



**ESCUELA POLITÉCNICA  
SUPERIOR DE CÓRDOBA**  
Universidad de Córdoba



**INGENIERÍA WEB**  
*Grado en Ingeniería Informática*

**UCOCAMP**

**Autores:** Adrián Leiva Rojano  
Alejandro Luna Puerto  
David Salcedo Sánchez  
Marcos Rivera Gavilán  
Marcos Rodríguez Moreno  
Nicolás José López Delgado

**Profesor:** David Guijo Rubio

**Grupo 1**



UNIVERSIDAD DE CÓRDOBA



# Índice

<b>Introducción</b>	<b>4</b>
<b>Definición del problema</b>	<b>5</b>
<b>Alcance del sistema</b>	<b>6</b>
<b>Objetivos del sistema</b>	<b>6</b>
<b>Recursos</b>	<b>7</b>
<b>Antecedentes</b>	<b>7</b>
Udemy	7
Codely	7
Coursera	8
Canvas	8
Captura de requisitos	8
<b>Definición de requisitos</b>	<b>8</b>
Requisitos Funcionales	8
Requisitos No Funcionales	10
Requisitos De Información	10
Validación de requisitos	11
Casos de uso en UWE	12
<b>Requisitos como actividades</b>	<b>25</b>
<b>SCRUM</b>	<b>53</b>
Backlog	53
Backlog	54
Sprint 1: 24-02-22 - 03/03/22	55
Sprint 2: 03-03-22 - 10/03/22	56
Sprint 3: 10-03-22 - 17/03/22	56

# Introducción



*“La imagen 1 contiene el logo de la aplicación”*

Esta es la memoria de la práctica 1 y 2 del grupo 1. En ella vamos a desarrollar la especificación del sistema que vamos a desarrollar en las prácticas de la asignatura.

En nuestro caso, UCOCAMP, es una plataforma de enseñanza online cuyo objetivo principal es ofrecer a los alumnos la posibilidad de obtener una formación complementaria a la que reciben en la universidad y que les permita validar estas aptitudes a través de una serie de certificados oficiales emitidos y respaldados por la universidad.

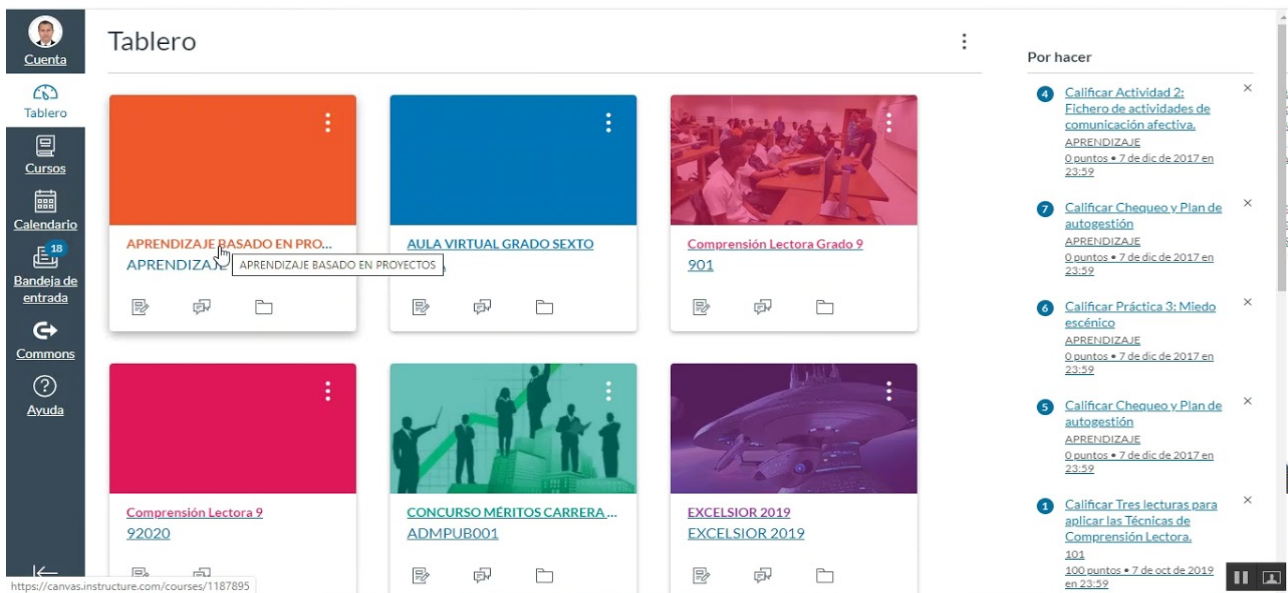
Esta práctica la hemos realizado a lo largo de 3 semanas. Esto aparece reflejado en la sección de SCRUM de este mismo documento. En ella explicamos que hemos hecho y cuando lo hemos hecho y como nos hemos organizado.

Para organizarnos hemos repartido las tareas que iba a tener cada sprint entre los distintos miembros del equipo de forma equitativa.

Como los sprints eran de una semana, hemos intentado que el valor de la suma de los puntos de función de las tareas que se introdujesen en ese sprint no fuese superior a 30 ya que consideramos que completar 5 puntos de función al día era algo asequible. La suma total de los puntos de función de los tres sprints es de 90 puntos de función.

El resultado de estas tres semanas de trabajo, aparece reflejado en este documento y es fruto del esfuerzo de los distintos miembros del equipo que siguiendo los roles asignados han realizado las distintas tareas propuestas.

# Definición del problema



*“La imagen 2 contiene un boceto de la aplicación”*

El objetivo de este proyecto, es crear una plataforma web que permita ofertar cursos online al alumnado de la UCO. Ya que sería genial disponer de una plataforma que nos permitiera validar de forma oficial nuestras diferentes aptitudes.

En este proyecto nos vamos a centrar en los cursos online asíncronos de programación, que se crearán tanto para el alumnado como para el profesorado.

Principalmente, tenemos un único usuario, que puede tener dos roles: creador y alumno. Los cursos pueden ser creados por cualquiera haciendo uso de la plataforma.

Además, como se ha mencionado, los creadores también pueden participar como alumnos en aquellos cursos que están ya creados.

A un curso pueden suscribirse muchos alumnos y puede tener más de un creador. El creador (o los creadores) son los encargados de toda la configuración del curso, así como de su contenido.

Simulando el comportamiento que tiene Udemy, la vista del curso por parte del creador debe ser diferente a la vista por parte del alumno, ya que el creador puede modificar y editar los contenidos, mientras que el alumno solamente debe poder descargar el contenido y hacer entregas de tareas.

En cuanto a las calificaciones, cada alumno debe ver las suyas únicamente. Además, hay que implementar una herramienta, que genere automáticamente los títulos, una vez que el curso haya sido completado por el alumno. Además los alumnos deben poder comentar en las secciones de los cursos y valorar el curso en general.

Por último, el creador debe tener acceso a estadísticas sobre el curso, como por ejemplo puede ser el número de alumnos o la proporción entre matriculados y completados.

# Alcance del sistema

El alcance de nuestro sistema será en un principio satisfacer y servir a los alumnos de la Universidad de Córdoba, aunque por como ha sido planteado, puede ser usado por cualquier persona externa a ella que quiera cursar alguno de los cursos siempre que se identifique y se dé de alta.

