



Módulo Teórico-Práctico

Entrega

Módulo
Sistemas Distribuidos
Nivel académico
Profesional
Tipo de entrega
<p>Cada documento debe contener la introducción, el marco teórico con reseñas de fuentes investigadas como ensayos, artículos científicos, libros de consulta, el desarrollo de lo correspondiente a la entrega como talleres, ejercicios y demás ítems solicitados en la respectiva entrega. Debe tener unas conclusiones y la bibliografía consultada.</p>

INSTRUCCIONES PARA REALIZAR LA ENTREGA

Nota

Tenga en cuenta que el tutor le indicará qué herramienta requiere y qué estrategia deberá desarrollar para evidenciar su participación individual en un trabajo colaborativo.



El estudiante debe ir elaborando el documento de las entregas usando normas APA. El documento de la entrega no debe tener errores de ortografía ni de semántica.

El documento de cada entrega es independiente, donde cada entrega contenga todos los ítems de presentación de un proyecto según la semana que corresponda con la entrega, conforme a las instrucciones para la elaboración del proyecto. Son tres entregas que van en documentos aparte y cada entrega se describe a continuación:

ENTREGA PREVIA 1

SEMANA 3

Primera entrega:

1. Para realizar esta entrega, es indispensable que realice las lecturas de los escenarios 1, 2 y 3 sobre fundamentos de los sistemas distribuidos y luego haga un repaso general del sistema operacional Linux.
2. Instale VirtualBox 6.x (Software libre) de 64 bits para Windows si su computador está con sistema operacional Windows o la versión correspondiente para el sistema operativo de su máquina.
3. Sobre esta plataforma, VirtualBox (o VMware o Quemus, etc.), se debe montar un sistema operacional Linux, Debian, Ubuntu o cualquier otro sistema operativo de la familia Unix, que permita instalar, configurar y poner en marcha un sistema de archivos en red **NFSv4**.
4. En la máquina virtual (VM) de 50 GB de disco, un Core y 1024 o 2048 GB de Memoria RAM (si dispone de buena memoria en su máquina real) instale el sistema operacional e identifique esta máquina como NFSv4-Server. (**instale el Java y el Eclipse para la versión de Linux instalada**)
5. Diseñe una **red LAN** (que conste de dos computadores) con sus respectivas IPs, máscaras, gateway y DNS.
6. Actualizar el sistema operacional Linux con el comando respectivo según su versión de Linux y luego exportarla para tener una **copia de respaldo**.
7. Clone la máquina virtual con una nueva *macaddress* llamada NFSv4-Cliente y asigne a cada una de las máquinas la IP respectiva del diseño de red LAN.
8. Haga los pings respectivos para verificar que las dos máquinas virtuales se conectan.
9. Proceda a la implementación de la instalación del NFSv4 en el servidor y vaya documentando el proceso en el documento maestro.

10. Instale el NFSv4 Cliente en la otra máquina virtual y vaya documentando el proceso en el documento maestro.
11. Para probar, cree un archivo desde la máquina cliente y verifique que haya quedado creada en la máquina servidor.
12. Para sustentar, haga un video que pruebe su funcionamiento creando un archivo desde el cliente, luego verifique que haya quedado en el servidor, después modifique el contenido del archivo desde la máquina cliente y muestre que los cambios quedaron hechos en el archivo, en la máquina servidor.
13. Elabore el documento con la documentación resumida del estado del arte del NFSv4, donde explique lo que es NFS y las ventajas y desventajas del NFS vs NFSv4. Explique los pasos de la instalación de server y el cliente NFSv4 del taller.
14. Agregue el diseño de su red LAN y la trazabilidad con las pruebas del proceso descrito anteriormente.
15. Construya un video donde muestre el funcionamiento del proyecto.
16. Publique el video en YouTube y pruebe su funcionamiento para estar seguro de que el video quedó bien cargado. Adjunte el enlace del video a la entrega.
17. Cargue la entrega completa a la plataforma dentro de los plazos establecidos.

ENTREGA PREVIA 2

SEMANA 5

Utilice las mismas máquinas virtuales de los pasos 6 y 7 de la primera entrega del escenario 3, que ya tienen eclipse, y proceda a desarrollar los pasos de la segunda entrega del escenario 5, de acuerdo con las siguientes indicaciones:

1. Monte una base de datos relacional en la máquina servidor solamente. La base de datos puede ser MariaDB, HsqlDB, Mysql, Postgres, Oracle Xpress, etc., o la que usted prefiera.

2. Cree dos tablas, colóqueles las llaves y relaciónelas, una de ciudades y una de personas, donde las personas vivan en una ciudad cualquiera. La tabla de ciudades debe tener los campos siguientes:

