



ITSSAT

LISTA DE COTEJO PARA INVESTIGACION

| | | |
|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------|----------------------------|
| INTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRES TUXTLA | | |
| CARRERA: INGENIERIA INFORMÁTICA | | |
| DATOS GENERALES DEL PROCESO DE EVALUACION | | |
| Nombre(s) del alumno(s): ACEVEDO MENDEZ JONATHAN EMMANUEL | | Firma del alumno(s): |
| Producto: Investigación Unidad I | Nombre del Proyecto: Lenguajes de Programación del lado del Cliente y del Lado del Servidor | Fecha: 14 / Marzo /2023 |
| Asignatura: PROGRAMACIÓN EN AMBIENTE CLIENTE/SERVIDOR | Grupo: 610 - A | Semestre: SEXTO |
| Nombre del Docente: MTI Lorenzo de Jesús Organista Oliveros | | Firma del Docente: |

| INSTRUCCIONES | | | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----|---------------|
| Revisar las actividades que se solicitan y marque en los apartados "SI" cuando la evidencia se cumple; en caso contrario marque "NO". En la columna OBSERVACIONES indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario. | | | | |
| Valor del reactivo | Características a cumplir (Reactivo) | CUMPLE | | OBSERVACIONES |
| | | SI | NO | |
| 1% | Presentación. El trabajo cumple con los requisitos de : a. Buena presentación | X | | |
| 1% | b. No tiene faltas de ortografía | X | | |
| 1% | c. Mismo formato (letra arial 14, títulos con negritas) | X | | |
| 1% | d. Misma calidad de hoja e impresión | X | | |
| 1% | e. Maneja el lenguaje técnico apropiado | X | | |
| 2% | Introducción y Objetivo. La introducción y el objetivo dan una idea clara del contenido del trabajo, motivando al lector a continuar con su lectura y revisión. | X | | |
| 5% | Sustento Teórico. Presenta un panorama general del tema a desarrollar y lo sustenta con referencias bibliográficas y cita correctamente a los autores. | X | | |
| 2% | Desarrollo. Sigue una metodología y sustenta todos los pasos que se realizaron al aplicar los conocimientos obtenidos, es analítico y bien ordenado. | X | | |
| 2% | Resultados. Cumplió totalmente con el objetivo esperado, tiene aplicaciones concretas. | X | | |
| 2% | Conclusiones. Las conclusiones son claras y acordes con el objetivo esperado. | X | | |
| 2% | Responsabilidad. Entregó el reporte en la fecha y hora señalada. | X | | |
| 20% | CALIFICACIÓN: | | | 20% |



**INSTITUTO TECNOLÓGICO SUPERIOR DE
SAN ANDRÉS TUXTLA**

Asignatura:

Programación en ambiente cliente - servidor

Docente:

Lorenzo de Jesús Organista Oliveros

Alumno:

Jonathan Emmanuel Acevedo Mendez

Trabajo:

Investigación

Fecha de entrega:

14/03/2023



Grupo: 610 A

Introducción:

Los lenguajes de programación buy-side y server-side son particularmente útiles para trabajos que necesitan acceso a información centralizada que reside en una base de datos en el servidor y cuando los cálculos no se pueden realizar en la PC del cliente por razones de estabilidad. Es importante recalcar que los lenguajes de programación son necesarios del lado del servidor, ya que para crear la mayoría de las aplicaciones web se debe tener acceso a diversos recursos fuera de la PC del comprador, principalmente bases de datos alojadas en servidores de Internet. Un caso claro es un banco: no tiene sentido que el comprador tenga acceso a toda la base de datos, sino solo a la información que le concierne.

La programación del lado del servidor, estos son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se transmiten al comprador en un formato que este pueda entender. Hay multitud de lenguajes que pueden o no haber sido diseñados para Internet. Todos ellos aprovechan aún más ciertas propiedades que los hacen más o menos útiles para el desarrollo de diversas aplicaciones. Un lenguaje del lado del servidor se ejecuta en el servidor web justo antes de que la página se envíe por Internet al comprador.

– Lenguajes de programación del Lado del Cliente.

Lado del cliente y lado del servidor son términos de desarrollo web que describen dónde se ejecuta el código de la aplicación. Los desarrolladores web también se refieren a esta distinción como frontend vs. backend, aunque lado del cliente/servidor y frontend/backend no son exactamente lo mismo. En una arquitectura sin servidor, el proveedor de servicios sin servidor aloja y asigna recursos a todos los procesos del lado del servidor, y los procesos escalan a medida que aumenta el uso de la aplicación.

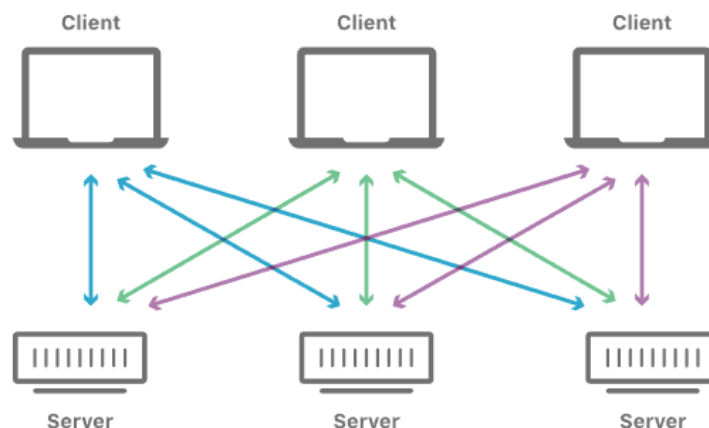
¿Qué es el modelo cliente-servidor?

