InvestUBA
Revista del Postgrado de la Universidad Bicentenaria de Aragua
DIEP
Decanato de Investigación, Extensión y Postgrado.
Volumen 6, No. 2, Año 2023 (julio - diciembre)

DESARROLLO DE UNA API PARA LA GESTIÓN DE MENÚS Y PEDIDOS EN LÍNEA

Mauro Hernández[4]
Universidad Bicentenaria De Aragua
mauro.uba2018@gmail.com

Resumen
El presente proyecto tiene como objetivo el desarrollo de una API (Interfaz de Programación de Aplicaciones) destinada a la gestión eficiente de restaurantes y pedidos en línea. Esta API ofrece una solución integral para optimizar los procesos de gestión de restaurantes y facilitar la realización de pedidos en línea. Se busca mejorar la experiencia tanto para los restaurantes como para los clientes, ofreciendo una plataforma eficiente y fácil de usar. La API está diseñada con una estructura sólida y funcionalidades bien definidas, lo que permitirá a los desarrolladores integrar estas capacidades en sus propias aplicaciones de manera sencilla. Además, se proporcionará una documentación detallada para guiar a los desarrolladores en el proceso de implementación y utilización del sistema.

Palabras Claves: Interfaz de Programación de Aplicaciones, Gestión, Autorización, Cliente

Abstract
The present project aims to develop an API (Application Programming Interface) for efficient restaurant and online order management. This API provides a comprehensive solution to optimize restaurant management processes and facilitate online ordering. The goal is to enhance the experience for both restaurants and customers by offering an efficient and user-friendly platform. The API is designed with a robust structure and well-defined functionalities, allowing developers to easily integrate these capabilities into their own applications. Furthermore, detailed documentation will be provided to guide developers through the implementation and utilization of the system.

Keywords: API (Application Programming Interface), Restaurant management, Online orders, Efficiency, Authorized emplees, Menus, Dishes, Search and filtering, Customer experience, Integration, Documentation, Scalable.

[4] Ingeniero en Sistemas

InvestUBA
Revista del Postgrado de la Universidad Bicentenaria de Aragua
DIEP
Decanato de Investigación, Extensión y Postgrado.
Volumen 6, No. 2, Año 2023 (julio - diciembre)

Introducción

En la actualidad, con el auge de la era digital, los servicios de pedidos en línea han experimentado un impresionante crecimiento, revolucionando la interacción entre los restaurantes y sus clientes, así como la entrega de sus productos. Sin embargo, este avance también ha traído consigo retos y oportunidades para mejorar la eficiencia y la experiencia del usuario. Es en este contexto que surge un ambicioso proyecto: el desarrollo de una aplicación de pedidos en línea, cuyo enfoque radica en atender tanto las necesidades de los restaurantes como las de los desarrolladores.

Uno de los principales problemas actuales es la complejidad asociada a gestionar los pedidos en línea y los menús de los restaurantes de manera eficiente. A pesar del crecimiento, muchos establecimientos aún dependen de métodos tradicionales para recibir y procesar los pedidos, lo que puede derivar en retrasos, errores y una experiencia insatisfactoria para los clientes. Asimismo, los desarrolladores enfrentan desafíos al intentar integrar la funcionalidad de pedidos en línea en sus aplicaciones de forma ágil y efectiva.

Ante esta demanda y necesidad de mejora, el objetivo central de este proyecto se centra en crear una aplicación de pedidos en línea que optimice los procesos de gestión para los restaurantes, al tiempo que facilite la integración para los desarrolladores. La propuesta se basa en la creación de una API sólida y escalable, que permita a los restaurantes gestionar sus menús y recibir pedidos en línea de manera altamente eficiente.

