVent": 324,  
    "TotalUSD": 384.47,  
    "PromMensualUSD": 13.29,  
    "PromCantidadMensual": 10.98,  
    "2021-01": 32,  
    "2021-02": 17,  
    "2021-03": 22,  
    "2021-04": 13,  
    "2021-05": 7,  
    "2021-06": 15,  
    "2021-07": 15,  
    "2021-08": 41,  
    "2021-09": 3,  
    "2021-10": 0,  
    "2021-11": 0,  
    "2021-12": 0,  
    "2022-01": 0,  
    "2022-02": 0,  
    "2022-03": 0,  
    "2022-04": 0,  
    "2022-05": 9,  
    "2022-06": 9,  
    "2022-07": 13,  
    "2022-08": 24,  
    "2022-09": 57,  
    "2022-10": 17,  
    "2022-11": 18,  
    "2022-12": 12  
},
```

Fuente: Elaboración propia (2023).

Figura 43.2: Datos JSON en formato de tabla

S.no	CodUbic	CodItem	Artículo	Disponible	PrecioUndDiv	InvDIV	TotalVent	TotalUSD	PromMensualUSD	PromCantidadMensual	2021-01	///-//
1	GS01	GUCP1227	CREMA DENTAL COLGATE MENTA 75ML	44	1.38	60.72	324	384.47	384.47	324	32	/
2	GS01	GUCP1228	JABON PROTEX ALOE TRIPACK 270GR		2.49		287	599.9	599.9	287		/
3	GS01	GUCP1229	LIMPIADOR FABULOSO ANTIBACTERIAL FRESCO	6	3.36	20.16	255	707.34	707.34	255	8	/
4	GS01	GUCP1230	ACONDICIONADOR DE TELAS SUAVITEL PRIMAVERA	1	4.67	4.67	65	258.89	258.89	65	1	/

Fuente: Elaboración propia (2023).

Figura 44.1: Datos JSON de la ruta /sales/category/:company_db/:periodo

```
{  
    "fecha": "2022-05-31T00:00:00.000Z",  
    "CodProd": "GUBE0768",  
    "NameProd": "BEBIDA ENERGETICA FURIA C-AZUCAR 250ML",  
    "UndVendidas": 73,  
    "FC": 73,  
    "NC": 0,  
    "Ventas $": 91.29,  
    "Ventas VEF": 462.82,  
    "CodInst": 5664,  
    "Descrip": "GUBE103-Bebidas (N.A)",  
    "Categoria": "5664-GUBE103-Bebidas (N.A)"  
},  
{  
    "fecha": "2022-05-31T00:00:00.000Z",  
    "CodProd": "GUCM0487",  
    "NameProd": "CHOCOLATE TOBLERONE LECHE 100GR",  
    "UndVendidas": 119,  
    "FC": 119,  
    "NC": 0,  
    "Ventas $": 183.31,  
    "Ventas VEF": 929.39,  
    "CodInst": 5673,  
    "Descrip": "GUCM102-Consumo Masivo",  
    "Categoria": "5673-GUCM102-Consumo Masivo"  
},  
{  
    "fecha": "2022-05-31T00:00:00.000Z",  
    "CodProd": "871200030599",  
    "NameProd": "CERVEZA HEINEKEN BOTELLA 250ML",  
    "UndVendidas": 156,  
    "FC": 300,  
    "NC": 144,  
    "Ventas $": 154.46,  
    "Ventas VEF": 783.12,  
    "CodInst": 5699,  
    "Descrip": "CERVEZAS",  
    "Categoria": "5699-CERVEZAS"  
},  
}
```

Fuente: Elaboración propia (2023).

Figura 44.2: Datos JSON en formato de tabla

S.no	fecha	CodProd	NameProd	UndVendidas	FC	NC	Ventas \$	Ventas VEF	CodInst	Descrip	Categoría
1	2022-05-31T00:00:00.000Z	GUCM3626	PAPAS CONGELADAS CRINKLE CUT 907GR	57	57		123.11	624.15	5673	GUCM102-Consumo Masivo	5673- GUCM102- Consumo Masivo
2	2022-05-31T00:00:00.000Z	50196388	BUCHANAN'S DELUXE 12 YEARS 0,75L	59	83	24	973.56	4935.94	5692	WHISKIES	5692- WHISKIES
3	2022-05-31T00:00:00.000Z	GUBE0768	BEBIDA ENERGETICA FURIA C-AZUCAR 250ML	73	73		91.29	462.82	5664	GUBE103-Bebidas (N.A)	5664- GUBE103- Bebidas (N.A)
4	2022-05-31T00:00:00.000Z	GUCM0487	CHOCOLATE TOBLERONE LECHE 100GR	119	119		183.31	929.39	5673	GUCM102-Consumo Masivo	5673- GUCM102- Consumo Masivo
5	2022-05-31T00:00:00.000Z	871200030599	CERVEZA HEINEKEN BOTELLA 250ML	156	300	144	154.46	783.12	5699	CERVEZAS	5699- CERVEZAS

Fuente: Elaboración propia (2023).

Figura 45.1: Datos JSON de la ruta /excel/:company_db/ventas/clientes/:group/:year

```
[{"Año": "2022", "Mes": "2022-05", "Fecha": "2022-05-16", "TipoDocumento": "FC", "NumeroControl": "2001", "Impresora": "Z1F0019693", "NumeroFactura": "Z1F0019693-2001", "Monto VEF": 33.62, "Monto USD": 7.04821803, "Factor": 4.77, "CodigoCliente": "10076", "Cliente": "EDUARDO RIVAS", "CodUbic": "GS01", "Almacen": "Almacén GUUAO Sambil"}, {"Año": "2022", "Mes": "2022-05", "Fecha": "2022-05-16", "TipoDocumento": "FC", "NumeroControl": "2002", "Impresora": "Z1F0019780", "NumeroFactura": "Z1F0019780-2002", "Monto VEF": 233.08, "Monto USD": 48.86412998, "Factor": 4.77, "CodigoCliente": "V-8320187", "Cliente": "ELIO MOYA", "CodUbic": "GS01", "Almacen": "Almacén GUUAO Sambil"}, {"Año": "2022", "Mes": "2022-05", "Fecha": "2022-05-16", "TipoDocumento": "FC", "NumeroControl": "2003", "Impresora": "Z1F0019693", "NumeroFactura": "Z1F0019693-2003", "Monto VEF": 233.08, "Monto USD": 48.86412998, "Factor": 4.77, "CodigoCliente": "V-8320187", "Cliente": "ELIO MOYA", "CodUbic": "GS01", "Almacen": "Almacén GUUAO Sambil"}]
```

Fuente: Elaboración propia (2023).

Figura 45.2: Datos JSON en formato de tabla

S.no	Año	Mes	Fecha	TipoDocumento	NumerоКontrol	Impresora	NumerоЦаrta	Monto VEF	Monto USD	Factor	CódigoCliente	Cliente	CodUbic	Almacen
1	2022	2022-05	2022-05-16	FC	2001	Z1F0019693	Z1F0019693-2001	33.62	7.04821803	4.77	10076	EDUARDO RIVAS	GS01	Almacén GUUAO Sambil
2	2022	2022-05	2022-05-16	FC	2002	Z1F0019780	Z1F0019780-2002	233.08	48.86412998	4.77	V-8320187	ELIO MOYA	GS01	Almacén GUUAO Sambil
3	2022	2022-05	2022-05-16	FC	2003	Z1F0019693	Z1F0019693-2003	47.7	10	4.77	v-15658817	CHRISTIAN RIOS	GS01	Almacén GUUAO Sambil
4	2022	2022-05	2022-05-16	FC	2004	Z1F0019780	Z1F0019780-2004	172.94	36.2566457	4.77	V-14840216	SOCRATES MOYA	GS01	Almacén GUUAO Sambil
5	2022	2022-05	2022-05-16	FC	2005	1FC2309779	1FC2309779-2005	31.01	6.50104822	4.77	V-14840216	SOCRATES MOYA	AL0JW01	ALMACEN DONDE ALFREDO VII

Fuente: Elaboración propia (2023).

Figura 46.1: Datos JSON de la ruta

/excel/:company_db/ventas/productos/:group/:rep/:range/:week/:brand

Fuente: Elaboración propia (2023).

Figura 46.2: Datos JSON en formato de tabla

S.no	Anio	Mes	Articulo	Almacen	CodUbic	CodItem	Categoría	Cantidad	TotalItem	Fecha
1	2022	2022-12	CHOCOLATE SAVOY CRI CRI 12X27GR	Almacén GUUAO Sambil	GS01	GUCM0944	5673-GUCM102-Consumo Masivo	8	99.44	2022-12-28
2	2022	2022-12	CHOCOLATE SAVOY GALAK BLANCO 12X30GR	Almacén GUUAO Sambil	GS01	GUCM0911	5673-GUCM102-Consumo Masivo	8	99.44	2022-12-28
3	2022	2022-12	CHOCOLATE SAVOY LECHE 12X30GR	Almacén GUUAO Sambil	GS01	GUCM0941	5673-GUCM102-Consumo Masivo	8	99.44	2022-12-28
4	2022	2022-12	COCOSETTE MAXI MULTIPACK 60X200GR	Almacén GUUAO Sambil	GS01	GUCM0934	5673-GUCM102-Consumo Masivo	11	519.31	2022-12-28
5	2022	2022-12	DISPLAY CHOC SAVOY CRI CRI 12X27GR	Almacén GUUAO Sambil	GS01	GUCM5518	5673-GUCM102-Consumo Masivo	1	141.12	2022-12-28
6	2022	2022-12	DISPLAY SUSY MINI CHOCOLATE 18X25GR	Almacén GUUAO Sambil	GS01	GUCM5494	5673-GUCM102-Consumo Masivo	1	120.96	2022-12-28

Fuente: Elaboración propia (2023).

Figura 47.1: Datos JSON de la ruta /excel/:company_db/credito/clientes/:fecha

