

CARPETA DE CAMPO - Software de una Cafeteria

Integrantes: Ciro Lobotrico - Joseph Guerrero - Alan Gonzalez

Curso: 6to 2da

Turno: Vespertino

Análisis del Problema: Sistema de Información para una Cafetería

Objetivo de Informatización

Creación de una página para una cafetería: El objetivo principal es proporcionar una plataforma digital que permita a los usuarios realizar pedidos, realizar un seguimiento de las ventas, gestionar roles de usuarios, y facilitar la administración de la cafetería.

Motivación

Eficiencia en la gestión: Busca mejorar la eficiencia en la toma de pedidos, seguimiento de ventas y gestión de usuarios.

Mejora de la experiencia del usuario: Proporciona a los clientes una forma conveniente de realizar pedidos y seguir su estado.

Funcionalidades Principales:

Registro de Usuarios: Permite a los clientes crear cuentas y tener perfiles personalizados.

Autenticación y Roles: Implementa un sistema de autenticación para garantizar la seguridad y asigna roles como administrador, empleado y usuario para controlar el acceso a diferentes funciones.

Realización de Pedidos: Facilita la realización de pedidos en línea para los usuarios registrados.

Seguimiento de Ventas: Permite a los administradores y empleados visualizar las ventas y generar informes.

Gestión de Roles: Los administradores pueden cambiar los roles de los usuarios, afectando así su acceso a diferentes secciones de la aplicación.

Problemas a Considerar

Seguridad: Asegurar que la información del usuario y las transacciones estén protegidas.

Usabilidad: Garantizar que la interfaz sea intuitiva y fácil de usar tanto para clientes como para administradores y empleados.

Rendimiento: Asegurarse de que la aplicación sea lo suficientemente rápida y eficiente incluso durante períodos de alta demanda.

Integración con Métodos de Pago: Si se implementan transacciones en línea, asegurar una integración segura con métodos de pago.

Escalabilidad: Considerar la capacidad del sistema para manejar un crecimiento futuro en el número de usuarios y transacciones.

Metodología de Desarrollo:

Desarrollo Iterativo: Puede ser beneficioso adoptar un enfoque iterativo para ir mejorando y ajustando la aplicación según las necesidades y el feedback.

Desarrollo Ágil: Dado que las necesidades del negocio pueden cambiar, un enfoque ágil permitiría adaptarse de manera más flexible a esos cambios.

Pruebas Rigurosas: Asegurar pruebas exhaustivas para garantizar la funcionalidad y la seguridad del sistema.

Visualización de Problemas:

Diagrama de Flujo de Usuario: Representar visualmente el proceso desde la realización del pedido hasta el seguimiento de la venta.

Mapa de Roles y Permisos: Ilustrar claramente los roles de administrador, empleado y usuario, y los permisos asociados a cada uno.

Wireframes de la Interfaz: Crear representaciones visuales de las páginas clave para comprender la navegación y la disposición de la información.

Fecha	Cantidad de horas trabajadas	Descripción de las actividades, responsable, aspectos positivos desarrolladas y negativos
26/10	8 horas	Diseño de Interfaz de Usuario
27/10	20 horas	Desarrollo de la Página Principal en PHP
01/11	12 horas	Integración de Base de Datos
01/11	10 horas	Prototipado Rápido
09/11	15 horas	Revisión de Código y Optimización

Aspectos Positivos y Negativos:

Positivos: La integración de la base de datos se completó antes de lo previsto, permitiendo avanzar rápidamente en el desarrollo.

Los prototipos fueron bien recibidos en la fase de revisión, generando comentarios constructivos.

Negativos: Se experimentaron algunos retrasos en el desarrollo de la interfaz debido a la complejidad de las interacciones esperadas.

La revisión de código reveló algunas áreas que requerían una optimización más profunda, lo que generó un tiempo adicional de trabajo.

3. Planificación: Proceso de Elaboración

3.1 Búsqueda y Selección de Información:

Entrevistas: Realización de entrevistas con partes interesadas, como usuarios finales y profesionales médicos, para comprender sus necesidades y expectativas con respecto a la página de citatorios.

