

EVALUACION	Obligatorio	GRUPO	TODOS	FECHA	XX/05/2014
MATERIA	Algoritmos y Estructuras de Datos 1				
CARRERA	Analista Programador / Analista en Tecnologías de la información				
CONDICIONES	<p>- Puntos: Máximo: 50 Mínimo: 1</p> <p>- Duración: 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> <p>- Los grupos deben estar conformados por dos personas.</p> <p>- Inscribirse (sacar la “boleta de entrega”).</p> <p>- Entregar en carpeta con elástico.</p> <p>- Entregar 2 copias en cd, donde se incluya la documentación en pdf con foto de los integrantes, y el proyecto JAVA con el código fuente.</p> <p>- Etiquetar documentación y cd con nombre, nº estudiante, carrera, grupo, materia y docente</p>				

Obligatorio: Reserva de Hoteles

Contenido

1.	Introducción	3
2.	Funcionalidades.....	4
2.1.	Crear Sistema de Reservas	4
2.2.	Destruir Sistema de Reservas.....	4
2.3.	Registrar Ciudad	4
2.4.	Registrar Hotel	5
2.5.	Ingresar Servicio	5
2.6.	Borrar Servicio	5
2.7.	Ingresar Comentario	5
2.8.	Realizar Reserva	6
2.9.	Cancelar Reserva	6
2.10.	Listado de Servicios	6
2.11.	Listado de Hoteles por Ciudad	7
2.12.	Listado de Hoteles por Ranking.....	7
2.13.	Listado de Comentarios.....	7
2.14.	Listado de Espera	8
3.	Ejercicio complementario	8
3.1.	Cargar Matriz de Distancias	8
3.2.	Camino más corto	9

1. Introducción

Se desea implementar una plataforma para la búsqueda y reserva de hoteles.

Los hoteles serán registrados en el sistema con la ciudad a la que pertenecen, su nombre (que podrá ser único dentro de la ciudad), una cierta cantidad de habitaciones (capacidad), la categoría (cantidad de estrellas), y una lista de servicios provistos.

Para simplificar la realidad consideraremos que las habitaciones de los hoteles serán todas unitarias, o sea que cada habitación podrá ser ocupada por un único cliente.

El sistema deberá proveer funcionalidades para que los clientes puedan gestionar sus reservas para los hoteles, realizar búsquedas por ciudad o por ranking, consultar los servicios de los mismos, etc.

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

Sistema	<pre>public class Sistema{ enum TipoRet{OK,ERROR_1, ERROR_2, ERROR_3, NO_IMPLEMENTADA}; /*Aquí introduzca la información que estime conveniente*/ }</pre>
----------------	---

Pueden definirse tipos de datos (clases) auxiliares.

2. Funcionalidades

2.1. Crear Sistema de Reservas

Firma: TipoRet crearSistemaReservas();

Descripción: Crea la estructura necesaria para representar el sistema de reservas.

Retornos posibles	
OK	<input type="checkbox"/> Siempre retorna OK.
ERROR	<input type="checkbox"/> No existen errores posibles.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

Nota: el sistema debe recibir como parámetro la cantidad de ciudades que va a gestionar, si el parámetro es 0 no se limita esta cantidad, de lo contrario se valida la misma.

2.2. Destruir Sistema de Reservas

Firma: TipoRet destruirSistemaReservas();

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

Retornos posibles	
OK	<input type="checkbox"/> Siempre retorna OK.
ERROR	<input type="checkbox"/> No existen errores posibles.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.3. Registrar Ciudad

Firma: TipoRet registrarCiudad(String Ciudad);

Descripción: Registra la Ciudad "Ciudad"

Nota: esto se utilizará para la parte 2 del obligatorio para definir la matriz de ciudades (mapa de ciudades)

Retornos posibles	
OK	<input type="checkbox"/> En caso que la ciudad "Ciudad" haya sido ingresado correctamente en el sistema.
ERROR	<input type="checkbox"/> 1. En caso que la ciudad "Ciudad" ya existe en el sistema.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.4. Registrar Hotel

Firma: TipoRet registrarHotel(String Ciudad, String Nombre, int Estrellas,int Capacidad);

Descripción: Registra el hotel de nombre “Nombre”, en la ciudad “Ciudad”, con cantidad de estrellas “Estrellas”, capacidad “Capacidad” y ranking 0 en el sistema de reservas.