```
    "Periodo": "2022-10-07",
    "CxCMonto": 41.7728,
    "Div": 5.09,
    "Credito": 41.7728,
    "NcMonto": null,
    "Diff": null,
    "DiffDiv": 5.09,
    "FechaNc": null
},
{
    "FechaFc": "2022-10-07",
    "CodUsua": "GERTG00",
    "CodEsta": "CAJA8",
    "Documento": "1FC2309779-00012057",
    "CodClie": "J-00000000",
    "Descrip": "GASTOS GENERALES",
    "FcMonto": 492.6,
    "CxCMonto": 492.6,
    "Div": 60,
    "Credito": 492.6,
    "NcMonto": 492.6,
    "Diff": 0,
    "DiffDiv": 0,
    "FechaNc": "2022-10-07T00:00:00.000Z"
},
{
    "FechaFc": "2022-10-07",
    "CodUsua": "GERTG00",
    "CodEsta": "CAJA8",
    "Documento": "1FC2309779-00012058",
    "CodClie": "J-00000000",
    "Descrip": "GASTOS GENERALES",
    "FcMonto": 2955.6,
    "CxCMonto": 2955.6,
    "Div": 360,
    "Credito": 2955.6,
    "NcMonto": null,
    "Diff": null,
    "DiffDiv": 360,
    "FechaNc": null
},
```

Fuente: Elaboración propia (2023).

Figura 47.2: Datos JSON en formato de tabla

S.no	FechaFc	CodUsua	CodEsta	Documento	CodClie	Descrip	FcMonto	CxCMonto	Div	Credito	NcMonto	Diff	DiffDiv	FechaNc
1	2022-10-07	GERTG00	CAJA8	1FC2309779-00012057	J-00000000	GASTOS GENERALES	492.6	492.6	60	492.6	492.6			2022-10-07T00:00:00.000Z
2	2022-10-24	GERTG00	CAJA6	Z1F0019780-00038611	V-111444710	AGUSTIN JOSE RODRIGUEZ VELASQUEZ	11821.51	11821.51	1409	11821.51	11821.51			2022-10-25T00:00:00.000Z
3	2020-12-25	GERGLEI01	CAJA5	Z1F0019467-00010183	J-00000000	GASTOS GENERALES	83.315	83.315	87.7	83.315	83.315			2021-07-15T00:00:00.000Z

Fuente: Elaboración propia (2023).

Figura 48.1: Datos JSON de la ruta /excel/:company_db/credito/productos

```
[{"  
    "FechaFc": "2022-12-14",  
    "CodUsua": "GERTG00",  
    "CodEsta": "CAJA6",  
    "Almacen": "GS01",  
    "Documento": "Z1F0019780-00040159",  
    "CodClie": "V-11309952",  
    "Descrip": "ENRIQUE URDANETA",  
    "Monto": 29814.6133,  
    "cxcMonto": 0,  
    "DivFC": 2019.96,  
    "DeudaDiv": 2019.96,  
    "Credito": 0.0042,  
    "NcMonto": null,  
    "Diff": 0,  
    "DiffDiv": 0,  
    "FechaNc": null,  
    "CodItem": "GUTH0783",  
    "Descrip1": "SARTEN VULCANO MAGEFLON 16CM",  
    "cantidad": 1,  
    "Precio": 224.35,  
    "LineDiV": 15.2  
},  
{  
    "FechaFc": "2022-12-14",  
    "CodUsua": "GERTG00",  
    "CodEsta": "CAJA6",  
    "Almacen": "GS01",  
    "Documento": "Z1F0019780-00040159",  
    "CodClie": "V-11309952",  
    "Descrip": "ENRIQUE URDANETA",  
    "Monto": 29814.6133,  
    "cxcMonto": 0,  
    "DivFC": 2019.96,  
    "DeudaDiv": 2019.96,  
    "Credito": 0.0042,  
    "NcMonto": null,  
    "Diff": 0,  
    "DiffDiv": 0,
```

Fuente: Elaboración propia (2023).

Figura 48.2: Datos JSON en formato de tabla

FechaFc	2020-06-26	2020-06-26	2020-06-30
CodUsua	GERGLEI01	GERGLEI01	GERGLEI01
CodEsta	CAJA01	CAJA01	CAJA4
Almacen	GS01	GS01	GS01
Documento	Z1F0019693-00000118	Z1F0019693-00000118	Z1F0019467-00000431
CodClie	V-4246358	V-4246358	V-18550808
Descrip	MILDA YEPES	MILDA YEPES	LUIS MENDOZA
Monto	9.0546	9.0546	146.9952
cxcMonto	0	0	0
DivFc	43.12	43.12	668.16
DeudaDiv	43.12	43.12	668.16
Credito	1.0061	1.0061	146.9952
NcMonto			
Diff	-1.01	-1.01	147
DiffDiv	0	0	668.16
FechaNc			
CodItem	GUCM1685	GUCM2334	GUTH0818
Descrip1	ARANDANOS SECOS OCEAN SPRAY CRAISINS	TOMATES SECOS EN ACEITE DE OLIVA BEL	TELEVISOR LG 60" SMART TV LED 4K 60U
cantidad	1	1	1
Precio	2.8501	3.5078	146.9952
LineDiv	13.57	16.7	668.16

Fuente: Elaboración propia (2023).

Figura 49.1: Datos JSON de la ruta /excel/:company_db/inventario

```
{  
    "CodUbic": "AL0JW01",  
    "Almacen": "ALMACEN DONDE ALFREDO VII",  
    "CodProd": "GUBE0002",  
    "Articulo": "TE VERDE DURAZNO VIDRIO 250ML",  
    "Categoria": "5664-GUBE103-Bebidas (N.A)",  
    "Existen": 0,  
    "PrecioUSD": 0.76,  
    "CostoUSD": 0  
},  
,  
{  
    "CodUbic": "GS01",  
    "Almacen": "Almacén GUUAO Sambil",  
    "CodProd": "GUBE0002",  
    "Articulo": "TE VERDE DURAZNO VIDRIO 250ML",  
    "Categoria": "5664-GUBE103-Bebidas (N.A)",  
    "Existen": 4,  
    "PrecioUSD": 0.76,  
    "CostoUSD": 0  
},  
,  
{  
    "CodUbic": "HM01",  
    "Almacen": "Barra H. Mas x Menos Av. Bolivar",  
    "CodProd": "GUBE0002",  
    "Articulo": "TE VERDE DURAZNO VIDRIO 250ML",  
    "Categoria": "5664-GUBE103-Bebidas (N.A)",  
    "Existen": 12,  
    "PrecioUSD": 0.76,  
    "CostoUSD": 0  
},  
]
```

Fuente: Elaboración propia (2023).

Figura 49.2: Datos JSON en formato de tabla

S.no	CodUbic	Almacen	CodProd	Articulo	Categoría	Existen	PrecioUSD	CostoUSD
1	GS01	Almacén GUUAO Sambil	GUBE0002	TE VERDE DURAZNO VIDRIO 250ML	5664-GUBE103-Bebidas (N.A)	4	0.76	
2	HM01	Barra H. Mas x Menos Av. Bolívar	GUBE0002	TE VERDE DURAZNO VIDRIO 250ML	5664-GUBE103-Bebidas (N.A)	12	0.76	
3	PG01	Almacen Guuaoo Cerro Verde	GUBE0002	TE VERDE DURAZNO VIDRIO 250ML	5664-GUBE103-Bebidas (N.A)		0.76	
4	AL0JW01	ALMACEN DONDE ALFREDO VII	GUBE0006	NECTAR DE MANZANA VIDRIO DEL MONTE 250ML	5664-GUBE103-Bebidas (N.A)		0.86	
5	GS01	Almacén GUUAO Sambil	GUBE0006	NECTAR DE MANZANA VIDRIO DEL MONTE 250ML	5664-GUBE103-Bebidas (N.A)		0.86	
6	HM01	Barra H. Mas x Menos Av. Bolívar	GUBE0006	NECTAR DE MANZANA VIDRIO DEL MONTE 250ML	5664-GUBE103-Bebidas (N.A)		0.86	

Fuente: Elaboración propia (2023).

Figura 50.1: Datos JSON de la ruta /cuadre/:company_db/:day

```
[{"Caja": "CAJA01", "ConDatosFiscales": 30, "Fecha": "2022-12-05T00:00:00.000Z", "PedidosSaint": 30, "NumeroZ": "891", "SerialIF": "Z1F0019693"}, {"Caja": "CAJA02", "ConDatosFiscales": 48, "Fecha": "2022-12-05T00:00:00.000Z", "PedidosSaint": 48, "NumeroZ": "455", "SerialIF": "Z7C0004836"}, {"Caja": "CAJA10GO", "ConDatosFiscales": 9, "Fecha": "2022-12-05T00:00:00.000Z", "PedidosSaint": 9, "NumeroZ": "22", "SerialIF": "Z7C7016550"}, {"Caja": "CAJA10SINU", "ConDatosFiscales": 6, "Fecha": "2022-12-05T00:00:00.000Z", "PedidosSaint": 6, "NumeroZ": "579", "SerialIF": "Z1F0019641"}, {"Caja": "CAJA11BDA", "ConDatosFiscales": 5, "Fecha": "2022-12-05T00:00:00.000Z", "PedidosSaint": 5, "NumeroZ": "151", "SerialIF": "1FC7002415"}]
```

Fuente: Elaboración propia (2023).

Figura 50.2: Datos JSON en formato de tabla

S.no	Fecha	Caja	ConDatosFiscales	PedidosSaint	SerialIF	NumeroZ
1	2022-12-05T00:00:00.000Z	CAJA01	30	30	Z1F0019693	891
2	2022-12-05T00:00:00.000Z	CAJA02	48	48	Z7C0004836	455
3	2022-12-05T00:00:00.000Z	CAJA10GO	9	9	Z7C7016550	22
4	2022-12-05T00:00:00.000Z	CAJA10SINU	6	6	Z1F0019641	579
5	2022-12-05T00:00:00.000Z	CAJA11BDA	5	5	1FC7002415	151
6	2022-12-05T00:00:00.000Z	CAJA11REMO	33	33	Z7C7016601	41
7	2022-12-05T00:00:00.000Z	CAJA11SINU	6	6	1FC5302297	723

Fuente: Elaboración propia (2023).

Figura 51.1: Datos JSON de la ruta /documentos/:company_db/:date