Cuestionarios para Entrevistas: Diseño de cuestionarios estructurados para guiar las entrevistas y recopilar información específica sobre los requisitos y preferencias de los usuarios.

Charlas: Sesiones de charlas informales con el equipo de desarrollo y otros stakeholders para intercambiar ideas, identificar desafíos potenciales y alinear objetivos.

Capturas de Pantalla:

Recopilación de capturas de pantalla de sistemas similares o herramientas existentes para tener una referencia visual y extraer mejores prácticas de diseño y funcionalidad.

3.2 Bibliografía Consultada:

Links: Investigación en línea para acceder a recursos relevantes, como artículos académicos, blogs especializados y sitios web confiables que aborden temas relacionados con el desarrollo de aplicaciones web y gestión de citas médicas.

Videos: Visualización de videos educativos y tutoriales en plataformas como YouTube para adquirir conocimientos prácticos sobre las tecnologías y metodologías que se utilizarán en el desarrollo.

Tutoriales: Estudio de tutoriales detallados sobre el uso de herramientas específicas, como Figma para el diseño, PHP para el desarrollo, y SQL para la gestión de bases de datos.

3.3 División de Tareas:

Roles y Responsabilidades: Identificación clara de roles dentro del equipo de desarrollo, asignando responsabilidades específicas a cada miembro según sus habilidades y conocimientos.

Colaboración Interdisciplinaria: Fomento de la colaboración interdisciplinaria, asegurando que los diseñadores, desarrolladores y otros miembros del equipo trabajen de manera conjunta para integrar eficientemente sus habilidades.

3.4 Cronograma de la División de Tareas:

Establecimiento de Fechas Límite: Definición de fechas límite realistas para cada fase del proyecto, considerando los plazos, la complejidad de las tareas y las posibles interdependencias entre ellas.

Fechas de Reuniones y Revisiones: Programación de reuniones regulares para revisar el progreso, abordar problemas emergentes y ajustar el cronograma según sea necesario.

Duración del Proyecto: Estimación de la duración total del proyecto, desde la planificación inicial hasta el lanzamiento final, teniendo en cuenta posibles imprevistos y la necesidad de iteraciones.

4. RESOLUCIÓN DE PROBLEMAS: ¿POR QUÉ? ¿CÓMO? (IDEAS PLANTEADAS)

Problema: Seguridad de la Información

¿Por qué es un problema? La seguridad es crucial para proteger la información confidencial del usuario y las transacciones.

¿Cómo resolverlo? Implementar protocolos de seguridad robustos, como encriptación de datos, autenticación de dos factores y certificados SSL para transacciones seguras.

Problema: Usabilidad de la Interfaz de Usuario

¿Por qué es un problema? Una interfaz confusa o complicada puede afectar negativamente la experiencia del usuario.

¿Cómo resolverlo? Realizar pruebas de usabilidad, recopilar feedback de usuarios y ajustar la interfaz para que sea intuitiva. Utilizar principios de diseño centrados en el usuario.

Problema: Rendimiento del Sistema

¿Por qué es un problema? Un rendimiento deficiente puede llevar a una experiencia del usuario lenta y frustrante.

¿Cómo resolverlo? Optimizar el código, utilizar técnicas de almacenamiento en caché, y escalar los recursos de servidor según sea necesario para garantizar un rendimiento fluido.

Problema: Integración con Métodos de Pago

¿Por qué es un problema? La integración incorrecta puede afectar la confianza del usuario y generar problemas en las transacciones.

¿Cómo resolverlo? Colaborar con proveedores de servicios de pago confiables, realizar pruebas exhaustivas de transacciones y garantizar la conformidad con estándares de seguridad.

Problema: Escalabilidad del Sistema

¿Por qué es un problema? Un sistema que no puede manejar el crecimiento puede resultar en caídas del servicio y pérdida de datos.

¿Cómo resolverlo? Diseñar la arquitectura del sistema considerando la escalabilidad, implementar soluciones en la nube y monitorear el rendimiento para realizar ajustes según sea necesario.

