



UNIVERSIDAD POLITÉCNICA DE TECAMAC

MATERIA: CLIENTE SERVIDOR

PROFESOR: EMMANUEL TORRES SERVÍN

GRUPO: 1523IS

ALUMNOS:

- | | |
|--------------------------------|------------|
| • CRUZ HERNÁNDEZ ESMERALDA | 1321124171 |
| • RODRÍGUEZ PÉREZ MOISES | 1321124194 |
| • DESALES DECARO ELIAN GAEL | 1321124306 |
| • ZAMORA AGUILERA BERNARDO | 1321124165 |
| • AULIZ NATIVIDAD JOSE ALBERTO | 1122134801 |

FECHA DE ENTREGA: 08/02/2023

Índice

Índice.....	1
Descripción del problema	2
Reporte sobre la comunicación de dispositivos de red y las arquitecturas cliente servidor	2
Diagrama de componentes de la arquitectura Cliente/Servidor	15
Cuadro comparativo entre los modelos IAAS, PAAS, SAAS y Cliente/Servidor	15
Propuesta técnica de arquitectura Cliente/Servidor contemplando los modelos de cómputo en la nube	17
Justificación de selección de sistema operativo móvil	19
Justificación de lenguaje de programación móvil.....	20
Conclusión.....	20

Descripción del problema

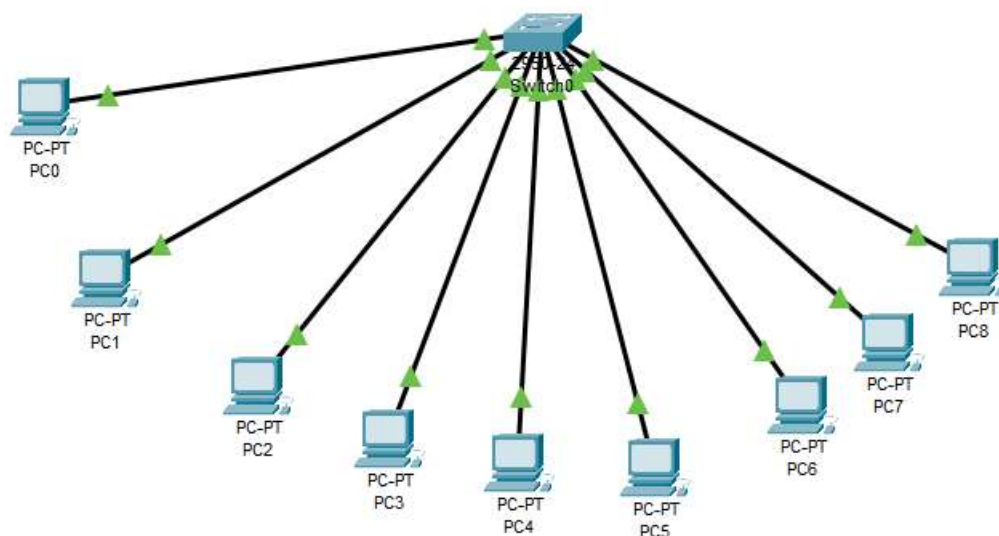
Juan tiene una empresa de patitos de hule, dicha empresa tiene 3 años y ya cuenta con 40 empleados, actualmente se encuentra en crecimiento constante, sin embargo el presupuesto sigue siendo limitado para cuestiones de inversión y mejora, por lo que ha perdido control de sus empleados en ciertas áreas 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.

Ayuda a Juan a encontrar la mejor opción para elegir como realizar los siguientes elementos:

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

La conexión prevista para la comunicación entre las máquinas de la compañía se tiene contemplada como una de tipo “LAN”, en la que se tendrá las limitaciones de dispositivos como la cantidad de ellos y el tamaño de las instalaciones de la empresa, como los puntos en los que se tiene previsto la conexión.

Para simular la conexión de dispositivos se ocupó el programa de “Cisco Packet Tracer”, en el que se ocuparon al menos nueve PC’s, que estaban conectadas a un switch de modelo “2950-24”.



