



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Actividad:

Architecture specification

Estudiante:

Ángel Amaury Tienda Lezama

Grupo:

10B

Materia:

Desarrollo móvil Integral

Docente:

Ray Brunett Parra Galaviz

Tijuana, Baja California, 07 de Enero del 2025

Introducción

La especificación de la arquitectura de software es un paso crítico en el proceso de desarrollo, ya que define la estructura, los componentes y las interacciones de un sistema para satisfacer los requisitos funcionales y no funcionales. Una arquitectura bien especificada sirve como plano para el diseño y desarrollo, asegurando que el software sea escalable, mantenible y eficiente. Este proceso implica tomar decisiones técnicas y organizativas que impactan en el ciclo de vida del proyecto.

I. Recolección y Análisis de Requisitos

- **Definición de Requisitos Funcionales:** Identificación de las funcionalidades específicas que el sistema debe ofrecer.
- **Requisitos No Funcionales:** Aspectos como rendimiento, seguridad, escalabilidad, usabilidad y disponibilidad.
- **Priorización:** Determinar qué requisitos son críticos y deben influir directamente en las decisiones arquitectónicas.
- **Resultado:** Un conjunto claro de necesidades que la arquitectura debe cumplir.

II. Selección del Estilo Arquitectónico

La elección del estilo arquitectónico define cómo se estructurarán y organizarán los componentes del sistema.

Estilos comunes:

- **Arquitectura Monolítica:** Todo el sistema está integrado en un único conjunto.
- **Arquitectura en Capas:** Divide el sistema en niveles funcionales (presentación, lógica de negocio, acceso a datos).
- **Arquitectura Cliente-Servidor:** Separa las responsabilidades entre un cliente que solicita servicios y un servidor que los proporciona.
- **Arquitectura de Microservicios:** Divide la funcionalidad en servicios independientes y escalables.
- **Arquitectura Basada en Eventos:** Enfocada en la comunicación asíncrona entre componentes mediante eventos.

Criterios de selección: Escalabilidad, modularidad, tiempo de desarrollo, y requisitos del negocio

III. Definición de Componentes y Módulos

Los componentes son las unidades funcionales del sistema, mientras que los módulos agrupan lógicamente estas funcionalidades.

Actividades clave:

- Identificación de los principales componentes del sistema.
- Definición de las interacciones entre los módulos.
- Uso de diagramas UML, como Diagramas de Componentes y Diagramas de Clases, para documentar la estructura.

IV. Diseño de la Comunicación entre Componentes

Establecer cómo se comunicarán los componentes para garantizar un flujo de datos eficiente.

Opciones comunes::

- Llamadas a procedimientos remotos (RPC).
- Interfaz de programación de aplicaciones (APIs) RESTful o GraphQL.
- Sistemas de mensajería, como RabbitMQ o Kafka.
- Comunicación síncrona vs. asíncrona.

Impacto: Garantiza interoperabilidad, eficiencia y reducción de errores.

V. Selección de Tecnologías

Identificar las herramientas, lenguajes de programación y frameworks más adecuados.

Factores clave:

- Compatibilidad con los requisitos del sistema.
- Experiencia del equipo.
- Escalabilidad y soporte a largo plazo.

Ejemplo: Uso de Spring Boot para microservicios, React para la interfaz, y PostgreSQL para la base de datos.

VI. Consideraciones de Infraestructura

La arquitectura debe definir cómo se desplegará el sistema y en qué entorno operará.

Aspectos esenciales:

- Selección de entornos de ejecución (nube, local, híbrido).
- Configuración de redes y servidores.
- Planificación para alta disponibilidad y balanceo de carga.
- Uso de contenedores (Docker) y orquestadores (Kubernetes) para despliegue.

Objetivo: Asegurar que la infraestructura soporte la operación estable del sistema.

VII. Documentación Arquitectónica

La especificación de la arquitectura debe estar documentada para guiar a los equipos de desarrollo.

Componentes clave:

- Descripción de alto nivel del sistema.
- Diagramas estructurales y de flujo de datos.
- Justificación de las decisiones tomadas.

Herramientas comunes: PlantUML, Lucidchart, o herramientas integradas en IDEs.

Conclusión

La especificación de la arquitectura es una etapa fundamental en el desarrollo de software que garantiza la alineación técnica con las necesidades del negocio. Una arquitectura bien diseñada no solo facilita el desarrollo y mantenimiento, sino que también reduce riesgos y asegura la escalabilidad y la adaptabilidad del sistema en el futuro. La clave del éxito radica en una planificación meticulosa, una documentación clara y una evaluación constante para adaptarse a los cambios y desafíos del entorno.

Prieto, E. (s. f.). *¿Cuáles son las etapas del Desarrollo de Software? - Tiffin*

University. Tiffin University.

<https://global.tiffin.edu/blog/cuales-son-las-etapas-del-desarrollo-de-software>

Alberto, P. G. B. L. (2020). *Propuesta de una guía de planificación y gestión de proyectos de desarrollo de software siguiendo los lineamientos del PMBOK y CMMI-DEV en una entidad pública.*

<https://repositorio.utp.edu.pe/handle/20.500.12867/3401>

Ibm. (2024, 24 septiembre). Desarrollo de software. *IBM.*

<https://www.ibm.com/mx-es/topics/software-development>