La Propuesta

El proyecto tiene como objetivo desarrollar una aplicación de pedidos en línea que se centrará en atender tanto a restaurantes como a desarrolladores. Para ello, se creará una API sólida y escalable que permita a los restaurantes gestionar sus menús y recibir pedidos en línea de manera eficiente. Por otra parte, los desarrolladores podrán hacer uso de la API proporcionada por el proyecto para integrar en sus propias

aplicaciones, permitiendo que sus usuarios realicen pedidos en línea desde sus plataformas.

Justificación

La justificación para desarrollar esta API se fundamenta en la necesidad de optimizar y agilizar los procesos de creación de futuras aplicaciones relacionadas con la gestión de restaurantes y pedidos en línea. Al proporcionar una API completa y bien documentada, se simplifica y agiliza el trabajo de los desarrolladores al utilizar servicios preexistentes en lugar de crear todas las funcionalidades desde cero.

Además, esta API ofrece a los restaurantes una solución sencilla y eficiente para administrar sus menús y recibir pedidos en línea. Al utilizar la API, los restaurantes pueden integrar de manera fácil estas funcionalidades en sus propios sistemas, sin la necesidad de desarrollar y mantener una plataforma de pedidos compleja.

Producto Final

Se obtendrá una API que ofrezca una solución eficiente y escalable, que permita realizar los procesos de gestión y realización de pedidos de un restaurante. La API se centrará en proporcionar a los restaurantes una plataforma completa para administrar su información, crear y gestionar sus menús con detalles completos y precios, así como gestionar empleados autorizados para tareas específicas. Además, el sistema de autenticación garantizará la seguridad y control de accesos.

La API estará compuesta por cuatro módulos: gestión de restaurantes, gestión de menús, autenticación y módulo de pedidos/pagos. El módulo de restaurantes permitirá a los establecimientos registrar y mantener su información actualizada, incluyendo dirección, horarios y descripción del negocio. El módulo de menús ofrecerá herramientas de búsqueda y filtrado para facilitar a los clientes la exploración de los diferentes platos disponibles según sus preferencias. El módulo de autenticación se encargará de los aspectos de seguridad, roles y permisos.

Finalmente, el módulo de pedidos/pagos permitirá que los restaurantes reciban y administren de manera eficiente los pedidos realizados por los clientes mediante la

plataforma. Sin embargo, esta funcionalidad no pudo ser implementada en la presente versión del sistema debido a ciertas limitaciones que serán detalladas más adelante. No obstante, se ha registrado esta tarea en el backlog para ser incluida en futuras versiones del proyecto.

La API será diseñada siguiendo un esquema de microservicios, lo que garantizará la flexibilidad y escalabilidad del sistema. Los desarrolladores encontrarán una interfaz amigable y una documentación completa para integrar la API en sus propias aplicaciones, brindando a los usuarios finales una experiencia de pedidos en línea de alta calidad. En definitiva, este producto tiene como objetivo optimizar y facilitar la gestión de restaurantes y pedidos, proporcionando una solución completa y bien estructurada para la industria de la restauración en la era digital.

Objetivos

Este proyecto tiene como objetivo *"Desarrollar una API para la gestión de restaurantes y pedidos en línea, que permita optimizar los procesos de gestión, realización de pedidos, la administración, pagos y otros aspectos clave del negocio"*, del cual podemos desglosar en los siguientes puntos:

- Identificar las funciones clave del sistema, como gestión de restaurantes, pedidos y pagos. Establecer requisitos y definir el alcance del proyecto.
- Determinar la estructura del sistema, componentes y su interconexión.
- Diseñar la base de datos y desarrollar cada módulo con las funciones pertinentes para cumplir sus criterios de aceptación tanto de seguridad como de rendimiento.
- Desarrollar pruebas unitarias para cada módulo, además de realizar pruebas de integración para verificar la comunicación y el funcionamiento conjunto.
- Desplegar el sistema en sus diferentes entornos (prueba, preproducción y producción).

