

Arquitecturas de Software



Universidad
del Cauca

Vigilada Mineducación

Laboratorio de Ingeniería de Software II

Practica de Laboratorio No. 2

Presentado por:

Jeferson Castaño Ossa

David Santiago Giron Muñoz

Profesor:

Ricardo Antonio Zambrano Segura

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Ingeniería de Sistemas

Popayán, Septiembre de 2023

Arquitecturas de Software

Objetivo: Analizar diferentes estilos de arquitecturas de software, y desarrollar criterios para seleccionar la arquitectura apropiada para un sistema.

1. ¿Qué es una arquitectura de software?

Una arquitectura de software es una estructura conceptual o modelo que define cómo los componentes de un sistema de software interactúan entre sí. Define la organización de los elementos del software, como módulos, capas y componentes, así como las relaciones y comportamientos entre ellos. Una buena arquitectura de software facilita la comprensión, el diseño, la implementación y el mantenimiento del sistema. De esta forma se dispone de una guía teórica detallada que nos permite entender cómo van a encajar cada una de las piezas de nuestro producto o servicio.

La arquitectura del software es particularmente importante porque la estructura del sistema afecta directamente su capacidad para cumplir con los llamados indicadores de calidad del sistema. Ejemplos de atributos de calidad son el rendimiento, que se refiere al tiempo de respuesta del sistema a las solicitudes que se le realizan, la usabilidad, que se refiere a lo fácil que es para los usuarios realizar operaciones en el sistema, o la modificabilidad, que se refiere a lo fácil que es hacer cambios en el sistema. Los indicadores de calidad forman parte de los requisitos (no funcionales) del sistema y son características que deben expresarse cuantitativamente. Por ejemplo, no tiene sentido decir que el sistema debería devolver una solicitud "rápidamente" o proporcionar una página "liviana", porque no es posible evaluar objetivamente si el sistema cumple con esos requisitos o no. La manera en que se estructura un sistema permitirá o impedirá que se satisfagan los atributos de calidad.

Una arquitectura de Software está compuesta por más arquitecturas de datos articuladas entre sí. Esta es la razón por la que la arquitectura de sistema abarca tantos elementos y herramientas para llevarse a cabo. Implantar una arquitectura de software capacitada para los intereses de una compañía deriva en la reducción de costos, mejora en la toma de decisiones, efectividad de rutas de acción y proyección acertada. Una arquitectura de software o arquitectura y diseño de software requiere de un determinado ciclo de etapas que constituyen sistemas: requisitos, diseño, documentación y evaluación de su funcionamiento.

2. ¿Cuál es el papel de un arquitecto de software?

Al igual que el papel de un arquitecto en la edificación de un edificio, un arquitecto de software es el encargado de proyectar, diseñar y dirigir la construcción de un

proyecto de software. En estas actividades, algunas de las funciones más importantes que realiza un arquitecto de software son:

- **Concepción del proyecto:** El arquitecto de software debe analizar e identificar las partes esenciales en un proyecto y las necesidades que se tengan. De tal manera que dirija estas necesidades hacia una solución técnica preliminar, además de velar por que se tengan atributos de calidad viables y requerimientos adecuados que direccionan el diseño arquitectónico de la solución.
- **Diseño y planificación del software:** El arquitecto de software es el encargado de diseñar todo el sistema en base a los requisitos ya definidos, de manera que se brinde una solución técnica que los satisfaga. Para esto define la arquitectura global que seguirá el sistema, así como los componentes y módulos esenciales que lo compondrán.
- **Tecnologías:** Debe poder elegir e identificar las tecnologías, herramientas y plataformas que sean apropiadas para el proceso de desarrollo del software y la aplicación de cada componente así como la conexión entre ellos.
- **Estándares de desarrollo:** Durante el proceso de diseño, establece y documenta estándares de desarrollo que se utilizaran en el proyecto, así como los patrones que se seguirán.
- **Comunicación:** Además de habilidades duras y proponer el diseño del sistema, el arquitecto de software también debe tener la habilidad de comunicar al equipo de desarrollo los diseños y decisiones que se toman. Esto con el fin de asegurar la implementación correcta de la arquitectura y colaborar también con su desarrollo. De esta manera el arquitecto juega un papel de mentor y supervisor de los otros miembros del equipo.
- **Revisiones y pruebas del sistema:** Al momento de realizar pruebas, el arquitecto toma un papel fundamental, pues es el encargado de revisar y realizar auditorías del código del proyecto para garantizar la calidad y la consistencia del software, así como de los atributos de calidad del sistema.
- **Garantizar la seguridad del sistema:** Debe identificar y mitigar los riesgos técnicos y limitaciones del sistema, para que se pueda lograr la eficiencia, adaptabilidad y seguridad del software.
- **Documentación:** El arquitecto documenta la arquitectura y el diseño técnico del sistema, así como colaborar en la documentación general del proyecto para que el equipo de desarrollo y otros interesados puedan comprender y colaborar efectivamente.
- **Seguimiento y mantenimiento:** Una vez el sistema está implantado en el ambiente productivo, la participación del arquitecto sigue siendo fundamental, pues muchas veces es necesario realizar ajustes de la aplicación para lograr un funcionamiento óptimo de ésta, por lo que se debe mantener un seguimiento y mejora continua de la arquitectura implantada.

