**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**

**21/11/2024**

​​**Índice**

[1. Representación gráfica de estructuras. 3](#_Toc183068216)

[2. Juego de prueba y evidencia 8](#_Toc183068217)

[2.1 Crear sistema de gestión. 8](#_Toc183068218)

[2.2 Agregar un estudiante. 8](#_Toc183068219)

[2.3 Obtener estudiante. 10](#_Toc183068220)

[2.4 Eliminar estudiante. 11](#_Toc183068221)

[2.5 Agregar Libro. 12](#_Toc183068222)

[2.6 Prestar libro. 13](#_Toc183068223)

[2.7 Reservar libro. 15](#_Toc183068224)

[2.8 Devolver libro. 17](#_Toc183068225)

[2.9 Eliminar libro. 19](#_Toc183068226)

[2.10 Listar estudiantes. 20](#_Toc183068227)

[2.11 Listar libros. 21](#_Toc183068228)

[2.12 Listar libros por categoría. 21](#_Toc183068229)

[2.13 Listar préstamos de un estudiante. 22](#_Toc183068230)

[2.14 Libros más prestados. 23](#_Toc183068231)

[2.15 Deshacer n eliminaciones. 23](#_Toc183068232)

[2.16 Cantidad de préstamos activos. 24](#_Toc183068233)

[2.17 Ranking de categorías. 25](#_Toc183068234)

[2.18 Resultados finales. 25](#_Toc183068235)

[2.19 Evidencia de test. 26](#_Toc183068236)

# 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.

El libro cuenta con una lista de préstamos y una cola de estudiantes que han hecho reservas de ese libro por no haber stock disponible.

Diagrama

Descripción generada automáticamente

ESTUDIANTE

Nombre: string

Apellido: string

Número: int

PrestamosActivos: ListaDoble<Prestamo>

PRESTAMO

Libro: Libro

Estudiante: Estudiante

Fecha: LocalDateTime

Activo: boolean

LIBRO

Nombre: string

ISBN: string

Categoria: string

Total: int

Disponible: int

Lista doble encadenada de **libros** ordenada decreciente por cantidad de préstamos.

Lista de libros indexada por cantidad de préstamos del libro.

Diagrama

Descripción generada automáticamente

ESTUDIANTE

Nombre: string

Apellido: string

Número: int

PrestamosActivos: ListaDoble<Prestamo>

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

PRESTAMO

Libro: Libro

Estudiante: Estudiante

Fecha: LocalDateTime

Activo: boolean

Pila de **libros**.

Se define este tipo de dato para cumplir por ejemplo con el requerimiento 3.6: Se deben deshacer las ultimas n eliminaciones de libros realizadas en el sistema. Si existen menos de n eliminaciones, se deshacen todas las realizadas. Se deberá cargar el resultado de los libros recuperados en el valor String del retorno, según el orden de recuperación (primer mostrar el último cancelado y recuperado)

Diagrama

Descripción generada automáticamente

LIBRO

Nombre: string

ISBN: string

Categoria: string

Total: int

Disponible: int

# Juego de prueba y evidencia

Se realizó un 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 Crear sistema de gestión.

p.ver(s.crearSistemaDeGestion().resultado, Retorno.Resultado.OK, "se crea sistema");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

## 2.2 Agregar un estudiante.

**Agregar estudiante caso de éxito.**

p.ver(s.agregarEstudiante("Luis", "Suarez", 1).resultado, Retorno.Resultado.OK, "OK: Se agrego a Luis Suarez");

Resultado:

Escala de tiempo

Descripción generada automáticamente

**Agregar estudiante caso error 1.**

p.ver(s.agregarEstudiante("", "Polenta", 11).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1 : Faltan parametros");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Agregar estudiante caso error 2.**

p.ver(s.agregarEstudiante("Diego", "Aguirre", -1).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2 : Numero fuera de rango");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Agregar estudiante caso error 3.**

p.ver(s.agregarEstudiante("Luis", "Suarez", 1).resultado, Retorno.Resultado.ERROR\_3, "Error 3: Se intento agregar un estudiante existente");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

## 2.3 Obtener estudiante.

**Obtener estudiante caso de éxito.**

p.ver(s.obtenerEstudiante(1).resultado, Retorno.Resultado.OK, "OK: se obtuvo estudiante");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Obtener estudiante error 1.**

p.ver(s.obtenerEstudiante(-1).resultado, Retorno.Resultado.ERROR\_1, "Error 1: numero fuera de rango");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Obtener estudiante error 2.**

p.ver(s.obtenerEstudiante(20).resultado, Retorno.Resultado.ERROR\_2, "Error 2 - el estudiante 20 no existe");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

