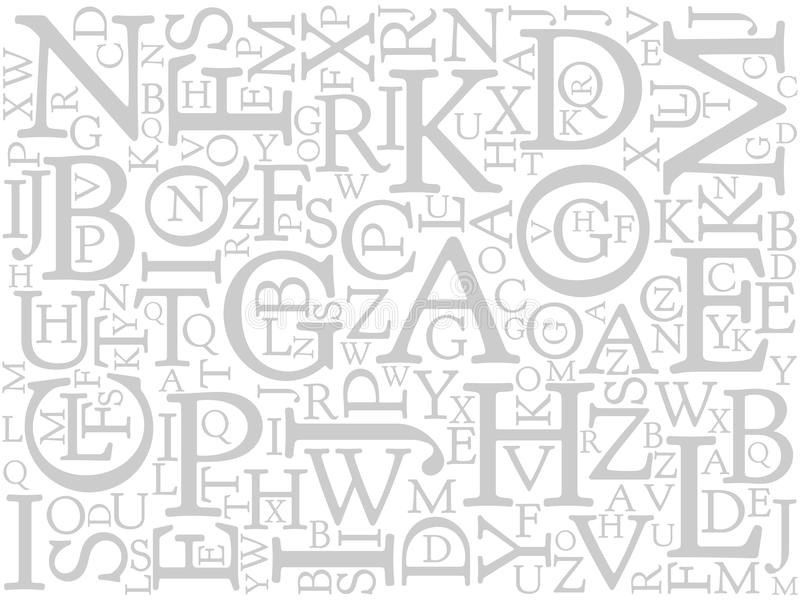
**Programación Avanzada**

**Trabajo integrador POO**





**Profesora:** Edith Schamberger

**Alumnos**: Franco Bravo, Marcos Alejandro Sena Ferreiro,

Costas Daniela Noemí, Federico Alexis Lambrech

**2023**

**1° CUATRIMESTRE**

**Biblioteca Virtual**

Contenido

[1. Análisis de la aplicación: 3](#_Toc136887914)

[2. Estrategia implementada: 3](#_Toc136887915)

[3. Decisiones de diseño establecidas: 3](#_Toc136887916)

[4. El diagrama de clases del programa sería el siguiente: 4](#_Toc136887917)

[5. Descripción de clases/objetos: 4](#_Toc136887918)

[**a.** Clase Usuario: 4](#_Toc136887919)

[**b.** Clase Administrador: 4](#_Toc136887920)

[**c.** Clase Cliente: 4](#_Toc136887921)

[**d.** Clase Libro: 5](#_Toc136887922)

[**e.** Clase Calificación: 5](#_Toc136887923)

[**f.** Clase Reserva: 5](#_Toc136887924)

[6. Anexo del código: 5](#_Toc136887925)

# Análisis de la aplicación:

El programa es una simulación de una biblioteca, donde se pueden reservar y prestar libros, devolverlos, cancelar reservas y calificar los ejemplares.

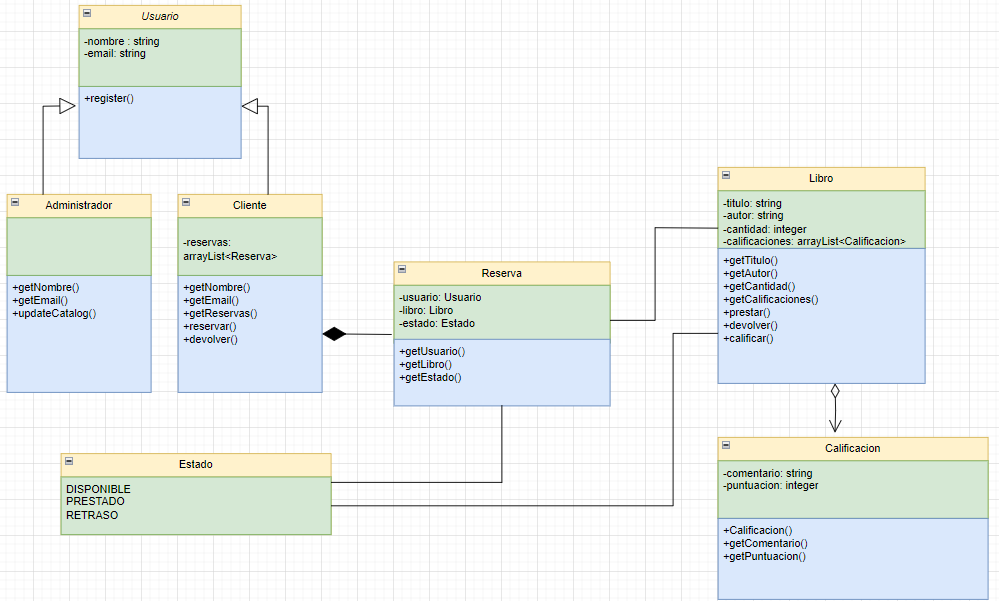
# Estrategia implementada:

Se usó el lenguaje de programación Java para desarrollar el programa y se ha hecho uso de clases y objetos para modelar la biblioteca, los usuarios y los libros.

# Decisiones de diseño establecidas:

Se utilizó enumeración Estado para representar el estado de los libros (disponible, prestado, retraso), una superclase Usuario que luego dio lugar a dos subclases (Administrador y Cliente) que heredan sus atributos. También incluimos el método reservar para la clase Cliente, de manera que desde allí pueda agregar una nueva entrada a su lista de reservas. Decidimos además, incorporar una clase Reserva para representar la reserva de un libro por parte de un cliente, y el método devolver de la clase Reserva para permitir la devolución del libro prestado.

# El diagrama de clases del programa sería el siguiente:



# Descripción de clases/objetos:

# Clase Usuario:

Representa a un usuario de la biblioteca. Tiene un nombre, un mail y el método asociado es el registro a la página de la biblioteca.

# Clase Administrador:

Representa al empleado de la biblioteca. Al ser una subclase de Usuario, hereda los atributos, también tiene un nombre y un mail, pero además tiene “updateCatalog” como método asociado, que indica que es quien puede agregar libros al catálogo, modificar la cantidad de ejemplares disponibles, eliminarlos, etc.

# Clase Cliente:

Representa a aquellas personas que quieren reservar libros a través de la página de la biblioteca. Es una subclase de usuario así que hereda sus atributos. Tiene varios métodos que indican las acciones que puede realizar: getNombre, getEmail, getReservas, reservar y devolver.

# Clase Libro:

Representa a un libro de la biblioteca. Tiene un título, un autor, una cantidad y una lista de calificaciones. Esta clase contiene los métodos getTitulo, getAutor, getCantidad, prestar, devolver y calificar.

# Clase Calificación:

Representa a una calificación que se ha dado a un libro. Tiene un comentario y una puntuación. Esta clase contiene los métodos getComentario y getPuntuacion.

# Clase Reserva:

Representa una reserva de un libro por parte de un usuario. Está asociada a un usuario, un libro y un estado (disponible, prestado, retraso). Esta clase contiene los métodos getUsuario, getLibro, getEstado, devolver, cancelar y calificar.

# Anexo del código: