

**Universidad de Costa Rica**  
**Facultad de Ingeniería**



**Escuela de Ciencias de la**  
**Computación e Informática**

**Curso CI-0123 PIRO**

**Proyecto Integrador**  
Entregable #3

**Profesores:**

Francisco Arroyo Mora  
José Antonio Brenes Carranza

**Asistente:**

Lester Cordero Murillo

**Estudiantes:**

(B70257) Ricardo Alfaro Víquez  
(B86875) Luis Eduardo Rojas Carrillo  
(B80986) Allan Barrantes Chaves  
(B67454) Mario Vargas Campos

**II Semestre, 2020**

# 1. Introducción

En este tercer entregable, se aumentó la complejidad del proyecto ya que se procedió a implementar un cliente al sistema operativo NachOS. Este sistema operativo tiene sus propios *system calls*, por lo que el uso de llamados al sistema de c y c++ no eran permitidos.

Durante esta entrega, se requiere de una implementación a un nivel mucho más bajo. Esto quiere decir que el desarrollador debe ser capaz de controlar el manejo de procesos en aspectos como sincronización, manejo de memoria, llamados al sistema, etc. La implementación de un cliente en un sistema operativo como NachOS requiere de dichos aspectos para su correcto desarrollo y funcionamiento.

Este cliente era capaz de conectarse a un servidor intermedio y por lo tanto realizar solicitudes. Al igual que en la entrega anterior, dicho cliente solicitaba datos tanto de países como de cantones de Costa Rica. Dichas solicitudes se realizaban a través del terminal.

Finalmente, la implementación para esta entrega, requiere que el desarrollador sea capaz de implementar un servidor intermedio funcional tanto para un cliente normal como para un cliente de NachOS. Este servidor debe atender solicitudes de ambos clientes, de manera eficiente y correcta, ya que para la última entrega, este servidor intermedio debe proporcionarle los datos al cliente y conectarse mediante un protocolo PIRO con múltiples servidores de datos.

## **2. Objetivo General**

A continuación se establece el objetivo general del proyecto:

Implementar un programa que permita conectarse a un servidor con el fin de obtener datos actualizados sobre el COVID-19 de países y cantones de Costa Rica, y mostrarlos al usuario desde un browser o desde la terminal.

## **3. Objetivos Específicos**

A continuación se establecen los objetivos específicos para esta etapa del proyecto:

- Construir un cliente funcional utilizando NachOS.
- Realizar una consulta a un servidor mediante un llamado de socket utilizando el protocolo HTTP.
- Enviar una respuesta desde el servidor al cliente del país o cantón que solicite en un browser o terminal de cliente nachOS.

## 4. Descripción de la Solución

Para este entregable lo principal fue implementar lo creado en el entregable 2 con la creación de un cliente en nachOS. El cliente que se creó en nachOS es similar al utilizado al de la entrega 2 para consola, sin embargo en este caso fue necesario sustituir los llamados a la librería **string** por variables de tipo **char \*** y además, se utilizaron los llamados a sistema **Read** y **Write** de nachOS para poder obtener datos de entrada y para imprimir resultados al usuario en la consola. De igual manera se utilizaron los llamados a sistema de socket de nachOS para poder realizar la conexión del cliente al puerto 8080 y así la conexión con el servidor.

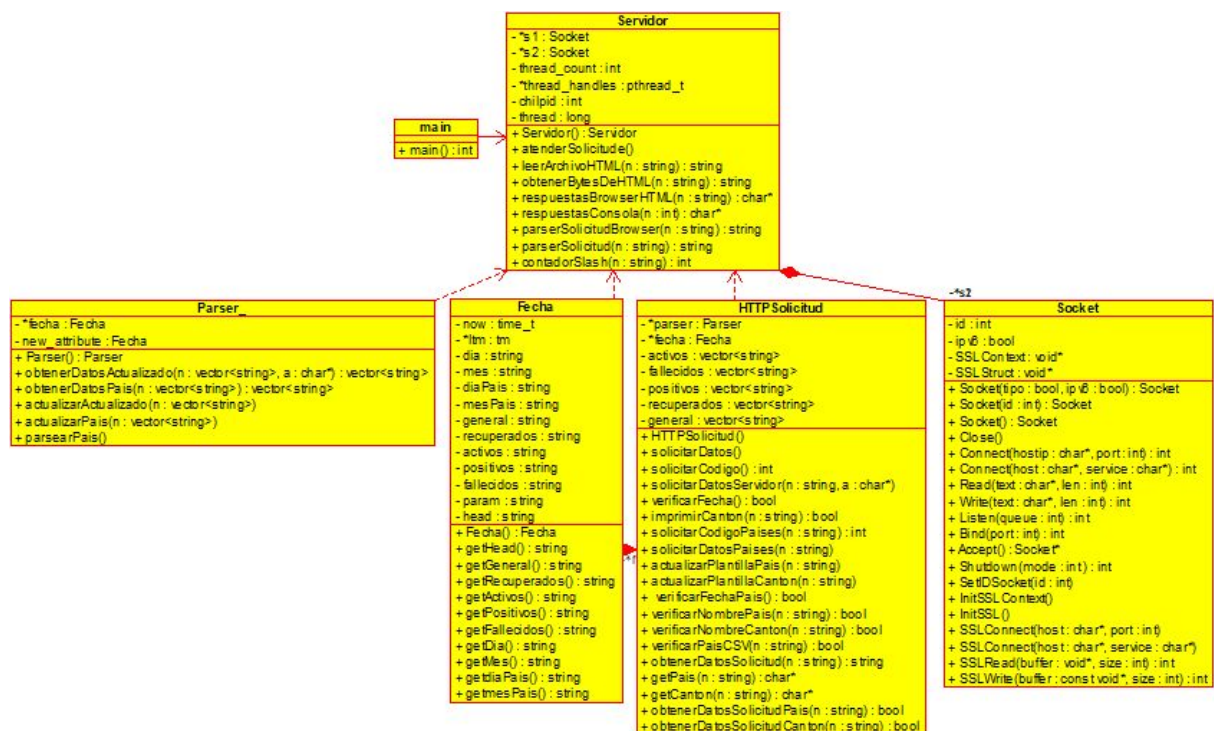
Con respecto a aspectos visuales, se realizaron un servidor y cliente más verbosos para que sea más amigable con el usuario desde consola y se pueda ver un mejor flujo del programa. Se utilizó para desplegar los resultados en consola una tabla para poder apreciar mejor cada dato, ya sea de país o de cantón, para esto es importante destacar que se despliega la tabla de países de manera vertical y la de cantones de manera horizontal. La decisión de desplegar las tablas de esa manera fue para evitar que en monitores pequeños la tabla de país se desfigurara por tener tantos datos y sea más fácil de entender para el usuario, en el caso de la tabla de país no existe el problema de países entonces se desplegó horizontal sin ningún problema.

Esta tabla implementada se tomó de código externo que se encuentra en el siguiente link: <https://github.com/seleznevae/libfort> y dentro del código del proyecto es específicamente los archivos que se llaman: **fort.h**, **fort.hpp**, **fort.c**

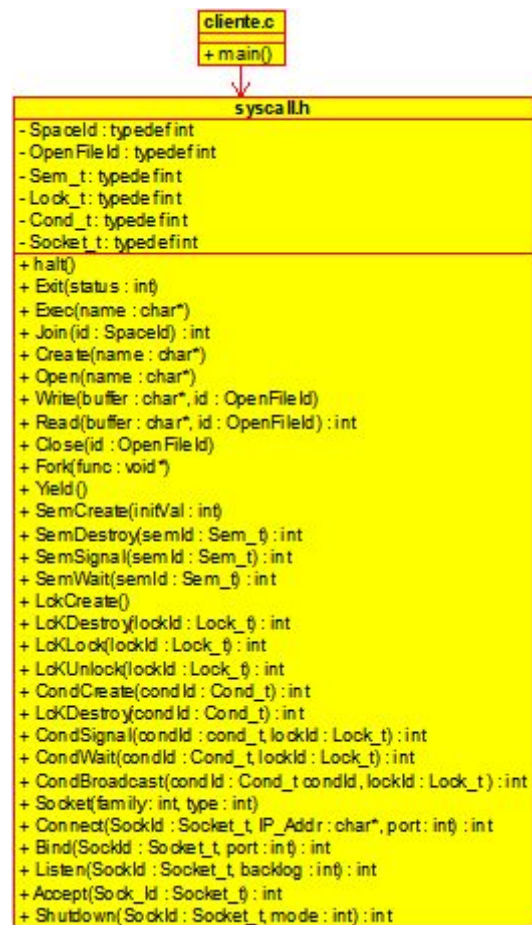
## 5. Esquema de la Solución

En el siguiente UML, se muestra la conformación de las clases del proyecto y cómo interactúan entre ellas. Además, se especifican las variables o los métodos propios de cada clase.