Criterios de Éxito y Parámetro de Calidad

El éxito del proyecto se alcanzará asegurando una funcionalidad completa de la API, es decir que esta ofrezca todas las funciones clave necesarias para administrar eficientemente un restaurante y mediante el despliegue exitoso de la misma en un entorno de producción, garantizando su disponibilidad y correcto funcionamiento.

A nivel técnico, la API debe poseer una estructura sólida y coherente con componentes bien definidos e interconectados, empleando tecnologías y herramientas adecuadas para asegurar un rendimiento óptimo y escalabilidad adecuada. Asimismo, la base de datos debe ser diseñada de manera efectiva, considerando las relaciones entre entidades y restricciones de integridad esenciales, y se deben implementar consultas y procedimientos almacenados eficientes para lograr un acceso rápido y seguro a los datos.

Es necesario seguir buenas prácticas de programación y considerar aspectos de seguridad, aplicando medidas adecuadas como encriptación y validación de entrada, con el fin de proteger los datos y prevenir posibles vulnerabilidades. Además, para garantizar el correcto funcionamiento del sistema se deben incluir test de aceptación para verificar la comunicación y funcionamiento conjunto de los módulos.

Entregables

Al finalizar el proyecto se entrega un repositorio de control de versiones alojado en la plataforma GitHub que contendrá el código fuente de la API desarrollada, permitiendo a los desarrolladores acceder, colaborar y realizar un seguimiento de los cambios realizados. Además de la documentación detallada de la API, ésta ofrecerá información sobre los endpoints disponibles, métodos HTTP admitidos, parámetros de entrada requeridos, formatos de datos y respuestas esperadas.

Se incluirá un informe detallado que mostrará los resultados de las pruebas realizadas, incluyendo pruebas unitarias, de integración y de rendimiento, para garantizar un funcionamiento correcto en diferentes escenarios. Y un manual con

instrucciones claras y concisas para instalar y configurar la API en distintos entornos, junto con los requisitos de software y hardware necesarios.

Duración e Hitos

El proyecto se planificó para desarrollarse en un periodo de tres meses (12 semanas). Principalmente se dividió en cuatro etapas: Lineamientos y Preparación, Desarrollo, Testing y Depuración, y Documentación y Entrega. Cada etapa tiene una duración específica, adaptada a la complejidad de las tareas involucradas en el proyecto. Podemos desglosar las etapas en la siguiente tabla:

Etapas del proyecto

Etapa	**Objetivos**	**Semanas**
Preparación	- Diseño de requerimientos. - Diseño y construcción de la DB. - Inicialización del repositorio.	3
Desarrollo	- Construcción y diseño de la lógica de negocios. - Diseño de rutas. - Desarrollo de servicios. - Integración de servicios.	4
Pruebas	- Diseño de pruebas unitarias. - Diseño de pruebas de integración. - Pruebas de seguridad y penetración.	3
Entrega	- Configuración de entornos. - Despliegue de sistemas. - Documentación del sistema.	2
Total		12

Autor: Hernández, M. (2023)

Costos

Se planificó una distribución de costos en tres áreas específicas: gastos generales, gastos de servicios y gastos de servidores. En cuanto a los gastos generales, se incluyen el salario de un programador especializado, licencias de software, equipos de computación y otros gastos relacionados. Los gastos de servicios abarcan el servicio eléctrico e internet, mientras que los gastos de servidores están asociados al uso de servicios de AWS (Amazon Web Services):

Para los gastos generales, el costo mensual estimado oscila entre $1,200 y $2,700, dependiendo de los recursos requeridos. Los gastos de servicios se estiman en $35 mensuales para cubrir internet y electricidad. En cuanto a los gastos de servidores en AWS, el costo mensual estimado varía entre $189.36 y $1,045.56, dependiendo del consumo de recursos. Además, se consideran gastos imprevistos con un fondo de $200 para enfrentar situaciones no previstas durante el desarrollo del proyecto.

