**PROPUESTA TÉCNICA DE UN PROYECTO DE SOFTWARE**

Una propuesta técnica de un proyecto de software debe estar estructurada de manera clara y detallada para que el cliente pueda comprender fácilmente los aspectos técnicos del proyecto. A continuación, se presenta una posible estructura para una propuesta técnica:

1. Resumen ejecutivo: El proyecto se centra en el desarrollo de un aplicativo web para un supermercado, con el objetivo de mejorar la experiencia de compra de los clientes y aumentar la eficiencia operativa. Este aplicativo permitirá a los usuarios realizar compras en línea de manera conveniente y segura, acceder a promociones exclusivas, recibir recomendaciones personalizadas y gestionar sus pedidos de forma eficiente. Tener un control en el inventario de entrada y salida de los productos. Los beneficios incluyen mayor comodidad para los clientes, aumento de las ventas y optimización de los procesos internos del supermercado, mejor experiencia del cliente, eficiencia operativa, reducción de costos, análisis de datos para mejorar la toma de decisiones y mayor competitividad en el mercado.
2. Introducción: El proyecto PowerFull Market se enmarca en la creciente demanda de compras en línea y la necesidad de adaptación de los supermercados tradicionales al entorno digital. También, presenta pérdidas en el inventario y descuadre de caja. La motivación detrás del desarrollo del software surge de la necesidad de ofrecer a los clientes una alternativa conveniente y accesible para realizar sus compras de alimentos y productos de primera necesidad. También hacer un control de inventario dentro del supermercado. El cliente busca satisfacer la demanda de un mercado cada vez más digitalizado, brindando una plataforma que no solo facilite las compras en línea, sino que también mejore la experiencia de usuario, fomente la fidelización y optimice la gestión de inventario y logística interna.
3. Objetivos del proyecto: Implementar un sistema de gestión de inventario en tiempo real para reducir pérdidas y optimizar el stock.

Desarrollar un programa de fidelización de clientes para aumentar la retención y las ventas recurrentes.

Integrar herramientas de análisis de datos para entender mejor las preferencias del cliente y ajustar la oferta de productos.

Mejorar la seguridad de los datos del cliente y del negocio para garantizar la confianza y la privacidad.

Automatizar procesos de gestión de compras y proveedores para reducir costos operativos.

Facilitar el análisis de datos para tomar decisiones estratégicas basadas en información precisa y oportuna.

1. Alcance del proyecto: El proyecto PowerFull Market incluirá las siguientes funcionalidades principales:

1. Catálogo de Productos: Presentación detallada de los productos disponibles, con imágenes, descripciones y precios actualizados.

2. Carrito de Compras: Capacidad para que los usuarios añadan y gestionen los productos que desean comprar.

3. Proceso de Pago Seguro: Integración de pasarelas de pago seguras para garantizar transacciones confiables.

4. Gestión de Cuenta de Usuario: Registro de usuarios, inicio de sesión seguro y opciones de perfil personalizado.

5. Promociones y Descuentos: Ofertas especiales y descuentos exclusivos para usuarios registrados.

6. Seguimiento de Pedidos: Funcionalidad para rastrear el estado de los pedidos en tiempo real.

7. Sistema de Recomendaciones: Utilización de algoritmos de recomendación para sugerir productos relevantes según el historial de compras del usuario.

8. Soporte al Cliente: Chat en vivo o sistema de soporte para resolver dudas y problemas de los usuarios.

9. Gestión de Inventario: Actualización en tiempo real del inventario para reflejar la disponibilidad de productos y evitar la venta de productos agotados. Esto incluirá la capacidad de realizar ajustes automáticos de inventario al completar una venta, así como alertas para reponer productos cuando sea necesario.

Restricciones y límites del proyecto:

- El aplicativo estará disponible solo para usuarios dentro de una determinada región geográfica.

- Se enfocará en la venta de productos de consumo masivo y no incluirá categorías especializadas.

- La disponibilidad de productos estará sujeta a la existencia en el inventario del supermercado físico.

- No se incluirán funcionalidades avanzadas como realidad aumentada o integración con dispositivos inteligentes en esta fase inicial del proyecto.

- Solo los usuarios registrados podrán realizar compras, asegurando así la autenticación y validación de los clientes antes de completar una transacción.

1. Metodología de desarrollo: La metodología Crystal es una aproximación ágil para el desarrollo de software que se adapta a diferentes tamaños y complejidades de proyectos. Se caracteriza por su enfoque flexible y orientado a las personas. El método Crystal lo adoptan principalmente equipos de desarrollo y líderes de proyecto que buscan un enfoque ágil personalizado a su situación única.