Buena parte de Internet se basa en el modelo cliente-servidor. En este modelo, los dispositivos de los usuarios se comunican mediante una red con servidores centralizados para obtener los datos que necesitan, en lugar de comunicarse entre sí. Los dispositivos de los usuarios finales, como portátiles, teléfonos inteligentes y ordenadores de escritorio, se consideran "clientes" de los servidores, como si fueran clientes que obtienen servicios de una empresa. Los dispositivos de los clientes envían solicitudes a los servidores de páginas web o aplicaciones, y los servidores les sirven respuestas.

El modelo cliente-servidor se utiliza porque los servidores suelen ser más potentes y fiables que los dispositivos de los usuarios. Además, se mantienen y se conservan permanentemente en entornos controlados para garantizar que estén siempre encendidos y disponibles; aunque haya servidores individuales que hayan dejado de funcionar, suele haber otros servidores de reserva. Mientras tanto, los usuarios pueden encender y apagar sus dispositivos, o perderlos o romperlos, y eso no debería afectar al servicio de Internet de otros usuarios.

Los servidores pueden servir a varios dispositivos clientes a la vez, y cada dispositivo cliente envía solicitudes a varios servidores durante el acceso y la navegación por Internet.

Varios clientes y servidores interactúan: Cada cliente se comunicará con varios servidores, y viceversa.



Supongamos que un usuario está navegando por Internet y escribe "netflix.com" en la barra del navegador. Esto da lugar a una solicitud a los servidores DNS para la dirección IP de netflix.com, y los servidores DNS responden a esta solicitud sirviendo la dirección IP al navegador. A continuación, el navegador del usuario realiza una solicitud a los servidores de Netflix (usando la dirección IP) para el contenido que aparece en la página, como las imágenes en miniatura de las películas, el logotipo de Netflix y la barra de búsqueda. Los servidores de Netflix lo entregan al navegador y esta carga la página en el dispositivo del cliente.

| Lenguajes del lado del cliente | |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HTML | El lenguaje llamado HTML indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. |
| JAVASCRIPT | JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. |
| APPLETS DE JAVA | Es otra manera de incluir código a ejecutar en los clientes que visualizan una página web. Se trata de pequeños programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página. |
| VISUAL BASIC SCRIPT | Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de JavaScript. |
| FLASH | Flash es una tecnología, y un programa, para crear efectos especiales en páginas web. Con Flash también conseguimos hacer páginas dinámicas del lado del cliente. Flash en realidad no es un lenguaje; Sin embargo, si tuviéramos que catalogarlo en algún sitio quedaría dentro del ámbito de las páginas dinámicas de cliente. |
| CSS | CSS, es una tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores... Incluso podemos definir nuestros propios estilos en un archivo externo a nuestras páginas; así, si en algún momento queremos cambiar alguno de ellos, automáticamente se nos actualizarán todas las páginas vinculadas de nuestro sitio. |

– Lenguajes de programación del lado del Servidor.

Se conoce como lenguaje del lado del servidor a aquel que se ejecuta en el servidor web, inmediatamente antes de que el sitio web se envíe a través de Internet al usuario. Los sitios web que se ejecutan en el servidor pueden realizar un amplio abanico de tareas hasta formar el propio sitio web que va a ver el usuario: acceso a base de datos, conexión en red.

¿Qué significa lado del servidor?

Como ocurre con el lado del cliente, "lado del servidor" se refiere a todo lo que ocurre en el servidor, en lugar de en el cliente. En el pasado, casi toda la lógica empresarial se ejecutaba en el lado del servidor, y esto incluía la representación de páginas web dinámicas, la interacción con las bases de datos, la autenticación de identidades y las notificaciones push.

El problema que hay con alojar todos estos procesos en el lado del servidor es que cada solicitud que implique a uno de ellos tiene que recorrer todo el camino desde el cliente hasta el servidor, y eso cada vez. Esto produce una gran cantidad de latencia. Por esta razón, las aplicaciones actuales ejecutan más código en el lado del cliente; un caso de uso es la representación de páginas web dinámicas en tiempo real mediante la ejecución de scripts en el navegador, que hacen cambios en el contenido que ve el usuario.

Al igual que "frontend" y "lado del cliente", backend también es un término para los procesos que tienen lugar en el servidor, aunque backend solo hace referencia a los tipos de procesos y lado del servidor hace referencia a la ubicación en la que se ejecutan los procesos.

Los datos van y vienen entre el cliente y el servidor, y los scripts se pueden ejecutar en ambos lados.

Así pues, podemos hablar de lenguajes de lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Cuando nosotros pinchamos sobre un enlace hipertexto, en realidad lo que pasa es que establecemos una petición de un archivo HTML residente en el servidor (un ordenador que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por nuestro navegador (el cliente).

Cada uno de estos tipos tiene por supuesto sus ventajas y sus inconvenientes. Así, por ejemplo, un lenguaje de lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas. Inversamente, un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo.

¿Qué ventajas tiene la programación del lado del servidor?

La gran ventaja del SSR es que el servidor carga previamente las páginas web. La petición del usuario se procesa casi de forma inmediata, lo que le permite acceder con gran rapidez a la página. Este enfoque es muy útil para las webs estáticas. Además, la rapidez de carga de la página tiene una influencia positiva en el posicionamiento en buscadores, pues gracias al SSR los crawlers pueden rastrear las páginas con mayor facilidad.

¿Cuáles son las desventajas de la programación del lado del servidor?

El server-side scripting requiere que el servidor entregue páginas HTML precargadas con cada solicitud. Si un cliente sigue enviando peticiones al servidor web para mostrar al usuario información nueva y modificada, se produce una sobrecarga de las capacidades del servidor. Por ello, el SSR no se recomienda en páginas web con un gran número de peticiones o que requieren un gran número de interacciones de los usuarios. En estos proyectos, el tiempo de respuesta del servidor web anularía la ventaja de la carga rápida de páginas.