### Esquema del Servidor:



## Esquema del Cliente:



## 6. Arquitectura de la Solución

La siguiente tabla, muestra la arquitectura del proyecto. Se define la conformación de las clases, y la funcionalidad principal para cada una de ellas. Además, de librerías o frameworks que destacan en la implementación del código.

| Concepto         | Descripción   |
|------------------|---|
| Socket.h         | <p>Esta clase define los métodos y los atributos correspondientes a la clase Socket.cc.</p> <p>Librerías:<br/>-OpenSSL: Esta librería permite el uso de herramientas para el manejo de protocolos SSL/TLS.</p>  |
| Socket.cc        | <p>Esta clase efectúa la comunicación con el servidor. Mediante una solicitud, permite leer y escribir, esto con el fin de obtener datos provenientes de un servidor a partir de dicha solicitud.</p>   |
| HTTPSolicitud.h  | <p>Esta clase define los métodos y los atributos correspondientes a la clase HTTPSolicitud.cc.</p> <p>Librerías:<br/>-Fstream, iostream: Son librerías que permiten herramientas para el manejo de I/O dentro de cualquier programa.</p>  |
| HTTPSolicitud.cc | <p>Esta clase realiza un manejo de los request que se realizarán al servidor para solicitar los datos. Realiza un manejo de los archivos .csv, en los cuales se parsean los datos solicitados. Realiza un manejo de errores de los source code siempre que se soliciten los datos mediante un HTTP Request.</p>   |
| MainCliente.cc   | <p>Esta es la clase main del programa. Maneja los menús que se le desplegarán al usuario para la interacción con el mismo. Despliega los resultados tabulares que el usuario solicite.</p> <p>Librerías:<br/>-Fstream, iostream: Son librerías que permiten herramientas para el manejo de I/O dentro de cualquier programa.</p> <p>-Socket.h, HTTPRequest.h: Incluye ambas clases para el manejo del programa.</p> |
| Fecha.h          | <p>Esta clase define los métodos y los atributos correspondientes a la clase Fecha.cc.</p> <p>Librerías:<br/>-Fstream, iostream: Son librerías que permiten herramientas para el manejo de I/O dentro de cualquier programa.</p>  |

|                 |   |
|-----------------|---|
| Fecha.cc        | Esta clase realiza un manejo de los formatos de fecha para el resto de clases que requieran de su implementación.   |
| Parser.h        | Esta clase define los métodos y los atributos correspondientes a la clase Parser.cc.<br><br>Librerías:<br>-Fstream, vector, ctime, algorithm: Estas librerías permiten el uso de objetos vector, manejo de variables de tiempo y fecha, así como el manejo de archivos.   |
| Parser.cc       | -Esta clase realiza un manejo de los datos del proyecto para así realizar los parser necesarios.<br>-Modifica archivos .csv y plantillas HTML necesarias para el almacenamiento y visualización de los datos.   |
| Servidor.h      | Define los métodos y los atributos correspondientes de la clase Servidor.cc.<br><br>Librerías:<br>-Fstream, iostream: Son librerías que permiten herramientas para el manejo de I/O dentro de cualquier programa.<br>-Pthread: Es una librería que permite el manejo de hilos, indispensable para el manejo de sockets. |
| Servidor.cc     | Esta clase realiza un manejo del programa con una función de servidor de datos.<br><br>Ejecuta solicitudes de un servidor de datos y se los provee al cliente cuando este lo solicite.  |
| MainServidor.cc | Esta clase permite que el servidor se ejecute.  |
| syscall.h       | Define todos los llamados al sistema de NachOS.   |
| Cliente.c       | Clase controladora que maneja un cliente que se conecta al servidor utilizando los llamados al sistema definidos en syscall.h de NachOs.  |



## **7. Limitaciones de su Solución**

A pesar de los intentos, no se logró implementar el control del código de error 505, esto se debió a que, estaba generando problemas en la compilación al interactuar con las otras excepciones que, por cuestiones de manejo de tiempo, no se lograron corregir.

El manejo de países con guiones en sus guiones generaban un fallo al buscarlas, pues justamente se utilizó el guión medio (-) como delimitador para los espacios en las palabras ingresadas en cada consulta, por lo que esta eventualidad no fue observada hasta el momento de las pruebas.

