­Obligatorio : Sistema de Emergencias

1. Introducción

El Ministerio de Salud Pública solicita el diseño e implementación de la mejor solución para la asignación de ambulancias ante una emergencia en alguna ciudad del país.

Para la eficiente asignación de las ambulancias ante una emergencia, el M.S.P. cuenta con la información del tiempo que requiere llegar desde una ciudad a otra y en función de esto desea obtener un mecanismo que le permita seleccionar la ambulancia más cercana (en tiempo) para asignarle un viaje.

Una ambulancia puede ser asignada a un viaje si se encuentra en buen estado y si no se encuentra asignada a otro viaje.

Se proveen los siguientes tipos de datos que deberán ser respetados.

**Sistema**

public class Sistema{

enum tipoRet{OK,ERROR,NO\_IMPLEMENTADA};

/\*Aquí introduzca la información que estime conveniente\*/

}

Pueden definirse tipos de datos (clases) auxiliares.

**2. Funcionalidades**

**2.1. Operaciones relativas al Sistema de Emergencias**

**2.1.1. Crear Sistema de Emergencias**

**Firma:**  TipoRet crearSistemaEmergencias(int cantCiudades);

**Descripción:** Crea la estructura necesaria para representar el sistema de emergencias capaz de albergar cantCiudades ciudades.

|  |
| --- |
| **Retornos posibles** |
| OK Si el sistema de emergencias pudo ser creado exitosamente. |
| ERROR Si cantCiudades es menor o igual a 0.  Se deberá imprimir en pantalla: “La cantidad de ciudades es inferior a 1.” |
| NO\_IMPLEMENTADA Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.1.2. Destruir Sistema de Emergencias**

**Firma:** TipoRet destruirSistemaEmergencias();

Descripción: Destruye el sistema de emergencias y todos sus elementos liberando la memoria utilizada.

Retornos posibles

OK Siempre retorna OK.

ERROR No hay errores posibles.

NO\_IMPLEMENTADA Cuando aún no se implementó. Es el tipo de retorno por defecto.

**2.2. Operaciones relativas a las Ambulancias**

**2.2.1. Registrar Ambulancia**

**Firma:** TipoRet registrarAmbulancia(String ambulanciaID, int ciudadID);

Descripción: Crea una nueva ambulancia con identificador ambulanciaID, estado DISPONIBLE y la asigna a la ciudad ciudadID.

|  |
| --- |
| Retornos posibles |
| OK Si la ambulancia ambulanciaID pudo ser creada exitosamente. |
| ERROR Si ciudadID no existe. Deberá imprimir en pantalla:  “La ciudad ciudadID no existe.”  Si ambulanciaID ya existe en el sistema de emergencias.  Deberá imprimir en pantalla:  “Ya existe una ambulancia con identificador ambulanciaID |
| NO\_IMPLEMENTADA Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.2. Deshabilitar Ambulancia**

**Firma:** TipoRet deshabilitarAmbulancia(String ambulanciaID);

Descripción: Cambia el estado de la ambulancia ambulanciaID a NO\_DISPONIBLE de forma que no pueda ser asignada a un viaje.

|  |
| --- |
| **Retornos posibles** |
| **OK**  Si se pudo deshabilitar la ambulancia exitosamente. |
|  |
| **ERROR** Si ambulanciaID no existe en el sistema de emergencias,  deberá imprimir en pantalla:  “No existe una ambulancia con identificador ambulanciaID.”  Si la ambulancia ambulanciaID está en estado NO\_DISPONIBLE, deberá imprimir en pantalla:  “La ambulancia ambulanciaID ya está en estado NO\_DISPONIBLE.”  Si la ambulancia ambulanciaID está asignada a un viaje, deberá imprimir en pantalla:  “No es posible deshabilitar la ambulancia ambulanciaID”. |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.3. Habilitar Ambulancia**

**Firma:**

TipoRet habiliatarAmbulancia(String ambulanciaID);

Descripción: Habilita a la ambulancia ambulanciaID para estar nuevamente operativa. Una ambulancia está no disponible para aceptar un viaje si fue deshabilitada (y no se habilitó nuevamente) o si está asignada a una emergencia.

|  |
| --- |
| **Retornos posibles** |
| **OK**  Si se pudo habilitar la ambulancia ambulanciaID exitosamente. |
| **ERROR** Si la ambulancia ambulanciaID no existe en el sistema deemergencias,  deberá imprimir en pantalla:  “No existe una ambulancia con identificador ambulanciaID.”  Si la ambulancia ambulanciaID está habilitada, deberá imprimir enpantalla:  “La ambulancia ambulanciaID ya está habilitada.” |
| NO\_IMPLEMENTADA Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.4. Eliminar Ambulancia**

