

Universidad Politécnica de Tecámac

Programación Cliente Servidor

Profesor: Torres Servín Emmanuel

Alumnos:

Márquez Rodríguez Jocelyn

1321124322

Montes de Oca Rodríguez Luis Alberto

1320114127

Santana Cortes Jose Luis

1321124292

Vega Ojeda Álvaro Joshua

1321124271

3523IS

índice

Descripción del problema	2
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 y 6
Propuesta técnica de arquitectura Cliente/Servidor contemplando los modelos de cómputo en la nube.....	7 y 8
JUSTIFICACIÓN DEL SISTEMA OPERATIVO MOVIL (ANDROID).....	9
JUSTIFICACIÓN DEL LENGUAJE DE PROGRAMACION ELEGIDO (KOTLIN).....	10
Conclusión.....	11

Descripción del problema:

El siguiente problema se presenta de un caso de estudio el cual se habla de Juan quien tiene una empresa de patitos de hule, dicha empresa cumple con 3 años en servicio y cuenta con 40 empleados, actualmente se encuentra en crecimiento constante, sin embargo el presupuesto sigue siendo limitado para cuestiones de inversión y mejora de esta misma, por lo que se ha perdido el control de sus empleados en ciertas áreas que integra la empresa, Juan quien es el dueño toma la iniciativa de contratar programadores para realizar una aplicación que consta en el poder administrar de forma general su empresa, sin embargo no cuenta con el conocimiento alguno sobre el mundo del desarrollo de software.

El objetivo de este proyecto es poder brindarle una solución afectiva, en la cual quede satisfecho con el trabajo que busca, lo principal es ayudar a Juan a encontrar una de las mejores opciones para elegir el cómo realizar los siguientes elementos:

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

Comunicación de dispositivos de red:

La comunicación de dispositivos de red es una parte fundamental de la informática y permite a los dispositivos compartir información y recursos. Esto se logra mediante el uso de protocolos de comunicación, que son reglas establecidas para garantizar que los dispositivos puedan comunicarse de manera efectiva y eficiente.

Hay muchos protocolos de comunicación de red diferentes, incluidos TCP/IP, HTTP, FTP, DNS, entre otros. Cada protocolo tiene una función específica y juega un papel importante en la comunicación de dispositivos de red.

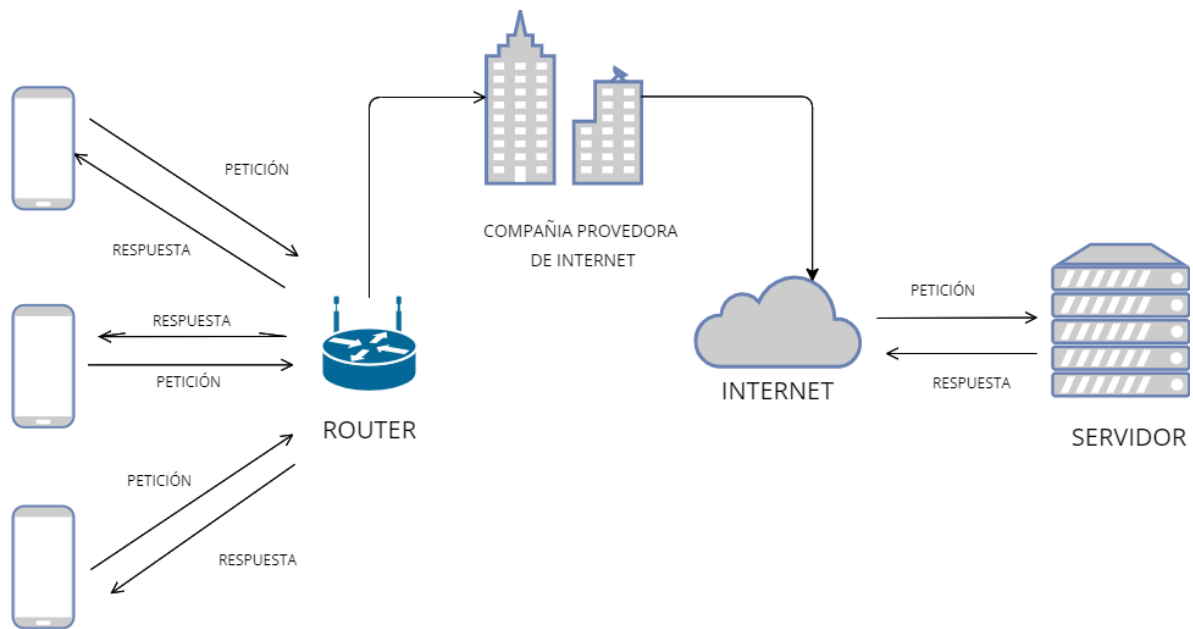
Arquitectura cliente-servidor:

La arquitectura cliente-servidor es un modelo de computación en el que un servidor proporciona servicios a uno o más clientes. En este modelo, el servidor es responsable de procesar los datos y brindar una respuesta a los clientes. Los clientes, por su parte, son responsables de realizar peticiones y presentar los resultados a los usuarios.

