



UNIVERSIDAD POLITÉCNICA
DE TECÁMAC

Proyecto – Parcial 1

Materia: Programación
cliente/servidor.

Profesor: Emanuel Torres
Servín.

Alumno:

- Erick Alejandro López Pacheco.
- Christian Uriel Ramírez Pérez.
- Oscar Luis Flores Vázquez.
- Rafael Alcantar Chávez.

Grupo: 3523IS

Carrera: Ingeniería en
Software

Contenido

Descripción del problema.	3
Reporte sobre la comunicación de dispositivos de red y las arquitecturas cliente servidor.....	3
Diagrama de componentes de la arquitectura Cliente/Servidor	4
Cuadro comparativo entre los modelos IAAS, PAAS, SAAS y Cliente/Servidor	5
Propuesta técnica de arquitectura Cliente/Servidor contemplando los modelos de cómputo en la nube.....	6
Justificación de selección de sistema operativo móvil.....	7
Justificación de lenguaje de programación móvil	8
Conclusión.	8

Descripción del problema.

Juan tiene una empresa de patitos de hule, dicha empresa tiene 3 años y ya cuenta con 40 empleados, actualmente se encuentra en crecimiento constante, sin embargo el presupuesto sigue siendo limitado para cuestiones de inversión y mejora, por lo que ha perdido control de sus empleados en ciertas áreas de la empresa, por lo que Juan toma la decisión de contratar unos programadores para realizar una aplicación en la cual pueda administrar de forma general su empresa, sin embargo no tiene conocimiento alguno sobre el mundo del desarrollo de software.

Reporte sobre la comunicación de dispositivos de red y las arquitecturas cliente servidor.

Una red es la combinación de dos o más sistemas y los enlaces de conexión de estos. Una red física es el hardware (equipo como adaptadores, cables y líneas de teléfono) que compone la red. El software y el modelo conceptual componen la red lógica. Existen distintos tipos de redes y emuladores que proporcionan funciones diferentes.

La complejidad de las redes de sistemas modernas ha dado origen a varios modelos conceptuales para explicar cómo funcionan las redes. Entre estos modelos, uno de los más comunes es el Modelo de referencia OSI (Open Systems Interconnection - Interconexión de sistemas abiertos) de la International Standards Organization (Organización internacional para los estándares), que también se conoce como modelo de siete capas OSI.

- **Redes físicas**

La red física consta de los cables (cable coaxial, par trenzado, fibra óptica y líneas telefónicas) que conectan el distinto hardware que reside en la red, el adaptador utilizado en los sistemas conectados a la red (sistemas principales) y los concentradores, repetidores, direccionadores o puentes utilizados en la red.

- **Sistemas de red**

Todas las comunicaciones de red implican el uso de hardware y software.

El soporte de comunicaciones de red lo determinan el hardware y el software necesarios para ejecutar dicho hardware y para intercambiar información con la red.

La arquitectura cliente-servidor es un modelo de diseño de software. Aquí, las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

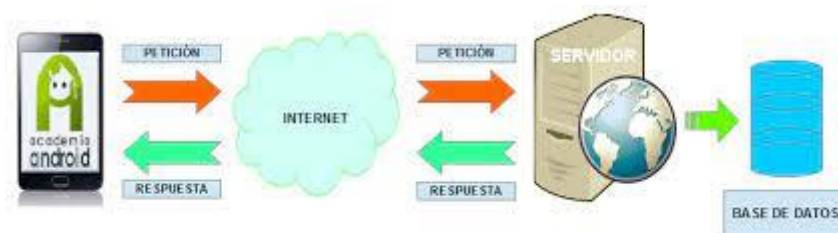
Un ejemplo de cliente es Chrome, que es ejecutado en nuestra computadora. En cambio, el servidor es aquel lugar al que pide datos para mostrar al usuario.

En la base de datos los datos están almacenados de forma remota y persistente (no se van a perder porque salgamos del sitio, ya que no están en nuestra máquina).

Mientras que del lado cliente, el lenguaje por defecto es JS. A su vez, del lado del servidor existen diversos lenguajes en que se puede programar: PHP, Java, Ruby, Python, y también pueden crearse servidores con JavaScript, usando Node.

