

EVALUACION	Obligatorio	GRUPO	TODOS	FECHA	Agosto 2021
MATERIA	<b>Algoritmos y Estructuras de Datos 1</b>				
CARRERA	Analista Programador / Analista en Tecnologías de la información				
CONDICIONES	<p>- <b>Puntos:</b> Máximo: 50 Mínimo: 0</p> <p>- <b>Fecha máxima de entrega:</b></p> <p><b>IMPORTANTE</b></p> <p>- Los grupos deben estar conformados por hasta 3 personas.</p> <p>- Deben Inscribirse (sacar la “boleta de entrega”).</p> <p>- La entrega será únicamente por gestión.</p>				

## Obligatorio: Sistema de libros

### Contenido

1.	Introducción.....	2
2.	Funcionalidades.....	3
2.1.	Crear Sistema de Reservas.....	3
2.2.	Destruir Sistema de Reservas.....	3
2.3.	Registrar Biblioteca.....	3
2.4.	Eliminar Biblioteca.....	3
2.5.	Registrar Libro.....	4
2.6.	Eliminar Libro.....	4
2.7.	Registrar calificación.....	4
2.8.	Realizar Reserva.....	5
2.9.	Cancelar Reserva.....	5
2.10.	Listado de libros.....	6
2.11.	Listado de libros de una biblioteca ordenado por calificación(promedio).....	6
2.12.	Listar el Ranking de todos los libros.....	6
2.13.	Listado de Comentarios.....	7
2.14.	Lista de Espera.....	7
2.15.	Cuadro de reservas hechas por biblioteca de los 5 libros más solicitados.....	8
3.	Documentation.....	9

## 1. Introducción

Se desea implementar una plataforma para la búsqueda y reserva de libros de una biblioteca

Los libros serán registrados en el sistema con un número (número único dentro de una biblioteca).

Cada libro tiene asociado un título, una editorial, una cantidad de ejemplares (cantidad de unidades reservable)

El sistema deberá proveer funcionalidades para que los clientes puedan gestionar sus reservas de libros, realizar búsquedas o consultas en general.

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

Sistema	<pre>public class Sistema{      /*Aquí introduzca la información que estime conveniente*/ } </pre>
Retorno	<pre>public class Retorno{     public enum Retorno{OK,ERROR, NO_IMPLEMENTADA};     /*Aquí introduzca la información que estime conveniente*/     public boolean valorBooleano     public int valorEntero;     public String valorString;     public Resultado resultado; }</pre>

Pueden definirse tipos de datos (clases) auxiliares.

## 2. Funcionalidades

### 2.1. Crear Sistema de Reservas

**Firma:** Retorno crearSistemaReservas();

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

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• Siempre retorna ok</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>no existen errores posibles</li></ul>
<b>NO-IMPLEMENTADA</b> – Cuando aún no se implementó Es el tipo de retorno por defecto

### 2.2. Destruir Sistema de Reservas

**Firma:** Retorno destruirSistemaReservas();

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

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• Siempre retorna ok</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>no existen errores posibles</li></ul>
<b>NO-IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

### 2.3. Registrar Biblioteca

**Firma:** Retorno registrarBiblioteca(String Biblioteca);

**Descripción:** Registra la Biblioteca “Biblioteca” en el sistema.

**Nota:** el sistema en un inicio no contempla Bibliotecas de igual nombre en instituciones diferentes.

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• En caso de agregar la Biblioteca con éxito</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>• 1. En caso de que la Biblioteca ya exista</li></ul>
<b>NO-IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

### 2.4. Eliminar Biblioteca

**Firma:** Retorno eliminarBiblioteca(String Biblioteca);

**Descripción:** Elimina la Biblioteca “Biblioteca” del sistema

**Nota:** Elimina la Biblioteca siempre y cuando no tenga reservas asociados, en cuyo caso debe emitir un mensaje.

Retornos posibles	
<b>OK</b>	<ul style="list-style-type: none"> <li>• En caso de eliminar la Biblioteca con éxito</li> </ul>
<b>ERROR</b>	<ul style="list-style-type: none"> <li>• 1. En caso de que la biblioteca no exista</li> </ul>
<b>NO-IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto	

## 2.5.Registrar Libro

**Firma:** Retorno registrarLibro(String titulo, String editorial, String biblioteca, int ejemplares);

**Descripción:** Registra el título “titulo” de la editorial “editorial” en la biblioteca “biblioteca”, con una cantidad de ejemplares “ejemplares”.

