

# Protocolo de Transmisión de Datos para Proyecto Integrador

## Especificación de Protocolo

Octubre 2022

Preparado para:

Profesores del curso CI0123: Proyecto Integrador de Sistemas Operativos y Redes de  
Comunicación

Universidad De Costa Rica

Realizado por:

Estudiantes del grupo 2 del curso CI0123

## Tabla de contenidos

1. Introducción	
a. Prefacio .....	3
2. Especificación	
a. Puertos y datagramas .....	3
b. Grupos asignados a cada servidor .....	3
c. Establecimiento de conexión .....	4
d. Comunicación entre cliente-servidor .....	5
e. Cerrando conexión .....	6
f. Formato de respuesta de los servidores .....	7

## 1. Introducción

### a. Prefacio

Este documento describe la implementación del protocolo diseñado por el grupo de estudiantes del grupo 2 del curso CI0123 para el envío de datos de servidores a clientes. Se especifica el formato en que los datos serán enviados de los servidores a los *routers*, como se realizara el establecimiento de una conexión entre clientes-servidores, como se llevara a cabo la comunicación entre ambos, como se debe cerrar la conexión, entre otros elementos.

## 2. Especificación

### a. Puertos y datagramas

Los siguientes puertos serán utilizados para el establecimiento de conexiones y envío de datos:

Dispositivo	TCP	UDP
<i>Router</i>	3022	2022
Servidor	3020	2020

Tabla 1: Puertos que se van a utilizar para el envío de datos.

Con respecto a los datagramas que se van a enviar, su contenido es el siguiente:

- Datagrama UDP: *Router* a Servidor  
Envía: IP de *router*  
Nota: la IP se va a mandar como un *string*. Por ejemplo: “192.18.6.10”.
- Datagrama UDP: Servidor a *Router*  
Envía: IP de servidor  
Código de grupo  
Nota: la IP se va a mandar como un *string*. Por ejemplo: “158.92.30.17”  
El código de grupo se va a enviar como un *char*. Por ejemplo: ‘C’

### b. Grupos asignados a cada servidor

Número de servidor	Grupo
1	E
2	G
3	F
4	B
5	D
6	H

c. Establecimiento de conexión

Para establecer una conexión, se tomo en consideración dos casos: cuando un *router* trata de establecer la conexión con un servidor o cuando es el servidor quien busca establecer una conexión con el *router*.

Caso A: *Router* a servidor

1. El *router* hace un *broadcast* al puerto 2020 UDP del servidor, enviándole un datagrama UDP con su IP.
2. El servidor responde con un *unicast* al puerto 2022 UDP del *router*, en donde envía un datagrama UDP con su IP y el código de grupo.

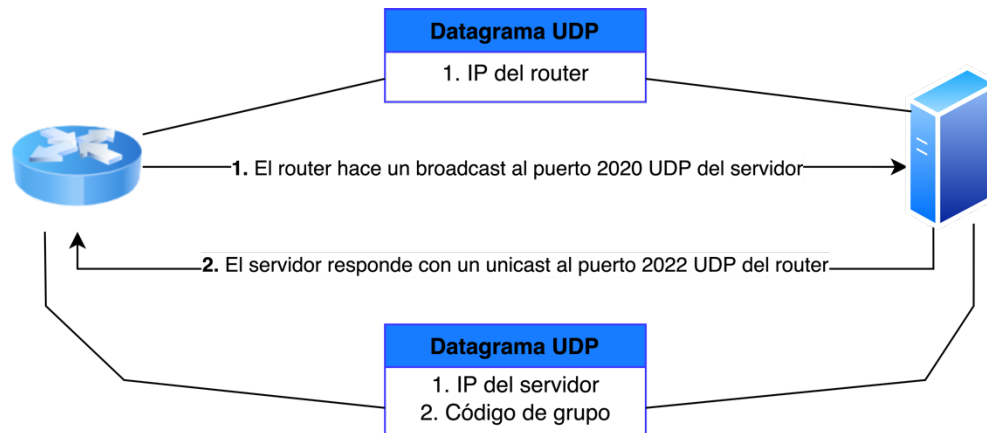


Figura 1: Diagrama de establecimiento de una conexión entre *router* y servidor

Caso B: Servidor a *router*

1. El servidor hace un *broadcast* al puerto 2022 UDP del *router*, enviándole un datagrama UDP con su IP y el código de grupo.
2. El *router* responde con un *unicast* al puerto 2020 UDP del servidor, en donde envía un datagrama UDP con su IP.

