

Investigación Diagrama de Despliegue

Que es

Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta.

Los diagramas de despliegue se utilizan normalmente para visualizar el hardware y el software físico de un sistema. Usándolo puedes entender cómo el sistema se desplegará físicamente en el hardware.

Notaciones del diagrama

Para poder familiarizarse para realizar el diagrama de despliegue debe conocer las siguientes notaciones :

Nodos

Nodo - anotaciones del diagrama de despliegue Un nodo, representado como un cubo, es una entidad física que ejecuta uno o más componentes, subsistemas o ejecutables. Un nodo podría ser un elemento de hardware o software.

Artefactos

- anotaciones Artefactos del diagrama de despliegue Los artefactos son elementos concretos que son causados por un proceso de desarrollo. Ejemplos de artefactos son las bibliotecas, archivos, archivos de configuración, archivos ejecutables, etc.

Asociación de Comunicación

Esto está representado por una línea sólida entre dos nodos. Muestra el camino de la comunicación entre los nodos.

Dispositivos

Dispositivo Un dispositivo es un nodo que se utiliza para representar un recurso físico computacional en un sistema. Un ejemplo de un dispositivo es un servidor de aplicaciones.

Especificaciones de despliegue

Especificación despliegue Las especificaciones de despliegue son un archivo de configuración, como un archivo de texto o un documento XML. Describe cómo se despliega un artefacto en un nodo.

Arquitectura de software

Nos traza un camino por el cual seguir para lograr cumplir con los requerimientos de la aplicación, por lo tanto tenemos que analizar cada uno de estos requerimientos para definir qué se va a hacer y cómo se va a hacer, aquí se comienza a definir temas como servidores, tecnologías, bases de datos entre otras cosas.

El definir las tecnologías es uno de los puntos más importantes de la arquitectura de software pero no quiere decir que si se toma una decisión sea algo definitivo que no se pueda modificar en el futuro. Por ejemplo Uber tenía la parte realtime de los mapas usando node.js, esta fue su primera implementación y funcionaba, llegó un punto donde no estaba escalando de la manera correcta y migraron a usar go para esta parte realtime.

Definición de la estructura

Para definir de la mejor forma la estructura de nuestra aplicación con base en la arquitectura de software tenemos 4 etapas principales que nos ayudarán a realizar de buena manera un proyecto:

Requerimientos: En esta etapa se recolecta la información y se documentan los requerimientos que influyen en la arquitectura de la aplicación, por ejemplo un color del menú no afecta al arquitectura, pero si por ejemplo la forma de guardar la información.

Diseño: Es la etapa más crucial, aquí se define el uso de tecnologías adecuadas para resolver el problema y no solo porque una tecnología está de moda. También se tienen en cuenta patrones como por ejemplo MVC (Modelo, Vista, Controlador) o arquitectura de microservicios.

Documentación: Una vez se ha definido el diseño es necesario comunicarlo de manera eficiente y eficaz a todos los involucrados, es importante crear documentación que sirva como referencia a todos y sea el marco de trabajo para todos.

Evaluación: Es importante luego de tener la documentación evaluar, esto se puede hacer incluso sin haber hecho una línea de código y ver con todos los involucrados si hay algo en el diseño que pueda no funcionar y reformarlo, esta evaluación se debería hacer posterior teniendo métricas por ejemplo del rendimiento de la aplicación y saber si un cambio mejora o no dicho rendimiento.

Infraestructura de T.I (Servidores -Servicios arquitectura en la nube)

La infraestructura de TI es el conjunto y compuestos de software, hardware, redes y servicios conectados que componen el entorno de TI de una organización. Cada componente de la infraestructura de TI brinda muchos servicios distintos y contribuye a la eficiencia general del sistema o proyecto que se realiza .

Arquitectura en la nube:

Hardware de TI

El hardware de TI hace referencia a todas las máquinas y todos los dispositivos físicos que una organización utiliza en su entorno de TI. Los dispositivos de almacenamiento y los servidores que brindan recursos de red a una empresa forman parte del hardware de TI. Todos los dispositivos de punto de conexión, como las computadoras, los teléfonos y las tabletas, entran en esta categoría.

Software de TI

La infraestructura de TI de software incluye lo siguiente:

- Sistemas operativos.
- Middleware.
- Bases de datos.
- Servidores de aplicación.
- Administración de relaciones con los clientes (CRM).
- Software de planificación de recursos empresariales.
- Sistemas de administración de contenidos.
- Software de virtualización.

Servicios en la nube

La infraestructura de servicios en la nube hace referencia a las plataformas, el software y la infraestructura que una empresa de terceros brinda a la organización. Puede utilizar aplicaciones de software como servicio (SaaS) para eliminar la necesidad de aplicaciones locales. Del mismo modo, podría utilizar la infraestructura como servicio (IaaS) para acceder a la infraestructura de TI desde la nube y externalizar la necesidad de servidores.

