



CI-0123 Proyecto integrador de redes y sistemas operativos

II Ciclo 2020. Grupo 2

Enunciado del proyecto programado *Minador de datos para Covid-19*

Fecha de entrega final: 04 de diciembre del 2020

Objetivo general

- Plasmar conceptos de las áreas de redes y sistemas operativos para programar un servicio que será atendido de manera redundante por varios servidores que pueden correr en máquinas distintas.

Objetivos específicos

- Crear un **cliente** web que obtenga datos sobre casos de Covid-19 por regiones determinadas, usando el protocolo HTTP.
- Crear un servicio de red básico: **servidor web de datos** regional para casos de Covid-19.
 - Se utilizará el protocolo HTTP básico en versión 1.1
 - Es posible utilizar un navegador web (Chrome, Firefox, etc.) puede ver los datos del servidor.
- Crear un **servidor intermedio** (proxy) para consulta agregada de casos de Covid-19.
 - Este servidor proxy será capaz de construir un “mapa” con los **servidores de datos** y determinar las regiones de cada uno, para atender las consultas de los clientes.
- Establecer esquemas de distribución de datos en distintos servidores.
 - Cada **servidor de datos** tendrá datos regionales de Covid-19. Inicialmente cada grupo se hará cargo de una región en específico.
- Definir protocolos de comunicación entre los nodos que componen el sistema.
 - Entre los **servidores de datos** y los **servidores intermedios** proxy.
 - Agregar un **servidor de datos** al **servidor intermedio**.
 - Consultas de datos al **servidor intermedio**.
 - Entre **servidores intermedios**.
 - Cualquier otro que los equipos de trabajo consideren pertinente.
- Registrar los eventos que se llevan a cabo en los servidores por medio de bitácoras que puedan ser consultadas en línea.
- Establecer los esquemas de comunicación.

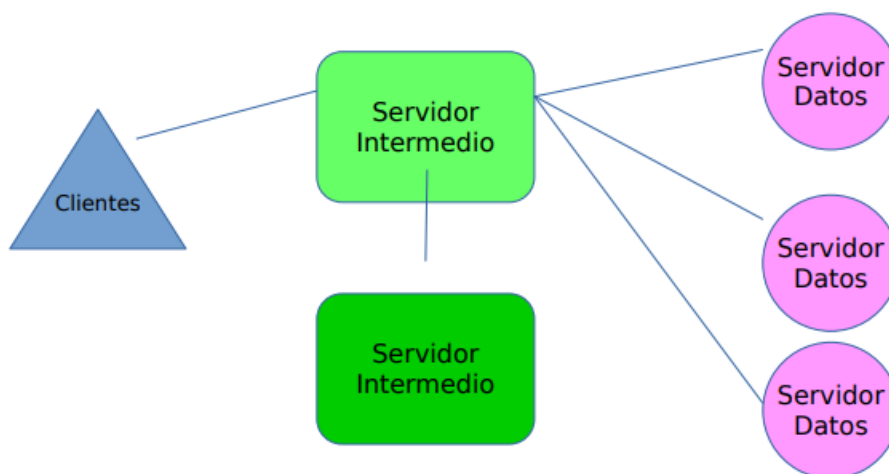


- Entre los **clientes** y los **servidores intermedios** se hará por medio de una red (pública) y se utilizará el puerto 80, inicialmente se hará entre los **clientes** y los **servidores de datos** directamente
- Entre los **servidores intermedios** y los **servidores de datos** se hace por medio de una red distinta a la primera (privada), se puede utilizar el puerto 80, pero este puede cambiar.

Descripción

Para el desarrollo de este proyecto se deberá programar distintos tipos de programas que se podrán integrar por medio de una red de comunicaciones. Específicamente debe construir programas para:

- Servidores intermedios
- Servidores de datos
- Clientes minadores



Esquema 1: Interacción de componentes



Clientes (primera etapa)

Los **clientes** serán capaces de obtener información desde los **servidores de datos**, en primera instancia y posteriormente desde los **servidores intermedios**, para ello deberán hacer uso del protocolo HTTP para el intercambio de mensajes.

Los **clientes** solicitarán datos sobre regiones específicas primero, por ejemplo, Costa Rica y desplegarán datos generales del país. Para ello, podrán hacer uso de los recursos expuestos en la sección “Fuentes de Información”. Los **clientes** deben ser capaces de atender consultas sobre regiones más grandes, por ejemplo, **Latinoamérica**, desplegar el resultado de cada país de esa región y el total de datos. El detalle de las regiones y sus miembros podrá ser consultado por medio de esta página web “<https://os.ecci.ucr.ac.cr/covid>”.