La idea es que esta plataforma se implante en la universidad para que los alumnos puedan complementar su formación universitaria.

El sistema debe de cumplir con todos los requisitos especificados en las siguientes páginas para que podamos considerar que su alcance es completo.

# Objetivos del sistema

Los objetivos que hemos planteado para este sistema son los siguiente:

- Crear una plataforma de cursos online.
- Proporcionar herramientas para la creación de cursos.
- Ofrecer una experiencia de usuario fluida
- Mejorar y facilitar el acceso a la formación
- Ofrecer un espacio de trabajo para alumnos y profesores

# Recursos

Los recursos que vamos a utilizar para desarrollar el sistema son los siguientes:

Dueño	Modelo	Procesador	Ram
Adrián	Lenovo-IdeaPad 5 Pro	Ryzen 7 5800u	16GB
Marcos Rivera	Asus-TP300LA	Intel i5-4210U	12GB
Marcos Rodríguez	Apple Macbook M1	Apple M1	8GB
Alejandro Luna	Lenovo IdeaPad 3	Intel i5-1035G1	8GB
Nicolás	Lenovo IdeaPad	Intel i7-8550U	8GB
David	Lenovo T440	Intel i5-4300U	8GB

*“La tabla 1 contiene los recursos utilizados para desarrollar el sistema”*

# Antecedentes

Algunas aplicaciones similares cuyos objetivos se asemejan a los nuestros:

## Udemy



*“La imagen 3 contiene el logo de Udemy”*

Udemy es una plataforma que ofrece cursos para adultos profesionales, principalmente sobre programación, desarrollo web y otros campos de la informática.

## Codely



*“La imagen 4 contiene el logo de CodelyTV”*

CodelyTV es una plataforma de cursos online que funciona a través de un modelo de suscripción enfocada sobre todo en el ámbito del desarrollo y la arquitectura software.

## Coursera



*“La imagen 5 contiene el logo de coursera”*

Coursera es una plataforma mucho más grande que las previamente mencionadas. Cuenta con cursos mucho más generales y pertenecientes a prácticamente todos los ámbitos del conocimiento

## Canvas



*“La imagen 6 contiene el logo de canvas”*

Canvas es la plataforma que la Universidad de Córdoba utiliza actualmente para impartir sus másteres y cursos online. La principal diferencia es que en este caso se vende el software de la aplicación para que la aloje ahí sus cursos propios pues ellos no generan contenidos de ningún tipo



## Requisitos Funcionales

ID	Descripción del requisito	Prioridad
RF13	El sistema debe permitir editar una lección dentro de un bloque	M
RF14	El sistema debe permitir borrar una lección dentro de un bloque	M
RF15	El sistema debe permitir crear un nuevo ejercicio dentro de un bloque	M
RF16	El sistema debe permitir editar un ejercicio dentro de un bloque	M
RF17	El sistema debe permitir borrar un ejercicio dentro de un bloque	M
RF18	El sistema debe permitir apuntarse a un curso	M
RF19	El sistema debe permitir desapuntarse de un curso	S
RF20	El sistema debe mostrar un listado con los cursos disponibles	M
RF21	El sistema debe mostrar un listado con los bloques pendientes y completados dentro de un curso	C
RF22	El sistema debe mostrar un listado con las lecciones de un bloque perteneciente a un curso	S
RF23	El sistema debe permitir entregar un ejercicio	M
RF24	El sistema debe permitir añadir eventos al calendario de un curso	C
RF25	El sistema debe permitir modificar eventos del calendario de un curso	C
RF26	El sistema debe permitir borrar eventos del calendario de un curso	C
RF27	El sistema debe permitir a los creadores de los cursos enviar comunicados a los alumnos	S

*“La tabla 3 contiene los requisitos funcionales del sistema”*



## Requisitos No Funcionales

ID	Descripción del requisito	Prioridad
RNF1	El sistema debe poder funcionar en cualquier navegador	M
RNF2	La interfaz de la aplicación debe ser responsive y adaptarse a cualquier dispositivo	M
RNF3	El sistema debe poder cargar los videos en menos de 5 segundos cuando haya 1000 usuarios conectados	M
RNF4	El sistema debe cumplir los estándares de accesibilidad para usuarios con necesidades especiales	M
RNF5	La interfaz debe seguir el código de colores y la tipografía de la universidad de córdoba	M

*“La tabla 4 contiene los requisitos no funcionales del sistema”*

## Requisitos De Información

ID	Descripción del requisito	Prioridad
RI1	El sistema debe almacenar de los alumnos: Nombre, Apellidos, Edad, Email, Cursos matriculados, Fecha en la que se matricularon, Cursos completados, Bloques completados, Secciones que se completaron, Fecha en la que se completaron cada uno, Calificaciones de los cursos	M
RI2	El sistema debe almacenar de los cursos: Nombre, Duración, Bloques que lo conforman, Alumnos matriculados, Alumnos que lo han superado, Fecha en la que se crearon, Creador	M
RI3	El sistema debe almacenar de los bloques de un curso: Nombre, Duración, Lecciones/Ejercicios que lo conforman, Alumnos que lo han superado	M
RI4	El sistema debe almacenar sobre las lecciones y ejercicios: Nombre, Duración, Contenido, Alumnos que lo han superado.	M
RI5	El sistema debe almacenar almacenar de los creadores de cursos: Nombre, Apellidos, Edad, Email, Cursos creados	M

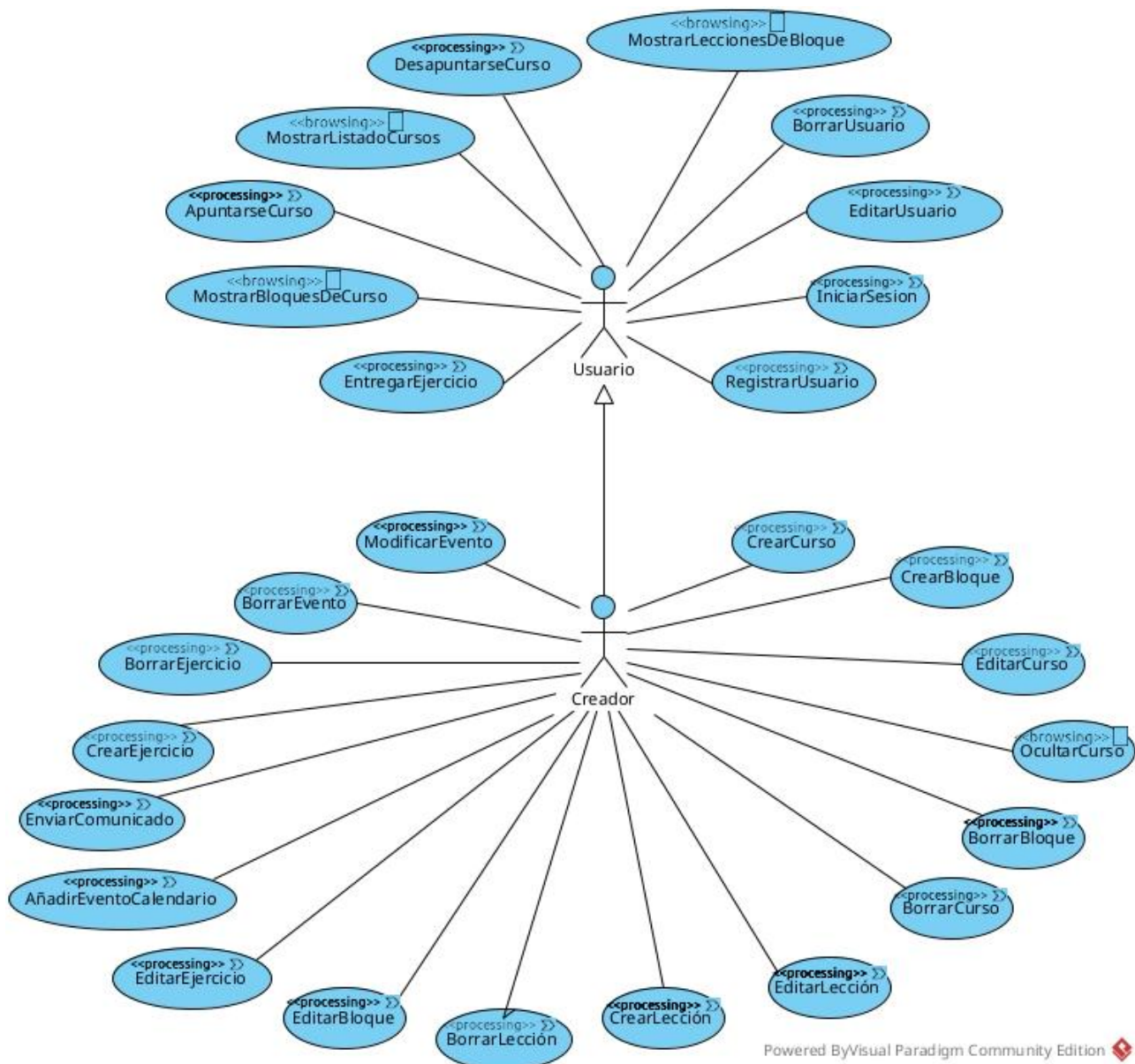
*“La tabla 5 contiene los requisitos de información del sistema”*