En Total, el presupuesto total requerido para el proyecto puede fluctuar entre $1,624.36 y $3,980.56 mensuales, según las etapas y exigencias específicas del desarrollo.

Talento Humano

El proyecto se fundamenta en la importancia del talento humano y la variedad de habilidades necesarias para su éxito. Se destacan varios roles clave, incluyendo al analista de requisitos, quien se encarga de recopilar y documentar las necesidades del cliente y traducirlas en especificaciones técnicas. Luego, el desarrollador backend asume la responsabilidad de diseñar y mantener la lógica y funcionalidad del lado del servidor, empleando JavaScript y el framework NestJS para crear APIs robustas y escalables.

Por otro lado, el especialista en bases de datos (DBA) es esencial para garantizar un almacenamiento eficiente y seguro de los datos del proyecto, aplicando su experiencia en diseño de bases de datos relacionales y optimización de consultas. El

especialista en DevOps juega un papel fundamental al configurar y asegurar el despliegue adecuado de la aplicación, utilizando herramientas de automatización como Docker y Kubernetes para gestionar los contenedores.

Asimismo, el equipo cuenta con un especialista en pruebas que planifica, diseña y ejecuta pruebas de software para asegurar la calidad y el correcto funcionamiento de la aplicación.

Alcance de la Solución

El proyecto tiene como objetivo la creación de una API completa y escalable para la gestión eficiente de restaurantes y pedidos en línea. La API estará estructurada en módulos:

- En el primer módulo, los restaurantes podrán registrar su información y gestionar a sus empleados.
- El segundo módulo se enfocará en la gestión de menús, permitiendo a los restaurantes crear y personalizar sus opciones.
- El tercer módulo se encargará de la autenticación y seguridad, garantizando el acceso autorizado a la plataforma.
- El cuarto módulo facilitará los pedidos en línea y los pagos.

Además, se implementará una base de datos eficiente y tecnologías adecuadas para asegurar un rendimiento óptimo y escalabilidad. Las pruebas exhaustivas serán realizadas para asegurar la calidad y robustez de la API. Se proporcionará documentación completa para que los desarrolladores puedan utilizarla efectivamente.

Riesgos e Imprevistos

El servicio propuesto será de gran relevancia para los usuarios finales, ya que las funciones que ofrece son críticas para el modelo de negocios de nuestros usuarios. Teniendo esto en cuenta, podemos concluir que el sistema debe ser estable, capaz de manejar altos niveles de concurrencia y reducir al máximo los posibles fallos. En cada uno de estos puntos, existen riesgos que amenazan el correcto funcionamiento de la API. Después de un análisis exhaustivo, podemos clasificar estos riesgos en tres

grupos: problemas de escalabilidad y disponibilidad, problemas de seguridad y problemas operativos.

El sistema deberá manejar aumentos repentinos en la demanda, la falta de capacidad en los servidores, los errores en la configuración de la red y las fallas en el hardware, para esto se plantea una arquitectura la cual busca redundancia en los servicios para poder manejar cargas de trabajo y fallos repentinos. Las vulnerabilidades de seguridad debilidades potenciales en sistemas y aplicaciones pueden ser explotadas por atacantes malintencionados, comprometiendo la integridad, confidencialidad y disponibilidad de los datos. Para manejar estos riesgos se implementarán rigurosas validaciones, cifrados de datos y análisis de componentes/librerías.

Por último, los problemas operacionales vienen relacionados estrechamente con las fallas de servicios básicos, como lo son el servicio de internet y el servicio eléctrico, este tipo de inconvenientes tienen gran relevancia debido a que, si ocurrieran, provocaría que la operación de desarrollo, mantenimiento y producción de la API se detuviera por un periodo de tiempo. Para prevenir esto se implementaron múltiples proveedores de los distintos servicios, así como sistemas alternos.