Servidores de datos (segunda etapa)

Estos programas deben ser capaces de atender solicitudes que utilizan el protocolo HTTP y devolver la información solicitada por los clientes. Estos programas deben obtener de la solicitud HTTP el recurso solicitado y entregarlo al cliente respectivo. Los **servidores de datos** tendrán información de regiones **específicas**, por ejemplo, para un continente, Costa Rica, México, Canadá, etc. El cambio de los datos se realiza a diario en el servidor provisto, por lo que es recomendable que estos servidores almacenen la información localmente en lugar de buscarla por Internet cada vez. Cada equipo debe planear su estrategia para mantener la información localmente.

Los **servidores de datos** deben crear otros trabajadores (hilos o procesos) para realizar la atención de las solicitudes de los clientes. El proceso principal no debe realizar esa atención de solicitudes.

Los **servidores de datos** no conocen la existencia de los otros servidores similares. Pero si deben ser capaces de unirse a los **servidores intermedios**. Los equipos de trabajo deben construir un protocolo, a nivel de toda la clase de CI0123 Grupo 02, para que los nuevos servidores de datos se unan a los **servidores intermedios** existentes. Esto significa que los **servidores de datos** de un grupo de trabajo de esta clase puedan unirse y funcionar adecuadamente con cualquier **servidor intermedio** de otro grupo de trabajo.

Adicionalmente, su **servidor de datos** debe permitir el uso de algunos de los clientes ya existentes para acceder a la información, por ejemplo: wget, curl, navegador web, etc. Similar a los ejemplos que se desarrollaron en la clase de creación de **Sockets**.

Durante esta etapa y para realizar pruebas, es permitido que los clientes conecten directamente con los **servidores de datos**. Sin embargo, para la entrega final, los clientes solo podrán conectarse a través de los **servidores intermedios**, ya que como se indicó anteriormente, estarán en redes lógicamente separadas.

También, debe ser posible acceder a los **servidores de datos** desde distintos tipos de dispositivos, por ejemplo, un teléfono móvil a través de su navegador, tableta, computadora, etc.

Clientes en NachOS (tercera etapa)

Cada grupo de trabajo debe implementar una versión del **cliente** que funcione como un programa de usuario del sistema operativo NachOS, para lo cual deben programar, entre otros aspectos, los llamados al sistema, el manejo de hilos, el manejo de la memoria, la memoria virtual y otros detalles requeridos en este ambiente.



En el curso de Sistemas Operativos se desarrollarán todos los elementos necesarios para completar esta etapa del proyecto. Este cliente debe funcionar de manera similar al presentado en la primera etapa.

Servidores intermedios (cuarta etapa)

Estos procesos van a integrar a los **servidores de datos** disponibles y comenzar a atender solicitudes de los **clientes**. Al inicio, es posible que cada grupo establezca un orden de ejecución de los programas, pero como requisito de este proyecto, los programas pueden ejecutarse en cualquier orden, al menos entre **servidores de datos** y **servidores intermedios**.

Cada **servidor intermedio** debe tener un “mapa” de las regiones atendidas por los **servidores de datos** que tiene conectados, de manera que pueda enrutar las solicitudes de los clientes y entregar las respuestas correspondientes. Si no es posible completar una solicitud, se deben utilizar elementos del protocolo HTTP para indicar los errores al cliente (códigos de error).

Los **servidores intermedios** están conectados con los clientes por medio de una red (pública, normalmente 10.1.x.x/16 de la ECCI) y con los **servidores de datos** por medio de una red distinta (privada, normalmente 172.16.123.x/24). Como fue indicado anteriormente, en esta etapa los clientes no tendrán acceso a la red de los **servidores de datos**, por lo que las consultas deberán dirigirse a través de los servidores intermedios.