Estos dispositivos tienen una red de “Clase C”, que posee una mascara de subred de “255.255.255.0”, también como ejemplo de una de las IP de las maquinas conectadas de la empresa se presenta “193.11.1.N”:

IP Configuration	
<input type="radio"/> DHCP	
<input checked="" type="radio"/> Static	
IPv4 Address	193.11.1.1
Subnet Mask	255.255.255.0

Para comprobar la conexión entre dispositivos se puede comprobar de dos maneras diferentes, la primera sería mandar un mensaje hipotético entre dispositivos, otro método usado será el uso del comando de "cmd" que es "ping", este método conlleva mayor tiempo, ya que es necesario introducir la IP de cada dispositivo para comprobar su conexión, pero gracias al programa de "Cisco Packet Tracer", basta con solo usar el icono de mensaje para comprobar conexión y si hay algún error se presentara para ser solucionado.

1. Dispositivo 1 conexión con dispositivo 2 al 9:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 193.11.1.2

Pinging 193.11.1.2 with 32 bytes of data:

Reply from 193.11.1.2: bytes=32 time<1ms TTL=128
Reply from 193.11.1.2: bytes=32 time<1ms TTL=128
Reply from 193.11.1.2: bytes=32 time<1ms TTL=128
Reply from 193.11.1.2: bytes=32 time=1ms TTL=128

Ping statistics for 193.11.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 193.11.1.3

Pinging 193.11.1.3 with 32 bytes of data:

Reply from 193.11.1.3: bytes=32 time=1ms TTL=128
Reply from 193.11.1.3: bytes=32 time<1ms TTL=128
Reply from 193.11.1.3: bytes=32 time<1ms TTL=128
Reply from 193.11.1.3: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 193.11.1.4

Pinging 193.11.1.4 with 32 bytes of data:

Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
Reply from 193.11.1.4: bytes=32 time=7ms TTL=128
Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
```

```
Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
Reply from 193.11.1.4: bytes=32 time=7ms TTL=128
Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
Reply from 193.11.1.4: bytes=32 time=8ms TTL=128

Ping statistics for 193.11.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 8ms, Average = 3ms

C:\>ping 193.11.1.5

Pinging 193.11.1.5 with 32 bytes of data:

Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time=7ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 7ms, Average = 1ms

C:\>ping 193.11.1.6

Pinging 193.11.1.6 with 32 bytes of data:

Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

C:\>ping 193.11.1.7

Pinging 193.11.1.7 with 32 bytes of data:

```

Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128

```

```

Ping statistics for 193.11.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

C:\>ping 193.11.1.8

Pinging 193.11.1.8 with 32 bytes of data:

```

Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128

```

```

Ping statistics for 193.11.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

C:\>ping 193.11.1.9

Pinging 193.11.1.9 with 32 bytes of data:

```

Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128

```

Pinging 193.11.1.9 with 32 bytes of data:

```

Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time=10ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128













```







```

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 2ms

```

C:\>

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)
	Successful	PC0	PC2	ICMP		0.000	N	1	(edit)
	Successful	PC0	PC3	ICMP		0.000	N	2	(edit)
	Successful	PC0	PC4	ICMP		0.000	N	3	(ec
	Successful	PC0	PC5	ICMP		0.000	N	4	(ec
	Successful	PC0	PC6	ICMP		0.000	N	5	(ec

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC0	PC4	ICMP		0.000	N	3	(ec
	Successful	PC0	PC5	ICMP		0.000	N	4	(ec
	Successful	PC0	PC6	ICMP		0.000	N	5	(ec

2. Dispositivo 2 conexión con dispositivo 3 al 9:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 193.11.1.3

Pinging 193.11.1.3 with 32 bytes of data:

Reply from 193.11.1.3: bytes=32 time<1ms TTL=128
Reply from 193.11.1.3: bytes=32 time<1ms TTL=128
Reply from 193.11.1.3: bytes=32 time<1ms TTL=128
Reply from 193.11.1.3: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.4