Lenguajes del lado del servidor

| | |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CGI | Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos. |
| PERL | Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código Perl. |
| ASP.NET | ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (JavaScript de Microsoft). |
| PHP | PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. |
| JSP | <p>JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas web con programación en Java.</p> <p>Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma.</p> |

Conclusión:

Los lenguajes de programación del lado del servidor son particularmente útiles para trabajos que necesitan acceso a información centralizada que reside en una base de datos en el servidor, y porque los cálculos no se pueden realizar en la PC del servidor-cliente por razones de estabilidad. También es importante tener en cuenta que se requieren lenguajes de programación en el lado del servidor, ya que para crear la mayoría de las aplicaciones web, debe tener acceso a muchos recursos externos en la PC del comprador, principalmente bases de datos alojadas en servidores de Internet.

Fuentes de información:

[1] Danny Goodman. Programación en JavaScript. Vía@Internet, Anaya Multimedia, México.

https://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html

[2] Eduardo Parra Murga. Lado del cliente - servidor. PC World, IDG Communications, Madrid.

<https://www.cloudflare.com/es-es/learning/serverless/glossary/client-side-vs-server-side/#:~:text=El%20scripting%20del%20lado%20del,este%20lenguaje%20es%20universalmente%20compatible.>

[3] Sergio Ríos Aguilar. Lenguajes HTML, Java y CGI. El diseño de páginas Web para Internet a su alcance. Abeto Editorial, Madrid, 2001.

<https://axarnet.es/blog/lenguajes-del-lado-del-servidor>

[4] Oscar González Moreno. Programación en JavaScript. Guías Prácticas, Anaya Multimedia, México, 2020.

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/lenguajes-del-lado-servidor-o-del-cliente-diferencias/>



GUIA DE OBSERVACIÓN PARA RESOLUCIÓN DE EJERCICIOS PRACTICOS

NOMBRE DE LA ASIGNATURA: **PROGRAMACIÓN EN AMBIENTE CLIENTE/SERVIDOR**

NOMBRE DE LA UNIDAD: **CONTEXTO DE LA PROGRAMACIÓN CLIENTE/SERVIDOR**

ALUMNO: **ACEVEDO MENDEZ JONATHAN EMMANUEL**

INSTRUCCIONES

Revisar los documentos o actividades que se solicitan y marque en los apartados “SI” cuando la evidencia a evaluar se cumple; en caso contrario marque “NO”. En la columna “OBSERVACIONES” ocúpela cuando tenga que hacer comentarios referentes a lo observado.

| Valor del reactivo | Características a cumplir (Reactivo) | CUMPLE | | OBSERVACIONES |
|---------------------------|-----------------------------------------------------------------------------------|---------------|-----------|----------------------|
| | | Si | NO | |
| 8% | ¿Identifico el problema planteado? | X | | |
| 4% | ¿Identifico los datos de entrada del problema? | X | | |
| 4% | ¿Identifico los datos de salida del problema? | X | | |
| 8% | ¿Generó la solución del problema en forma clara y comprensible (orden)? | X | | |
| 12% | ¿Elaboró el programa respetando la sintaxis del lenguaje de programación (orden)? | X | | |
| 4% | Comprobó los resultados esperados a través de pruebas de escritorio? | X | | |
| 40% | CALIFICACIÓN: | | | 40% |



**INSTITUTO TECNOLÓGICO SUPERIOR DE
SAN ANDRÉS TUXTLA**

Asignatura:

Programación en ambiente cliente - servidor

Docente:

Lorenzo de Jesús Organista Oliveros

Alumno:

Jonathan Emmanuel Acevedo Mendez

Trabajo:

Practica U1

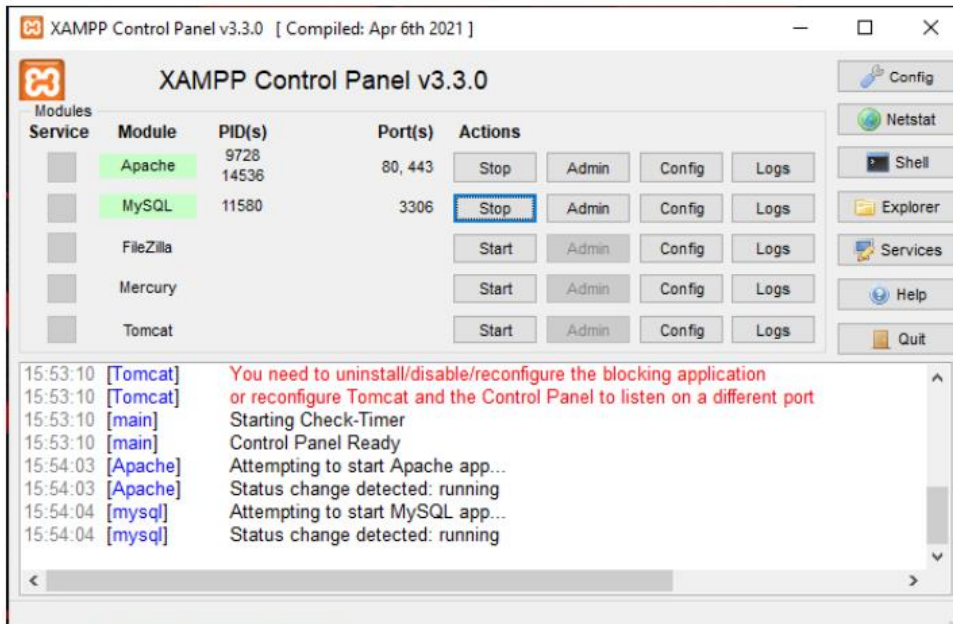
Fecha de entrega:

