

## Lab session 5: Create a Complete Application

El objetivo de esta práctica era implementar el diseño de un hospital sencillo visto en la sesión de seminario 5. Había unos huecos a completar de un código que nos daban para que se ejecutara correctamente la clase principal Hospital Manager.

Hemos seguido también el esquema proporcionado como referencia, que es un esquema tipo Prototipo.

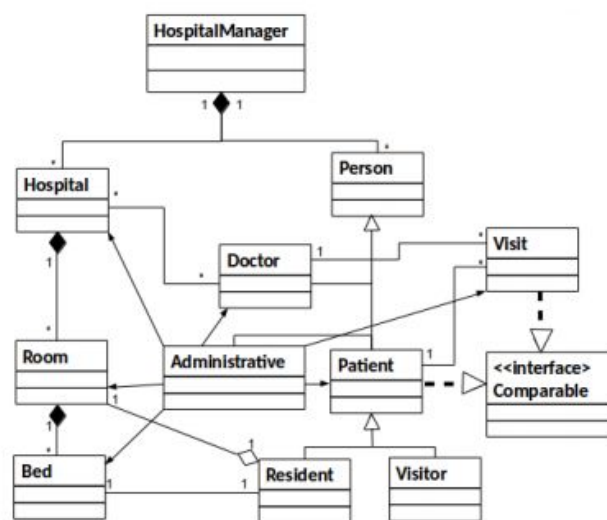


Figure 1: HospitalManager Class Diagram

Ya que gran parte del código venía dado, vamos a explicar un poco solo las funciones más relevantes.

### Administrative

```
19 public boolean assignBed( Resident resident ) {
20     LinkedList<Room> rooms = hospital.getRooms();
21     for(int i=0; i<rooms.size();i++){
22         Bed b = rooms.get(i).getAvailableBed();
23         if(b!=null){
24             resident.assignBed(b);
25             resident.assignRoom(rooms.get(i));
26             System.out.println(toString()+" has assigned bed to\n"+resident.toString());
27             return true;
28         }
29     }
30     System.out.println(toString()+" has not found bed for\n"+resident.toString());
31     return false;
32 }
```

Antonio Pintado Lineros u172771  
Amanda Pintado Lineros u137702

En esta clase, el método que ha requerido más trabajo ha sido el que asignaba camas a un residente en concreto, donde debíamos comprobar para todas las habitaciones si había una cama disponible y si la hay, asignarla al residente.

## Hospital

```
88 public String toString(){
89     String str="Hospital "+name+"\n";
90     str=str+"Administratives:\n";
91     for(int i=0; i<admins.size();i++){
92         str=str+admins.get(i).toString()+"\n";
93     }
94     str=str+"Doctors:\n";
95     for(int i=0; i<doctors.size();i++){
96         str=str+doctors.get(i).toString()+"\n";
97     }
98     str=str+"Patients:\n";
99     for(int i=0; i<patients.size();i++){
100         str=str+patients.get(i).toString();
101     }
102     str=str+"Rooms:\n";
103     for(int i=0; i<rooms.size();i++){
104         str=str+rooms.get(i).toString()+rooms.get(i).listBeds()+"\n";
105     }
106     return str;
}
```

Un método relevante para esta clase podría ser ToString, que interactúa directa e indirectamente con los métodos toString del resto de clases, cada una con el suyo propio.

## Patient

```
37 public int compareTo( Patient p ){
38     if(age < p.getAge()){
39         return -1;
40     }else if(age > p.getAge()){
41         return 1;
42     }
43     return 0;
}
```

Este método se parece un poco a los métodos que tuvimos que implementar en el pasado Lab4, así que la implementación era bastante parecida y sencilla.

## Conclusión

La práctica ha sido bastante sencilla, sobre todo gracias al código ya dado, lo que nos ha ahorrado un buen rato de crear cada clase e inicializar los miembros, que es un trabajo bastante mecánico y que realmente no es complicado.

Es de agradecer que la práctica haya sido sencilla teniendo en cuenta que dentro de poco empezará el periodo de exámenes finales y también que haya sido colgada con antelación para poder adelantar trabajo.