

---

# Diseño Casos de uso

**Proyecto: Aplicación para la gestión de una protectora de animales**



**Versión:** 4.0

**Fecha:** 11/06/2023

**Equipo de desarrollo:** Soft4U

**Carmen Esau Navas**

**Ibrahim Pérez de León de los Santos**

Cliente: ARCA

Carlos Sánchez Lancha

## LISTA DE CONTROL DE CAMBIOS

Versión	Causa del Cambio	Fecha del Cambio
1.0	Versión inicial, índice y portada	11/04/2023
1.1	Introducción, caso de uso y dependencias	12/04/2023
2.0	Diagramas: de clase, de caso de uso, de actividad y de estados	14/04/2023
2.1	Patrón DAO diagrama de clases	16/04/2023
2.2	Diagramas: secuencia y objetos	17/04/2023
3.0	Organización del documento y formato de documento	18/04/2023
3.1	Añadir casos de uso nuevos y sus diagramas	14/05/2023
4.0	Versión final	11/06/2023

# ÍNDICE

<b>LISTA DE CONTROL DE CAMBIOS.....</b>	<b>1</b>
<b>1. INTRODUCCIÓN.....</b>	<b>3</b>
<b>2. ORGANIZACIÓN DEL DOCUMENTO.....</b>	<b>3</b>
<b>3. CASOS DE USO MODELADOS.....</b>	<b>3</b>
a. Dependencias .....	7
<b>4. PATRONES.....</b>	<b>10</b>
<b>5. MODELADO PARA EL CASO DE USO 02: SOLICITAR ADOPCIÓN .....</b>	<b>12</b>
a. Diagrama de casos de uso .....	12
b. Diagrama de clases .....	13
c. Diagrama de actividad .....	14
d. Diagrama de estados.....	15
e. Diagrama de secuencia .....	15
f. Diagrama de objetos.....	16
<b>6. MODELADO PARA EL CASO DE USO 03: REALIZAR DONACIÓN MONETARIA .....</b>	<b>17</b>
a. Diagrama de casos de uso .....	17
b. Diagramas de clases .....	17
c. Diagrama de actividad .....	19
d. Diagrama de estados.....	20
e. Diagrama de secuencia .....	20
f. Diagrama de objetos.....	21
<b>7. MODELADO PARA EL CASO DE USO 04 .....</b>	<b>22</b>
a. Diagrama de casos de uso .....	22
b. Diagrama de clases .....	22
c. Diagrama de actividad .....	23
d. Diagrama de estados.....	24
e. Diagrama de secuencia .....	25
f. Diagrama de objetos.....	26

## 1. INTRODUCCIÓN

El objetivo de este documento es presentar el diseño de tres de los casos de uso planteados en nuestro proyecto 'Aplicación para la gestión de una protectora de animales'. En específico trabajaremos el caso de uso 02: Solicitar adopción, el caso de uso 03: Realizar donación monetaria, y el caso de uso 04: Generar recibo de donación, recogidos en el documento 'Especificación de Requisitos del Sistema'. El diseño se realizará mediante diagramas UML realizados con el software 'MagicDraw UML'.

## 2. ORGANIZACIÓN DEL DOCUMENTO

En cuanto a la organización del documento comenzamos viendo las tablas de los casos de uso que vamos a modelar, así como las tablas de los requisitos de información con los que tienen dependencias. Después pasamos a comentar los patrones que hemos implementado en el diseño de nuestro modelo y las razones por las cuales los usamos. Por último mostramos los diagramas UML que consideramos que resumen el modelado de los casos de uso que hemos seleccionado, además de algún diagrama extra que ayuda a la comprensión y entendimiento del diseño de los casos de uso.

## 3. CASOS DE USO MODELADOS

<b>CU02</b>	<b>Solicitar adopción</b>
<b>[Versión]</b>	3.0
<b>[Dependencias]</b>	<ul style="list-style-type: none"><li>• Caso de Uso 01: Iniciar sesión</li><li>• Requisito de Información 01 : Datos de un animal</li><li>• Requisito de Información 03 : Formulario de adopción</li><li>• Requisito de Fiabilidad 01: Validación de datos</li><li>• Requisito de Usabilidad 01: Interfaz de Usuario</li></ul>
<b>Actores</b>	<ul style="list-style-type: none"><li>• ACT 02: Cliente</li></ul>

Precondición	El cliente ha iniciado sesión.			
Descripción	La aplicación deberá comportarse como se describe en el siguiente caso de uso cuando un cliente quiera solicitar la adopción de un animal de la protectora.			
Secuencia Normal	Paso	Acción		
	1	Seleccionar en el menú que muestra la aplicación la opción 'Solicitar adopción'.		
	2	Seleccionar animal que desee adoptar del listado que se le muestra por pantalla.		
	3	Rellenar el formulario de adopción. (Los campos del formulario de adopción están descritos en el Requisito de Información 03.)		
	4	El cliente envía el formulario de adopción relleno.		
	5	La aplicación verifica que los campos del formulario están correctamente rellenos.		
	6	La aplicación guarda la información del formulario.		
	7	La aplicación muestra por pantalla un mensaje de que el formulario se ha enviado correctamente.		
Postcondición	El estado de adopción del animal seleccionado pasa de 'libre' a 'reservado'.			
Excepciones	Caso	Acción		
	1	Si el formulario no está correctamente relleno		
		1.1	La aplicación muestra un mensaje con los campos que están incorrectamente rellenos.	
		1.2	El cliente rellena de nuevo el formulario y lo envía.	
		1.3	La aplicación comprueba que todos los campos estén correctamente rellenos.	
		1.4	Se muestra por pantalla un mensaje de que el formulario se ha enviado correctamente.	

<b>Comentarios</b>	Los diferentes estados de adopción de un animal están recogidos en el Requisito de Información 01.
--------------------	--

<b>CU03</b>	<b>Realizar donación monetaria</b>	
<b>[Versión]</b>	3.0	
<b>[Dependencias]</b>	<ul style="list-style-type: none"> <li>• Caso de Uso 04: Generar recibo de donación</li> <li>• Caso de Uso 05: Iniciar sesión</li> <li>• Requisito de Información 04: Datos de una donación</li> <li>• Regla de Negocio 02: Cantidades donaciones monetarias</li> <li>• Regla de Negocio 03: Métodos de pago</li> <li>• Requisito de Usabilidad 01: Interfaz de usuario</li> </ul>	
<b>Actores</b>	<ul style="list-style-type: none"> <li>• ACT 02: Cliente</li> </ul>	
<b>Precondición</b>	El cliente ha iniciado sesión.	
<b>Descripción</b>	La aplicación deberá comportarse como se describe en el siguiente caso de uso cuando un cliente quiera realizar una donación monetaria.	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El cliente selecciona en el menú de inicio 'Realizar donación'.
	2	El cliente selecciona el tipo de donación, monetaria. (Opciones especificadas en Requisito de Información 04: Datos de una donación)
	3	El cliente selecciona la cantidad de dinero que se quiera donar. (Opciones especificadas en Requisito de Información 04: Datos de una donación)
	4	El cliente selecciona el método de pago. (Opciones especificadas en Requisito de Información 04: Datos de una donación)
	5	Según el método de pago elegido, el cliente tendrá que rellenar la información de su tarjeta o la aplicación le mostrará por pantalla el número de teléfono a la que enviar el bizum.

	6	Una vez realizado el pago, la aplicación muestra por pantalla un mensaje de que se ha realizado con éxito la donación.		
	7	La aplicación guarda los datos de la donación en la base de datos.		
<b>Postcondición</b>	La información correspondiente a la donación se ha almacenado en la base de datos y se ejecuta el Caso de Uso 04: Generar recibo de donación.			
<b>Excepciones</b>	<b>Caso</b>	<b>Acción</b>		
	1	El pago ha sido rechazado por falta de fondos.		
		1.1	La aplicación muestra por pantalla un mensaje de pago rechazado.	
		1.2	Se cancela la operación.	

<b>CU04</b>	<b>Generar recibo de donación</b>		
<b>[Versión]</b>	3.0		
<b>[Dependencias]</b>	<ul style="list-style-type: none"> <li>• Caso de Uso 03: Realizar donación monetaria</li> <li>• Requisito de Información 02: Datos de un usuario</li> <li>• Requisito de Información 04: Datos de una donación</li> <li>• Requisito de Información 05: Recibo de donación</li> </ul>		
<b>Actores</b>			
<b>Precondiciones</b>	El cliente ha realizado una donación.		
<b>Descripción</b>	La aplicación deberá comportarse como se describe en el siguiente caso de uso cuando haya que generar y enviar el recibo de una donación.		
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	La aplicación consulta en la base de datos la información de la donación realizada.	
	2	La aplicación rellena los campos de un recibo de donación con los datos correspondientes. (Los campos del recibo de	

		donación están especificados en el Requisito de Información 05: Recibo de donación.)
	3	La aplicación almacena el recibo en la base de datos.
	4	La aplicación envía el recibo al correo del cliente que ha realizado la donación.
<b>Postcondición</b>	El recibo de la donación ha sido almacenado en la base de datos.	
<b>Excepciones</b>	<b>Caso</b>	<b>Acción</b>
	1	Error al acceder a la base de datos.
	1.1	Tras un tiempo de 10 minutos, la aplicación vuelve a intentar realizar el caso de uso 04.

### a. Dependencias

Con la finalidad de aportar más claridad al caso de uso, presentamos los requisitos de información con los que tienen dependencias.

<b>RI01</b>	<b>Datos de un animal</b>
<b>Versión</b>	1.2
<b>Dependencias</b>	Ninguna
<b>Descripción</b>	La aplicación deberá gestionar la información correspondiente al animal.



<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Fecha de registro: en formato dd/mm/aaaa indicando fecha en la que se ha registrado el animal en el sistema.</li> <li>• Especie: palabra de máximo 5 caracteres indicando la especie del animal. (perro o gato)</li> <li>• Raza: palabra de máximo 20 caracteres indicando la raza del animal. (p.ej: gato pardo, pastor alemán, etc).</li> <li>• Edad: número de dos cifras indicando la edad del animal.</li> <li>• Estado: palabra de máximo 20 caracteres indicando el estado del animal. Opciones: cuarentena, pendiente de revisión médica, ingresado, sano.</li> <li>• Adopción: palabra de máximo 10 caracteres indicando si el animal está reservado por adoptantes, en casa de acogida, o libre para ser adoptado.</li> <li>• Tamaño: campo que siga el siguiente formato: altura en centímetros x el peso del animal en kilogramos. (p.ej: 50cm x 20kg)</li> <li>• Género: palabra indicando el sexo del animal, dos opciones posibles: macho, hembra.</li> <li>• Foto: imagen de máximo 25 mb en formato jpg o png donde se identifique al animal en cuestión.</li> </ul>
--------------------------	--

<b>RI02</b>	<b>Datos de un usuario</b>
<b>[Versión]</b>	1.2
<b>[Dependencias]</b>	Ninguna
<b>Descripción</b>	La aplicación deberá gestionar la información correspondiente a los datos del usuario.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Nombre: palabra de máximo 20 caracteres indicando el nombre del usuario.</li> <li>• Apellidos: palabra de máximo 20 caracteres indicando los apellidos del usuario.</li> <li>• DNI: campo compuesto por 8 cifras y terminado con 1 letra indicando el DNI del usuario.</li> <li>• Teléfono: número de 9 cifras indicando el teléfono personal del usuario.</li> </ul>

	<ul style="list-style-type: none"> <li>• Rol: 1 o 2 palabras de máximo 20 caracteres indicando el tipo de usuario dentro de la aplicación: voluntario senior, voluntario junior o cliente.</li> </ul>
--	---

RI03	Formulario de adopción
Versión	3.1
Dependencias	<ul style="list-style-type: none"> <li>• Requisito de Información 02: Datos de un usuario</li> </ul>
Descripción	El sistema deberá almacenar la información correspondiente al formulario de adopción de un animal.
Datos específicos	<ul style="list-style-type: none"> <li>• Información personal del cliente: <ul style="list-style-type: none"> <li>- Nombre completo</li> <li>- Dirección</li> <li>- Número de teléfono</li> <li>- Correo electrónico</li> </ul> </li> <li>• Tipo de vivienda en la que vivirá el animal: campo de selección en el que se tendrá que escoger una de las siguientes opciones: "Casa con jardín grande", "Casa con jardín pequeño", "Apartamento con terraza o balcón", "Apartamento sin terraza o balcón", "Finca o terreno grande", "Otro" (en este caso, se podría incluir un campo de texto libre para que el adoptante pueda especificar el tipo de vivienda).</li> <li>• El cliente es propietario de la vivienda: Botón de opción (radio button). Marcar si el cliente es el propietario de la vivienda.</li> <li>• El cliente tiene otros animales en el hogar: Botón de opción (radio button). En el caso de que se marque este campo se abrirá un campo de selección con las siguientes opciones: "Sí, esterilizados y vacunados", "Sí, esterilizados pero no vacunados", "Sí, vacunados pero no esterilizados" y "Sí, no esterilizados ni vacunados".</li> <li>• Número de niños en el hogar: número de 2 cifras como máximo.</li> <li>• El cliente tiene experiencia previa con animales: campo de selección con las siguientes opciones: "Experiencia previa con perros", "Experiencia previa con gatos", "Experiencia previa con animales exóticos", "Ninguna experiencia previa".</li> <li>• Motivación para adoptar un animal: campo de texto de 200 caracteres como máximo.</li> </ul>

	<ul style="list-style-type: none"> <li>• El cliente ya tiene veterinario seleccionado: Botón de opción (radio button). En el caso de que se marque ésta opción se deberá rellenar los siguientes campos: <ul style="list-style-type: none"> <li>◦ Nombre del veterinario</li> <li>◦ Nombre de la clínica</li> </ul> </li> <li>• El cliente se compromete a proporcionar atención médica y nutricional adecuada al animal: Botón de opción (radio button).</li> <li>• El cliente se compromete a mantener al animal en su hogar y no abandonarlo: Botón de opción (radio button).</li> <li>• El cliente se compromete a permitir visitas de seguimiento por parte de la protectora: Botón de opción (radio button).</li> <li>• El cliente entiende que la adopción es un compromiso a largo plazo: Botón de opción (radio button).</li> <li>• El cliente se compromete a cumplir con todas las leyes y regulaciones pertinentes en relación con la tenencia de animales: Botón de opción (radio button).</li> </ul>
--	--

<b>RI04</b>	<b>Datos de una donación</b>
<b>[Versión]</b>	3.1
<b>[Dependencias]</b>	<ul style="list-style-type: none"> <li>• Requisito de Información 02: Datos de un usuario</li> <li>• Regla de Negocio 02: Cantidades donaciones monetarias</li> <li>• Regla de Negocio 03: Métodos de pago</li> </ul>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a las donaciones que hagan los clientes
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• ID: número identificador de la donación.</li> <li>• Cliente: datos del cliente que realiza la donación (Datos de usuario recogidos en el Requisito de Información 02).</li> <li>• Fecha: en formato día/mes/año, fecha en la que se realiza la donación.</li> <li>• Tipo: monetaria o material.</li> <li>• Estado: palabra de máximo 10 caracteres que describe el estado de una donación. Estados posibles: iniciada, en proceso, pagada, rechazada, registrada.</li> <li>• Importe: las opciones posibles se recogen en la Regla de Negocio 02: Cantidades donaciones monetarias.</li> <li>• Método de pago: los posibles métodos de pago se especifican en la Regla de Negocio 03: Métodos de pago.</li> </ul>

