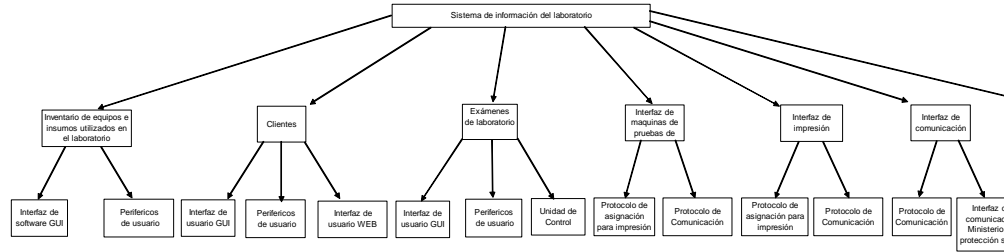


Punto I



# Req	Requerimiento	Inventario maquinas e insumos laboratorio	Clientes	Exámenes de laboratorio	Interfaz de Maquinas de pruebas de laboratorio	Interfaz de impresión	Interfaz de comunicación
1	Datos Básicos de clientes propios		X	X			
2	Datos Básicos clientes centros hospitalarios		X				X
3	Registro tipo de exámen médico requerido para realizar el análisis		X	X			
4	Impresión factura		X	X		X	
5	Gestón de tipo de exámen de laboratorio			X			
6	Gestón de insumos	X					
7	Alerta por falta de insumos			X			
8	Gestón de Maquinas	X					
9	Recepción resultados análisis maquina				X		
10	Registro de realización de pago		X				
11	Imprimir resultado de los clientes propios después de cancelar valor de factura		X			X	
12	Generar e impresión de código de barras		X	X	X	X	
13	Actualización BD M PROT SOCIAL						X
14	Envío de los resultados por internet después de cancelar valor		X				X

1. Analice el particionamiento y asignación de requerimientos planteado, la propuesta de subsistemas y subsistemas, en términos de **acoplamiento** (tip: revise cada subsistema, su particionamiento y la razón cual tiene requerimientos asignados y por qué no hay un subsistema de interfaces de usuario, como lo hay para las máquinas o para la comunicación, y sí lo hay para la BD?)

2. Qué periféricos de usuario hay ?

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 p**  
**cubrimiento de código:**

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