

Contrato de Servicios Web para el Sistema de Gestión de Restaurante

Fecha de Vigencia: 25 de octubre de 2024

Autorizado por:

Mariana Osorio Rojas

Sebastián Restrepo Yepes

Arquitectura de Software

Facultad de Las Tecnologías de la Información y la Comunicación

Universidad Pontificia Bolivariana

Link del Repositorio:

<https://github.com/MinervaStarfish/restaurant-service>

Tabla de Contenido

| | |
|--|----|
| 1. Nombre del Servicio | 3 |
| 2. Descripción del Servicio..... | 3 |
| 3. Interfaces y Protocolos | 3 |
| 4. Operaciones del Servicio | 4 |
| 5. Parámetros de la Operación | 5 |
| 6. Formato de Datos | 5 |
| 7. Autenticación y Seguridad..... | 5 |
| 8. Respuestas y Códigos de Estado | 7 |
| 9. Manejo de Errores..... | 7 |
| 10. Ejemplos y Escenarios de Uso | 8 |
| 11. Requisitos de Rendimiento y Tiempo de Respuesta..... | 9 |
| 12. Versionado | 10 |
| 13. Acuerdo de Nivel de Servicio (SLA) | 10 |
| 14. Responsabilidades de las Partes..... | 12 |
| 15. Fecha de Vigencia y Actualización..... | 13 |
| 16. Firma y Aprobaciones..... | 13 |

1. Nombre del Servicio

API de Gestión de Restaurante

Este servicio permite la administración completa de las operaciones de un restaurante; incluyendo la gestión de pedidos, visualización de ítems de menú, y el control del estado de las mesas.

2. Descripción del Servicio

La **API de Gestión de Restaurante**, está diseñada para facilitar la administración de un restaurante mediante una serie de operaciones que se integran en una aplicación web o móvil. A través de la API, los usuarios pueden realizar las siguientes operaciones:

- **Consultar y administrar el menú.**
- **Gestionar pedidos y órdenes de los clientes.**
- **Actualizar y consultar el estado de las mesas del restaurante.**

El objetivo principal del proyecto, es brindar una plataforma eficiente para automatizar las operaciones del restaurante, optimizar el tiempo de atención y mejorar la experiencia del cliente.

3. Interfaces y Protocolos

- **Protocolo de Comunicación:** HTTP/1.1, RESTful API.
- **Base URL:** http://localhost:3000
- **Formato de Datos:** JSON (JavaScript Object Notation).
- **Métodos HTTP:** Utiliza métodos HTTP estándar (GET, POST, PUT, DELETE).

4. Operaciones del Servicio

Servicio 1: Menú

1. **Operación 1.1:** Obtener todos los ítems del menú.
 - **Endpoint:** /menu
 - **Método:** GET
 - **Descripción:** Devuelve la lista completa de los ítems del menú.
2. **Operación 1.2:** Consultar un ítem específico del menú.
 - **Endpoint:** /menu/:id
 - **Método:** GET
 - **Descripción:** Devuelve los detalles del ítem del menú solicitado según su ID.

Servicio 2: Órdenes

1. **Operación 2.1:** Crear una nueva orden.
 - **Endpoint:** /orders
 - **Método:** POST
 - **Descripción:** Permite la creación de una nueva orden de pedido.
2. **Operación 2.2:** Consultar todas las órdenes.
 - **Endpoint:** /orders
 - **Método:** GET
 - **Descripción:** Devuelve una lista de todas las órdenes en proceso.

Servicio 3: Mesas

1. **Operación 3.1:** Consultar el estado de todas las mesas.
 - **Endpoint:** /tables

- **Método:** GET
 - **Descripción:** Devuelve el estado actual de cada mesa.
2. **Operación 3.2:** Actualizar el estado de una mesa.
- **Endpoint:** /tables/:tableNumber
 - **Método:** PUT
 - **Descripción:** Permite actualizar el estado de una mesa específica, como disponible, ocupada o reservada.

5. Parámetros de la Operación

- **/menu/:** id(número entero que representa el ID único del ítem del menú).
- **/orders:** JSON con estructura de pedido.
- **/tables/:** tableNumber (número entero que identifica la mesa a actualizar).