# Validación de requisitos

Para realizar la validación de requisitos con el cliente, realizamos una última entrevista y le mostramos el siguiente diagrama de casos de uso que representaba las funcionalidades del sistema

## Casos de uso en UWE

En este apartado modelamos los casos de uso del sistema haciendo uso de UWE



“La imagen 7 contiene el diagrama de casos de uso”

## Caso de Uso: IniciarSesion

Nombre	IniciarSesion
Requisito que referencia	RF1
Descripción	Un usuario anónimo podrá iniciar sesión en el sistema.
Actor	Usuario
Precondición	No tener una sesión iniciada.
Postcondición	El usuario debe haber iniciado la sesión.

*“La tabla 6 contiene el caso de uso IniciarSesion”*

## Caso de Uso: RegistrarUsuario

Nombre	RegistrarUsuario
Requisito que referencia	RF2
Descripción	Un usuario no registrado podrá registrarse en el sistema.
Actor	Usuario
Precondición	No tener una sesión iniciada.
Postcondición	El usuario debe haber sido registrado.

*“La tabla 7 contiene el caso de uso RegistrarUsuario”*

## Caso de Uso: BorrarUsuario

Nombre	BorrarUsuario
Requisito que referencia	RF3
Descripción	Un usuario puede borrar su cuenta.
Actor	Usuario
Precondición	El usuario debe estar registrado en el sistema.
Postcondición	El usuario debe de ser borrado.

*“La tabla 8 contiene el caso de uso BorrarUsuario”*

## Caso de Uso: EditarUsuario

Nombre	EditarUsuario
Requisito que referencia	RF4
Descripción	Un usuario puede editar su cuenta.
Actor	Usuario
Precondición	El usuario debe estar registrado en el sistema.
Postcondición	Ninguno

*“La tabla 9 contiene el caso de uso EditarUsuario”*

## Caso de Uso: CrearCurso

Nombre	CrearCurso
Requisito que referencia	RF5
Descripción	El creador crea un curso.
Actor	Creador
Precondición	Ninguno
Postcondición	El creador tendrá un nuevo curso a su nombre.

*“La tabla 10 contiene el caso de uso CrearCurso”*

## Caso de Uso: OcultarCurso

Nombre	OcultarCurso
Requisito que referencia	RF6
Descripción	El creador elige la visibilidad de los cursos (visible o no visible).
Actor	Creador
Precondición	Debe estar el curso creado.
Postcondición	Se debe producir un cambio en la visibilidad del curso

*“La tabla 11 contiene el caso de uso OcultarCurso”*

## Caso de Uso: EditarCurso

Nombre	EditarCurso
Requisito que referencia	RF7
Descripción	El creador edita el contenido de un curso (nombre, información...)
Actor	Creador
Precondición	Debe haber al menos un curso creado que editar.
Postcondición	Ninguna

*“La tabla 12 contiene el caso de uso EditarCurso”*

## Caso de Uso: BorrarCurso

Nombre	BorrarCurso
Requisito que referencia	RF8
Descripción	El creador borrará uno de sus propios cursos.
Actor	Creador
Precondición	El creador debe tener al menos un curso creado, no deben haber alumnos realizando el curso
Postcondición	El creador tendrá un curso menos

*“La tabla 13 contiene el caso de uso BorrarCurso”*

## Caso de Uso: CrearBloque

Nombre	CrearBloque
Requisito que referencia	RF9
Descripción	El creador creará un nuevo bloque dentro de un curso ya existente
Actor	Creador
Precondición	El creador debe tener al menos un curso creado
Postcondición	El curso tendrá un bloque más que antes

*“La tabla 14 contiene el caso de uso CrearBloque”*

## Caso de Uso: EditarBloque

Nombre	EditarBloque
Requisito que referencia	RF10
Descripción	El creador editará el bloque de uno de sus cursos
Actor	Creador
Precondición	El creador debe tener al menos un curso creado con un bloque
Postcondición	Ninguna

*“La tabla 15 contiene el caso de uso EditarBloque”*

## Caso de Uso: BorrarBloque

Nombre	BorrarBloque
Requisito que referencia	RF11
Descripción	El creador borrará uno de los bloques de un curso específico.
Actor	Creador
Precondición	El creador debe tener al menos un curso creado y con un bloque
Postcondición	El curso tendrá un bloque menos.

*“La tabla 16 contiene el caso de uso BorrarBloque”*

## Caso de Uso: CrearLección

Nombre	CrearLección
Requisito que referencia	RF-12
Descripción	El creador crea una nueva lección dentro de un bloque de un curso.
Actor	Creador
Precondición	-El curso debe existir. -La lección a crear no debe existir previamente en ese curso.
Postcondición	Se añade la nueva lección al correspondiente bloque del curso.

*“La tabla 17 contiene el caso de uso CrearLección”*

## Caso de Uso: EditarLección

Nombre	EditarLección
Requisito que referencia	RF-13
Descripción	El creador edita una lección dentro de un bloque de un curso.
Actor	Creador
Precondición	-El curso debe existir. -La lección a editar debe existir.
Postcondición	Se aplican los cambios realizados sobre la lección editada.

*“La tabla 18 contiene el caso de uso EditarLección”*

## Caso de Uso: BorrarLección

Nombre	BorrarLección
Requisito que referencia	RF-14
Descripción	El creador borra una lección dentro de un bloque de un curso.
Actor	Creador
Precondición	-El curso debe existir. -La lección a borrar debe existir.
Postcondición	Se borra la lección del curso.

