



ITLA

Instituto Tecnológico de Las Américas

Arquitectura de software, Infraestructura de software y Patrones y
Diseños de Software

Definición, Origen, Evolución, Características, tipos y roles

Francisco Javier Batista Acosta

2021-0952

Arquitectura de Software

➤ Definición

La arquitectura de software son patrones o lineamientos que ayudan a la construcción de una aplicación. Los patrones permiten tener una guía para los analistas, desarrolladores y los cargos relacionados para lograr cumplir con los requerimientos de la aplicación.

Básicamente la arquitectura de software nos traza un camino por seguir para obtener nuestra meta y cumplir los requerimientos de la aplicación.

La arquitectura a nivel de software.

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

Una arquitectura de software se selecciona y diseña con base en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como el mantenimiento, la auditoría, flexibilidad e interacción con otros sistemas de información.

➤ Origen

El origen de la arquitectura de software se remonta al año 1960, no obstante toma popularidad en los años 1990 después de reconocer la denominada crisis del software y como tema de interés de la incipiente disciplina de la ingeniería del software.

➤ Evolución

La evolución de arquitecturas de software en consecuencia de cambios como la redefinición de requerimientos, o mejoras en la infraestructura/tecnología del sistema.

El desarrollo de la arquitectura ha pasado el proceso de evolución desde la arquitectura monolítica, la arquitectura vertical, la arquitectura distribuida hasta la computación móvil.

➤ Características

Los puntos que influyen en la aptitud de la arquitectura de software son la planificación de proyectos, el análisis de riesgo, la organización, el proceso de desarrollo, los ciclos de trabajo, el hardware, la garantía de calidad y los requerimientos.

➤ Tipos de arquitecturas

Las Arquitecturas más comunes y utilizadas son:

- **Descomposición Modular.** En este el software se estructura en grupos funcionales y muy acoplados.
- **Cliente-Servidor.** En este el software divide sus cargas en dos partes independientes de la otra, pero sin reparto claro de funciones.
- **Arquitectura de 3 niveles.** Es un tipo mejorado del cliente-servidor donde las cargas se dividen en 3 partes o capas donde las funciones están claras, una capa se encarga de la interfaz del usuario, otra para los cálculos donde se encuentra el negocio y otra para el almacenamiento. Cada una tiene relación con solo una capa no entre las 3

➤ Roles o responsabilidades de un arquitecto de Software

- Un Arquitecto de software tienes varios roles claves como:
- La responsabilidad de conciliar las demandas de los clientes y de la organización de desarrollo.
- Encargarse de los proyectos de desarrollo y comunicar los avances a la dirección de la compañía.
- Satisfacer las necesidades de los clientes y/o de la empresa
- Llevar cada cierto tiempo una verificación de código para prevenir y corregir posibles errores.

Infraestructura de Software

➤ Definición

Es una agrupación de conjuntos de elementos que sirven para el almacenamiento de los datos de un ente. Donde se incluyen el software y los diferentes servicios necesarios para optimizar la gestión interna y seguridad de información.

➤ Origen

El origen de la Infraestructura de software tiene un largo recorrido, en comparación con el hardware hubo menos novedades ya que el desarrollo de ambos iba paralelo o superpuesto. Algunos historiadores al ingeniero mecánico e inventor musulmán Al-Jazari con el primer ejemplo de programación.

➤ Evolución

La infraestructura de TI en las organizaciones actuales es el fruto de más de 50 años de evolución en las plataformas de computadora. Han transcurrido cinco etapas en esta evolución, cada una de las cuales representa una distinta configuración de poder de cómputo y elementos de la infraestructura (vea la figura 5-2). Las cinco eras son la computación con mainframes y minicomputadoras de propósito general, las microcomputadoras mejor conocidas como computadoras personales, las redes cliente/servidor, la computación empresarial y la computación en la nube y móvil.

➤ Características

Características y ventajas principales

- Reduce costes.
- Aumenta eficiencias operativas.
- Simplifica el mantenimiento y soporte.
- Escalabilidad y agilidad de adaptación.
- Disminuye el riesgo de fallos.
- Incrementa la seguridad.
- Mayor capacidad de respuesta.

➤ Tipos de Infraestructura de software

Infraestructura tradicional. Esta compuesta por los componentes habituales del hardware y software: instalaciones, centros de datos, servidores, computadoras de escritorio de hardware de red y soluciones de software de aplicaciones empresariales. Normalmente este tipo de infraestructura requiere de mucha energía, espacio físico y dinero que otras.

Infraestructura de la nube. Es muy similar a la tradicional, con la única diferencia de que los usuarios finales pueden acceder a través de Internet y tienen capacidad de usar la virtualización para ocupar recursos informáticos sin realizar instalaciones locales. Virtualizar conecta los servidores físicos de un proveedor de servicios en cualquier ubicación geográfica.