La arquitectura cliente-servidor es ampliamente utilizada en la industria de la tecnología de la información y se puede encontrar en muchas aplicaciones, como los servidores web, correo electrónico y bases de datos o aplicaciones especializadas.

Esto le puede ayudar a su empresa a tener más control sobre todas las áreas y los empleados facilitando la vista general del manejo y administración de la empresa dando una vista clara de las áreas de oportunidad así como también tener un registro o Historial de los movimientos que se hagan desde cualquier parte del mundo siempre y cuando tenga el cliente (algún dispositivo con conexión a internet) le brindara esa facilidad.

- **Diagrama de componentes de la arquitectura Cliente/Servidor.**



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

Modelo	Definición	Características	Ventajas	Desventajas
<i>IaaS</i>	Infraestructura como Servicio	Provee acceso a recursos de infraestructura como servidores, almacenamiento y redes. El cliente tiene control total sobre el software y el hardware	- Mayor flexibilidad y escalabilidad - Menor costo inicial - Control completo sobre el hardware y el software	- Mayor complejidad y responsabilidad - Requiere habilidades técnicas avanzadas
<i>PaaS</i>	Plataforma como Servicio	Provee una plataforma para desarrollar, ejecutar y alojar aplicaciones. El proveedor se encarga de la gestión de la infraestructura subyacente	- Menor complejidad y responsabilidad - Mayor productividad y eficiencia	- Menor flexibilidad y control - Costos más altos a largo plazo
<i>SaaS</i>	Software como Servicio	Provee acceso a un software a través de Internet sin la necesidad de descargar o instalarlo en un dispositivo local. El proveedor se encarga de la gestión y el mantenimiento del software y la	Provee acceso a un software a través de Internet sin la necesidad de descargar o instalarlo en un dispositivo local. El proveedor se encarga de la gestión y el mantenimiento del software y la	- Menor flexibilidad y control - Dependencia del proveedor de la nube

		infraestructura subyacente	infraestructura subyacente	
<i>Cliente-Servidor</i>	Arquitectura en la que un dispositivo (cliente) solicita servicios a otro dispositivo (servidor)	El servidor proporciona servicios y el cliente los consume. La arquitectura permite una separación clara de responsabilidades y una mejor escalabilidad y rendimiento	<ul style="list-style-type: none"> - Escalabilidad y flexibilidad - Mejor rendimiento y eficiencia - Mayor seguridad y control 	<ul style="list-style-type: none"> - Requiere una configuración y mantenimiento adecuados - Dependencia del servidor - Costos más altos a largo plazo

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

La propuesta técnica se centra en el desarrollo de una aplicación web que permita a Juan administrar de manera general su empresa de patitos de hule. La aplicación se desarrollará utilizando una arquitectura cliente-servidor y se implementará en la nube, lo que permitirá a Juan acceder a la aplicación desde cualquier lugar y en cualquier momento.

Arquitectura cliente-servidor:

La arquitectura cliente-servidor será utilizada para el desarrollo de la aplicación, lo que significa que la aplicación se dividirá en dos partes: el cliente y el servidor. El cliente será una aplicación web que se ejecutará en el navegador del usuario y permitirá a Juan interactuar con la aplicación. El servidor alojará la aplicación y proporcionará los servicios necesarios para que el cliente funcione correctamente.

Modelos de cómputo en la nube:

La aplicación se implementará en la nube utilizando uno de los modelos de cómputo en la nube más adecuados para las necesidades de Juan y su empresa. Algunos de los modelos de cómputo en la nube que se pueden considerar incluyen:

Infraestructura como servicio (IaaS): En este modelo, Juan alquilaría los recursos de hardware y software necesarios para ejecutar la aplicación.

Plataforma como servicio (PaaS): En este modelo, Juan alquilaría una plataforma de desarrollo en la nube en la que los programadores puedan desarrollar y ejecutar la aplicación.

Software como servicio (SaaS): En este modelo, Juan utilizaría una aplicación en la nube que ya está desarrollada y que se ejecuta en el servidor de un proveedor de servicios en la nube.

Funcionalidades:

Gestión de empleados: Juan podrá gestionar y asignar tareas a sus empleados, así como verificar el estado de sus tareas y su progreso.

Gestión de inventario: Juan podrá verificar el inventario de patitos de hule y realizar pedidos de más productos si es necesario.

Reportes y análisis: Juan podrá generar informes.

En conclusión, la implementación de una aplicación cliente-servidor en la nube sería una excelente opción para Juan, ya que le permitiría tener un mejor control de su empresa y brindaría una mayor eficiencia en la administración de sus empleados y recursos. Además, al estar basada en la nube, la aplicación estaría accesible en cualquier momento y desde cualquier lugar, lo que facilitaría el trabajo de los empleados y mejoraría la productividad de la empresa.

JUSTIFICACIÓN DEL SISTEMA OPERATIVO MOVIL (ANDROID)

Hay varias razones por las cuales se puede elegir Android como sistema operativo para administrar una empresa de forma general como lo son las siguientes:

Amplia disponibilidad: Android es un sistema operativo muy popular y está disponible en una amplia gama de dispositivos, desde teléfonos inteligentes hasta tabletas y computadoras. Esto significa que es fácil para los empleados acceder a la información y las aplicaciones que necesitan para realizar sus tareas.