## 2.4 Eliminar estudiante.

**Eliminar estudiante caso de éxito.**

p.ver(s.eliminarEstudiante(10).resultado, Retorno.Resultado.OK, "OK: Se elimina estudiante 10");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Eliminar estudiante error 1.**

p.ver(s.eliminarEstudiante(-10).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: Se intenta eliminar estudiante con nro fuera de rango");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Eliminar estudiante error 2.**

p.ver(s.eliminarEstudiante(20).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: no existe ese estudiante");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Eliminar estudiante error 3.**

p.ver(s.eliminarEstudiante(1).resultado, Retorno.Resultado.ERROR\_3, "ERROR 3: Se intenta eliminar estudiante con prestamos");

Resultado:

Escala de tiempo

Descripción generada automáticamente

## 2.5 Agregar Libro.

**Agregar libro caso de éxito.**

p.ver(s.agregarLibro("Libro 15", "15", "categoria 1", 1).resultado, Retorno.Resultado.OK, "OK: se agrega libro 15 con 1 ejemplar");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Agregar libro error 1.**

p.ver(s.agregarLibro("", "11", "categoria 3", 1).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: se intenta agregar libro con falta de parametros");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Agregar libro error 2.**

p.ver(s.agregarLibro("Libro 10", "10", "categoria 3", 1).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: Se intenta agregar libro existente");

Resultado:

Imagen que contiene Texto

Descripción generada automáticamente

**Agregar libro error 3.**

p.ver(s.agregarLibro("Libro 13", "20", "ategoria 3", -1).resultado, Retorno.Resultado.ERROR\_3, "OK: se intenta agregar libro con cantidad de ejemplares incorrecta");

Resultado:

Texto

Descripción generada automáticamente con confianza media

## 2.6 Prestar libro.

**Prestar libro caso de éxito.**

p.ver(s.prestarLibro("3", 1).resultado, Retorno.Resultado.OK, "OK: se presto Libro 1 ejemplar 1 libro 3");

Resultado:

Escala de tiempo

Descripción generada automáticamente

**Prestar libro error 1.**

p.ver(s.prestarLibro("", 1).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: ISBN vacio");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Prestar libro error 2.**

p.ver(s.prestarLibro("25", 1).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: no existe libro con ese ISBN");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Prestar libro error 3.**

p.ver(s.prestarLibro("3", -1).resultado, Retorno.Resultado.ERROR\_3, "ERROR 3: numero de estudiante fuera de rango");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Prestar libro error 4.**

p.ver(s.prestarLibro("3", 31).resultado, Retorno.Resultado.ERROR\_4, "ERROR 4: no Existe ese estudiante");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Prestar libro error 5.**

p.ver(s.prestarLibro("10", 6).resultado, Retorno.Resultado.ERROR\_5, "ERROR 5: Libro sin stock");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Prestar libro error 6.**

p.ver(s.prestarLibro("3", 1).resultado, Retorno.Resultado.ERROR\_6, "ERROR 6: ya existe ese prestamo para ese estudiante libro");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

## 2.7 Reservar libro.

**Reservar libro caso de éxito.**

p.ver(s.reservarLibro("15", 2).resultado, Retorno.Resultado.OK, "OK: Se reserva libro 15");

Resultado:

Escala de tiempo

Descripción generada automáticamente con confianza baja

**Reservar libro error 1.**

p.ver(s.reservarLibro("", 1).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: ISBN null");

Resultado:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Reservar libro error 2.**

p.ver(s.reservarLibro("34", 1).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: no existe ese libro");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Reservar libro error 3.**

p.ver(s.reservarLibro("3", -1).resultado, Retorno.Resultado.ERROR\_3, "ERROR 3: numero fuera de rango");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Reservar libro error 4.**

p.ver(s.reservarLibro("3", 22).resultado, Retorno.Resultado.ERROR\_4, "ERROR 4: no existe ese estudiante");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Reservar libro error 5.**

p.ver(s.reservarLibro("2", 5).resultado, Retorno.Resultado.ERROR\_5, "ERROR 5: ese libro aún tiene stock");

Resultado:

Escala de tiempo

Descripción generada automáticamente con confianza media

## 2.8 Devolver libro.

**Devolver libro caso de éxito.**

p.ver(s.devolverLibro("1", 1).resultado, Retorno.Resultado.OK, "OK: se devuelve el libro 1, estudiante 1");

Resultado:

Escala de tiempo

Descripción generada automáticamente con confianza baja

**Devolver libro error 1.**

p.ver(s.devolverLibro("", 1).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: ISBN vacío");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

**Devolver libro error 2.**

p.ver(s.devolverLibro("100", 1).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: no existe libro con ese ISBN");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Devolver libro error 3.**

p.ver(s.devolverLibro("1", -1).resultado, Retorno.Resultado.ERROR\_3, "ERROR 3: número fuera de rango");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Devolver libro error 4.**

p.ver(s.devolverLibro("1", 150).resultado, Retorno.Resultado.ERROR\_4, "ERROR 4: no existe estudiante con ese número");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Devolver libro error 5.**

p.ver(s.devolverLibro("7", 1).resultado, Retorno.Resultado.ERROR\_5, "ERROR 5: no existe un prestamo activo de ese libro para ese estudiante");