X|

Diagrama de componentes de la arquitectura Cliente/Servidor



Cuadro comparativo entre los modelos IAAS, PAAS, SAAS y Cliente/Servidor

	IAAS	PAAS	SAAS	CLIENTE SERVIDOR
DESCRIPCION	este proporcionará un infraestructura hardware virtualización en la red para aplicación y Atender	Esto es un plataforma llegar desarrollar A ejecutar y administrar diferente solicitud	Esto es un software como un Servicios para que puede accesible para él La Internet	todo es lo mismo Es un solicitud cual va a ejecutar servidor y cliente Dónde gracias a esto será capaz de Ingresar todo es lo mismo aprobar La Internet
Ejemplos	Amazon, Microsoft y google	Google App Engine, Microsoft Azure	Salesforce Google Apps, Dropbox	Aplicaciones dentro del escritorio o aplicaciones web
Características	<ul style="list-style-type: none"> - Los usuarios tienen control total sobre la infraestructura. - Arquitectura multiinquilino que sirve a múltiples usuarios - Las tarifas varían según el consumo. - Infraestructura de nivel empresarial - Recursos altamente escalables 	<ul style="list-style-type: none"> - Los equipos remotos pueden trabajar juntos fácilmente - Las bases de datos y los servicios web están integrados - El usuario no tiene un control total sobre la infraestructura - Varios usuarios pueden 	<ul style="list-style-type: none"> El proveedor es responsable de todo - Las aplicaciones son accesibles desde cualquier dispositivo vinculado 	<ul style="list-style-type: none"> - Facilita la integración entre diferentes sistemas y comparte información - Al favorecer el uso del interfaz de gráficas interactivas, los sistemas contruidos bajo este esquema tienen una mayor

		acceder a la misma aplicación en desarrollo		interacción con el usuario.
Desventajas	<ul style="list-style-type: none"> -Se requiere formación del equipo para saber cómo manejar la nueva infraestructura -Las interrupciones del proveedor hacen que los usuarios no puedan acceder a sus datos -Problemas de seguridad de los datos a la arquitectura multitenant. 	<ul style="list-style-type: none"> -Dependencia del soporte, la fiabilidad y la rapidez del proveedor Compatibilidad de la infraestructura existente -Problemas de seguridad de los datos 	<ul style="list-style-type: none"> -Se necesita conectividad -Variedad limitada de soluciones -Pérdida de control 	<ul style="list-style-type: none"> -Este modelo representa una limitación importante en cuanto a los costos económicos debido a que estos servidores son computadoras de alto nivel con un hardware y software específicos para poder dar un correcto funcionamiento a nuestras apps

Propuesta técnica de arquitectura Cliente/Servidor contemplando los modelos de cómputo en la nube

Se propone utilizar PaaS (Platform as a Service) ya que proporciona un entorno de desarrollo listo para usar en el que los desarrolladores pueden centrarse en escribir y ejecutar código de calidad para crear aplicaciones personalizadas.

La Plataforma como Servicio se distribuye a través de la web y permite que los desarrolladores puedan construir aplicaciones escalables y altamente disponibles, sin preocuparse del sistema operativo, el almacenamiento o las actualizaciones. Proporciona un espacio de trabajo que los desarrolladores pueden usar para desarrollar, gestionar, distribuir y testear sus aplicaciones de software.

Este modelo de servicio cloud hace que el proceso de desarrollo de aplicaciones sea mucho más simple, eficiente y competitivo. Varios usuarios pueden acceder a una plataforma PaaS a través de la misma aplicación de desarrollo. Esta plataforma integra servicios web, motores de bases de datos, etc., para ayudar a los desarrolladores con el desarrollo, testeo y despliegue de apps.

Los servicios de plataforma en la nube permiten innovar más rápido, pero los clientes de PaaS solo tienen control sobre lo que construyen en la plataforma. Así que, si hay algún problema con el sistema operativo o el hardware de la plataforma, no tienen ningún control sobre el impacto que pueda tener en el rendimiento de su software. Sin embargo, mantienen el control sobre otros aspectos como el anti-malware o el control de acceso.

Justificación de selección de sistema operativo móvil.

Se eligió el sistema operativo Android, ya que, al estar presente en la mayoría de las marcas comerciales, indudablemente es el sistema más utilizado, esto ayudara a ser compatible con cualquier celular que el dueño decida usar (siempre y cuando tenga el SO Android), dicho sistema es de código abierto esto permite que cualquiera cree aplicaciones para el

sistema, con lo cual aumenta la oferta en el universo Android y los posibles errores pueden ser revisados y solucionados con mayor rapidez.

Justificación de lenguaje de programación móvil

Se pretende utilizar JavaScript ya que es rápido, versátil, sencillo y destaca por su funcionalidad. Además, se ocupa de mejorar la experiencia de usuario, gracias a su interactividad.

Conclusión.

La aplicación será viable ya que solamente busca llevar un control sobre sus empleados, y no se necesitan de muchos requerimientos para que esta funcione, como el cliente no tenía mucho presupuesto se optó por las opciones ya descritas.