**Orden de Servicio**

**Datos del Cliente**

|  |  |
| --- | --- |
| **Nombre** | *Omar Cabrera* |
| **Rut** | *9.999.999-0* |
| **Teléfono** | *9 89786756* |
| **Email** | *omar.cabrera@inacapmail.cl* |

**Datos de los Consultores técnicos** *(Coloque aquí la identificación de los integrantes del grupo)*

|  |  |
| --- | --- |
| **Nombre** | Fernando Carrasco Mellado |
| **Rut** | 21.558.167-5 |
| **Nombre** | Ignacio Colun Torres |
| **Rut** | 21.432.304-4 |
| **Nombre** | Alejandro Peña Lagos |
| **Rut** | 21.210.090-1 |
| **Nombre** | Issa Peña Mondaca |
| **Rut** | 20.840.456-3 |
| **Nombre** | Eduardo Rapiman Acuña |
| **Rut** | 21.008.805-9 |

**Observaciones generales**

|  |
| --- |
| Se va a desarrollar un sitio web para un servicio de venta de pizzas. Esta página permitirá a los usuarios hacer pedidos en línea, los cuales serán entregados directamente en sus domicilios. El sitio ofrecerá un amplio menú de pizzas, y se planea incluir funciones adicionales, como la opción de personalizar las pizzas para destacar frente a la competencia. También se implementarán recompensas para motivar a los clientes a realizar pedidos frecuentes y fomentar la lealtad hacia la marca. |