Nota: Al inicio los hoteles registrados no contarán con ningún servicio.

Retornos posibles	
OK	<input type="checkbox"/> En caso que el hotel haya sido ingresado correctamente en el sistema.
ERROR	<input type="checkbox"/> 1. En caso que “Estrellas” sea menor a 1 o mayor a 5.
	<input type="checkbox"/> 2. En caso que “Capacidad” sea menor a 0.
	<input type="checkbox"/> 3. En caso que ya exista un hotel de nombre “Nombre” dentro de la ciudad “Ciudad”.
	<input type="checkbox"/> 4. En caso que no exista la ciudad “Ciudad”
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.5. Ingresar Servicio

Firma: TipoRet ingresarServicio(String Ciudad, String Hotel, String Servicio);

Descripción: Ingresar el servicio “Servicio” al hotel “Hotel”.

Nota 1: No se hará ningún control de unicidad sobre los servicios.

Nota 2: El ingreso de los servicios deberá realizarse en el menor tiempo posible.

Retornos posibles	
OK	<input type="checkbox"/> En caso que el servicio “Servicio” haya sido ingresado correctamente en el hotel “Hotel”.
ERROR	<input type="checkbox"/> 1. En caso que no exista un hotel “Hotel” registrado dentro de la ciudad “Ciudad”.
	<input type="checkbox"/> 2. En caso que no exista la ciudad “Ciudad”.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.6. Borrar Servicio

Firma: TipoRet borrarServicio(String Ciudad, String Hotel, String Servicio);

Descripción: Borra la primera ocurrencia del servicio “Servicio” del Hotel “Hotel”.

Retornos posibles	
OK	<input type="checkbox"/> En caso de que el servicio “Servicio” haya sido borrado correctamente del hotel “Hotel”.
ERROR	<input type="checkbox"/> 1. En caso de que no exista un hotel “Hotel” registrado dentro de la ciudad “Ciudad”.
	<input type="checkbox"/> 2. En caso de que no exista el servicio “Servicio” registrado en el hotel “Hotel”.
	<input type="checkbox"/> 3. En caso que no exista la ciudad “Ciudad”.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.7. Ingresar Comentario

Firma: TipoRet ingresarComentario(String Ciudad, String Hotel, String Comentario, int Ranking);

Descripción: Ingresar el comentario “Comentario” al hotel “Hotel”. Cada comentario llevará una calificación para el hotel. El ranking general del hotel quedará definido por el promedio de todas sus calificaciones.

Nota: El ranking será representado con un número entre 0 y 5 inclusive.

Retornos posibles	
OK	<input type="checkbox"/> En caso de que el comentario “Comentario” haya sido ingresado correctamente en el hotel “Hotel”.
ERROR	<input type="checkbox"/> 1. En caso que el ranking sea menor a 0 o mayor a 5.
	<input type="checkbox"/> 2. En caso que no exista un hotel de nombre “Hotel” registrado dentro de la ciudad “Ciudad”.
	<input type="checkbox"/> 3. En caso que no exista la ciudad “Ciudad”.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.8. Realizar Reserva

Firma: TipoRet realizarReserva(int cliente, String Ciudad, String Hotel);

Descripción: Realiza la reserva de una habitación en el hotel "Hotel" dentro de la ciudad "Ciudad". En caso de que no haya cupos de habitaciones disponibles, el cliente "cliente" quedará en lista de espera.

