

# Informe Tarea X

Nombre de la Tarea X

Fecha: <fecha entrega>  
Autor: <nombre alumno>  
e-mail: <correo del alumno>

Figura 1: Formato para la portada.

- Portada. El formato de la portada se muestra en la Figura 1.

## 1. Introducción

## 2. Análisis del problema

El problema consiste basicamente en solucionar la simulacion de una fila de un banco, utilizando sistemas de colas de prioridad, y colas normales, para luego establecer una estadistica con los resultados de tiempo de atencion por grupo etario. Esta simulacion esta regida por un numero finito de horas (valor entero entre 1.. 12), ademas como antecedente se tiene que llega una persona a la cola cada minuto a un ritmo constante, con edad aleatoria distribuida de forma uniforme entre 11.. 100; al igual que un tiempo de atencion variable entre 20.. 140 segundos, distribuidos de forma uniforme.

Como estructuras necesarias, se plantea el uso de una cola de prioridad, una cola simple, y una estructura basica para almacenar la informacion de cada una de las personas. Abondar en como esta compuesta cada una de ellas, es trivial ya que es algo visto en la clase.

### 3. Solución del problema

#### 3.1. Algoritmo de solución

Primero, dentro de el programa solicitado, comienza con la etapa de verificación. Se validan los argumentos entregados de modo que estos no produzcan algún error propagable dentro de la ejecución. De no ser válido algún argumento, el programa termina su ejecución informándole al usuario del error.

Luego comienza la etapa de la ejecución propiamente tal. La clase “Simulador.java” se encarga de iniciar la ejecución de la simulación en el mismo constructor de la clase. Este configura las variables de la ejecución, y luego define el tipo de cola a utilizar. Luego entra a un ciclo infinito, que dependiendo del retardo que tenga provoca que la simulación tome más o menos tiempo.

Cada vez que se entra al ciclo, imprime las estadísticas correspondientes a la simulación, acto seguido, verifica si la simulación está corriendo dentro del tiempo especificado del usuario. De ser así, entonces al final de los cálculos, realiza la extracción de una persona de la cola. Independiente de que lo mencionado anteriormente ocurra, se extrae una persona de la cola, y se le consume el tiempo.

Una vez que el tiempo necesario para llevar a cabo sus operaciones se ha acabado, esta persona sale de la cola, y entra otra. De no haber ninguna, se pasa a la siguiente iteración. Cada iteración equivale a un segundo de simulación.