6. Formato de Datos

- **Solicitud y Respuesta:** JSON

7. Autenticación y Seguridad

1. Autenticación

- **Estado Actual:**
 - En la versión inicial del servicio, no se implementa ningún mecanismo de autenticación. Esto permite facilitar el acceso y la integración inicial con el servicio.
- **Recomendaciones para Futuras Versiones:**

- Para las versiones futuras del servicio, se recomienda implementar autenticación basada en **JSON Web Tokens (JWT)**. Esta metodología proporcionará un método seguro y escalable para autenticar a los usuarios y asegurar que solo los consumidores autorizados puedan acceder a las funciones sensibles del servicio.
- **Proceso de Autenticación Propuesto:**
 - Los usuarios se autenticarán enviando sus credenciales (nombre de usuario y contraseña) a un endpoint de autenticación.
 - Tras una autenticación exitosa, se generará un token JWT que deberá ser incluido en las cabeceras de las solicitudes posteriores al servicio.

2. Seguridad

- **Protocolo de Comunicación:**
 - Todas las comunicaciones entre el cliente y el servicio deben realizarse a través de **HTTPS**. Esto asegura que los datos en tránsito estén cifrados, protegiendo la información contra intercepciones y ataques de tipo "man-in-the-middle".
- **Buenas Prácticas de Seguridad:**
 - Implementar medidas de protección contra ataques comunes, como **inyecciones SQL, cross-site scripting (XSS) y cross-site request forgery (CSRF)**.
 - Realizar auditorías de seguridad periódicas para identificar y remediar posibles vulnerabilidades.
- **Gestión de Errores:**
 - Evitar la exposición de información sensible en los mensajes de error devueltos al cliente. En su lugar, proporcionar mensajes de error genéricos que no revelen detalles del sistema.

8. Respuestas y Códigos de Estado

- **200 OK:** Solicitud procesada correctamente.
- **201 Created:** Orden creada (solo en creación de órdenes).
- **400 Bad Request:** Parámetros incorrectos.
- **404 Not Found:** Recurso no encontrado.
- **500 Internal Server Error:** Error interno del servidor.

9. Manejo de Errores

1. Elementos de la Respuesta de Error

- **Código de Error:** Un identificador único del tipo de error ocurrido, que permite categorizar el problema (ej., ITEM_NOT_FOUND).
- **Mensaje:** Una descripción clara y concisa del error que se produjo, diseñada para ser comprensible por el consumidor del servicio.
- **Timestamp:** La fecha y hora en la que se generó el error, en formato ISO 8601, que ayuda en la auditoría y seguimiento de problemas.
- **Detalles Adicionales:** Información adicional que puede incluir el requestId, que permite al proveedor del servicio rastrear la solicitud específica que causó el error, y la path, que indica la URL que fue solicitada.

2. Códigos de Estado HTTP

- El servicio devolverá códigos de estado HTTP apropiados junto con la respuesta de error para indicar la naturaleza del problema:
 - **404 Not Found:** Cuando el recurso solicitado no existe.
 - **400 Bad Request:** Cuando la solicitud no se puede procesar debido a parámetros inválidos.
 - **500 Internal Server Error:** Para errores no manejados en el servidor.

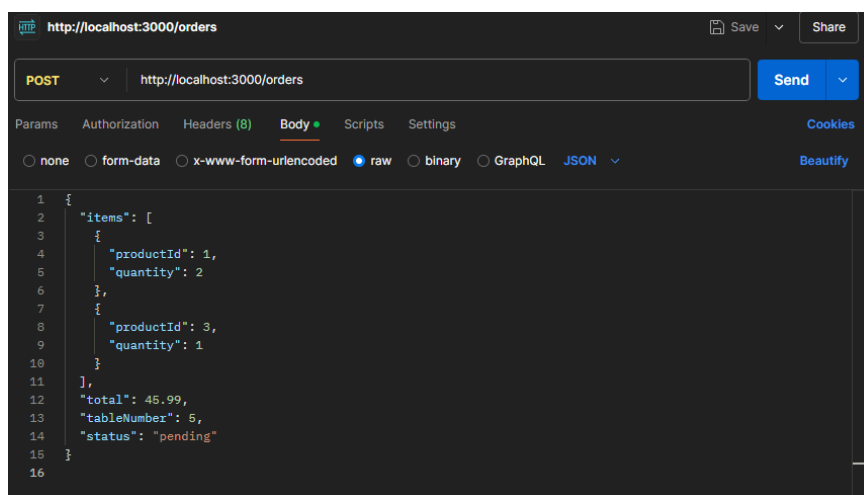
3. Manejo de Errores en el Cliente

- Los consumidores del servicio deben implementar un manejo de errores adecuado en su aplicación para procesar las respuestas de error de manera efectiva, proporcionando a los usuarios finales información útil y opciones de recuperación.

