**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**Escuela de Tecnología**

**OBLIGATORIO ALGORITMOS Y ESTRUCTURAS DE DATOS 1**

**DOCUMENTO**

Hombre con barba y bigote

Descripción generada automáticamente

**Nicolas Gimenez – 291950**

Un hombre con una playera de color azul

Descripción generada automáticamente

**Cristian García – 317010**

**Grupo N3D**

**Docente: Rafael Cohen**

**Analista en tecnologías de la información**

**14/10/2024**

​​**Índice**

[1. Representación gráfica de estructuras. 3](#_Toc179754133)

[2. Juego de prueba y evidencia 7](#_Toc179754134)

[2.1 Agregar estudiante y mostrar el listado. 7](#_Toc179754135)

[2.2 Obtener un estudiante por su número. 7](#_Toc179754136)

[2.3 Eliminar estudiante y mostrar la lista. 8](#_Toc179754137)

[2.4 Agregar 3 estudiantes de forma ordenada y mostrar la lista. 8](#_Toc179754138)

[2.5 Vaciar la lista de estudiantes y mostrar. 9](#_Toc179754139)

[2.6 Agregar 4 estudiantes de forma desordenada y mostrar. 9](#_Toc179754140)

[2.7 Mostrar lista de libros (vacía). 9](#_Toc179754141)

[2.8 Agregar un libro y mostrar la lista. 10](#_Toc179754142)

[2.9 Agregar 4 libros ordenados y mostrar la lista. 10](#_Toc179754143)

[2.10 Vaciar la lista de libros. 11](#_Toc179754144)

[2.11 Agregar 4 libros desordenados y mostrar la lista. 11](#_Toc179754145)

[2.12 Mostrar la lista de libros según la categoría indicada. 12](#_Toc179754146)

[2.13 Intentar eliminar un estudiante con préstamo activo. 12](#_Toc179754147)

# Representación gráfica de estructuras.

Lista doble encadenada de **estudiantes** ordenado creciente por número de estudiante.

Se define este tipo de dato para cumplir por ejemplo con el requerimiento 3.1: Se listan los datos de los estudiantes ordenados creciente por número, cargando el resultado en el valor String del retorno.

Cada estudiante contiene una lista de sus préstamos.

Diagrama

Descripción generada automáticamente

PRESTAMO

Libro: Libro

Estudiante: Estudiante

Fecha: LocalDateTime

Activo: boolean

ESTUDIANTE

Nombre: string

Apellido: string

Número: int

PrestamosActivos: ListaDoble<Prestamo>

Lista doble encadenada de **libros** ordenada creciente por ISBN.

Se define este tipo de dato para cumplir por ejemplo con el requerimiento 3.2: Se listan los datos de todos los libros de la biblioteca ordenados creciente por ISBN cargando el resultado en el valor String del retorno.

Diagrama

Descripción generada automáticamente

Lista doble encadenada de **préstamos** ordenada creciente por IBSN.

PRESTAMO

Libro: Libro

Estudiante: Estudiante

Fecha: LocalDateTime

Activo: boolean

LIBRO

Nombre: string

ISBN: string

Categoria: string

Total: int

Disponible: int

Se define este tipo de dato para cumplir por ejemplo con el requerimiento 3.2: Se deben listar el/los libros más prestados ordenado por ISBN, cargando el resultado en el valor String del retorno. En caso de existir libros con la misma cantidad “máxima de préstamos”, se deberán mostrar todos.

Diagrama

Descripción generada automáticamente

Cola de **reservas**.

PRESTAMO

Libro: Libro

Estudiante: Estudiante

Fecha: LocalDateTime

Activo: boolean

Se define este tipo de dato para cumplir por ejemplo con el requerimiento 2.7: Registrar una reserva de un ejemplar de un libro para un estudiante. Solo se puede realizar una reserva si no existe stock disponible en el momento para realizar el préstamo.

Diagrama

Descripción generada automáticamente

RESERVA

Libro: Libro

Estudiante: Estudiante

Fecha: LocalDateTime

# Juego de prueba y evidencia

Se realizó u juego de prueba dentro del proyecto donde se ejecutan los métodos solicitados y se muestran los resultados obtenidos.

A continuación se detalla el código con la prueba y print del resultado obtenido.

