

Documentación de Evaluación de Arquitectura ATAM (Arquitectura Tecnológica para la Evaluación de Modelos)

Sistema de Gestión de Tareas

Contexto:

Este documento presenta una evaluación de arquitectura utilizando el modelo ATAM para el Sistema de Gestión de Tareas. El objetivo principal de este sistema es permitir a los usuarios organizar, administrar y realizar un seguimiento eficiente de sus tareas y proyectos. El sistema proporcionará funcionalidades clave de creación, edición, visualización y eliminación de tareas.

1. Introducción

El Sistema de Gestión de Tareas se propone como una solución para mejorar la productividad y la organización personal y profesional de los usuarios. Este documento tiene como objetivo evaluar la arquitectura propuesta para el sistema y analizar su capacidad para cumplir con los requisitos funcionales y no funcionales.

2. Requisitos Funcionales

2.1. Creación de Tareas

- Como usuario, quiero poder crear una nueva tarea de manera fácil y rápida.
- Debo poder asignar un título descriptivo a la tarea.
- Debo poder asignar una prioridad a la tarea, como alta, media o baja.
- Debo poder cambiar el estado a la tarea, los cuales tendrán dos estados “En progreso” y “Completado”, por defecto la tarea se crea con el estado “En proceso”.
- Quiero tener la capacidad de agregar una descripción detallada de la tarea si es necesario.
- Una vez creada, la tarea debe mostrarse en mi lista de tareas pendientes.

2.2. Edición de Tareas

- Como usuario, quiero poder editar una tarea existente si surgen cambios o actualizaciones.
- Debo poder cambiar el título, la prioridad, la descripción de la tarea.
- Los cambios realizados en una tarea deben reflejarse de manera instantánea y precisa.

2.3. Visualización de Tareas

- Como usuario, quiero poder ver todas mis tareas pendientes de manera clara y organizada.
- Debo poder filtrar mis tareas por título, categoría, prioridad para una búsqueda más específica.
- Quiero poder marcar una tarea como completada una vez que la haya realizado.

- Debo poder ver tareas pasadas o completadas para hacer un seguimiento de mis logros y para futuras referencias.

2.4. Eliminación de Tareas

- Como usuario, quiero poder eliminar una tarea cuando ya no sea relevante o necesaria.
- Debo recibir una confirmación antes de que se elimine una tarea para evitar eliminaciones accidentales.
- Una vez eliminada, la tarea debe desaparecer de mi lista de tareas pendientes.

3. Requisitos No Funcionales

3.1. Usabilidad

- La interfaz de usuario debe ser intuitiva y de fácil navegación.
- La creación y edición de tareas deben ser procesos simples y eficientes.

3.2. Rendimiento

- El sistema debe responder de manera rápida y eficiente incluso cuando haya un gran número de tareas almacenadas (20 tareas).
- La visualización y filtrado de tareas debe ser ágil.

3.3. Seguridad

- Los datos de los usuarios y las tareas deben estar protegidos y almacenados de manera segura.
- Debe implementarse autenticación y autorización para proteger la información confidencial.

4. Evaluación de Arquitectura ATAM

4.1. Escenarios

Se han identificado varios escenarios clave para evaluar la arquitectura del Sistema de Gestión de Tareas:

Escenario 1: Creación de Tareas

- Un usuario desea crear una nueva tarea de manera rápida y asignarle una fecha de vencimiento y prioridad.
- El sistema debe mostrar la tarea creada en su lista de tareas pendientes.

Escenario 2: Edición de Tareas

- Un usuario necesita editar una tarea existente para cambiar su título y su descripción.
- Los cambios deben reflejarse de manera instantánea y precisa en la tarea.

Escenario 3: Visualización de Tareas

- Un usuario desea ver todas sus tareas pendientes y filtrarlas por categoría.
- Debe poder marcar una tarea como completada y ver tareas pasadas.

Escenario 4: Eliminación de Tareas

- Un usuario quiere eliminar una tarea que ya no es relevante.
- Debe recibir una confirmación y la tarea debe desaparecer de su lista.

4.2. Evaluación

La evaluación de arquitectura se realizará utilizando los escenarios mencionados anteriormente. Se verificará la eficacia de la arquitectura para cumplir con los requisitos funcionales y no funcionales. Además, se evaluará la escalabilidad, la seguridad y la capacidad de mantener y actualizar el sistema.

4.3. Resultados

Los resultados de la evaluación de arquitectura se documentarán en un informe separado después de realizar las pruebas y análisis correspondientes.

5. Conclusiones

El Sistema de Gestión de Tareas es una solución que busca mejorar la productividad y organización personal de los usuarios. La evaluación de arquitectura ATAM permitirá identificar posibles desafíos y asegurar que la arquitectura sea eficiente.

