|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DEPARTAMENTO:** | CIENCIAS DE LA COMPUTACIÓN | **CARRERA:** | INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN | | |
| **ASIGNATURA:** | APLICACIONES DE SISTEMAS OPERATIVOS | **PERIODO LECTIVO:** | MAYO 2023-  SEP 2023 | **NIVEL:** | 5to |
| **DOCENTE:** | ING. ANDREA LÓPEZ | **NRC:** | 10035 | **PRÁCTICA N°:** | 2.1 |
| **TEMA DE LA PRÁCTICA:** | ARQUITECTURA CLIENTE - SERVIDOR | | | | |
| **OBJETIVOS:** | | | | | |
| 1. Implementar un modelo de paso de mensajes en un arquitectura cliente-servidor (se sugiere usar los siguientes lenguajes: C, C++, Python) 2. Se pueden usar las máquinas en donde Ubuntu sea el servidor y Windows 10 el cliente. (opcional) 3. Utilizar una herramienta de gestión como Putty para acceder al servidor y ejecutar. | | | | | |
| **MARCO TEÓRICO:** | | | | | |
| * **Arquitectura cliente-servidor**   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. Llamadas servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor es quien da la respuesta.  Describe la configuración detallada de cada servidor e incluye: Hardware que se necesita para cada servidor. Sistema operativo que se necesita para cada servidor.     * **Lenguaje Python**   Pyhton es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo.  Es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites.   * **Máquina virtual**   Es un software que simula un sistema de computación y puede ejecutar programas como si fuese una  computadora real. Este software en un principio fue definido como “un duplicado eficiente y aislado de una máquina física”.  Tiene su propio disco duro, memoria, tarjeta gráfica y demás componentes de hardware, aunque todos ellos son virtuales.   * **Herramientas de gestión**   Son una serie de herramientas que sirven para mejorar el funcionamiento del sistema y proteger la información. Se dividen en tres herramientas que son:   1. De aplicación. 2. De configuración.   De optimización. | | | | | |
| **MATERIALES:** | | | | | |
| **EQUIPOS:**   * PC | | | | | |
| **PROCEDIMIENTOS Y ACTIVIDADES REALIZADAS EN LA PRÁCTICA:** | | | | | |

|  |
| --- |
| 1. **Implementar un modelo de paso de mensajes en una arquitectura cliente-servidor (se sugiere usar los siguientes lenguajes: C, C++, Python)**   **Instalación de Python en Ubuntu.**        **Instalación de Visual Studio Code en Ubuntu**    **Visual Studio Code Instalado en máquina virtual Ubuntu**    **Creación de código en Python para el Servidor y Cliente**    **Seridor**    **Client**      Ejecutamos el servidor con la dirección IP e ingresamos la dirección por el puerto que se va ingresar en este caso por el puerto 9099    Ejecutamos al cliente con la dirección IP del Servidor y el puerto por donde se va ingresar en este caso 9099    Desde el cliente escribimos  ESTE ES UN MENSAJE ES ENVIADO DESDE EL CLIENTE WINDOWS AL SERVIDOR UBUNTU    Respondemos desde el Servidor con el siguiente mensaje  MENSAJE RECIBIDO DESDE EL SERVIDOR UBUNTU    Y verificamos que en el cliente Windows que el servidor Ubuntu respondió     1. **c) Utilizar una herramienta de gestión como Putty para acceder al servidor y ejecutar. Ingresamos a la herramienta Putty, con la dirección IP de nuestro servidor, en este caso el servidor esta desde Ubuntu, por los cual se ingresa las credenciales de Ubuntu.**   Una vez ingresado, nos dirigimos al directorio en donde esto código de servidor para ejecutar desde la consola de Putty.    Una vez ingresado al Servidor ingresamos la dirección IP de nuestro servidor y el puerto 9099.    Iniciamos de la misma manera el cliente en Windows con la dirección IP del servidor y el mismo puerto para que exista conexión y vemos que se estableció la conexión.    Mandamos el primer mensaje desde el cliente al servidor.    Servidor abierto desde Putty para el paso de mensajes |
|  |
| **RESULTADOS OBTENIDOS:** |
|  |
| **CONCLUSIONES:** |
| 1. La red cliente servidor ha sido una red de comunicación en la cual el cliente está conectado con el servidor y se puede centralizar los diversos recursos y aplicaciones con que se cuenta cada vez que son solicitados. 2. Con este laboratorio se puede concluir que establecer una conexión para el paso de mensajes del servidor cliente, no es tan complicado, se ha logrado realizar las conexiones solicitadas sin inconvenientes y recibiendo los mensajes tanto al servidor, como al cliente. 3. Se realizaron pruebas de conectividad entre las máquinas virtuales y la máquina anfitrión, antes de realizar cualquier otra actividad, y todo fue exitoso. 4. Esta tecnología nos proporciona el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro servicio del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. |
| **RECOMENDACIONES:** |
| 1. Es recomendable antes de iniciar la práctica, consultar en fuentes de información acerca de la arquitectura cliente – servidor, en máquinas virtuales con la implementación de un lenguaje de programación ya sea Python, C, C++, entre otros. 2. Es necesario conocer acerca de los sockets que se han implementado. 3. Para la programación de los códigos del cliente como del servidor es importante conocer las librerías que permiten crear la conexión para el paso de mensajes, mediante las direcciones IP y puerto por donde se va ingresar para la respectiva conexión. 4. Antes de iniciar la práctica, es recomendable revisar las configuraciones de red de la máquina virtual ya que, si se tiene conectado mediante, una dirección IP estática, el cliente no va responder a dicha dirección asignada, por lo que es recomendable tener en DHCP para así establecer conexión a la misma red. |
| **ELABORADO POR:** |
|  |