

Reporte tarea 11 - Programación y algoritmos

Erick Salvador Alvarez Valencia, Centro de Investigación en Matemáticas, A.C.

Index Terms—Programación orientada a objetos, UML, Triage.

I. INTRODUCCIÓN

En el presente reporte se analizará la simulación de un protocolo de intervención llamado triage, el cual se realiza en un hospital a los pacientes que van ingresando. Para la simulación se utilizó el lenguaje C++ en su paradigma de Programación Orientada a Objetos (POO) ya se permite generar una abstracción muy buena del problema. Para un mejor entendimiento se proporcionará un diagrama en formato UML de las clases y los métodos.

mds

Noviembre 13, 2017

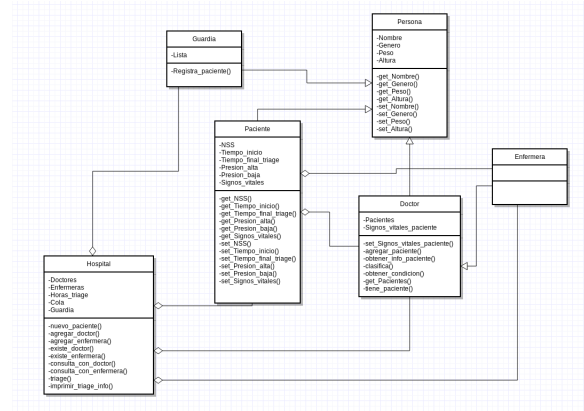
II. TRIAGE

II-A. Descripción

El triage es un protocolo de intervención que en este caso se ejecuta en cierto hospital a los pacientes que ingresan al mismo, los pasos a ejecutar son los siguientes:

1. El paciente llega al servicio de urgencias.
2. El vigilante registra al paciente con su reloj checador.
3. El paciente va a la sala de espera.
4. En caso de que el paciente llegue en la ambulancia, pasa directamente al consultorio del triage.
5. El médico o la enfermera pasan al paciente a la sala del triage cuando es su turno.
6. Se registran los datos básicos del paciente, así como sus signos vitales.
7. Se realiza la clasificación en base a los signos vitales (para ellos se utiliza un formato).
8. Se decide cuál es el nivel de gravedad del paciente, para el cual puede ir desde *Sin urgencia* hasta *Rehabilitación inmediata*.
9. Se firma el formato del triage.
10. Los pacientes pasan a otra sala o a urgencias dependiendo del resultado anterior a esperar para ser atendidos.

Para modelar o simular los pasos descritos anteriormente se utilizó el paradigma POO (Programación Orientada a Objetos) con el que cuenta C++ ya que con él se puede generar una abstracción muy buena de la situación anterior, y en general de muchas otras situaciones, esa es la gran ventaja que nos brinda este paradigma. Para la simulación primero se formuló una estructura de clases las cuales contuvieran atributos y métodos que se acomplanaran bien, y por lo cual se elaboró un diagrama UML que facilitó el trabajo. A continuación se muestra el diagrama UML de las clases elaboradas:



(a) Figura 1. Diagrama UML del triage.

Podemos ver en la Figura 1. que tenemos 6 clases elaboradas, a continuación se explicará su composición y jerarquía.

- **Clase persona:** Esta es una clase base, sabemos que todo individuo que está contenido en el triage es una persona como tal, así que se comenzó definiendo esta clase de la cual heredarían todas las que tengan que ver con personas. Esta clase contiene atributos y métodos básicos, se sabe que una persona tiene: Nombre, Género, Peso, Altura, entre otros atributos, y como estos mismos se definieron en modo privado se declararon algunos métodos que funcionarían como getters y setters.
- **Clase paciente:** Esta clase es muy necesaria para la simulación, un paciente es la persona que llega enferma al hospital y a la que se le realiza su clasificación. Por lo cual esta clase hereda de la clase *persona* y aparte adquiere nuevos atributos, tales como: un número de Seguridad Social, sus tiempos de ingreso al hospital, inicio del triage, final del triage, sus signos vitales, etc. De la misma forma se definieron los getters y setters apropiados para los atributos antes descritos.
- **Clase doctor:** Otro pilar fundamental de la simulación es el doctor, este personaje se encarga de examinar y clasificar a los pacientes. Primero hay que volver a mencionar que esta clase hereda de la clase *persona* y por lo cual adquiere sus mismos atributos y métodos, además de ello se añaden algunos métodos y atributos importantes, primeramente se sabe que un médico puede tener varios pacientes (también las enfermeras pero esto se verá a continuación) y por lo tanto se genera una relación de agregación, esto porque cualquier médico puede revisar al paciente sin ser necesario uno en específico. Un método fundamental en la clase