➤ Rol y Funciones de un Gerente de Infraestructura de software

- Dirigir el equipo y proyectos
- Mantener buenas relaciones comerciales con los proveedores de servicios y productos
- Elaborar registros, documentaciones e informes
- Gestionar y medir las métricas operativas
- Desarrollar estrategias
- Aplicar y optimizar los procesos y procedimientos

Patrones de Diseño y Desarrollo de Software

➤ Definición

Los patrones de diseño y desarrollo de software son tipos de soluciones generales, son reutilizables y aplicables a diferentes problemas de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de pruebas y error.

➤ Origen

En el año de 1994, cuatro autores Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, a los que llamaron Gang Four (GoF), publicaron un libro titulado Design Patterns, elementos de **software orientado a objetos** reutilizables.

Con ese trabajo se inicia el concepto de patrón de diseño en el desarrollo de software y recoge 23 patrones de diseño comunes. Cada uno de estos definen la solución para resolver un determinado problema, facilitando además la reutilización del código.

➤ Evolución

- Proyectos Harvey
- El Apple Learning Interchange (ALI)
- El proyectos Ariadne
- El repositorio Merlot
- El Proyecto Ceti
- Los Generative Learning Object (GLO)
- Los Objetos generativos (GLO)

➤ Características

- Facilitar el aprendizaje a las nuevas generaciones de diseñadores condensando conocimiento ya existente
- Formalizar un vocabulario común entre diseñadores
- Estandarizar el modo en que se realiza el diseño
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente
- Proporcionar catálogos de elementos reusables en el diseño de sistemas de software

➤ Tipos de Patrones de Diseño y Software

Son 3 tipos:

Patrones Creacionales. Esto le da al programa más flexibilidad para decidir qué objetos deben crearse para un caso de uso dado.

Patrones Estructurales. Facilitan soluciones y estándares eficientes con respecto a las composiciones de clase y las estructuras de objetos.

Patrones de Comportamiento. Se utilizan para detectar la presencia de patrones de comunicación ya presentes y pueden manipular estos patrones.

Conclusión

En conclusión, en este trabajo pudimos comprobar la importancia de las tres, Infraestructura, arquitectura y diseño, pudimos ver de donde se originan y que básicamente una se compenetra de la otra.

También pudimos ver las responsabilidades, cargos y roles que tiene un encargado de cada uno de estas.

Bibliografías.

➤ Arquitectura de software

- <https://platzi.com/blog/que-es-arquitectura-de-software/> (definición)
- https://es.wikipedia.org/wiki/Arquitectura_de_software#Breve_rese%C3%B1a_hist%C3%B3rica (definición) (tipos)
- <https://1library.co/document/y69m9j4y-breve-historia-de-la-arquitectura-de-software.html> (historia)
- <https://ri.conicet.gov.ar/handle/11336/6898#:~:text=La%20evoluci%C3%B3n%20de%20arquitecturas%20de%20software%20es%20consecuencia,requerimientos%2C%20o%20mejoras%20en%20la%20infraestructura%2Ftecnolog%C3%ADa%20del%20sistema.> (evolución)
- <https://www.voigtmann.de/es/desarrollo-de-software/arquitectura-de-software/#:~:text=Caracter%C3%ADsticas%20de%20una%20arquitectura%20de%20software%20exitosa&text=Factores%20importantes%20que%20influyen%20la,de%20calidad%20y%20los%20requerimientos.> (características)
- <https://www.michaelpage.es/advice/profesi%C3%B3n/tecnolog%C3%ADa/perfil-de-arquitecto-de-software#:~:text=El%20Arquitecto%20de%20Software%20tiene,la%20direcci%C3%B3n%20de%20la%20compa%C3%B1%C3%ADa.> (roles)

➤ Infraestructura de software

- <https://veggestion.es/la-infraestructura-tecnologica-definicion-tipos-e-importancia/> (definición y características)
- <https://www.ibm.com/es-es/topics/infrastructure> (tipos)
- cavsi.com (Origen)
- scrib.com (evolución)
- <https://www.freelancermap.com/blog/es/que-hace-gerente-infraestructura/#Funciones> (funciones)

➤ Patrones de Diseño y Software

- imetoast.com (evolución patrones de diseño)
- <https://www.fabrigadigital.com/la-importancia-de-los-patrones-de-diseno-software/> (tipos)
- [https://profile.es/blog/patrones-de-diseno-de-software/#%C2%BFQue son los patrones de diseno design patterns](https://profile.es/blog/patrones-de-diseno-de-software/#%C2%BFQue%20son%20los%20patrones%20de%20diseno%20design%20patterns) (definición, historia)