	<ul style="list-style-type: none"> <li>Material: si la donación es material, breve descripción del material donado.</li> </ul>
--	--

<b>RI05</b>	<b>Recibo de donación</b>
<b>[Versión]</b>	3.1
<b>[Dependencias]</b>	<ul style="list-style-type: none"> <li>Requisito de Información 04: Datos de una donación.</li> <li>Caso de Uso 04: Generar recibo de donación</li> </ul>
<b>Descripción</b>	La aplicación deberá generar un recibo de la donación hecha por un cliente que contenga la siguiente información
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>ID: número identificador del recibo.</li> <li>Donación: datos correspondientes a la donación específica especificados en el Requisito de Información 04: Datos de una donación.</li> <li>Estado: vacío, completado, guardado o enviado.</li> <li>Fecha: en la que se genera el recibo.</li> </ul>

## 4. PATRONES

- DAO : objeto de acceso a datos

Una de las razones por las que hemos decidido implementar este patrón es separar la lógica de acceso a datos de la lógica de negocio que hace que el código sea más fácil de mantener y modificar. Además, el patrón DAO facilita la implementación de cambios en la base de datos, ya que los cambios sólo deben realizarse en las clases DAO, en lugar de en toda la aplicación. Al utilizar este patrón, se puede mejorar la calidad del código y hacer que el mantenimiento de la aplicación sea más sencillo a largo plazo.

En nuestro caso, nos ha servido de mucha utilidad a la hora de manejar los datos de usuarios así como los datos de animales a lo largo de todo el proceso

de diseño. Ya que cuando el sistema esté completo y en funcionamiento se prevé que haya un elevado número de datos e información que manejar.

Podemos observar con claridad la implementación de éste patrón en los diagramas de clases denominados DAO, disponibles para cada uno de los casos de uso.

- MVC: modelo vista controlador

En nuestro caso, aplicado al diagrama de secuencia llamado Secuencia Adopción, el actor Cliente interactúa con la interfaz de usuario ClienteUI que forma parte de la vista, el controlador corresponde a la clase Aplicación que se encarga de procesar la entrada del usuario y coordinar la interacción entre la vista y el modelo, el modelo corresponde a la interfaz AdopciónDAO que se comunica con el sistema externo BBDD para obtener o actualizar los datos relevantes.

Similar al diagrama anterior, podemos ver el patrón MVC aplicado al diagrama de secuencia Realizar Donación, el actor Cliente interactúa con la interfaz de usuario ClienteUI que forma parte de la vista, el controlador corresponde al contenedor de Servlets Tomcat junto con los Servlets, y el modelo corresponde a la entidades DonacionMonetaria y JpaDonacionMonetariasDAO. Se utiliza el mismo procedimiento de aplicación de dicho patrón en el diagrama de secuencia Generar Recibo, a falta de incluir en dicho diagrama los Servlets correspondientes.

Las ventajas del uso del patrón MVC en este diagrama son:

- Separación de responsabilidades: El patrón MVC permite separar las responsabilidades de las diferentes partes del sistema (vista, controlador, modelo), lo que facilita la modificación y el mantenimiento del código.
- Reutilización de código: Al separar las diferentes partes del sistema, se puede reutilizar el código de manera más eficiente.
- Facilidad para la implementación de pruebas unitarias: Al separar el modelo de la vista, se puede probar la lógica del modelo sin necesidad de involucrar la vista o el controlador.

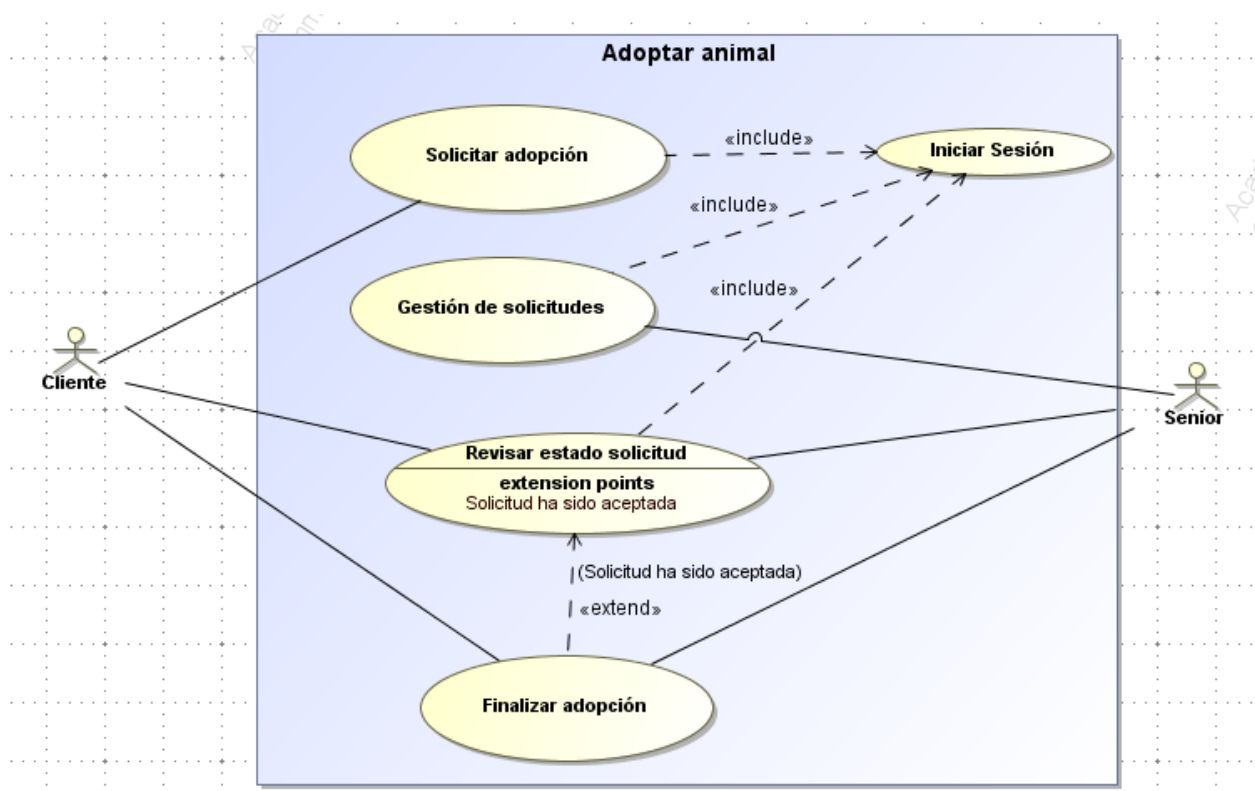
Las desventajas del uso del patrón MVC en este diagrama podrían ser:

- Complejidad: Al dividir el sistema en tres partes distintas, puede haber una mayor complejidad en el diseño y en la implementación.
- Sobrecarga de clases e interfaces: Al utilizar tres partes distintas para el sistema, puede haber una sobrecarga de clases e interfaces.

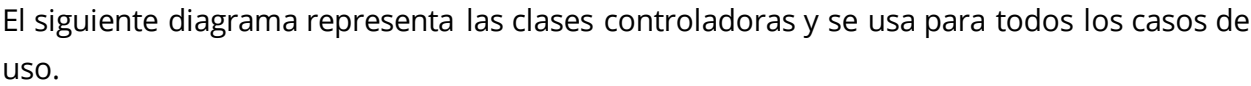
## 5. MODELADO PARA EL CASO DE USO 02: SOLICITAR ADOPCIÓN

### a. Diagrama de casos de uso

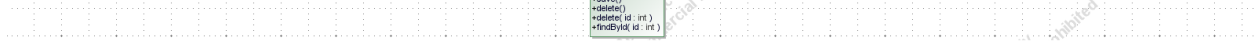
Observamos que para adoptar a un animal se requiere realizar varios casos de uso, para los cuales hay que iniciar sesión previamente. La adopción sólo será finalizada si la solicitud ha sido aceptada.



En este diagrama se representan las clases relacionadas con este caso de uso. Además de un diagrama de las clases controladoras en nuestra aplicación.

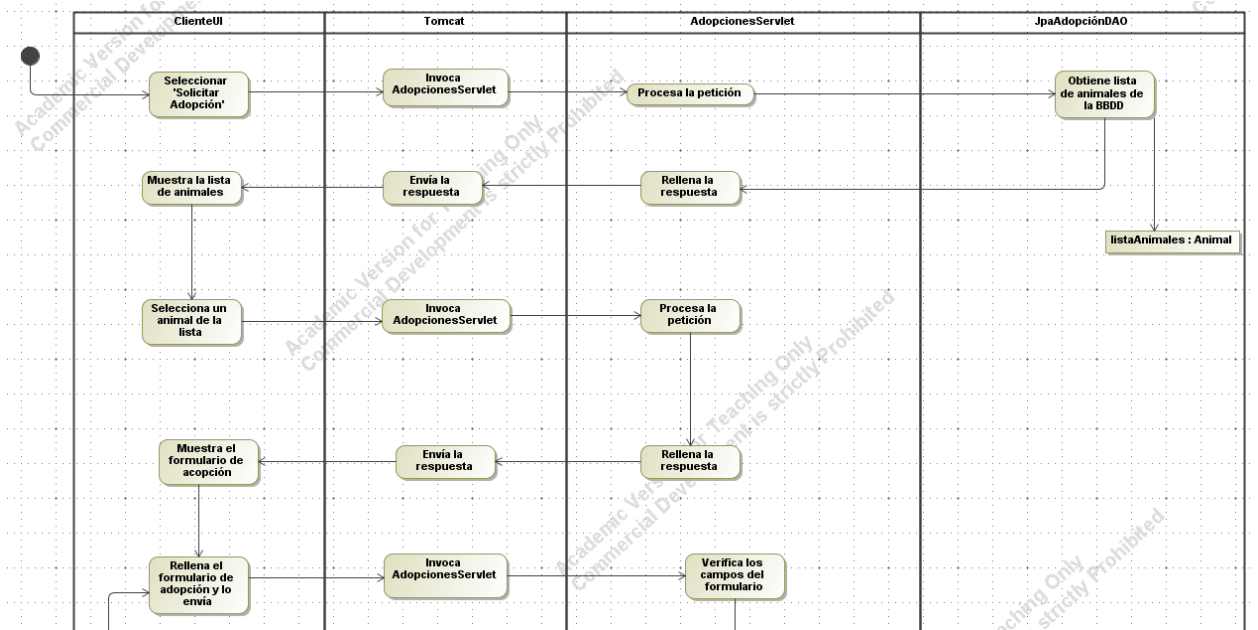


El siguiente diagrama representa las clases controladoras y se usa para todos los casos de uso.

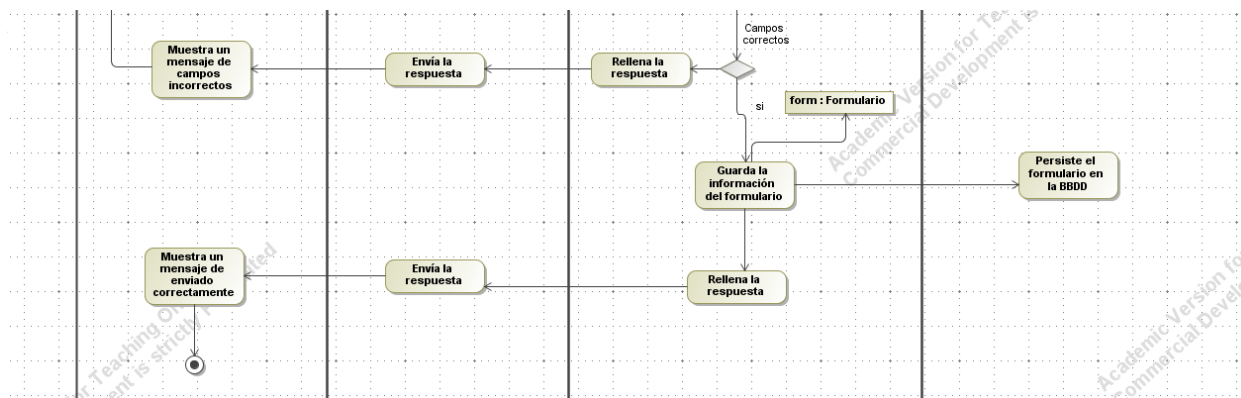


c. Diagrama de actividad

En el diagrama de actividad podemos observar las distintas acciones realizadas para completar el caso de uso y quién las realiza.

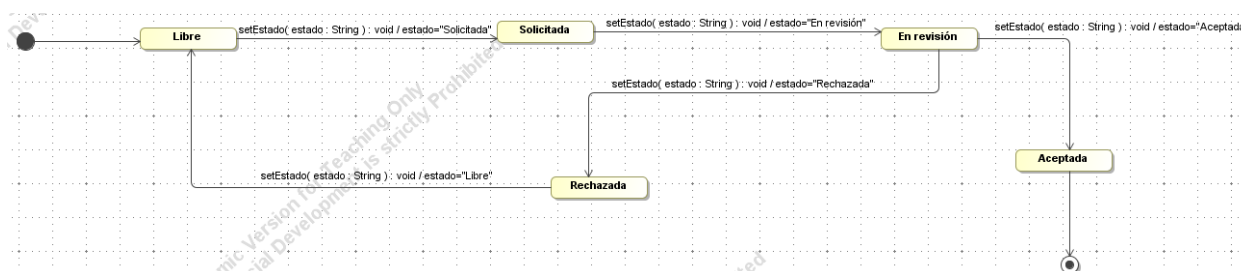






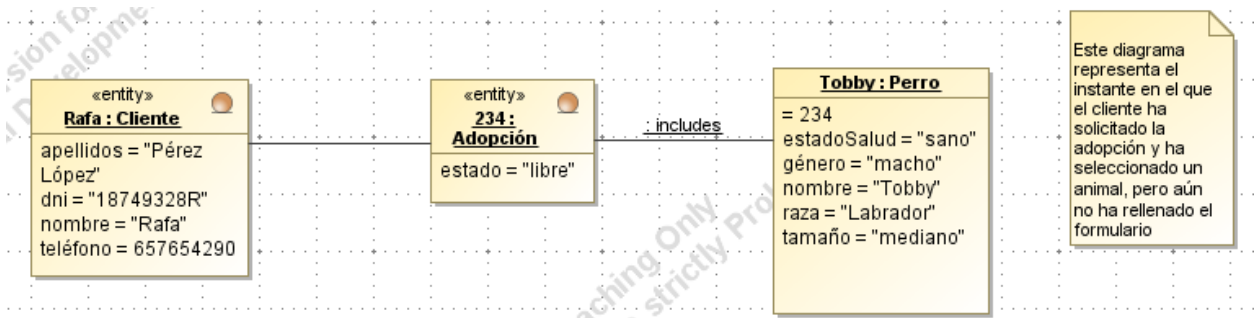
#### d. Diagrama de estados

En este diagrama podemos observar los distintos estados en los que puede estar un objeto de tipo adopción. También observamos las distintas operaciones que desencadenan el cambio de estado.

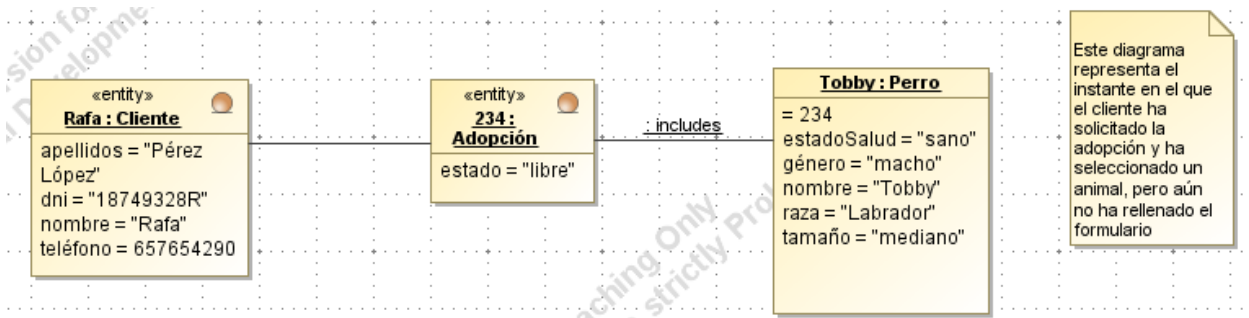


#### e. Diagrama de secuencia

En este diagrama implementamos el patrón M/V/C para representar la secuencia que se debe seguir desde que el cliente pide solicitar una adopción hasta que envía el formulario de adopción.



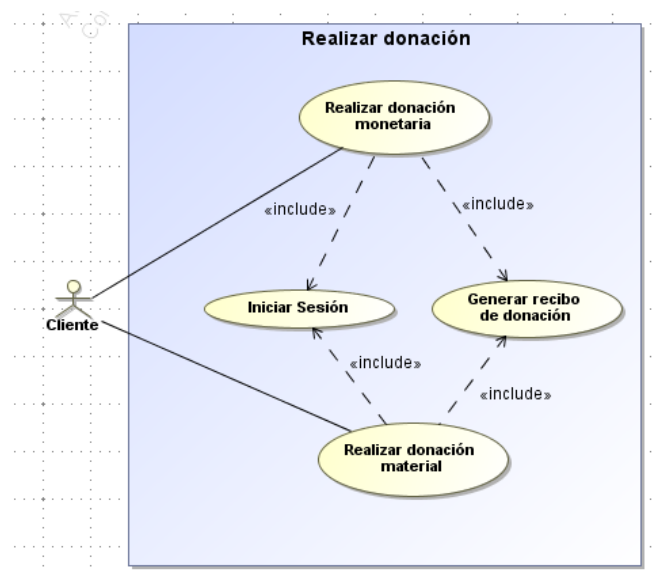
En este diagrama se muestra el estado de los objetos cuando un cliente ha solicitado una adopción y ha seleccionado el animal que quiere adoptar.



## 6. MODELADO PARA EL CASO DE USO 03: REALIZAR DONACIÓN MONETARIA

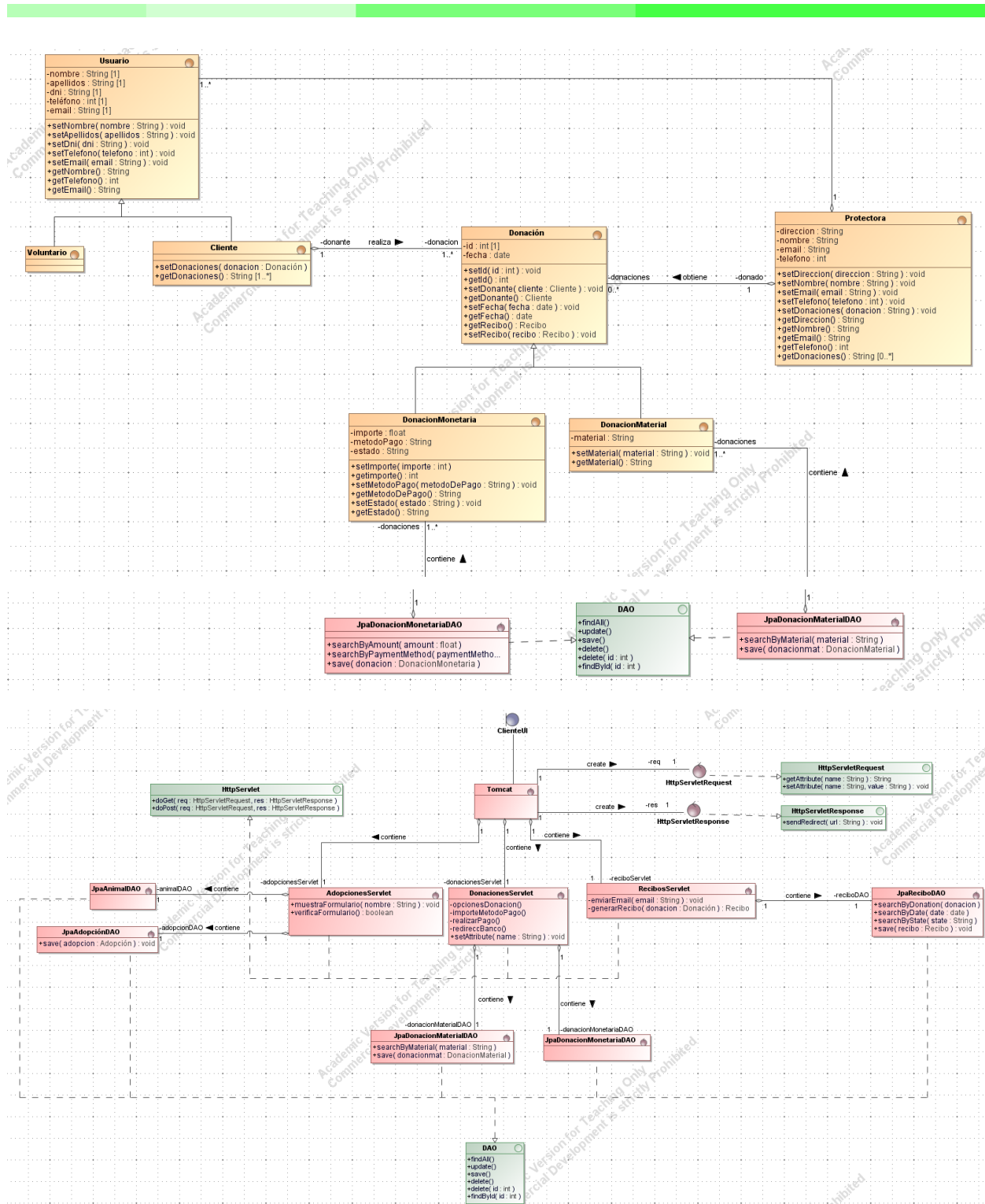
### a. Diagrama de casos de uso

En este diagrama podemos observar el conjunto "Realizar donación" compuesto por varios casos de uso. En esta ocasión vamos a diseñar el modelado para los casos de uso 'Realizar donación monetaria' y 'Generar recibo de donación'.