**Nota:** puede existir el mismo libro en distintas bibliotecas

Retornos posibles	
<b>OK</b>	<ul style="list-style-type: none"> <li>• En caso de que el libro haya sido ingresado correctamente en el sistema</li> </ul>
<b>ERROR</b>	<ul style="list-style-type: none"> <li>• 1. En caso de que el libro ya exista en esa biblioteca</li> <li>• 2. En caso de que la biblioteca no exista.</li> </ul>
<b>NO-IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto	

## 2.6.Eliminar Libro.

**Firma:** Retorno eliminarLibro(String titulo, String editorial, String biblioteca);

**Descripción:** elimina el libro de la biblioteca “biblioteca”

**Nota 1:** Se elimina el libro y todos sus ejemplares

Retornos posibles	
<b>OK</b>	<ul style="list-style-type: none"> <li>• En caso de que el título “titulo” de la editorial “editorial” haya sido eliminado correctamente en la biblioteca “biblioteca”</li> </ul>
<b>ERROR</b>	<ul style="list-style-type: none"> <li>• 1. En caso de que no exista el título “titulo” de la editorial “editorial” en la biblioteca “biblioteca”</li> </ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto	

## 2.7.Registrar calificación.

**Firma:** Retorno RegistrarCalificacion(String titulo, String editorial, int calificacion, String biblioteca, String comentario);

**Descripción:** Registra una calificación sobre el libro. La misma, es un valor comprendido entre 0 y 5 que indica el grado de aceptación del libro y si lo recomienda (0 no lo recomienda - 5 es muy recomendable).

**Nota:** El ranking será representado con un número entre 0 y 5 inclusive.  
El ranking general de la biblioteca quedará definido por el promedio de todas

sus calificaciones.

Todo método de alta ordenada deberá ser realizado **recursivamente**.

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• En caso de que la calificación haya sido registrada correctamente para el título "título" de la editorial "editorial" de la biblioteca "biblioteca".</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>• 1. En caso de que no exista el título "título" de esa "editorial" en esa "biblioteca"</li><li>• 2. En caso de que la calificación no este comprendida entre 0 y 5</li></ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

## 2.8. Realizar Reserva.

**Firma:** Retorno realizarReserva(int cliente, int numero, String biblioteca, String título, String editorial, String fecha);

**Descripción:** Realiza la reserva para el cliente "cliente" con reserva número "numero" en la biblioteca "biblioteca", título "título" de la editorial "editorial". En caso de que no haya ejemplares disponibles, el cliente "cliente" quedará en lista de espera. El cliente podrá realizar más de una reserva.

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• En caso de que la reserva haya sido efectuada correctamente</li><li>• En caso de que la reserva haya quedado en lista de espera</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>• 1. En caso de que no exista el libro buscado en la biblioteca</li></ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

## 2.9. Cancelar Reserva

**Firma:** Retorno cancelarReserva(int cliente, int numero, String biblioteca);

**Descripción:** Cancela la reserva de un cliente "cliente" para la reserva número "numero" de la biblioteca "biblioteca". 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 de espera. La búsqueda de las reservas se efectuará primero dentro de las reservas de libros y luego en la lista de espera.

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• En caso de que la cancelación del número "numero" de la biblioteca "biblioteca" se lleve a cabo correctamente para el cliente "cliente".</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>• 1. En caso de que no exista un numero de reserva "numero" registrado en la biblioteca "biblioteca".</li></ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

## 2.10. Listado de libros.

**Firma:** Retorno listarLibros(String biblioteca);

**Descripción:** Lista todos los libros ofrecidos por la biblioteca “biblioteca” con el siguiente formato:

Libros de la biblioteca < biblioteca >

1 - <libro 1>

-----

N - <libro N>

En caso de que no haya libros registrados en la biblioteca “biblioteca” el sistema deberá imprimir: No existen libros registrados en la biblioteca <biblioteca>

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• En caso de que se imprima el listado correctamente.</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>• 1. En caso de que no exista la biblioteca “biblioteca” registrada</li></ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

## 2.11. Listado de libros de una biblioteca ordenado por calificación (promedio).

**Firma:** Retorno listarLibrosBiblioteca (String biblioteca);

**Descripción:** Lista todos los libros registrados en la biblioteca “biblioteca” ordenados por Ranking con el siguiente formato.

libros de la biblioteca <biblioteca>

<Titulo> <editorial> <calificación>

...

<Titulo> <editorial> <calificación>

En caso de que no haya ningún título “titulo” registrado en la biblioteca “biblioteca” el sistema deberá imprimir: No existen libros registrados.

