Trabajos de Arquitectura Jhovani Pineda Alvarez

¿Qué es un Servidor FTP?

es un software que corre en los servidores conectados a Internet (o bien a una red local LAN), y tiene como función primordial permitir que diferentes clientes FTP puedan conectarse a él, para luego descargar o subir información (archivos).

En una conexión FTP siempre interviene un servidor FTP, pues sin éste nunca podría accederse a los archivos para interactuar con ellos (descarga, subir o eliminación de los mismos). Un elemento necesario para que un servidor FTP funcione es estar bajo una red TCP / IP, la cual estará conectada al servidor VPS, Cloud o Dedicado.

¿Qué es nodeJs y en que lo puedo aplicar?

Se basa en el lenguaje JavaScript: sencillo, versátil y ampliamente conocido. NodeJS complementa a JS con librerías de sistema que sacan un gran rendimiento a los ejecutables.

El uso más inmediato de NodeJS está en la **programación de servicios web.** Tanto en la creación de APIs como sirviendo páginas tradicionales. Esto le ha valido que se le conozca como el JavaScript del lado del servidor.

**Netflix,uber,ebay,paypal,linkedln.**

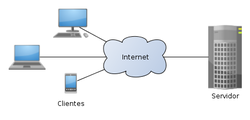
¿Que es API?

Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos

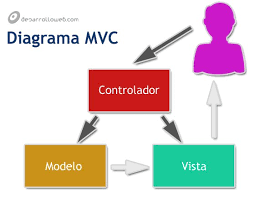
¿Qué es Arquitectura Servidor-Cliente?

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.



¿Qué es MVC?

es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.



¿Qué es Serverless?

En general, el términoserverless se emplea para referirse al modelo de computación según el cual el proveedor de la capa de computación nos **permite ejecutar durante un periodo de tiempo determinado porciones de código denominadas "funciones"** sin necesidad de hacernos cargo de la infraestructura subyacente que se provisiona para dar el servicio. En este modelo, el proveedor se encarga de ofrecer los recursos de forma transparente, de escalarlos automáticamente si crece la demanda y de liberarlos cuando no son utilizados, definiendo una serie de restricciones referentes al procesamiento y un modelo de pago por el consumo de los recursos derivados de la ejecución.

## Software-as-a-Service (SaaS)

## Básicamente se trata de cualquier servicio basado en la web. Tenemos ejemplos claros como el Webmail de Gmail, los CRM online. En este tipo de servicios nosotros accedemos normalmente a través del navegador sin atender al software. Todo el desarrollo, mantenimiento, actualizaciones, copias de seguridad es responsabilidad del proveedor.

Platform-as-a-Service (PaaS)

Es un modelo que reduce bastante la complejidad a la hora de desplegar y mantener aplicaciones ya que las soluciones PaaS gestionan automáticamente la escalabilidad usando más recursos si fuera necesario. Los desarrolladores aun así tienen que preocuparse de que sus aplicaciones estén lo mejor optimizadas posibles para consumir menos recursos posible.

Ejemplos populares son Google App Engine que permite desarrollar aplicaciones en Java o Python desplegándolas en la infraestructura que provee Google, cosa que también hace Heroku con Rails y Django.

## Infraestructure-as-a-Service (IaaS)

## En este caso con IaaS tendremos mucho más control que con PaaS, aunque a cambio de eso tendremos que encargarnos de la gestión de infraestructura.

## El ejemplo perfecto es el proporcionado por Amazon Web Service (AWS) que no provee una serie de servicios como EC2 que nos permite manejar maquinas virtuales en la nube o S3 para usar como almacenamiento. Nosotros podemos elegir qué tipo de instancias queremos usar LInux o Windows, así como la capacidad de memoria o procesador de cada una de nuestras maquinas. El hardware para nosotros es transparente, todo lo que manejamos es de forma virtual.