*“La tabla 19 contiene el caso de uso BorrarLección”*

## Caso de Uso: CrearEjercicio

<b>Nombre</b>	CrearEjercicio
<b>Requisito que referencia</b>	RF-15
<b>Descripción</b>	El creador crea un ejercicio dentro de un bloque de un curso.
<b>Actor</b>	Creador
<b>Precondición</b>	-El curso debe existir. -El ejercicio a crear no debe existir ya en ese curso.
<b>Postcondición</b>	Se añade el ejercicio creado al bloque del curso.

*“La tabla 20 contiene el caso de uso CrearEjercicio”*

## Caso de Uso: EditarEjercicio

<b>Nombre</b>	EditarEjercicio
<b>Requisito que referencia</b>	RF-16
<b>Descripción</b>	El creador edita un ejercicio dentro de un bloque de un curso.
<b>Actor</b>	Creador
<b>Precondición</b>	-El curso debe existir. -El ejercicio a editar debe existir.
<b>Postcondición</b>	Se aplican los cambios realizados sobre el ejercicio.

*“La tabla 21 contiene el caso de uso EditarEjercicio”*



## Caso de Uso: BorrarEjercicio

<b>Nombre</b>	BorrarEjercicio
<b>Requisito que referencia</b>	RF-17
<b>Descripción</b>	El creador borra un ejercicio dentro de un bloque de un curso.
<b>Actor</b>	Creador
<b>Precondición</b>	-El curso debe existir. -El ejercicio a borrar debe existir.
<b>Postcondición</b>	Se borra el ejercicio del curso.

*“La tabla 22 contiene el caso de uso BorrarEjercicio”*

## Caso de Uso: ApuntarseCurso

<b>Nombre</b>	ApuntarseCurso
<b>Requisito que referencia</b>	RF-18
<b>Descripción</b>	Un usuario registrado se apunta a un curso abierto.
<b>Actor</b>	Usuario
<b>Precondición</b>	1. El usuario debe tener una sesión iniciada en el sistema. 2. El curso debe de estar abierto. 3. El usuario debe de haber entrado a la vista previa del curso.
<b>Postcondición</b>	1. El usuario debe de tener acceso al contenido del curso. 2. Debe de quedar registrado en su usuario que está apuntado a dicho curso.

*“La tabla 23 contiene el caso de uso ApuntarseCurso”*

## Caso de Uso: DesapuntarseCurso

<b>Nombre</b>	DesapuntarseCurso
<b>Requisito que referencia</b>	RF-19
<b>Descripción</b>	Un usuario se desapunta a un curso en el que estaba apuntado.
<b>Actor</b>	Usuario
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe tener una sesión iniciada en el sistema.</li> <li>2. El usuario debe pertenecer al curso.</li> </ol>
<b>Postcondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe de dejar de tener acceso al contenido del curso.</li> <li>2. Debe de quedar registrado en su usuario que no está apuntado a dicho curso.</li> </ol>

*“La tabla 24 contiene el caso de uso DesapuntarseCurso”*

## Caso de Uso: MostrarListadoCursos

<b>Nombre</b>	MostrarListadoCursos
<b>Requisito que referencia</b>	RF-20
<b>Descripción</b>	Un usuario consulta el listado de cursos disponibles
<b>Actor</b>	Usuario
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe tener una sesión iniciada en el sistema.</li> <li>2. Deben de existir cursos abiertos en el sistema.</li> </ol>
<b>Postcondición</b>	

*“La tabla 25 contiene el caso de uso MostrarListadoCursos”*

## Caso de Uso: MostrarBloquesDeCurso

<b>Nombre</b>	MostrarBloquesDeCurso
<b>Requisito que referencia</b>	RF-21
<b>Descripción</b>	Un usuario consulta el listado de bloques pendientes y completados de un curso al que está apuntado
<b>Actor</b>	Usuario
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe tener una sesión iniciada en el sistema.</li> <li>2. El usuario debe estar apuntado a dicho curso.</li> </ol>
<b>Postcondición</b>	

*“La tabla 26 contiene el caso de uso MostrarBloquesDeCurso”*

## Caso de Uso: MostrarLeccionesDeBloque

<b>Nombre</b>	MostrarLeccionesDeBloque
<b>Requisito que referencia</b>	RF-22
<b>Descripción</b>	Un usuario consulta el listado de lecciones pertenecientes a un bloque
<b>Actor</b>	Usuario
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe tener una sesión iniciada en el sistema.</li> <li>2. El usuario debe estar apuntado al curso que contiene dicho bloque.</li> </ol>
<b>Postcondición</b>	

*“La tabla 27 contiene el caso de uso MostrarLeccionesDeBloque”*

## Caso de Uso: EntregarEjercicio

<b>Nombre</b>	EntregarEjercicio
<b>Requisito que referencia</b>	RF-23
<b>Descripción</b>	Un usuario entrega un ejercicio perteneciente a un bloque o lección de un curso al que está apuntado
<b>Actor</b>	Usuario, Sistema
<b>Precondición</b>	<ol style="list-style-type: none"> <li>1. El usuario debe tener una sesión iniciada en el sistema.</li> <li>2. El usuario debe estar apuntado al curso que contiene la lección o bloque al que pertenece el ejercicio.</li> </ol>
<b>Postcondición</b>	<ol style="list-style-type: none"> <li>1. El sistema verificará la validez de dicho ejercicio y notificará al usuario.</li> </ol>

*“La tabla 28 contiene el caso de uso EntregarEjercicio”*

## Caso de Uso: AñadirEventoCalendario

<b>Nombre</b>	AñadirEventoCalendario
<b>Requisito que referencia</b>	RF-24
<b>Descripción</b>	El creador añade un evento al calendario del curso
<b>Actor</b>	Creador
<b>Precondición</b>	El curso debe existir
<b>Postcondición</b>	Se añade el evento al calendario

*“La tabla 29 contiene el caso de uso AñadirEventoCalendario”*

## Caso de Uso: ModificarEvento

<b>Nombre</b>	ModificarEvento
<b>Requisito que referencia</b>	RF-25
<b>Descripción</b>	El creador del curso modifica un evento ya creado
<b>Actor</b>	Creador
<b>Precondición</b>	El curso debe existir El evento debe existir
<b>Postcondición</b>	Se modifica el evento

*“La tabla 30 contiene el caso de uso ModificarEvento”*

## Caso de Uso: BorrarEvento

<b>Nombre</b>	BorrarEvento
<b>Requisito que referencia</b>	RF-26
<b>Descripción</b>	El creador del curso borra un evento ya creado
<b>Actor</b>	Creador
<b>Precondición</b>	El curso debe existir El evento debe existir
<b>Postcondición</b>	Se borra el evento