**Firma:**

TipoRet eliminarAmbulancia(String ambulanciaID);

Descripción: Destruye la ambulancia ambulanciaID y todos sus elementos y referencias liberando la memoria utilizada.

|  |
| --- |
| **Retornos posibles** |
| **OK** Si se pudo eliminar la ambulancia ambulanciaID exitosamente. |
| **ERROR**  Si la ambulancia ambulanciaID no existe en el sistema de emergencias,  deberá imprimir en pantalla:  “No existe una ambulancia con identificador ambulanciaID.”  Si la ambulancia ambulanciaID está asignada a un viaje, deberá imprimir  en pantalla: “No es posible eliminar la ambulancia ambulanciaID |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.5. Buscar Ambulancia**

**Firma:** TipoRet buscarAmbulancia(String ambulanciaID)

Descripción: Imprime por pantalla los datos de la ambulancia ambulanciaID en pantalla. Los datos a desplegar deben ser los siguientes respetando el formato especificado:

Datos ambulancia: ambulanciaID

Estado: estado\_ambulancia

Ciudad: ciudad\_ambulancia

#Emergencias: cantidad\_emergencias

**Ejemplo:**

Datos ambulancia: RAM1392

Estado: NO\_DISPONIBLE

Ciudad: Paso de los Toros

#Emergencias: 24

|  |
| --- |
| **Retornos posibles** |
| **OK** Si se pudo encontrar la ambulancia ambulanciaID exitosamente. |
| **ERROR** Si la ambulancia ambulanciaID no existe en el sistema de  emergencias, deberá imprimir en pantalla:  “No existe una ambulancia con identificador ambulanciaID.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.6. Informe de Ambulancia**

**Firma:** TipoRet informeAmbulancia();

Descripción: Imprime por pantalla todas las ambulancias ordenadas según ambulanciaID (matrícula), detallando su estado y la ciudad donde se encuentra. Si no se han ingresado ambulancias deberá imprimir: “No se han registrado ambulancias.”

|  |
| --- |
| **Retornos posibles** |
| **OK** Siempre retorna OK. |
| **ERROR** No hay errores posibles. |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de impresión deberá ser el siguiente:

Listado de ambulancias:

ambulanciaID> - <estado\_ambulancia> - <ciudad\_ambulancia>

…..

ambulanciaID> - <estado\_ambulancia> - <ciudad\_ambulancia>

**Ejemplo 1:**

**Listado de ambulancias:**

No se han registrado ambulancias.

**Ejemplo 2:**

**Listado de ambulancias:**

AAM1629 – NO\_DISPONIBLE - Pando

AAM3902 – DISPONIBLE – Las Piedras

AAM6314 – ATENDIENDO\_EMERGENCIA

BAM0621 – DISPONIBLE – Pan de Azúcar

CAM0472 – DISPONIBLE – Las Flores

FAM1434 – ATENDIENDO\_EMERGENCIA

GAM3194 – ATENDIENDO\_EMERGENCIA

RAM1392 – NO\_DISPONIBLE – Paso de los Toros

SAM0341 – ATENDIENDO\_EMERGENCIA

SAM1712 – ATENDIENDO\_EMERGENCIA

SAM6524 – NO\_DISPONIBLE - Montevideo

**2.2.7. Ambulancias por Ciudad**

**Firma:** TipoRet informeAmbulancia (int ciudadID);

Descripción: Imprime por pantalla todas las ambulancias disponibles que se encuentran en la ciudad ciudadID (ordenadas por ambulanciaID de forma ascendente)

|  |
| --- |
| **Retornos posibles** |
| **OK** Si se pudo imprimir el informe existosamente. |
| **ERROR** Si ciudadID no existe. Deberá imprimir en pantalla:  “La ciudad ciudadID no existe.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de impresión deberá ser el siguiente:

**Informe de ambulancias en: <ciudadID> - <nombreCiudad>**

Ambulancia: ambulanciaID

…

Ambulancia: ambulanciaID

Total Ambulancias disponibles: cant\_ambulancias

**Ejemplo 1:**

**Informe de ambulancias en:**  14 - Colonia

No hay ambulancias disponibles en la ciudad.

