

EVALUACION	Obligatorio	GRUPO	Todos	FECHA	MAYO 2015
MATERIA	Algoritmos y Estructuras de Datos 2				
CARRERA	AP, ATI				
CONDICIONES	<p>- Puntos: Máximo: 50 Mínimo: 1</p> <p>- Fecha máxima de entrega:</p> <p>- Horario de Coordinación adjunta para entregas: de lunes a jueves de 8:00 a 12:30 y de 17:30 a 19:00 hs.</p> <p>IMPORTANTE</p> <ul style="list-style-type: none"> - Los grupos deben estar conformados por dos personas. - Inscribirse (sacar la "boleta de entrega"). - Entregar en carpeta con elástico. - Entregar 2 copias en cd, donde se incluya la documentación en pdf con foto de los integrantes. - No entregue el código impreso. - Etiquetar documentación y cd con nombre, n° estudiante, carrera, grupo, materia y docente 				

SISTEMA DE GESTIÓN INMOBILIARIO

Introducción

El lugar en el que se ubica una propiedad inmobiliaria, es una de las principales variables que inciden en su precio de venta o alquiler. No solamente el barrio, sino también la cercanía a ciertos puntos de interés, como por ejemplo el acceso al transporte público, centros comerciales o de negocios, y zonas de alto interés o crecimiento.

Es por esto, que se desea desarrollar un sistema para una inmobiliaria de la ciudad de Montevideo, que permita manejar la publicación de propiedades a la venta o alquiler y sus características usuales, pero además brinde información geográfica de la misma, y su relación (por ejemplo la distancia) con puntos de interés como los nombrados antes.

Cuando se ingresa una propiedad al sistema automáticamente se la asigna a un vendedor, el cuál será seleccionado mediante una política de asignación FIFO.

Generalidades

Se define una clase Retorno, la cual se utilizará como tipo de retorno para todas las operaciones del sistema. Dicha clase contiene:

- Un resultado, que especifica si la operación se pudo realizar correctamente (OK), o si ocurrió algún error (según el número de error).
- Un valor entero, para las operaciones que retornen un número entero.
- Un valor String, para las operaciones que retornen un String, o un valor más complejo (por ejemplo una lista o clase), la cuál será formateada según lo indicado en el Anexo I de este documento.

Se proveen una interfaz llamada ISistema, la cual no podrá ser modificada en ningún sentido, y una clase sistema que la implemente, donde el estudiante deberá completar la implementación de las operaciones solicitadas.

Sistema	<pre>public class Sistema implements ISistema{ /* Aquí van las operaciones del sistema */ }</pre>
Retorno	<pre>public class Retorno { enum Resultado {OK, ERROR_1, ERROR_2, ERROR_3, ERROR_4, ERROR_5, NO_IMPLEMENTADA}; int valorEntero; string valorString; Resultado resultado; ... }</pre>

Además, se provee un enumerado con tipos de puntos, rubros y tipos de propiedad para facilitar la tarea.

Pueden definirse los tipos de datos (clases) auxiliares que crea necesario, SIN MODIFICAR LA INTERFAZ DADA.

Funcionalidades

1. Operaciones globales

1.1. Inicializar Sistema Gestión Inmobiliaria

Firma: `Retorno inicializarSistema (int cantPuntos);`

Descripción: Inicializa las estructuras necesarias para representar el sistema especificado, capaz de albergar como máximo *cantPuntos* nodos en el mapa.

Resultados posibles	
OK	<ul style="list-style-type: none">Si el sistema pudo ser inicializado exitosamente.
ERROR	<ul style="list-style-type: none">1. Si <i>cantPuntos</i> es menor o igual a 0.
NO_IMPLEMENTADA	<ul style="list-style-type: none">Cuando aún no se implementó. Es el tipo de retorno por defecto.

1.2. Destruir Sistema

Firma: `Retorno destruirSistema();`

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

Resultados posibles	
OK	<ul style="list-style-type: none">Siempre retorna OK.
ERROR	<ul style="list-style-type: none">No hay errores posibles.
NO_IMPLEMENTADA	<ul style="list-style-type: none">Cuando aún no se implementó. Es el tipo de retorno por defecto.

2. Operaciones relativas al mapa

2.1. Registrar punto de interés en el mapa

Los puntos a registrar en el mapa serán los puntos de interés con su respectivo rubro y su nombre.

Las propiedades también serán puntos a registrar en el mapa pero se ingresarán mediante otra operación.