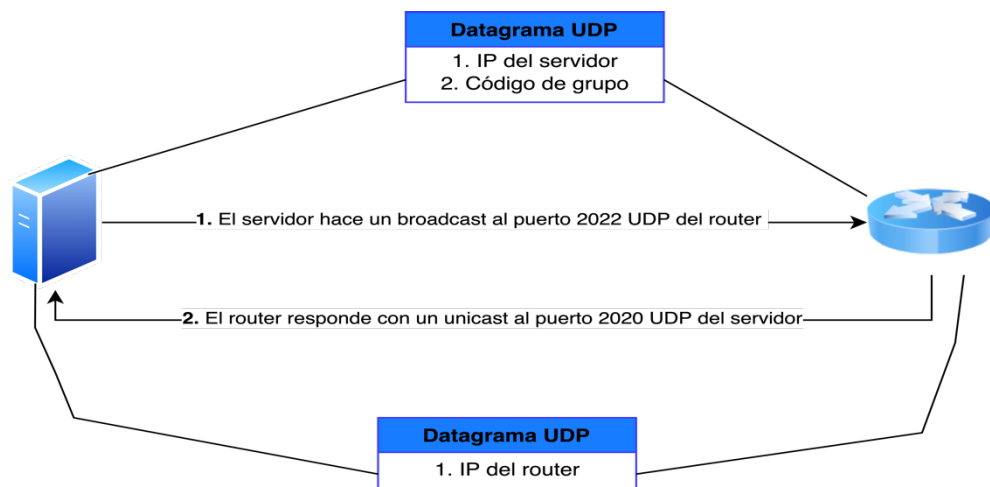


Figura 2: Diagrama de establecimiento de una conexión entre servidor y *router*

#### d. Comunicación entre cliente-servidor

Una vez establecida la conexión, se va a comenzar a recibir solicitudes por parte de los clientes:

1. El cliente realiza un *request* al servidor (la cual llega al *router*).  
Nota: El *request* que se va a utilizar es el del protocolo HTTP.
2. El *router* realiza la consulta al servidor para verificar si el país está contenido en la tabla que maneja el servidor:

##### a. Si el país esta en la tabla:

El *router* envía la solicitud (GET) al servidor. Este envía los datos (en el formato establecido entre los grupos del proyecto integrador) al *router* y este los pasa al cliente. Además, envía el código 200 (OK) al cliente.

El formato del GET va a ser el siguiente:

GET /fifa/año/\_\_\_\_  
                                  |\_\_\_\_ código de país de acuerdo al formato establecido por la Fifa

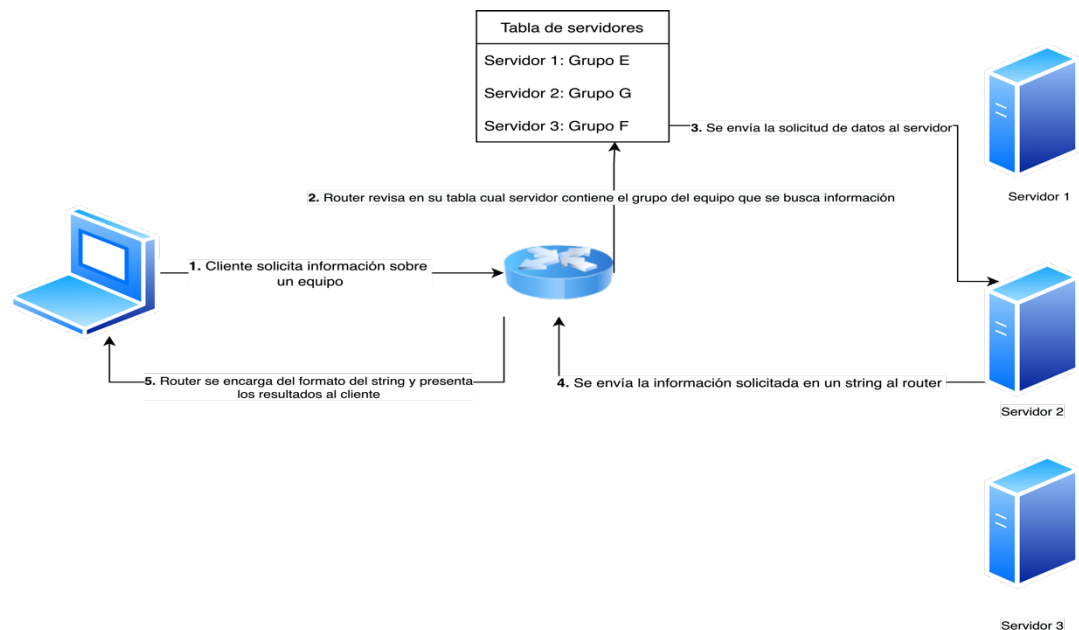


Figura 3: Diagrama de un *request* siendo atendido de manera exitosa

- b. Si el país no esta en la tabla:  
El *router* se encarga de enviar el código 404 (NOT FOUND) al cliente.