Integración de Google: Como Android es propiedad de Google, es fácil integrar otros productos y servicios de Google, como Gmail, Google Drive y Google Calendar, en el sistema. Esto puede mejorar la eficiencia y la colaboración en la empresa.

Personalización y escalabilidad: Android es un sistema operativo altamente personalizable y escalable, lo que significa que puede adaptarse a las necesidades de la empresa a medida que cambian. Además, existen una amplia gama de aplicaciones disponibles en la Play Store que pueden ayudar a mejorar la productividad y la eficiencia de la empresa.

Seguridad: Aunque Android ha sido criticado en el pasado por su seguridad, Google ha tomado medidas para mejorar la seguridad en las últimas versiones del sistema operativo. Además, existen opciones de seguridad avanzadas disponibles para las empresas, como el cifrado de dispositivos y la gestión de aplicaciones.

Además, desarrollar aplicaciones móviles para Android es una excelente idea debido a varias razones:

Base de usuarios grande y en crecimiento: Android es el sistema operativo móvil más popular en todo el mundo, con una base de usuarios cada vez más grande. Esto significa que, al desarrollar una aplicación para Android, puede tener acceso a una audiencia global significativa.

Gran variedad de dispositivos: Android está disponible en una amplia gama de dispositivos, incluidos teléfonos inteligentes, tabletas y computadoras. Esto significa que una aplicación desarrollada para Android tendrá una gran cantidad de dispositivos compatibles, lo que puede aumentar su alcance y su potencial de ingresos.

Menor tiempo de desarrollo y menores costos: Debido a la amplia disponibilidad de herramientas de desarrollo y recursos en línea para Android, el tiempo de desarrollo y los costos asociados con el

desarrollo de una aplicación para Android pueden ser significativamente menores en comparación con otras plataformas móviles.

Libertad y flexibilidad: Al contrario de algunas otras plataformas móviles, Android ofrece una gran libertad y flexibilidad a los desarrolladores. Pueden utilizar cualquier lenguaje de programación y herramienta que prefieran para desarrollar su aplicación, lo que significa que pueden personalizar la experiencia de usuario y añadir funcionalidades únicas.

JUSTIFICACIÓN DEL LENGUAJE DE PROGRAMACION ELEGIDO (KOTLIN)

Nosotros elegimos kotlin como lenguaje de programación ya que es más fácil de utilizar y de manipular y aparte es más fácil decirle al programa que es lo que vamos a necesitar y también nos facilita la parte del desarrollo y además al realizarlo ya que su sistema es fácil de operar y esto nos ayuda a evaluar por que elegimos este lenguaje para el desarrollo de la página.

También ya investigando un poco los costos nos percatamos que es menos costoso utilizar el programa kotlin ya que tiene una facilidad de usar ya que investigando la aplicación tiene un uso muy practico y es muy sencillo de utilizar.

Kotlin se puede utilizar para cualquier tipo de desarrollo, desde web del lado del servidor y del lado del cliente hasta Android e iOS. Debido a que el lenguaje se ejecuta en la JVM, permite que el código se comparta entre diferentes plataformas.

También ya hemos enumerado los principales beneficios y características que ofrece Kotlin, pero el acalorado debate anterior sobre la "batalla" entre Kotlin y Java ha resurgido.

Al realizar una comparación entre Java y kotlin, Java tiene una larga historia en la comunidad más grande del mundo web. Sin embargo, el uso de Kotlin está aumentando día a día, especialmente para desarrollar aplicaciones móviles en Android.

Analizando hemos visto que programar en kotlin nos lleva menos tiempo a la hora de programar y además una de las ventajas de Kotlin es que elimina el código redundante y es compacto y conciso, lo que simplifica enormemente el proceso de codificación y evita la repetición.

Conclusión:

En conclusión, la decisión de Juan de contratar a programadores para crear una aplicación de gestión de su empresa es una buena iniciativa para mejorar el control y la eficiencia de su negocio. Al considerar los modelos de cómputo en la nube y el modelo cliente-servidor, podemos ver que hay varias opciones disponibles para satisfacer las necesidades de su empresa.

Los modelos IaaS, PaaS y SaaS ofrecen soluciones escalables y eficientes en términos de costos, y permiten a Juan ahorrar en la inversión en infraestructura y tecnología. Además, estos modelos en la nube brindan flexibilidad y accesibilidad remota, lo que significa que Juan y sus empleados pueden acceder a la aplicación en cualquier lugar y en cualquier momento.

Sin embargo, el modelo cliente-servidor también ofrece ventajas únicas, como un mayor control y personalización de la aplicación. Además, este modelo puede ser más adecuado para empresas con requisitos de seguridad y privacidad más estrictos, ya que la información se almacena y se procesa en servidores internos.

En última instancia, la elección del modelo adecuado dependerá de las necesidades específicas de la empresa de Juan y de sus objetivos a largo plazo. Es importante considerar cuidadosamente todas las ventajas y desventajas de cada modelo antes de tomar una decisión.