Metodología Técnica

En el desarrollo de este proyecto, se utilizó una metodología ágil basada en el marco de trabajo Scrum. El equipo adoptó buenas prácticas de desarrollo de software en todas las etapas del proceso para garantizar la calidad y eficiencia del producto final. Se llevaron a cabo pruebas exhaustivas para verificar la correcta implementación de las funcionalidades requeridas, así como para asegurar la estabilidad y confiabilidad de la API.

Además, se realizaron revisiones periódicas de código entre los miembros del equipo para identificar y corregir posibles problemas y mejorar la coherencia del código. Asimismo, se contó con el asesoramiento técnico de expertos en áreas

específicas para abordar desafíos técnicos y asegurar el cumplimiento de los criterios de calidad y seguridad establecidos.

Aspectos Técnicos

Para la realización de este sistema (API), se optó por utilizar tecnologías de JavaScript, específicamente el framework NestJS, el cual ofrece características deseables para el proyecto. NestJS es una elección adecuada debido a su enfoque en la creación de aplicaciones escalables y bien estructuradas. Además, su uso de TypeScript proporciona una ventaja adicional al permitir un desarrollo más seguro y robusto.

En cuanto al gestor de base de datos, se eligió PostgreSQL gracias a que ofrece un alto rendimiento y una gran capacidad para manejar grandes volúmenes de datos, lo que es esencial para un sistema de gestión de restaurantes y pedidos en línea. Además, cuenta con sólidas características de seguridad y es compatible con NestJS, lo que facilita su integración y uso en el proyecto.

Esta combinación de tecnologías de JavaScript y PostgreSQL garantiza una base sólida para el desarrollo de la API, permitiendo crear un sistema eficiente, escalable y seguro para la gestión de restaurantes y pedidos en línea.

Se tomó la decisión de transportar este proyecto utilizando contenedores de Docker, los cuales nos ofrecen un ambiente de desarrollo y despliegue altamente portable y consistente. De esta manera, garantizamos que la aplicación funcione de manera uniforme en diferentes entornos y evitamos problemas de compatibilidad.

Además, para el despliegue final, optamos por utilizar instancias de AWS (Amazon Web Services) con conexión a un servidor S3 y RDS (Relational Database Service). Esta elección nos permite aprovechar la escalabilidad y la disponibilidad que AWS ofrece, asegurando que la aplicación esté disponible y funcione de manera eficiente en todo momento. La combinación de contenedores de Docker y el poder de AWS nos proporciona un enfoque moderno y robusto para el desarrollo y despliegue exitoso de este proyecto.

Además de configurar una instancia de Kubernetes para manejar la escalabilidad y el balanceo de carga de los contenedores, se implementarán estrategias avanzadas de gestión de recursos para optimizar el rendimiento del sistema. Mediante la utilización de Kubernetes, se logrará una distribución eficiente de la carga de trabajo, permitiendo que los contenedores se escalen automáticamente según la demanda.

Jerarquía de la Información y Reglas del Sistema La jerarquía de información se organiza de la siguiente manera:

- Restaurantes:
  - Cada restaurante es único en el sistema y se identifica mediante un ID único.
  - Cada restaurante tiene asignado un único usuario administrador, que se registra como un empleado en la tabla de empleados.
  - Los restaurantes pueden tener múltiples empleados trabajando para ellos.
  - Los restaurantes pueden tener N registros de menús.
- Empleados:
  - Cada empleado se identifica mediante un ID único y se asocia a un restaurante específico mediante el ID del restaurante al que pertenece.
  - Cada empleado tiene su propio registro de información, que incluye datos como nombre, apellido, correo electrónico, entre otros.
  - Cada empleado tiene un registro de sus credenciales para el acceso al sistema.
  - Cada empleado tiene asociado un rol.
  - Cada empleado puede modificar su información personal, pero únicamente el usuario administrador del restaurante al que pertenece puede modificar su información sensible.
- Menús:
  - Cada restaurante puede tener múltiples menús, cada uno identificado por un ID único.