15/03/2023

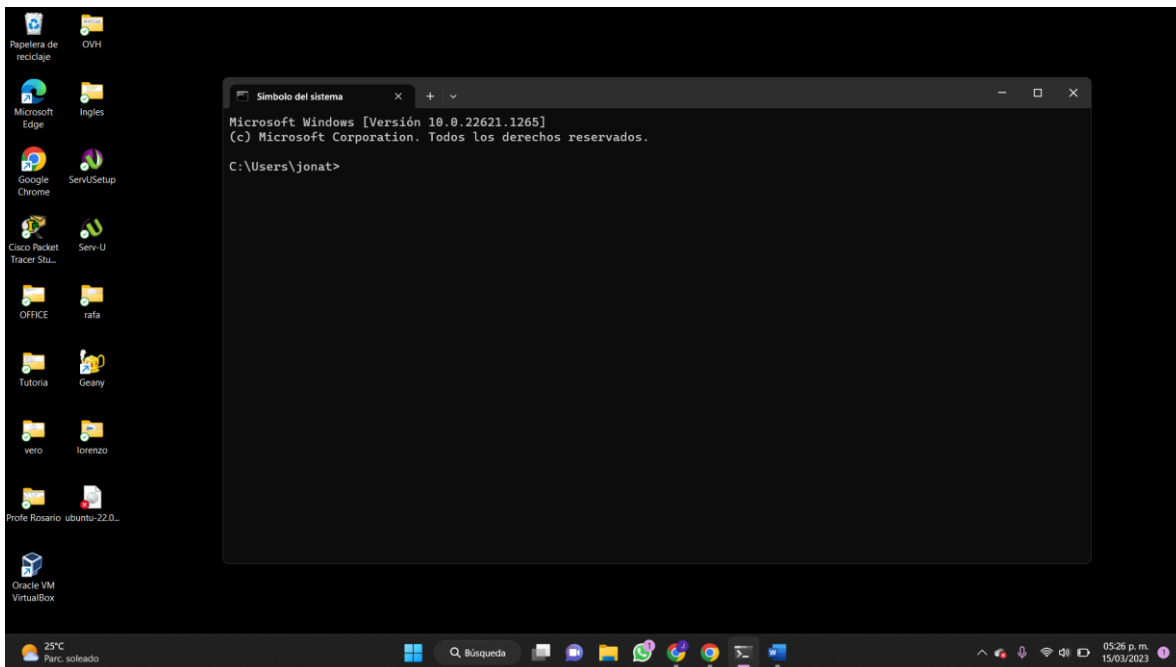


Grupo: 610 A

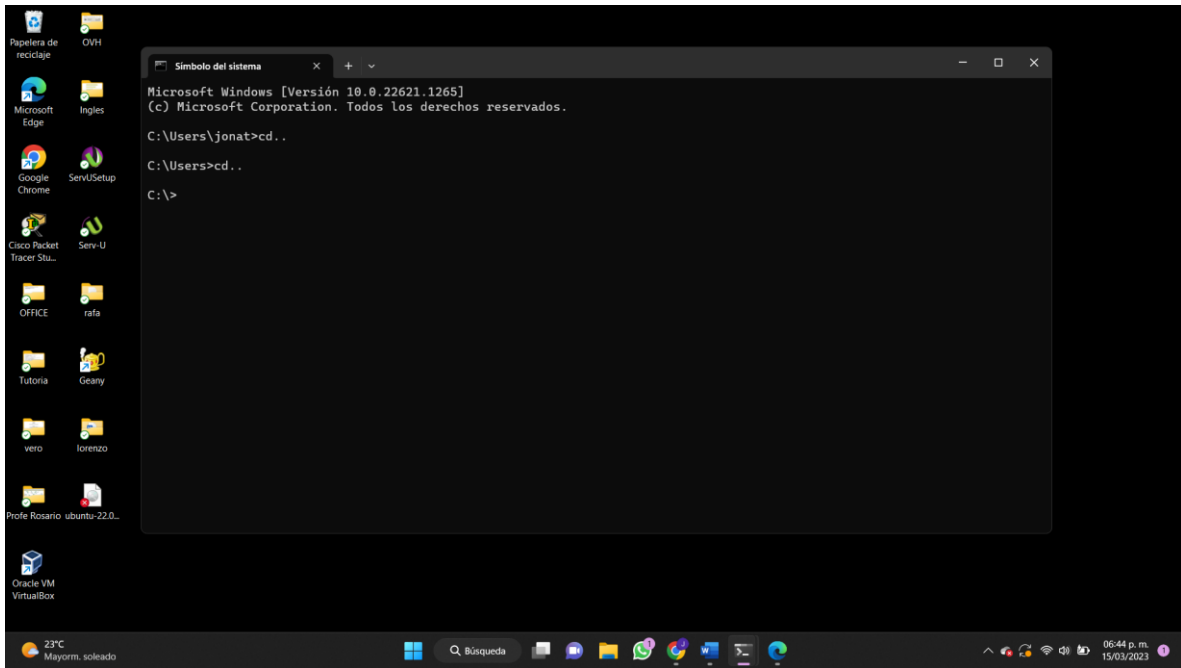
- ✓ Primero activaremos lo que es el Apache y MySQL para crear el servidor.