*“La tabla 31 contiene el caso de uso BorrarEvento”*

## Caso de Uso: EnviarComunicado

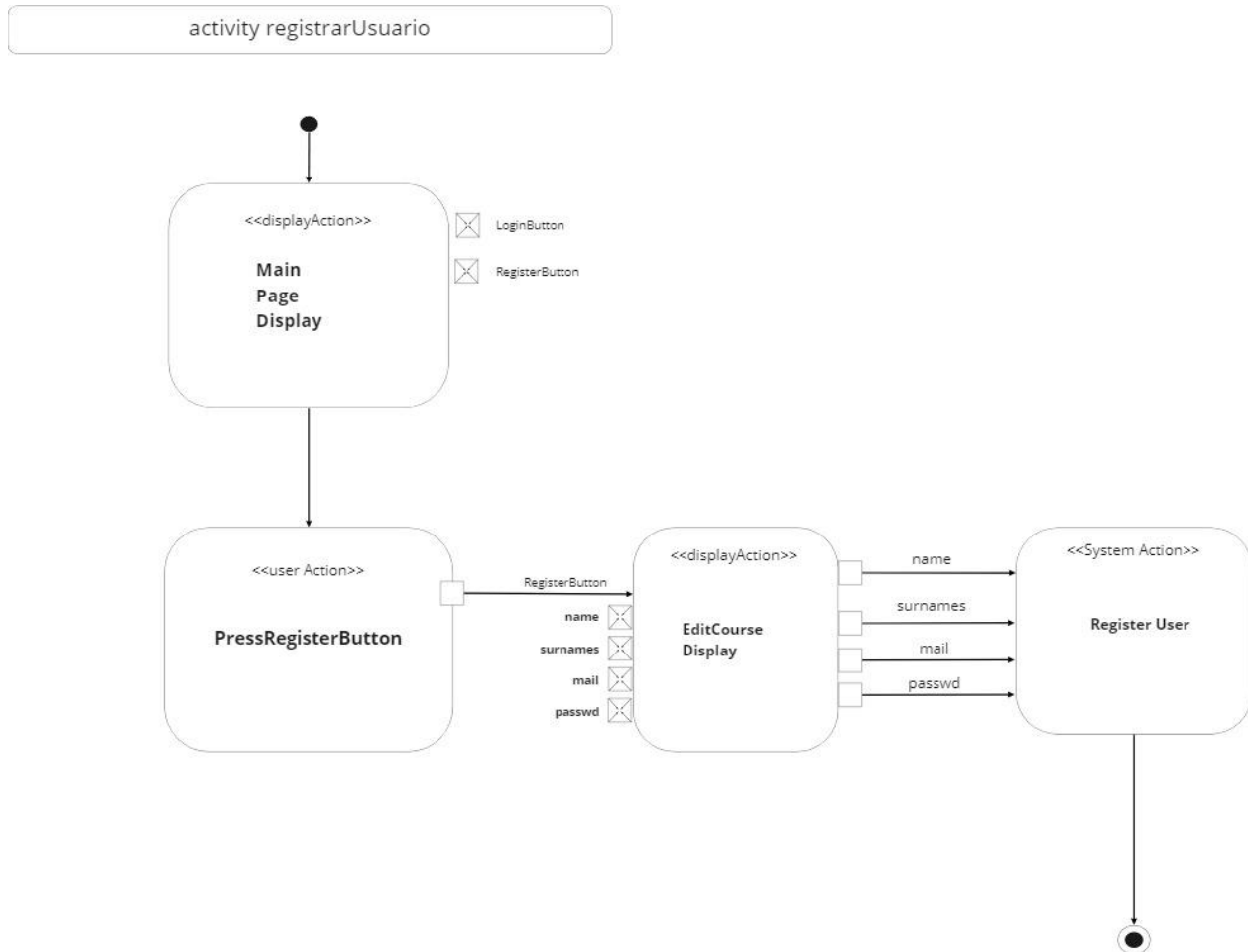
<b>Nombre</b>	EnviarComunicado
<b>Requisito que referencia</b>	RF-27
<b>Descripción</b>	El creador del curso envia un comunicado a sus alumnos
<b>Actor</b>	Creador
<b>Precondición</b>	El curso debe existir Los alumnos deben estar matriculados en el curso
<b>Postcondición</b>	Se envia el comunicado

*“La tabla 32 contiene el caso de uso EnviarComunicado”*

# Requisitos como actividades

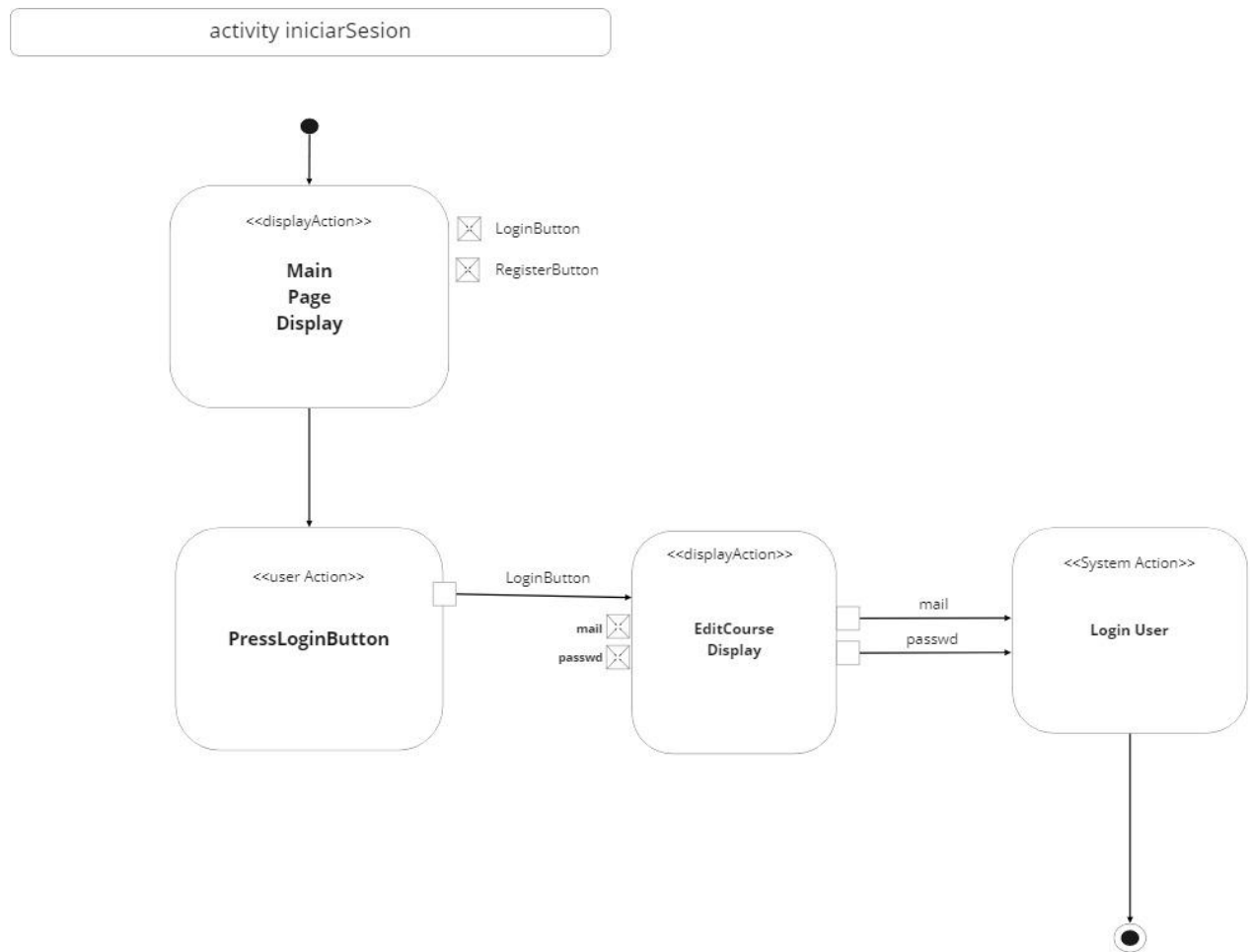
En este apartado realizamos los diagramas de actividades haciendo uso de UWE

