UNIVERSIDAD POLITÉCNICA DE TECÁMAC

UNIVERSIDAD POLITÉCNICA DE TECÁMAC.

Materia: Programación cliente/servidor.

Docente: Emmanuel Torres Servín.

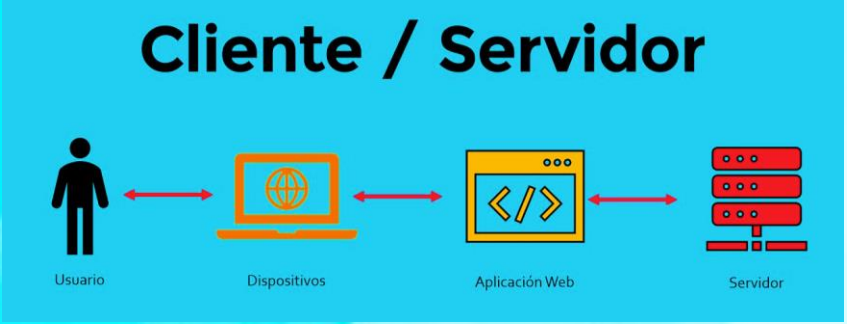
Trabajo: **Rúbrica Unidad 1.**

Team:

- Ángel Velasco Marco Joel.
- Fuentes Cortes Luz Alexia.
- Alemán Perez Natali Joselin.

Grupo: 1523IS.

Fecha de entrega: 08 de Febrero del 2023.



Índice:

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.

Justificación de selección de sistema operativo móvil:

Hay varias razones por las que Android puede ser una buena elección como sistema operativo:

Popularidad: Android es el sistema operativo más popular en el mundo, con una cuota de mercado del 68,8% en diciembre de 2021. Esto significa que hay una amplia variedad de dispositivos móviles y tabletas que lo utilizan, lo que lo hace accesible para una gran cantidad de usuarios.

Personalización: Android permite una gran cantidad de personalización a través de la instalación de aplicaciones, widgets y temas, lo que lo hace atractivo para usuarios que desean tener control sobre su experiencia móvil.

Integración con Google: Android es un producto de Google, lo que significa que está estrechamente integrado con los servicios de Google, como Google Maps, Gmail y Google Drive.

Precio: Muchos dispositivos Android son más asequibles que los dispositivos de otros sistemas operativos, lo que lo hace atractivo para aquellos que buscan un buen valor por su dinero.

Android es un sistema operativo popular, personalizable, integrado con los servicios de Google y con una amplia variedad de aplicaciones disponibles. Además, su bajo costo puede ser un factor atractivo para aquellos que buscan un dispositivo asequible.

Justificación de lenguaje de programación móvil:

El lenguaje que se eligió fue con base a los conocimientos que tiene el programador y el que más conviene para el desarrollo del sistema, ya que dependiendo de esto es como se puede avanzar más rápido o retrasarse, pero siempre se verá por el lado de poder sacar adelante el sistema con el menor tiempo posible para lograr corregir los errores que se encuentren, por eso debemos de tener en cuenta todos los datos que se nos proporcionan.

En este caso nosotros ocuparemos el lenguaje de C#, procederemos a hacer una comparación entre otros lenguajes y el porqué del que lo elegimos.

Rapidez

C#: Suele ser más rápido a comparación de java, ya que su tiempo de respuesta es más corto y esto es aceptable para este proyecto.

Java: En cambio este es un tipo de lenguaje intermedio y para funcionar necesita un JIT (Just inTime Compiler) que traduzca a código de maquina al momento de ejecutar.

C++: Es más rápido que estos lenguajes, puesto que este genera código más nativo.

Facilidad de lectura de código

C#: Tiene una lectura fácil.

Java: Tiene una lectura fácil.

C++: Es más difícil de leer y entender que los demás lenguajes.

Disponibilidad de herramientas de desarrollo

C#: Cuenta con muchas herramientas útiles que podemos usar para este proyecto.

Java: Cuente igual con buenas herramientas para el desarrollo de proyectos.

C++: Las herramientas de las cuales dispone no son tan variadas.

Multiplataforma

C#: Es multiplataforma orientada a objetos y de código abierto.

Java: Es multiplataforma orientada a objetos y de código abierto

C++: Este no cuenta con esa función.

Conexión a la Base de Datos

C#: Se puede realizar, pero debemos de instalar los componentes requeridos para poder conectarlo a la base de datos.

Java: Se puede realizar, pero se deben de conseguir el driver necesario para poder hacer la conexión a la base de datos.

C++: Dispone de librerías de terceros para poder realizar la conexión a la base de datos.