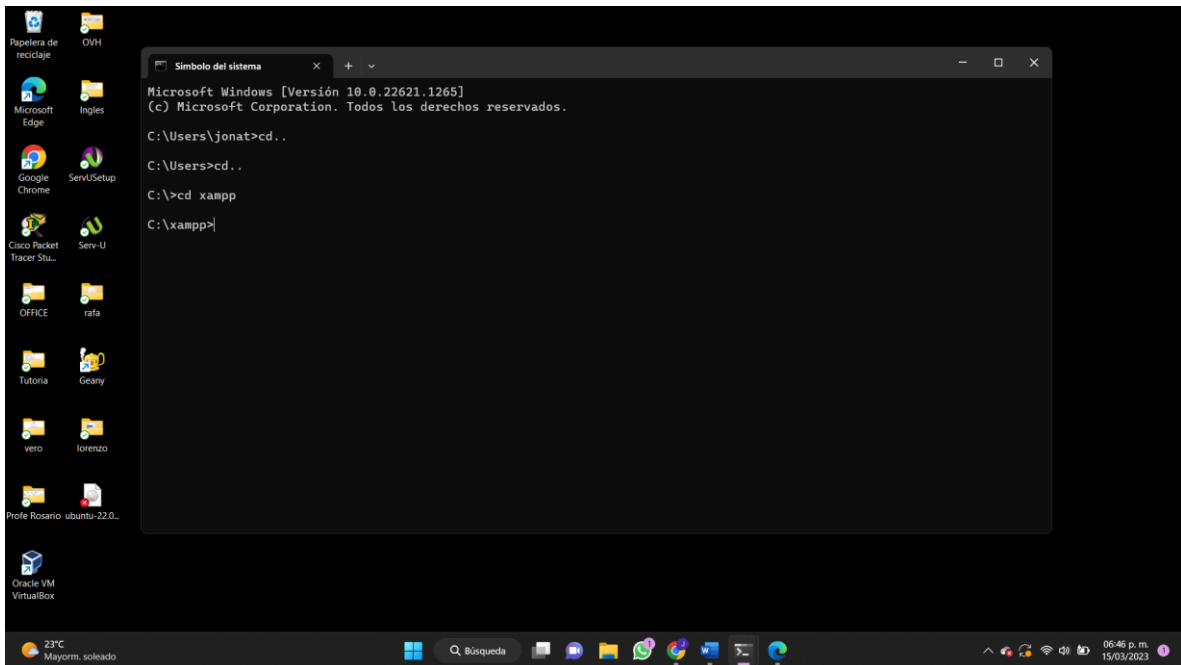
- ✓ Abrir el símbolo del sistema.



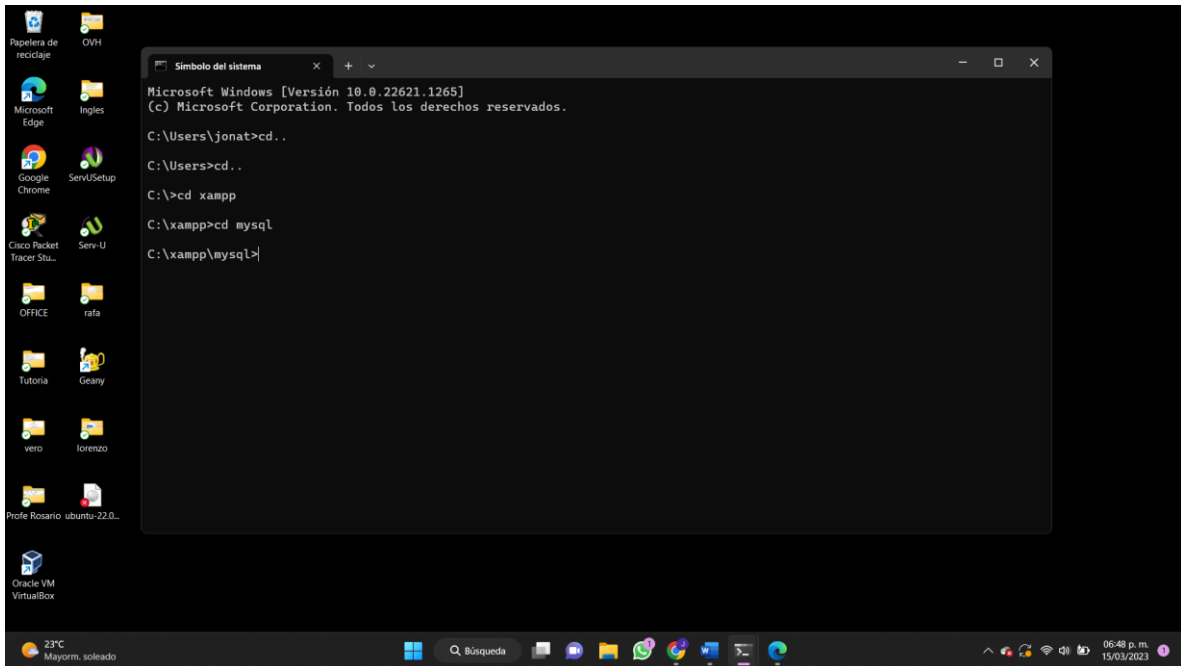
✓ Ingresaremos los comandos de cd.



✓ Luego iniciamos el XAMPP.