```
{  
    "NoSaint": "0002038",  
    "credito": 0,  
    "Descto": 0,  
    "Doc": "NC",  
    "div": 23.99,  
    "Tipo": "B",  
    "Fecha": "2022-12-05T00:00:00.000Z",  
    "Name": "GOPARTY C.A",  
    "Total": 280.44,  
    "Num": "00000305",  
    "SerialIF": "Z1F0019467",  
    "CodEsta": "CAJA5",  
    "CodUsua": "CAJA 5 TGSM"  
},  
{  
    "NoSaint": "9637330",  
    "credito": 0,  
    "Descto": 0,  
    "Doc": "F",  
    "div": 38.4,  
    "Tipo": "A",  
    "Fecha": "2022-12-05T00:00:00.000Z",  
    "Name": "gnadio mujica",  
    "Total": 448.89,  
    "Num": "00000337",  
    "SerialIF": "Z7C7016550",  
    "CodEsta": "CAJA10GO",  
    "CodUsua": "RHINA MONTILLA"  
},  
{  
    "NoSaint": "9637333",  
    "credito": 0,  
    "Descto": 0,  
    "Doc": "F",  
    "div": 3.29,  
    "Tipo": "A",  
    "Fecha": "2022-12-05T00:00:00.000Z",  
    "Name": "INVERSIONES 170103,C.A.",  
    "Total": 38.46,
```

Fuente: Elaboración propia (2023).

Figura 51.2: Datos JSON en formato de tabla

S.no	NoSaint	credito	Descto	Doc	div	Tipo	Fecha	Name	Total	Num	SerialIF	CodEsta	CodUsua
1	0002038			NC	23.99	B	2022-12-05T00:00:00.000Z	GOPARTY C.A	280.44	00000305	Z1F0019467	CAJA5	CAJA 5 TGSM
2	9637330			F	38.4	A	2022-12-05T00:00:00.000Z	gnadio mujica	448.89	00000337	Z7C7016550	CAJA10GO	RHINA MONTILLA
3	9637333			F	3.29	A	2022-12-05T00:00:00.000Z	INVERSIONES 170103,C.A.	38.46	00000338	Z7C7016550	CAJA10GO	RHINA MONTILLA
4	9637346			F	1.23	A	2022-12-05T00:00:00.000Z	IRENE ALVARADO	14.38	00000339	Z7C7016550	CAJA10GO	RHINA MONTILLA

Fuente: Elaboración propia (2023).

Figura 52.1: Datos JSON de la ruta /gerencia/:company_db/:mes

```
[{"Detalle": [{"Fecha": "2022-05-01", "Ordenes": 275, "Clientes": 273, "Monto": 19458.11, "Div": 4333.52, "GUUAO": 3835.11}, {"Fecha": "2022-05-02", "Ordenes": 216, "Clientes": 215, "Monto": 16972.73, "Div": 3780.07, "GUUAO": 3423.9}, {"Fecha": "2022-05-03", "Ordenes": 215, "Clientes": 209, "Monto": 16671.19, "Div": 3696.49, "GUUAO": 3293.71}, {"Fecha": "2022-05-04", "Ordenes": 305, "Clientes": 300, "Monto": 16120.32, "Div": 3574.39, "GUUAO": 2338.06}, {"Fecha": "2022-05-05", "Ordenes": 276, "Clientes": 272, "Monto": 21133.64, "Div": 4644.62, "GUUAO": 3025.98}]}]
```

Fuente: Elaboración propia (2023).

Figura 52.2: Datos JSON en formato de tabla

Detalle	Object [21]						
	S.no	Fecha	Ordenes	Clientes	Monto	Div	GUUAO
1	2023-06-01	253	252	125446.7	4775.27	4751.27	
2	2023-06-02	325	323	145166.48	5484.2	5480.01	
3	2023-06-03	421	417	487582.34	18420.21	18383.5	
4	2023-06-04	333	332	133256.81	5034.26	5022.26	
5	2023-06-05	232	232	125385.07	4713.72	4704.71	
6	2023-06-06	213	213	161155.16	6090.49	6082.99	
7	2023-06-07	255	253	223363.26	8387.66	8387.66	
8	2023-06-08	250	247	132052.2	4945.79	4916.1	
9	2023-06-09	351	349	415863.77	15482.66	15462.47	
10	2023-06-10	473	470	275529.59	10257.94	10200.08	
11	2023-06-11	327	324	119759.74	4458.64	4448.14	
12	2023-06-12	214	214	110834.1	4126.33	4125.83	
13	2023-06-13	208	207	93166.4	3448.04	3427.04	
14	2023-06-14	261	260	168770.67	6246.1	6243.1	
15	2023-06-15	326	322	263031.48	9702.35	9691.85	
16	2023-06-16	463	463	353449.93	13008.84	13005.84	
17	2023-06-17	682	677	396985.85	14611.21	14575.21	
18	2023-06-18	346	345	153380.05	5645.18	5624.18	
19	2023-06-19	259	259	290771.16	10701.93	10691	
20	2023-06-20	284	282	196162.05	7190.7	7172.7	
21	2023-06-21	47	47	57411.58	2106.84	2106.84	

Total	Object [1]						
	S.no	Fecha	Almacen	Monto	TotalDiv	Tasa	
1	2023-06-21T00:00:00.000Z	GS01	57411.5814	2106.84	27.25		

Fuente: Elaboración propia (2023).

Figura 53.1: Datos JSON de la ruta /report/:company_db/:day

```
{  
    "Detalle": [  
        {  
            "Fecha": "2022-05-12T00:00:00.000Z",  
            "Impreso": "12-05-2022",  
            "SerialIF": "1FC2309779",  
            "ReporteZ": "495",  
            "FacturasSaint": 13,  
            "Facturas": 13,  
            "FacturaInicial": "11336",  
            "FacturaFinal": "11348",  
            "Facturado(Bs)": 19310.73,  
            "IGTF Facturado(Bs)": 0,  
            "Facturado($)": 4161.8,  
            "DevolucionesSaint": 2,  
            "Devoluciones": 2,  
            "DevolucionInicial": "85",  
            "DevolucionFinal": "86",  
            "Devuelto(Bs)": 8860.08,  
            "IGTF Devuelto(Bs)": 0,  
            "Devuelto($)": 1909.5,  
            "TotalVenta": 17401.23,  
            "TotalUsd": 2252.3  
        },  
        {  
            "Fecha": "2022-05-12T00:00:00.000Z",  
            "Impreso": "12-05-2022",  
            "SerialIF": "1FC5302297",  
            "ReporteZ": "510",  
            "FacturasSaint": 27,  
            "Facturas": 27,  
            "FacturaInicial": "15101",  
            "FacturaFinal": "15127",  
            "Facturado(Bs)": 3194.28,  
            "IGTF Facturado(Bs)": 0,  
            "Facturado($)": 688.42,  
            "DevolucionesSaint": 0,  
            "Devoluciones": 0,  
            "DevolucionInicial": "",  
            "DevolucionFinal": "",  
            "Devuelto(Bs)": 0,  
            "Devuelto($)": 0  
        }  
    ]  
}
```

Fuente: Elaboración propia (2023).

Figura 53.2: Datos JSON en formato de tabla

Fecha	2022-05-12T00:00:00.000Z	2022-05-12T00:00:00.000Z	2022-05-12T00:00:00.000Z
Impreso	12-05-2022	12-05-2022	12-05-2022
SerialIF	1FC2309779	1FC5302297	Z1F0019106
ReporteZ	495	510	497
FacturasSaint	13	27	26
Facturas	13	27	26
FacturaInicial	11336	15101	20485
FacturaFinal	11348	15127	20510
Facturado(Bs)	19310.73	3194.28	2310.25
IGTF Facturado(Bs)	0	0	44.73
Facturado(\$)	4161.8	688.42	497.9
DevolucionesSaint	2	0	0
Devoluciones	2	0	0
DevolucionInicial	85		
DevolucionFinal	86		
Devuelto(Bs)	8860.08	0	0
IGTF Devuelto(Bs)	0	0	0
Devuelto(\$)	1909.5	0	0
TotalVenta	17401.23	3194.28	2310.25
TotalUsd	2252.3	688.42	497.9

Fuente: Elaboración propia (2023).

Figura 54.1: Datos JSON de la ruta /conciliar/:company_db/:day

```
{  
    "Detalle": [  
        {  
            "Fecha": "2022-05-12",  
            "Caja": "CAJA01",  
            "Serial": "Z1F0019693",  
            "ReporteZ": "683",  
            "MP": "EFECTIVO (USD $)",  
            "Declarado": 0,  
            "Bs": 127.45,  
            "Div": 27.46,  
            "Tasa": 4.64,  
            "GUUAO": 26.67  
        },  
        {  
            "Fecha": "2022-05-12",  
            "Caja": "CAJA10SINU",  
            "Serial": "Z1F0019641",  
            "ReporteZ": "478",  
            "MP": "EFECTIVO (USD $)",  
            "Declarado": 0,  
            "Bs": 33.74,  
            "Div": 7.27,  
            "Tasa": 4.64,  
            "GUUAO": 7.06  
        },  
        {  
            "Fecha": "2022-05-12",  
            "Caja": "CAJA3",  
            "Serial": "Z1F0019106",  
            "ReporteZ": "497",  
            "MP": "EFECTIVO (USD $)",  
            "Declarado": 0,  
            "Bs": 248.82,  
            "Div": 53.63,  
            "Tasa": 4.64,  
            "GUUAO": 52.06  
        },  
        {  
            "Fecha": "2022-05-12",  
            "Caja": "CAJA4",  
            "Serial": "Z1F0019107",  
            "ReporteZ": "498",  
            "MP": "EFECTIVO (USD $)",  
            "Declarado": 0,  
            "Bs": 248.82,  
            "Div": 53.63,  
            "Tasa": 4.64,  
            "GUUAO": 52.06  
        }  
    ]  
}
```

Fuente: Elaboración propia (2023).

Figura 54.2: Datos JSON en formato de tabla

Detalle	Object [17]											
	S.no	Fecha	Caja	Serial	ReporteZ	MP	Declarado	Bs	Div	Tasa	GUUAO	
1	2022-05-12	CAJA01	Z1F0019693	683	EFFECTIVO (USD \$)	0	127.45	27.46	4.64	26.67		
2	2022-05-12	CAJA10SINU	Z1F0019641	478	EFFECTIVO (USD \$)	0	33.74	7.27	4.64	7.06		
3	2022-05-12	CAJA3	Z1F0019106	497	EFFECTIVO (USD \$)	0	248.82	53.63	4.64	52.06		
4	2022-05-12	CAJA4	1FC5302297	510	EFFECTIVO (USD \$)	0	76.56	16.5	4.64	0		
5	2022-05-12	CAJA6	Z1F0019780	665	EFFECTIVO (USD \$)	0	45.91	9.9	4.64	9.61		
6	2022-05-12	SERVIDOR	1FC2309779	495	EFFECTIVO (USD \$)	0	17.03	3.67	4.64	0		
7	2022-05-12	CAJA01	Z1F0019693	683	EFFECTIVO (VES Bs)	0	128.98	27.8	4.64	30.71		
8	2022-05-12	CAJA02	Z7C0004836	250	EFFECTIVO (VES Bs)	0	18.03	3.89	4.64	35.51		
9	2022-05-12	CAJA3	Z1F0019106	497	EFFECTIVO (VES Bs)	0	23.06	4.98	4.64	76.27		
10	2022-05-12	CAJA5	Z1F0019467	688	EFFECTIVO (VES Bs)	0	5.58	1.2	4.64	1.2		
11	2022-05-12	CAJA6	Z1F0019780	665	EFFECTIVO (VES Bs)	0	19.54	4.21	4.64	10.03		
12	2022-05-12	CAJA8	1FC2309779	495	EFFECTIVO (VES Bs)	0	31.52	6.79	4.64	0		
13	2022-05-12	CAJA9HEINE	1FC5302297	510	EFFECTIVO (VES Bs)	0	134.48	28.99	4.64	0		
14	2022-05-12	CAJA01	Z1F0019693	683	EFFECTIVO DOLARES (USD \$)	0	270.51	58.29	4.64	82.97		
15	2022-05-12	CAJA02	Z7C0004836	250	EFFECTIVO DOLARES (USD \$)	0	277.33	59.77	4.64	62.41		
16	2022-05-12	CAJA11SINU	1FC5302297	510	EFFECTIVO DOLARES (USD \$)	0	166.84	35.96	4.64	0		
17	2022-05-12	CAJA3	Z1F0019106	497	EFFECTIVO DOLARES (USD \$)	0	1015.88	218.94	4.64	215		

Cajas	Object [11]										
	CAJA01	CAJA10SINU	CAJA3	CAJA4	CAJA6	SERVIDOR	CAJA02	CAJA5	CAJA8	CAJA9HEINE	CAJA11SINU

Fuente: Elaboración propia (2023).

Figura 55.1: Datos JSON de la ruta /detalle/:company_db/:day/:cajas

