

**Parte Teórica - Valor 1.8**

1. (0.4) Cuando se define una red o subred, ya sea utilizando direcciones tipo classfull o classless, se debe tener en cuenta que siempre se van a tener \_\_\_ direcciones menos disponibles de las especificadas en el rango para asignar a los equipos; esto debido a que esas direcciones se utilizan obligatoriamente para: (Seleccione todas las que apliquen)
  - a) Permitir el enlace hacia otras redes
  - b) La dirección de Multicast
  - c) La dirección de Broadcast
  - d) La dirección de Unicast
  - e) La dirección que identifica la red
  - f) Reservarse para uso futuro
2. (0.3) Explique claramente qué significa DNS y cuál es su utilidad en la comunicación en red, principalmente en Internet.
3. (0.3) Realice un paralelo entre los niveles del modelo OSI y los de la pila TCP, donde se identifique la equivalencia entre dichos niveles. No es necesario explicar los niveles, solamente mencionarlos.
4. (0.8) Indique cuáles de las siguientes afirmaciones son ciertas, justifique su respuesta solamente en caso de ser falsa:
  - a. Es posible enviar cualquier objeto por la red, ya sea por TCP o UDP utilizando las clases ObjectOutputStream y ObjectInputStream.
  - b. Es posible utilizar un DatagramSocket para enviar datos a múltiples destinatarios, sin embargo, el envío no se puede hacer simultáneo.
  - c. Objetos de las clases PApplet y PImage no se pueden enviar por la red debido a que implementan Serializable.
  - d. La clase Client, utilizada en Bluetooth, no permite el envío de objetos, solamente de algunos tipos de datos básicos.

**Parte práctica - Valor 3.2**

Como se ha visto en algunos casos, no es posible mover objetos PImage entre aplicaciones, dado que esta clase no implementa Serializable.

Para solucionar este inconveniente se requiere que desarrolle una solución que permita solicitar a un servidor una imagen y el mismo la envíe al cliente que la está solicitando.

El servidor iniciará con un conjunto de imágenes en su carpeta data y las organizará de algún modo, de tal forma que cuando le soliciten la imagen 1 envíe una, la imagen 2 envíe otra y así sucesivamente.

Cuando un cliente requiera una imagen, enviará un byte al servidor con el número solicitado y se quedará esperando a que el servidor le conteste con la imagen solicitada. Se puede asumir que se tienen 5 imágenes disponibles.

Al momento de recibirla, la guardará y mostrará por pantalla al usuario.

El servidor debe poder atender varios usuarios al tiempo, en caso que llegue más de una solicitud de imagen, lo cual no implica que tenga que utilizar multihilos. Al tener que atender a varios usuarios, el servidor debe responder al cliente que le solicitó la información (no puede tener la dirección y puerto de respuesta establecido).

1. (1.0) Explique claramente la secuencia de pasos para realizar la comunicación. Indique qué se ejecuta en el cliente y en el servidor y en qué método se realizaría.
  2. (1.2) Elabore el código la aplicación Cliente. Debe permitir al usuario escoger cual imagen desea obtener desde el servidor y mostrarla cuando se haya recibido.
  3. (1.0) Elabore el código la aplicación Servidor. No se requiere que tenga interfaz gráfica.
- Para resolver el ejercicio, debe apoyarse en los métodos en entrada y salida de archivos que permiten cargar y grabar archivos hacia/desde un arreglo de bytes, a saber loadBytes() y saveBytes().
  - Puede asumir que las imágenes a enviar pesan menos de 64Kb (65536 bytes).
  - Para probar su aplicación cliente, puede intentar conectarse al servidor que se encuentra corriendo en la IP 192.168.\_\_\_\_. Este servidor le responderá con una imagen cuando Ud le envíe el número.