

Arquitectura Alissta

Sobre el documento

El presente documento contiene el diseño elaborado para el proyecto POSITIVA SG-SST, el cual es producto de un análisis minucioso de los requisitos del sistema, según estos pueden ser satisfechos con las tecnologías y características discutidas con los clientes y usuarios.

El documento está organizado alrededor de tres ideas principales:

1. Las características generales del diseño.
2. Los requisitos atendidos por el diseño.
3. Los modelos y vistas que lo detallan.

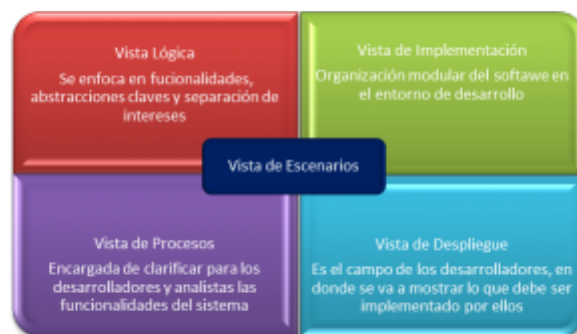
Al contrario de muchas otras actividades técnicas, el desarrollo de sistemas intensivos en software dedica la mayoría de sus esfuerzos a la especificación y modelado. Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación. Los modelos son presentados por medio de vistas o diagramas, generalmente utilizando notaciones gráficas como el UML. Por otro lado, los programas de computadora son construidos por medio del uso de herramientas de traducción automáticas llamados compiladores, para los cuales es construida la forma final y más detallada del software del sistema: el código fuente. La última sección del documento indica la forma en que se puede obtener el código fuente del proyecto, así como las instrucciones de compilación necesarias para lograr la ejecución de los componentes que este código detalla.

Descripción General

Se entiende por arquitectura del software, como el conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución.

Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto. Esta sección describe en términos generales, las ideas principales detrás de la arquitectura escogida para el mismo.

El diseño será representado por medio del modelo de "4+1 Vistas" (Kruchten, vol. 12), cuyo objetivo es mostrar, en cada una de las vistas, una perspectiva o visión de un conjunto de elementos del proyecto y sus relaciones, esto desde el punto de vista de la arquitectura. En unión, las 4+1 vistas representan las decisiones de diseño y la forma como se desarrollará el proyecto. Este modelo posee un alto grado de importancia debido que está estrechamente relacionado con todos los Stakeholders según su rol dentro del desarrollo del proyecto. El modelo se divide en 4+1 vistas que se describen en la Figura 1.



El modelo de vistas múltiples, organiza una descripción de la arquitectura de software utilizando cinco vistas concurrentes, las cuales permiten aproximar de manera aislada los intereses de los diferentes stakeholders de la arquitectura: los usuarios finales, los desarrolladores, entre otros; y manejar de manera separada los requerimientos funcionales y no funcionales. Se capturan las decisiones de diseño en cuatro de las vistas y utilizan la quinta vista para ilustrar y validarlas.

El modelo propone las siguientes perspectivas o vistas:

1. **Vista lógica:** Ofrece soporte a los requerimientos funcionales, lo que el sistema debe proveer en términos de servicios a sus usuarios.
2. **Vista de procesos:** La vista de procesos permite describir los procesos del sistema y como estos se comunican.
3. **Vista física o de despliegue:** La vista física describe como es instalada la aplicación y como se ejecuta en una red de computadores. Para describir esta vista, en el presente documento se utilizó un diagrama de despliegue.
4. **Vista de desarrollo o de implementación:** Esta vista se concentra en la organización en módulos del software. En el presente documento, fue representada por el diagrama de paquetes.
5. **Vista de casos de uso:** La vista de casos de uso consolida las vistas anteriores, donde los escenarios se convierten en una abstracción de los requerimientos más importantes. Para describir esta vista, en el presente documento como anexo se encuentran los casos de uso.

Una vez explicadas las vistas que propone Kruchten y la forma de documentarlas, se puede apreciar que es un modelo bastante bueno para documentar la arquitectura de un sistema software “complejo”, ya que todos los stakeholders pueden entender el sistema que se está desarrollando desde diferentes perspectivas.

Posicionamiento y Alcance

From:
<http://wiki.adacsc.co/> - Wiki

Permanent link:
http://wiki.adacsc.co/doku.php?id=ada:arquitectura_alissta&rev=1642717763

Last update: 2022/01/20 22:29