```
{  
    "Detalle": [  
        {  
            "Fecha": "2022-05-12",  
            "MP": "EFECTIVO (USD $)",  
            "Declarado": 0,  
            "Fiscal": 0,  
            "Bs": 127.45,  
            "Div": 27.46,  
            "Tasa": 4.64,  
            "GUUAO": 127.45  
        },  
        {  
            "Fecha": "2022-05-12",  
            "MP": "EFECTIVO (VES Bs)",  
            "Declarado": 0,  
            "Fiscal": 0,  
            "Bs": 128.98,  
            "Div": 27.8,  
            "Tasa": 4.64,  
            "GUUAO": 128.98  
        },  
        {  
            "Fecha": "2022-05-12",  
            "MP": "EFECTIVO DOLARES (USD $)",  
            "Declarado": 0,  
            "Fiscal": 0,  
            "Bs": 270.51,  
            "Div": 58.29,  
            "Tasa": 4.64,  
            "GUUAO": 270.51  
        },  
        {  
            "Fecha": "2022-05-12",  
            "MP": "PAGO MOVIL (VES Bs)",  
            "Declarado": 0,  
            "Fiscal": 0,  
            "Bs": 13.87,  
            "Div": 2.99,  
            "Tasa": 4.64,  
            "GUUAO": 13.87  
        }  
    ]  
}
```

Fuente: Elaboración propia (2023).

Figura 55.2: Datos JSON en formato de tabla

S.no	Fecha	MP	Declarado	Fiscal	Bs	Div	Tasa	GUUAO
1	2022-05-12	EFFECTIVO (USD \$)			127.45	27.46	4.64	127.45
2	2022-05-12	EFFECTIVO (VES Bs)			128.98	27.8	4.64	128.98
3	2022-05-12	EFFECTIVO DOLARES (USD \$)			270.51	58.29	4.64	270.51
4	2022-05-12	PAGO MOVIL (VES Bs)			13.87	2.99	4.64	13.87
5	2022-05-12	POS TDD BANPLUS (VES Bs)			21.88	4.72	4.64	21.88
6	2022-05-12	VPOS DEBITO (Bs)			801.75	172.82	4.64	801.75
7	2022-05-12	ZELLE DIV (VES Bs)			364.12	78.47	4.64	364.12

Fuente: Elaboración propia (2023).

Figura 56.1: Datos JSON de la ruta /soporte/documentos/diferencias/:company_db/:date

```
{  
    "fecha": "2022-05-15T00:00:00.000Z",  
    "CodEsta": "CAJA4",  
    "NumeroZ": "513",  
    "Inicial": "15255",  
    "Final": "15291",  
    "diff": "36",  
    "TipoFac": "A",  
    "CorrelativosFiscales": 11,  
    "CorrelativosSaint": 11,  
    "Faltantes": "25"  
},  
{  
    "fecha": "2022-05-15T00:00:00.000Z",  
    "CodEsta": "CAJA4MGNUS",  
    "NumeroZ": "513",  
    "Inicial": "15256",  
    "Final": "15289",  
    "diff": "33",  
    "TipoFac": "A",  
    "CorrelativosFiscales": 5,  
    "CorrelativosSaint": 5,  
    "Faltantes": "28"  
},  
{  
    "fecha": "2022-05-15T00:00:00.000Z",  
    "CodEsta": "CAJA5",  
    "NumeroZ": "691",  
    "Inicial": "25917",  
    "Final": "25932",  
    "diff": "15",  
    "TipoFac": "A",  
    "CorrelativosFiscales": 15,  
    "CorrelativosSaint": 15,  
    "Faltantes": "0"  
},
```

Fuente: Elaboración propia (2023).

Figura 56.2: Datos JSON en formato de tabla

S.no	fecha	CodEsta	NumeroZ	Inicial	Final	diff	TipoFac	CorrelativosFiscales	CorrelativosSaint	Faltantes
1	2022-05-15T00:00:00.000Z	CAJA4	513	15255	15291	36	A	11	11	25
2	2022-05-15T00:00:00.000Z	CAJA4MGNUS	513	15256	15289	33	A	5	5	28
3	2022-05-15T00:00:00.000Z	CAJA5	691	25917	25932	15	A	15	15	0
4	2022-05-15T00:00:00.000Z	CAJA6	668	32348	32385	37	A	37	37	0
5	2022-05-15T00:00:00.000Z	CAJA8	498	11390	11400	10	A	10	10	0

Fuente: Elaboración propia (2023).

Figura 57.1: Datos JSON de la ruta /soporte/documentos/detalle/:company_db/:station/:date