## Caso de Uso: RegistrarUsuario



miro

## Caso de Uso: IniciarSesión

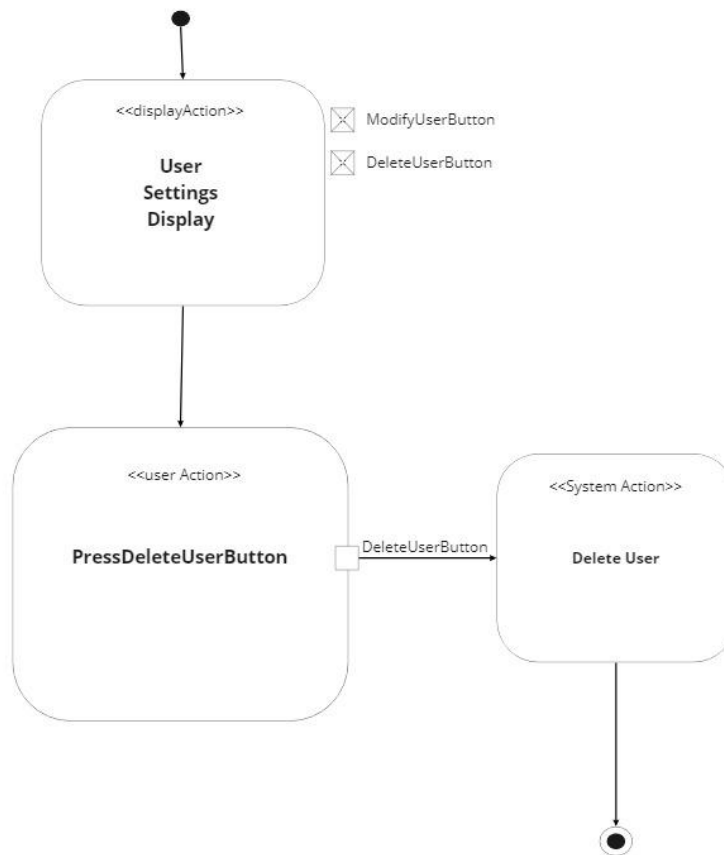


miro



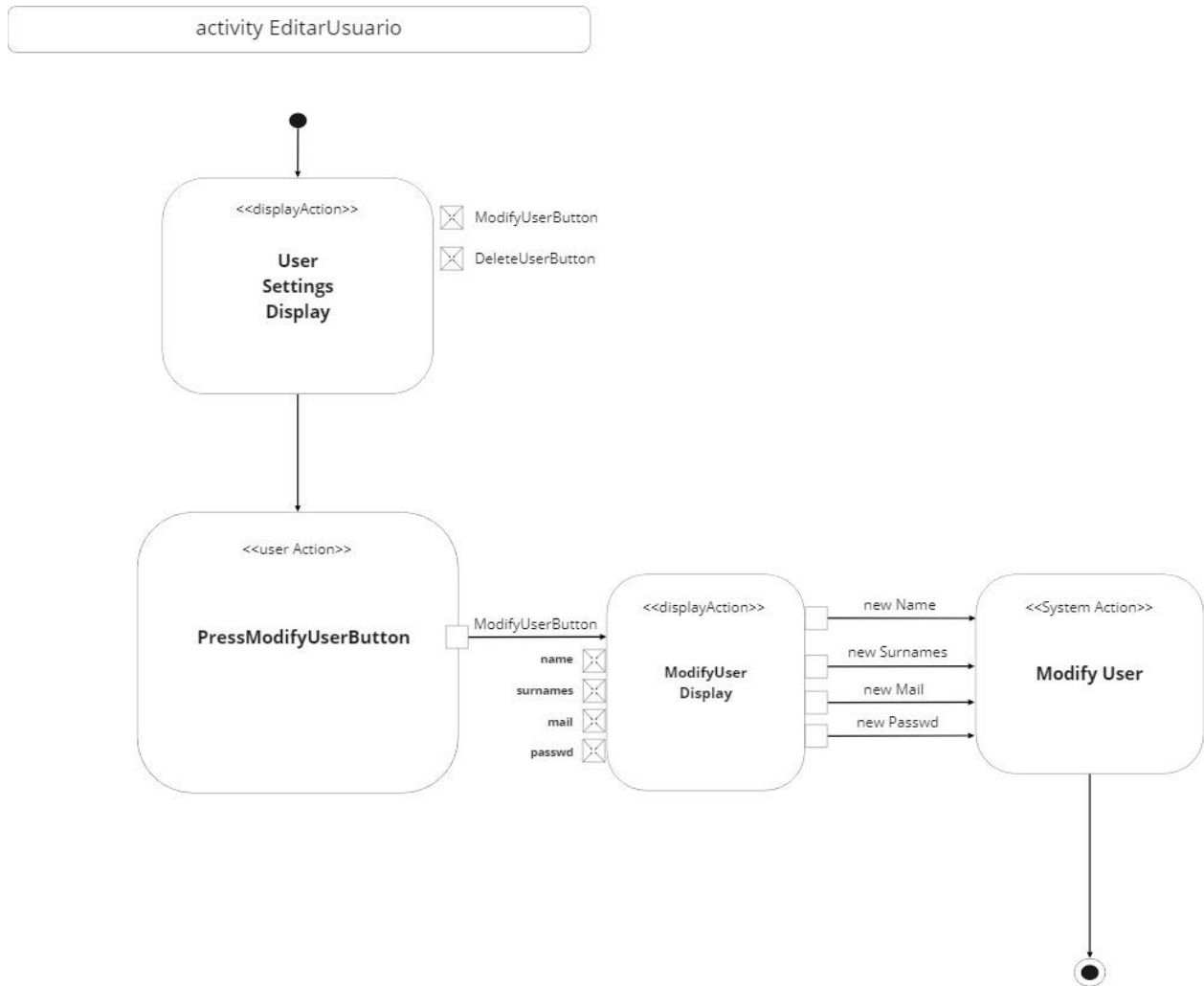
