DISEÑO DE SISTEMAS

Juan Carlos Molina Lozano

Docente

CONTENIDO

- Objetivos
- Propósito
- Introducción a Fichas CRC
- Características de las Fichas CRC
- Ventajas y Desventajas de las Fichas CRC
- Como se crean las Fichas CRC
- Introducción a Diagramas de Clases
- Caracteristicas de los Diagramas de Clases
- Ventajas y Desventajas de los Diagramas de Clases
- Herramientas para la creación de Diagramas de Clases
- Preguntas

OBJETIVOS DE LA CLASE

- Explicar el propósito y la importancia de las fichas CRC y los diagramas de clases en el diseño de software orientado a objetos.
- Analizar casos de uso para identificar clases, sus responsabilidades y colaboraciones dentro de un sistema.
- Aplicar la metodología de fichas CRC para definir las clases, sus responsabilidades y sus interacciones en un proyecto de software.
- Construir diagramas de clases básicos a partir de fichas CRC para representar visualmente la estructura y relaciones de un sistema.
- Fomentar el pensamiento crítico y la colaboración en la identificación y modelado de clases en un problema real.
- Reflexionar sobre la utilidad de estas herramientas en la planificación y desarrollo de software antes de su implementación.

PROPÓSITO DE LA CLASE

Brindar a los estudiantes las herramientas conceptuales y prácticas para modelar sistemas de software mediante fichas CRC y diagramas de clases, permitiéndoles estructurar de manera clara y organizada los componentes de un sistema antes de su implementación. Esta comprensión facilitará la transición del análisis a la programación, mejorando la calidad del diseño y reduciendo errores en el desarrollo de software.

FICHAS CRC: CONCEPTO, USO Y CONSTRUCCIÓN

¿Qué son las fichas CRC?

Las fichas CRC (Class-Responsibility-Collaborator) son una técnica utilizada en el diseño de software orientado a objetos para identificar y definir las clases, sus responsabilidades y las colaboraciones entre ellas. Fueron introducidas por Kent Beck y Ward Cunningham en los años 80 como una herramienta sencilla y colaborativa para modelar sistemas antes de pasar a la codificación.

Cada ficha CRC representa una clase, especificando:

- Su nombre (qué representa en el sistema).
- Sus responsabilidades (qué debe hacer).
- Sus colaboradores (con qué otras clases se relaciona para cumplir sus responsabilidades).

FICHAS CRC: CONCEPTO, USO Y CONSTRUCCIÓN

¿Cómo se construyen las fichas CRC?

Para crear fichas CRC, se sigue un proceso estructurado:

1. Identificación de clases
   - A partir de un caso de uso o requerimiento, se identifican los conceptos clave del sistema.
   - Cada concepto relevante puede representar una posible clase.
2. Definición de responsabilidades
   - Se describen las acciones que la clase debe realizar.
   - Las responsabilidades deben ser concisas y enfocadas en lo que hace la clase.
3. Determinación de colaboradores
   - Se identifican las clases con las que interactúa para cumplir sus responsabilidades.
4. Revisión y validación
   - Se analizan las fichas para asegurar que cada clase tiene un rol bien definido y que las responsabilidades están bien distribuidas.

CARACTERÍSTICAS DE LAS FICHAS CRC

- **Simplicidad:** Son fáciles de entender y modificar sin herramientas especializadas.
- **Colaboración:** Facilitan la discusión y el trabajo en equipo en la fase de diseño.
- **Iteración rápida:** Se pueden ajustar rápidamente antes de formalizar un modelo UML o empezar a codificar.
- **Enfoque en la responsabilidad:** Se centran en definir lo que hace cada clase, evitando diseños confusos o sobrecargados.

VENTAJAS DE LAS FICHAS CRC

☑ **Fácil comprensión:** Son intuitivas y no requieren conocimientos avanzados de modelado UML.

☑ **Útiles en la fase de diseño:** Ayudan a visualizar el sistema antes de la implementación.