Firma: `Retorno registrarPuntoInteres(Double coordX, Double coordY, TipoRubro rubro, String nombre);`

Descripción: Registra el punto en el sistema.

Resultados posibles	
OK	<ul style="list-style-type: none">Si el punto fue registrado exitosamente.
ERROR	<ul style="list-style-type: none">1. Si en el sistema ya hay registrados <i>cantPuntos</i> puntos.2. Si el nombre del punto de interés es vacío o nulo3. Si el punto de coordenadas <i>coordX</i>, <i>coordY</i> ya existe en el sistema.
NO_IMPLEMENTADA	<ul style="list-style-type: none">Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.2. Registrar esquina en el mapa

Mediante esta operación se registrarán las esquinas del mapa de la ciudad.

Firma: `Retorno registrarEsquina(Double coordX, Double coordY);`

Descripción: Registra la esquina en el sistema.

Resultados posibles	
OK	<ul style="list-style-type: none">Si el punto fue registrado exitosamente.
ERROR	<ul style="list-style-type: none">1. Si en el sistema ya hay registrados <i>cantPuntos</i> puntos.

	<ul style="list-style-type: none"> 2. Si el punto de coordenadas <i>coordX</i>, <i>coordY</i> ya existe en el sistema.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.3. Registrar una propiedad en el mapa

Registra la propiedad en el sistema asignándole automáticamente un vendedor que la gestione. El vendedor asignado será aquel que hace más tiempo que no se le asigna una propiedad.

Firma: Retorno registrarPropiedad(Double coordX, Double coordY, TipoPropiedad tipo, String direccion);

Descripción: Registra la propiedad en el sistema.

Resultados posibles	
OK	<ul style="list-style-type: none"> Si la propiedad fue registrada exitosamente.
ERROR	<ul style="list-style-type: none"> 1. Si en el sistema ya hay registrados <i>cantPuntos</i> puntos. 2. Si la dirección es vacía o nula 3. Si el punto de coordenadas <i>coordX</i>, <i>coordY</i> ya existe en el sistema.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.4. Eliminar punto del mapa

Firma: Retorno eliminarPunto(Double coordX, Double coordY);

Descripción: Elimina el punto en el sistema (sin importar si es una propiedad, punto de interés, o una esquina). Si existen tramos conectados con el punto, deberán ser eliminados también.

Resultados posibles	
OK	<ul style="list-style-type: none"> Si el punto fue eliminado exitosamente.

ERROR	<ul style="list-style-type: none"> 1. Si el punto de coordenadas <i>coordX</i>, <i>coordY</i> no existe en el sistema.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.5. Registrar tramo

Firma: Retorno registrarTramo(Double coordXi, Double coordYi, Double coordXf, Double coordYf, int peso);

Descripción: Registra un tramo en el sistema desde la coordenada inicio (*coordXi*, *coordYi*) hasta la coordenada destino (*coordXf*, *coordYf*) de peso *peso*. El tramo se puede registrar entre puntos de interés, propiedades, esquinas, o cualquier combinación de ellos.

Nota: Los tramos serán considerados sin dirección por lo cual al ingresar el tramo del punto A al punto B, también deberá ser posible navegar desde el punto B al punto A.

Resultados posibles	
OK	<ul style="list-style-type: none"> Si el tramo pudo ser registrado exitosamente.
ERROR	<ul style="list-style-type: none"> 1. Si peso es menor o igual a 0. 2. Si no existe algunos de los puntos <i>coordi</i> o <i>coordf</i>. 3. Si ya existe el tramo <i>coordi</i> - <i>coordf</i> en el sistema.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.6. Eliminar tramo

Firma: Retorno eliminarTramo(Double coordXi, Double coordYi, Double coordXf, Double coordYf);

Descripción: Elimina el tramo desde la coordenada inicio (*coordXi*, *coordYi*) hasta la coordenada destino (*coordXf*, *coordYf*).

Resultados posibles

OK	<ul style="list-style-type: none"> Si el tramo pudo ser eliminado exitosamente.
ERROR	<ul style="list-style-type: none"> 1. Si no existe alguno de los puntos <i>coordi</i> o <i>coordf</i>. 2. Si no existe un tramo desde <i>coordi</i> a <i>coordf</i>.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> Cuando aún no se implementó. Es el tipo de retorno por defecto.

Nota: Como los tramos son no dirigidos, al eliminar el tramo desde el punto A al punto B, también deberá eliminarse el tramo del punto B al punto A.