#### 3.2. Diagrama de Estados

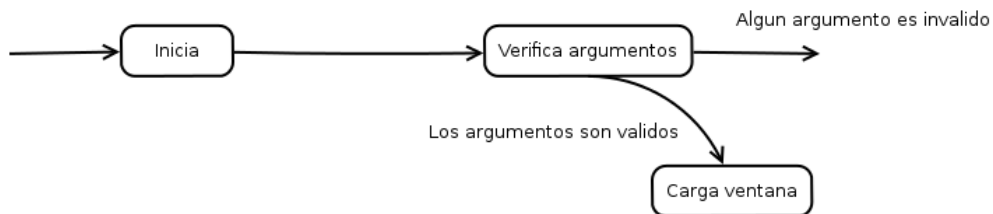


Figura 2: Inicio

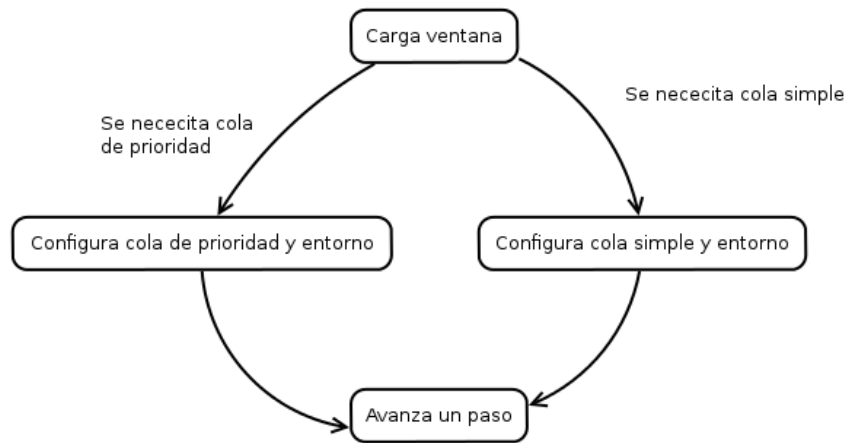


Figura 3: Configuración

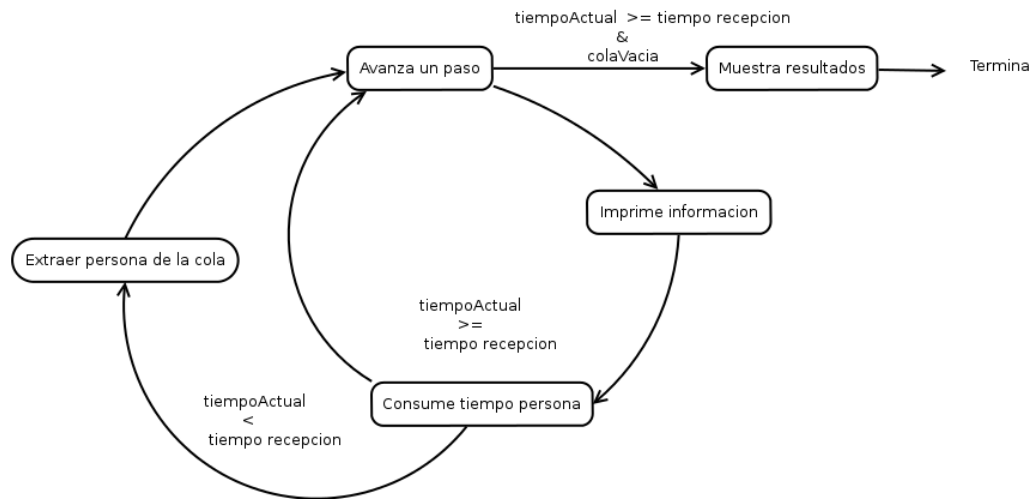


Figura 4: Simulación

### 3.3. Diseño

Explicitar las Pre y Post condiciones consideradas, mostrar los invariantes empleados.

### 3.4. Implementación

---

**Algorithm 1** computo de pasos para la simulacion

---

**Require:** cola no vacia

**Ensure:** que todas las personas de la cola hayan sido atendidas

```
imprimirInformacion()
 $step \leftarrow step + 1$ 
if  $step > maxTime$  then
    if  $actual = null$  then
        return false
    end if {e}n caso de que la persona haya terminado lo que tenia que hacer
    if  $actual.stepTimeOut()$  then
         $actual \leftarrow siguiente$ 
         $siguiente \leftarrow popCurrent()$ 
    end if
    if  $siguiente = null$  then
        return false
    end if
else
    if  $siguiente = null$  then
         $siguiente \leftarrow popCurrent()$ 
    end if
    if  $actual \neq null$  then
        if  $actual.stepTimeOut()$  then
             $actual \leftarrow siguiente$ 
             $siguiente \leftarrow popCurrent()$ 
        end if
    end if
    if  $siguiente = null$  then
        return false
    end if
    if  $step \% 60 = 0$  then
         $pushNew()$ 
    end if
end if
if  $step \% 60 = 0$  then
     $imprimeInformacion()$ 
end if
return true
```

---

### 3.5. Modo de uso

Se entrega un proyecto de netbeans, compilante con la version 7.0.1 RC2. Para poder compilar usando este IDE, se recomienda importar el proyecto. Caso contrario, se sugiere ejecutar la

compilacion desde el directorio raiz de las fuentes.

## **4. Pruebas**

Se debe mostrar las pruebas realizadas y sus resultados. El número de pruebas puede variar dependiendo del problema. En esta sección debe incluir las tablas y gráficos necesarios solicitados, y el análisis de los resultados. En este capítulo adjunten las conclusiones obtenidas de los resultados de la tarea.

## **5. Anexos**

De ser necesario, cualquier información adicional se debe agregar en los anexos y debe ser referenciada en alguna sección del informe de la tarea. Dentro de los anexos se puede incluir un listado con el programa completo que efectivamente fue compilado.