(**ciud_id, ciud_nombre** donde la llave es **ciud_id**). La tabla de personas debe tener los siguientes Campos (**dir_tel, dir_tipo_tel, dir_nombre, dir_direccion, dir_ciud_id**), donde **dir_tel** sea la llave primaria y **dir_ciud_id** sea la llave foránea.
3. Cargue cinco datos de prueba en cada tabla.
4. Elabore un programa socket Server que recoja las peticiones de un programa cliente, el cual le va a pedir el nombre, la dirección y la ciudad donde vive una persona que tiene un número de teléfono dado.
5. Elabore un programa socket Cliente que solicite por consola el número de teléfono de una persona y lo envíe al server.
6. El server entra a la base de datos, extrae la información si la encuentra y le devuelve al cliente los siguientes datos: el número de teléfono, el nombre de la persona, la dirección y el nombre de la ciudad si encuentra la información, de lo contrario devuelva un mensaje que diga: Persona dueña de ese número telefónico no existe.
7. El cliente recibe el mensaje, publica el resultado y repite la consulta por lo menos dos veces, luego cierra la transacción, pero el socket servidor queda abierto esperando por si otro cliente solicita servicio.
8. Elabore el documento con la documentación resumida del estado del arte de lo que es un socket y la explicación de las líneas de código de los sockets construidos para el taller. Agregue el diseño de su aplicación y la trazabilidad con las pruebas del proceso descrito anteriormente. Adjunte el código fuente a la entrega.
9. Construya un video donde explique el código y muestre el funcionamiento.
10. Suba el video a YouTube y pruebe su funcionamiento. Adjunte el enlace del video a la entrega.
11. Cargue la entrega completa a la plataforma dentro de los plazos establecidos.
12. Durante el proceso de instalación habilite la aplicación de eclipse y el Java.

ENTREGA FINAL

SEMANA 7

Haga un diseño de una red LAN interna y una red pública NAT para enlazar tres máquinas virtuales. Este taller se puede hacer en Google HPC, pero requiere pago, por tal razón se sugiere usar software libre como RocksHPC, OpenHPC, Warewulf HPC, etc., (se recomienda usar RocksHPC la versión 6.1.1 o 6.2).

1. Monte una máquina virtual máster con VirtualBox de 50 GB, un Core y 1024 de RAM. Configure una red LAN interna por el dispositivo 1 llamada numeroPI y una red Pública por el dispositivo 2, como red NAT.
2. Cree una segunda máquina virtual de 25 GB, 1 Core y 1024 bytes de RAM con el nombre “nodo-0” Configure una red LAN interna llamada numeroPI por el dispositivo de red 1.
3. Cree una tercera máquina virtual de 25 GB, 1 Core y 1024 bytes de RAM con el nombre “nodo-1” Configure una red LAN interna llamada numeroPI por el dispositivo de red 1.
4. Proceda a montar el sistema operacional distribuido hpc.
 - a. En el evento que decida usar Rocks, descargue el Rocks 6.1.1 o 6.2 de la página oficial de Rockscluster.org y proceda a su instalación. <https://www.rocksclusters.org/downloads.html>
 - b. Instale el Rocks servidor en el Máster y vaya documentando el proceso.
 - c. Instale el Rocks cliente en los nodos y vaya documentado el proceso.
 - d. Aprenda a usar los comandos básicos de Rocks, tales como ssh, rocks list, qstat, qsub, rocks run host, ganglia, etc. Ver manual en: http://digital.csic.es/bitstream/10261/36977/1/ContrerasB_MatDidact_Programac_rocks.pdf
 - e. Escriba el programa “holamundo.c” que se muestra a continuación, compílelo y córralo usando comandos de compilación MPI y ejecución con MPI explicados en el manual y en las respectivas lecturas. Luego adjunte las evidencias al documento de la segunda entrega. (Solicite asistencia al tutor por el correo y en los encuentros).

```

#include <stdio.h>
#include "mpi.h"
int main(int argc, char *argv[ ])
{
    int rank, size;
    MPI_Init(&argc, &argv);

    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    printf("Soy el core nro. %d de %d\n", rank, size);

    MPI_Finalize();
    return 0;
}

```

5. Como ya aprendió a montar el sistema HPC, ahora monte la solución de la entrega 7 con el siguiente ejercicio: Haciendo uso de la librería MPI realice un programa que halle el número π con el método de Leibniz haciendo 9×10^9 (9 por 10 elevado a la 9) iteraciones, se debe calcular el tiempo de ejecución y observar el comportamiento con 1, 2 y 3 procesadores.
6. Elabore el documento con la documentación resumida del estado del proceso del programa en paralelo para hallar π , la documentación de la ejecución del programa y el análisis de los resultados.
7. Construya un video donde explique el código y muestre el funcionamiento.
8. Suba el video a YouTube y pruebe su funcionamiento. Adjunte el enlace del video a la entrega.
9. Cargue la entrega completa a la plataforma dentro de los plazos establecidos.

Algunos enlaces sugeridos:

- » <https://www.javenda.me/2021/02/05/lenguajes-y-frameworks-para-la-programacion-paralela/#:~:text=Cactu%3A%20es%20un%20framework%20de,aplicaciones%20en%20clusters%20o%20superordenadores.>

- » <https://www.aspsys.com/solutions/hpc-applications/mpi-applications/>
- » <http://recursostic.educacion.es/observatorio/web/gl/software/software-general/733-nfs-sistema-de-archivos-de-red>
- » <https://www.aspsys.com/solutions/hpc-applications/mpi-applications/>
- » <https://warewulf.org/>

Criterios de evaluación

Entrega 1:

En la evaluación de las entregas se tiene en cuenta que los estudiantes trabajen en grupo, usen normas APA, no tengan errores de ortografía, no haya errores de redacción y el objetivo de la entrega se haya logrado.

Entrega 2:

El tutor evalúa que todos los ítems de la entrega se hayan desarrollado, pero el video tiene un peso de 40 % del puntaje de la entrega y el otro 60 % lo constituye el documento en normas Apa.

El video debe ser claro, donde se evidencie que fue desarrollado por el equipo de trabajo.

Entrega 3:

Revisar que el proyecto no haya sido bajado de Internet ni plagiado.

Revisar que las recomendaciones de la entrega anterior se hayan tenido en cuenta en la siguiente entrega.

Revisar que el documento no tenga errores de semántica ni ortografía.