☑ **Favorecen el trabajo en equipo:** Pueden usarse en dinámicas grupales para discutir y mejorar el diseño.

☑ **Flexibles:** Permiten cambios rápidos sin necesidad de rehacer diagramas complejos.

DESVENTAJAS DE LAS FICHAS CRC

❌ No son una herramienta formal: No reemplazan UML ni otros diagramas más detallados.

❌ Limitadas en sistemas grandes: Para proyectos con muchas clases, se vuelven difíciles de gestionar sin una estructura adicional.

❌ Pueden depender de interpretación subjetiva: Si no se documentan bien, pueden generar diferentes entendimientos entre desarrolladores.

¿CÓMO SE HACEN LAS FICHAS CRC?

Las fichas CRC pueden hacerse en papel, en tarjetas físicas o con herramientas digitales. Su estructura básica es:

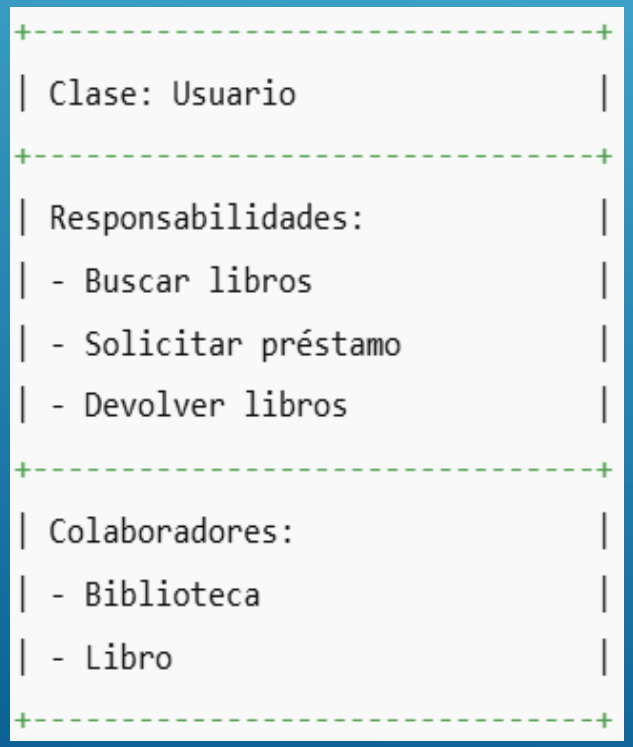
Formato de ficha CRC

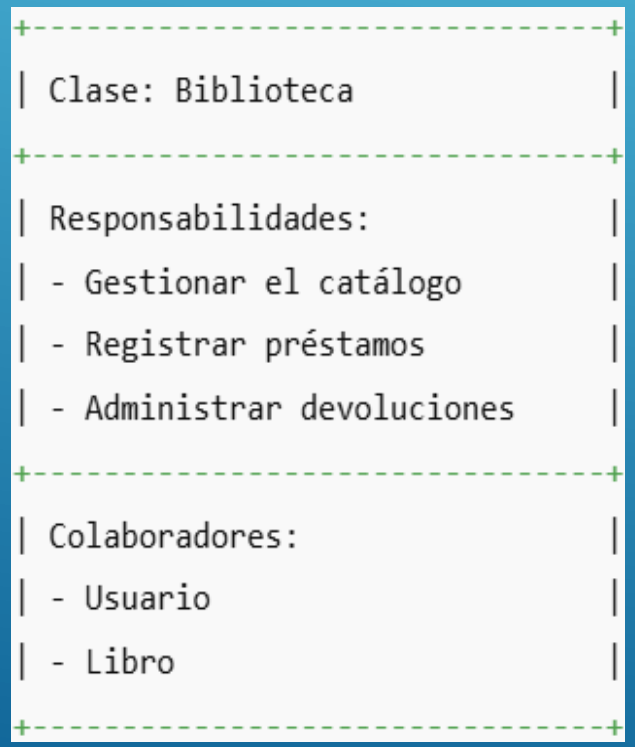
```
+---------------------------------+
| Clase: [Nombre de la clase]     |
+---------------------------------+
| Responsabilidades:              |
| - Acción 1                      |
| - Acción 2                      |
| - Acción 3                      |
+---------------------------------+
| Colaboradores:                  |
| - Clase 1                       |
| - Clase 2                       |
+---------------------------------+
```