2. Evaluación de Trade-offs y Decisiones Clave

En el proceso de diseño y desarrollo del Sistema de Gestión de Tareas, se han enfrentado varios trade-offs y se han tomado decisiones clave para equilibrar los requisitos y limitaciones del proyecto. A continuación, se presenta una evaluación de algunos de los trade-offs y decisiones más significativas tomados durante el proceso de arquitectura.

Trade-off 1: Usabilidad vs. Funcionalidad Compleja

Descripción: La decisión clave aquí se centró en encontrar el equilibrio entre una interfaz de usuario extremadamente simple y una con una funcionalidad más compleja. Si bien una interfaz simple es fácil de usar, una funcionalidad compleja podría abordar las necesidades de usuarios avanzados.

Decisión Tomada: Se optó por una interfaz de usuario simple y fácil de navegar. Si bien esto puede limitar algunas funcionalidades avanzadas, se priorizó la facilidad de uso y la accesibilidad para una amplia gama de usuarios.

Trade-off 2: Escalabilidad vs. Rendimiento

Descripción: En el diseño de la base de datos, se enfrentó el trade-off entre la escalabilidad futura y el rendimiento inmediato. Una estructura de datos altamente normalizada puede mejorar la escalabilidad, pero podría afectar negativamente el rendimiento a corto plazo.

Decisión Tomada: Se eligió una estructura de datos que prioriza la escalabilidad a largo plazo. Aunque esto podría implicar ciertas limitaciones en cuanto al rendimiento inmediato, se considera que es una inversión estratégica para garantizar que el sistema pueda crecer de manera sostenible en el futuro.

Trade-off 3: Seguridad vs. Accesibilidad

Descripción: La implementación de medidas de seguridad adicionales, como autenticación de dos factores, podría aumentar la seguridad del sistema, pero también dificultar el acceso para algunos usuarios.

Decisión Tomada: Se implementó la autenticación estándar como medida de seguridad inicial. Se dejó abierta la posibilidad de agregar medidas adicionales en el futuro, pero se priorizó la accesibilidad y la facilidad de uso en esta etapa inicial.

Trade-off 4: Actualizaciones Frecuentes vs. Estabilidad

Descripción: Se debatió si implementar actualizaciones frecuentes con nuevas características o mantener una versión estable.

Decisión Tomada: Se optó por un enfoque de actualizaciones más frecuentes para responder a las necesidades cambiantes de los usuarios y mantener el sistema competitivo en el mercado.

Decisiones Clave:

Decisión 1: Tecnología Backend

Descripción: Se eligió utilizar el lenguaje de programación C# y un framework de desarrollo .NET para la implementación del backend.

Justificación: La elección se basó en la experiencia del equipo de desarrollo en estas tecnologías, lo que permitirá una implementación más rápida y eficiente.

Decisión 2: Almacenamiento de Datos

Descripción: Se decidió utilizar una base de datos relacional para almacenar las tareas y los datos de los usuarios, se eligió SQLServer.

Justificación: La elección se basó en la necesidad de mantener la integridad de los datos y permitir futuras consultas y análisis complejos de manera eficiente.

Decisión 3: Estrategia de copias de seguridad

Descripción: Se estableció una estrategia de copias de seguridad periódicas de la base de datos, con la capacidad de restaurar los datos en caso de pérdida o corrupción

Justificación: Esto asegura la disponibilidad y recuperación de datos importantes en caso de fallos, garantizando la integridad de las tareas y la información del usuario.

Decisión 4: Tecnología Front-End

Descripción: Se optó por utilizar un enfoque de desarrollo de front-end basado en una SPA (Single Page Application), utilizando el framework de JavaScript, React. Para el CSS se utilizó el framework TailwindCSS

Justificación: La elección de un enfoque SPA con un framework como React ofrece una experiencia de usuario más fluida y rápida, al minimizar los tiempos de carga de páginas completas. Esto mejora la velocidad de respuesta y la eficiencia del sistema al permitir la actualización dinámica de

contenido sin recargar la página completa. Además, React es ampliamente adoptado y cuenta con una comunidad activa y una gran cantidad de recursos disponibles para el desarrollo.

Conclusiones

La evaluación de trade-offs y decisiones clave es esencial en el proceso de diseño y desarrollo de un sistema. Cada decisión tiene implicaciones significativas en términos de usabilidad, rendimiento, seguridad y mantenimiento del sistema. Las decisiones tomadas hasta ahora se han centrado en proporcionar una experiencia de usuario eficiente y accesible, con la capacidad de escalar y evolucionar el sistema a medida que las necesidades cambian. El monitoreo constante y la retroalimentación de los usuarios serán cruciales para ajustar las decisiones y trade-offs a medida que el sistema se desarrolle y crezca.