Evaluación

Etapas	Actividad	Lenguaje	Porcentaje
1	Clientes	C++, C	30%
2	Servidores de datos	C++, C	25%
3	Clientes	NachOS	15%
4	Servidores intermedios	C++, C	30%
		Total	100%
Final	Nota grupal Nota individual	Porcentaje asociado al curso CI0123	80%

Tabla 1: Porcentajes de evaluación

Entregables

- Primera entrega (30%)
 - Elementos a entregar
 - Cliente
 - Casos de prueba
 - Definición de protocolo de comunicación para adicionar servidores de datos a servidores intermedios
 - Documento de entrega (plantilla)
 - Valoración del trabajo en equipo e individual



- Presentación de resultados a los evaluadores
- Fecha y hora máxima: 11 de septiembre del 2020 a las 23:59 horas.
- Segunda entrega (25%)
 - Elementos a entregar
 - Servidores de datos con información regional específica
 - Casos de pruebas
 - Definición **grupal** del protocolo de agregado de servidores
 - Documento de entrega
 - Valoración del trabajo en equipo e individual
 - Presentación de resultados a los evaluadores
 - De ser necesario, completar elementos de la etapa anterior
 - Fecha y hora máxima: 09 de octubre del 2020 a las 23:59 horas.
- Tercera entrega
 - Elementos a entregar (15%)
 - Cliente en NachOS
 - Casos de prueba
 - Documento de entrega
 - Valoración del trabajo en equipo e individual
 - Completar elementos incompletos de etapas anteriores
 - Presentación de resultados a los evaluadores
 - Fecha y hora máxima: 06 de noviembre del 2020 a las 23:59 horas.
- Cuarta entrega (**final**) (25%)
 - Elementos a entregar
 - Integración de elementos, agregado de servidores de datos
 - Esquema de atención de solicitudes de clientes
 - Pruebas de integración (propias y con otros grupos)
 - Documento de entrega
 - Valoración del trabajo en equipo e individuales
 - Completar elementos incompletos de etapas anteriores
 - Presentación del proyecto 2020/Dic/08 de 1 p.m. a 7 p.m.
 - Fecha y hora máxima: 03 de diciembre del 2020 a las 23:59 horas.

Fuentes de información

Para obtener la información de los casos de coronavirus, puede hacer uso de las siguientes opciones:

- Casos locales (Costa Rica)



- Servicio web REST disponible en la dirección https://services.arcgis.com/LjCtRQt1uf8M6LGR/arcgis/rest/services/CASOS_COVID_19_VISTA2/FeatureServer/2/query?where=1%3D1&outFields=*&outSR=4326&f=js on
- Casos regionales/globales:
 - Sitio web con datos globales de coronavirus disponible en la dirección <https://www.worldometers.info/coronavirus/>
 - Servicio web REST disponible en la dirección <https://api.covid19api.com/total/country/costa-rica> Además, para obtener la lista completa de países con los códigos a utilizar en las consultas se puede hacer uso de los datos disponibles en la dirección <https://api.covid19api.com/countries>

Es importante considerar que para el intercambio de mensajes entre clientes y servidores se debe hacer uso del protocolo HTTP. Además, los servicios web REST antes listados devuelven en todos los casos, los datos en formato JSON.

Requerimientos adicionales

- Crear un repositorio privado para el control de versiones en la plataforma GitUCR (<https://git.ucr.ac.cr/>). Deben invitar como colaboradores a los docentes y al asistente a este repositorio.
- Debe existir una manera automática de generar los programas ejecutables, por ejemplo mediante el uso de “*Makefiles*” (a través del comando “make <<nombre_archivo>>”). También deben crear un documento, “*Read.Me*” con las instrucciones para construir y ejecutar los programas.
- El código fuente de los programas deben estar documentado adecuadamente (documentación interna).
- Para la evaluación individual del proyecto se tomará en cuenta los **aportes individuales** de cada miembro del equipo de trabajo en el repositorio de control de versiones del proyecto.
- El desarrollo debe hacerse para ambientes Linux. Todos los programas deben correr en cualquiera de los equipos del laboratorio 3-5 de la ECCI, por lo que en las instrucciones deben indicar los paquetes requeridos para lograr su ejecución.
- Cada entrega incluye una presentación de los resultados, puede ser a toda la clase, pero al menos se hará para los profesores y el asistente. Estas presentaciones, se harán de manera virtual.
- Pueden utilizar varios lenguajes de programación, pero se requiere la aprobación por escrito de ambos profesores para realizarlo. Si es aprobado, entonces los distintos componentes podrán programarse en diferentes lenguajes, excepto la versión del servidor en NachOS que debe ser en C++. El grupo que utilice distintos lenguajes es responsable de lograr que sus componentes puedan ser integrados entre sí. Además, deben considerar que la integración con componentes de otros equipos de trabajo pueda ocurrir con programas desarrollados en otros lenguajes.