```
[  
  {  
    "fecha": "2022-05-15T00:00:00.000Z",  
    "CodEsta": "CAJA02",  
    "NumeroZ": "253",  
    "Inicial": "11882",  
    "Final": "11912",  
    "diff": "30",  
    "TipoFac": "A",  
    "CorrelativosFiscales": 30,  
    "CorrelativosSaint": 30,  
    "Faltantes": "0"  
  },  
  {  
    "fecha": "2022-05-15T00:00:00.000Z",  
    "CodEsta": "CAJA10SINU",  
    "NumeroZ": "481",  
    "Inicial": "19365",  
    "Final": "19370",  
    "diff": "5",  
    "TipoFac": "A",  
    "CorrelativosFiscales": 5,  
    "CorrelativosSaint": 5,  
    "Faltantes": "0"  
  },  
  {  
    "fecha": "2022-05-15T00:00:00.000Z",  
    "CodEsta": "CAJA11SINU",  
    "NumeroZ": "513",  
    "Inicial": "15257",  
    "Final": "15290",  
    "diff": "33",  
    "TipoFac": "A",  
    "CorrelativosFiscales": 5,  
    "CorrelativosSaint": 5,  
    "Faltantes": "28"  
  },  
]
```

Fuente: Elaboración propia (2023).

Figura 57.2: Datos JSON en formato de tabla

S.no	fecha	CodEsta	NumeroZ	Inicial	Final	diff	TipoFac	CorrelativosFiscales	CorrelativosSaint	Faltantes
1	2022-05-15T00:00:00.000Z	CAJA02	253	11882	11912	30	A	30	30	0
2	2022-05-15T00:00:00.000Z	CAJA10SINU	481	19365	19370	5	A	5	5	0
3	2022-05-15T00:00:00.000Z	CAJA11SINU	513	15257	15290	33	A	5	5	28

Fuente: Elaboración propia (2023).

5.4.3 Rutas POST API

Figura 58: Diagrama POST a la ruta /update/:company/precios

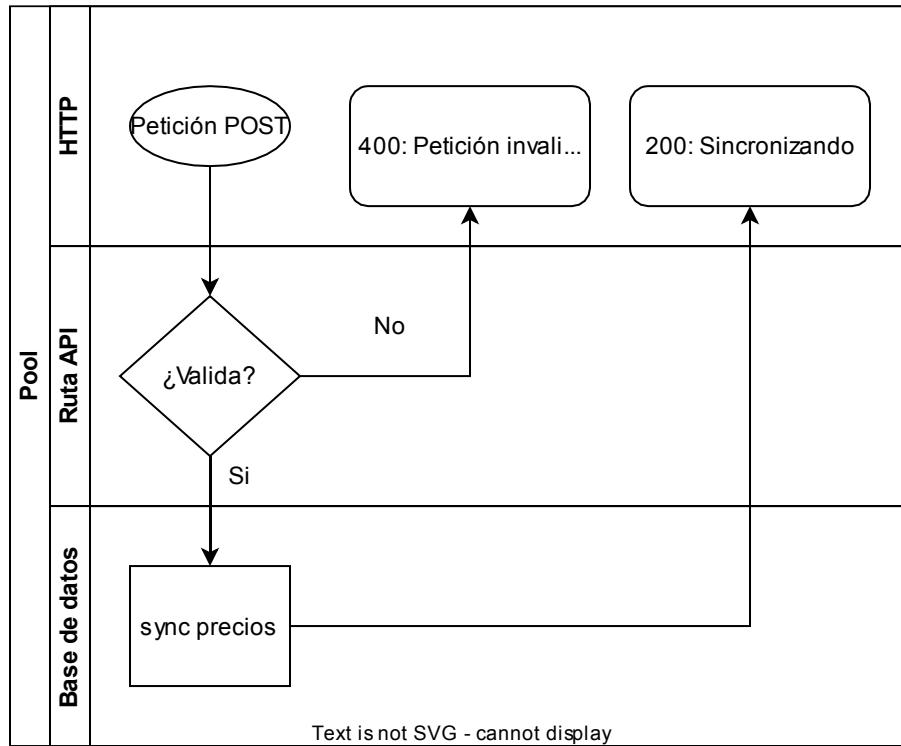
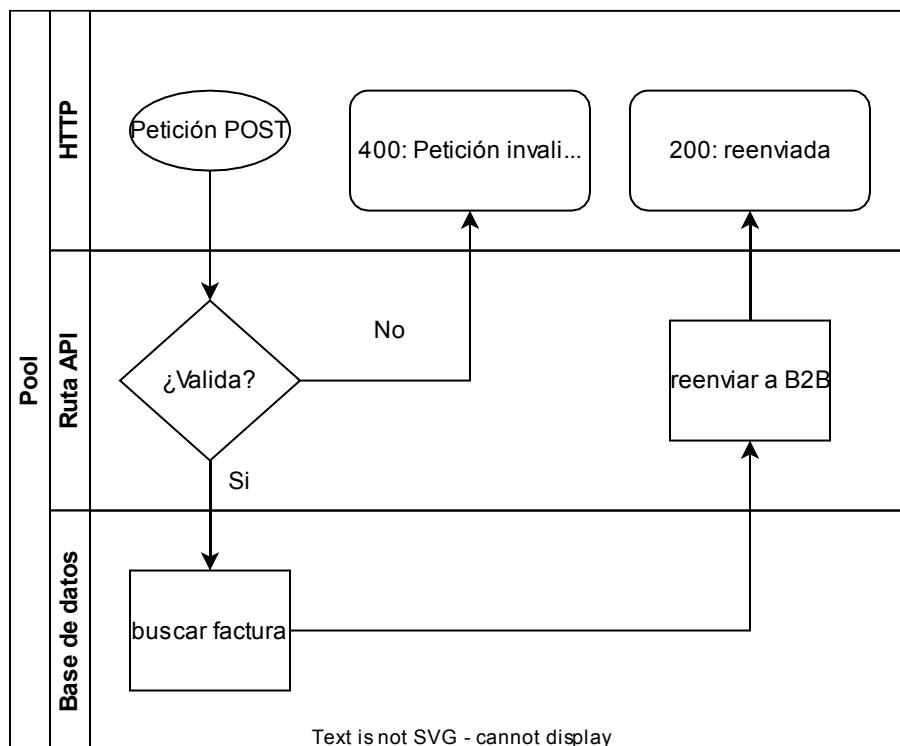


Figura 58 Diagramas de las rutas POST de la API

Fuente: Elaboración propia (2023).

La presente figura representa diagrama de uso de la ruta para la actualización de precios de manera manual con el método POST, la misma recibe un cuerpo en la petición realizada para validar el usuario haciendo la petición. Una vez validado, ejecutará el método sincronizador de precios en la base de datos del POS.

Figura 59: diagrama POST a la ruta /soporte/:company/missing



Fuente: Elaboración propia (2023).

La presente figura representa diagrama de uso de la ruta para el reenvío de documentos, en casos de facturas creadas con datos faltantes en el ERP, en la misma se aprecia la validación de la petición POST, para luego buscar la factura en la base de datos del POS y reenviarla al ERP de la empresa.

5.4.4 Implementación remota de la API

Figura 60: Acceso remoto al servidor



Figura 60 Acceso remoto al servidor

Fuente: Elaboración propia (2023).

La presente figura muestra el método de conexión remota con el servidor, se hizo uso de un programa que poseen todos los sistemas operativos Windows y permite la conexión remota a un equipo, la cual es accesible desde cualquier equipo gracias a la dirección IP pública del servidor remoto. En la figura se ha censurado la dirección y nombre del equipo por medidas de seguridad.

Figura 61: Descripción de inicio automático del servicio

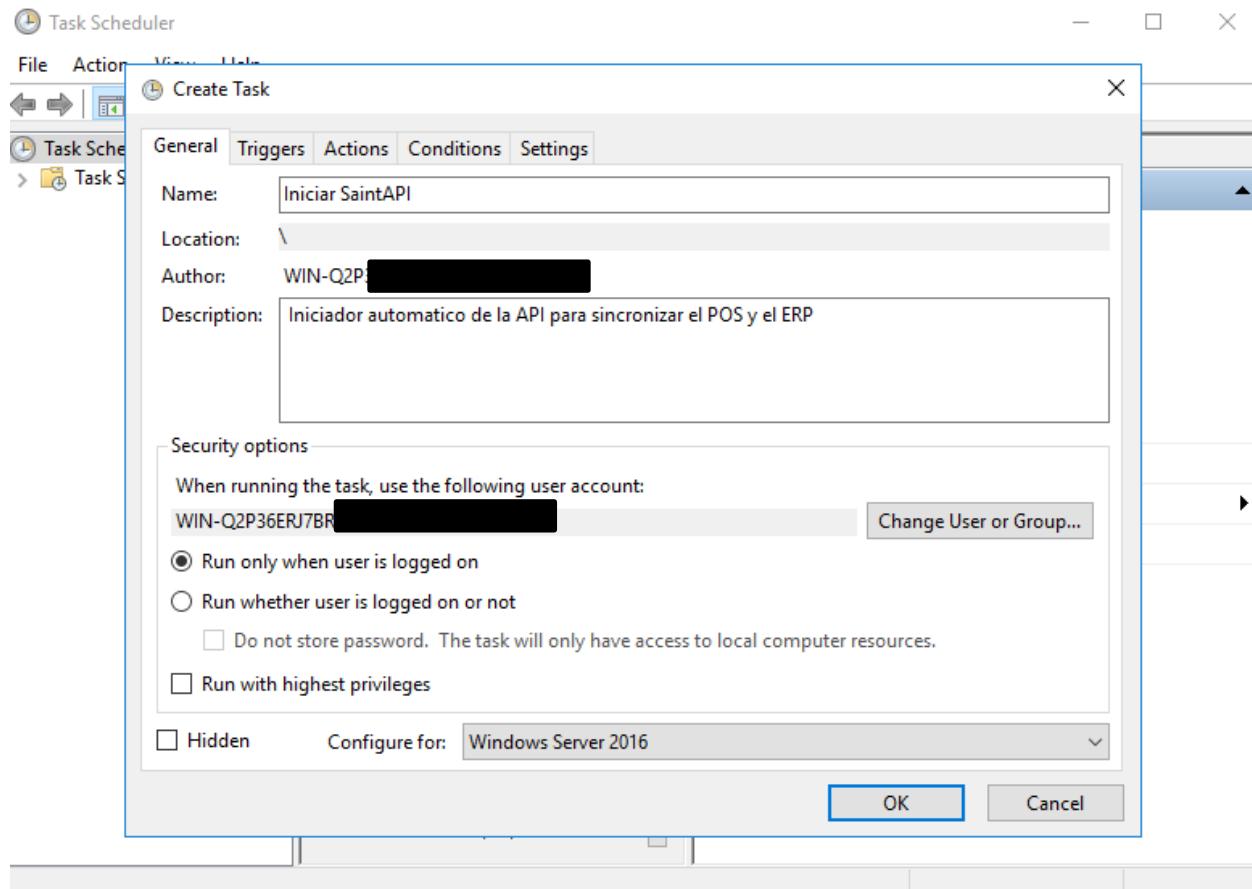


Figura 39 Describiendo el inicio automático del servicio

Fuente: Elaboración propia (2023).

La presente figura muestra las configuraciones iniciales para el inicio automático del servicio API, se hizo uso del planificador de tareas que poseen los sistemas operativos Windows. En la figura se aprecia el nombre de la tarea y su descripción.

Figura 62: Configuración del desencadenador al iniciar el servidor

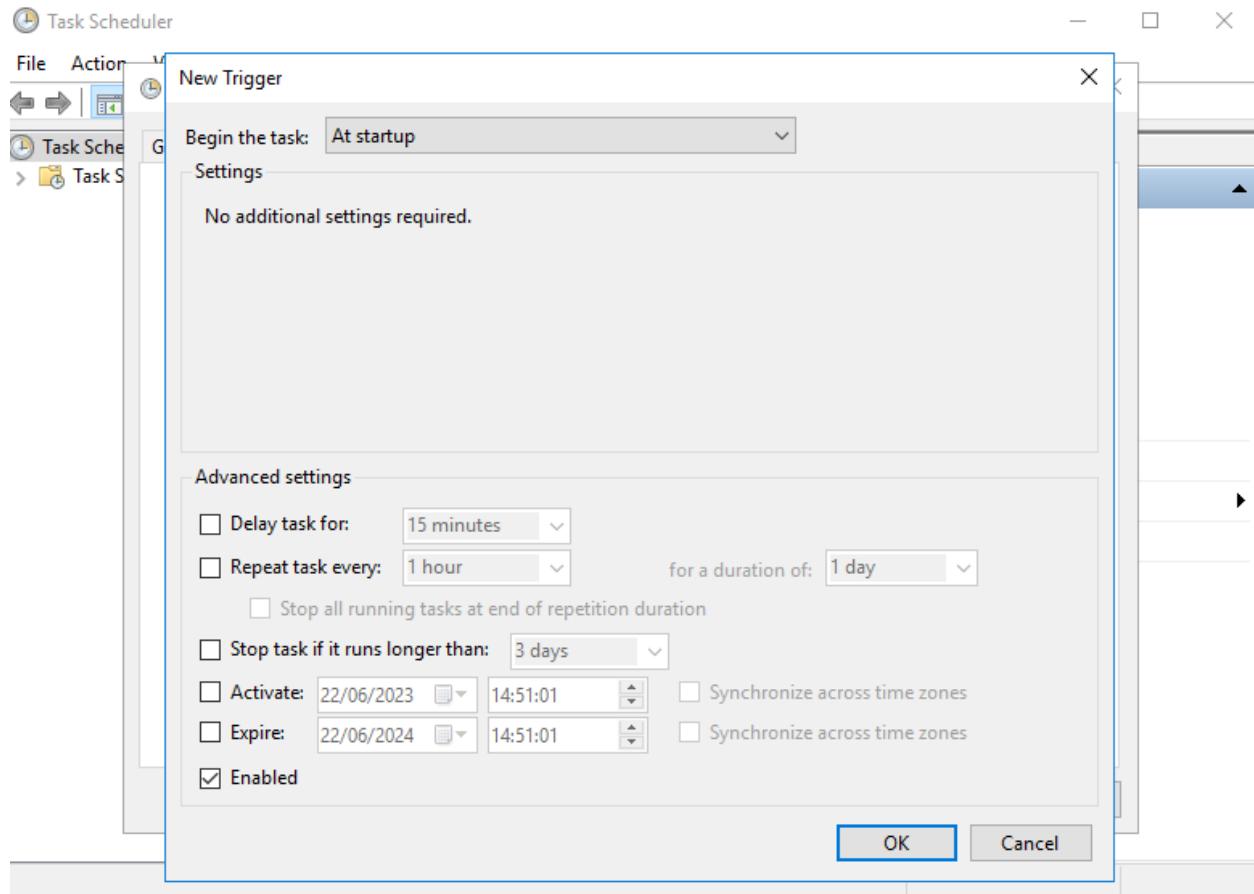


Figura 40 Desencadenador al iniciar el servidor

Fuente: Elaboración propia (2023).

La presente figura muestra la configuración para la actividad que desencadenará el inicio del servicio API, en la misma se aprecia la configuración elegida como “al iniciar”, esto permitirá que, cada vez que se inicie o reinicie el servidor por algún problema, el servicio se ejecute de manera automática.

Figura 63: Configuración de acción al iniciar el servidor

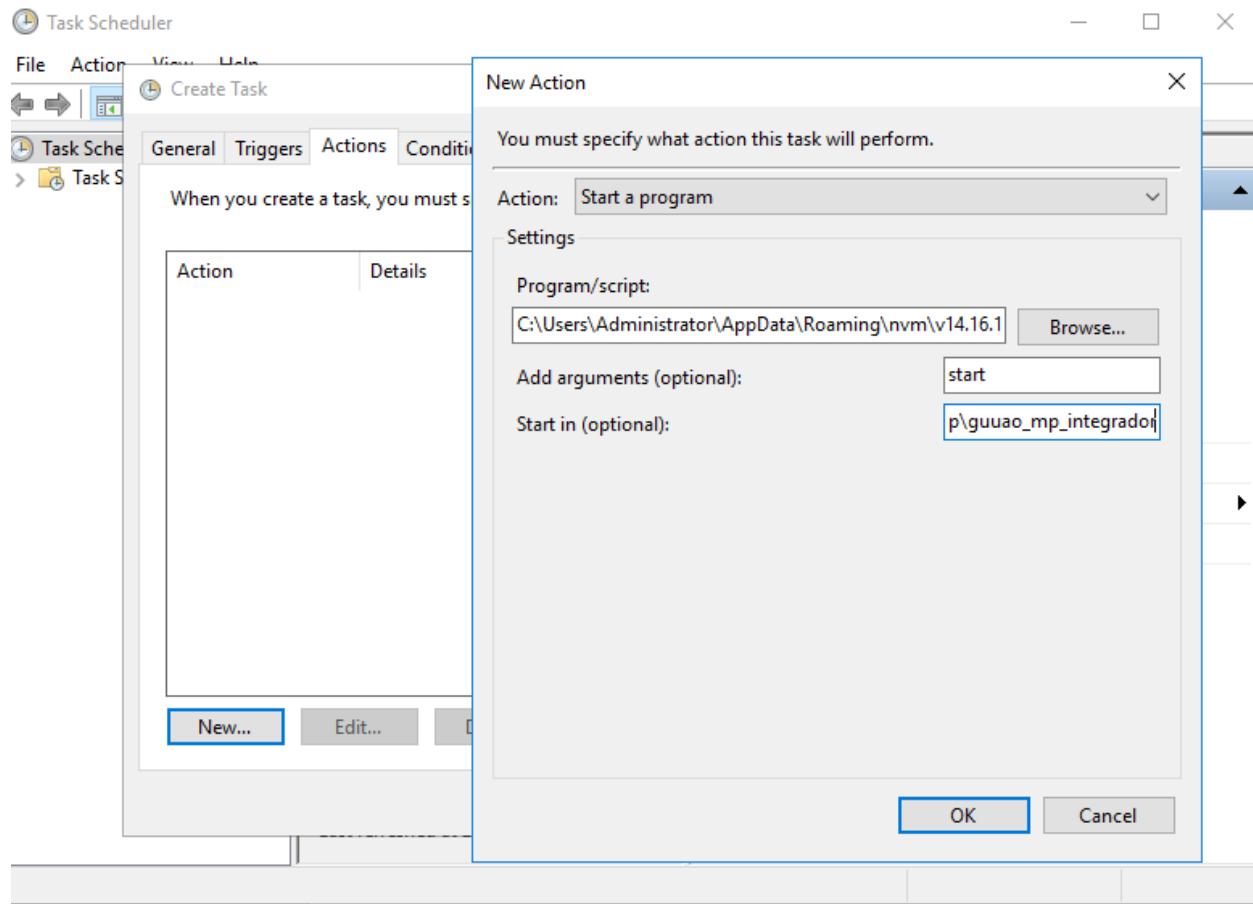


Figura 41 Acción al iniciar el servidor

Fuente: Elaboración propia (2023).

La presente figura muestra las acciones a ejecutar, una vez se lleve a cabo el desencadenador, en la misma se aprecia la configuración de “iniciar un programa” y la ruta de la ubicación de los archivos de la API.

CONCLUSIONES

En un mundo cada vez más conectado, la sincronización y la eficiencia en la gestión de datos son de vital importancia. En el contexto de la economía digital y del comercio electrónico, empresas como GUUAO Marketplace se han visto en la necesidad de optimizar sus procesos internos para responder de forma eficaz a las demandas de un mercado cada vez más exigente. Esta realidad ha llevado a la implementación de sistemas como las APIs, que posibilitan la comunicación y la sincronización eficaz entre diferentes plataformas. Dentro de este marco, el objetivo de la presente investigación fue la implementación de una API para sincronizar el sistema de punto de venta (POS) con el sistema de planificación de recursos empresariales (ERP) de GUUAO Marketplace, cuya consecución produjo las siguientes conclusiones:

- Antes de sincronizar un Punto de Venta (POS) y un sistema de Planificación de Recursos Empresariales (ERP) como los de GUUAO Marketplace, fue imprescindible obtener un entendimiento profundo de sus funcionalidades clave. Por lo tanto, al inicio, este se convirtió en el primer y más crucial objetivo. A través de entrevistas con el personal familiarizado con ambos sistemas, se pudo identificar y comprender a fondo las distintas operaciones, como la gestión de ventas, inventario, datos de los clientes, y funciones de back-end, lo cual permitió esclarecer el panorama para futuras implementaciones y trazar una hoja de ruta sólida para la creación de una API. De esta forma, se logró reunir los componentes esenciales para una sincronización efectiva.
- Después de identificar y comprender a fondo las funcionalidades clave del POS y ERP de GUUAO Marketplace, se usó la revisión documental detallada para realizar el desarrollo de la estructura de una solución de integración óptima. En efecto, se seleccionaron tecnologías y herramientas apropiadas como Node.js, TypeScript, Express.js y una arquitectura REST para la implementación de la API. Se utilizaron sincronizadores a intervalos regulares entre el POS y el ERP para sincronizar los datos. Para la seguridad en el manejo de credenciales y configuraciones, se implementó la librería DotEnv. Por último, se realizó la estructura de la API de una manera modular, escalable y con las mejores prácticas de desarrollo.
- Para finalizar, se aplicó la observación participante durante los procesos de la empresa y la observación en tiempo real de la sincronización de los datos, para determinar si la API se

adaptó y aumentó la eficiencia de las operaciones diarias. De tal manera, se determinó, detallando las fortalezas, oportunidades, debilidades y amenazas de la implementación, que pudo agilizar los procesos, haciéndolos automáticos y, además, es lo suficientemente flexible para futuros cambios, en función de las necesidades de la empresa.

RECOMENDACIONES

Al haber concluido con el desarrollo de la presente investigación, se procede a anexar ciertas recomendaciones con el propósito de mejorar los aspectos funcionales y de implementación del servicio API para la sincronización del POS y el ERP de GUUAO MarketPlace, buscando, de tal modo, aumentar su funcionalidad y capacidad:

1. Implementar la API mediante un servicio en la nube, como DigitalOcean, lo que permitirá aislarla de otros programas que se encuentran en el mismo servidor remoto, además, solucionará posibles problemas, como la necesidad de borrar cíclicamente los logs generados por la API por problemas de espacio.
2. Estandarizar más los formatos de los JSON que se pueden consultar de la API, ya que, muchos de ellos tienen su actual estructura basada en los formatos necesarios ya estructurados como la página del Marketplace y los endpoint de creación de documentos en el ERP, ocasionando que el formato JSON se tuviese que adaptar al formato en que estos sistemas reciben los datos.
3. Eliminar de la base de datos del POS datos de almacenes que ya no existan, estén inactivos o ya no sean socios de negocios de GUUAO, ya que esto ocasiona que la API entregue datos que ya no son utilizados por la empresa.

REFERENCIAS BIBLIOGRAFÍCAS

- Allen, P. (2006). Orientación al servicio: estrategias ganadoras y mejores prácticas. Prensa de la Universidad de Cambridge.
- Anushka, V. (s.f.). Beneficio de usar MVC. Extraído de <https://www.geeksforgeeks.org/benefit-of-using-mvc/>
- Ambler, S. (2002). Técnicas de bases de datos ágiles: estrategias efectivas para el desarrollador de software ágil. John Wiley & Sons.
- Babbie, E. (2009). La práctica de la investigación social (12a ed.). Belmont, CA: Wadsworth Cengage.
- Berners-Lee, T. y Cailliau, R. (1990). WorldWideWeb: Propuesta de Proyecto de Hipertexto. CERN.
- Bobrowski, K. y Markewicz, I. (2018). Sistemas de Punto de Venta Móvil: Problemas y Perspectivas. *Fundamentos de Gestión*, 10(1), 127-138.
- Bowen, G. A. (2009). Análisis de documentos como método de investigación cualitativa. *Revista de Investigación Cualitativa*, 9(2), 27-40.
- Box, D. y Ehnebuske, D. (2000). Protocolo simple de acceso a objetos (SOAP) 1.1. Nota del W3C. Extraído de <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- Bray, T., Paoli, J. y Sperberg-McQueen, C. (1998). Lenguaje de marcado extensible (XML) 1.0. W3C.
- Bryman, A. (2012). Métodos de investigación social (4a ed.). Oxford: Oxford University Press.
- Chappel, D. (2004). Bus de servicios empresariales. O'Reilly Media, Inc.
- Chung, S. y Hossain, MA (2009). Hacia un enfoque holístico para la integración de sistemas empresariales. *Revista de gestión de la información empresarial*
- Coulouris, G., Dollimore, J. y Kindberg, T. (2012). Sistemas distribuidos: conceptos y diseño (5^a ed.). Addison-Wesley.
- Comer, D. (2000). Internetworking con TCP/IP: Principios, Protocolos y Arquitectura (Vol. 1). Prentice Hall.

Crockford, D. (2006). El tipo de medio application/json para la notación de objetos de JavaScript (JSON). RFC 4627.

Creswell, J. W. (2009). Diseño de investigación: enfoques cualitativos, cuantitativos y de métodos mixtos (3a ed.). Thousand Oaks, CA: Sage Publications.

Creswell, J. W. (2012). Investigación educativa: planificación, realización y evaluación de investigaciones cuantitativas y cualitativas (4a ed.). Boston, MA: Pearson Education.

CONSTITUCIÓN DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA. (1999). Gaceta Oficial Extraordinaria N° 36.860 de fecha 30 de diciembre de 1999.

Denscombe, M. (2010). La buena guía de investigación: para proyectos de investigación social a pequeña escala (4a ed.). Maidenhead, Berkshire, Inglaterra: McGraw-Hill/Open University Press.

Davenport, TH (1998). Poner la empresa en el sistema empresarial. Revisión de negocios de Harvard.

DeWalt, K. M., y DeWalt, B. R. (2011). Observación participante: una guía para investigadores de campo (2a ed.). Lanham, MD: Rowman & Littlefield Publishers.

Evans, J., y Lindsay, W. (2011). Administración y control de la calidad. Cengage Learning.

Erl, T. (2005). Arquitectura orientada a servicios: conceptos, tecnología y diseño. PTR de Prentice Hall.

Fransson, A. y Pascoa, R. (2018). El futuro del comercio minorista: una perspectiva sobre la tecnología emergente y los formatos de tienda.

Fielding, R. y Taylor, R. (2002). Diseño fundamentado de la arquitectura web moderna. ACM Transactions on Internet Technology, 2(2), 115-150.

García, A. (2018). Definición e implementación de una API REST para sistemas de recomendación. Trabajo de fin de grado, Universidad Autónoma de Madrid, Madrid, España.
<https://repositorio.uam.es/handle/10486/688292>

Gattiker, T. F. y Goodhue, D. L. (2005). Qué sucede después de la implementación de ERP: comprender el impacto de la interdependencia y la diferenciación en los resultados a nivel de planta. MIS Trimestral

Gravetter, F. J. y Wallnau, L. B. (2016). Estadísticas para las ciencias del comportamiento (10a ed.). Boston, MA: Cengage Learning.

Hernández, R., Fernández, C. y Baptista, P. (2010). Metodología de la investigación (5ta ed.). México, D.F.: McGraw-Hill Interamericana.

He, Y., & Siau, K. (2017). Inquietudes sobre la seguridad y la privacidad de los datos en los sistemas de puntos de venta móviles. Gestión industrial y sistemas de datos.

Hohpe, G. y Woolf, B. (2003). Patrones de integración empresarial: diseño, creación e implementación de soluciones de mensajería. Addison-Wesley.

Klaus, H. y Rosemann, M. (2000). Características y consecuencias del software ERP. En M. O. J. Hawkings (Ed.), Actas de la 7^a Conferencia Europea sobre Sistemas de Información.

Krafcik, D., Banke, K. y Slama, D. (2004). Enterprise SOA: mejores prácticas de arquitectura orientada a servicios. PTR de Prentice Hall.

Kumar, K., & Hillegersberg, J. (2000). Reingeniería de procesos de negocio: Perspectivas de la implementación de sistemas ERP. En M. Hammer & G. Grant (Eds.)

Kim, D. y Lim, J. (2011). Seguridad y privacidad en la era de la computación en la nube: una encuesta de problemas y desafíos.

Kvale, S. (1996). InterViews: Una introducción a la investigación cualitativa mediante entrevistas. Thousand Oaks, CA: Sage Publications.

Lee S y Han I. (2018). Investigación sobre técnicas de venta minorista para una participación y experiencias óptimas del consumidor.

LEY ORGÁNICA DE CIENCIA, TECNOLOGÍA E INNOVACIÓN. (2005). Gaceta Oficial de la República Bolivariana de Venezuela N° 38.493 del 2 de agosto de 2005. Extraído de https://www.oas.org/juridico/PDFs/mesicic4_ven_anexo7.pdf

LEY ORGÁNICA DEL TRABAJO, LOS TRABAJADORES Y LAS TRABAJADORAS. (2012). Gaceta Oficial de la República Bolivariana de Venezuela N° 6.076 Extraordinaria del 7 de mayo de 2012.

Extraído de
https://www.oitcinterfor.org/sites/default/files/file_documento_base/gaceta_oficial_n_6.076.pdf

LEY PARA PERSONAS CON DISCAPACIDAD. (2007). Gaceta Oficial de la República Bolivariana de Venezuela N° 39.118 del 5 de enero de 2007. Extraído de http://www.asambleanacional.gob.ve/uploads/documentos/doc_4f3dd6d4c6f92.pdf

Themistocleous, M. (2004). Justificando las decisiones para implementaciones de EAI: una propuesta validada de factores influyentes. Revista de gestión de la información empresarial.

Martínez, J. (2019). Desarrollo de una API REST para la gestión integral de inventarios y puntos de venta. Trabajo de fin de grado, Universidad de Sevilla, Sevilla, España.
<https://core.ac.uk/download/pdf/61473672.pdf>

Martín, J. y Nwulu, N. (2017). Sistemas de punto de venta basados en la nube: beneficios, desafíos y factores de adopción.

Martínez, M. (2006). La investigación cualitativa (síntesis conceptual). Ediciones Trillas.

Masse, M. (2011). Libro de reglas de diseño de API REST. O'Reilly Media, Inc.

Meneses, K. (2018). Desarrollo de una API REST para la integración de CRM con el sistema de gestión de colas en empresa de fondos colectivos. Trabajo de grado, Universidad Nacional Mayor de San Marcos, Lima, Perú.
https://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/16431/Meneses_rk.pdf?sequence=1

Morales, A., & Vásquez, B. (2021). Desarrollo de API REST para automatizar proceso de recepción, seguimiento y calificación de proyectos de titulación-ESFOT. Trabajo de fin de grado, Escuela Politécnica Nacional, Quito, Ecuador. <https://bibdigital.epn.edu.ec/handle/15000/21994>

Moreno, K. (2022). Introducción a los archivos .env. Recuperado de <https://dev.to/khriztianmoreno/introduccion-a-los-archivos-env-3e2f>

Mozilla Developer Network. (s. f.). HTTP. Extraído de <https://developer.mozilla.org/es/docs/Web/HTTP>

Mulesoft. (2021). ¿Qué es una API? Recuperado de <https://www.mulesoft.com/resources/api/what-is-an-api>

Murdick, R., Ross, J. y Claggett, R. (2004). Sistemas de información para la gestión moderna (3^a ed.). Pearson Prentice Hall.