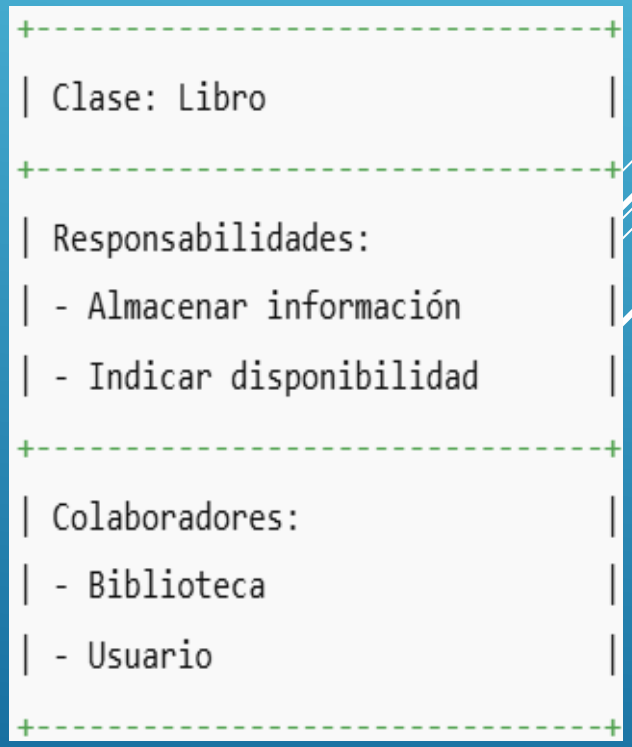
Tarjeta CRC	
[[Nombre del Proyecto]]	
Datos de la clase	
Nombre de la clase:	
Responsabilidades	**Colaboradores**