ISSN: 2739-0500 Depósito Legal: AR2021000099

- Cada menú contiene sus detalles asociados, como nombre, descripción, precio y estado (activo o inactivo).
- Los empleados con los permisos necesarios pueden crear y modificar los menús del restaurante al que están asociados.

- Roles:
  - Un rol puede tener asociados múltiples permisos.

Estructura de Base de Datos

La estructura de la base de datos está compuesta por siete tablas, que son las siguientes:

- Restaurantes: Contiene la información general de los restaurantes.
- Menús: Almacena la información relevante para los menús.
- Employees: Contiene los datos generales de los empleados.
- Auth: Guarda las credenciales e información sensible de los empleados.
- Roles: Sirve como referencia a los permisos que posee un usuario.
- Permissions: Define las acciones disponibles en el sistema.
- Roles_Permissions: Contiene las relaciones entre los roles y los permisos.

Para interactuar con la base de datos, se utiliza el ORM Prisma, el cual proporciona amplias funcionalidades para llevar a cabo operaciones en la base de datos. Además, este ORM (Object-Relational Mapping) permite montar la base de datos en diferentes entornos a partir de los modelos y las migraciones.

Para gestionar la base de datos se utilizó PGAdmin, esta es una herramienta de administración de bases de datos de código abierto que se utiliza específicamente para gestionar bases de datos PostgreSQL. Proporciona una interfaz gráfica fácil de usar que permite a los usuarios realizar diversas tareas relacionadas con la base de datos, como la creación y modificación de tablas, vistas y funciones, así como la ejecución de consultas y la supervisión del rendimiento. A continuación, se presenta el diagrama de la estructura de la base de datos:

InvestUBA
Revista del Postgrado de la Universidad Bicentenaria de Aragua
DIEP
Decanato de Investigación, Extensión y Postgrado.
Volumen 6, No. 2, Año 2023 (julio - diciembre)