## 2.1 Agregar estudiante y mostrar el listado.

Código:

sis.agregarEstudiante("Nombre", "Apellido", 1111);

sis.agregarEstudiante("Nombre", "Apellido", 1112);

System.out.println("Lista de estudiantes: " + sis.listarEstudiantes().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.2 Obtener un estudiante por su número.

Código:

System.out.println("Obtengo estudiante 1111: " + sis.obtenerEstudiante(1111).valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.3 Eliminar estudiante y mostrar la lista.

Código:

sis.eliminarEstudiante(1111);

System.out.println("Elimino el estudiante 1111 muestro la lista: " + sis.listarEstudiantes().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.4 Agregar 3 estudiantes de forma ordenada y mostrar la lista.

Código:

sis.agregarEstudiante("Nombre2", "Apellido2", 2222);

sis.agregarEstudiante("Nombre3", "Apellido3", 3333);

sis.agregarEstudiante("Nombre4", "Apellido4", 4444);

System.out.println("Agrego 3 estudiantes ordenados y muestro lista: " + sis.listarEstudiantes().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.5 Vaciar la lista de estudiantes y mostrar.

Código:

sis.Estudiantes.vaciar();

System.out.println("Vaciamos la lista: " + sis.listarEstudiantes().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.6 Agregar 4 estudiantes de forma desordenada y mostrar.

Código:

sis.agregarEstudiante("Nombre7", "Apellido7", 7777);

sis.agregarEstudiante("Nombre1", "Apellido1", 1111);

sis.agregarEstudiante("Nombre4", "Apellido4", 4444);

sis.agregarEstudiante("Nombre3", "Apellido3", 3333);

System.out.println("Agrego 4 estudiantes desordenados y muestro lista: \n" + sis.listarEstudiantes().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:

Texto

Descripción generada automáticamente con confianza media

## 2.7 Mostrar lista de libros (vacía).

Código:

System.out.println("Lista de libros vacia: " + sis.listarLibros().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.8 Agregar un libro y mostrar la lista.

Código:

sis.agregarLibro("NombreLibro", "ISBN", "Categoria", 150);

System.out.println("Agrego un libro y muestro la lista: " + sis.listarLibros().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.9 Agregar 4 libros ordenados y mostrar la lista.

Código:

sis.agregarLibro("NombreLibro1", "ISBN1", "Filosofia", 150);

sis.agregarLibro("NombreLibro2", "ISBN2", "Matematica", 250);

sis.agregarLibro("NombreLibro3", "ISBN3", "Filosofia", 350);

sis.agregarLibro("NombreLibro4", "ISBN4", "Matematica", 450);

System.out.println("Agrego 4 libros ordenados y muestro la lista: " + sis.listarLibros().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:

Texto

Descripción generada automáticamente con confianza media

## 2.10 Vaciar la lista de libros.

Código:

sis.Libros.vaciar();

System.out.println("Vaciamos la lista: " + sis.listarLibros().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:



## 2.11 Agregar 4 libros desordenados y mostrar la lista.

Código:

sis.agregarLibro("NombreLibro8", "ISBN8", "Filosofia", 250);

sis.agregarLibro("NombreLibro7", "ISBN7", "Matematica", 150);

sis.agregarLibro("NombreLibro2", "ISBN2", "Filosofia", 450);

sis.agregarLibro("NombreLibro5", "ISBN5", "Arquitectura", 350);

System.out.println("Agrego 4 libros desordenados y muestro la lista: \n" + sis.listarLibros().valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:

Forma

Descripción generada automáticamente con confianza media

## 2.12 Mostrar la lista de libros según la categoría indicada.

Código:

System.out.println("Libros de Filosofia: " + sis.listarLibros("Filosofia").valorString);

System.out.println("-----------------------------------------------------------\n");

Resultado:

Imagen que contiene Forma

Descripción generada automáticamente

## 2.13 Intentar eliminar un estudiante con préstamo activo.

Código:

Libro libro = sis.Libros.obtenerElemento(new Libro(null, "ISBN10", null, 0));

Estudiante estudiante = sis.Estudiantes.obtenerElemento(e1);

estudiante.agregarPrestamo(libro);

System.out.println("Intentar eliminar estudiante con préstamo activo: " + sis.eliminarEstudiante(1111).resultado);

Resultado:



## 2.14 Evidencia de test.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente