

## Redes de computadores I (ICI 224)

### Tarea 2: Implementar servidor web TCP sencillo.



Integrantes: -Carolina Echiburú.

-Sebastián Serey.

-Franco Calabriano.

Profesor: Gabriel Astudillo.

Fecha: 13/12/2018.

## Índice:

<b>1.Introducción:</b>	3
<b>2.Objetivos:</b>	3
<b>3.Trabajo a realizar:</b>	3
<b>3.1Correo Electrónico:</b>	3
<b>3.2Repositorio:</b>	4
<b>4.Vista Lógica:</b>	4

## 1.Introducción:

Alumnos de la Escuela de Ingeniería Civil Informática de la Universidad de Valparaíso deberán crear un software en lenguaje de programación C++ 2014, para lograr implementar un Servidor web TCP sencillo.

Un servidor web es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo.

TCP (Protocolo de Control de Transmisión) es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación, posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él, (es decir, el protocolo IP).

## 2.Objetivos:

- Conocer y aplicar el API de Networking en C++.
- Implementar servidor web TCP sencillo.

## 3. Trabajo a realizar:

El software crear se debe implementar en C++14, en Sistema Operativo Linux y debe compilar sin warnings en los computadores de los laboratorios. Se pueden utilizar todas las facilidades de este lenguaje para resolver el problema como la *Standard Library Template* y *boost c++ libraries*. Bajo ningún punto de vista, debe utilizar librerías que ya implementen el problema.

El servidor web iterativo simple a crear, debe ser capaz de servir páginas web sencillas, que contengan sólo texto HTML. Debe tener parametrizados en un archivo de configuración JSON, la dirección IP, puerto TCP, el directorio raíz donde se ubican las páginas web, entre otros. El servidor debe estar programado de tal manera que si se coloca una nueva página en el directorio especificado por la variable **root\_dir**, y el usuario la pide a través de su navegador, el sistema se la debe transmitir. Si no la encuentra, debe transmitir la página que está referenciada en la variable **notFoundFile** en el archivo de configuración. Para leer el archivo JSON, se recomienda utilizar la librería disponible en el archivo "json.hpp".

El entregable final de todo el trabajo realizado será un correo electrónico dirigido al ayudante de la asignatura con la información de los integrantes del grupo además del enlace del repositorio. Este repositorio debe ser creado en GitHub, y debe ser clonado por el ayudante a la hora de corrección.

### 3.1 Correo Electrónico:

El correo electrónico debe ser enviado al ayudante de la asignatura (correo: **pablo.olivares@alumnos.uv.cl**), el día Jueves 13/12/2018 antes de las 23:55 horas. Además, debe tener la siguiente información:

-Asunto: [Tarea02\_RedesNueva]Echiburu-Serey-Calabriano.

-Integrantes:

Carolina Viviana Echiburú Cueto – [carolina.echiburu@alumnos.uv.cl](mailto:carolina.echiburu@alumnos.uv.cl)

Sebastián Tadeo Serey Carrillo – [sebastian.serey@alumnos.uv.cl](mailto:sebastian.serey@alumnos.uv.cl)

Franco Enzo David Calabriano Garrido – [franco.calabriano@alumnos.uv.cl](mailto:franco.calabriano@alumnos.uv.cl)

-URL del repositorio: <https://github.com/carolinaechiburu/Tarea02-Echiburu-Serey-Calabriano>

### 3.2 Repositorio:

En el repositorio deben ir todos los archivos a utilizar para el servidor junto a su correcta documentación. La carpeta principal se llama Tarea02, luego sus debidas carpetas que contengan los archivos ejecutables y todos los que sean necesarios para su compilación.

También, hay una carpeta “Documentos” la que incluye el informe del trabajo realizado, junto a la documentación de la arquitectura del software (por lo menos la Vista Lógica y Vista de Procesos).

Dentro del repositorio, debe ir un archivo README.md, donde se especifican los integrantes del grupo, sus respectivos correos y una descripción del funcionamiento del software, desde como compilarse hasta ejemplos de uso.

### 4.Vista Lógica:

```
---->Documentos
----->INFORME-TAREA02.pdf
----->VISTA-DE-PROCESOS.txt
----->VISTA-LOGICA.txt
----> Tarea02
----->bin
----->config.json
----->servidor
----->src
----->includes
----->json.hpp
----->YASL.h
----->Makefile
----->servidorWeb.cc
----->www-data
----->INTERNETWORKING.HTML
----->IP.HTML
----->NODOS.HTML
----->www-error
```