

Universidad Mariano Gálvez de Guatemala

Ingeniería en sistemas de la información



Avance de Proyecto 07 – Diseño Arquitectónico

Curso: Análisis de Sistemas II

Catedrático: Ing Miguel Catalán

06-10-2023

INTEGRANTES

Jairo Samuel de León de León 7590-20-19534

Madelline Yocary Coronado Duarte carne: 7590-19-2044

Marco Vinicio Rac Cotzoyay 7590-20-17448

Angel Jeronimo Quiej Chacach 7590-20-3143

José Benjamin Méndez Salguero 7590-20-812

MODELADO C4

El modelo C4 es un enfoque para visualizar la arquitectura de software que se basa en abstracciones y diagramas jerárquicos.

Proporciona abstracciones jerárquicas: sistemas de software, contenedores, componentes y código.

Utiliza diagramas jerárquicos que incluyen:

- Diagrama de contexto del sistema.
- Diagrama de contenedor.
- Diagrama de componentes.
- Diagrama de código.

Es independiente de la notación, lo que significa que no prescribe una notación específica.

Es independiente de las herramientas utilizadas.

Usos y beneficios:

- Facilita la comunicación de la arquitectura de software tanto dentro como fuera de los equipos de desarrollo.
- Ayuda en la incorporación eficiente de nuevo personal.
- Es útil para revisiones y evaluaciones de arquitectura.
- Facilita la identificación de riesgos y el modelado de amenazas.
- Compara la forma en que los arquitectos de software suelen comunicar la arquitectura con la industria de la construcción.
- Señala que el modelo C4 se creó para resolver problemas de comunicación visual en el desarrollo de software.

Niveles del modelo C4:

Diagrama de contexto del sistema: Muestra cómo el sistema de software encaja en su entorno.

Diagrama de contenedor: Detalla los componentes técnicos de alto nivel del sistema.

Diagrama de componentes: Descompone un contenedor en componentes individuales.

Diagrama de código: Amplía un componente individual para mostrar su implementación.

Abstracciones del modelo C4:

Persona: Representa usuarios humanos del sistema.

Sistema de software: El nivel más alto de abstracción que ofrece valor a los usuarios.

Contenedor: Representa aplicaciones o almacenes de datos que se ejecutan por separado.

Componente: Agrupación de funcionalidades relacionadas detrás de una interfaz bien definida.

Diagrama de Contexto del Sistema: Este tipo de diagrama proporciona una vista general del sistema de software, mostrando el sistema central rodeado por usuarios y otros sistemas con los que interactúa. Está diseñado para presentar una visión panorámica del sistema y es útil para personas tanto técnicas como no técnicas. El enfoque está en actores, roles y sistemas de software.

Diagrama de Contenedor: El diagrama de contenedor se centra en la estructura de alto nivel del sistema de software y cómo se distribuyen las responsabilidades en él. Muestra los contenedores (como aplicaciones web, aplicaciones móviles, bases de datos, etc.) y cómo se comunican. Es útil para desarrolladores y personal de soporte/operaciones.

Diagrama de Componentes: Este diagrama descompone aún más los contenedores en componentes, mostrando los bloques estructurales, sus responsabilidades y detalles de tecnología/implementación. Está dirigido a arquitectos y desarrolladores de software.

Diagrama de Código: El diagrama de código se centra en la implementación del código, utilizando diagramas de clases UML o similares. Suele ser detallado y se recomienda solo para componentes críticos o complejos. Está destinado a arquitectos y desarrolladores de software.

Diagrama de Paisaje del Sistema: Este diagrama proporciona una visión más amplia de cómo encajan múltiples sistemas de software en una empresa u organización. Es similar al Diagrama de Contexto del Sistema pero a nivel de empresa u organización.

Diagrama Dinámico: Este tipo de diagrama se utiliza para mostrar cómo interactúan los elementos del modelo estático en tiempo de ejecución para

implementar características o historias de usuario. Puede basarse en diagramas de comunicación UML o de secuencia.

Diagrama de Implementación: Este diagrama ilustra cómo se implementan las instancias de sistemas de software y contenedores en la infraestructura en un entorno de implementación específico. Puede incluir nodos de infraestructura como servicios DNS, equilibradores de carga, etc.

Notación Sugerida:

- Utiliza cuadros para representar abstracciones.
- Usa etiquetas para describir elementos.
- Las relaciones entre elementos se representan con flechas etiquetadas.
- Se pueden utilizar colores y formas para complementar la notación.

C4 y UML:

Se puede implementar utilizando UML, pero los diagramas tienden a ser menos descriptivos en términos de texto que los diagramas tradicionales UML.

C4 y ArchiMate:

Es posible crear diagramas C4 utilizando ArchiMate.

Clave/Leyenda del diagrama:

Cada diagrama debe tener una clave que explique la notación utilizada.

Visualizaciones alternativas:

No es necesario utilizar siempre diagramas tradicionales; existen otras visualizaciones interactivas que pueden mostrar las mismas abstracciones del modelo C4 de manera diferente.