Planificación del Proyecto de Desarrollo de Software

1. Análisis de Requerimientos

Objetivo: Comprender a fondo lo que el cliente necesita para garantizar que la aplicación cumpla con todas sus expectativas y funcionalidades deseadas.

- **Reunión con el cliente**: Organizar una reunión inicial donde el cliente pueda expresar sus necesidades y preocupaciones sobre la aplicación. Es importante hacer preguntas abiertas para captar todos los detalles relevantes. Asegurarse de que el cliente esté involucrado en el proceso, pues esto facilitará la obtención de información clara sobre las funcionalidades, como la consulta del estado de las pistas, el pago con tarjeta, y los requisitos específicos de reserva (con o sin luz).

- **Requerimientos funcionales**:
  - El sistema debe permitir a los usuarios consultar la disponibilidad de las pistas en tiempo real.
  - Implementar una opción para que los usuarios puedan seleccionar si desean reservar la pista con o sin luz, ya que esto puede influir en el costo de la reserva.
  - Integrar un sistema seguro para realizar pagos en línea con tarjeta, lo cual es fundamental para la experiencia del usuario.
  - Generar informes o listados de reservas que se pueden filtrar por horas y por clientes, lo cual permitirá al personal del gimnasio gestionar las reservas de manera más eficiente.

- **Requerimientos no funcionales**:
  - La aplicación debe ser accesible desde diferentes dispositivos, incluyendo móviles y tablets, asegurando que la experiencia sea óptima en todas las plataformas.
  - Es importante que la aplicación esté construida bajo principios de diseño accesible, para que personas con diferentes capacidades puedan utilizarla.
  - Cumplir con la normativa sobre protección de datos, asegurando la privacidad de los usuarios, es esencial para construir confianza con el cliente.

- **Documentación**: Crear un documento de especificaciones que contenga todos los detalles sobre los requerimientos. Este documento servirá como referencia durante el desarrollo y ayudará a prevenir malentendidos o cambios de última hora.

2. Diseño

Objetivo: Definir cómo se va a construir la aplicación, seleccionando las tecnologías, lenguajes y herramientas necesarias para su desarrollo.

- **Lenguaje de programación**:
  - Utilizar **Python** con el framework **Django**, que es conocido por su simplicidad y robustez, ideal para el desarrollo web. También se puede considerar **PHP** con **Laravel**, que es una opción popular en el desarrollo de aplicaciones web.

- **Base de datos**:
    - Usar **PostgreSQL** como sistema de gestión de bases de datos. Este sistema es conocido por su estabilidad y por ser una solución robusta para manejar grandes volúmenes de datos, lo que es importante dado que la aplicación manejará múltiples reservas y usuarios.

- **Frontend**:
    - La interfaz del usuario se desarrollará utilizando **HTML5**, **CSS3** y **JavaScript**. Considerar el uso de frameworks como **React** o **Vue.js** puede hacer que el desarrollo sea más eficiente, ya que permiten construir interfaces más dinámicas y reactivas.

- **Entorno de desarrollo**:
    - Usar un entorno de desarrollo libre como **Visual Studio Code** o **Atom**, que ofrece extensiones y herramientas útiles para facilitar la programación. Es fundamental que el equipo esté familiarizado con el entorno elegido para maximizar la productividad.
    - Utilizar **Git** como sistema de control de versiones. Esto permitirá al equipo colaborar de manera eficiente, llevando un registro de los cambios y facilitando el trabajo en equipo.

- **Arquitectura**: La aplicación se desarrollará utilizando una arquitectura **cliente-servidor**. Esto significa que el cliente (el navegador del usuario) enviará solicitudes al servidor, que será el encargado de gestionar las reservas, pagos y toda la lógica del negocio. Este enfoque separa claramente la lógica de presentación de la lógica de negocio.

3. Desarrollo

Objetivo: Implementar la aplicación, desarrollando todas las funcionalidades necesarias de acuerdo con el diseño previamente establecido.

