

## Tarea Corta II

Tecnológico de Costa Rica  
Escuela de Ingeniería en Computación  
Redes (IC 7602)  
Primer Semestre 2022



### 1. Objetivo General

- Implementar una calculadora de subredes IPv4 utilizando lenguaje de programación C y Docker compose.

### 2. Objetivos Específicos

- Desarrollar habilidades en sockets, operadores bitwise y lenguaje de programación C.
- Implementar un protocolo para intercambiar información entre un cliente y un servidor.

### 3. Datos Generales

- El valor del proyecto: 3%
- La tarea debe ser implementada de forma individual.
- La **fecha de entrega** es 13/05/2022 antes de las 11:59 pm. Si la tarea es entregada después de esta hora, será calificada con una nota de 0.
- Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo con el reglamento. La copia incluye código/configuraciones que se puede encontrar en Internet y que sea utilizado parcial o totalmente sin el debido reconocimiento al autor.
- Se debe incluir documentación con instrucciones claras para ejecutar la tarea.
- Se deben seguir buenas prácticas de programación en el caso de que apliquen. Por ejemplo, documentación interna y externa, estándares de código, diagramas de arquitectura, diagramas de flujo, pruebas unitarias son algunas de las buenas prácticas que se esperan de un estudiante de Ingeniería en Computación en sus cursos finales.
- Toda documentación debe ser implementada en Markdown.
- Si la tarea no se encuentra automatizado con Docker / Docker compose se obtendrá una nota de 0. Si la tarea no se entrega completo, la porción que sea entregada debe estar completamente automatizada, de lo contrario se obtendrá una nota de 0.
- En caso de no entregar documentación se obtendrá una nota de 0.
- El email de entrega debe contener una copia de la tarea en formato tar.gz y un link al repositorio dónde se encuentra almacenado, debe seguir los lineamientos en el programa de curso.
- Si los cálculos no se realizan utilizando operadores bitwise, se obtendrá una nota de 0.

### 4. Descripción

Implementar un servidor en lenguaje de programación, el mismo escuchará en el puerto TCP/9666, cuando un cliente se conecta al servidor, podrá enviar los siguientes mensajes (primitivas):

- GET BROADCAST IP {dirección IP} MASK {mascara en formato /bits o notación X.X.X.X}  
Por ejemplo:
  1. GET BROADCAST IP 10.8.2.5 MASK /29
    - a. Retorna 10.8.2.7
  2. GET BROADCAST IP 172.16.0.56 MASK 255.255.255.128
    - a. Retorna 172.16.0.127
- GET NETWORK NUMBER IP {dirección IP} MASK {mascara en formato /bits o notación X.X.X.X}  
Por ejemplo:
  1. GET NETWORK NUMBER IP 10.8.2.5 MASK /29
    - a. Retorna 10.8.2.0
  2. GET NETWORK NUMBER IP 172.16.0.56 MASK 255.255.255.128
    - a. Retorna 172.16.0.0
- GET HOSTS RANGE IP {dirección IP} MASK {mascara en formato /bits o notación X.X.X.X}  
Por ejemplo:
  1. GET HOSTS RANGE IP 10.8.2.5 MASK /29
    - a. Retorna 10.8.2.{1-6}
  2. GET HOSTS RANGE IP 172.16.0.56 MASK 255.255.255.128
    - a. Retorna 172.16.0.{1-126}
- GET RANDOM SUBNETS NETWORK NUMBER {Y.Y.Y.Y} MASK {mascara en formato /bits o notación X.X.X.X} NUMBER {número de redes} SIZE {mascara en formato /bits o notación X.X.X.X}  
Por ejemplo:
  1. GET RANDOM SUBNETS NETWORK NUMBER 10.0.0.0 MASK /8 NUMBER 3 SIZE /24
    - a. Puede retornar:
      - i. 10.20.10.0/24
      - ii. 10.33.11.0/24
      - iii. 10.42.13.0/24

Únicamente se tendrá que implementar el servidor, como cliente utilizaremos el comando telnet. Se deben validar errores y proporcionar mensajes de error útiles para el usuario.

## 5. Entregables

- Documentación.
- Docker compose junto con todos los archivos/scripts requeridos para ejecutar su tarea.

## 6. Evaluación

Funcionalidad / Requerimiento	Porcentaje
Documentación	10%
Implementación: <ul style="list-style-type: none"><li>• Calculadora</li></ul>	90%