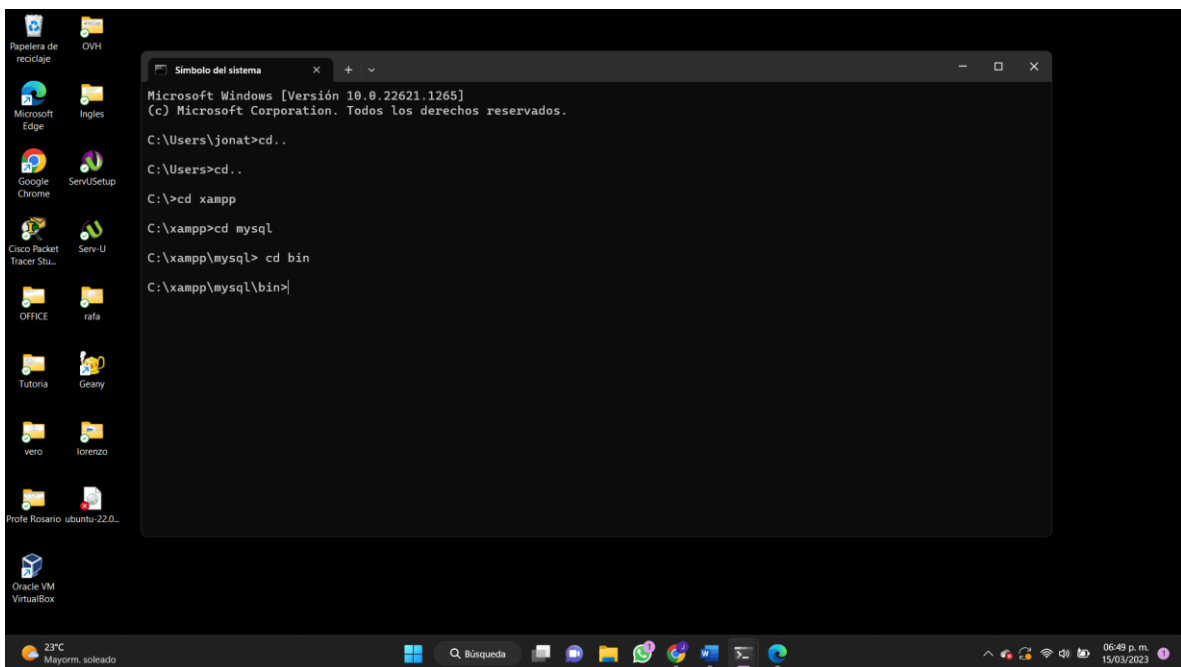
✓ Después del iniciar el XAMPP, iniciamos el MySQL.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.1265]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jonat>cd..
C:\Users>cd..
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql>
```

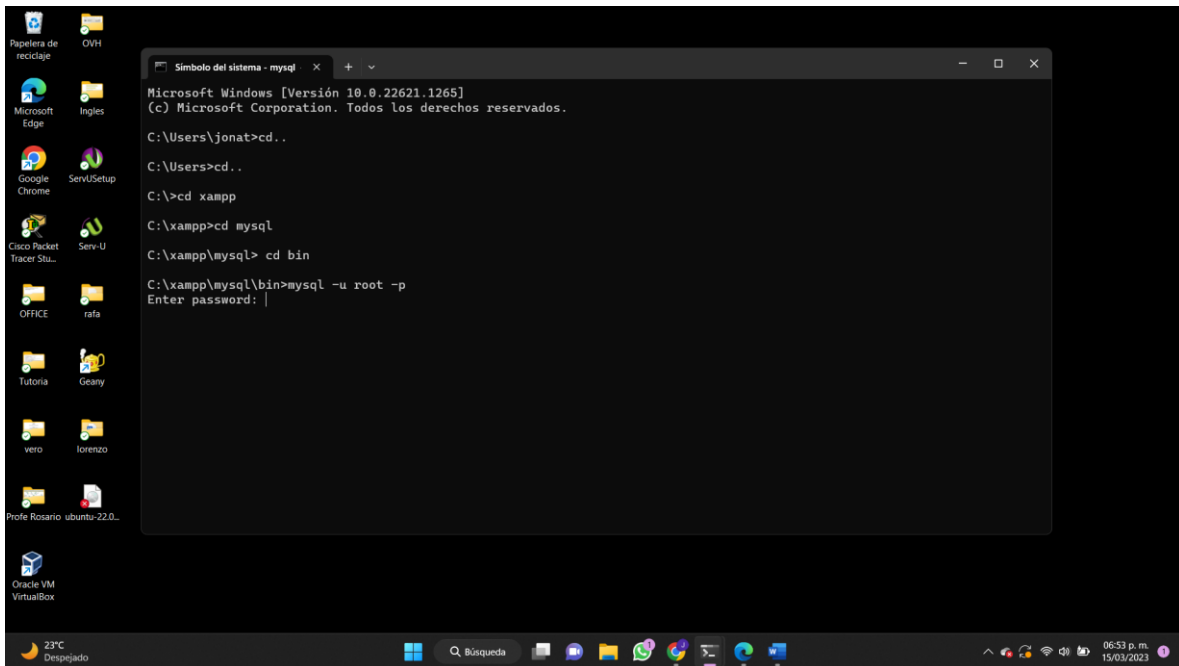
✓ Y como tercero ponemos el comando cd bin.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.1265]
(c) Microsoft Corporation. Todos los derechos reservados.

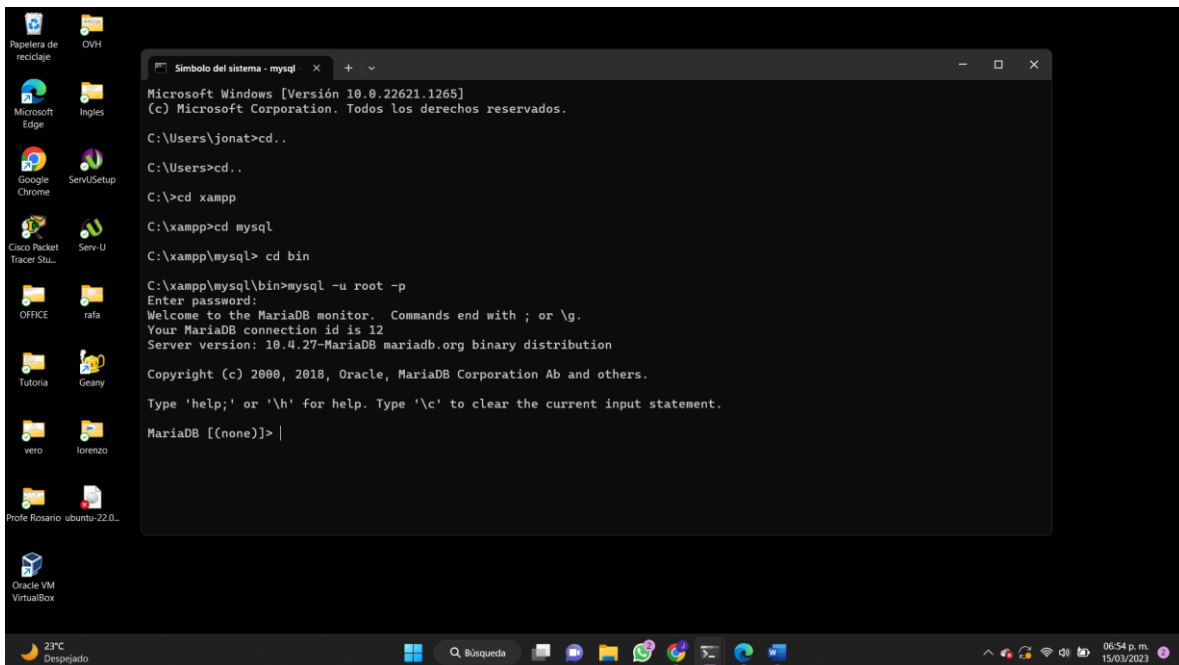
C:\Users\jonat>cd..
C:\Users>cd..
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql> cd bin
C:\xampp\mysql\bin>
```