- **Backend**: Programar la lógica del servidor, que incluirá la gestión de reservas, autenticación de usuarios, y procesamiento de pagos. Es esencial que se implementen medidas de seguridad adecuadas para proteger la información sensible de los usuarios.
- **Frontend**: Diseñar y desarrollar la interfaz visual, asegurando que sea intuitiva y fácil de navegar. La experiencia del usuario es crucial, por lo que se deben realizar pruebas de usabilidad para identificar áreas de mejora.
- **Integración de pagos**: Incorporar una pasarela de pago confiable que permita a los usuarios pagar con tarjeta. Se debe asegurar que el sistema de pago cumpla con los estándares de seguridad y protección de datos, utilizando protocolos como HTTPS.
- **Control de luz**: Implementar la funcionalidad que permita a los usuarios seleccionar si desean reservar la pista con o sin luz. Esta opción debe estar claramente visible durante el proceso de reserva, y el sistema debe calcular el precio automáticamente según la selección realizada.

4. Pruebas

Objetivo: Asegurarnos de que la aplicación funcione correctamente y cumpla con todos los requerimientos antes de su lanzamiento al público.

- **Pruebas unitarias**: Desarrollar pruebas para cada componente del sistema. Esto ayuda a detectar errores en etapas tempranas y asegura que cada parte funcione de manera independiente.
- **Pruebas de integración**: Comprobar que los diferentes módulos de la aplicación interactúen correctamente. Esto es especialmente importante en un sistema donde diferentes componentes (como la gestión de reservas y la pasarela de pagos) deben comunicarse entre sí.
- **Pruebas funcionales**: Realizar pruebas exhaustivas para asegurarse de que la aplicación cumpla con todos los requerimientos establecidos. Esto incluye verificar que los usuarios puedan reservar pistas, realizar pagos y acceder a informes de manera correcta.
- **Pruebas de usabilidad**: Evaluar la interfaz de usuario con personas que no estén familiarizadas con la aplicación. Esto ayudará a identificar problemas de navegación y a hacer ajustes para mejorar la experiencia del usuario.
- **Pruebas de rendimiento**: Simular un alto volumen de usuarios simultáneos para ver cómo responde la aplicación. Esto es crucial para asegurarse de que la app pueda manejar la carga durante momentos de alta demanda.

5. Despliegue

Objetivo: Hacer que la aplicación esté disponible para el público y asegurarnos de que funcione sin problemas en un entorno real.

- **Servidor**: Elegir un servidor que soporte software libre, como **Apache** o **Nginx**. Esto asegurará que la aplicación esté alojada en un entorno estable y seguro.
- **Dominio**: Configurar un nombre de dominio adecuado para que los usuarios puedan acceder a la aplicación fácilmente. También es importante asegurarse de que el sitio esté optimizado para buscadores (SEO) para atraer más usuarios.
- **Base de datos**: Configurar la base de datos en el servidor y asegurarse de que esté bien estructurada para soportar el volumen de datos que se generará. Implementar copias de seguridad automáticas para evitar pérdidas de datos.

6. Mantenimiento

Objetivo: Asegurarse de que la aplicación siga funcionando correctamente y realizar mejoras según las necesidades del cliente y los usuarios.

- **Corrección de errores**: Establecer un sistema para que los usuarios puedan reportar errores y problemas. Esto incluye la creación de un formulario de contacto dentro de la aplicación para facilitar la comunicación.
- **Actualizaciones**: Programar actualizaciones regulares para añadir nuevas funciones o mejorar las existentes. Mantener la aplicación al día es crucial para retener a los usuarios y responder a sus necesidades cambiantes.
- **Soporte técnico**: Proporcionar asistencia técnica al cliente y a los usuarios para resolver cualquier problema que pueda surgir. Esto puede incluir la creación de una sección de preguntas frecuentes (FAQ) o tutoriales para ayudar a los usuarios a entender mejor la aplicación.

Conclusión

Seguir todas estas etapas en el ciclo de vida del software es fundamental para garantizar que la aplicación cumpla con las expectativas del cliente y funcione sin problemas. La elección de tecnologías libres, así como un enfoque en la calidad y la experiencia del usuario, son clave para el éxito de este proyecto y para satisfacer las necesidades de los usuarios finales.

Gabriel Trujillo Vallejo

Entornos de Desarrollo DAM Ud1