Etapas del proyecto:

-Iniciación: Definición del alcance, objetivos y requisitos del proyecto.

-Elaboración: Diseño detallado del sistema y planificación de las iteraciones.

-Construcción: Implementación iterativa del software, enfocándose en entregas parciales y continuas.

-Transición: Pruebas, integración y despliegue del software final.

Roles y responsabilidades:

Líder del proyecto: Responsable de la coordinación general y la comunicación con el cliente.

Equipo de desarrollo: Encargado de diseñar, implementar y probar el software.

Cliente o representante del usuario: Proporciona retroalimentación continua sobre el progreso del proyecto y los requisitos.

Plazos de entrega:

Iteraciones cortas: Entregas regulares de funcionalidades completas, generalmente cada 2-4 semanas.

Revisiones frecuentes: Se realizan revisiones periódicas con el cliente para validar el progreso y ajustar las prioridades.

Con la metodología CRYSTAL, se fomenta la colaboración entre los miembros del equipo, la adaptación a los cambios y la entrega temprana de valor al cliente.

1. Tecnologías utilizadas: Lenguajes de Programación:

HTML, CSS, PHP, Python para la lógica de negocio y JavaScript para la interfaz de usuarios.

Bases de Datos:

Puede ser MySQL, PostgreSQL dependiendo de los requisitos de almacenamiento y consulta de datos.

Herramientas de Desarrollo:

Visual Studio Code,

Control de versiones con Git y GitHub o GITLAB

1. Arquitectura del software: La arquitectura del software estará diseñada siguiendo un enfoque de cliente-servidor, con los siguientes componentes principales:

1. Cliente Web: La interfaz de usuario que los clientes utilizarán para acceder al aplicativo desde sus navegadores web. Esta parte del sistema se encargará de presentar la información de manera clara y amigable, permitiendo a los usuarios realizar compras, gestionar su cuenta y acceder a otras funcionalidades.

2. Servidor de Aplicación: El núcleo del sistema donde se procesarán las solicitudes de los clientes, gestionará la lógica de negocio y se conectará con la base de datos. Estará encargado de manejar la autenticación de usuarios, la gestión de pedidos, la generación de recomendaciones y otras funcionalidades críticas.

3. Base de Datos: Almacenará toda la información relevante para el funcionamiento del aplicativo, incluyendo datos de usuarios, catálogo de productos, historial de compras, información de inventario, entre otros. Se utilizará un sistema de gestión de bases de datos relacional MySQL o PostgreSQL, para garantizar la integridad y la consistencia de los datos.

En cuanto a seguridad, se implementarán medidas de protección como:

- Autenticación de usuarios mediante contraseñas seguras y tokens de sesión.

- Encriptación de datos sensibles en la base de datos y durante la transmisión entre el cliente y el servidor.

- Validación de entradas para prevenir ataques de inyección de código SQL o XSS.

Para garantizar la escalabilidad, se diseñará el sistema de manera modular y con capacidad de crecimiento horizontal. Se podrán añadir más servidores de aplicación y bases de datos según sea necesario para manejar un mayor número de usuarios y transacciones.

En cuanto a la integración con otros sistemas, se podría considerar la integración con sistemas de gestión empresarial (ERP) del supermercado para sincronizar datos de inventario, ventas y clientes. Además, se podrían explorar integraciones con sistemas de análisis de datos para mejorar las recomendaciones de productos y la personalización de la experiencia del usuario.

1. Plan de pruebas: El plan de pruebas se estructurará en varias etapas para garantizar la calidad y el correcto funcionamiento del software:

1. Pruebas de Unidad:

- Se realizarán pruebas individuales en cada componente del sistema, incluyendo funciones y módulos específicos.

- Se verificará que cada unidad funcione correctamente de acuerdo con sus especificaciones y requisitos.

- Se utilizarán herramientas de pruebas unitarias para automatizar este proceso y facilitar la detección temprana de errores.

2. Pruebas de Integración:

- Se probará la integración entre los diferentes componentes del sistema para asegurar que trabajen juntos de manera eficiente.

- Se verificará que la comunicación entre los distintos módulos sea correcta y que no haya conflictos en la interoperabilidad.

- Se evaluará la consistencia de los datos transmitidos entre los componentes.

3. Pruebas de Sistema:

- Se realizarán pruebas exhaustivas del sistema en su conjunto para validar su funcionamiento global.