### b. Diagramas de clases

En este diagrama se representan las clases utilizadas en el Caso de Uso 'Realizar donación monetaria'.

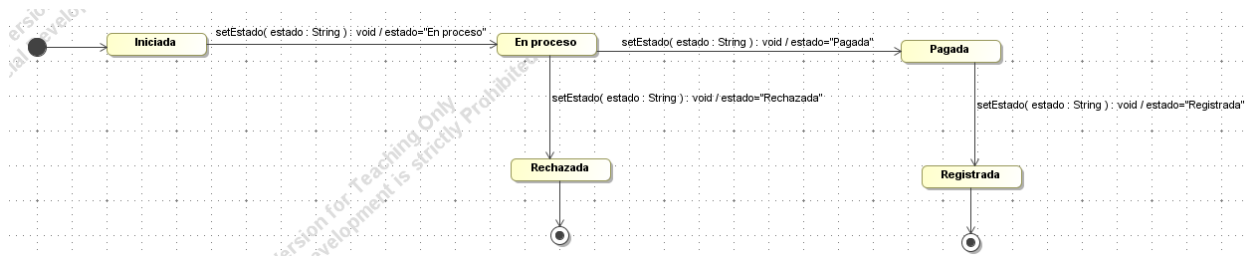


c. Diagrama de actividad

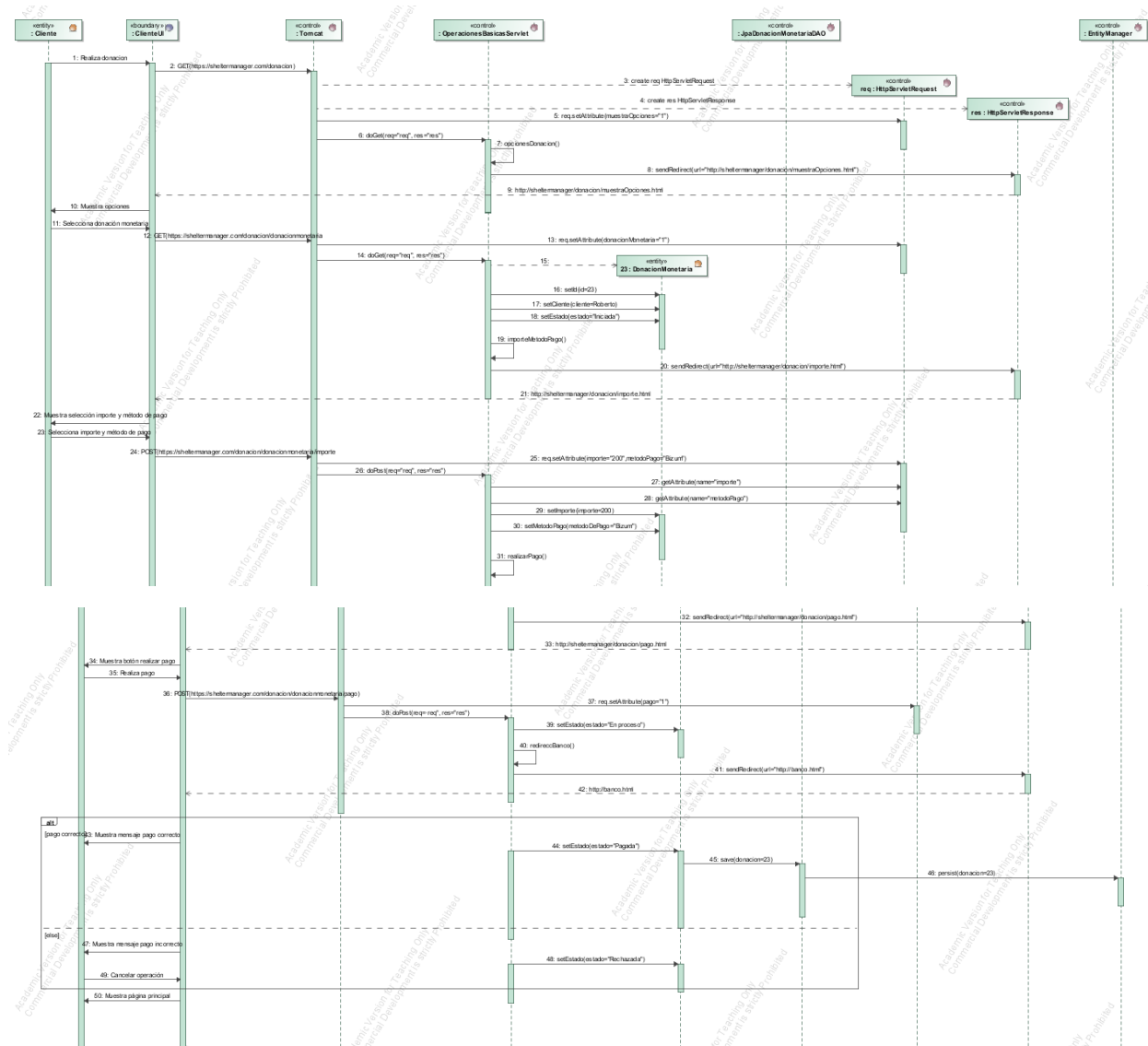


## d. Diagrama de estados

En este diagrama se representan los diferentes estados que puede tener una donación y los métodos que desencadenan esos estados.

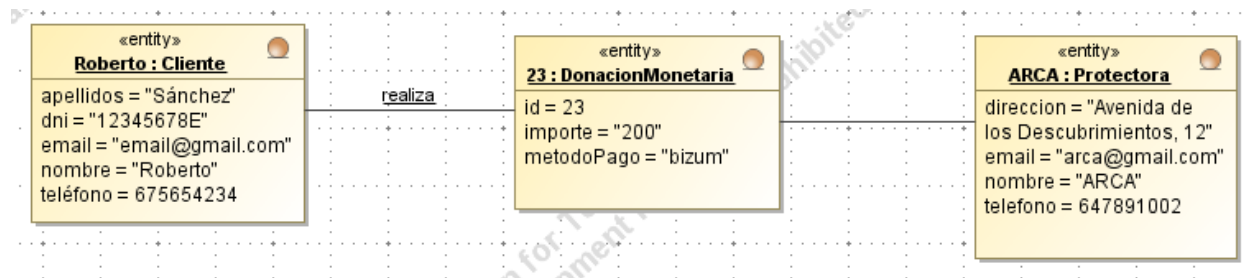


## e. Diagrama de secuencia



## f. Diagrama de objetos

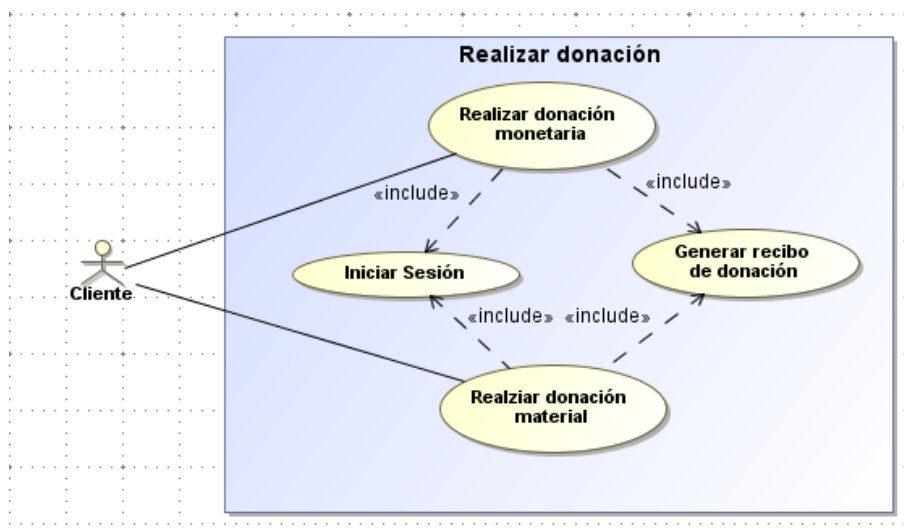
En este diagrama se muestra un ejemplo de la instancia de las clases.



## 7. MODELADO PARA EL CASO DE USO 04

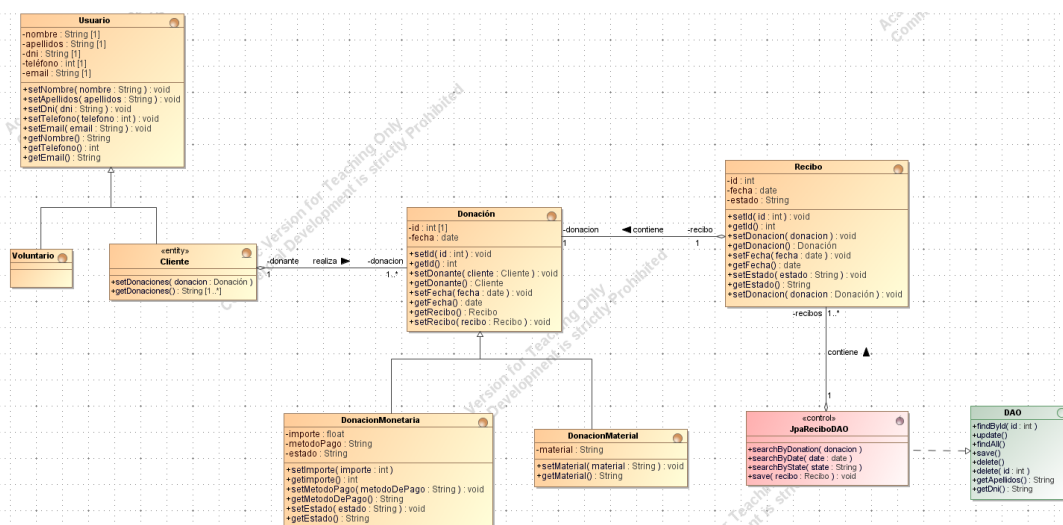
### a. Diagrama de casos de uso

En este diagrama podemos observar el conjunto "Realizar donación" compuesto por varios casos de uso. En esta ocasión vamos a diseñar el modelado para los casos de uso 'Realizar donación monetaria' y 'Generar recibo de donación'.

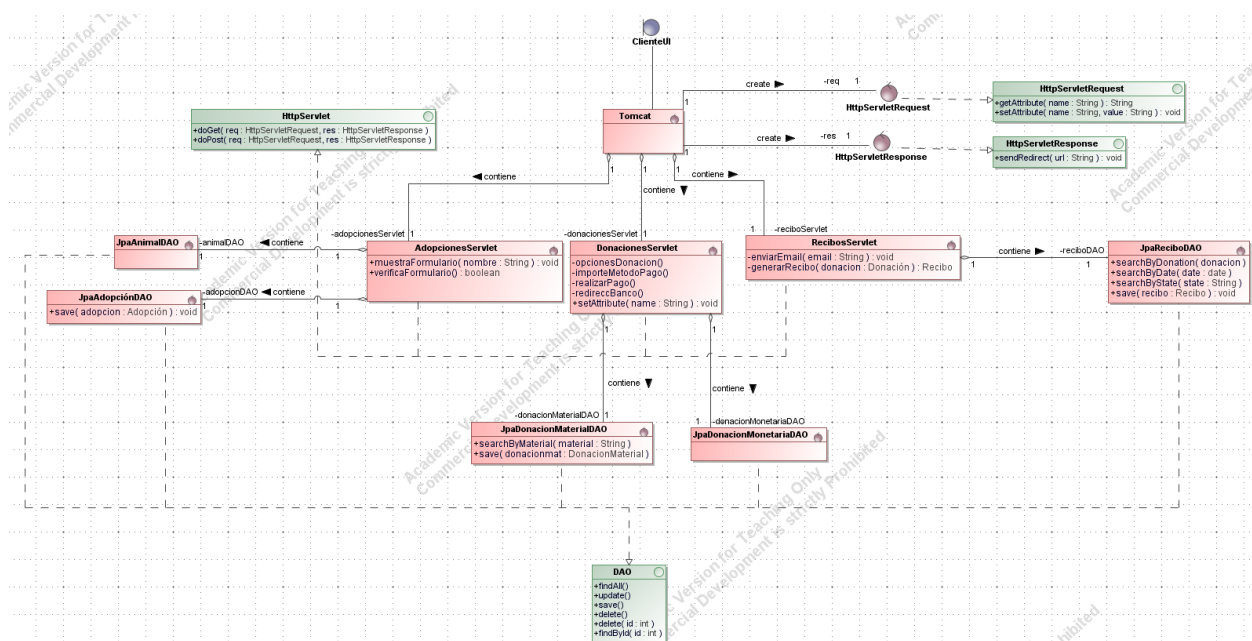


## b. Diagrama de clases

En este diagrama se representan las clases utilizadas en el caso de uso.

