

Infraestructura I

Profesor: David Pignalberi

Módulo 1: Inmersión

Sesión 1: Introducción a la materia y The Big Picture **Marzo 2 de 2022**

¿Qué es Infraestructura de IT?

La infraestructura de tecnología de la información, o infraestructura de IT, se refiere a los componentes combinados necesarios para el funcionamiento y la gestión de los servicios de IT de la empresa y los entornos de IT.

Arquitectura Cliente-Servidor

La arquitectura cliente-servidor persigue el objetivo de procesar la información de un modo distribuido. Además de la transparencia y la independencia del hardware y del software, una implementación cliente-servidor debe tener las siguientes características:

- Utilizar protocolos asimétricos
- El acceso es transparente, multiplataforma y multiarquitectura.
- Se facilitará la escalabilidad, de manera que sea fácil añadir nuevos clientes a la infraestructura —escalabilidad horizontal— o aumentar la potencia del servidor o servidores, aumentando su número o su capacidad de cálculo —escalabilidad vertical—.

Servidor

Genéricamente un servidor es un ordenador, pero con prestaciones elevadas. Sin embargo, desde este enfoque, un servidor es un proceso que ofrece recursos y servicios a los clientes que lo solicitan (back end). Debido a que los programas y datos se encuentran centralizados, se facilita la integridad y el mantenimiento.

Cliente

De manera genérica, nos referimos a una computadora, normalmente con prestaciones ajustadas, sin embargo, en entornos cliente-servidor, se utiliza el término front end, ya que es un proceso que solicita los servicios del servidor a través de una petición del usuario. Un proceso cliente se encarga de interactuar con el usuario, por lo que estará construido con alguna herramienta que permita implementar interfaces gráficas (GUI).

Middleware

Es la parte del software del sistema que se encarga del transporte de los mensajes entre el cliente y el servidor y facilita la interconexión de sistemas

heterogéneos sin utilizar tecnologías propietarias. Por lo cual, se ejecuta en ambos lados de la estructura.

El middleware permite independizar a los clientes y a los servidores. Además, ofrece más control sobre el negocio, debido a que permite obtener información desde diferentes orígenes —uniendo tecnologías y arquitecturas distintas— y ofrecer de manera conjunta.

Otra característica es que los sistemas están débilmente acoplados ya que interactúan mediante el envío de mensajes.

Ejemplos de Middleware:

DBMS: Sistema de Administración de Base de Datos, que permite crear, recuperar, actualizar y administrar datos.

API: Interfaz de Programación de Aplicaciones. Brinda un conjunto de subrutinas, funciones y procedimientos para ser utilizados en otros software.

Centros de Datos (Datacenters)

Inmuebles construidos específicamente para albergar sistemas de información y comunicación, debidamente protegidos contra incendios, accesos indebidos e interrupción de energía.

Sistema Operativo

Es un conjunto de programas capaces de administrar recursos físicos del servidor, así como los protocolos de ejecución de otros software.

Características Fundamentales de un Sistema Operativo

- Soporte de Red
- Amplia compatibilidad con el hardware
- Seguridad
- Tolerancia a Fallos

Sistemas Operativos de Servidores:

- Manejo diferente del hardware
- Características soportadas
- Soporte

Servidor UNIX (75% del mercado):

- GNU Linux
- FreeBSD
- macOSServer

Servidor Windows (25% del mercado):

- Windows Server (2003 – 2019)

Sincrónico

Escalabilidad horizontal: Ej. agregar otro servidor

Escalabilidad vertical: Agregar mejores características del servidor que está ya en operación

Ej: Si se tienen muchos requests, probablemente se requiere agregar servidores.

Ej: Si la aplicación requiere almacenar más datos, se podría aumentar la capacidad verticalmente.

También depende de factores económicos.

El middleware no es un elemento físico.

¿Reverse Proxy es un middleware? **Se podría considerar**

Serverless: No es necesario hacerse cargo del servidor; solo de que la aplicación funcione. En AWS, lambda. => Capacidad de computo sin tener que provisionar ni administrar los servidores.

Roadmap:

<https://roadmap.sh/devops>

Canal de Discord:

<https://discord.gg/2b26m3yw>

Comando:

?rank Canada 1

Instalar:

- VirtualBox
- Vagrant

Ejercicio 1

De forma grupal proponemos configuraciones de Sistemas Operativos y Software para los siguientes escenarios, una vez consensuado con nuestra mesa, subimos nuestras propuestas a un Jamboard.:

Escenario 1: Una página web realizada en PHP, con una base de datos en MySQL. Presupuesto escaso.

Escenario 2: Una página web para uso interno de una empresa realizada en .NET, integración necesaria con Active Directory.

Escenario 3: Un conjunto de APIs Web, que deben presentar un rendimiento excelente. Realizadas en Java, se posee amplia experiencia en Linux y se cuenta con un presupuesto holgado.

Escenario 4: Debemos soportar una base de datos relacional estable y probada, no tenemos disponibilidad para pagar ningún tipo de licencias.

Escenario 5: Debemos soportar una base de datos no relacional y un servidor Web eficiente, se posee experiencia en Linux. Mauricio.

Ejercicio 2

De forma grupal representamos en forma de diagrama por lo menos 2 ejemplos de la arquitectura Cliente – Servidor.

<https://jamboard.google.com/d/1MuoNGRVWw6YcF9i4EC6j90Nos3tgspL9WUUt-k9B5J0/viewer?f=5>