

# Proceso de Análisis de Casos de Uso

Análisis del Proceso Siguiendo Ejemplo Biblioteca.

## Proceso De Realización CU

### CU-Devolución de ejemplares a domicilio.v

Alcance del caso de uso: **Sistema**. Meta: **Usuario**. Interacción: **Dialogal**

Precondiciones:

- El sistema posee los préstamos del socio registrado.

Actores:

- Socio
- Bibliotecario.

Disparador:

- El socio se presenta con el ejemplar de uno o varios libros ante el bibliotecario.

### Camino basico:

1. El bibliotecario ingresa identificación del socio
2. El sistema:
  - 2.1. Valida que el socio exista.
  - 2.2. Muestra nombre y apellido del socio.
  - 2.3. Muestra los ejemplares pendientes de devolución (n° de ejemplar y título).
3. El bibliotecario selecciona los ejemplares a devolver y confirma la devolución.
4. El sistema:
  - 4.1. Registra la devolución.
  - 4.2. Emite comprobante de devolución.

### Caminos alternativos:

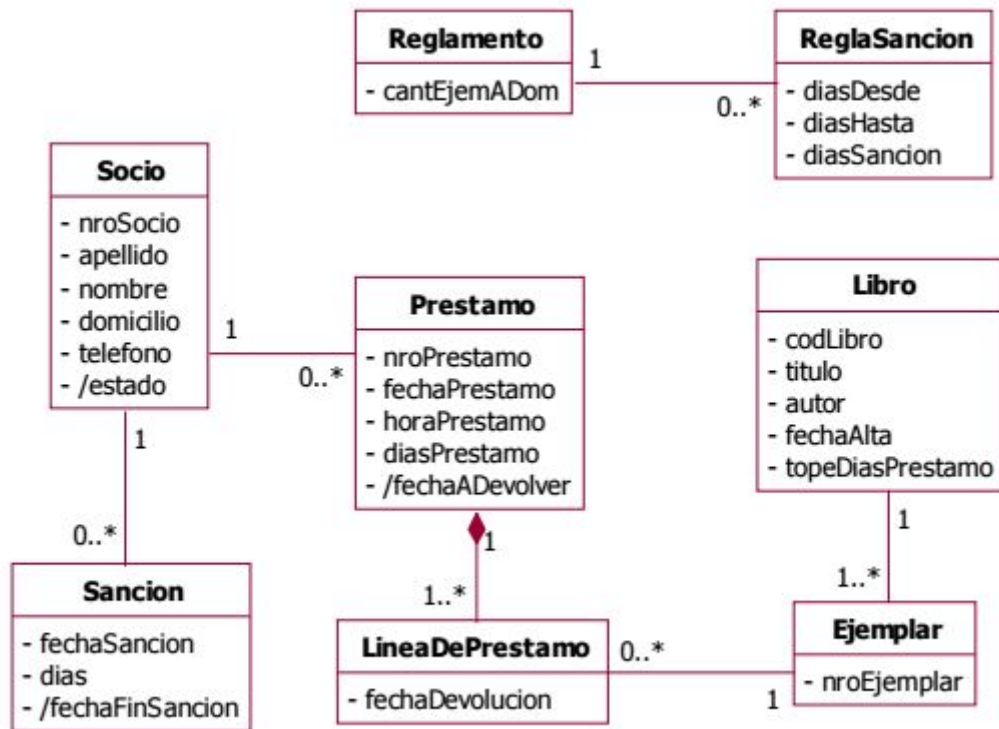
- 21A El socio no existe.
  1. El sistema muestra el mensaje "El número de socio no existe".
  2. Termina el caso de uso.
- 41A Cuando corresponda sanción (RN n° 5).
  1. Se registra la sanción
  2. Vuelve al paso 4.2 .