De esta manera el arquitecto de software juega un papel fundamental durante el desarrollo de un proyecto de software, tomando decisiones críticas que definen el curso de este. Es responsable de diseñar, guiar y supervisar la arquitectura técnica del sistema de software y garantizar que sea robusto y eficiente, siendo su papel crucial para el éxito del proyecto.

3. Seleccione dentro de la siguiente lista de estilos arquitectónicos dos para su estudio en profundidad:

Tuberías y filtros
Pizarra
Capas
Orientados a Objetos
REST
Máquina Virtual
Modelo Vista controlador

Analice sus características, principales ventajas, desventajas y sistemas en los cuales es recomendable su uso. Construya una tabla comparativa.

	Tuberías y filtros	Capas
Características	<p>División en Etapas: El sistema se divide en una serie de etapas o pasos independientes, cada una de las cuales realiza una tarea específica. Estas etapas se conocen como "filtros".</p> <p>Procesamiento Secuencial: Los datos fluyen a través de las etapas en secuencia, de una a otra, en un flujo continuo. Cada filtro toma los datos de entrada, realiza alguna transformación o procesamiento y pasa los datos transformados a la siguiente etapa.</p> <p>Independencia de Filtros: Cada filtro es independiente de</p>	<p>Separación de Responsabilidades: Se basa en la separación de componentes en capas lógicas, donde cada capa tiene una responsabilidad específica y única. Una capa es una colección conexa de módulos, donde cada uno de los cuales puede ser accedido o invocado por otra capa inferior o superior.</p> <p>Jerarquía de Capas: Las capas se organizan en una jerarquía, con capas superiores que dependen de capas inferiores. Las capas superiores tienen</p>

	<p>los demás y no necesita conocer detalles sobre los filtros adyacentes. Esto facilita la modularidad y la reutilización de componentes.</p> <p>Conexiones Simples: Las conexiones entre filtros son simples y unidireccionales, lo que facilita la expansión y la reconfiguración del sistema agregando o eliminando filtros.</p>	<p>una gran flexibilidad y se centran en el modelo del negocio y servicios disponibles para su uso. Las capas inferiores son mucho más restringidas y dependientes de la plataforma. Una capa se descompone en segmentos.</p> <p>Interfaz: Cada capa se comunica con las capas adyacentes a través de interfaces definidas, donde se especifica cómo se puede acceder y utilizar los servicios de cada capa.</p> <p>Relación “Allow-to-use”: Es la relación esencial en la arquitectura. Es una autorización la cual determina quién tiene permiso para acceder o utilizar componentes y recursos específicos en cada capa.</p>
Ventajas	<p>Modularidad: El enfoque de tuberías y filtros permite diseñar sistemas altamente modularizados, lo que facilita el desarrollo, el mantenimiento y la reutilización de componentes.</p> <p>Paralelismo: Si las etapas son independientes y no dependen de los resultados de otras etapas, es posible implementar procesamiento paralelo para mejorar el rendimiento en sistemas multicore.</p>	<p>Flexibilidad y Adaptabilidad: Tener capas independientes permite agregar, quitar o modificar una capa sin afectar el funcionamiento de las demás, lo que permite que el sistema sea más flexible y adaptable a cambios.</p> <p>Seguridad: El control basado en capas permite aplicar medidas de seguridad específicas en</p>