doctor es el de poder clasificar, para ello se toman los signos vitales de un paciente, se suman y se hace una evaluación según una escala ya existente, dicha clasificación devuelve un entero indicando el nivel de urgencia del paciente.

- **Clase enfermera:** Este es otro personaje enfocado en recibir y atender a los pacientes, como se puede apreciar en el archivo de datos que se dió en conjunto con la descripción, a cada paciente que ingresa al hospital ya se le tiene asignado un médico o una enfermera que se encargan de realizar el proceso de triage. Por lo anterior se determinó que la enfermera en este caso tiene las mismas facultades de recibir y clasificar pacientes, para ello se decidió que esta clase heredó de la clase *doctor* ya que de esta forma recibiría los métodos y atributos básicos de las personas y a su vez la de los doctores.
- **Clase guardia:** Como se menciona en la descripción, existe un guardia que recibe por primera vez a los pacientes que ingresan al hospital y los registra en su reloj checador, para ello se define una clase que hereda de *persona* y se define un atributo para él, el cual es un vector con el que registra las entradas, también se definió un método para hacer checkin a la hora que llega un paciente.
- **Clase hospital:** Esta clase se encarga de conjuntar a todas las demás, se sabe que un hospital cuenta con doctores, enfermeras, guardias, etc. De la misma forma aquí llegan los pacientes para ser tratados, por lo cual se determinó que esta clase por medio de agregación usará a las otras, esto porque los doctores, enfermeras y pacientes no dependen solamente de un hospital, ya que si por alguna razón desapareciera, estos pueden irse a otro. La clase hospital tendrá un objeto cola de prioridad, esto para saber cuál es el siguiente paciente en relación al tiempo de inicio de triage. De la misma forma se cuenta con un vector de doctores y uno de enfermeras, los cuales son los que atienden a los pacientes. Para concluir se tiene el método *triage* el cual realiza todo el proceso de atender a los pacientes en cola con sus respectivos médicos y enfermeras.

III. EJEMPLO DE EJECUCIÓN

El programa recibe de entrada el archivo de datos provisto en conjunto con la tarea, realiza el *parsing* correspondiente, genera los médicos, enfermeras, pacientes y guardia necesarios y al final realiza el proceso de triage. Se podrá observar que se imprime cuando el guardia registra un paciente, así como cuando se atiende al mismo. Se imprime quién lo atiende, su información, así como el resultado de la clasificación y la firma de atención. Al concluir el proceso se imprimirá la media y la desviación estándar de los tiempos transcurridos en todos