**Éxito:** Se registró la devolución.

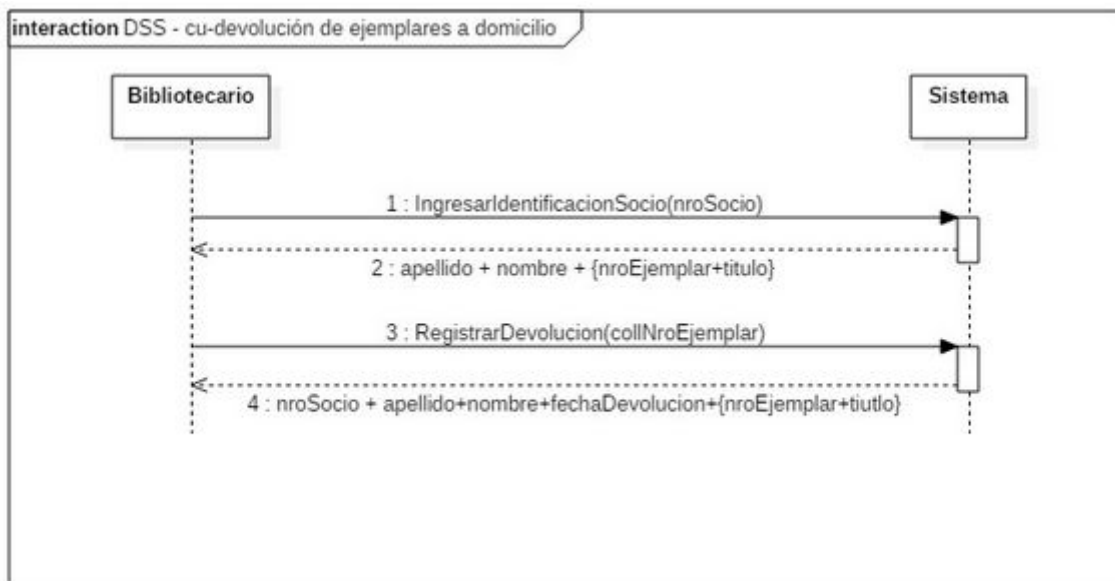
**Éxito de los caminos alternativos:** Se registró la sanción.

**Fracaso:** No se registró la devolución

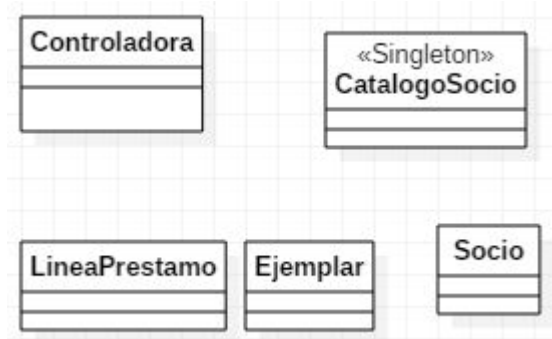
## Modelo de Dominio(MD)



## Diagrama De Secuencia de Sistema(DSS)



## Diagrama de clases de diseño (DCD) Inicial



**Agregamos las clases (sin atributos) que están en el Modelo de Dominio (MD) y que estén involucradas en el caso de uso en estudio (Devolución de los Ejemplares a Domicilio.)**

**Agregamos las clases de Diseño que hacen falta a priori y no están en el MD.**

**(Controladora y CatálogoSocio)**

A la clase **CatálogoSocio** le agregamos el estereotipo **Singleton**, para indicar que se trata de una clase de **instancia única**, donde se van a encontrar todos los libros del universo de la biblioteca. (Siempre los catálogos van a poseer como propiedad una lista del tipo de objeto a cual haga referencia el catálogo, por ejemplo el CatalogoSocio posee una propiedad que es una lista del objeto Socio)

## Diagrama de Secuencia de Diseño (DSD)

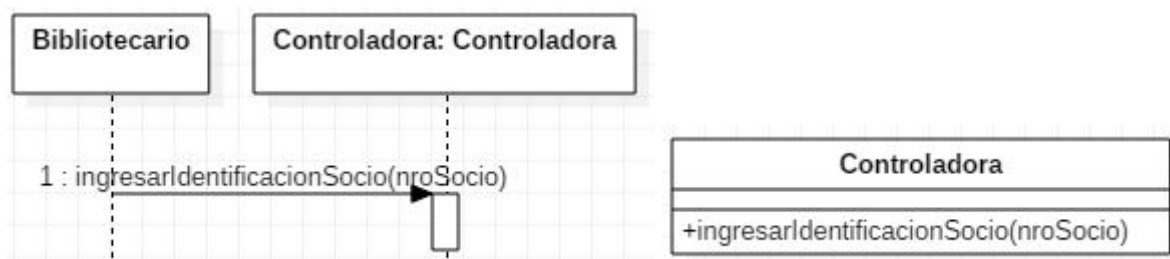
### IngresarIdentificacionSocio

Líneas de Vida iniciales

Creamos en el diagrama de secuencia IngresarIdentificacionSocio la línea de vida del actor Bibliotecario y la Línea de vida de la clase Controladora.