**Ejemplo 2:**

**Informe de ambulancias en: 12 - Montevideo**

AAM3902

CAM0472

SAM6524

Total Ambulancias disponibles: 3

**2.2.8. Cambiar Ubicación de una Ambulancia**

Firma: TipoRet cambiarUbicacion(String ambulanciaID, int ciudadID);

Descripción: Al atender una emergencia la ambulancia finaliza en una ciudad distinta a la de origen por lo que es necesario cambiar su ubicación. Este método cambia la ubicación de la ambulancia asignándola a la ciudad ciudadID. Se deberá quitar la ambulancia ambulanciaID de la ciudad donde se encontrab

|  |
| --- |
| **Retornos posibles** |
| **OK** Si se pudo cambiar la ubicación de la ambulancia ambulanciaID exitosamente. |
| **ERROR** Si ciudadID no existe. Deberá imprimir en pantalla: “La ciudad ciudadID no  existe.”  Si la ambulancia ambulanciaID no existe en el sistema de emergencias, deberá  imprimir en pantalla: “No existe una ambulancia con identificador  ambulanciaID.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2. Operaciones relativas al Mapa**

**2.2.1. Agregar Ciudad al Mapa**

**Firma:** TipoRet agregarCiudad(String ciudadNombre);

Descripción: El sistema deberá permitir crear ciudades y para cada una de ellas deberá asignarle un identificador único (el menor valor libre no asignado a otra ciudad) entre 1 y cantCiudades.

|  |
| --- |
| **Retornos posibles** |
| **OK** Si se pudo crear la ciudad *ciudadNombre* exitosamente. |
| **ERROR** Si ya hay registradas cantCiudades en el Sistema de emergencias  Se deberá imprimir en pantalla: “No se pueden ingresar la *ciudadNombre* al  sistema por no tener más capacidad.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.2. Listar Ciudades**

**Firma:** TipoRet listarCiudades();

Descripción: Imprime en pantalla las ciudades ordenadas de forma ascendente por ciudadID.

|  |
| --- |
| **Retornos posibles** |
| **OK** Siempre retorna OK. |
| **ERROR** No hay errores posibles. |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de salida deberá ser el siguiente:

**Ciudades en el Mapa:**

ID> - <nombreCiudad>

----

ID> - <nombreCiudad>

En caso de que no existan ciudades en el mapa se deberá imprimir en pantalla:

No existen ciudades en el mapa.

**2.2.3. Agregar Ruta al Mapa**

**Firma:** TipoRet agregarRuta(int ciudadOrigen, int ciudadDestino, int minutosViaje);

Descripción: Ingresa la ruta desde ciudadOrigen a ciudadDestino con demora minutosViaje. Tenga en cuenta que al crear una ruta desde un origen se deberá también agregar la ruta ciudadDestino a ciudadOrigen de igual demora.

|  |
| --- |
| **Retornos posibles** |
| **OK** Si se pudo agregar la ruta exitosamente. |
| **ERROR** Si ciudadOrigen no existe. Deberá imprimir en pantalla: “La ciudad  ciudadOrigen no existe.”  Si ciudadDestino no existe. Deberá imprimir en pantalla: “La ciudad  ciudadDestino no existe.”  Si minutosViaje es menor o igual a 0. Deberá imprimir en pantalla: “La  duración del viaje debe ser mayor que 0.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.4. Modificar demora de viaje**

**Firma:** TipoRet modificarDemora(int ciudadOrigen, int ciudadDestino, int minutosViaje);

Descripción: Modifica la demora del viaje desde ciudadOrigen a ciudadDestino (y viceversa).

|  |
| --- |
| **Retornos posibles** |
| **OK**  Si se pudo modificar la demora exitosamente. |
| **ERROR**  Si ciudadOrigen no existe. Deberá imprimir en pantalla:  “La ciudad ciudadOrigen no existe.”  Si ciudadDestino no existe. Deberá imprimir en pantalla:  “La ciudad ciudadDestino no existe.”  Si minutosViaje es menor o igual a 0. Deberá imprimir en pantalla:  “La duración del viaje debe ser mayor que 0.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.2.5. Ambulancia más cercana**

**Firma:** TipoRet ambulanciaMasCercana(int ciudadID);

Descripción: Muestra por pantalla los datos de la ambulancia más cercana a la ciudad ciudadID y el tiempo que demoraría en llegar. Si hay alguna ambulancia en la ciudad ciudadID el tiempo de demora será 0.

|  |
| --- |
| **Retornos posibles** |
| **OK** Si la operación se pudo efectuar correctamente. |
| **ERROR** Si ciudadID no existe. Deberá imprimir en pantalla: “La ciudad ciudadID  no existe.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de impresión deberá ser el siguiente:

**Ambulancia más cercana a: <ciudadID> - <nombreCiudad>**

Ambulancia: ambulanciaID>

Demora del viaje: <minutosViaje>

**Ejemplo 1:**

**Ambulancia más cercana a: 14 - Colonia**

No hay ambulancias creadas.