## Caso de Uso: BorrarUsuario

activity BorrarUsuario



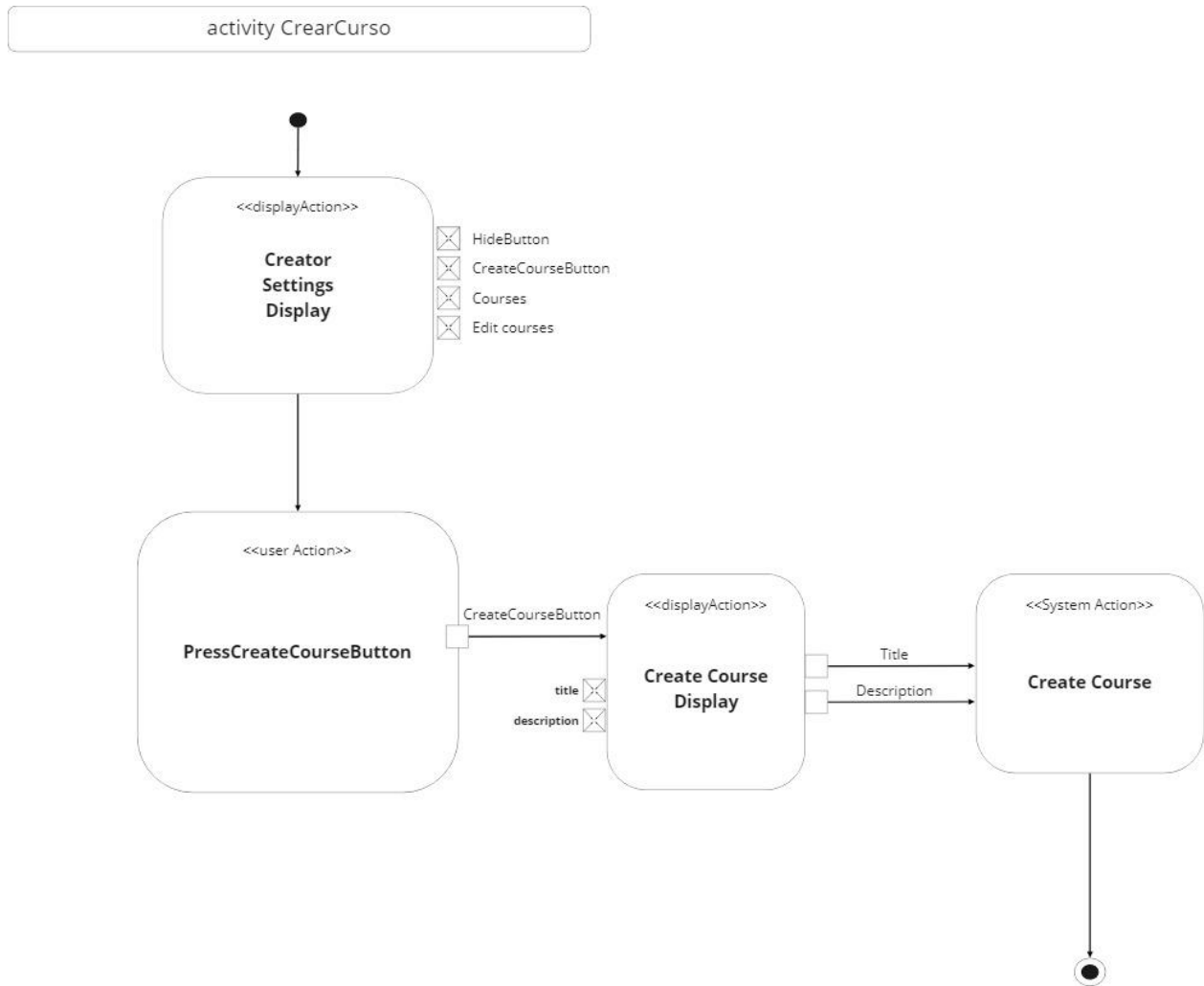
miro

## Caso de Uso: EditarUsuario



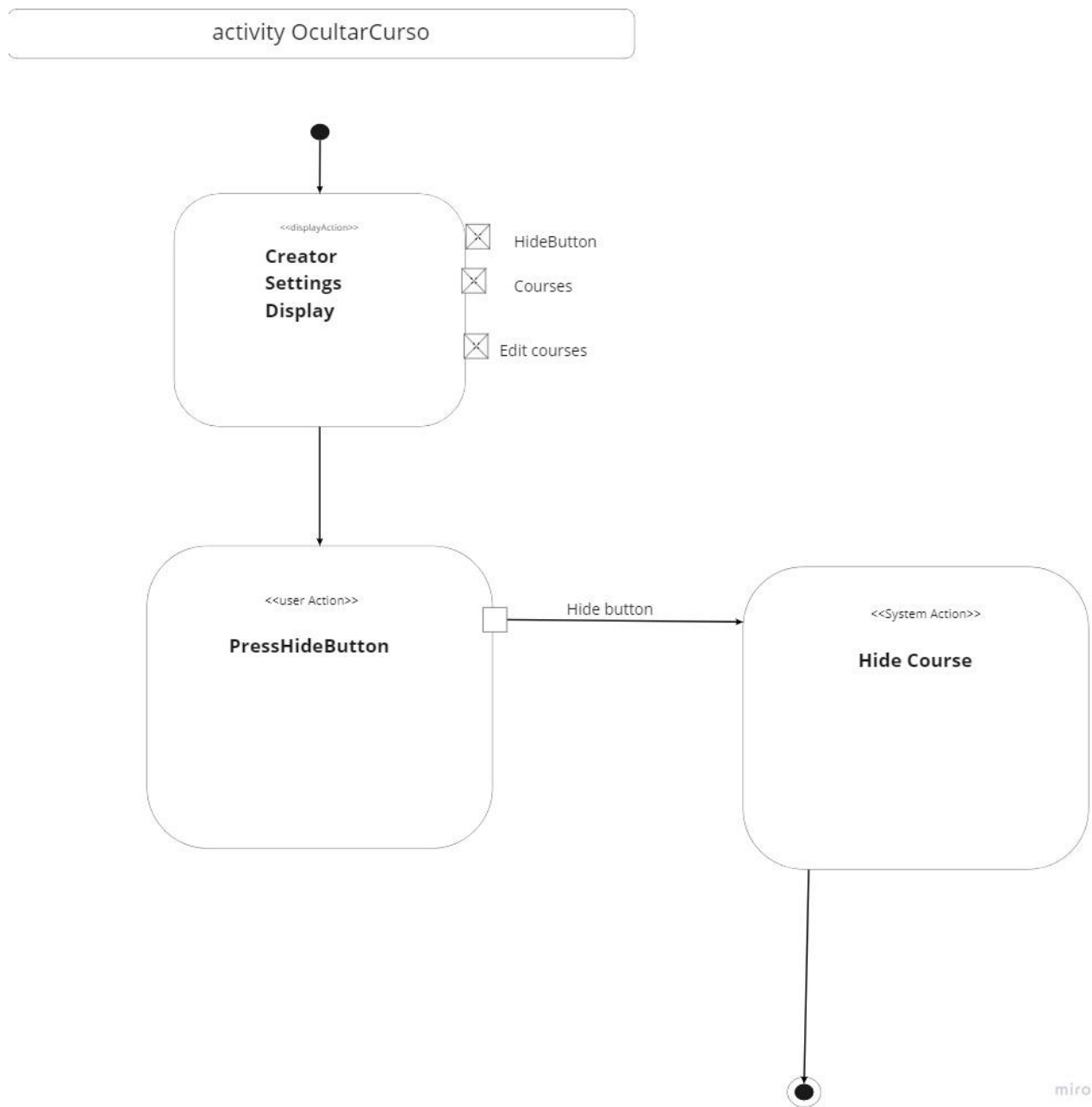
miro