Pinging 193.11.1.4 with 32 bytes of data:

Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
Reply from 193.11.1.4: bytes=32 time<1ms TTL=128
Reply from 193.11.1.4: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.5

Pinging 193.11.1.5 with 32 bytes of data:

Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
```



```
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.6

Pinging 193.11.1.6 with 32 bytes of data:

Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time=1ms TTL=128

Ping statistics for 193.11.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 193.11.1.7

Pinging 193.11.1.7 with 32 bytes of data:

Reply from 193.11.1.7: bytes=32 time=1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

```

C:\>ping 193.11.1.8

Pinging 193.11.1.8 with 32 bytes of data:

Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.9

















Pinging 193.11.1.9 with 32 bytes of data:

Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>

```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC1	PC2	ICMP		0.000	N	8	(ec
	Successful	PC1	PC3	ICMP		0.000	N	9	(ec
	Successful	PC1	PC4	ICMP		0.000	N	10	(ec
	Successful	PC1	PC5	ICMP		0.000	N	11	(ec
	Successful	PC1	PC6	ICMP		0.000	N	12	(ec
	Successful	PC1	PC7	ICMP		0.000	N	13	(ec
	Successful	PC1	PC7	ICMP		0.000	N	13	(ec
	Successful	PC1	PC8	ICMP		0.000	N	14	(ec

3. Dispositivo 3 conexión con dispositivo 4 al 9:

```
C:\>ping 193.11.1.5

Pinging 193.11.1.5 with 32 bytes of data:

Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128
Reply from 193.11.1.5: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.6

Pinging 193.11.1.6 with 32 bytes of data:

Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.7

Pinging 193.11.1.7 with 32 bytes of data:

Reply from 193.11.1.7: bytes=32 time=10ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

```

C:\>ping 193.11.1.8

Pinging 193.11.1.8 with 32 bytes of data:

Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.9







Pinging 193.11.1.9 with 32 bytes of data:







Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>

```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC3	PC4	ICMP		0.000	N	16	(ec
	Successful	PC3	PC5	ICMP		0.000	N	17	(ec
	Successful	PC3	PC6	ICMP		0.000	N	18	(ec

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC3	PC6	ICMP		0.000	N	18	(ec
	Successful	PC3	PC7	ICMP		0.000	N	19	(ec
	Successful	PC3	PC8	ICMP		0.000	N	20	(ec

4. Dispositivo 4 conexión con dispositivo 5 al 9:

```
C:\>ping 193.11.1.6

Pinging 193.11.1.6 with 32 bytes of data:

Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128
Reply from 193.11.1.6: bytes=32 time=13ms TTL=128
Reply from 193.11.1.6: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 13ms, Average = 3ms

C:\>ping 193.11.1.7

Pinging 193.11.1.7 with 32 bytes of data:

Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time=4ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

C:\>ping 193.11.1.8

Pinging 193.11.1.8 with 32 bytes of data:

Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:


C:\>ping 193.11.1.9




Pinging 193.11.1.9 with 32 bytes of data:

Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC4	PC5	ICMP		0.000	N	21	(ec
	Successful	PC4	PC6	ICMP		0.000	N	22	(ec
	Successful	PC4	PC7	ICMP		0.000	N	23	(ec

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC4	PC6	ICMP		0.000	N	22	(ec
	Successful	PC4	PC7	ICMP		0.000	N	23	(ec
	Successful	PC4	PC8	ICMP		0.000	N	24	(ec