EJEMPLO 1: SISTEMA DE GESTIÓN DE BIBLIOTECA

- Clase: Usuario
- Clase: Biblioteca
- Clase: Libro

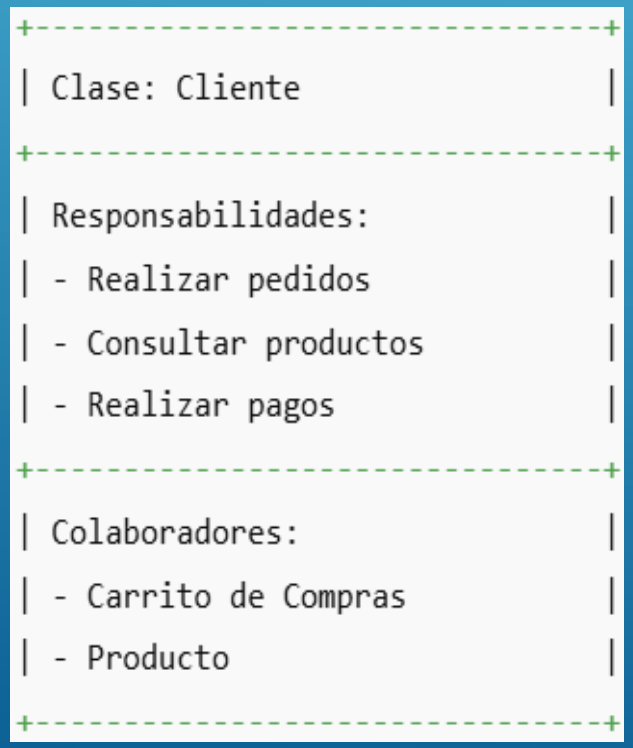


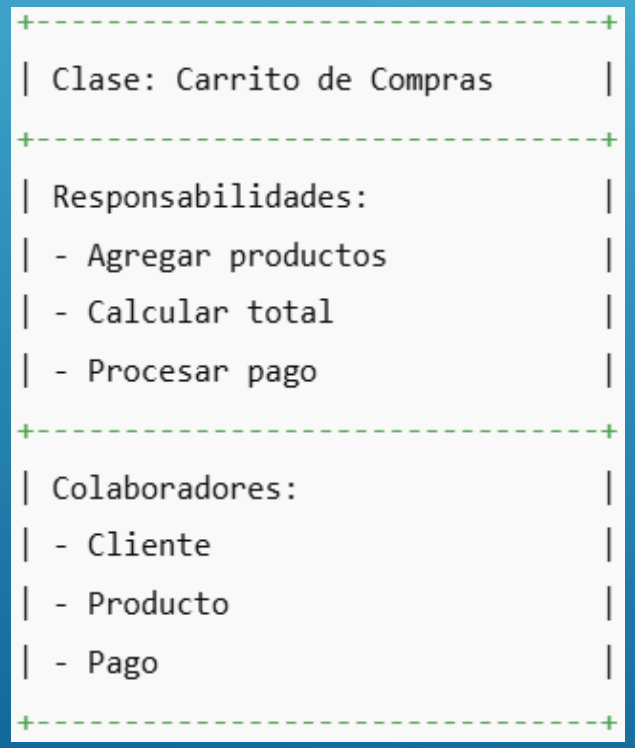


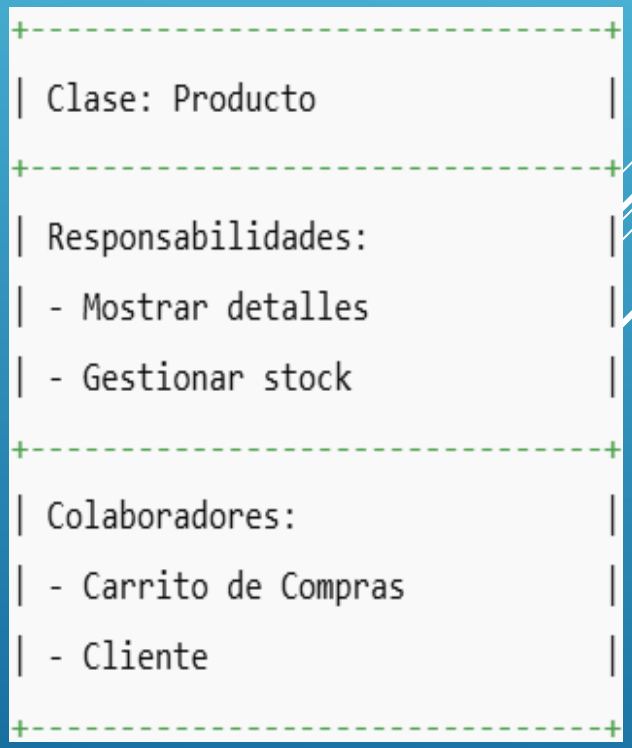


EJEMPLO 2: SISTEMA DE VENTA EN LÍNEA

- Clase: Cliente
- Clase: Carrito de Compras