3. Operaciones relativas a los vendedores

3.1. Registrar vendedor

Firma: Retorno `registrarVendedor(String cedula, String nombre, String email, String celular);`

Descripción: Registra al vendedor de cédula “cedula” en el sistema. La operación deberá controlar que el email, el número de celular y la cédula de identidad tengan un formato correcto.

Resultados posibles	
OK	<ul style="list-style-type: none"> Si el vendedor pudo ser registrado exitosamente.
ERROR	<ul style="list-style-type: none"> 1. Si la cédula no cumple el formato N.NNN.NNN-N 2. Si el número de celular no cumple el formato 09NNNNNNNN 3. Si el email no cumple el formato de direcciones de e-mail 4. Si el vendedor de cédula “cedula” ya está registrado en el sistema.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> Cuando aún no se implementó. Es el tipo de retorno por defecto.

Se recomienda el uso de **expresiones regulares** para lograr fácilmente el control de los formatos. Ver Anexo con links de interés.

3.2. Eliminar vendedor

Firma: `Retorno eliminarVendedor(String cedula);`

Descripción: Elimina al vendedor de cédula “cedula” del sistema. Las propiedades que estén asignadas al mismo, deberán ser reasignadas, utilizando la misma estrategia que cuando se ingresa una nueva propiedad. Si el vendedor a eliminar tiene N propiedades asignadas, se asignarán dichas propiedades a los primeros N vendedores del pool.
En caso de que el vendedor que se elimina sea el único vendedor del sistema, las propiedades asignadas a éste no se podrán reasignar (porque no quedan otros vendedores). En ese caso, se deberá retornar error.

Resultados posibles	
OK	<ul style="list-style-type: none">Si el vendedor pudo ser eliminado exitosamente.
ERROR	<ul style="list-style-type: none">1. Si el vendedor de cédula “cedula” no está registrado en el sistema.2. Si no se pueden reasignar las propiedades por no haber más vendedores registrados
NO_IMPLEMENTADA	<ul style="list-style-type: none">Cuando aún no se implementó. Es el tipo de retorno por defecto.

3.3. Listado de vendedores

Firma: `Retorno listadoVendedores();`

Descripción: Retorna en valorString un listado de los vendedores registrados en el sistema, ordenados por CI en forma ascendente. El string de salida deberá tener el formato:

CIvendedor1;nombreVend1;celularVend1|CIvendedor2;nombreVend2;celularVend2|...

Ejemplo:

1.234.567-8;Pablo;099123456|2.345.678-9;Juan;091999111|3.123.123-9;Jose;098111222

Resultados posibles	
OK	<ul style="list-style-type: none">Siempre retorna OK.
ERROR	<ul style="list-style-type: none">No hay errores posibles

NO_IMPLEMENTADA	<ul style="list-style-type: none"> · Cuando aún no se implementó. Es el tipo de retorno por defecto.
------------------------	---

4. Reportes

4.1. Mapa de propiedades, puntos de interés y esquinas

Firma: `Retorno verMapa();`

Descripción: Abre una ventana del navegador y muestra en un mapa de Google Maps todos los puntos registrados en el mapa: propiedades, puntos de interés y esquinas. Marcar con un color diferente según el tipo de propiedad.

Resultados posibles	
OK	<ul style="list-style-type: none"> · Siempre retorna OK.
ERROR	<ul style="list-style-type: none"> · No hay errores posibles.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> · Cuando aún no se implementó. Es el tipo de retorno por defecto.

4.2. Punto de interés más cercano

Firma: `Retorno PuntoMasCercano(String direccion, Rubro rubro);`

Descripción: Devuelve en valorString las coordenadas del punto de interés más cercano a la dirección especificada, siendo el punto del rubro dado. El formato deberá ser: coordX;coordY

Resultados posibles	
OK	<ul style="list-style-type: none"> · Si el reporte pudo ejecutar correctamente.
ERROR	<ul style="list-style-type: none"> · 1. Si la propiedad de dirección "direccion" no existe en el sistema. · 2. Si no hay ningún punto de interés del rubro "tipo" en el sistema.

NO_IMPLEMENTADA	<ul style="list-style-type: none"> • Cuando aún no se implementó. Es el tipo de retorno por defecto.
------------------------	---

4.3. Camino mínimo a punto de interés

Firma: Retorno `caminoMinimo(String direccion, Double coordX, Double coordY);`

Descripción: Retorna el camino mínimo para ir desde la propiedad especificada por su dirección hasta el punto de interés determinado por sus coordenadas. El camino deberá estar formado por todos los puntos del mapa que forman el camino, incluyendo la propiedad de inicio y el punto de llegada.