Entorno de desarrollo grafico

C#: Cuenta con un buen desarrollo gracias a .NET

Java: Cuenta con un buen desarrollo y simplifica el trabajo (NetBeans y Eclipse).

C++: Como tal no cuenta con un desarrollo gráfico y necesita de librerías.

Reporte de comunicación de dispositivos de red y arquitecturas de cliente servidor:

La comunicación de dispositivos de red es el intercambio de datos entre dispositivos conectados a una red. Esto permite que los dispositivos compartan recursos y realicen tareas en conjunto.

Algunos de los dispositivos de red más comunes y sus diferentes tareas que realizan son los siguientes:

SWITCH

Conmutador es un equipo de capa de enlace de datos, multipuerto que busca mejorar la eficiencia de la red gracias a la gran capacidad del circuito virtual. También se encarga de la conexión de segmentos de red y del mantenimiento de una información de enrutamiento limitada sobre los nodos de la red interna. El encaminamiento se realiza de acuerdo con las tablas de direcciones MAC que le permite conocer en qué puerto físico se ubica cada dirección.

ROUTER

Un router es un dispositivo de interconexión de redes que se encarga establecer cuál es la mejor ruta para transmitir paquetes de datos a su lugar de destino, utilizando un enrutado de esos datos en función de la dirección IP para enviar la información a una interfaz u otra.

Ademas, un router puede almacenar información acerca de la red donde se encuentren conectados y hacen parte del nivel de red del Modelo de Interconexión de Sistemas Abiertos.

GATEWAY

Los gateway o puertas de enlace son equipos electrónicos que permiten la conexión entre dos redes, normalmente, entre una red privada y una red pública. Estas puertas de enlace se ubican entre las capas de transporte y sesión del modelo OSI.

FIREWALL

Un cortafuegos o firewall es un dispositivo de seguridad que permite realizar el filtrado de paquetes de acuerdo a ciertas reglas. Además, permite la segmentación de la red para crear zonas con mayor nivel de seguridad. Esto ayuda para que no haya desvío de información en las empresas.

IDS e IPS

Un IDS es un sistema de detección de intrusiones en función de firmas, que solo detecta y notifica posibles actividades maliciosas; mientras que un IPS es un sistema de prevención de intrusiones, similar a IDS, pero que puede llegar a bloquear tráfico.

Proxy

Es un dispositivo intermedio para el filtrado de conexiones de usuario a nivel de aplicación, es decir, se encarga de facilitar las peticiones de recursos realizadas por un cliente a otro servidor.

Las arquitecturas de cliente-servidor son un modelo común de comunicación en redes de computadoras. En este modelo, los dispositivos cliente envían solicitudes a un dispositivo servidor, el cual procesa la solicitud y envía una respuesta.

La arquitectura Cliente/Servidor centraliza la información y la separación de responsabilidades. El servidor será la única entidad que tendrá acceso a los datos y los servirá solo a los clientes del cual el confía, y de esta forma, protegemos la información y la lógica detrás del procesamiento de los datos, además, el servidor puede atender simultáneamente a varios clientes, por lo que suele ser instalado en un equipo con muchos recursos. Por otro lado, el cliente suele ser instalado en computadoras con bajos recursos, pues desde allí no se procesa nada, simplemente actúa como un visor de los datos y delega las operaciones pesadas al servidor.

El server recibe, procesa y almacena todos los datos provenientes de todos los clientes. Si bien los clientes por lo general solo se conectan a un solo servidor, existen variantes donde hay clientes que se conectan a múltiples servidores para funcionar, tal es el caso de los navegadores.

