

Índice

1. Definición de Proyecto	2
1.1. Introducción	2
1.2. Planteamiento del problema	2
1.3. Objetivo General	2
1.4. Objetivos Específicos	2
1.5. Justificación	2
1.6. Alcances y Limitaciones	2
1.6.1. Alcances del Proyecto	2
1.6.2. Limitaciones del Proyecto	2
2. Marco teorico	3
2.1. Descripción	3
2.1.1. Metodología de Analisis	3
2.2. Entrevista de requerimientos	4
2.3. Analisis	4
2.3.1. Entrevista	4
2.3.2. Registro de miembros	4
2.3.3. Seguimiento Administrativo	4
2.3.4. Accesos al sistema	5
2.3.5. Integración	5
2.3.6. Soporte y mantenimiento	5
2.3.7. Últimas palabras	5
2.4. Historias de Usuario	6
2.5. Diseño	6
2.6. Desarrollo	6
2.7. Prioridad de historias de usuario	6

1. Definición de Proyecto

1.1. Introducción

1.2. Planteamiento del problema

1.3. Objetivo General

1.4. Objetivos Específicos

1.5. Justificación

1.6. Alcances y Limitaciones

1.6.1. Alcances del Proyecto

1. **Distribución Geográfica de Datos:** Implementación de una base de datos distribuida que permita almacenar y gestionar datos en múltiples ubicaciones físicas —en nuestro caso, las sedes de gimnasio gestionadas por el cliente— para mejorar la disponibilidad y la redundancia.
2. **Sincronización y Consistencia de Datos:** Desarrollo de mecanismos que garanticen que los datos sean consistentes entre las distintas ubicaciones y que cualquier actualización se propague adecuadamente.
3. **Escalabilidad:** Diseño de la base de datos para que pueda escalar horizontalmente, añadiendo más nodos según sea necesario para manejar el aumento del volumen de datos y la carga de trabajo.
4. **Seguridad de Datos:** Implementación de medidas de seguridad que protejan los datos distribuidos contra accesos no autorizados y posibles ataques cibernéticos.
5. **Resiliencia y Recuperación ante Desastres:** Establecimiento de procedimientos y tecnologías para asegurar la recuperación rápida y efectiva en caso de fallos o desastres.

1.6.2. Limitaciones del Proyecto

1. **Latencia de Red:** La latencia de la red entre los nodos puede afectar el rendimiento del sistema, especialmente si los nodos están ubicados en regiones geográficas distantes.
2. **Complejidad de Gestión:** La administración y el mantenimiento de una base de datos distribuida pueden ser significativamente más complejos en comparación con una base de datos centralizada.
3. **Costos de Infraestructura:** Los costos asociados con el hardware, el software y la red para soportar una base de datos distribuida pueden ser elevados.
4. **Consistencia Eventual:** Algunas arquitecturas distribuidas optan por la consistencia eventual, lo que puede llevar a periodos en los que los datos no están sincronizados en todos los nodos.
5. **Problemas de Sincronización:** La sincronización de datos entre nodos puede enfrentar problemas, especialmente en redes con alta latencia o en escenarios de particionamiento de red.

2. Marco teorico

2.1. Descripción

2.1.1. Metodología de Analisis

¿Qué es metodología Scrum? Es un pequeño equipo de personas, un equipo Scrum. El equipo Scrum consta de: - un Scrum Master, - un propietario de producto (Product Owner) y - desarrolladores.

Dentro de un equipo de Scrum, no hay sub-equipos ni jerarquías. Es una unidad cohesionada de profesionales enfocada en un objetivo a la vez, el objetivo del Producto.

Scrum (nombre que proviene de cierta jugada que tiene lugar durante un partido de rugby) es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990.

Retraso: lista de prioridades de los requerimientos o características del proyecto que dan al cliente un valor del negocio. Es posible agregar en cualquier momento otros aspectos al retraso (ésta es la forma en la que se introducen los cambios). El gerente del proyecto evalúa el retraso y actualiza las prioridades según se requiera. Sprints: consiste en unidades de trabajo que se necesitan para alcanzar un requerimiento definido en el retraso que debe ajustarse en una caja de tiempo predefinida (lo común son 30 días). Durante el sprint no se introducen cambios (por ejemplo, aspectos del trabajo retrasado). Así, el sprint permite a los miembros del equipo trabajar en un ambiente de corto plazo pero estable. Reuniones Scrum: son reuniones breves (de 15 minutos, por lo general) que el equipo Scrum efectúa a diario. Hay tres preguntas clave que se pide que respondan todos los miembros del equipo. • ¿Qué hiciste desde la última reunión del equipo? • ¿Qué obstáculos estás encontrando? • ¿Qué planeas hacer mientras llega la siguiente reunión del equipo? Un líder del equipo, llamado maestro Scrum, dirige la junta y evalúa las respuestas de cada persona.

¿Qué es una Metodología de análisis? Nosotros enfocamos un marco teorico que influye en el proceso estructurado para guiar el desarrollo desde los elementos de entrada(información recopilada), hasta las herramientas que se utilizaran para la elaboración de ellas. Asi facilitando su implementación en las diferentes etapas del proceso. Pressman promueve un enfoque iterativo e incremental en el desarrollo de software, en este caso lo utilizaremos en el desarrollo de nuestra base de datos distribuida en base al proyecto del GYM, esto para que se contruya y mejore partes del proyecto de manera gradual. Esto tambien tomando en cuenta los aspectos de gestión de proyectos, estimación, planificación y control, asi tambien la detección temprana de los defectos y su corrección inmediata.