Resultado:

Texto

Descripción generada automáticamente

## 2.9 Eliminar libro.

**Eliminar libro caso de éxito.**

p.ver(s.eliminarLibro("ISBN2").resultado, Retorno.Resultado.OK, "OK: se elimina el libro ISBN2");

Resultado:

Escala de tiempo

Descripción generada automáticamente

**Eliminar libro error 1.**

p.ver(s.eliminarLibro("").resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: el ISBN es vacío a null");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Eliminar libro error 2.**

p.ver(s.eliminarLibro("1").resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: el libro tiene préstamos");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

## 2.10 Listar estudiantes.

**Listar estudiantes caso de éxito.**

p.ver(s.listarEstudiantes().resultado, Retorno.Resultado.OK, "OK: se listan los estudiantes");

System.out.println(s.listarEstudiantes().valorString + "\n");

Resultado:

Texto

Descripción generada automáticamente

## 2.11 Listar libros.

**Listar libros caso de éxito.**

p.ver(s.listarLibros().resultado, Retorno.Resultado.OK, "OK: se listan los libros");

System.out.println(s.listarLibros().valorString + "\n");

Resultado:

Texto

Descripción generada automáticamente

## 2.12 Listar libros por categoría.

**Listar libros por categoría caso de éxito.**

p.ver(s.listarLibros("categoria 1").resultado, Retorno.Resultado.OK, "OK: se listan los libros de la categoria 1");

System.out.println(s.listarLibros("categoria 1").valorString + "\n");

Resultado:

Escala de tiempo

Descripción generada automáticamente

**Listar libros por categoría error 1.**

p.ver(s.listarLibros("").resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: la categoria es vacia o null");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

## 2.13 Listar préstamos de un estudiante.

**Listar préstamos de un estudiante caso de éxito.**

p.ver(s.listarPrestamos(1).resultado, Retorno.Resultado.OK, "OK: se listan los prestamos del estudiante 1");

System.out.println(s.listarPrestamos(1).valorString + "\n");

Resultado:

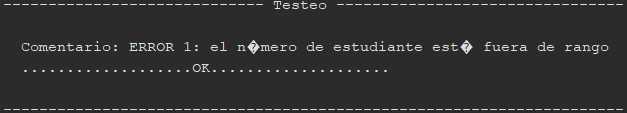
Texto

Descripción generada automáticamente

**Listar préstamos de un estudiante error 1.**

p.ver(s.listarPrestamos(0).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: el número de estudiante está fuera de rango");

Resultado:



**Listar préstamos de un estudiante error 2.**

p.ver(s.listarPrestamos(2505).resultado, Retorno.Resultado.ERROR\_2, "ERROR 2: no existe un estudiante con ese número");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

## 2.14 Libros más prestados.

**Libros más prestados caso de éxito.**

p.ver(s.librosMasPrestados().resultado, Retorno.Resultado.OK, "OK: se listan el/los libro/s más prestado/s");

System.out.println(s.librosMasPrestados().valorString + "\n");

Resultado:

Imagen que contiene Diagrama

Descripción generada automáticamente

## 2.15 Deshacer n eliminaciones.

**Deshacer n eliminaciones de libros caso de éxito.**

p.ver(s.deshacerEliminaciones(2).resultado, Retorno.Resultado.OK, "OK: se deshacen las ultimas 2 eliminaciones de libros");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

**Deshacer n eliminaciones de libros error 1.**

p.ver(s.deshacerEliminaciones(0).resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: n es menor o igual a 0");

Resultado:

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

## 2.16 Cantidad de préstamos activos.

**Cantidad de préstamos activos caso de éxito.**

p.ver(s.cantidadPrestamosActivos("1").resultado, Retorno.Resultado.OK, "OK: se muestra la cantidad de préstamos del libro 1");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Cantidad de préstamos activos error 1.**

p.ver(s.cantidadPrestamosActivos("").resultado, Retorno.Resultado.ERROR\_1, "ERROR 1: el IBSN es vacío o null");

Resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

## 2.17 Ranking de categorías.

**Ranking de categorías caso de éxito.**

p.ver(s.prestamosXCategoría().resultado, Retorno.Resultado.OK, "OK: se muestra el ranking de prestamos por categoria");

Resultado:

Imagen que contiene Gráfico

Descripción generada automáticamente

## 2.18 Resultados finales.

**Juego de prueba alumnos.**

Texto

Descripción generada automáticamente

**Juego de prueba propuesto en clase.**

Texto

Descripción generada automáticamente

## 2.19 Evidencia de test.

Captura de pantalla de computadora

Descripción generada automáticamente

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

Descripción generada automáticamente