10. Ejemplos y Escenarios de Uso

Ejemplo de Crear una Orden

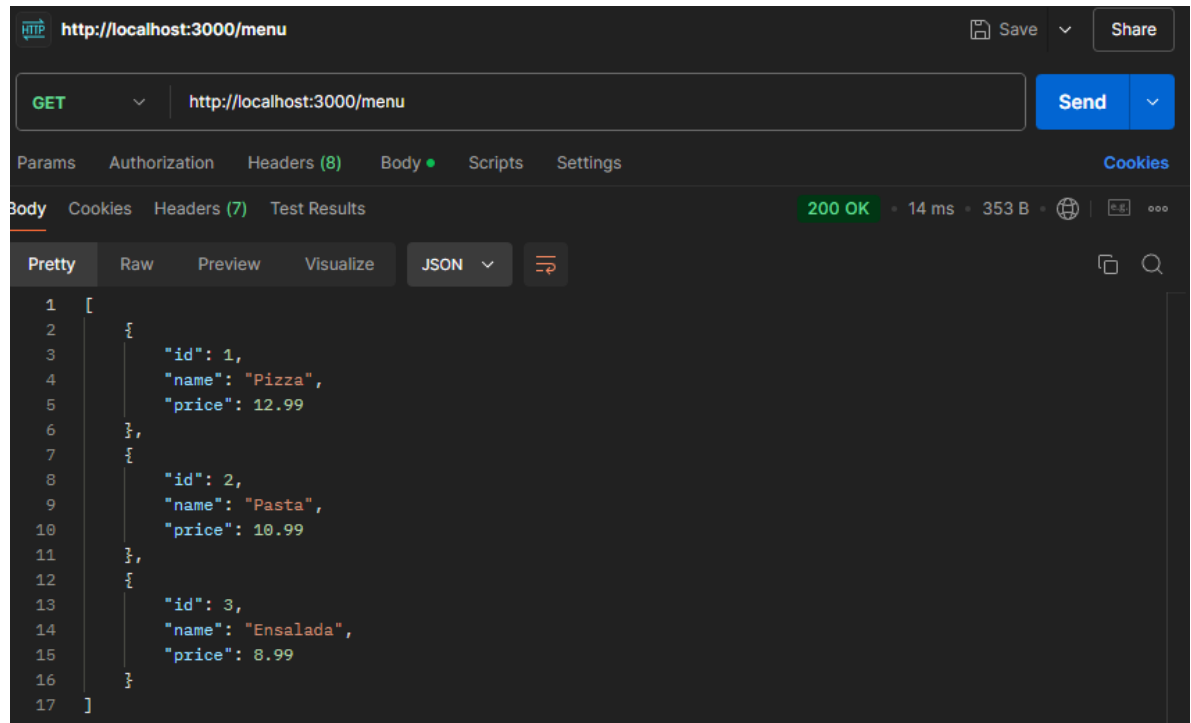
- Solicitud:** POST <http://localhost:3000/orders>
- Body:**



```
1 {
2   "items": [
3     {
4       "productId": 1,
5       "quantity": 2
6     },
7     {
8       "productId": 3,
9       "quantity": 1
10    }
11  ],
12  "total": 45.99,
13  "tableNumber": 5,
14  "status": "pending"
15 }
16 }
```

Ejemplo de Consultar Menú Completo

- Solicitud:** GET <http://localhost:3000/menu>
- Respuesta:**



11. Requisitos de Rendimiento y Tiempo de Respuesta

1. Tiempo de Respuesta

- **Compromiso de Tiempo de Respuesta:**
 - El servicio debe garantizar un tiempo de respuesta inferior a 2 segundos para el 95% de todas las solicitudes procesadas.
 - Este tiempo de respuesta se medirá desde que se recibe la solicitud hasta que se envía la respuesta al consumidor.

2. Capacidad de Solicitudes

- **Límite de Solicitudes Concurrentes:**
 - El servicio debe ser capaz de manejar un mínimo de 60 solicitudes por minuto, asegurando que pueda atender picos de demanda sin degradar el rendimiento.

- Se implementará un mecanismo de escalabilidad para gestionar un mayor volumen de solicitudes en situaciones de alta carga, con el objetivo de mantener la eficiencia y la disponibilidad del servicio.

3. Pruebas de Rendimiento

- Se realizarán pruebas de rendimiento periódicas utilizando herramientas como Apache JMeter o Gatling para validar el cumplimiento de estos requisitos.
- Las pruebas incluirán escenarios de carga simulando hasta el doble de la capacidad esperada, evaluando tanto el tiempo de respuesta como la estabilidad del servicio.