5. Dispositivo 5 conexión con dispositivo 6 al 9:

```
Pinging 193.11.1.7 with 32 bytes of data:

Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128
Reply from 193.11.1.7: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.7:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.8

Pinging 193.11.1.8 with 32 bytes of data:

Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 193.11.1.9

Pinging 193.11.1.9 with 32 bytes of data:

Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC5	PC6	ICMP		0.000	N	25	(ec
	Successful	PC5	PC7	ICMP		0.000	N	26	(ec
	Successful	PC5	PC8	ICMP		0.000	N	27	(ec

6. Dispositivo 6 conexión con dispositivo 7 al 9:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 193.11.1.8

Pinging 193.11.1.8 with 32 bytes of data:

Reply from 193.11.1.8: bytes=32 time=8ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128
Reply from 193.11.1.8: bytes=32 time<1ms TTL=128

Ping statistics for 193.11.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 8ms, Average = 2ms

C:\>ping 193.11.1.9

Pinging 193.11.1.9 with 32 bytes of data:

Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time=6ms TTL=128
Reply from 193.11.1.9: bytes=32 time=1ms TTL=128

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 6ms, Average = 1ms

C:\>|
```

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC5	PC8	ICMP		0.000	N	27	(ec
	Successful	PC6	PC7	ICMP		0.000	N	28	(ec
	Successful	PC6	PC8	ICMP		0.000	N	29	(ec

7. Dispositivo 7 conexión con dispositivo 8 al 9:

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Successful	PC6	PC7	ICMP		0.000	N	28	(ec
	Successful	PC6	PC8	ICMP		0.000	N	29	(ec
	Successful	PC7	PC8	ICMP		0.000	N	30	(ec

```

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 193.11.1.9

Pinging 193.11.1.9 with 32 bytes of data:

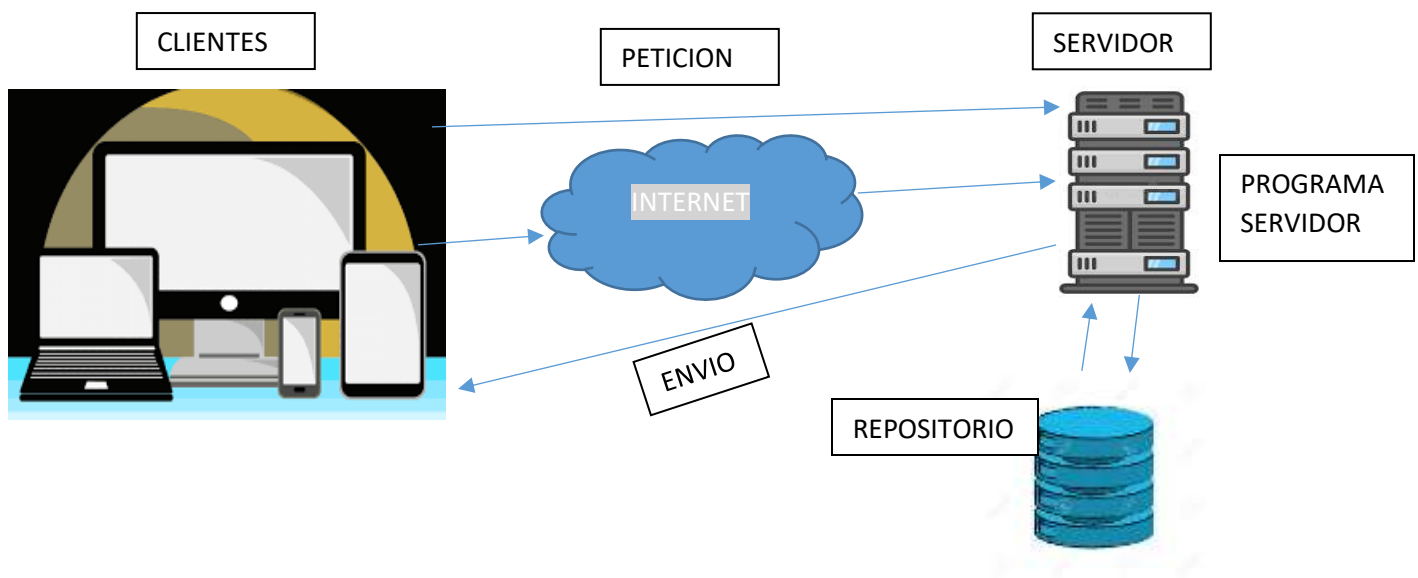
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time=7ms TTL=128
Reply from 193.11.1.9: bytes=32 time<1ms TTL=128
Reply from 193.11.1.9: bytes=32 time=1ms TTL=128

Ping statistics for 193.11.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 7ms, Average = 2ms

C:\>