- ✓ Una vez estando dentro tenemos que iniciar sesión como administrando y para eso ocuparemos el siguiente comando, MySQL -u root -p, nos aparecerá poner un password solo le daremos enter.



```
Símbolo del sistema - mysql x + -
Microsoft Windows [Versión 10.0.22621.1265]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jonat>cd ..
C:\Users>cd ..
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql> cd bin
C:\xampp\mysql\bin>mysql -u root -p
Enter password: |
```



```
Símbolo del sistema - mysql x + -
Microsoft Windows [Versión 10.0.22621.1265]
(c) Microsoft Corporation. Todos los derechos reservados.

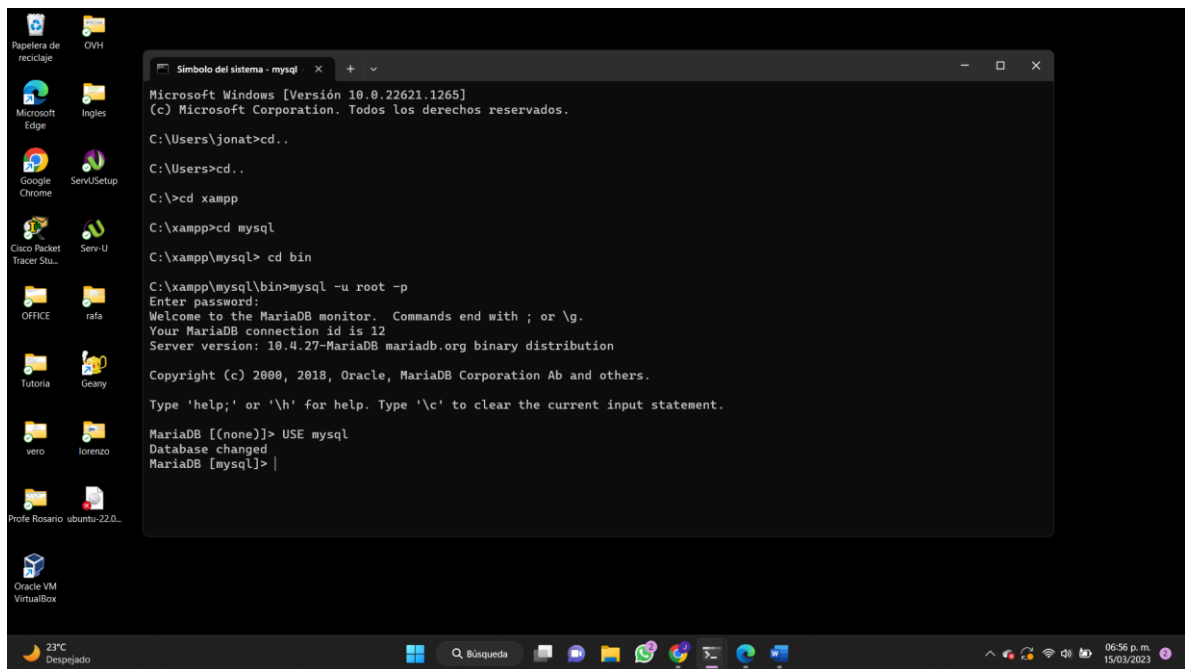
C:\Users\jonat>cd ..
C:\Users>cd ..
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql> cd bin
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```


✓ Luego pondremos el comando se USE MySQL.



```
Microsoft Windows [Versión 10.0.22621.1265]
(c) Microsoft Corporation. Todos los derechos reservados.