Mensaje - Método IngresarIdentificacionSocio



DSD: Agregamos el mensaje como IngresarIdentificacionSocio(nroSocio)

DCD: Agregamos el método IngresarIdentificacionSocio(nroSocio) a la clase Controladora.

Nota: Si utilizamos herramientas como starUML, u otras herramientas para el modelado UML por

cada mensaje agregado en el diagrama de secuencia se agrega en forma automática un método a la clase en el diagrama de clase.

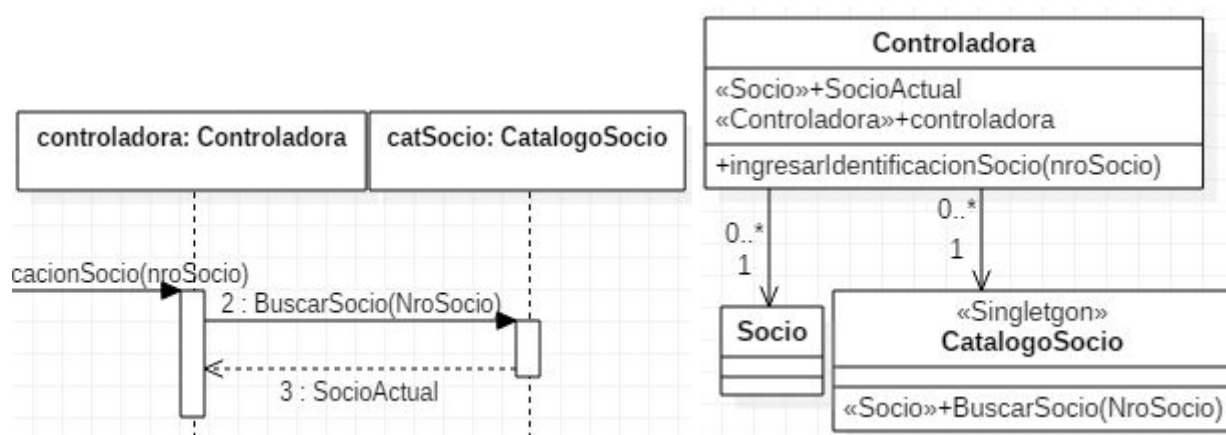
#### Mensaje - Método BuscarSocio

**Antes de agregar un mensaje** en el DSD primero tenemos que plantearnos lo siguiente:

- 1) ¿A qué **clase** se enviará el mensaje ?. En este caso la clase CatalogoSocio
- 2) ¿Debemos **guardar** el **objeto obtenido** o creado, para que lo use otro evento, u otro paso de nuestro DSS? Si es así, necesito asignarlo a una variable de instancia (Propiedad) de la controladora y crear la asociación.

La **variable de instancia** la podemos representar como, un **atributo** en la **clase** o como un **rol** en la **asociación**. Podríamos adoptar algunas de las dos maneras para nuestro curso, pero lo vamos a dejar a criterio del alumno.

En este caso es claro que una vez encontrado el socio, lo vamos a necesitar para completar el caso de uso.



DCD: Agregamos la **asociación** entre **Controladora** y **CatalogoSocio**, la multiplicidad [0..1] y la navegabilidad desde la clase Controladora hacia la clase CatalogoSocio. Los catálogos singleton serán siempre propiedades de las controladoras, en caso de que sea necesario enviarle mensajes.