- Clase: Producto

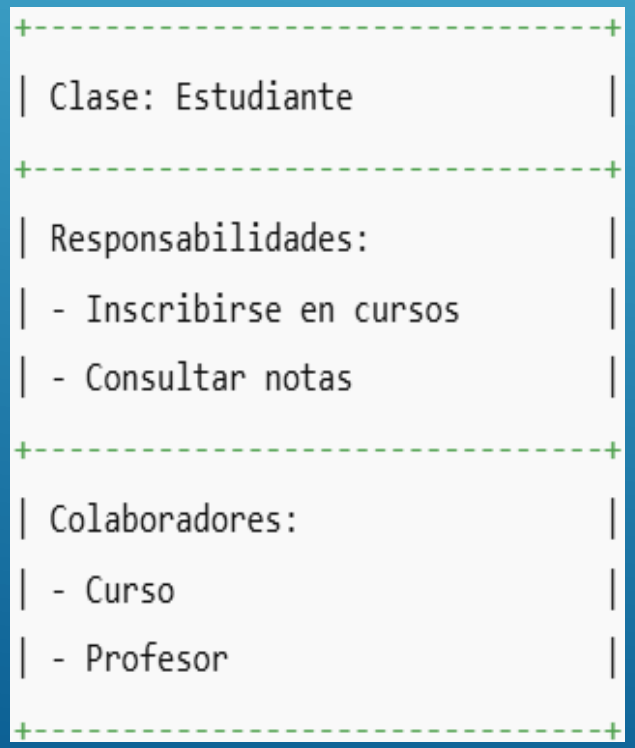


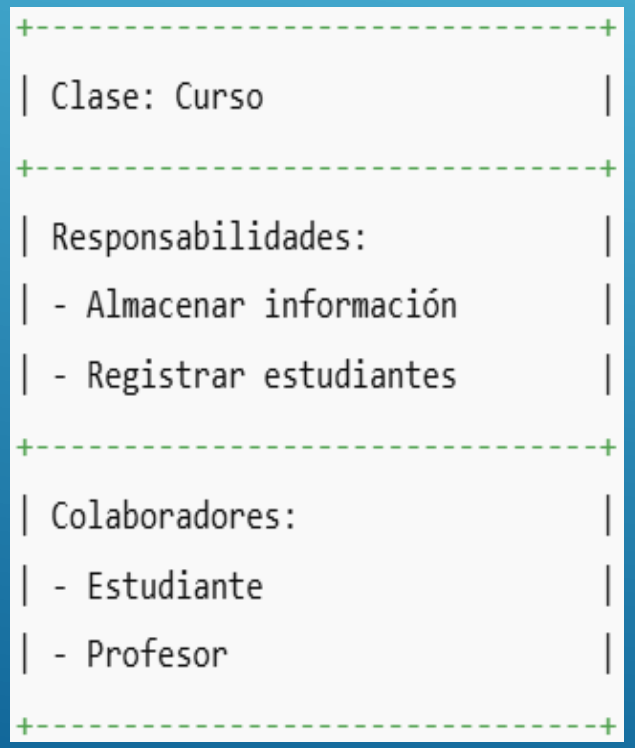


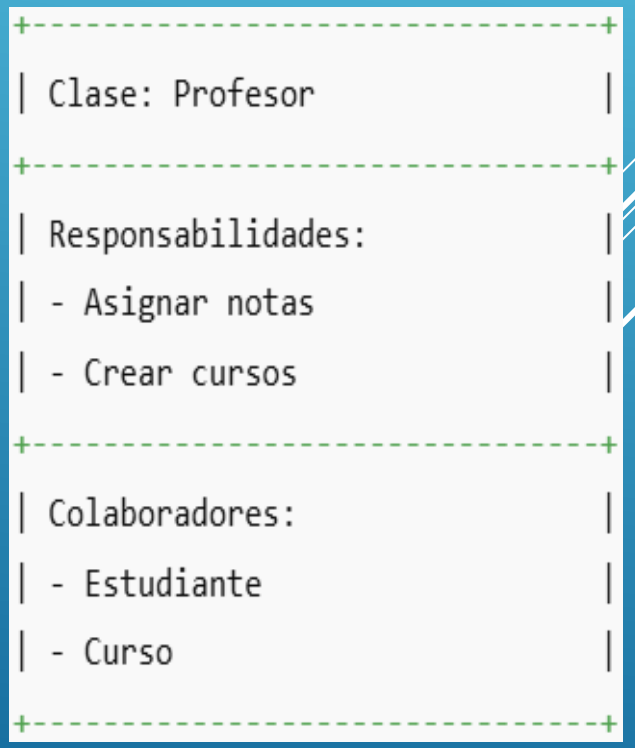


EJEMPLO 3: SISTEMA DE GESTIÓN DE ESTUDIANTES

- Clase: Estudiante
- Clase: Curso
- Clase: Profesor







¿QUÉ ES UN DIAGRAMA DE CLASES?