los triages realizados. A continuación se verá el resultado de la ejecución del programa:

```

e-82037-8460025761:/Documents/Neatrix/C21/Semestre 1/programacion y algoritmos/programa11/Alvarez_855 ./notn triage_sin.dat
El guardia registró la hora de llegada del paciente: Paciente_01 a las: 00:05:42
Paciente: Paciente_01, inicia proceso de triage a las: 00:20:01
Lo atiende: Dr. Javier Barba
Registro de información básica del paciente:
Nombre: Paciente_01
Número de seguridad social: 000000000001
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Urgencia
Hora de término del triage: 00:43:01
El guardia registró la hora de llegada del paciente: Paciente_02 a las: 00:10:09
Paciente: Paciente_02, inicia proceso de triage a las: 00:30:51
Lo atiende: Enf. Beatriz Aguirre
Registro de información básica del paciente:
Nombre: Paciente_02
Número de seguridad social: 000000000002
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Urgencia menor
Hora de término del triage: 00:55:11
El guardia registró la hora de llegada del paciente: Paciente_03 a las: 00:11:43
Paciente: Paciente_03, inicia proceso de triage a las: 00:30:01
Lo atiende: Dr. Javier Barba
Registro de información básica del paciente:
Nombre: Paciente_03
Número de seguridad social: 000000000003
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Sin urgencia
Hora de término del triage: 00:55:53
El guardia registró la hora de llegada del paciente: Paciente_04 a las: 00:15:31
Paciente: Paciente_04, inicia proceso de triage a las: 01:24:01
Lo atiende: Enf. Beatriz Aguirre
Registro de información básica del paciente:
Nombre: Paciente_04
Número de seguridad social: 000000000004
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Emergencia inmediatamente
Hora de término del triage: 02:03:03
El guardia registró la hora de llegada del paciente: Paciente_05 a las: 01:00:02
Paciente: Paciente_05, inicia proceso de triage a las: 01:45:01
Lo atiende: Enf. Beatriz Aguirre
Registro de información básica del paciente:
Nombre: Paciente_05
Número de seguridad social: 000000000005
Peso: 78
Altura: 124

```

(b) Figura 2. Ejecución del programa (1/2).

```

Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Urgencia
Hora de término del triage: 02:14:23
El guardia registró la hora de llegada del paciente: Paciente_06 a las: 03:05:51
Paciente: Paciente_06, inicia proceso de triage a las: 03:19:01
Lo atiende: Enf. Beatriz Aguirre
Registro de información básica del paciente:
Nombre: Paciente_06
Número de seguridad social: 000000000006
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Emergencia inmediatamente
Hora de término del triage: 03:25:54
El guardia registró la hora de llegada del paciente: Paciente_07 a las: 03:05:18
Paciente: Paciente_07, inicia proceso de triage a las: 03:24:01
Lo atiende: Dr. Andrea Ugalde
Registro de información básica del paciente:
Nombre: Paciente_07
Número de seguridad social: 000000000007
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Emergencia inmediatamente
Hora de término del triage: 03:39:02
El guardia registró la hora de llegada del paciente: Paciente_08 a las: 05:14:04
Paciente: Paciente_08, inicia proceso de triage a las: 05:29:01
Lo atiende: Dr. Andrea Ugalde
Registro de información básica del paciente:
Nombre: Paciente_08
Número de seguridad social: 000000000008
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Urgencia
Hora de término del triage: 05:40:02
El guardia registró la hora de llegada del paciente: Paciente_09 a las: 06:13:03
Paciente: Paciente_09, inicia proceso de triage a las: 06:29:01
Lo atiende: Dr. Javier Barba
Registro de información básica del paciente:
Nombre: Paciente_09
Número de seguridad social: 000000000009
Peso: 78
Altura: 124
Presión alta: 120
Presión baja: 80
Se ha clasificado al paciente con resultado: Urgencia menor
Hora de término del triage: 06:34:24
Media (minutos): 17.6
Desviación estándar (minutos): 9.57288

```

(c) Figura 3. Ejecución del programa (2/2).

En la Figura 1. se puede ver una parte de los resultados donde se atiende paciente por paciente de acuerdo a la hora de inicio del triage, se imprime su información y se realiza la clasificación.

En la Figura 2. se aprecia la segunda parte de la clasificación pero al final se ve la media y desviación estándar (en minutos) de los tiempos obtenidos por el triage.

IV. CONCLUSIONES

En el presente reporte se analizó una simulación de un sistema de triage realizado en un hospital usando el enfoque orientado a objetos. Se pudo ver que el programa funciona de forma correcta así como que los resultados obtenidos fueron los

esperados. Como extra se propone una simulación en tiempo real usando funciones como *sleep* o *this_thread::sleep_for*, generando intervalos de tiempo escalados con respecto a los reales.