## 8. Pruebas de funcionalidad

Describa los casos de prueba para esta etapa del proyecto y sus resultados esperados. Utilice tablas cuando requiera indicar que se necesita ejecutar una instrucción en terminal.

### Pruebas con el Terminal

#### ***Caso#1: Muestra los datos de un cantón solicitado.***

```
Elija alguna de las opciones del menu
1. Consultar pais
2. Consultar cantón
3. Salir del programa
2
Por favor digite el cantón a consultar
San-Ramon
Código  Provincia      Código  Cantón      Positivos  Activos  Recuperados  Fallecidos
2        Alajuela        202     San Ramón    865        378     480         7

Elija alguna de las opciones del menu
1. Consultar pais
2. Consultar cantón
3. Salir del programa
```

#### ***Caso#2: Muestra los datos generales de un país.***

```
ricardopricardo-VirtualBox:/media/sf_ShareUbuntu/GitLab/Piro/cl0123_proyectointegradorredessistemasoperativos_grupo0/src$ ./cliente
Elija alguna de las opciones del menu
1. Consultar pais
2. Consultar cantón
3. Salir del programa
1
Por favor digite el país a consultar
Costa-Rica

Pais          Total Casos  + Casos  Total Muertes  Muertes  Total Recup  Recuperados  Activos  Criticos  Casos/mill  Muertes/mill  Total tests  Tests/mill  Población
Costa Rica    86,853      +1,225   1,055          +15      52,327      +545        32,671   197      16,851     207          251,285     49,207     5,106,729

Elija alguna de las opciones del menu
1. Consultar pais
2. Consultar cantón
3. Salir del programa
```

**Caso#3: Muestra manejo de error, recurso no encontrado.**

```
Elija alguna de las opciones del menu
1. Consultar país
2. Consultar cantón
3. Salir del programa
1
Por favor digite el país a consultar
asdfasdf

Error! Recurso no encontrado

Elija alguna de las opciones del menu
1. Consultar país
2. Consultar cantón
3. Salir del programa
```

**Caso#4: Muestra manejo de error, solicitud inválida.**


```
Elija alguna de las opciones del menu
1. Consultar país
2. Consultar cantón
3. Salir del programa
2
Por favor digite el cantón a consultar

Error! Solicitud inválida

Elija alguna de las opciones del menu
1. Consultar país
2. Consultar cantón
3. Salir del programa
3
ricardo@ricardo-VirtualBox: /media/sf_ShareUbuntu/GitLab/Piro/cl0123_proyectointegradorredessistemasoperativos_grupoC/src$
```

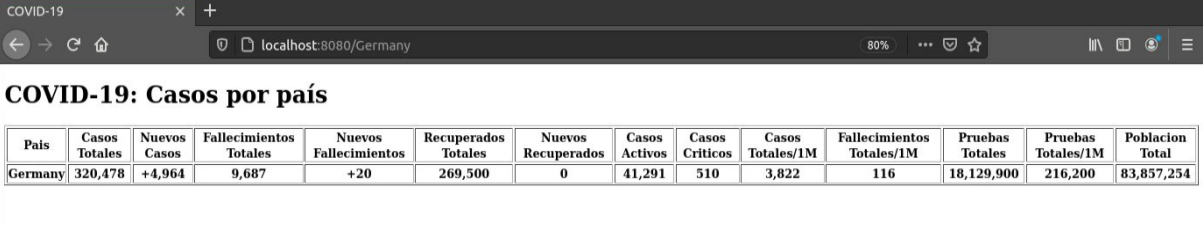
Pruebas con el Browser

**Caso#1: Muestra manejo de error, recurso no encontrado.**



The screenshot shows a web browser window with a single tab titled "COVID-19". The address bar displays "localhost:8080/Costa-Rica/asdas". Below the address bar, the page content shows the message "Error! Recurso no encontrado".