Nguyen, T. y Kelly, L. (2016). Implementación y mantenimiento de sistemas POS minoristas: una imagen rica.

Pantano, E. y Viassone, M. (2015). Involucrar a los consumidores en nuevos entornos minoristas multicanales integrados: desafíos para los minoristas. Revista de venta minorista y servicios al consumidor.

Preece, J., Rogers, Y. y Sharp, H. (2002). Diseño de interacción: más allá de la interacción humano-computadora. Wiley.

Pressman, R. y Maxim, B. (2018). Ingeniería de software: el enfoque de un profesional. Educación McGraw-Hill.

Real Academia Española. (s. f.). Diccionario de la lengua española (23.^a ed.). Extraído de <https://dle.rae.es>

Roh, J., Kunnathur, A. y Tarafdar, M. (2014). Impacto de un sistema de recopilación de datos de punto de venta en tiempo real basado en RFID en la gestión de inventario. Revista Internacional de Investigación de Producción.

Roh, J., Kunnathur, A. y Tarafdar, M. (2014). Impacto de los sistemas de información en el rendimiento de las tiendas minoristas: un estudio de tres segmentos minoristas.

Rubio, D. M., Berg-Weger, M., Tebb, S. S., Lee, E. S. y Rauch, S. (2003). Objetivando la validez de contenido: realizando un estudio de validez de contenido en la investigación en trabajo social. Investigación en Trabajo Social.

Salazar, S. (2023). Documentación de SaintAPI. Elaboración propia, Ubicada en https://s3w.gitlab.io/guia_apis_integrador

Saldaña, J. (2015). El manual de codificación para investigadores cualitativos (3a ed.). Thousand Oaks, CA: Sage Publications.

Schrock, N. (2015). GraphQL: un lenguaje de consulta de datos. Blog de ingeniería de Facebook.
Recuperado de <https://engineering.fb.com/2015/09/14/core-data/graphql-a-data-query-language/>

Sharma, G. y Roy, R. (2019). Transformando el comercio minorista con inteligencia artificial y realidad aumentada: una revisión y clasificación de las aplicaciones. Revista de venta minorista y servicios al consumidor.

Stallings, W. (2004). Datos y Comunicaciones Informáticas. Prentice Hall.

Stallings, W. (2013). Sistemas operativos: componentes internos y principios de diseño. Prentice Hall.

Tanenbaum, A. y Steen, M. (2007). Sistemas Distribuidos: Principios y Paradigmas. Prentice Hall.

Taylor, R. (2010). Arquitectura de software: fundamentos, teoría y práctica. John Wiley & Sons.

Umble, E. y Umble, M. (2003). Planificación de recursos empresariales: procedimientos de implementación y factores críticos de éxito. Revista europea de investigación operativa

Vargo, S. L., & Lusch, R. F. (2008). Lógica de servicio dominante: continuando la evolución. Revista de la Academia de Ciencias del Marketing.

Vukovic, M. y Bartolini, C. (2016). Plataformas basadas en API: un modelo para el negocio digital. En Actas de la 8.^a ACM/SPEC sobre la Conferencia internacional sobre ingeniería de rendimiento.

Wächter, L. (2020). Cómo estructuro mis API REST de Node.js.. <https://medium.com/swlh/how-i-structure-my-node-js-rest-apis-4e8904ccd2fb>

Zang, N. y Rosson, M. B. (2014). ¿Qué hay en una API? En Actas de la 18^a Conferencia Internacional sobre Trabajo en Grupo de Apoyo

ANEXOS

Figura 57: Ejemplo real del uso de la ruta /bar/:company_db



Figura 42 Ejemplo real del uso de la API, Sambil Margarita

Fuente: Lector de precios GUUAO MP, Sambil Margarita (2023).

Figura 58: Consumo real de la ruta `products/:company_db` en la página de GUUAO MP.

COD. DE PRODUCTO	COD. DE BARRA	DESCRIPCION	EXISTENCIA	ALMACEN	MARCA	PRECIO TACHADO	REF TACHADA	DESCUENTO %	PRECIO	REF
GUCM2360	096619338696	HARINA DE ALMENDRA KIRKLAND SIGNATURE 13	1	GS01	PROTECOM	Bs.626,75	23,00	0%	Bs.626,75	23,00
GUCM0667	75910022000011	HARINA PAN MAIZ 1KG	23	GS01	ALIMENTOS POLAR	Bs.37,33	1,37	0%	Bs.37,33	1,37
GUCM0667	GUCM0667	HARINA PAN MAIZ 1KG	23	GS01	ALIMENTOS POLAR	Bs.37,33	1,37	0%	Bs.37,33	1,37
GUCM0647	7591002001803	HARINA PAN MAIZ AMARILLA 1KG	21	GS01	ALIMENTOS POLAR	Bs.37,33	1,37	0%	Bs.37,33	1,37
GUCM0648	7591002200046	HARINA PAN MAIZ BLANCO Y ARROZ 1KG	4	GS01	ALIMENTOS POLAR	Bs.34,34	1,26	0%	Bs.34,34	1,26
GUCM0533	7591184002056	HARINA DE AVENA QUAKER BOLSA 400GR	13	GS01	ALIMENTOS POLAR	Bs.51,78	1,90	0%	Bs.51,78	1,90
GUCB0181		VINAGRE + BG + MAYONESA 445GR + HARINA PAN 1KG + ARROZ PRIMOR 1KG + OREO AMR 216			ALIMENTOS POLAR	Bs.0,00	0,00	0%	Bs.0,00	0,00
GUCB0296		HARINA PAN MAIZ 20X1KG			ALIMENTOS POLAR	Bs.0,00	0,00	0%	Bs.0,00	0,00
GUCB0297		HARINA PAN MAIZ BLANCO Y ARROZ 20X1KG			ALIMENTOS POLAR	Bs.0,00	0,00	0%	Bs.0,00	0,00
GUCB0299		HARINA PAN MAIZ AMARILLA 20X1KG			ALIMENTOS POLAR	Bs.0,00	0,00	0%	Bs.0,00	0,00
GUCM0742		HARINA DE TRIGO TIGOLAR LEUDANTE 1KG			OXFORD	Bs.0,00	0,00	0%	Bs.0,00	0,00
GUCM1997	8007290841308	HARINA DE TRIGO GRANORO 1000GR			GRANORO	Bs.36,52	1,34	0%	Bs.36,52	1,34
GUAM0051		HARINA PAN MAIZ 20X1KG			ALIMENTOS POLAR	Bs.664,90	24,40	0%	Bs.664,90	24,40
GUAM0052		HARINA PAN MAIZ BLANCO Y ARROZ 20X1KG			ALIMENTOS POLAR	Bs.594,05	21,80	0%	Bs.594,05	21,80
GUAM0054		HARINA PAN MAIZ AMARILLA 20X1KG			ALIMENTOS POLAR	Bs.654,00	24,00	0%	Bs.654,00	24,00
GUCM2360	8007290841308	CHITAS SUCESAS HABLA INTEGRAL 1KG			KIDS WORLD	Bs.32,58	2,30	0%	Bs.32,58	2,30

Figura 43 Ejemplo de uso de la API, página local de GUUAO MP

Fuente: Pagina local GUUAO MP (2023).

Figura 59: Documentación de funciones de la API.

Figura 44 Captura de documentación de la funciones de la API.

Fuente: Repositorio en Gitlab (2023).

Figura 60: Documentación de variables de la API.

Variable	Description	Source Code
B2bSync	(src/.../B2BSynchronizer.ts)	B2bSync
b2bSynchronizer	(src/.../logger.ts)	b2bSynchronizer
dbSync	(src/.../DbSyncronizer.ts)	dbSync
dbSyncronizer	(src/.../logger.ts)	dbSyncronizer
env	(src/.../discounts.ts)	env
env	(src/.../invoices.ts)	env
env	(src/.../invoicesB2B.ts)	env
env	(src/.../products.ts)	env
env	(src/.../index.ts)	env
env	(src/.../index.ts)	env
env	(src/.../invoices.ts)	env
env	(src/.../products.ts)	env
env	(src/.../reports.ts)	env
env	(src/.../barCodes.ts)	env
env	(src/.../categories.ts)	env
env	(src/.../discounts.ts)	env
env	(src/.../invoicesB2B.ts)	env
env	(src/.../prices.ts)	env
env	(src/.../products.ts)	env
errorLog	(src/.../logger.ts)	errorLog
errorLog	(src/.../logger.ts)	errorLog
exceptionsLog	(src/.../logger.ts)	exceptionsLog
exceptionsLog	(src/.../logger.ts)	exceptionsLog
invoiceController	(src/.../routesDataManagers.ts)	invoiceController
jobSyncB2B	(src/.../index.ts)	jobSyncB2B
jobSyncDB	(src/.../index.ts)	jobSyncDB
jobSyncDiscounts	(src/.../index.ts)	jobSyncDiscounts
JobSyncFactor	(src/.../index.ts)	JobSyncFactor
jobSyncPrice	(src/.../index.ts)	jobSyncPrice
logDir	(src/.../logger.ts)	logDir
logDir	(src/.../logger.ts)	logDir
loggerAPI	(src/.../logger.ts)	loggerAPI
loggerBarCodes	(src/.../logger.ts)	loggerBarCodes
loggerCategories	(src/.../logger.ts)	loggerCategories
loggerCreditB2B	(src/.../logger.ts)	loggerCreditB2B
loggerDiscounts	(src/.../logger.ts)	loggerDiscounts
loggerMain	(src/.../logger.ts)	loggerMain
loggerOrdersB2B	(src/.../logger.ts)	loggerOrdersB2B
loggerPaymentsB2B	(src/.../logger.ts)	loggerPaymentsB2B
loggerPrices	(src/.../logger.ts)	loggerPrices
loggerProducts	(src/.../logger.ts)	loggerProducts
priceInterval	(src/.../index.ts)	priceInterval
productController	(src/.../routesDataManagers.ts)	productController
queryBarCodFromDB	(src/.../products.ts)	queryBarCodFromDB
queryCreateSyncB2BTable	(src/.../syncStatus.ts)	queryCreateSyncB2BTable
queryCurrencyRateFromDB	(src/.../products.ts)	queryCurrencyRateFromDB
queryGlobalDiscountLine	(src/.../discounts.ts)	queryGlobalDiscountLine
queryPaymentsDB	(src/.../invoicesB2B.ts)	queryPaymentsDB
queryProductsFromDB	(src/.../products.ts)	queryProductsFromDB
sqlConfig	(db/.../index.ts)	sqlConfig
storage	(src/.../routes.ts)	storage
upload	(src/.../routes.ts)	upload
upload	(src/.../reports.ts)	upload

Figura 45 Captura de documentación de las variables

Fuente: Repositorio en Gitlab (2023).

Figura 61: Ejemplo de explicación de una función.