Un Diagrama de Clases es un modelo visual en UML (Unified Modeling Language) que representa la estructura de un sistema de software mediante clases, atributos, métodos y relaciones entre ellas. Es una herramienta fundamental en la programación orientada a objetos para definir cómo los objetos interactúan dentro de un sistema.

El diagrama de clases se usa para:

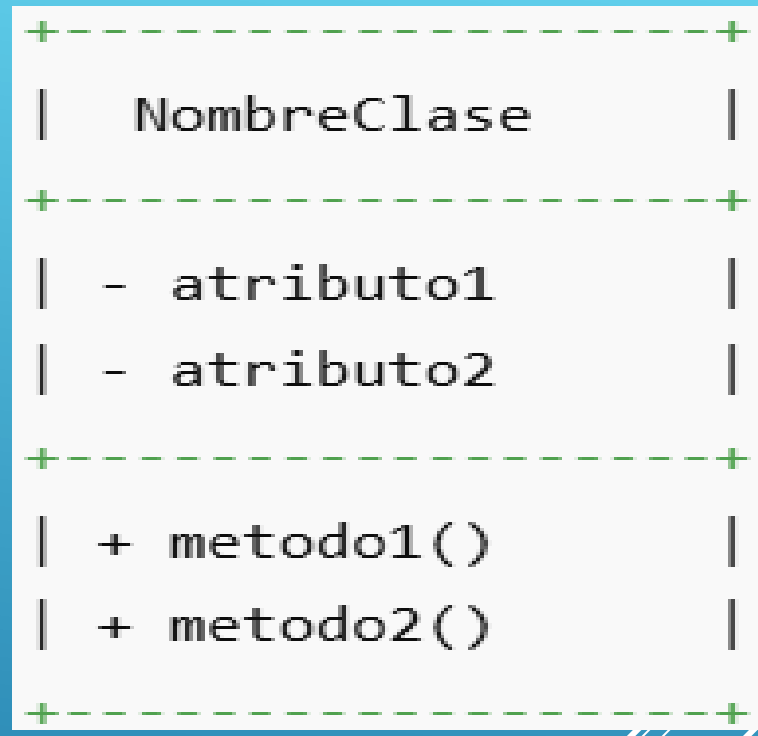
- Modelar la estructura del sistema antes de su implementación.
- Definir relaciones entre clases (herencia, asociación, composición, agregación).
- Visualizar la organización del código en sistemas grandes.
- Facilitar la comunicación entre desarrolladores, diseñadores y clientes.
- Mejorar la mantenibilidad y escalabilidad del software.

¿CÓMO SE REALIZA UN DIAGRAMA DE CLASES?

Para construir un diagrama de clases, se siguen estos pasos:

- Identificar las clases: Basándose en el análisis del sistema y fichas CRC.
- Definir los atributos: Determinar qué datos manejará cada clase.
- Definir los métodos: Especificar las operaciones que realizará cada clase.
- Establecer relaciones: Determinar cómo se conectan las clases (asociación, herencia, agregación, composición).
- Dibujar el diagrama: Representar gráficamente las clases y sus relaciones en UML.

EJEMPLO DE ESTRUCTURA DE UNA CLASE EN UML



Símbolos utilizados en UML:

- Clase: Representada como un rectángulo.
- Atributos: Propiedades de la clase.
- Métodos: Funcionalidades de la clase.
- Relaciones:
  - Asociación: Relación simple entre clases.
  - Herencia (Generalización): Una clase hija hereda de una clase padre.
  - Agregación: Relación "tiene un", donde una clase puede existir sin la otra.
  - Composición: Relación "es parte de", donde una clase no puede existir sin la otra.

¿DE DÓNDE SALE UN DIAGRAMA DE CLASES?

El diagrama de clases se deriva del análisis de requisitos del sistema, específicamente de:

- Casos de uso: Determinan los actores y funcionalidades del sistema.
- Fichas CRC: Ayudan a definir clases, responsabilidades y colaboraciones.
- Modelado conceptual: Identificación de entidades y relaciones clave.
- Diseño detallado: Refinamiento del modelo con atributos y métodos.

