**SERVICIO NACIONAL DE APRENDIZAJE – SENA**

Logotipo

Descripción generada automáticamente

**CENTRO DE COMERCIO REGIONAL ANTIOQUIA**

**TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE - 2675805**

**Evidencia de producto: GA4-220501095-AA2-EV06 - Taller arquitectura de software**

**DANIEL FELIPE ARIAS CORREDOR**

**2023**

*Introducción*

El presente documento tiene como fin dar solución a las preguntas formuladas para el taller de arquitectura de software, mediante la consulta bibliográfica acerca de este tema.

*Preguntas*

* ¿Qué entiende por arquitectura de software?

Una arquitectura de software es un patrón o lineamiento comúnmente usado en la construcción de una aplicación de software. Estos lineamientos están basados en modelos teóricos, patrones y prácticas comúnmente usadas por la industria. La implementación de una arquitectura de software precede al desarrollo de la aplicación.

* ¿Cuál es su función?

La arquitectura de software permite una planificación previa de la estructura interna que va a tener la aplicación. Provee información acerca del tiempo de desarrollo y los recursos económicos y humanos que se van a necesitar. Puesto que las arquitecturas son estándares comúnmente usados en la industria, permiten la reutilización de módulos y la modificación de código por parte de otros miembros del equipo, ya que se convierte en un lenguaje común para los desarrolladores. Aparte de lo anterior, una arquitectura bien estructurada aportará a la aplicación orden, seguridad en el acceso a datos y futura conexión con otras aplicaciones.

* ¿Cómo se elabora la arquitectura de software?

La arquitectura de software se realiza mediante la definición de la estructura de la aplicación a partir de un modelo de arquitectura escogido. Este proceso consta de cuatro etapas:

1. Levantamiento de requerimientos: Etapa de recolección de información y de los requerimientos (funcionales y no funcionales) del producto, en tanto afecten la estructura de la aplicación. A partir de estos requerimientos y de la información que provean las partes interesadas del proyecto, se empieza a pensar en su estructura.
2. Diseño: Etapa donde se define el uso de tecnologías necesarias para resolver los requerimientos. Es en este momento donde se escoge el modelo de arquitectura que pueda ejecutar satisfactoriamente las funcionalidades de la aplicación. Ejemplos de arquitectura son el patrón cliente-servidor, por capas, hexagonal o microservicios. Cada una cuenta con una estructura, forma de responder a solicitudes e interactuar con el usuario.
3. Documentación: A partir de la arquitectura escogida, se crea una documentación que sirva como marco de referencia para el trabajo de todo el equipo de desarrollo.
4. Evaluación: Fase de escrutinio. Se revisa si el diseño pueda funcionar o si requiere modificaciones. El diseño puede evaluarse antes de escribir código, por medio de la retroalimentación de los desarrolladores o durante el desarrollo realizando pruebas y métricas.

* ¿Cómo lograr una buena arquitectura?

Para lograr una buena arquitectura se deben seguir una serie de principios, los cuales apelan sobre todo a facilitar la realización de cambios. Entre los principios (Cervantes et.al, 2016) se encuentran:

* Modularidad: Se logra al descomponer en partes o paquetes el código de la aplicación. Permite el desarrollo en paralelo de varios componentes. Si se debe realizar un cambio, basta con modificar un módulo y no un script de código completo. El nivel de modularidad se debe considerar de acuerdo con los atributos de la aplicación.
* Cohesión alta y acoplamiento bajo: los módulos deben estar enfocados a la realización de una única tarea, por lo que un módulo multitarea se considera de cohesión baja. El acoplamiento bajo se refiere al encapsulamiento de los módulos. Al no conocer su funcionamiento y variables internas y no depender tanto uno de otro, modificar un módulo tiene un impacto nulo o bajo en los demás.
* Mantener simples las cosas: Conocido como principio KISS, hace hincapié en mantener el diseño lo más simple posible, enfocándose en satisfacer los atributos de calidad requeridos y no más de estos, en lugar de diseñar soluciones demasiado elaboradas.
* ¿Cuáles son los elementos de diseño de una arquitectura de software?

Loes elementos que conforman el diseño de la arquitectura pueden variar de un patrón a otro, pero existen una serie de elementos comunes, que sirven para dar solución a los requerimientos. Entre estos se encuentran:

* Cliente: El usuario, ya sea persona o una API que hace solicitudes de información o bien introduce data nueva. Es el usuario a quien va destinada la aplicación
* Interfaces de usuario: Es la forma en que se muestra la aplicación y los datos a presentar al usuario. Consisten en una serie de interfaces intuitivas y de fácil manejo, las cuales presentan o captan la data del sistema. Se conoce como la parte frontend de la aplicación.
* Backend: Es el conjunto de módulos o paquetes que trabajan tras bambalinas para manejar la información que viene del frontend y se almacena en base de datos o viceversa. A parte de ser el transporte de la información, es donde la aplicación internamente la manipula y realiza sus operaciones internas. Su estructura puede variar de acuerdo con el patrón de arquitectura, pero es donde reside la lógica de negocio. Puede contener controladores, módulos unifuncionales, entre otros.
* Gestión de base de datos: Es la parte enfocada en la persistencia de memoria. Consiste en las bases de datos propiamente dichas y los módulos de acceso a data. Las bases de datos generalmente están aisladas de las interfaces y según el diseño están separadas lo mas que se pueda de las interfaces de usuario.
* Servidor: Es la plataforma que soporta y atiende las peticiones del usuario. Es el encargado de ofrecer el servicio de almacenamiento de la data y de la aplicación.

Conclusión

La arquitectura de software es un proceso fundamental en la planeación de un proyecto de software. Una correcta implementación permite un proceso de desarrollo mejor planificado, entendible para todo el equipo de desarrollo y un producto final que atienda oportunamente a sus requerimientos. La familiarización con los patrones de software mas usados en la industria es de vital importancia para todo desarrollador.

*Referencias*

Cervantes, H., Velaso-Elizondo, P., Castro, L. (2016). *Arquitectura de software. Conceptos y ciclo de desarrollo*. Cengage Learning Editores, Mexico DF.

OpenWebinars. *Arquitectura de software: Qué es y qué tipos existen*. Recuperado de: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>

Platzi. *Qué es la arquitectura de software: más allá de la programación.* Recuperado de: <https://platzi.com/blog/que-es-arquitectura-de-software/>