3. Historias de usuario y Casos de uso

3.1. Casos de Uso

Analiza el comportamiento del sistema en distintas condiciones en las que el sistema responde a una petición de alguno de sus participantes. En esencia, un caso de uso narra una historia estilizada sobre cómo interactúa un usuario final (que tiene cierto número de roles posibles) con el sistema en circunstancias específicas. Un caso de uso ilustra el software o sistema desde el punto de vista del usuario final.

3.2. Historias de Usuario

Las historia de usuario consta de ser un texto narrativo, un lineamiento de tareas o interacciones, una descripción basada en un formato o una representación diagramática. Sin importar su forma

3.3. Entrevista de requerimientos

Éste es el enfoque más directo, los miembros del equipo de software se reúnen con los usuarios para entender mejor sus necesidades, motivaciones, cultura laboral y una multitud de aspectos adicionales. Esto se logra en reuniones individuales o a través de grupos de enfoque.

3.4. Analisis

3.4.1. Entrevista

Entrevistador: Buenas tardes. Gracias por tomarse el tiempo para esta entrevista. Para empezar, ¿puede darme una visión general de su negocio y qué es lo que espera lograr con este nuevo sistema?

Administrador: Buenas tardes. Claro, tengo varios gimnasios en diferentes ubicaciones, y actualmente gestionamos todo manualmente o con sistemas muy básicos. Lo que realmente necesito es un sistema que me permita gestionar las inscripciones de los miembros, el seguimiento de su actividad y productividad, y también la administración de los empleados y horarios.

Entrevistador: Entiendo. Vamos a desglosar esto en partes. Primero, ¿puede describir cómo maneja actualmente el registro de miembros y qué aspectos le gustaría mejorar?

Administrador: En este momento, usamos un sistema de hojas de cálculo para el registro de nuevos miembros y el seguimiento de pagos. Esto es bastante ineficiente y propenso a errores. Me gustaría que el nuevo sistema permitiera registrar automáticamente nuevos miembros y gestionar sus pagos.

3.4.2. Registro de miembros

Entrevistador: Perfecto. Para el registro de miembros, ¿qué información específica necesita capturar?

Administrador: Necesitamos capturar información básica como:

- nombre,
- número de teléfono principal y
- número de teléfono alternativo.

También necesitamos registrar:

- el tipo de membresía,
- la fecha de inicio y
- los pagos realizados.

3.4.3. Seguimiento Administrativo

Entrevistador: En relación a la administración de empleados y horarios, ¿qué funcionalidades específicas está buscando?

Administrador: Necesito un módulo para gestionar:

- los horarios de los empleados, incluyendo:
 - las asignaciones de turno y
 - las solicitudes de vacaciones.

También sería útil poder hacer un seguimiento de:

- sus horas trabajadas y
- sus pagos.

3.4.4. Accesos al sistema

Entrevistador: Entiendo. ¿Qué tipo de acceso y permisos deberían tener los diferentes usuarios del sistema? Por ejemplo, ¿qué pueden ver o hacer los administradores frente a los empleados o los entrenadores?

Administrador:

- Los administradores deberían tener acceso completo a todas las funciones del sistema, incluyendo: la gestión de miembros, empleados, informes y configuraciones.
- Los empleados y entrenadores solo deberían tener acceso a funciones relacionadas con su trabajo, como ver sus horarios y gestionar sus propias clases.

3.4.5. Integración

Entrevistador: Muy bien. En términos de integración, ¿hay otros sistemas o herramientas con los que el nuevo sistema debería integrarse, como sistemas de contabilidad o plataformas de marketing?

Administrador: Sí, el sistema debería integrarse con nuestro software de contabilidad para que los pagos se registren automáticamente.

3.4.6. Soporte y mantenimiento

Entrevistador: Perfecto. Para finalizar, ¿hay algún otro requerimiento o aspecto específico que no hayamos cubierto y que sea importante para usted?

Administrador: Creo que lo hemos cubierto bastante bien. Solo me gustaría:

- que el sistema fuera fácil de usar,
- que tuviera un buen soporte técnico y
- una interfaz llamativa.

La facilidad de uso es importante para que tanto los empleados como los miembros puedan adaptarse rápidamente.

3.4.7. Últimas palabras

Entrevistador: Entendido. Muchas gracias por la información detallada. Con estos requisitos, podemos empezar a definir el alcance del proyecto y trabajar en el diseño del sistema. ¿Hay algo más que quiera agregar antes de que terminemos?

Administrador: No, eso es todo. Gracias a usted.

Entrevistador: De nada. Le mantendremos informado sobre el avance del proyecto. Que tenga un buen día.

Administrador: Igualmente. Que tenga un buen día.

3.5. Historias de Usuario

- Diagramas de Casos de uso
- Historias de Usuario

3.6. Diseño

- Arquitectura de Sistema de Base de Datos Distribuido
- Esquema Conceptual Global
- Esquema Conceptual Local

3.7. Desarrollo

- Implementación del SBDD
- Implementación De Interfaces de Usuario
- Pruebas Unitarias

3.8. Prioridad de historias de usuario