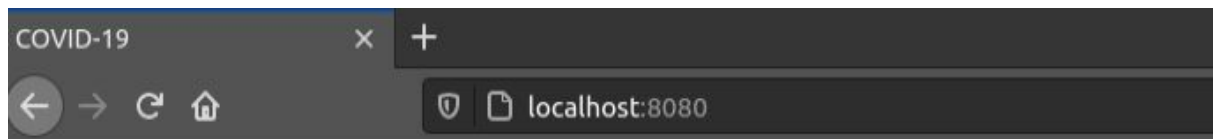
**Caso#2: Muestra los datos generales de un país.**



The screenshot shows a web browser with the address bar at localhost:8080/Germany. The page title is "COVID-19: Casos por país". Below the title is a table with 13 columns: País, Casos Totales, Nuevos Casos, Fallecimientos Totales, Nuevos Fallecimientos, Recuperados Totales, Nuevos Recuperados, Casos Activos, Casos Críticos, Casos Totales/1M, Fallecimientos Totales/1M, Pruebas Totales, Pruebas Totales/1M, and Poblacion Total. The data for Germany is as follows:

| País    | Casos Totales | Nuevos Casos | Fallecimientos Totales | Nuevos Fallecimientos | Recuperados Totales | Nuevos Recuperados | Casos Activos | Casos Críticos | Casos Totales/1M | Fallecimientos Totales/1M | Pruebas Totales | Pruebas Totales/1M | Poblacion Total |
|---------|---------------|--------------|------------------------|-----------------------|---------------------|--------------------|---------------|----------------|------------------|---------------------------|-----------------|--------------------|-----------------|
| Germany | 320,478       | +4,964       | 9,687                  | +20                   | 269,500             | 0                  | 41,291        | 510            | 3,822            | 116                       | 18,129,900      | 216,200            | 83,857,254      |

**Caso#3: Muestra manejo de error, solicitud inválida.**



**Error! Petición inválida**

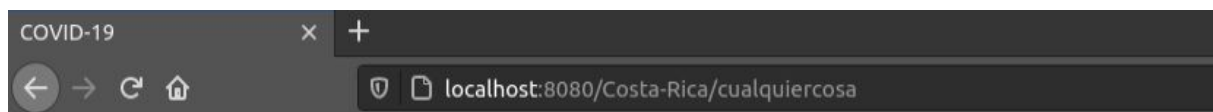
**Caso#4: Muestra los datos de un cantón solicitado**



The screenshot shows a web browser with the address bar at localhost:8080/Costa-Rica/Leon-Cortes-Castro. The page title is "COVID-19: Casos por cantones". Below the title is a table with 8 columns: Codigo Provincia, Provincia, Codigo Canton, Canton, Casos Positivos, Casos Activos, Casos Recuperados, and Fallecidos. The data for León Cortés Castro is as follows:

| Codigo Provincia | Provincia | Codigo Canton | Canton             | Casos Positivos | Casos Activos | Casos Recuperados | Fallecidos |
|------------------|-----------|---------------|--------------------|-----------------|---------------|-------------------|------------|
| 1                | San José  | 120           | León Cortés Castro | 41              | 6             | 35                | 0          |

**Caso#5: Muestra manejo de error, recurso no encontrado.**



**Error! Recurso no encontrado**

## Pruebas Terminal NachOS

### Caso#1: Muestra los datos de un cantón solicitado.

```
-----
Elija alguna de las opciones del menu
1. Consultar un País
2. Consultar un Cantón
3. Salir del Programa
-----

2

Por favor, digite el nombre del Cantón a consultar:

Sarchi
```

| Codigo | Provincia | Codigo | Canton | Positivos | Activos | Recuperados | Fallecidos |
|--------|-----------|--------|--------|-----------|---------|-------------|------------|
| 2      | Alajuela  | 212    | Sarchí | 259       | 203     | 56          | 0          |

### Caso#2: Muestra los datos generales de un país.

```
-----
Elija alguna de las opciones del menu
1. Consultar un País
2. Consultar un Cantón
3. Salir del Programa
-----

1

Por favor, digite el nombre del País a consultar:

Nigeria
```

| Pais          | Nigeria     |
|---------------|-------------|
| Total Casos   | 63,328      |
| + Casos       | 0           |
| Total Muertes | 1,155       |
| Muertes       | 0           |
| Total Recup   | 59,675      |
| Recuperados   | 0           |
| Activos       | 2,498       |
| Criticos      | 10          |
| Casos/mill    | 305         |
| Muertes/mill  | 6           |
| Total tests   | 668,729     |
| Tests/mill    | 3,217       |
| Poblacion     | 207,866,456 |

**Caso#3: Muestra manejo de error, recurso no encontrado con País.**

```
-----  
Elija alguna de las opciones del menu  
1. Consultar un País  
2. Consultar un Cantón  
3. Salir del Programa  
-----  
  
1  
  
Por favor, digite el nombre del País a consultar:  
  
aaaaaa  
  
Error! Recurso no encontrado
```