12. Versionado

- **Versión Actual:** v1.0.0
- **Estrategia de Versionado:** Semántico (MAJOR.MINOR.PATCH).
- **Política de Actualización:** Se notificarán cambios incompatibles con 30 días de anticipación.

13. Acuerdo de Nivel de Servicio (SLA)

- **Disponibilidad:** 99.5%
- **Tiempo de Respuesta Promedio:** 1 segundo.
- **Horas de Operación:** 24/7
- **Tiempo de Respuesta a Incidencias:** 4 horas.

1. Disponibilidad

- **Compromiso de Disponibilidad:** El servicio garantiza una disponibilidad del 99.5% durante un periodo mensual.
- **Método de Verificación:**
 - Se implementará un sistema de monitoreo continuo utilizando herramientas como **UptimeRobot** o **Pingdom** para realizar verificaciones de disponibilidad cada 1 o 5 minutos.
 - Se calculará el porcentaje de uptime mensual. La disponibilidad se considerará cumplida si se mantiene por encima del 99.5%.

2. Tiempo de Respuesta Promedio

- **Compromiso de Tiempo de Respuesta:** El servicio asegura un tiempo de respuesta promedio de 1 segundo para las solicitudes.
- **Método de Verificación:**
 - Se llevarán a cabo pruebas de carga utilizando herramientas como **Apache JMeter**, **k6**, o **Locust** para medir el tiempo de respuesta promedio en condiciones de carga normal.
 - Las herramientas de monitoreo también registrarán el tiempo de respuesta promedio en producción, asegurando que se mantenga dentro del límite establecido.

3. Horas de Operación

- **Compromiso de Horas de Operación:** El servicio estará disponible 24 horas al día, 7 días a la semana (24/7).
- **Método de Verificación:**
 - Se utilizarán herramientas de monitoreo para garantizar que el servicio esté disponible en todo momento.
 - Se configurarán alertas para detectar y notificar cualquier inactividad no programada.

4. Tiempo de Respuesta a Incidencias

- **Compromiso de Respuesta a Incidencias:** El tiempo de respuesta para incidencias será de un máximo de 4 horas.
- **Método de Verificación:**
 - Se establecerá un sistema de notificación para alertar al equipo sobre incidencias. Se utilizarán herramientas como **PagerDuty** o **Slack** para facilitar la comunicación y respuesta.
 - Se llevará un registro de incidencias documentando las marcas de tiempo de inicio y respuesta, permitiendo la evaluación del cumplimiento del SLA en cuanto al tiempo de respuesta.

14. Responsabilidades de las Partes

1. Responsabilidades del Proveedor del Servicio

- **Mantenimiento del Servicio:**
 - Asegurar la disponibilidad continua del servicio, garantizando un tiempo de actividad del 99.5%.
 - Realizar actualizaciones y mejoras de manera regular para optimizar el rendimiento y la seguridad del servicio.
- **Resolución de Incidencias:**
 - Investigar y resolver de manera proactiva los errores y problemas reportados por los consumidores del servicio en un plazo no mayor a 4 horas.
 - Proporcionar comunicación oportuna a los consumidores sobre el estado de las incidencias reportadas y las acciones tomadas para su resolución.
- **Documentación:**
 - Mantener actualizada la documentación técnica y de usuario del servicio, asegurando que esté disponible para los consumidores.

2. Responsabilidades del Consumidor del Servicio

- **Configuración de Solicitudes:**
 - Configurar y enviar solicitudes adecuadas y válidas al servicio, siguiendo las especificaciones y formatos establecidos en el contrato.
- **Notificación de Errores:**
 - Informar al proveedor del servicio sobre cualquier error, incidencia o anomalía en el funcionamiento del servicio de manera inmediata.
 - Proporcionar detalles claros y completos sobre el problema encontrado para facilitar su diagnóstico y resolución.
- **Cumplimiento de Requisitos:**
 - Asegurarse de que su entorno (software y hardware) cumpla con los requisitos necesarios para el correcto funcionamiento del servicio.
 - Seguir las mejores prácticas recomendadas en la documentación para maximizar la eficacia del servicio.

15. Fecha de Vigencia y Actualización

- **Fecha de Vigencia:** 25 de Octubre de 2024.
- **Actualización:** Revisión anual o en cada versión.

16. Firma y Aprobaciones

- **Autorizado por:**
 - Mariana Osorio Rojas
 - Sebastián Restrepo Yepes