```

filterProducts
filterProducts(records: string, product: Product)

Compares if product received from B2B is already updated in the DB.

Parameters :


| Name    | Type    | Optional | Description                  |
|---------|---------|----------|------------------------------|
| records | string  | No       | Products stored from the DB. |
| product | Product | No       | Product from B2B.            |


Returns : boolean
True if product is not updated in the DB, false if otherwise.

```

Figura 46 Ejemplo de explicación de una función en la documentacion

Fuente Repositorio en Gitlab (2023).

Figura 62: readme.md de configuraciones y credenciales de la API



VARIABLES DE ENTORNO

- `filedir`: Ruta en la que se almacenan imágenes en el servidor local. (NO SE USA)
- `img_route`: Ruta relativa para acceder a las imágenes desde el servidor. (NO SE USA)
- `PORT`: Puerto en el que se ejecuta la aplicación.
- `db_user`: Nombre de usuario para la base de datos.
- `db_password`: Contraseña para la base de datos.
- `db_server`: Dirección IP del servidor de la base de datos.
- `db_database`: Nombre de la base de datos.
- `db_port`: Puerto del servidor de la base de datos.
- `sap_url_base`: URL base para la API de SAP.
- `api_key_sap`: Clave API para autenticarse con la API de SAP.
- `api_key_b2b`: Clave API para autenticarse con la API B2B.
- `b2b_url_base`: URL base para la API B2B.
- `toFactor`: Id de divisas usadas para conversión.
- `range_days`: Número de días a considerar para alguna operación.
- `interval_sync_db`: Intervalo (en minutos) entre sincronizaciones con la base de datos.
- `interval_sync_documents`: Intervalo (en minutos) entre sincronizaciones de documentos.
- `interval_sync_prices`: Intervalo (en formato hh:mm) entre sincronizaciones de precios.
- `active_sync_documents`: Booleano para activar o desactivar la sincronización de documentos.
- `manual_prices_sync`: Booleano para activar o desactivar la sincronización manual de precios.
- `NODE_ENV`: El entorno en el que se está ejecutando la aplicación (p. ej., desarrollo, producción).
- `subsidiary`: Código que identifica a la subsidiaria.
- `warehouses_sap`: Lista de almacenes a considerar en las consultas a la API de SAP.
- `price_list_table`: Configuración para el manejo de precios.

Figura 47 readme.md de configuraciones y credenciales de la API

Fuente Repositorio en Gitlab (2023).