	<p>Facilidad de Mantenimiento: Debido a la separación clara de las etapas, es más fácil depurar y mantener cada filtro individualmente sin afectar otros componentes.</p> <p>Permite entender el sistema global en términos de la combinación de componentes.</p> <p>Soporta de buena manera la reutilización. Los filtros son independientes de sus vecinos.</p>	<p>cada capa, de tal manera que se limita el acceso a determinadas capas y se fortalece la seguridad general del sistema.</p> <p>Mantenimiento: Con capas las cuales tienen responsabilidades específicas, es más fácil identificar y corregir problemas, así como realizar actualizaciones y mejoras sin afectar todo el software.</p> <p>Evolución: El sistema resulta fácilmente escalable, además de la incorporación de nuevas tecnologías y características en el futuro.</p>
Desventajas	<p>Overhead: La comunicación entre etapas a través de tuberías introduce cierto overhead en términos de rendimiento y consumo de recursos.</p> <p>Limitado a Flujos Secuenciales: El estilo de tuberías y filtros es adecuado para flujos de datos secuenciales y procesamiento en serie, pero puede no ser la mejor opción para problemas que requieran interacción o retroalimentación entre etapas.</p> <p>Complejidad en la Coordinación: La coordinación de múltiples filtros y la gestión de errores puede ser</p>	<p>Complejidad: Con el enfoque de capas se puede aumentar la complejidad del sistema a medida que se implementan múltiples capas, lo que dificulta la comunicación y sincronización de datos entre ellas.</p> <p>Recursos: La modularidad de capas requiere de un mayor tiempo de desarrollo y costos para diseñar e implementar adecuadamente las interfaces y el sistema en general. Además de recursos computacionales al mantener múltiples capas y la comunicación</p>

	complicada, especialmente en sistemas grandes.	entre ellas. Comunicación: Se tiene una mayor latencia y dificultad de sincronización y coordinación, pues llegar a una capa o realizar un proceso significa haber hecho un recorrido por todas las demás, teniendo un sistema complejo que requiere una gestión y mantenimiento cuidadoso.
Sistemas recomendados	<p>Procesamiento de Datos en Batch: Es útil para aplicaciones de procesamiento en lotes donde se pueden dividir los datos en etapas de procesamiento independientes.</p> <p>Procesamiento de Datos en Tiempo Real: En aplicaciones donde se requiere procesamiento en tiempo real de flujos de datos, como sistemas de filtrado de señales, el estilo de tuberías y filtros puede ser apropiado.</p> <p>Procesamiento de Datos de ETL (Extract, Transform, Load): En el campo de la inteligencia empresarial y la extracción de datos, se puede utilizar para limpiar, transformar y cargar datos desde múltiples fuentes.</p> <p>Procesamiento de Imágenes y Multimedia: En aplicaciones que implican manipulación de imágenes, audio o video, el estilo de tuberías y filtros puede</p>	<p>Aplicaciones Empresariales: La arquitectura de capas ayuda a organizar y gestionar estos sistemas los cuales son complejos al integrar funciones como finanzas, recursos humanos, inventario y producción.</p> <p>Sistemas de Planificación de Recursos Empresariales: Debido al uso de múltiples módulos y funcionalidad, la arquitectura de capas resulta útil, pues permite separar la lógica de negocio, la presentación y el almacenamiento de datos.</p> <p>Aplicaciones Web: Ayuda a separar la lógica de presentación front-end y la lógica de negocio back-end.</p> <p>Aplicaciones Móviles: La</p>

	ser beneficioso para aplicar filtros y efectos.	<p>lógica de negocio de la interfaz de usuario y de gestión de datos se pueden implementar mediante una arquitectura de capas.</p> <p>Sistemas de Gestión de Almacenes y Logística: Necesitan gestionar inventario, envíos y seguimiento de productos, donde la arquitectura de capas facilita la gestión de estas funcionalidades.</p>
--	---	--

4. Recientemente ha habido muchas inconformidades con el censo en Colombia que se realizó en 2018, el gobierno ha decidido estudiar la posibilidad de realizar uno nuevo para el año 2022 y así corregir los errores del anterior para tener unas cifras más reales, las cuales además han cambiado dado el fenómeno migratorio venezolano. Su empresa ha ganado la licitación para este nuevo proyecto, y una de sus primeras decisiones será la de seleccionar una arquitectura adecuada para el nuevo sistema.

¿Considera que alguna de las arquitecturas estudiadas sería útil para esta tarea?, ¿por qué sí?, ¿por qué no?

Para ello revise bien que es un censo, ¿cómo se realiza?, ¿qué tipo de información se colecta?, ¿qué información debe arrojar?, ¿qué componentes debe tener un sistema que lo soporte, igualmente revise que atributos de calidad debe tener.

Recuerde somos más de 50 millones de Colombianos hoy (50.270.571).

Teniendo de base los aspectos característicos que se deben tener presentes a la hora de realizar un censo para un país como Colombia, con su gran número de habitantes y sus especiales características geográficas, consideramos que para la arquitectura general del sistema, la mas optima (de las estudiadas) es la arquitectura de software en capas.