UML diagrama de despliegue

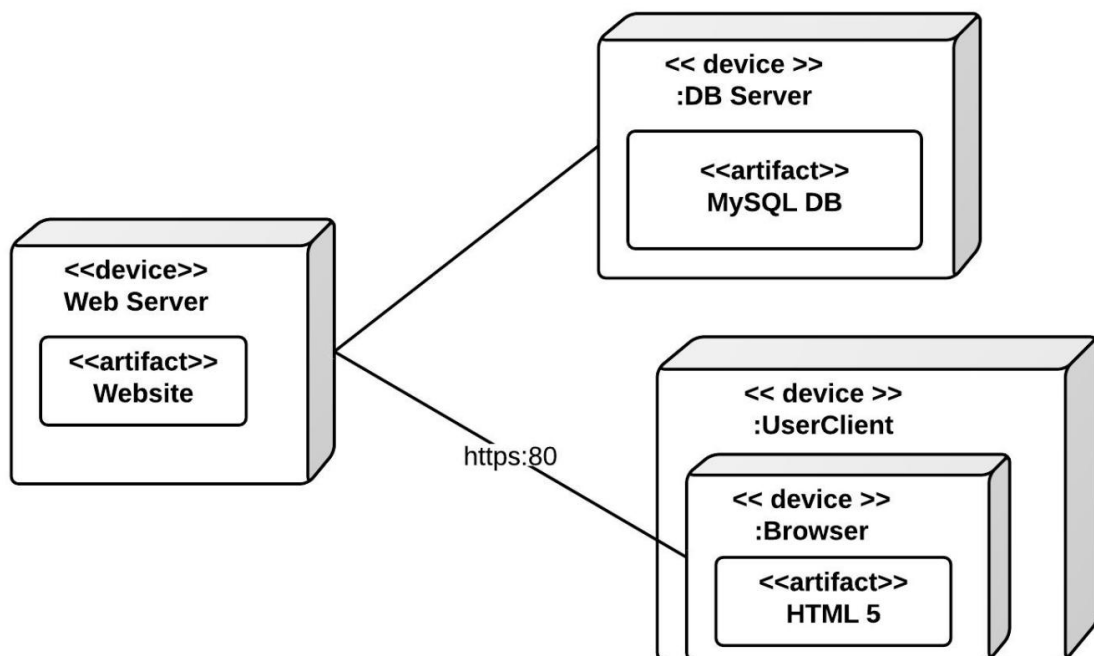
En el contexto del Lenguaje Unificado de Modelado (UML), un diagrama de implementación se encuentra dentro de la familia de diagramas estructurales porque describe un aspecto del sistema en sí. En este caso, el diagrama de implementación describe la implementación física de la información generada por el programa de software en los componentes de hardware. A la información que el software genera se la conoce como artefacto.

Elementos del diagrama:

- **Artefacto:** Un producto desarrollado por el software, representado por un rectángulo con el nombre y la palabra "artefacto" encerrado por flechas dobles.
- **Asociación:** Una línea que indica un mensaje u otro tipo de comunicación entre nodos.
- **Componente:** Un rectángulo con dos pestañas que indica un elemento de software.
- **Dependencia:** Una línea discontinua que termina en una flecha, que indica que un nodo o componente depende de otro.
- **Interfaz:** Un círculo que indica una relación contractual. Aquellos objetos que se dan cuenta de que la interfaz debe completar cierto tipo de obligación.
- **Nodo:** Un objeto de hardware o software, mostrado por un cuadro tridimensional.
- **Nodo como contenedor:** Un nodo que contiene otro nodo dentro de sí, como en el ejemplo siguiente, en el que los nodos contienen componentes.
- **Estereotipo:** Un dispositivo contenido dentro del nodo, presentado en la parte superior del nodo, con el nombre entre flechas dobles a manera de corchetes.

Ejemplo de diagrama de implementación

Este ejemplo muestra un diagrama de implementación básico para Lucidchart. Hay un servidor web, un servidor de base de datos y la máquina donde el usuario visualiza el sitio web. Puedes agregar más complejidad mostrando las diferentes partes del servidor web y la forma en que funciona Javascript en el cliente usuario, pero este ejemplo te da una idea de cómo se ve una implementación en una notación UML.



REFERENCIAS

Siriwardhana, S. (2020, octubre 26). *Tutorial de Diagrama de Despliegue*. Creately Blog; Creately.

<https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>

¿Qué es la arquitectura de software? (2018, marzo 22). <https://platzi.com>; Platzi.

<https://platzi.com/blog/que-es-arquitectura-de-software/>

(S/f). Amazon.com. Recuperado el 8 de mayo de 2025, de

<https://aws.amazon.com/es/what-is/it-infrastructure/#:~:text=La%20infraestructura%20de%20TI%20es%20el%20conjunto%20de%20software%2C%20hardware,la%20eficiencia%20general%20del%20sistema.>

Tutorial de diagramas de despliegue. (2025, enero 2). Lucidchart.

<https://www.lucidchart.com/pages/es/tutorial-de-diagramas-de-despliegue>