Retornos posibles	
OK	<input type="checkbox"/> En caso de que la reserva haya sido efectuada correctamente en el hotel "Hotel". <input type="checkbox"/> En caso de que la reserva haya quedado en lista de espera.
ERROR	<input type="checkbox"/> 1. En caso de que no exista un hotel de nombre "Hotel" registrado dentro de la ciudad Ciudad". <input type="checkbox"/> 2. En caso que no exista la ciudad "Ciudad".
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.9. Cancelar Reserva

Firma: TipoRet cancelarReserva(int cliente, String Ciudad, String Hotel);

Descripción: Cancela la reserva en el hotel "Hotel" para el cliente "cliente". En caso de que la cancelación se lleve a cabo correctamente y haya clientes en lista de espera, el cupo liberado será ocupado por el primer cliente de la lista. La búsqueda de las reservas se efectuará primero dentro de las reservas de habitaciones y luego en la lista de espera. En caso de que el cliente "cliente" tenga más de una reserva para el hotel "Hotel" se cancelará solo la primera reserva encontrada en el orden establecido.

Retornos posibles	
OK	<input type="checkbox"/> En caso que la cancelación de la habitación se lleve a cabo correctamente en el hotel "Hotel".
ERROR	<input type="checkbox"/> 1. En caso de que no exista un hotel de nombre "Hotel" registrado en la ciudad "Ciudad". <input type="checkbox"/> 2. En caso de que el cliente "cliente" no tenga ninguna reserva en el hotel "Hotel" registrado en la ciudad "Ciudad". <input type="checkbox"/> 3. En caso que no exista la ciudad "Ciudad".
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.10. Listado de Servicios

Firma: TipoRet listarServicios(String Ciudad, String Hotel);

Descripción: Lista todos los servicios prestados por el hotel "Hotel" de la ciudad "Ciudad" con el siguiente

Formato.

Servicios del Hotel <Hotel> <Ciudad>

1 - <Servicio1>

...

N - <ServicioN>

En caso que no haya servicios registrados en el hotel "Hotel", el sistema deberá imprimir:

No existen servicios registrados en el hotel <Hotel> <Ciudad>

Retornos posibles	
OK	<input type="checkbox"/> En caso que se imprima el listado correctamente.
ERROR	<input type="checkbox"/> 1. En caso que no exista el hotel "Hotel" registrado en la ciudad "Ciudad". <input type="checkbox"/> 2. En caso que no exista la ciudad "Ciudad".
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.11. Listado de Hoteles por Ciudad

Firma: TipoRet listarHotelesCiudad(String Ciudad);

Descripción: Lista todos los hoteles registrados en la ciudad "Ciudad" ordenados por Nombre con el siguiente formato.

Hoteles en <Ciudad>

<NombreHotel1> <Estrellas1> <Ranking1>

...

<NombreHotelN> <EstrellasN> <RankingN>

En caso que no haya ningún hotel registrado en la ciudad "Ciudad" el sistema deberá imprimir:
No existen hoteles registrados en <Ciudad>

Nota: La impresión de este listado, por ser muy utilizado, deberá realizarse en el menor tiempo posible.

Retornos posibles	
OK	<input type="checkbox"/> En caso que se imprima el listado correctamente.
ERROR	<input type="checkbox"/> 1. En caso que no exista la ciudad "Ciudad".
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.12. Listado de Hoteles por Ranking

Firma: TipoRet listarHotelesRanking();

Descripción: Lista todos los hoteles registrados en el sistema ordenados por ranking descendente.

El formato de impresión deberá ser el siguiente:

Hoteles ordenados por ranking

<Ciudad1> - <Hotel1> - <Ranking1>

...

<CiudadN> - <HotelN> - <RankingN>

En caso que no exista ningún hotel registrado en el sistema, se deberá imprimir:
No hay registros de hoteles en el sistema.

Retornos posibles	
OK	<input type="checkbox"/> Siempre retorna OK.
ERROR	<input type="checkbox"/> No existen errores posibles.
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.13. Listado de Comentarios

Firma: TipoRet listarComentarios(String Ciudad, String Hotel);

Descripción: Lista todos comentarios ingresados para el hotel "Hotel" dentro de la ciudad "Ciudad". Los comentarios más recientes deberán aparecer primeros en el listado.