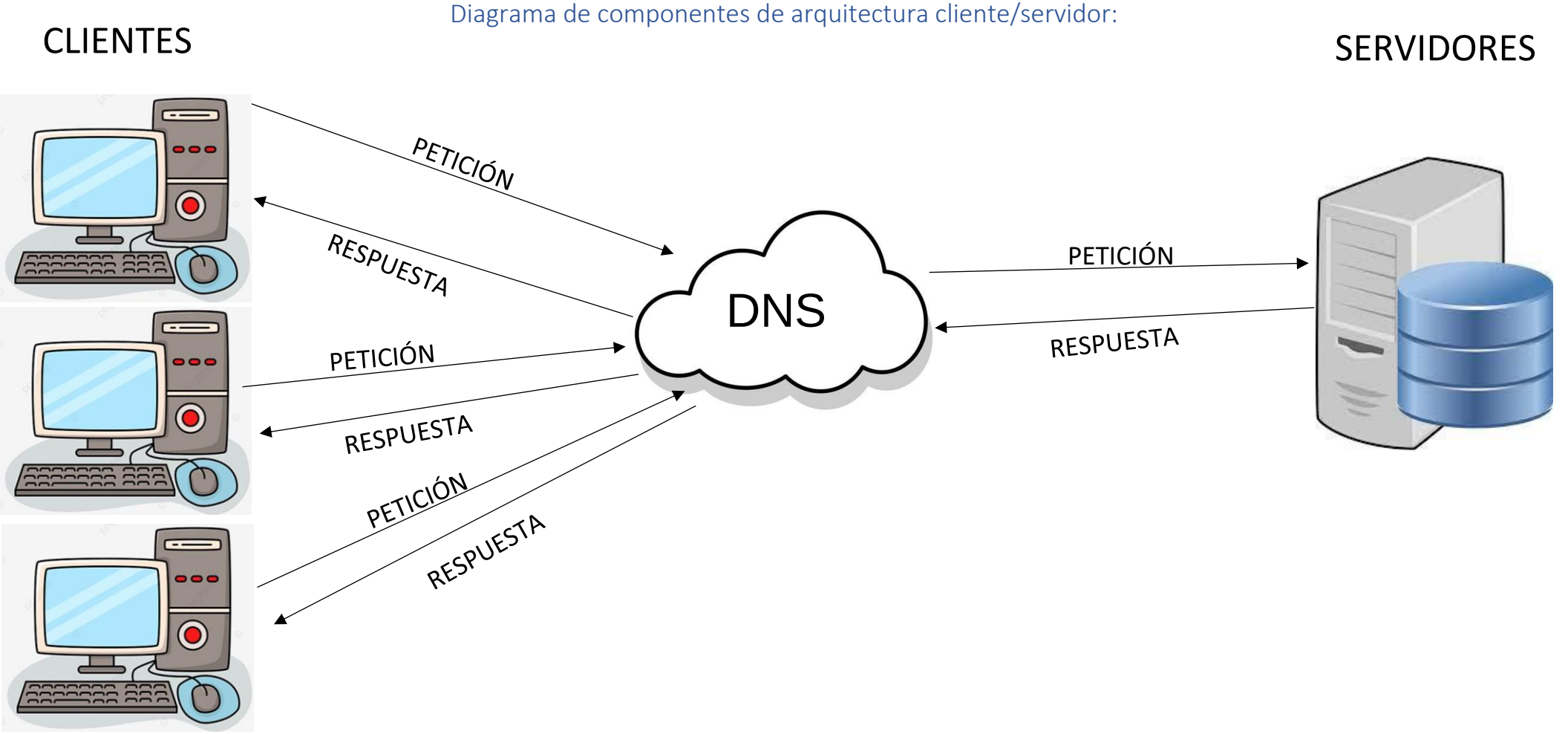
La comunicación en las arquitecturas de cliente-servidor se realiza mediante el uso de protocolos de red estándar, como TCP/IP. Los dispositivos cliente y servidor pueden ser cualquier tipo de dispositivo conectado a una red, como computadoras, smartphones, impresoras, etc.

Son ampliamente utilizadas en aplicaciones como el correo electrónico, la transferencia de archivos, los juegos en línea, las bases de datos, entre otros.

En resumen, la comunicación de dispositivos de red y las arquitecturas de cliente-servidor son fundamentales para el funcionamiento eficiente de las redes de

computadoras y para permitir la interacción y el intercambio de información entre los dispositivos conectados.

Diagrama de componentes de arquitectura cliente/servidor:



Cuadro comparativo entre los modelos IAAS, PAAS, SAAS y cliente/servidor:

Características	IAAS	PAAS	SAAS	Cliente servidor.
Definición.	La infraestructura como servicio (IAAS) es un modelo de servicio en la nube que ofrece recursos de infraestructura bajo demanda, como computación, almacenamiento, redes y virtualización, a empresas y particulares a través de la nube.	El modelo PAAS ofrece una plataforma basada en la nube para gestionar, ejecutar y desarrollar aplicaciones. El proveedor de la nube mantiene, gestiona y aloja todo el software y el hardware incluido en el modelo. Normalmente, el proveedor gestionará las herramientas de desarrollo, los frameworks, los tiempos de ejecución, el middleware, las bases de datos, la red, el almacenamiento, el sistema operativo y los servidores, entre otros. Esto incluye la respuesta, el despliegue, la entrega, las pruebas,	El SAAS es un software de aplicación listo para usar y alojado en la nube. Los usuarios pagan una cuota anual o mensual para utilizar una aplicación desde una aplicación móvil, escritorio o navegador web. La aplicación es gestionada y alojada por el proveedor de SaaS. También gestiona toda la infraestructura necesaria para ofrecer la aplicación, como el almacenamiento de datos, el software de la	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.

		la integración y la codificación.	aplicación, el middleware, los servidores, etc.	
Ventajas.	• La empresa ya no necesita adquirir hardware para implementar infraestructura informática. Permite optimizar los gastos corrientes. • Es posible implementar nuevos proyectos de forma rápida. • Es flexible y permite escalar los recursos fácilmente. • Desaparece el gasto por instalación, mantenimiento y	• Permite una fácil migración a la nube híbrida. • Menor cantidad de codificación necesaria. • Política de empresa automatizada. • Se reducen los costes de lanzamiento, prueba y creación de aplicaciones. • El proceso de desarrollo es sencillo y rápido. • El software construido en PAAS es multitenant, altamente	• Fácil de personalizar. • Escalabilidad. • Modelo de pago por uso. • Accesible desde cualquier lugar. • Compatibilidad entre dispositivos. • Sin gastos de hardware.	• Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P) • Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores). • Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre

	modernización del hardware. • Diversas sedes y trabajadores pueden conectarse entre sí y colaborar fácilmente gracias a IaaS.	disponible y escalable.		varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
Desventajas.	• El proveedor es el responsable de la disponibilidad y seguridad de los servicios. • El acceso online es fundamental, los problemas de conexión pueden interrumpir el uso de IaaS. • La localización de datos es crucial para	• Dependencia del soporte, la fiabilidad y la rapidez del proveedor. • Compatibilidad de la infraestructura existente. • Problemas de seguridad de los datos.	• Se necesita conectividad. • Variedad limitada de soluciones. • Pérdida de control.	La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envía peticiones simultáneas al mismo servidor, este debe gestionarlas y por ello puede verse saturado (a mayor número de clientes, más problemas para el servidor). Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene. • El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está caído, las

	evitar la pérdida de datos y lograr un óptimo nivel de seguridad.			peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red. • El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el coste. • El cliente no dispone de los recursos que puedan existir en el servidor. Por ejemplo,

				si es una aplicación web, no podremos escribir en el disco duro del cliente o imprimir directamente sobre las impresoras sin sacar antes la ventana previa de impresión de los navegadores.

Propuesta técnica de arquitectura cliente/servidor contemplando los modelos de cómputo en la nube:

Diseño de la arquitectura: La arquitectura se basará en un modelo de nube híbrida, donde se utilizarán servidores en la nube y en el centro de datos para alojar servicios.

Servidores en la nube: Se utilizarán servicios de nube pública como Amazon Web Services (AWS) o Microsoft Azure para alojar aplicaciones y servicios críticos para la empresa. Estos servidores proporcionarán escalabilidad y disponibilidad a demanda.

Servidores en el centro de datos: Para alojar aplicaciones y servicios sensibles a los datos, se utilizarán servidores en el centro de datos para asegurar la seguridad de los datos.

Conexión de red: Se implementará una conexión de red privada virtual (VPN) para permitir la comunicación segura entre los servidores en la nube y en el centro de datos.

Balanceo de carga: Se implementará un equilibrador de carga para distribuir la carga entre los servidores y garantizar la disponibilidad y la escalabilidad de la aplicación.

Copias de seguridad y recuperación ante desastres: Se llevarán a cabo copias de seguridad regulares de los datos en ambos servidores en la nube y en el centro de datos y se implementará un plan de recuperación ante desastres para garantizar la continuidad del negocio en caso de una interrupción.

Monitoring y gestión: Se implementará un sistema de monitoreo y gestión para supervisar el funcionamiento de los servidores y aplicaciones y para recibir alertas en caso de problemas.

Esta propuesta combinará la escalabilidad y disponibilidad de los servidores en la nube con la seguridad y control de los servidores en el centro de datos, para ofrecer una solución robusta y escalable para las necesidades de la empresa en la nube.

Conclusión:

El documento se realizó estructuradamente cada punto que se nos pidió para entregar una documentación en buen estado, el cliente podrá administrar su empresa de forma adecuada para no perder el control de ninguna área.

También se le proporcionará el documentó para que pueda entender el tipo de trabajo que se le esta entregando y realizando y no tenga dudas de este.

Los dispositivos de red juegan un papel crucial en la implementación de redes informáticas. Desde routers hasta switches y firewalls, cada uno de ellos tiene una función específica en el procesamiento y transmisión de datos.

En cuanto al modelo cliente-servidor, es una estructura de red en la que uno o más dispositivos actúan como servidores y otros como clientes. El servidor es responsable de proporcionar recursos y servicios a los clientes, mientras que los clientes solicitan y utilizan esos recursos. Este modelo es ampliamente utilizado en muchos sistemas informáticos, incluyendo bases de datos, correo electrónico, servicios web y muchos otros.

En conclusión, los dispositivos de red y el modelo cliente-servidor son componentes clave de las redes informáticas modernas y trabajan juntos para permitir la eficiente transmisión y acceso a los datos.