Resultados posibles	
OK	<ul style="list-style-type: none"> • Si el reporte pudo ejecutar correctamente.
ERROR	<ul style="list-style-type: none"> • 1. Si la propiedad de dirección "direccion" no existe en el sistema. • 2. Si el punto de interés de coordenadas coordX, coordY no existe en el sistema. • 3. En caso de no encontrar ningún camino posible entre la propiedad y el punto de interés.
NO_IMPLEMENTADA	<ul style="list-style-type: none"> • Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato de salida para imprimir las coordenadas del camino mínimo:

Ejemplo: -30.4;-30.4|-35.4;-36.4|-35.3;-36.3|-35.3;-36.21|-38.1;-34.1

El camino debe contener el punto origen y el punto final.

4.4. Listado de propiedades

Firma: Retorno `listadoPropiedades(String cedula);`

Descripción: Retorna un listado con las coordenadas de todas las propiedades gestionadas por el vendedor especificado por su cédula.

Resultados posibles	
OK	<ul style="list-style-type: none">• Si el reporte pudo ejecutar correctamente.
ERROR	<ul style="list-style-type: none">• 1. Si el vendedor de cédula “cedula” no existe en el sistema.• 2. Si el vendedor de cédula “cedula” no tiene ninguna propiedad asignada.
NO_IMPLEMENTADA	<ul style="list-style-type: none">• Cuando aún no se implementó. Es el tipo de retorno por defecto.

Formato de salida para imprimir las coordenadas de las propiedades:

Ejemplo: -30.4;-30.4|-35.4;-36.4|-35.3;-36.3|-35.3;-36.21|-38.1;-34.1

Información importante

- Se deberán **respetar los formatos de retorno** dados para las operaciones que devuelven datos.
- **Ninguna** de las operaciones 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.
- Puntos acerca de la documentación impresa:
 - **Incluir las pre y post condiciones de los métodos solicitados**
 - **Incluir un diagrama de la estructura de datos que se implementó para representar el sistema junto con una breve explicación indicando por qué eligió dichas estructuras**
 - **La documentación tiene un valor de 10 puntos del total del obligatorio, que podrán restarse en caso de estar incompleta o mal presentada.**
 - **NO incluir el código fuente.**
- 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 ISistema que se publicará en el sitio de la materia en aulas.ort.edu.uy (El uso de esta interfaz es obligatorio, y **NO DEBE SER MODIFICADA**).
- El proyecto entregado debe compilar y ejecutar correctamente en Eclipse.
- No se contestarán dudas sobre el obligatorio en las 48 horas previas a la entrega.
- No se contestarán dudas a través del mail del docente. Las preguntas se deberán hacer en el foro de consultas de aulas.

Anexo I: Formatos de retorno

Para las operaciones en las que se debe retornar un tipo complejo (varios valores, o una colección de valores), se define el siguiente formato de manera de serializar el valor y encapsularlo en un único String.

- Si el valor a retornar es una colección de datos, se separarán cada ítem de la colección por un carácter "|".
- Si el valor a retornar es un tipo complejo y tiene más de un atributo (por ejemplo la CI y nombre), se separarán ambos valores por un carácter ";"

Ejemplos:

Retornar la CI y el nombre de una persona:

32551567;Fernando

Retornar una colección de nombres:

Fernando|Esteban|Fabián

Retornar una colección de personas, con sus CI y nombres:

32551567;Fernando|1234567;Esteban|98765432;Fabián

Anexo II: Información útil

Expresiones regulares:

<http://www.mkyong.com/regular-expressions/how-to-validate-email-address-with-regular-expression/>
<http://regexpal.com/>

Crear mapas de Google Maps con marcadores:

<https://developers.google.com/maps/documentation/staticmaps/?csw=1>

Ejemplo:

<http://maps.googleapis.com/maps/api/staticmap?center=Montevideo,Uruguay&zoom=13&size=1200x600&mapttype=roadmap&markers=color:blue%7Clabel:1%7C-34.90,-56.16&markers=color:red%7Clabel:2%7C-34.91,-56.17&markers=color:green%7Clabel:3%7C-34.905,-56.19&sensor=false>

Parsear un string:

<http://stackoverflow.com/questions/3481828/how-to-split-a-string-in-java>