**Caso#4: Muestra manejo de error, solicitud inválida con País.**

```
-----  
Elija alguna de las opciones del menu  
1. Consultar un País  
2. Consultar un Cantón  
3. Salir del Programa  
-----  
  
1  
  
Por favor, digite el nombre del País a consultar:  
  
  
  
Error! Solicitud inválida
```

**Caso#5: Muestra manejo de error, recurso no encontrado con Cantón.**

```
-----  
Elija alguna de las opciones del menu  
1. Consultar un País  
2. Consultar un Cantón  
3. Salir del Programa  
-----  
  
2  
  
Por favor, digite el nombre del Cantón a consultar:  
  
aaaaa  
  
Error! Recurso no encontrado
```

**Caso#6: Muestra manejo de error, solicitud inválida con Cantón.**

```
-----  
Elija alguna de las opciones del menu  
1. Consultar un País  
2. Consultar un Cantón  
3. Salir del Programa  
-----
```

```
2
```

```
Por favor, digite el nombre del Cantón a consultar:
```

```
Error! Solicitud inválida
```

**Caso#7: Muestra el servidor intermedio ejecutándose.**

```
-----  
Hola! Soy el Servidor Intermedio!  
Me encuentro escuchando solicitudes atraves del puerto 8080  
  
Puedes realizar solicitudes usando el siguiente formato:  
Para solicitar un pais tienes que indicar:  
    localhost:8080/ + Nombre-Pais  
Para solicitar un canton tienes que indicar:  
    localhost:8080/ + Costa-Rica + / + Nombre-Canton  
-----
```

**Caso#7: Muestra el cliente de NachOs ejecutándose.**

```
-----  
Hola! Soy el Cliente!  
Puedo realizar solicitudes al Servidor Intermedio!  
  
Para realizar una solicitud de un Pais tienes que indicar:  
    Nombre-Pais  
Para realizar una solicitud de un Canton tienes que indicar:  
    Nombre-Canton  
-----
```

```
-----  
Elija alguna de las opciones del menu  
1. Consultar un País  
2. Consultar un Cantón  
3. Salir del Programa  
-----
```

**Caso#8: Se sale del programa correctamente.**

```
-----  
Elija alguna de las opciones del menu  
1. Consultar un País  
2. Consultar un Cantón  
3. Salir del Programa  
-----  
  
3  
  
Se salió del programa.  
  
No threads ready or runnable, and no pending interrupts.  
Assuming the program completed.  
Machine halting!  
  
Ticks: total 3967, idle 0, system 20, user 3947  
Disk I/O: reads 26, writes 6  
Console I/O: reads 13, writes 6590  
Paging: faults 0  
Network I/O: packets received 0, sent 0  
  
Cleaning up...
```



## 9. Protocolo PIRO

Para la última entrega del proyecto, se definirá un nuevo protocolo para la comunicación entre los servidores intermedios y los servidores de datos. Para la dinámica, se requerirá un uso híbrido entre el protocolo HTTP y el protocolo PIRO. Dentro de las funcionalidades generales que se implementarán, se encuentran las siguientes:

- El sistema se compone de 3 personajes importantes: Los clientes, los servidores intermedios y los servidores de datos.
- Se requiere del uso de 2 protocolos de comunicación, HTTP y PIRO.
- Un cliente puede comunicarse únicamente a un solo servidor intermedio correspondiente.
- Un servidor intermedio puede comunicarse con diversos servidores de datos mediante un broadcast.
- Los servidores de datos responden a los servidores intermedios que les realizaron la solicitud.
- La solicitud entre un servidor intermedio y un servidor de datos se realiza mediante el siguiente formato:

URL: región="nombre-región"&pais="nombre-país"

- El formato para manejar la solicitud requiere el uso únicamente de mayúsculas, los espacios se deben cambiar por guiones intermedios y los nombres de los países deben manejarse en inglés.
- El puerto para comunicarse mediante el protocolo HTTP es el 51000 y el puerto para la comunicación mediante el protocolo PIRO es el 50000.
- Los servidores intermedios se identifican con un Id = 1 y los servidores de datos se identifican con un Id = 0.