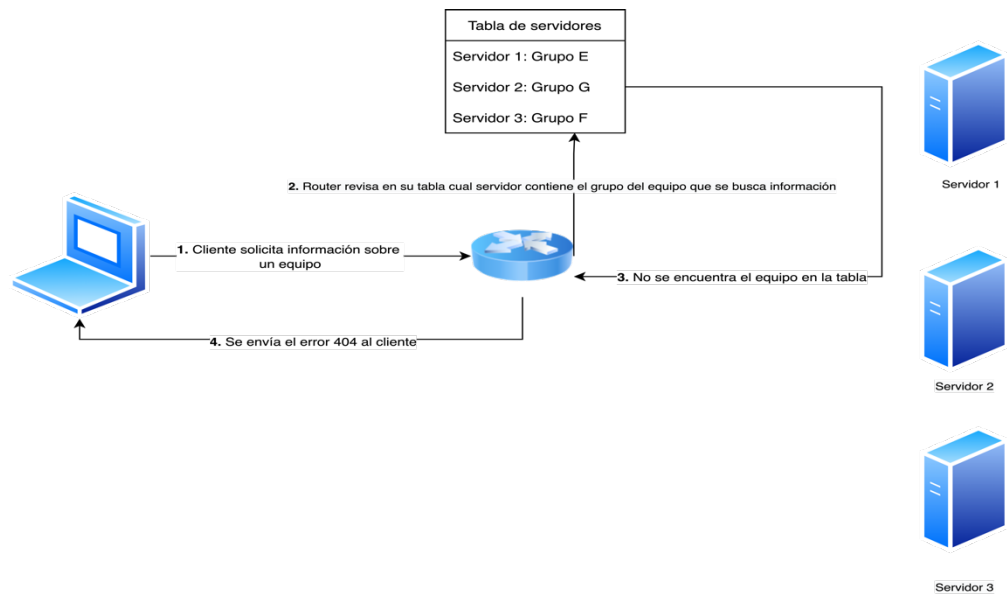


Figura 4: Diagrama de *request* inválido

#### e. Cerrando conexión

Para finalizar la conexión, se consideran dos casos:

1. Si el *router* envía una solicitud y no recibe respuesta alguna del servidor después de cierto tiempo, entonces el *router* hace un *timeout* en donde elimina al servidor de su tabla. Esto se considera por si hubiera algún problema que provocara que el servidor se apague/cierre inesperadamente.

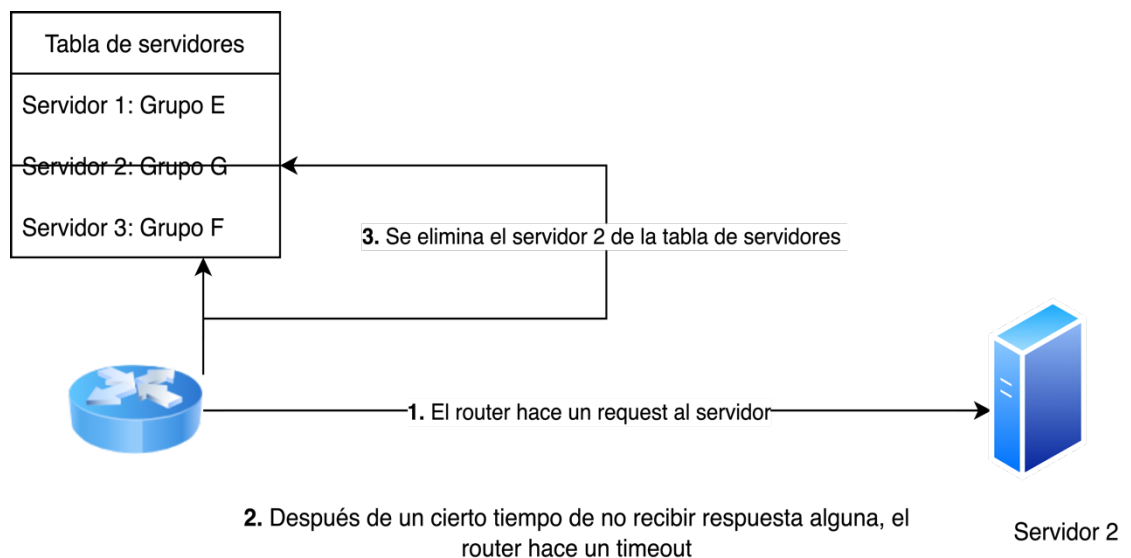


Figura 5: Diagrama de cierre de conexión si el servidor se apaga de manera inesperada

2. Si el servidor se cierra de manera normal, este se encarga de hacer un *broadcast* a los *routers* para que los eliminen de su tabla de servidores disponibles.

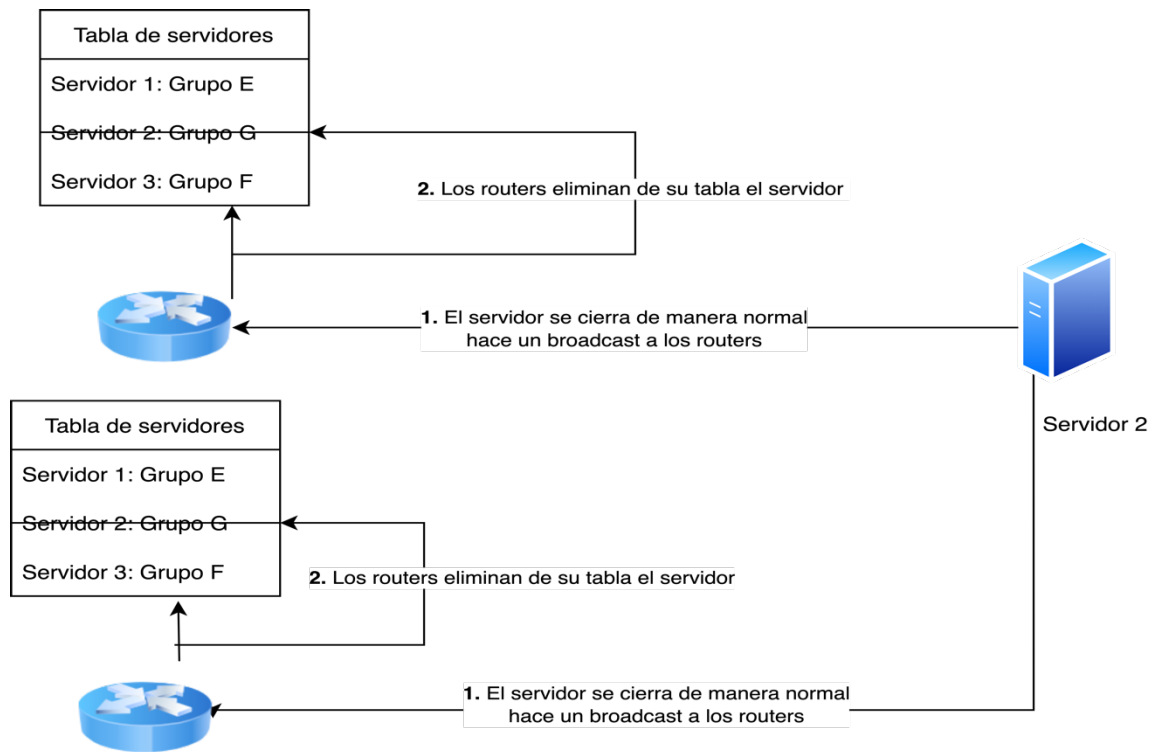


Figura 6: Diagrama de cierre de conexión si el servidor es cerrado de manera normal

#### f. Formato de respuesta de los servidores

Debido a que entre los grupos del proyecto se trabajó con distintos formatos de archivos (JSON, CSV, txt) se decidió que los servidores envíen los datos en un *string* y cada equipo se encarga de manipularlos de acuerdo al formato que implementaron. El formato de respuesta de los servidores corresponde a:

Para fase de grupos:

Sigla\_Grupo\n

Sigla\_Equipo\tPosicion\tPartidos\_ganados\tPartidos\_empatados\tPartidos\_perdidos\tGoles\_a\_favor\tGoles\_en\_contra\tGoles\_diferencia\tPuntos\n

Para octavos, semifinales, etc:

Fase\tSigla\_Equipo1\tSigla\_Equipo2\tResultado\n

En el caso que se tuviera que definir el resultado mediante penales, estos se deberán de escribir entre paréntesis. Por ejemplo:

Final\tCRC\FRA\t5(1)-5(2)\n

Para la parte de fases y final, se decidió utilizar el siguiente formato:

round\_16

round\_8

round\_4

round\_2

round\_2\_loser