Un sistema en capas nos brinda la capacidad de separación de responsabilidades; una característica de gran ayuda en un proyecto donde se involucran una gran variedad de tareas como lo es un censo nacional,

donde se tiene desde la recopilación de datos hasta el procesamiento y la presentación de resultados.

Además es de recalcar que se necesita un sistema capaz de manejar un alto volumen de datos durante un amplio periodo de tiempo, por lo que pueden surgir un gran número de problemas con las tecnologías usadas durante ese lapso, además de nuevas reglas de negocio en el sistema. Por ello, el software debe ser fácil de mantener, así como tener la capacidad de crecer en el tiempo, cualidades que son posibles de lograr con la arquitectura de capas, pues cada capa se puede trabajar, probar, mantener y escalar de manera independiente, lo que nos permite manejar óptimamente el proyecto.

Dado que la seguridad y la privacidad de los datos del censo son de suma importancia, la arquitectura de capas permite implementar medidas de seguridad en las interfaces entre capas para proteger los datos y garantizar que solo se acceda a funciones autorizadas.

De esta manera, la arquitectura de capas nos proporciona una estructura organizada, mantenible, modulada, segura y clara para el censo nacional. Aspectos fundamentales para la calidad y el éxito del proyecto. Sin embargo el desarrollo de esta arquitectura puede resultar en costos altos en términos de tiempo y dinero por su gran complejidad, misma que también puede ser un aspecto negativo si no se planea con correcta antelación, pues fácilmente puede desembocar en una complejidad innecesaria que haría la compresión y mantenimiento del software más difícil.

Por otro lado, con lo estudiado en lo que respecta al funcionamiento especial que debe haber dentro de un censo, llegamos a la conclusión de que usar una única arquitectura para abordar diferentes aspectos del sistema no sería completamente óptimo, pues es casi inherente combinar diferentes arquitecturas para adaptarse a las necesidades específicas del proyecto. Es por esto que consideramos que la arquitectura de tuberías y filtros también sería útil en la implementación del nuevo sistema en el componente de procesamiento de datos para el censo nacional.

La arquitectura de Tuberías y Filtros es adecuada para procesar los datos de un censo nacional que suelen ser recopilados a medida que los encuestadores o sistemas de recolección recopilan información. Esto mediante su enfoque secuencial a través de los filtros, donde se aplican transformaciones y análisis específicos. De esta manera, dentro del censo los filtros pueden representar diferentes etapas de procesamiento, como validación, limpieza, agregación, análisis estadístico, etc.

Es de aclarar que este tipo de arquitectura solo puede ser utilizada dentro del campo de procesamiento de datos, pues en otros componentes no

puede ser la más óptima. Así, junto con la arquitectura de capas pueden trabajar de forma tal que beneficie enormemente las necesidades específicas del nuevo sistema para el censo nacional.

Referencias

- Huet, P. (24 de Agosto de 2022). *Arquitectura de software: Qué es y qué tipos existen*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- ¿Qué es una Arquitectura de Software? (s.f.). Obtenido de KEEPCODING: <https://keepcoding.io/blog/arquitectura-de-software/>
- Cervantes, H. (s.f.). *Arquitectura de Software*. Obtenido de SG: <https://sg.com.mx/revista/27/arquitectura-software>
- Perfil de Arquitecto de Software*. (s.f.). Obtenido de MichaelPage: <https://www.michaelpage.es/advice/profesión/tecnología/perfil-de-arquitecto-de-software#:~:text=El%20Arquitecto%20de%20Software%20tiene,la%20dirección%20de%20la%20compañía>
- Vilela, C. (04 de Mayo de 2022). *¿Qué hace un arquitecto de software?* Obtenido de freelancemap: <https://www.freelancemap.com/blog/es/que-hace-arquitecto-software/>
- Cervantes, H. (s.f.). *El Rol del Arquitecto de Software*. Obtenido de SG: <https://sg.com.mx/revista/33/el-rol-del-arquitecto-software>
- gomez, k. r. (26 de Febrero de 2015). *Arquitectura flujo de datos(filtros y tuberías)*. Obtenido de slideshare: <https://www.slideshare.net/ktrevelo/arquitectura-flujo-de-datosfiltros-y-tuberías>
- Durán, M. (11 de Abril de 2023). *Qué es la arquitectura en capas, ventajas y ejemplos*. Obtenido de HubSpot: <https://blog.hubspot.es/website/que-es-arquitectura-en-capas#:~:text=La%20arquitectura%20en%20capas%20es,adyacentes%20mediante%20interfaces%20bien%20definidas.>