- Se simularán escenarios de uso real para comprobar la estabilidad, el rendimiento y la escalabilidad del sistema.

- Se verificará que todas las funcionalidades principales estén operativas y que no existan errores críticos que puedan afectar la experiencia del usuario.

4. Pruebas de Aceptación:

- Se llevarán a cabo pruebas finales con usuarios reales o simulados para validar que el software cumple con los requisitos y expectativas del cliente.

- Se evaluará la usabilidad, la accesibilidad y la satisfacción del usuario durante la interacción con el aplicativo.

- Se recopilarán comentarios y sugerencias de los usuarios para realizar ajustes finales antes del lanzamiento oficial.

1. Plan de implementación: Preparación y Capacitación del Personal:

Se organizarán sesiones de capacitación presenciales para el personal de caja, estanterías, administración y cualquier otro departamento relevante.

Se proporcionarán manuales de usuario detallados y se ofrecerá soporte continuo durante el proceso de aprendizaje.

Migración de Datos:

Se evaluarán los sistemas actuales de gestión de inventario, ventas y finanzas para determinar los datos que necesitan ser migrados al nuevo software.

Implementación Gradual:

Se comenzará con una implementación piloto en una sucursal o sección específica del supermercado para probar la funcionalidad del software en un entorno real.

Post-Implementación:

Se establecerá un equipo de soporte técnico dedicado para abordar cualquier problema técnico o pregunta de los usuarios después de la implementación.

Se realizarán reuniones regulares con el personal para recopilar comentarios y realizar ajustes según sea necesario.

Se proporcionará formación continua para asegurar que el personal esté familiarizado y cómodo con el nuevo software a largo plazo.

1. Ficha Técnica de software y Hardware mínimos y recomendados
2. Presupuesto:

1. Honorarios del equipo de desarrollo:

- 3 personas x 80.000 COP/hora x 8 horas = 1.920.000 COP.

2.Costos de licencia de Software:

- Son libres, no habría costos asociados.

. Costos de dominio:

- 90.000 COP. Con el dominio de google.

3. Costos de hosting:

- 8.000 COP. Con GoDaddy

4. Costos de pruebas de calidad:

- Por día de prueba 5.000 COP.

Sumando todos estos elementos, el presupuesto total sería de 2.023.000 COP.

1. Cronograma de trabajo:

1.Análisis y Requisitos (2 semanas): Identificación de necesidades del supermercado y recopilación de requisitos del software.

2.Diseño (3 semanas): Creación del diseño arquitectónico, interfaz de usuario y estructura de la base de datos.

3.Desarrollo (6 semanas): Implementación del software según los requisitos y diseños establecidos.

4.Pruebas y Ajustes (4 semanas): Verificación del funcionamiento del software y corrección de errores.

5.Capacitación y Preparación (2 semanas): Formación del personal y preparación para la transición al nuevo sistema.

6.Implementación y Migración de Datos (2 semanas): Puesta en marcha del software y migración de datos desde sistemas antiguos.

7.Soporte

Post-Implementación (Ongoing): Ofrecimiento de asistencia continua y mantenimiento del software después de la implementación

1. **Capacitación:**

**Descripción de la capacitación:**

La capacitación para el uso del aplicativo web de un supermercado PowerFull Market estará diseñada para garantizar que todo el personal esté familiarizado y competente en su uso. Se proporcionará formación sobre la navegación en la plataforma, la gestión de inventario, la realización de transacciones, la atención al cliente en línea y cualquier otra función relevante del aplicativo. La capacitación puede ser presencial o virtual, dependiendo de las necesidades y preferencias del cliente.

**Objetivos de la capacitación:**

1. Familiarizar al personal del supermercado con todas las funciones y características del aplicativo web.

2. Mejorar la eficiencia en la gestión de inventario, ventas y atención al cliente a través del uso del aplicativo.

3. Garantizar la seguridad y confidencialidad de la información del cliente y de la empresa al utilizar la aplicación.

4. Fomentar la adopción y el uso efectivo del aplicativo en las operaciones diarias del supermercado, promoviendo una transición fluida hacia el entorno digital.Conclusiones: Resumen de los puntos clave de la propuesta técnica y una invitación al cliente para discutir cualquier duda o inquietud que pueda tener.

Al presentar una propuesta técnica de un proyecto de software a un cliente privado, es importante adaptar la estructura y el contenido de la propuesta a las necesidades y expectativas del cliente, asegurándose de presentar la información de manera clara y concisa para facilitar su comprensión.