El formato deberá ser el siguiente:

N - <Comentario N> - <RankingN>

...

1 - <Comentario 1> - <Ranking1>

En caso que no existan comentarios para el hotel "Hotel" el sistema deberá imprimir:
No se han agregado comentarios al hotel <Hotel> <Ciudad>

Retornos posibles	
OK	<input type="checkbox"/> En caso que se imprima el listado correctamente.
ERROR	<input type="checkbox"/> 1. En caso que no exista el hotel "Hotel" registrado en la ciudad "Ciudad". <input type="checkbox"/> 2. En caso que no exista la ciudad "Ciudad".
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.14. Listado de Espera

Firma: TipoRet listarEspera(String Ciudad, String Hotel);

Descripción: Lista los clientes en lista de espera para el hotel "hotel" dentro de la ciudad "Ciudad". Los clientes deberán imprimirse en el orden que serán considerados para tomar una posible habitación libre. Se considerará primero al cliente que esté hace más tiempo en la lista de espera.

El formato del listado deberá ser el siguiente:

Lista de espera para el hotel <Hotel> <Ciudad>

1 - <CedulaCliente1>

...

N - <CedulaClienteN>

En caso que no existan clientes en la lista de espera se deberá imprimir en pantalla:

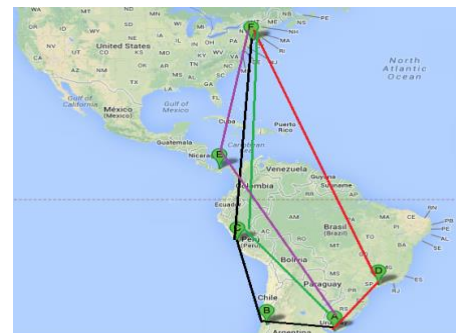
No existen reservas pendientes para el hotel <Hotel> <Ciudad>

Retornos posibles	
OK	<input type="checkbox"/> En caso que se imprima el listado correctamente.
ERROR	<input type="checkbox"/> 1. En caso que no exista el hotel "Hotel" registrado en la ciudad "Ciudad". <input type="checkbox"/> 2. En caso que no exista la ciudad "Ciudad".
NO_IMPLEMENTADA	<input type="checkbox"/> Cuando aún no se implementó. Es el tipo de retorno por defecto.

3. Ejercicio complementario

Dada la siguiente matriz en la que se almacena las distancias que existen entre las ciudades.

	A Montevideo	B Santiago	C Lima	D San Pablo	E Panamá	F New York
A	0	10		15	30	0
B	10	0	20	0	0	0
C	25	20	0	0	0	40
D	15	0	0	0	0	45
E	30	0	0	0	0	25
F	0	0	40	45	25	0



Montevideo, Lima, New York

Montevideo, Santiago, New York

Montevideo, San Pablo, New York

Montevideo, panamá, New York

Se desea implementar los siguientes métodos:

3.1. Cargar Matriz de Distancias

Firma: TipoRet CargarDistancias(int[][] Ciudades)

Nota: La matriz debe ser creada en función de la cantidad de ciudades definidas en el sistema. Debemos definir en la creación del sistema un parámetro cantidad de ciudades para luego validar y crear la matriz en función de esta cantidad.

3.2. Camino más corto

Firma: TipoRet BuscarCamino (int [][] M, string origen, string destino)

Retorna una lista



Se debe retornar el camino más corto para llegar de un origen a un destino restringiendo la búsqueda a caminos que solo tengan una ciudad intermedia.

Nota: Las casillas que tienen valor distinto de 0 son ciudades que tienen conexión y las casillas que tienen valor 0 no tienen conexión.

Defina las estructuras que considere necesarias para resolver el problema.

Información importante

- ☐ Lectura de Obligatorio: _____.
- ☐ Puntaje mínimo/máximo: 1/50
- ☐ Los obligatorios se realizan por equipos de **2 estudiantes**.
- ☐ Plazo máximo de segunda entrega (con boleta): _____.
- ☐ 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 reservas 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).