```

Diagrama de componentes de la arquitectura Cliente/Servidor



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

IAAS (Infrastructure as a Service), PAAS (Platform as a Service), SAAS (Software as a Service) y Cliente/Servidor son diferentes modelos de entrega de tecnología que se utilizan para diferentes fines.

CARACTERÍSTICA	IAAS	PAAS	SAAS	CLIENTE/SERVIDOR
INFRAESTRUCTURA	Proporciona infraestructura (servidores, almacenamiento, red).	Proporciona plataforma e infraestructura.	Proporciona software e infraestructura.	Requiere que el cliente tenga su propia infraestructura.
MANTENIMIENTO	El proveedor se encarga de la administración y mantenimiento.	El proveedor se encarga de la administración y mantenimiento.	El proveedor se encarga de la administración y mantenimiento.	El cliente se encarga de la administración y mantenimiento.
PERSONALIZACIÓN	Alta personalización.	Menor personalización que IAAS.	Mínima personalización.	Alta personalización.
COSTO	Costoso.	Menos costoso que IAAS.	Generalmente más económico que IAAS y PAAS.	Puede ser costoso si se requiere una gran cantidad de recursos.

ESCALABILIDAD	Alta escalabilidad	Alta escalabilidad	Limitada	Limitada
Responsabilidad	Infraestructura básica	Plataforma	Aplicación	Todo

En resumen, IAAS es adecuado para organizaciones que desean tener control total sobre su infraestructura tecnológica, mientras que PAAS y SAAS son opciones más económicas y convenientes para organizaciones que buscan una solución más rápida y fácil de implementar y administrar. La solución de cliente/servidor es adecuada para aplicaciones que requieren una alta personalización y control sobre la infraestructura.

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

Esta propuesta describe una solución para construir una arquitectura cliente-servidor escalable, segura y de alto rendimiento utilizando módulos de computación en la nube. La solución propuesta tiene como objetivo proporcionar una forma robusta y eficiente para que los clientes accedan a los recursos y servicios del servidor a través de una red.

La arquitectura cliente-servidor constará de dos componentes principales, el cliente y el servidor. El cliente interactuará con el servidor utilizando un protocolo estándar, como HTTP o HTTPS, para solicitar recursos y servicios. El servidor, por su parte, almacenará y gestionará los recursos y servicios a los que puede acceder el cliente.

La solución propuesta utilizará módulos de computación en la nube para brindar beneficios de escalabilidad, seguridad y rendimiento. Los módulos de computación en la nube permitirán que el servidor asigne dinámicamente recursos, como capacidad de procesamiento y almacenamiento, según sea necesario. Esto

asegurará que el servidor pueda manejar las crecientes demandas de los clientes mientras mantiene un alto rendimiento.

Los módulos de computación en la nube también proporcionarán funciones de seguridad integradas, como firewalls, encriptación y controles de acceso, para proteger el servidor y sus recursos de ataques maliciosos.

La solución propuesta será altamente escalable, lo que permitirá que el servidor se adapte a las crecientes demandas de los clientes sin sacrificar el rendimiento. Esto se logrará mediante el uso de módulos de computación en la nube, que permitirán que el servidor asigne recursos dinámicamente según sea necesario.

La seguridad es una prioridad máxima en la solución propuesta. Los módulos de computación en la nube proporcionarán funciones de seguridad integradas, como firewalls, encriptación y controles de acceso, para proteger el servidor y sus recursos de ataques maliciosos. Además, el uso de protocolos estándar, como HTTP o HTTPS, garantizará que la comunicación entre el cliente y el servidor sea segura.