A continuación se especifican las funcionalidades de cada personaje dentro del sistema:

### Cliente

Un cliente es capaz de solicitarle datos a un servidor intermedio acerca de un país en específico. Esta comunicación se realiza mediante una solicitud del protocolo HTTP y un cliente se puede comunicar únicamente con un único servidor intermedio.

### Servidor Intermedio

Un servidor intermedio es el encargado de realizar una funcionalidad tipo proxy entre un cliente y un servidor de datos. Este servidor recibe peticiones por parte de un cliente y a partir de esto, el servidor se comunica con un servidor de datos para obtener dicha información y brindarle al cliente. Este debe recibir la

petición HTTP del cliente y enviarle la misma petición al servidor de datos. Este servidor intermedio almacena una tabla con la información de los servidores de datos. Aca se almacenan las regiones asignadas a cada servidor de datos, por lo que se registran de la siguiente manera:

“región+IP”, siguiendo las mismas reglas mencionadas anteriormente.

La comunicación entre un servidor intermedio y un servidor de datos se da de la siguiente manera:

- Se levanta el servidor intermedio.
  - Enviar broadcast a todos los servidores para avisar que estoy activo. Este envío se realiza usando el siguiente formato: string separado con coma que tenga un 1 + [IPv4]. Ej: “1,172.16.4.205”.
  - Al enviar el broadcast, pueden darse los siguientes casos:
    - No hay otros servidores de datos activos, por lo que no hay respuestas para procesar y la tabla de servidores de datos queda en su estado actual.
    - Hay otros servidores de datos activos, por lo que se espera una respuesta que contenga un string con el siguiente formato: “Id, IP, region”.
    - Hay servidores intermedios activos, se reciben de 1 a N respuestas en el siguiente formato de string: “1,ipv4”, pero no se procesa la respuesta.
  - Se deja un oído activo para escuchar los broadcast de otros servidores. Si es un servidor de datos, se agrega a la tabla, pero si es un servidor intermedio, se descarta.