**Ejemplo 2:**

**Ambulancia más cercana a:12 – Montevideo**

Ambulancia: SAM6524

Demora del viaje: 0

**2.2.6. Ruta más rápida**

**Firma:** TipoRet rutaMasRapida(int ciudadOrigen, int ciudadDestino);

Descripción: Muestra por pantalla la ruta más rápida para ir desde la ciudad ciudadOrigen a la ciudad ciudadDestino. Deberá desplegar las ciudades intermedias (limitando a una ciudad intermedia) y el tiempo estimado de cada tramo y el viaje completo.

|  |
| --- |
| **Retornos posibles** |
| **OK** Si la operación se pudo efectuar existosamente. |
| **ERROR** Si ciudadOrigen no existe. Deberá imprimir en pantalla: “La ciudad  ciudadOrigen no existe.”  Si ciudadDestino no existe. Deberá imprimir en pantalla: “La ciudad  ciudadDestino no existe.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de impresión deberá ser el siguiente:

**Ruta más rápida:**

**nombreCiudadOrigen> - < duración\_viaje >**

nombreCiudad> - < duración\_viaje >

…

nombreCiudad> - < duración\_viaje >

nombreCiudadDestino> - <duración\_viaje>

Demora total Ambulancias: <demora\_viaje>

**Ejemplo 1:**

**Ruta más rápida:**

No hay ruta desde Paso de los Toros a Colonia

**Ejemplo 2:**

**Ruta más rápida:**

Montevideo - 0

Las Piedras – 32

Canelones – 54

Trinidad - 62

Demora total Ambulancias: 148

**Nota:** Recordar que debe limitar el viaje a solo una ciudad intermedia, por lo que este ejmeplo se descartaría.

**2.2.7. Informe de Ciudades**

**Firma:** TipoRet informeCiudades();

Descripción: Imprime por pantalla todas las ciudades detallando las ciudades con las que tiene ruta directa y la demora del viaje. También deberá detallar la cantidad de ambulancias según su estado que se encuentran cada ciudad.

|  |
| --- |
| **Retornos posibles** |
| **OK** Siempre retorna OK. |
| **ERROR** No hay errores posibles. |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de impresión deberá ser el siguiente:

**Informe Ciudad: <ciudadID>**

Ruta directa a <ciudadID>, minutos <minutos\_viaje>

Ambulancias disponibles: <cant\_ambulancias\_disponibles>

Ambulancias no disponibles: <cant\_ambulancias\_no\_disponibles>

**2.2.8. Informe de Ciudades en un radio**

**Firma:** TipoRet ciudadesEnRadio(int ciudadID, int duracionViaje);

Descripción: Imprime por pantalla todas las ciudades que se encuentran a duracionViaje minutos o menos de la ciudad ciudadID.

Retornos posibles

OK Si se pudo efectuar la operación exitosamente.

ERROR Si ciudadID no existe. Deberá imprimir en pantalla: “La ciudad ciudadID

no existe.”

Si duracionViaje es menor o igual a 0. Deberá imprimir en pantalla: “La

duración del viaje debe ser mayor que 0.”

NO\_IMPLEMENTADA Cuando aún no se implementó. Es el tipo de retorno por defecto.

El formato de impresión deberá ser el siguiente:

**Ciudades en radio de <duracionViaje> minutos:**

Ciudad <ciudadID1> a <duracionViaje1> minutos

…

Ciudad <ciudadIDN> a <duracionViajeN> minutos

**2.3. Operaciones relativas a los Choferes**

Cada ambulancia contará con una lista de choferes habilitados. De cada chofer interesará saber su Nombre y Cédula de Identidad.

**2.3.1. Registrar Chofer**

**Firma:** TipoRet registrarChofer(String ambulanciaID, String nombre, String cedula);

**Descripción:** Registra al chofer de nombre nombre y cédula cédula como chofer habilitado para conducir la ambulancia ambulanciaID.

**Precondición:** No existe un chofer de cédula cedula como chofer habilitado para conducir la ambulancia ambulanciaID.

Retornos posibles

OK Si se pudo efectuar la operación exitosamente.

ERROR Si la ambulancia ambulanciaID no existe en el sistema de emergencias,

deberá imprimir en pantalla: “No existe una ambulancia con

identificador ambulanciaID.”

NO\_IMPLEMENTADA Cuando aún no se implementó. Es el tipo de retorno por defecto.