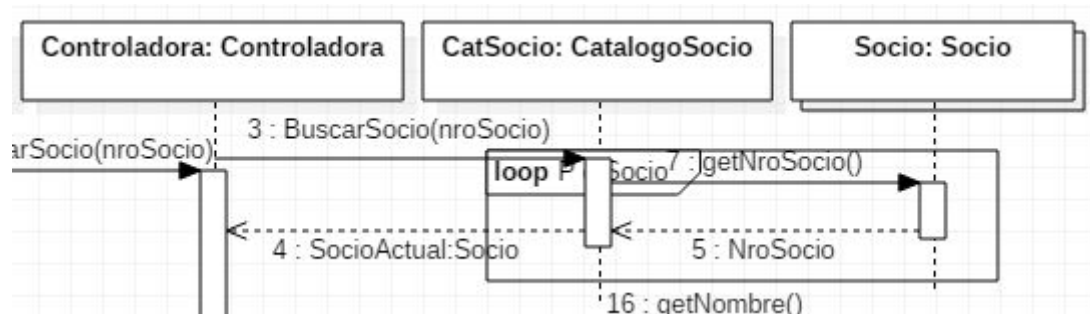
DSD: Agregamos la línea de vida de la instancia **catSocio** de la clase **CatalogoSocio**.

Agregamos el mensaje BuscarSocio(nroSocio) desde controladora a catSocio:CatalogoSocio

DSD: Agregamos el mensaje de retorno desde el catálogo a la controladora que devuelve el SocioActual.

DCD: Agregamos como propiedad SocioActual a la controladora. Y también agregamos la asociación.

#### Mensaje - Método getNroSocio

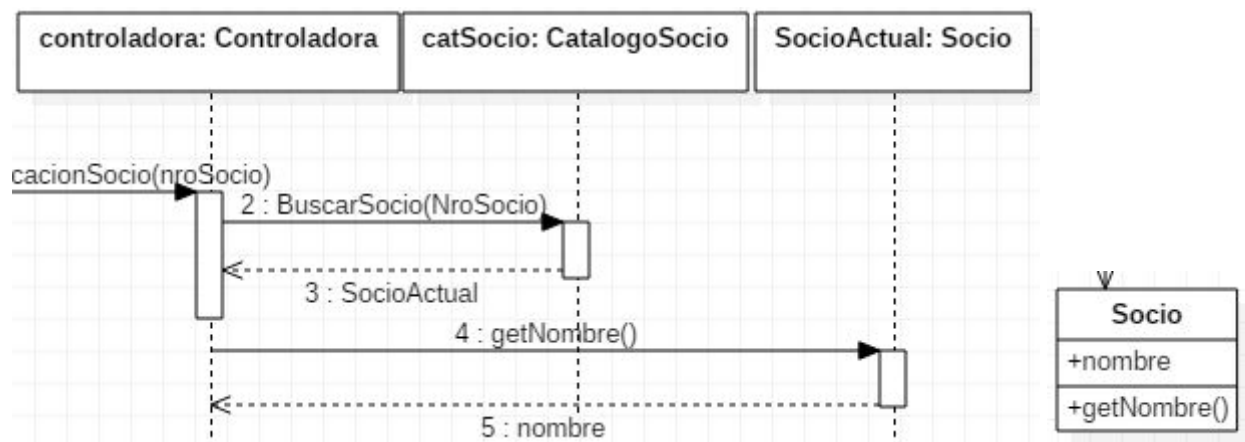


El catálogo de socios posee una lista de objetos Socios. En esa colección vamos a buscar el socio.

DSD:(Esto sucede en la línea de vida del catalogo del socio, es decir dentro del método BuscarSocio) Agregamos la colección de socios; cada una de las instancias de la colección se llama "Socio". Para indicar que se trata de una colección, hacemos un doble cuadro. Agregamos la línea de vida de la instancia Socio. Para buscar el socio (Método Contenedor), pedimos a cada una de las instancias de la colección el número de socio, mediante el método getNroSocio(). Para indicar que enviamos el mensaje a c/u de las instancias de la colección, utilizamos el recuadro, indicando que se trata de un loop. El mensaje de retorno devuelve el nroSocio. Mediante una nota indicamos que comparamos ese número con el ingresado, cuando lo encontramos paramos el loop.

DCD: Agregamos como propiedad del Catalogo de socios la colección de socios. Además agregamos el método getNroSocio() a la clase Socio.

Mensaje - Método getNombre



DSD: Agregamos la línea de vida de la instancia SocioActual de la clase socio. Agregamos el mensaje `getNombre` desde la controladora al SocioActual.

Agregamos también el mensaje de retorno que devuelve el nombre.

DCD: Agregamos a la clase socio, la propiedad `nombre`, y el método `getNombre`.

.....