Problema: Cambios en los Requisitos del Negocio

¿Por qué es un problema? Los cambios no anticipados en las necesidades del negocio pueden afectar la eficiencia y funcionalidad del sistema.

¿Cómo resolverlo? Adoptar metodologías de desarrollo ágil para permitir adaptaciones rápidas, mantener una comunicación constante con los stakeholders y realizar revisiones periódicas de requisitos.

Problema: Eficiencia en la Gestión y Toma de Pedidos

¿Por qué es un problema? La falta de eficiencia en la gestión puede afectar la velocidad y precisión en la toma de pedidos.

¿Cómo resolverlo? Implementar un sistema de gestión de pedidos intuitivo, integrar tecnologías como códigos de barras o QR para agilizar procesos y proporcionar capacitación adecuada al personal.

Problema: Experiencia del Usuario en la Realización de Pedidos

¿Por qué es un problema? Si los usuarios encuentran dificultades al realizar pedidos, puede afectar la adopción de la plataforma.

¿Cómo resolverlo? Simplificar el proceso de pedido, ofrecer opciones de personalización claras, y proporcionar asistencia en línea o tutoriales para los usuarios.

5. Análisis de Viabilidad/ Discusión:

5.1 Viabilidad Técnica:

Desarrollo en PHP: La elección de PHP para el desarrollo de la página de citatorios es técnicamente viable, dada su amplia compatibilidad y funcionalidades para aplicaciones web.

Figma y Herramientas de Diseño: La integración de Figma en el proceso de diseño es técnica y creativamente sólida, facilitando una colaboración efectiva entre diseñadores y desarrolladores.

Base de Datos con SQL: El uso de SQL para la base de datos (médico 1.sql) es estándar y viable, pero se debe considerar la escalabilidad y el rendimiento a medida que la plataforma crezca.

5.2 Viabilidad Financiera:

Costos de Desarrollo: El costo inicial del desarrollo dependerá del tamaño del equipo, la complejidad de las funcionalidades y las herramientas utilizadas. Se debe considerar también el mantenimiento a largo plazo.

Licencias y Herramientas: Si se utilizan herramientas de diseño o bases de datos que requieren licencias, es necesario evaluar los costos asociados.

5.3 Viabilidad Operativa:

Capacitación del Equipo: La formación del equipo en las herramientas y tecnologías seleccionadas es esencial para una operación eficiente.

Soporte Técnico: Establecer un sistema de soporte técnico para abordar problemas operativos y garantizar la continuidad del servicio.

5.4 Discusión de Desafíos:

Escalabilidad: A medida que la plataforma crezca, se deben implementar medidas para garantizar la escalabilidad, especialmente en términos de rendimiento de la base de datos y del código PHP.

Aceptación del Usuario: Garantizar que la interfaz sea intuitiva y cumpla con las expectativas de los usuarios es crucial para la adopción exitosa.

Mantenimiento Continuo: Se debe planificar el mantenimiento continuo para abordar posibles problemas de seguridad, actualizaciones de software y mejoras funcionales.

6. Desarrollo del Prototipo:

6.1 Definición de Requisitos del Prototipo:

Revisión de la información recopilada: Utiliza la información recopilada durante las entrevistas, cuestionarios y charlas para definir claramente los requisitos del prototipo.

Identificación de Funcionalidades Clave: Destaca las funcionalidades clave que deben estar presentes en el prototipo para asegurar una representación fiel de la página de citatorios.

6.2 Herramientas y Tecnologías:

Figma: Utiliza Figma para traducir los requisitos en un diseño visual. Crea wireframes interactivos y define la estructura de la página, la navegación y la disposición de los elementos.

HTML/CSS: Comienza a traducir los elementos del diseño de Figma a código HTML y CSS para construir una representación estática y visualmente precisa del prototipo.

6.3 Iteración y Feedback:

Sesiones de Revisión con el Equipo: Realiza sesiones de revisión periódicas con el equipo para obtener feedback sobre el diseño y la usabilidad.