**Nota:** La impresión de este listado (por ser muy utilizado) deberá realizarse en el menor tiempo posible. Se valorará especialmente la eficiencia en este método.

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• Siempre retorna OK.</li></ul>
<b>ERROR</b> <p>No hay errores posibles.</p>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

## 2.12. Listar el Ranking de todos los libros

**Firma:** Retorno listarBibliotecaRanking();

**Descripción:** Lista todos los libros registrados en el sistema ordenadas por ranking

descendente.

El formato de impresión deberá ser el siguiente:

Bibliotecas ordenadas por ranking

<libro> - <Ranking>

...

<libros> - <Ranking>

En caso de que no exista ninguna biblioteca con libros registrados en el sistema, se deberá imprimir: No hay registros de libros en el sistema.

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• Siempre retorna OK.</li></ul>
<b>ERROR</b> No hay errores posibles.
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

### 2.13. Listado de Comentarios

**Firma:** Retorno listarComentarios (String biblioteca)

**Descripción:** Lista todos comentarios ingresados de la biblioteca “biblioteca”. Los comentarios más recientes deberán aparecer primeros en el listado. El formato deberá ser el siguiente:

N - <Comentario N>

...

1 - <Comentario 1>

En caso de que no existan comentarios para el sistema deberá imprimir:

No se han agregado comentarios de la biblioteca <biblioteca>

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"><li>• En caso de que el listado se haya imprimido correctamente.</li></ul>
<b>ERROR</b> <ul style="list-style-type: none"><li>• 1. En caso de que no exista la biblioteca “biblioteca” registrada.</li></ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

### 2.14. Lista de Espera

**Firma:** Retorno listarEspera (String titulo, String editorial, String biblioteca);

**Descripción:** Lista los clientes en lista de espera para un titulo “titulo” de la editorial “editorial” dentro de la biblioteca “biblioteca”.

Los clientes deberán imprimirse en el orden que serán considerados para tomar el libro. 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 del titulo “titulo” de la editorial “editorial” en la biblioteca “biblioteca”

1 - <CedulaCliente1>

...

N - <CedulaClienteN>

En caso de que no existan clientes en la lista de espera se deberá imprimir en pantalla:  
No existen reservas pendientes para el título “titulo” de la editorial “editorial” en la biblioteca “biblioteca”

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"> <li>• En caso de que el listado se haya imprimido correctamente.</li> </ul>
<b>ERROR</b> <ul style="list-style-type: none"> <li>• 1. En caso de que no exista el título “titulo” de la editorial “editorial” en la biblioteca “biblioteca”</li> </ul>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

## 2.15. Cuadro de reservas hechas por biblioteca de los 5 libros más solicitados.

**Firma:** Retorno mostrarReservasBiblioteca ();

**Descripción:** Muestra por pantalla una tabla conteniendo las reservas de los 5 libros más solicitados en el sistema. Solamente se mostrarán las cantidades por biblioteca y el total del libro en todas las bibliotecas

Retornos posibles
<b>OK</b> <ul style="list-style-type: none"> <li>• En caso de que el listado se haya imprimido correctamente.</li> </ul>
<b>ERROR</b> <p>No hay errores posibles</p>
<b>NO IMPLEMENTADA</b> – • Cuando aún no se implementó Es el tipo de retorno por defecto

El siguiente es un ejemplo de impresión.

**Nota:** se imprime el top 5 de libros más solicitados.

Reservas/biblioteca	Biblioteca 1	Biblioteca2	Biblioteca3	Biblioteca4	Biblioteca5	total
Libro1						
Libro2						
Libro3						
Libro4						
Libro5						



### 3. Documentation

Se pide que la documentación incluya:

1. Las pre y post condiciones de los métodos solicitados.
2. 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.
3. El conjunto de pruebas diseñadas y ejecutadas y el resultado obtenido de esta ejecución con un screenshot de la pantalla mostrando el resultado de cada uno de los tests.
4. Los nombres de cada @Test (caso de prueba) debe ser descriptivos, dejando claro que es lo que se pretende testear.
5. No se aceptará Tests con nombres Test1, Test2, etc.

#### Información importante

Puntaje mínimo/máximo: 0/50

Los obligatorios se realizan por equipos de hasta 3 **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 la selección adecuada de las estructuras para modelar el problema y la eficiencia en cada una de las operaciones es muy importante.**

Deberá aplicar la metodología vista en el curso.

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 debe respetar las firmas que se describen en el documento. (El uso de esta interfaz es obligatorio y no puede ser modificada).