CARACTERÍSTICAS DEL DIAGRAMA DE CLASES

✓ Es visual y estructurado: Representa el sistema de manera clara.

✓ Muestra relaciones entre clases: Permite entender cómo interactúan.

✓ Es estático: No representa el comportamiento en tiempo de ejecución.

✓ Facilita la implementación: Ayuda a los programadores a escribir código organizado.

✓ Puede evolucionar: Se adapta a cambios en los requisitos del sistema.

VENTAJAS Y DESVENTAJAS DEL DIAGRAMA DE CLASES

☑ Claridad en el diseño: Permite ver cómo está estructurado el sistema.

☑ Facilita la comunicación: Entre desarrolladores, diseñadores y clientes.

☑ Reduce errores en el código: Al definir previamente la estructura.

☑ Favorece la reutilización: Se pueden reutilizar clases en otros proyectos.

☑ Permite detectar inconsistencias antes de programar.

❌ Puede volverse complejo en sistemas grandes.

❌ No representa la lógica de negocio en detalle: no detalla el comportamiento del sistema ni su flujo de ejecución.

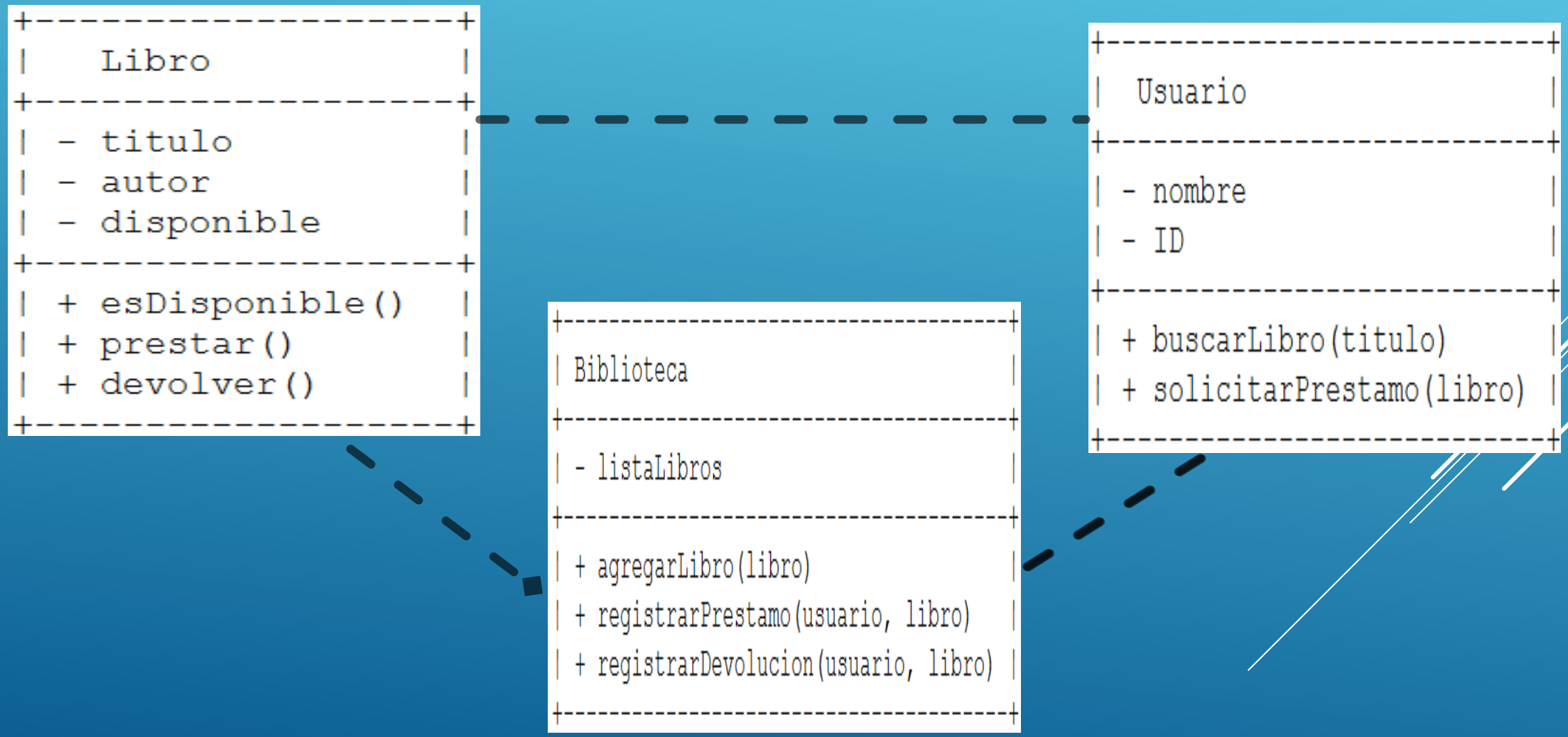
❌ Puede ser difícil de actualizar si el diseño cambia constantemente.

