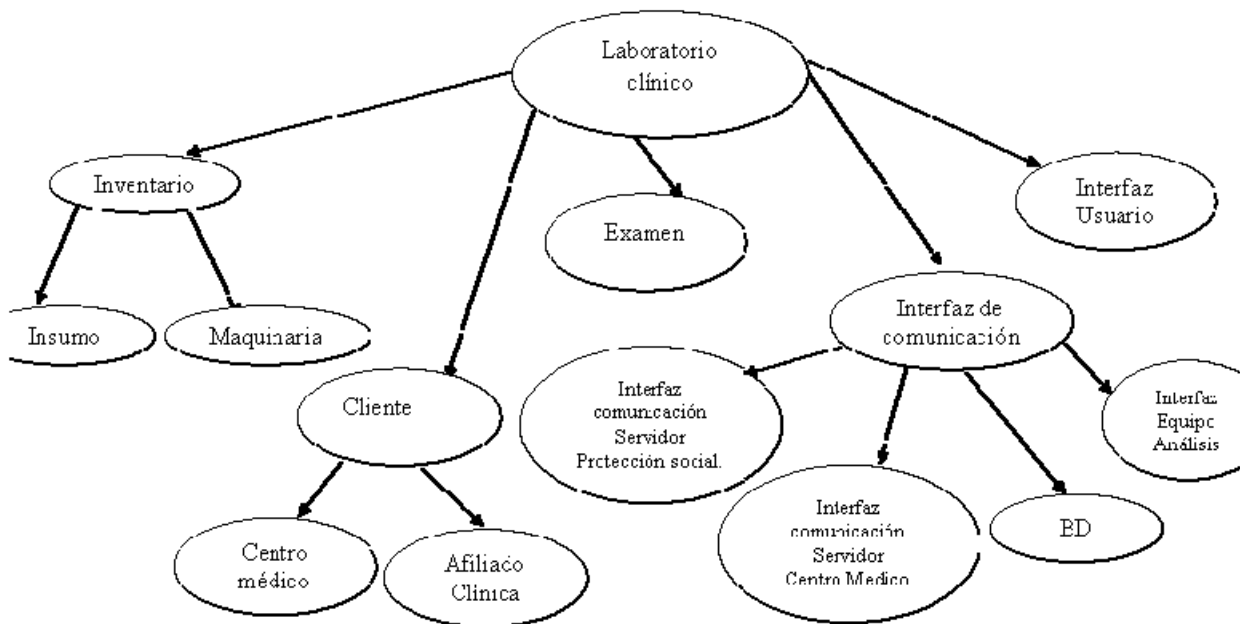


Punto I

Con respecto a los requerimientos y el planteamiento de análisis a continuación:

1. Registrar datos básicos de los pacientes.
2. Administrar la información de las máquinas.
3. Administrar la información de los insumos necesarios para las muestras.
4. Recibir por internet información de clientes de centros médicos.
5. Clasificar los exámenes bajo parámetros establecidos por el ministerio de protección social.
6. Codificar los exámenes.
7. Enviar información de los clientes y sus respectivos exámenes a la máquina.
8. Recibir información de los resultados de los exámenes realizados por la máquina.
9. Actualizar la información del cliente.
10. Enviar información a centros médicos.



| Requerimiento | Inventario | Examen | Interfaz de comunicación | Cliente | Interfaz Usuario |
|---------------|------------|--------|--------------------------|---------|------------------|
| 1 | | X | X | X | X |
| 2 | X | | X | | X |
| 3 | X | | X | | X |
| 5 | | X | X | | X |
| 6 | | X | X | X | |
| 7 | | X | X | X | |
| 8 | | X | X | | |
| 9 | | X | | X | |
| 10 | | | X | X | |

1. Analice el particionamiento y asignación de requerimientos planteado, la propuesta de subsistemas y sub-subsistemas, en términos de **acoplamiento** (tip: revise cada subsistema y la razón por la cual tiene requerimientos asignados).

2. De acuerdo al análisis del punto anterior, replantee el diagrama de particionamiento, explicando **cómo mejora el acoplamiento**.

Punto II

Para el siguiente “código a evaluar” planteado en su trabajo, escriba el **conjunto de casos de prueba para cubrimiento de código** (distinga claramente cada caso de prueba):

```
public void run(){
chat.getEnviar().addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evento) {
try{
String cadena = getChat().getTexto().getText();

Imagelcon icono = (lmagelcon)chat.getImagenes().getSelectedItem();

String cadenaNueva = "";
Imagelcon icon = null;

switch(Integer.parseInt(icono.getDescription())){
case 1: cadenaNueva +=cadena;
break;
case 2: cadenaNueva += ");"+" "+cadena;
icon = new lmagelcon("guiño.jpg");
break;
case 3:  cadenaNueva += ":%."+" "+cadena;
icon = new lmagelcon("pena.jpg");
break;
case 4: cadenaNueva += "):"."+" "+cadena;
icon = new lmagelcon("smile.jpg");
break;
case 5: cadenaNueva += ":D."+" "+cadena;
icon = new lmagelcon("supersmile.jpg");
break;
case 6: cadenaNueva += ":(."+" "+cadena;
icon = new lmagelcon("triste.jpg");
break;
}
System.out.println(cadenaNueva);
System.out.println("casas teta.....");

if(icon != null) {
getChat().getConversación().insertlcon(icon);
}else{
getChat().getConversación().setText(getChat().getConversación().getTex
t()+"\n"
+Control.getYo().getLogin()+" dice: "+" \n"+cadenaNueva+"\n");
getChat().getTexto().setText("");
Control.guardarConversación(Control.getYo().getLogin()+" dice:
"+cadenaNueva);
}

byte datos Q = null;
datos = cadena.getBytes();
DatagramPacket enviado = new
DatagramPacket(datos, datos.length,chat.getDir(), 65000);
socketUDP.send(enviado);
}
catch (IOException e) {
e.printStackTrace();
}
}
}
```