

Sistema de Información para Urgencias

El Modelo o Mundo

Esta parte de la aplicación contará con las siguientes clases:

1. Comenzar con la Ubicación. Estas clases comparables y usaremos primero la calle y luego la carrera como criterio de comparación. Escriba métodos constructores, analizadores y modificadores para esta clase.
2. Construya una función externa para hallar la distancia entre dos ubicaciones. El resultado de esta función es un número entero y usa el esquema de la distancia Manhattan para encontrar tal distancia.
3. Sigamos con la ambulancia. Agregue los atributos. Tenga en cuenta que cuando la ambulancia está vacía, el accidentado debe ser **null**. Agregue métodos para
 - ingresar un accidentado a la ambulancia: la ambulancia debe estar vacía, y se recibe el accidentado como dato de entrada. No olvide cambiar el estado de la ambulancia.
 - desocupar la ambulancia: la ambulancia debe estar ocupada. Coloca el accidentado en null y cambia el estado.
 - Cambiar la ubicación de la ambulancia.
 - Analizadores para obtener los atributos
4. La clase Accidentado también será comparable. Para esta clase, el criterio de comparación será el nombre del accidentado.
5. Un Hospital también será comparable. El criterio de comparación será el código del hospital. Escriba el constructor y los analizadores. Agregue los siguientes métodos:
 - Determinar si un paciente con un nombre dado está en ese hospital.
 - Saber si el hospital atiende un determinado accidente.
 - Ingresar un accidentado al hospital. Solo se admite el paciente si no está ya en el hospital y el accidente del accidentado es atendido por el hospital.
 - Dar de alta a un paciente: dado el nombre del paciente, se debe eliminar ese paciente del hospital.

6. Ahora vamos a diseñar una clase que responde al patrón Singleton, que será el controlador en nuestra aplicación: El Sistema De Urgencias. Esta clase tendrá una lista de los hospitales y ambulancias que hay alrededor de la ciudad. Además de los atributos y del constructor por defecto, agregue métodos que realicen las siguientes tareas:
- Agregar una ambulancia: recibe el código de la ambulancia, la calle y la carrera donde está ubicada la misma. Sencillamente verifica que no haya una ambulancia con el mismo código, crea la ambulancia (en un estado libre y sin paciente) y la anexa al final de la lista de ambulancias del Sistema de Urgencias.
 - Agregar un hospital: recibe el código del hospital, el nombre, la calle y la carrera, así como los dos nombres de los accidentes. La operación verifica que no exista otro hospital con ese código, y crea el hospital y lo anexa al final de la lista de hospitales del sistema.
 - Ocurrió un accidente en cierta ubicación de la ciudad. Este método recibe un accidentado y determina la ambulancia disponible (libre) más cerca al accidentado dado. Si no hay tal ambulancia, deberá retornarse **null**.
 - Actualizar la ubicación de una ambulancia. A medida que se mueve la ambulancia, es necesario cambiar la ubicación de la misma. Este método necesita el código de la ambulancia y la nueva ubicación. El método busca la ambulancia que tiene el código dado y cambia la ubicación de la misma siempre y cuando esté libre. Las ambulancias ocupadas no cambian de ubicación.
 - Asignar un accidentado a una ambulancia. En este caso necesitamos el accidentado y la ambulancia. Es obligatorio que la ambulancia esté libre. Lo único que debe hacer esta función es asignar el paciente a la ambulancia dada, cambiando el estado de la ambulancia a ocupada.
 - Buscar hospital para una ambulancia. El método recibe una ambulancia (precondición: la ambulancia debe estar ocupada), y retorna el primer hospital de la lista que pueda atender el accidente del accidentado que lleva dentro la ambulancia. Este hospital debe tener entre sus especialidades aquella que sufrió el accidentado y debe ser el más cerca a la ubicación actual de la ambulancia y no debe tener un accidentado que se llame igual al que lleva la ambulancia. Si no hay tal hospital, deberá retornarse **null**.

- Llegada de una ambulancia a un hospital. Este método recibe la ambulancia (debe estar ocupada), busca el hospital para el accidentado que lleva la ambulancia y deja la ambulancia libre. Cambia la posición de la ambulancia a la misma posición del hospital e ingresa el paciente al hospital. No debe haber un paciente con el mismo nombre en el hospital o no podrá ser admitido.
- Dar de alta a un accidentado. Dado el código del hospital y el nombre de un paciente, da de alta al accidentado del hospital que tiene el código dado. Precondición: debe existir el hospital con el código dado y debe existir dentro del hospital, el accidentado con el nombre dado.

La Vista

Su aplicación Android desarrollada en Kotlin deberá ofrecer al usuario final al menos las siguientes opciones:

1. Ingresar la información de una Ambulancia. Deberá solicitar el código y ubicación de la nueva ambulancia (calle y carrera) solamente.
2. Ingresar la información de un nuevo hospital al sistema. La aplicación deberá pedirle al usuario el código del nuevo hospital, el nombre del mismo, la ubicación (calle y carrera), y los nombres de los dos accidentes que se manejan en ese hospital.
3. Actualizar la información de una ambulancia: solicita el código de la ambulancia, y la calle y la carrera de su nueva ubicación.
4. Informar un accidente: la aplicación debe solicitar el nombre del accidentado, el nombre del accidente que éste sufrió y la calle y carrera donde ocurrió el accidente. A partir de esa información se crea el objeto accidentado y se le busca una ambulancia que lo lleve a un hospital cercano. Luego de montar el accidentado a la ambulancia, se debe desplegar el código de la ambulancia que va a transportar al accidentado. Este código debe aparecer en pantalla.
5. Llevar un accidentado a un hospital: este caso de uso solamente recibe el código de la ambulancia. Si la ambulancia está ocupada, busca el hospital para llevar al paciente de la ambulancia. Si existe ese hospital, mostrar el código y nombre en la pantalla, e ingresar el paciente al hospital.

6. Dar de alta un accidentado: con el código del hospital y el nombre del paciente. Se informa por pantalla si fue posible realizar la operación.
7. Mostrar todas las ambulancias del sistema
8. Mostrar todos los hospitales del sistema
9. Mostrar todos los pacientes de un hospital: dado el código del hospital
10. Mostrar todas las ambulancias libres
11. Mostrar todas las ambulancias ocupadas
12. Mostrar la información de un hospital: se ingresa el código, y se presenta el nombre, ubicación y accidentes de ese hospital.