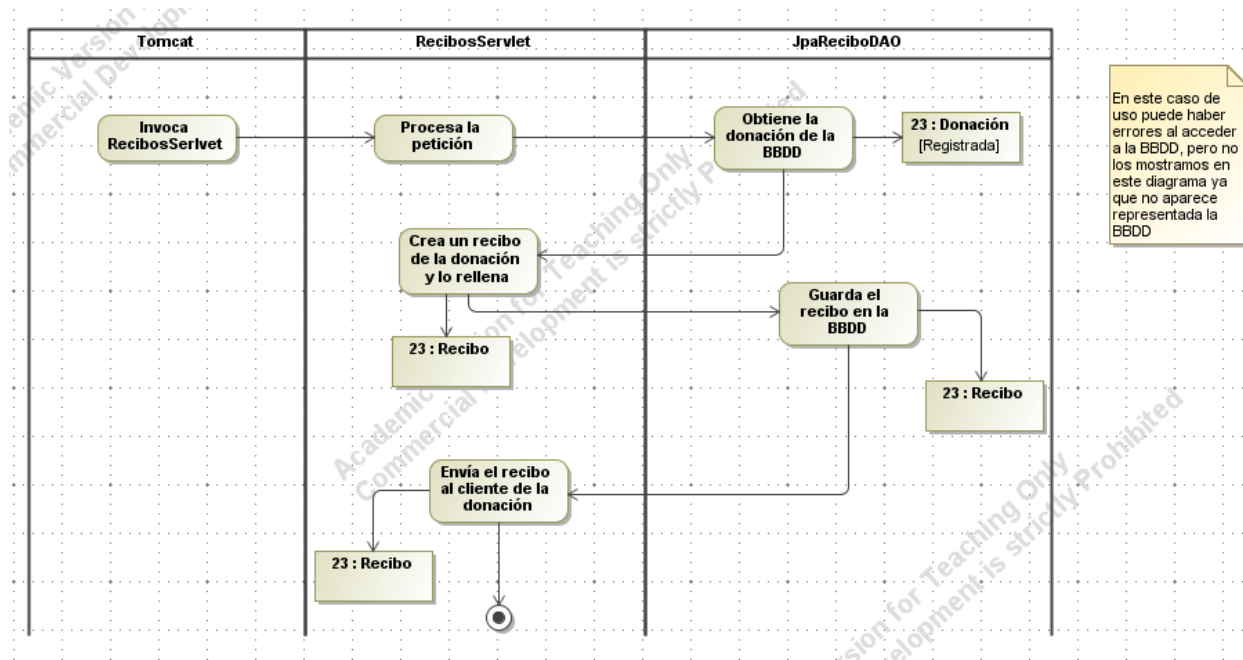






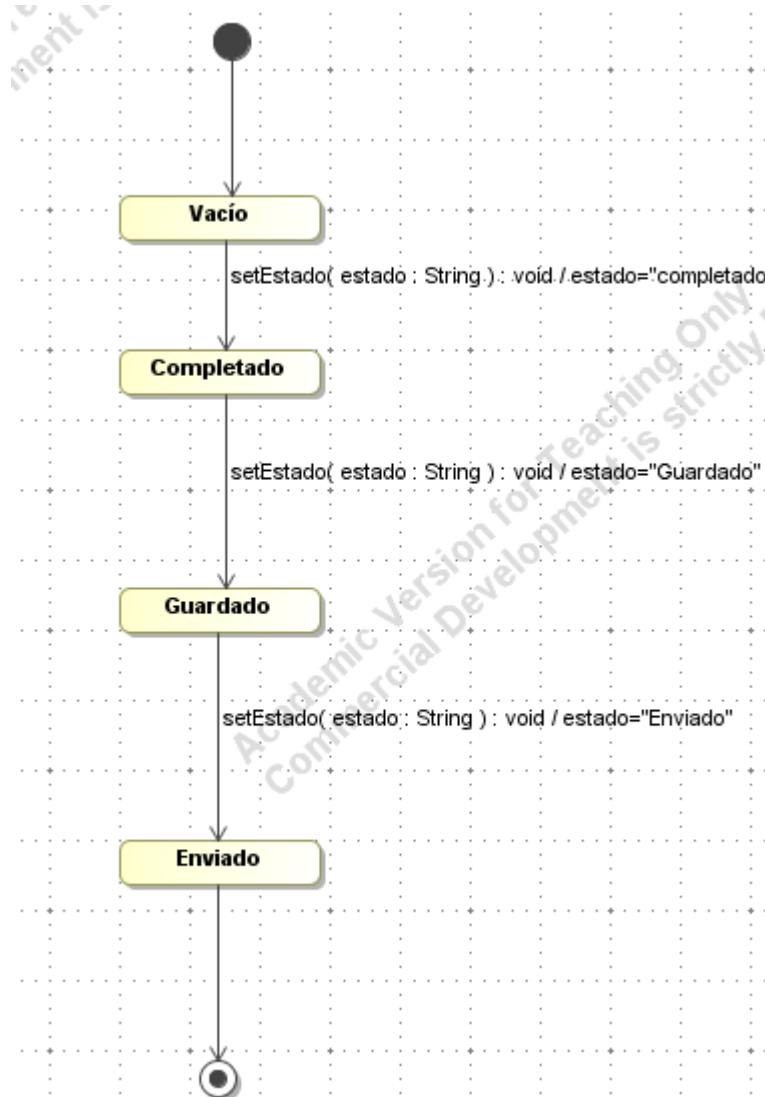
### c. Diagrama de actividad

Aquí podemos observar como la aplicación, sin interactuar con ningún actor, genera un recibo de donación.

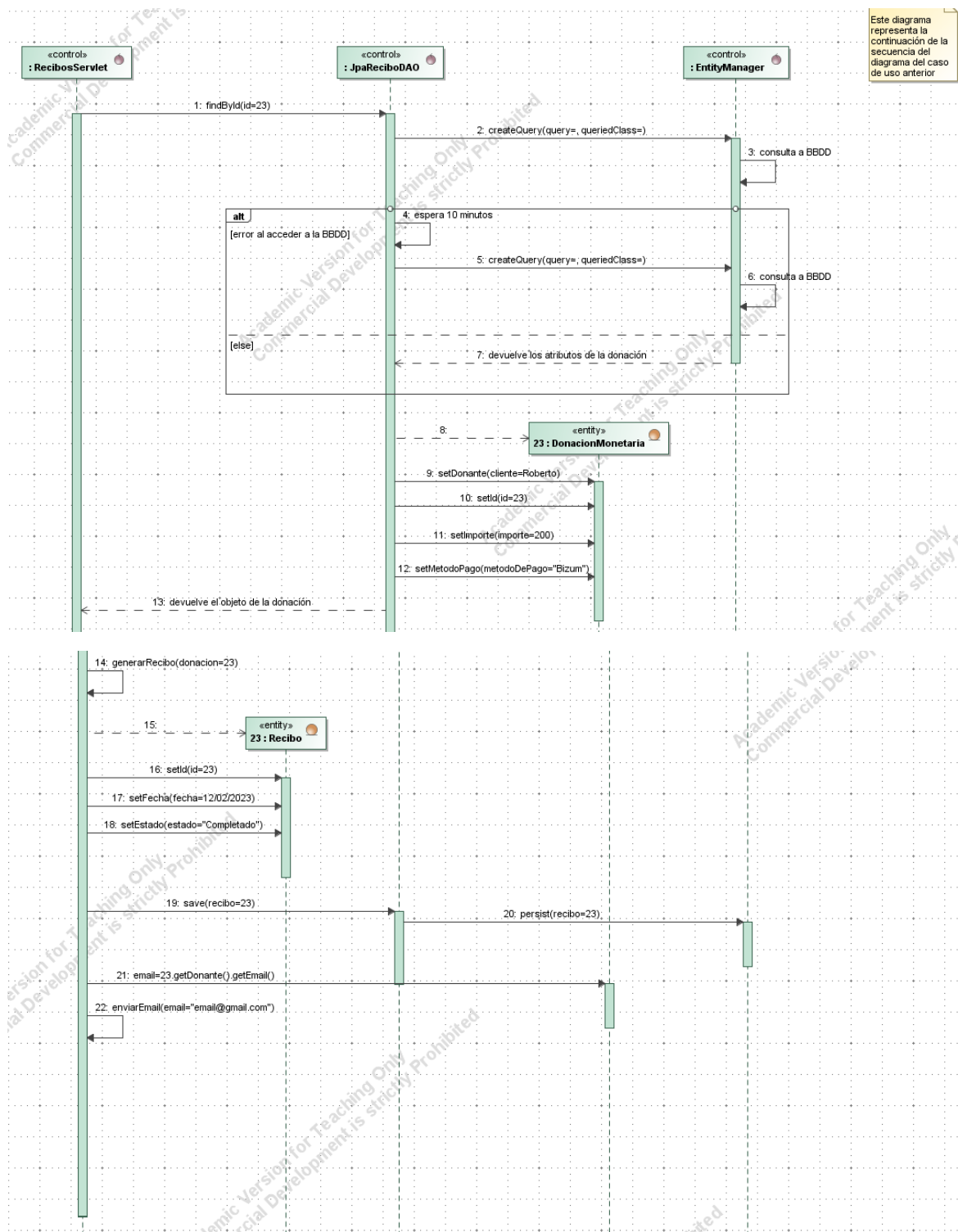


#### d. Diagrama de estados

En este diagrama podemos observar los distintos estados en los que puede estar un objeto de tipo recibo de donación. También observamos las distintas operaciones que desencadenan el cambio de estado.



## e. Diagrama de secuencia



## f. Diagrama de objetos

En este diagrama se muestra un ejemplo de la instancia de las clases.