Ajustes Interactivos: Realiza ajustes en el diseño y la funcionalidad del prototipo según las sugerencias y comentarios recibidos.

6.4 Desarrollo de Funcionalidades Clave:

Formularios de Citatorios: Implementa formularios interactivos para la emisión y gestión de citatorios, asegurando que la información clave sea capturada de manera efectiva.

Navegación Intuitiva: Asegúrate de que la navegación entre las diferentes secciones de la página sea lógica y fácil de entender.

6.5 Integración con Figma:

Actualización Continua: Mantén una conexión activa entre el diseño en Figma y el desarrollo del prototipo para asegurar coherencia y precisión.

Feedback de Diseñadores: Obtén feedback regular de los diseñadores para asegurar que la implementación del prototipo refleje fielmente el diseño original.

6.6 Pruebas del Prototipo:

Pruebas de Usabilidad: Realiza pruebas de usabilidad con usuarios representativos para identificar posibles problemas de navegación o diseño.

Correcciones según Pruebas: Realiza correcciones en el prototipo según los resultados de las pruebas de usabilidad.

6.7 Documentación:

Registro de Proceso: Documenta el proceso de desarrollo del prototipo, incluyendo decisiones de diseño, diagramas, cambios realizados y problemas resueltos.

6.8 Presentación y Aprobación:

Sesión de Presentación: Presenta el prototipo al equipo de desarrollo y a partes interesadas para obtener aprobación y feedback adicional.

Iteraciones Finales: Realiza las iteraciones finales según los comentarios y aprueba el prototipo para pasar a la siguiente fase de desarrollo.

7 TESTEO O EVALUACIÓN DE COMPORTAMIENTO}

Pruebas de Seguridad:

Objetivo: Evaluar la robustez de las medidas de seguridad implementadas.

Cómo: Realizar pruebas de penetración para identificar vulnerabilidades, comprobar la resistencia a ataques de seguridad y asegurar el cumplimiento de estándares de seguridad.

Pruebas de Usabilidad:

Objetivo: Evaluar la facilidad de uso de la interfaz de usuario.

Cómo: Realizar pruebas de usabilidad con usuarios reales, recopilar comentarios sobre la navegación, el diseño y la eficiencia de la interfaz, y realizar ajustes según sea necesario.

Pruebas de Rendimiento:

Objetivo: Evaluar la velocidad y eficiencia del sistema bajo diferentes cargas de trabajo.

Cómo: Utilizar herramientas de prueba de carga para simular condiciones de uso intensivo, monitorear el tiempo de respuesta del sistema y optimizar el rendimiento.

Pruebas de Integración con Métodos de Pago:

Objetivo: Evaluar la eficacia de la integración con métodos de pago.

Cómo: Realizar transacciones de prueba con diferentes métodos de pago, verificar la precisión de las transacciones y garantizar la seguridad de la información financiera.

Pruebas de Escalabilidad:

Objetivo: Evaluar la capacidad del sistema para manejar un crecimiento de usuarios y transacciones.

Cómo: Incrementar gradualmente la carga de trabajo, monitorear el rendimiento y verificar que el sistema pueda escalar de manera eficiente.

Pruebas de Cambios en los Requisitos del Negocio:

Objetivo: Evaluar la capacidad del sistema para adaptarse a cambios en los requisitos del negocio.

Cómo: Introducir cambios simulados en los requisitos, evaluar la flexibilidad del sistema y la facilidad de implementación de modificaciones.

Pruebas de Eficiencia en la Gestión y Toma de Pedidos:

Objetivo: Evaluar la eficiencia de las funciones relacionadas con la gestión y la toma de pedidos.

Cómo: Simular escenarios de gestión de pedidos, medir el tiempo de procesamiento y la precisión, y ajustar los procesos según sea necesario.

Pruebas de Experiencia del Usuario en la Realización de Pedidos:

Objetivo: Evaluar la experiencia del usuario al realizar pedidos.

Cómo: Observar a usuarios mientras realizan pedidos, recopilar comentarios sobre la claridad de las opciones, la personalización y la asistencia disponible.