**2.3.2. Eliminar Chofer**

**Firma:** TipoRet eliminarChofer(String ambulanciaID, String cedula);

**Descripción:** Elimina al chofer de cédula cedula como chofer habilitado para conducir la ambulancia ambulanciaID.

|  |
| --- |
| **Retornos posibles** |
| **OK**  Si se pudo efectuar la operación exitosamente. |
| **ERROR**  Si la ambulancia ambulanciaID no existe en el sistema de emergencias,  deberá imprimir en pantalla: “No existe una ambulancia con  identificador ambulanciaID.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

**2.3.3. Informe de Choferes Habilitados**

**Firma:**

TipoRet informeChoferes(String ambulanciaID);

**Descripción:** MImprime en pantalla un informe con todos los choferes habilitados para conducir la

ambulancia ambulanciaID.

|  |
| --- |
| **Retornos posibles** |
| **OK**  Si se pudo efectuar la operación exitosamente. |
| **ERROR**  Si la ambulancia ambulanciaID no existe en el sistema de emergencias,  deberá imprimir en pantalla: “No existe una ambulancia con  identificador ambulanciaID.” |
| **NO\_IMPLEMENTADA** Cuando aún no se implementó. Es el tipo de retorno por defecto. |

El formato de impresión deberá ser el siguiente:

**Informe Choferes de <ambulanciaID>**

Nombre: <nombre\_chofer1>, Cédula: <cedula\_chofer1>

…

Nombre: <nombre\_choferN>, Cédula: <cedula\_choferN>

**Información importante**

* Lectura de Obligatorio: semana 6)
* Puntaje mínimo/máximo: 1/50
* Los obligatorios se realizan por equipos de 2 estudiantes.
* Plazo máximo de segunda entrega (con boleta): semana 14
* Se deberán respetar los formatos de impresión dados para las operaciones que imprimen en consola.
* El resto de las operaciones no deben imprimir nada en consola.
* El sistema no debe requerir ningún tipo de interacción con el usuario por consola.
* Es obligación del estudiante mantenerse al tanto de las aclaraciones que se realicen en clase o a través del foro de aulas.
* Se valorará la selección adecuada de las estructuras para modelar el problema y la eficiencia en
* cada una de las operaciones.
* Deberá aplicar la metodología vista en el curso.
* Deberá entregar dos copias en CD: conteniendo el código fuente y la documentación (en pdf con las fotos de los integrantes) que entregó impresa.
* Deberá entregar impreso: La las pre y post condiciones de los métodos solicitados, un diagrama de la estructura de datos que se implementó para representar el sistema de emergencias junto con una breve explicación indicando por qué eligió dichas estructuras y las pruebas realizadas y el resultado obtenido.
* Para la presentación de la documentación se publicará en aulas.ort.edu.uy un template. (El uso de este template es obligatorio).
* El proyecto será implementado en lenguaje JAVA sobre una interfaz que se publicará en el sitio de la materia en aulas.ort.edu.uy (El uso de esta interfaz es obligatorio).

**Utilizar las siguiente interface:**

**public** interface Sistema {

**public** enum TipoRet {OK, NO\_IMPLEMENTADA, ERROR;}

**public** TipoRet eliminarAmbulancia(String ambulanciaID);

**public** TipoRet crearSistemaDeEmergencias(**int** cantidadCiudades);

**public** TipoRet destruirSistemaEmergencias();

**public** TipoRet registrarAmbulancia(String ambulanciaId, int ciudadID);

**public** TipoRet deshabilitarAmbulancia(String ambulanciaId);

**public** TipoRet habilitarAmbulancia(String ambulanciaID);

**public** TipoRet buscarAmbulancia(String ambulanciaID);

**public** TipoRet informeAmbulancia();

**public** TipoRet informeAmbulancia(**int** ciudadID);

**public** TipoRet cambiarUbicacion(String ambulanciaID, **int** ciudadID);

**public** TipoRet agregarCiudad(String ciudadNombre);

**public** TipoRet listarCiudades();

**public** TipoRet agregarRuta(**int** ciudadOrigen, **int** ciudadDestino, **int** minutosViaje);

**public** TipoRet modificarDemora(**int** ciudadOrigen, **int** ciudadDestino, **int** minutosViaje);

**public** TipoRet ambulanciaMasCercana(**int** ciudadID);

**public** TipoRet rutaMasRapida(**int** ciudadOrigen, **int** ciudadDestino);

**public** TipoRet informeCiudades();

**public** TipoRet ciudadesEnRadio(**int** ciudadID, **int** duracionViaje);

**public** TipoRet registrarChofer(String ambulanciaID, String nombre, String cedula);

**public** TipoRet eliminarChofer(String ambulanciaID, String cedula);

**public** TipoRet informeChoferes(String ambulanciaID);

}