**Descripción del Servicio:**

|  |
| --- |
| **Para efectuar el desarrollo de la aplicación se requiere del siguiente entorno de trabajo:**  *1.- Describa la infraestructura necesaria para la implementación de la aplicación*:  el servidor web, que entrega contenido a los usuarios; el hosting, que proporciona espacio en un servidor; el dominio, que es la dirección web; y el DNS, que traduce nombres de dominio en direcciones IP. Además, se requiere una base de datos para gestionar datos, frontend para la interfaz de usuario, backend para procesar la lógica de la aplicación, y APIs para integrar diferentes sistemas. La integración de todos estos elementos asegura que la página web funcione de manera eficiente, sea segura y ofrezca una buena experiencia de usuario.  *2.- Describa la configuración necesaria del hardware para que funcione la aplicación*:  **Requisitos de Hardware**  -Servidores web y del sitio  CPU: 4vCPUs  RAM: 8GB  Almacenamiento: 100GB SSD  Cantidad: 2 (Balanceo de carga y alta disponibilidad)  Justificación: Dos servidores con alta configuración soportaran los 50 usuarios concurrentes, distribuyendo la carga entre ellos y proporcionando redundancia en caso de fallos.  -Servidor de base de datos  CPU: 8 vCPUs  RAM: 16GB  Almacenamiento: 50GB SSD  Cantidad: 1  Justificación: Para mejorar el rendimiento y reducir la carga en la base de datos, el servidor de caché almacenara datos de acceso frecuente.  -Servidor de Archivos/Medios  CPU: 2 vCPUs  RAM: 4 GB  Almacenamiento: 200 GB SSD  Cantidad: 1  Justificación: Almacenar y servir imágenes y otros archivos estáticos desde un servidor dedicado mejora la eficiencia y la rapidez del sitio.  -Balanceador de Carga  Cantidad: 1  Justificación: Distribuir el tráfico entre los servidores web y de aplicación, asegurando que ningún servidor individual se sobrecargue.  Diseño de la Aplicación  Arquitectura  Frontend (Cliente):  Tecnologías: HTML, CSS, JavaScript  Responsivo: Adaptable a dispositivos móviles y de escritorio.  Backend (Servidor):  Tecnologías: Node.js, Django, Ruby on Rails, o similar.  API para la comunicación con el Frontend.  Base de Datos:  Relacional: MySQL para transacciones.  NoSQL: MongoDB para datos no estructurados.   1. *- Describa el software necesario para el desarrollo, y posterior implementación de la aplicación*:  Herramientas de Desarrollo Para el desarrollo antes de la publicación, utilizaremos las siguientes herramientas:   * **Visual Studio Code**: Un editor de código fuente potente y ligero que soporta diversas extensiones para mejorar el desarrollo. * **XAMPP**: Una distribución de Apache que incluye MySQL, PHP y Perl, útil para el desarrollo y pruebas locales del servidor. * **Adobe Photoshop**: Para el diseño y edición de imágenes, creando gráficos y elementos visuales de alta calidad para la interfaz de usuario.  Frontend Para el frontend se requiere una interfaz intuitiva y atractiva para proporcionar una experiencia de usuario óptima. Utilizaremos las siguientes tecnologías: Librerías/Frameworks:  * + - **React.js**: Es una biblioteca de JavaScript para construir interfaces de usuario. Es conocida por su rendimiento y modularidad, lo que facilita el desarrollo y mantenimiento de componentes reutilizables.     - **Bootstrap**: Es un framework de CSS que simplifica el diseño responsivo y la creación de una interfaz atractiva y funcional.   **Justificación**: React.js es ideal para aplicaciones dinámicas y ofrece una experiencia de usuario fluida. Bootstrap nos ayudará a crear interfaces responsivas, modernas y minimalistas. Backend El backend maneja la lógica del servidor, la base de datos y la autenticación. Necesitamos una arquitectura robusta y escalable, por lo cual utilizaremos: Lenguaje y Frameworks:  * + - **Node.js con Express.js**: Node.js permite ejecutar JavaScript en el servidor y Express.js es un framework minimalista y flexible para construir aplicaciones web.  Base de Datos:  * + - **MongoDB**: Utilizaremos una base de datos NoSQL orientada a documentos ya que es flexible y permite almacenar datos semi-estructurados, lo cual es útil para almacenar información de pedidos y usuarios.   **Justificación**: Node.js y Express.js son conocidos por su alto rendimiento y escalabilidad, lo que es esencial para una aplicación que puede tener un alto volumen de tráfico. MongoDB es altamente escalable y permite cambios rápidos en la estructura de datos, lo que es beneficioso en una aplicación con posibles cambios frecuentes en el menú y la personalización de pizzas. Autenticación y Autorización Para manejar la autenticación de usuarios y la autorización de roles, implementaremos:   * + **JSON Web Tokens (JWT)**: Para asegurar las sesiones de usuario y manejar la autenticación de manera segura y eficiente.   **Justificación**: JWT es una solución ligera y segura para manejar la autenticación y autorización en aplicaciones web. Despliegue y Hosting La infraestructura para desplegar y hospedar la aplicación debe ser escalable y confiable. Utilizaremos:   * + **AWS (Amazon Web Services)**: Servicios como EC2 para servidores, S3 para almacenamiento y RDS para bases de datos gestionadas.   **Justificación**: AWS nos proporciona una infraestructura más robusta y escalable, esperando así un crecimiento a largo plazo. Seguridad La seguridad es crítica para proteger los datos del usuario y garantizar transacciones seguras. Implementaremos:   * + **SSL/TLS**: Para encriptar la comunicación entre el cliente y el servidor.   + **OWASP Security Practices**: Implementar prácticas de seguridad recomendadas por OWASP para prevenir vulnerabilidades comunes.  Integración de Pagos Para manejar los pagos en línea de manera segura, utilizaremos:   * + **PayPal**: Una plataforma de pagos segura y fácil de integrar.   **Justificación**: PayPal es un servicio confiable y ampliamente utilizado que simplifica la integración de pagos en línea. Funcionalidades Adicionales  * + **Sistema de Recompensas y Fidelización**:     - **Base de datos para recompensas**: Puede ser parte de la misma base de datos principal (MongoDB) o una base de datos separada si la aplicación crece significativamente.   + **Justificación**: Implementar un sistema de recompensas integrado en la misma base de datos simplifica el desarrollo y la gestión de datos.  Personalización de Pizzas:  * + - **Configurador de Productos**: Implementar un módulo en el frontend para personalizar ingredientes y características de las pizzas.   + **Justificación**: Ofrecer un configurador en tiempo real mejora la experiencia del usuario al permitir la visualización inmediata de las personalizaciones de su pedido. |