La solución propuesta proporcionará un alto rendimiento al utilizar módulos de computación en la nube para asignar recursos de forma dinámica según sea necesario. Esto asegurará que el servidor pueda manejar las crecientes demandas de los clientes mientras mantiene un alto rendimiento.

Por ende la solución propuesta para construir una arquitectura cliente-servidor utilizando módulos de computación en la nube proporciona una solución escalable, segura y de alto rendimiento para que los clientes accedan a los recursos y servicios del servidor a través de una red. El uso de módulos de computación en la nube proporcionará beneficios de escalabilidad, seguridad y rendimiento, lo que la convierte en una solución ideal para cualquier organización que busque construir una arquitectura cliente-servidor sólida y eficiente.

Justificación de selección de sistema operativo móvil

La elección de un sistema operativo móvil depende de muchos factores, incluyendo los requisitos del usuario, las características del hardware y las preferencias personales. Aquí hay algunos de los motivos más comunes por los que la gente elige un sistema operativo móvil en particular:

La forma en que se presenta y se interactúa con un sistema operativo móvil puede ser un factor importante para muchos usuarios. Algunos sistemas operativos tienen una interfaz más intuitiva y fácil de usar, mientras que otros ofrecen más personalización y configuración avanzada.

La disponibilidad de aplicaciones es otro factor importante a la hora de elegir un sistema operativo móvil. Algunos sistemas operativos tienen una amplia selección de aplicaciones en su tienda en línea, mientras que otros tienen una selección más limitada.

Para algunos usuarios, la privacidad y la seguridad son una preocupación importante. Algunos sistemas operativos ofrecen características de privacidad y seguridad avanzadas, como la encriptación de datos y la autenticación de dos factores.

Si el usuario tiene otros dispositivos con un sistema operativo en particular, puede ser más fácil y conveniente elegir el mismo sistema operativo para su dispositivo móvil.

Finalmente, el costo puede ser un factor importante a la hora de elegir un sistema operativo móvil. Algunos sistemas operativos son gratuitos, mientras que otros pueden tener un costo adicional.

En resumen, la elección de un sistema operativo móvil depende de las necesidades y preferencias individuales de cada usuario. Es importante evaluar cuidadosamente las opciones disponibles antes de tomar una decisión.

Justificación de lenguaje de programación móvil

El lenguaje que vamos ocupar será Javascript ya que es un programa muy versátil para la creación de la aplicación, ya que nos permite utilizar múltiples plataformas por ejemplo (iOS, Android y Windows), Uno de sus beneficios es el marco React Native, esto nos permite escribir el código una vez, para así ejecutarlo en los diferentes dispositivos, por lo cual nos reducirá el tiempo y el esfuerzo necesario para poder desarrollarla.

Ya que proporciona una gran cantidad de recursos, bibliotecas y herramientas para ayudarnos al desarrollo de la creación de la aplicación móvil de alta calidad. Además, tiene una sintaxis relativamente simple y es fácil de aprender, lo que lo hace accesible y muy entendible a la hora de programar dicha aplicación.

Finalmente, lenguaje dinámico y flexible que nos permitirá realizar cambios rápidamente y probar nuevas funciones en tiempo real. Esto puede ayudar a acelerar el proceso de desarrollo y facilitar la respuesta a las necesidades cambiantes de los usuarios y las tendencias del mercado.

Conclusión

Como conclusión, hemos visto que JavaScript ha demostrado ser un componente crucial del desarrollo moderno de aplicaciones web y móviles. Con su naturaleza dinámica y de alto nivel, ofrece a los desarrolladores la flexibilidad y eficiencia que necesitan para crear interfaces de usuario rápidas y receptivas. La gran y activa comunidad de JavaScript también proporciona amplios recursos y soporte, por lo que es una opción preferida para muchos desarrolladores. Además, con el auge de los marcos de desarrollo móvil multiplataforma como React Native, JavaScript se ha convertido en un lenguaje popular para crear aplicaciones móviles que en este caso es importante para realizar esta problemática.