❌ Requiere conocimiento de UML para su interpretación.

HERRAMIENTAS DE ESCRITORIO PARA CREAR DIAGRAMAS DE CLASES

- StarUML (🔗 staruml.io)Software ligero y rápido con soporte para UML 2.0.
- Visual Paradigm (🔗 visual-paradigm.com)Potente herramienta UML con generación de código.
- Enterprise Architect (🔗 sparxsystems.com)Profesional y orientada a grandes proyectos de software.
- Astah (🔗 astah.net)Ligera y enfocada en modelado UML con diagramas intuitivos.
- Modelio (🔗 modelio.org)

EJEMPLO


Libro
- titulo
- autor
- disponible
+ esDisponible()
+ prestar()
+ devolver()
Usuario
- nombre
- ID
+ buscarLibro(titulo)
+ solicitarPrestamo(libro)
Biblioteca
- listaLibros
+ agregarLibro(libro)
+ registrarPrestamo(usuario, libro)
+ registrarDevolucion(usuario, libro)

EJEMPLO

📌 Explicación de las relaciones en el diagrama:

- ◇ Usuario —(Asociación)—> Biblioteca
  - Un usuario interactúa con la biblioteca (puede solicitar préstamos, devolver libros, etc.).

- ◇ Biblioteca ◇—(Agregación)—> Libro
  - La biblioteca contiene libros, pero los libros pueden existir sin la biblioteca (por ejemplo, en otra sucursal o colección personal).

- ◇ Usuario —(Asociación)—> Libro
  - Un usuario puede pedir prestado uno o varios libros.

INTEGRACIÓN DE FICHAS CRC CON DIAGRAMAS DE CLASES

Las Fichas CRC y los Diagramas de Clases están estrechamente relacionados, ya que ambos se usan en la fase de diseño de la programación orientada a objetos para estructurar un sistema antes de su implementación.

📌 ¿Cómo se integran?

Las fichas CRC ayudan a identificar y definir las clases de un sistema antes de representarlas en un diagrama de clases UML. Siguen este flujo:

1️⃣ Se crean las Fichas CRC para definir las clases, sus responsabilidades y colaboradores.

2️⃣ Se analizan las relaciones entre las clases a partir de los colaboradores identificados en las fichas CRC.

3️⃣ Se traducen las fichas CRC a un diagrama de clases UML, asignando atributos, métodos y relaciones entre clases.

CONCLUSIÓN

Las fichas CRC y los diagramas de clases se complementan dentro del proceso de diseño de software.

Las fichas CRC sirven como una herramienta exploratoria y colaborativa, permitiendo identificar responsabilidades y colaboraciones de manera simple. A partir de estas, se puede desarrollar un diagrama de clases estructurado, que represente con mayor precisión los detalles técnicos del sistema.

Usar ambos enfoques en conjunto mejora la comprensión, la organización y la calidad del diseño del software, asegurando una distribución adecuada de responsabilidades y una correcta implementación del sistema en el código.

PREGUNTAS