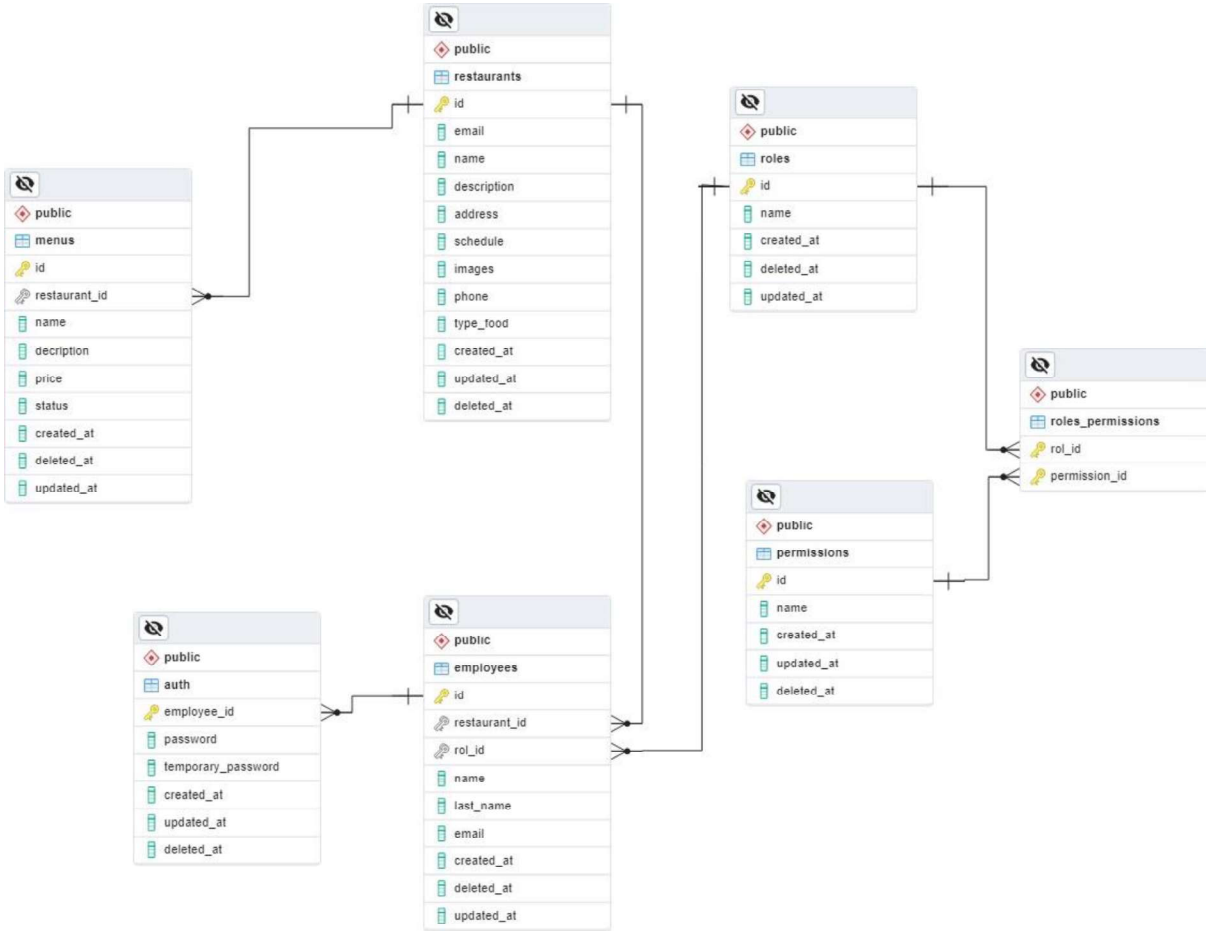

Figura 1. Diagrama de la estructura de la base de datos

ISSN: 2739-0500 Depósito Legal: AR2021000099

Análisis De Resultados

Durante el período de tres meses de desarrollo, alcanzamos las metas establecidas para la creación de una API completa y escalable para la gestión eficiente de restaurantes y pedidos en línea. Se logró implementar con éxito el primer y segundo módulo, permitiendo a los restaurantes registrar su información y gestionar sus menús de manera efectiva. Sin embargo, el módulo de gestión de pedidos se pospuso para la versión 2.0 debido a problemas surgidos durante su desarrollo.

Uno de los principales problemas que surgieron en el módulo de pedidos fue la integración de los métodos de pago en la plataforma. Implementar una solución segura y confiable para que los usuarios pudieran realizar pagos en línea de manera conveniente resultó ser más complejo de lo esperado.

Por otro lado, se estableció con éxito el tercer módulo, encargado de la autenticación y seguridad, asegurando el acceso autorizado a la plataforma mediante mecanismos de autenticación sólidos. La infraestructura del proyecto, incluida una base de datos eficiente y tecnologías adecuadas, garantiza un rendimiento óptimo y escalabilidad adecuada. Asimismo, se llevaron a cabo pruebas exhaustivas para asegurar el correcto funcionamiento y cumplimiento de los requisitos establecidos.

Metas	
Registro de la información del restaurante	COMPLETADO
Gestión de empleados	COMPLETADO
Gestión de menús	COMPLETADO
Gestión de pedidos	NO COMPLETADO
Autenticación y seguridad	COMPLETADO
Implementación de base de datos	COMPLETADO
Realización de pruebas	COMPLETADO
Documentación de la API	COMPLETADO

InvestUBA
Revista del Postgrado de la Universidad Bicentenaria de Aragua
DIEP Decanato de Investigación, Extensión y Postgrado.
Volumen 6, No. 2, Año 2023 (julio - diciembre)

Análisis De Calidad

El desarrollo de la API para la gestión de operaciones en un restaurante fue llevado a cabo de manera exitosa y cumplió con los criterios de funcionalidad, interacción y eficiencia, escalabilidad, flexibilidad, documentación, seguridad, protección contra ataques y vulnerabilidades, estabilidad y confiabilidad. Durante el proceso, se realizaron pruebas exhaustivas para verificar el correcto funcionamiento de todas las funcionalidades requeridas.