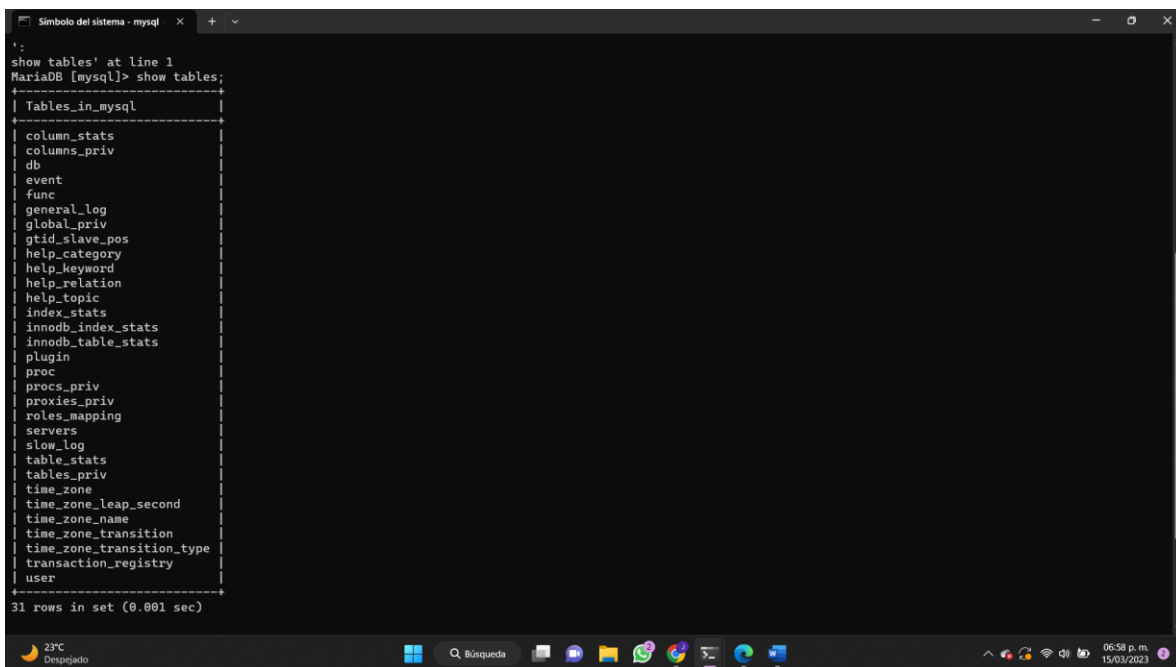
C:\Users\jonat>cd..
C:\Users>cd..
C:\>cd xampp
C:\xampp>cd mysql
C:\xampp\mysql> cd bin
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> USE mysql
Database changed
MariaDB [mysql]> |
```

✓ Después pondremos el siguiente comando show tables; .



```
show tables' at line 1
MariaDB [mysql]> show tables;
+-----+
| Tables_in_mysql |
+-----+
| column_stats    |
| columns_priv   |
| db              |
| event          |
| func           |
| general_log    |
| global_priv    |
| gtid_slave_pos |
| help_category  |
| help_keyword   |
| help_relation  |
| help_topic     |
| index_stats    |
| innodb_index_stats |
| innodb_table_stats |
| plugin         |
| proc          |
| procs_priv     |
| proxies_priv  |
| roles_mapping  |
| servers       |
| slow_log      |
| table_stats    |
| tables_priv   |
| time_zone     |
| time_zone_leap_second |
| time_zone_name |
| time_zone_transition |
| time_zone_transition_type |
| transaction_registry |
| user          |
+-----+
31 rows in set (0.001 sec)
```


✓ Y lo cerraremos con un exit y seria todo.

```
Simbolo del sistema x + -
| time_zone_transition |
| time_zone_transition_type |
| transaction_registry |
| user |
+-----+
31 rows in set (0.001 sec)

MariaDB [mysql]> CREATE USER "JONATHAN@"%"IDENTIFIDE BY "JONATHAN";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server versi
on for the right syntax to use near 'IDENTIFIDE BY "JONATHAN"' at line 1
MariaDB [mysql]> CREATE USER "JONATHAN@"%" IDENTIFIDE BY "JONATHAN";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server versi
on for the right syntax to use near 'IDENTIFIDE BY "JONATHAN"' at line 1
MariaDB [mysql]> exit
Bye

C:\xampp\mysql\bin>mysql -u jonathan -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Bye

C:\xampp\mysql\bin>
```

1. Contexto de la programación cliente/servidor :



Evaluación - Unidad I

Fecha de entrega: 21 mar, 11...

Publicado: 21 mar

Resuelve correctamente lo que se te pide: (Valor 40%)

15

Entregadas

0

Asignadas



Examen - Unidad I

Examen - Unidad I

Contexto de la programación Cliente/Servidor

Se ha registrado el correo del encuestado (**201u0221@alumno.itssat.edu.mx**) al enviar este formulario.

1. Modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático. *

- Aplicación monolítica
- Modelo cliente/servidor
- Aplicación 2,3 y n capas
- Aplicación distribuida

2. Dividen una aplicación entre un cliente y un servidor estableciendo un middleware que controla las comunicaciones entre ambos. *

- Aplicación monolítica
- Aplicación cliente/servidor
- Aplicación 2 capas
- Aplicación 3 y n capas

3. Es el que inicia un requerimiento de servicio. *

- Usuario
- Sistema Operativo
- Cliente
- Servidor

4. Pueden estar conectados a través de redes LANs o WANs, para proveer de múltiples servicios. *

- Usuario
- Sistemas Operativo
- Cliente
- Servidor

5. Su función es permitir el acceso remoto a archivos almacenados en él. *

- Servidor de Base de Datos
- Servidor de Archivos
- Servidor Web
- Servidor de Transacciones

6. Software diseñado para permitir colaborar a los usuarios sin importar la localización. *

- Servidor Web
- Servidor de Archivos
- Servidor Groupware
- Servidor de Transacciones

7. Sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador del usuario. *

- Servidor de Base de Datos
- Servidor Web
- Servidor de Archivos
- Servidor de Transacciones

8. Procesa todas las transacciones, valida y genera un pedido al servidor de base de datos. *

- Servidor de Base de Datos
- Servidor de Archivos
- Servidor Web
- Servidor de Transacciones

9. Es la forma de comunicarse la interfaz de usuario y la regla de negocio en un modelo de tres capas. *

- COM
- ODBC
- JAVA
- HTML

10. Es la forma de comunicarse la regla de negocio y las Base de Datos en un modelo de tres capas. *

- COM
- ODBC
- MYSQL
- SQL SERVER

Untitled Question

- Option 1

Este formulario se creó en INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA.

Google Formularios