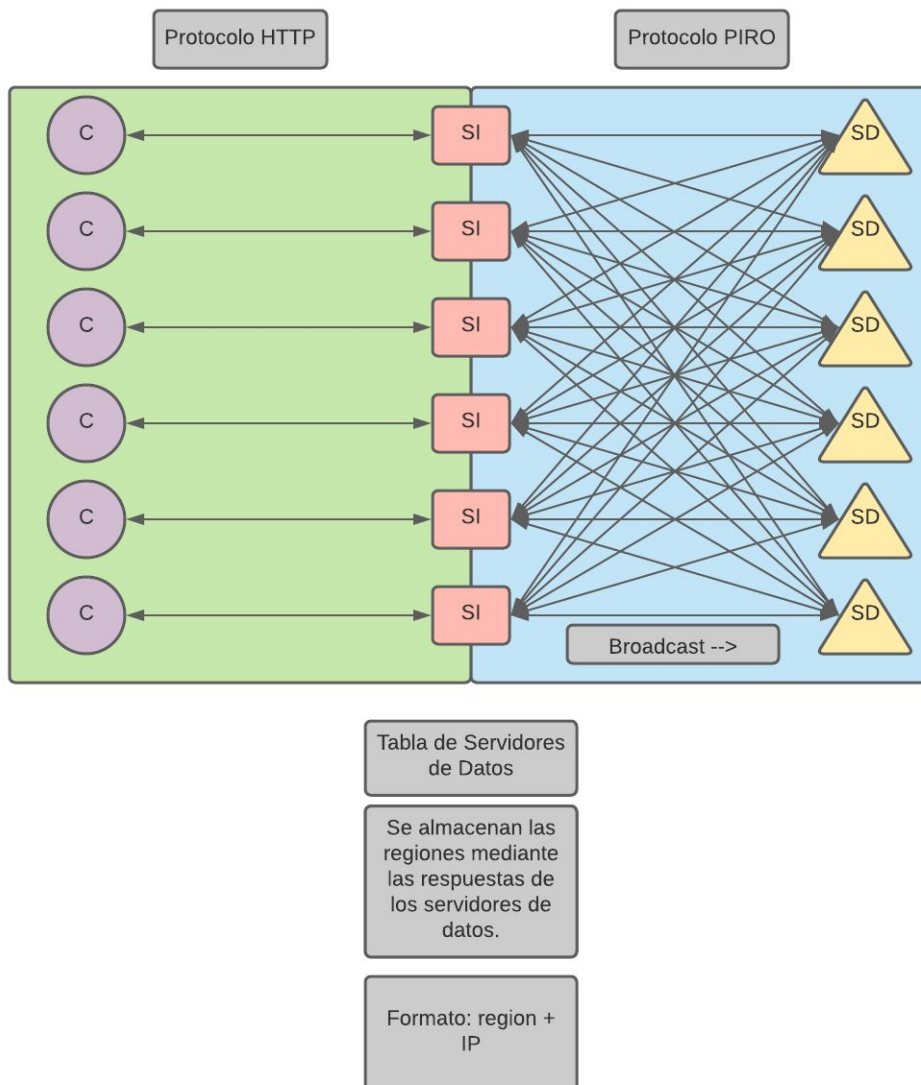
### Servidores de Datos

Un servidor de datos es el encargado de recibir la petición HTTP proveniente del servidor intermedio, y a partir de esto, ir a solicitar los datos de la petición y enviarle dichos datos de regreso al servidor intermedio. Estos servidores de datos pueden recibir peticiones de múltiples servidores intermedios. Cada servidor de datos tiene asignado una región de países, los cuales serán solicitados por el cliente.

La comunicación entre un servidor de datos y un servidor intermedio se da de la siguiente manera:

- Se levanta un servidor de datos.
  - Enviar broadcast para avisar al resto de servidores que estoy activo. El formato del envío se da mediante un string de la siguiente manera: “0, [IPv4], región. La región se maneja como un string, en minúscula, y los espacios se manejan con dash.
  - Se deja un oído activo para escuchar los broadcast de otros servidores. Si es un servidor de datos, se descarta, pero si es un servidor intermedio, se espera por la confirmación de la solicitud.

A continuación se presenta un esquema del sistema completo anteriormente mencionado:



## 10. Valoración

| Valoración Proyecto    | Primera Etapa   | Nota Proyecto |
|------------------------|-----------------|---------------|
| Miembro del Equipo     | Nota Individual | 100           |
| Ricardo Alfaro Víquez  | 100             |               |
| Allan Barrantes Chaves | 100             |               |
| Luis Rojas Carrillo    | 100             |               |
| Mario Vargas Campos    | 100             |               |

## 11. Bitácora

| Fecha      | Trabajo realizado   |
|------------|---|
| 10/12/2020 | Se hizo una reunión en la cual se condensaron los conocimientos y dudas respecto al sistema operativo NachOS, se complementó con la documentación y los videos del profesor para entender mejor su funcionamiento |
| 10/15/2020 | Se trabajó en pair programming para realizar los metodos de sincronización del sistema operativo NachOS   |
| 10/22/2020 | Se trabajó en pair programming para realizar los metodos de llamado al sistema pertenecientes al sistema operativo NachOS   |
| 10/23/2020 | Se completaron los metodos de llamado a sistema, y se implementaron los metodos que se necesitaban para el funcionamiento del socket  |
| 10/29/2020 | Se trabajó en pair programming para realizar los metodos manejo de memoria pertenecientes al sistema operativo NachOS   |
| 10/30/2020 | Se continuó trabajando en los métodos de manejo de memoria, se hicieron correcciones al manejo de tabla de archivos para su funcionamiento con Fork y Yield   |
| 11/2/2020  | Se empezó a implementar la conexión entre el sistema NachOS y el servidor   |
| 11/3/2020  | Se hicieron las correcciones en la clase cliente de NachOS para poder conectar con el servidor, se hicieron modificaciones para poder leer datos desde consola  |
| 11/5/2020  | Se añadieron nuevos diseños de impresión para las tablas, además se añadió un menú para la consola de cliente.  |
| 11/6/2020  | Se corrigieron errores, se realizaron las pruebas y se completó la documentación  |

## 12. Referencias Bibliográficas

- [1] Contributors, MDN. (2020). HTTP response status codes. Septiembre 8, 2020, de MDN Web Docs. Sitio web: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>.
- [2] Cplusplus.com. (2020). General C++ Programming. Septiembre 8, 2020, de cplusplus.com. Sitio web: <http://www.cplusplus.com/forum/general/>.
- [3] OpenSSL Software Foundation. (1997-2018). Open SSL Cryptography and SSL/TLS Toolkit. Septiembre 7, 2020, de OpenSSL Software Foundation. Sitio web: <https://www.openssl.org>.
- [4] <https://github.com/seleznevae/libfort>