El diseño de la API demostró una capacidad excepcional para manejar un alto volumen de transacciones y consultas simultáneas, adaptándose de manera eficiente a medida que aumentaba la demanda. La arquitectura modular y orientada a servicios permitió la incorporación de nuevas características sin interrupciones, lo que brinda una solución flexible para futuras expansiones y actualizaciones.

La documentación elaborada fue completa y precisa, proporcionando una guía detallada sobre las funcionalidades disponibles, endpoints, parámetros y métodos de autenticación, facilitando su implementación y utilización.

En términos de seguridad, la API implementa medidas sólidas de autenticación, autorización y protección contra vulnerabilidades y posibles ataques, lo que garantiza la integridad y confidencialidad de los datos y la información sensible. Las pruebas de resistencia y gestión de errores aseguraron la estabilidad y confiabilidad de la API en el entorno de producción, evitando fallos y asegurando una experiencia de usuario consistente.

Conclusión

En resumen, el proyecto de desarrollo de la aplicación de pedidos en línea para restaurantes y desarrolladores fue un logro sobresaliente que superó las expectativas. Durante tres meses, se trabajó diligentemente para completar con éxito la API, que se estructuró en tres módulos clave: restaurantes, autenticación y gestión de menús.

ISSN: 2739-0500 Depósito Legal: AR2021000099

Aunque hubo dificultades con la integración de los métodos de pago en el módulo de gestión de pedidos, el equipo tomó una decisión acertada al posponer esta funcionalidad para una futura versión, lo que permitió enfocarse en otros módulos. Las pruebas exhaustivas contribuyeron al éxito, asegurando el cumplimiento de los estándares de rendimiento.

La documentación detallada y el código abierto facilitan la integración y colaboración de otros desarrolladores. El equipo humano desempeñó un papel fundamental con su compromiso, experiencia y comunicación efectiva, impulsando el proyecto hacia el éxito.

Referencias

Universidad Bi centenaria de Aragua (2020). Manual para la elaboración. Presentación y evaluación del trabajo de grado y tesis doctoral de los programas de postgrado. San Joaquín de Turmero, fundó Editorial UBA.

Project Management Institute, Inc. (2017). La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK). Newton Square, Pennsylvania 190733299 EE. UU

NestJS (2023). A progressive Node.js framework. Recuperado 28 de mayo de 2023 en: https://docs.nestjs.com

RedHat (2023). ¿Qué es una API y cómo funciona? Recuperado 28 de mayo de 2023 en: https://www.redhat.com/es/topics/api/what-are-applicationprogramming-interfaces

Open Source Initiative (2023). MIT sin licencia de atribución. Recuperado el 12 de junio de 2023 en: https://opensource.org/license/mit-0/

Debian.org (2022). Contrato social de Debian. Recuperado el 12 de junio de 2023 en: https://www.debian.org/social_contract#guidelines

Microsoft. (2021). Azure Documentation. Recuperado el 12 de junio de 2023, de https://docs.microsoft.com/en-us/azure/

Amazon Web Services. (2021). AWS Documentation. Recuperado el 12 de junio de 2023, de https://docs.aws.amazon.com/index.html

Syntonize (2022). Stacks de Desarrollo web Y ejemplos. Recuperado el 15 de junio de 2023, de https://www.syntonize.com/stack-tecnologico-que-es-y-como-crearuno

Prisma.io (2021). Next-generation Node.js and TypeScript ORM. Recuperado el 27 de julio del 2023, de https://www.prisma.io/.