## Caso de Uso: CrearCurso



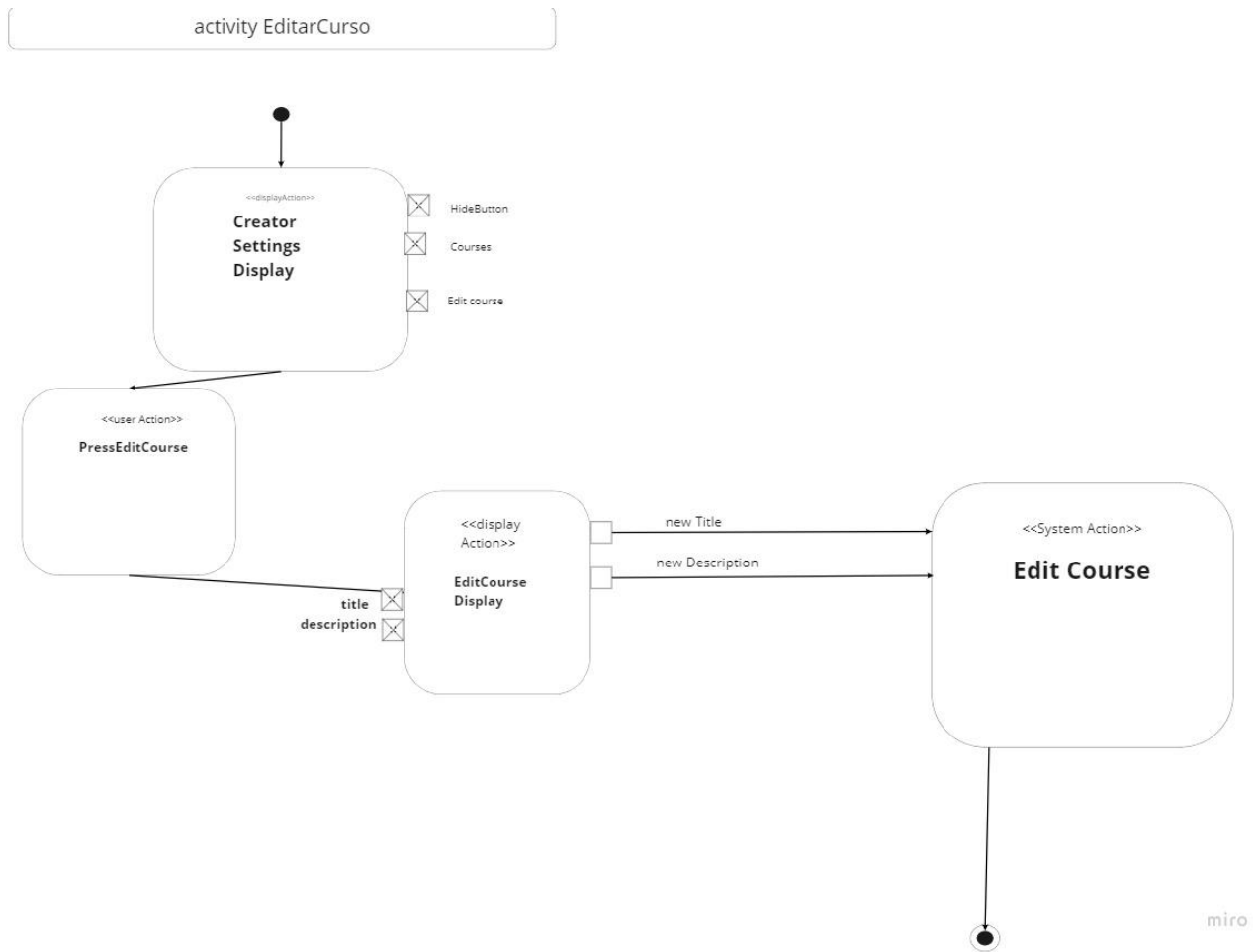
miro

## Caso de uso: OcultarCurso

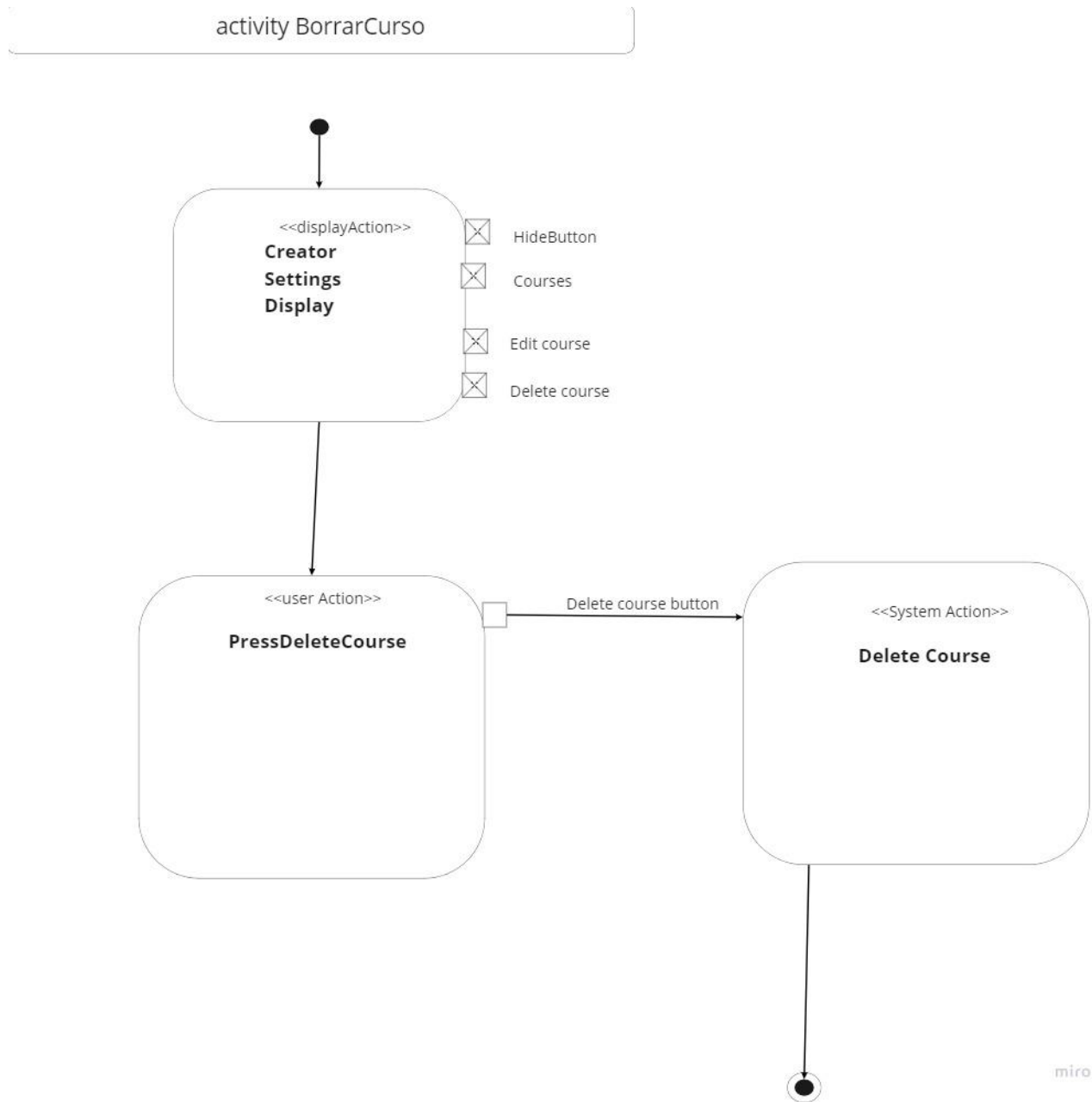


miro

## Caso de uso: EditarCurso

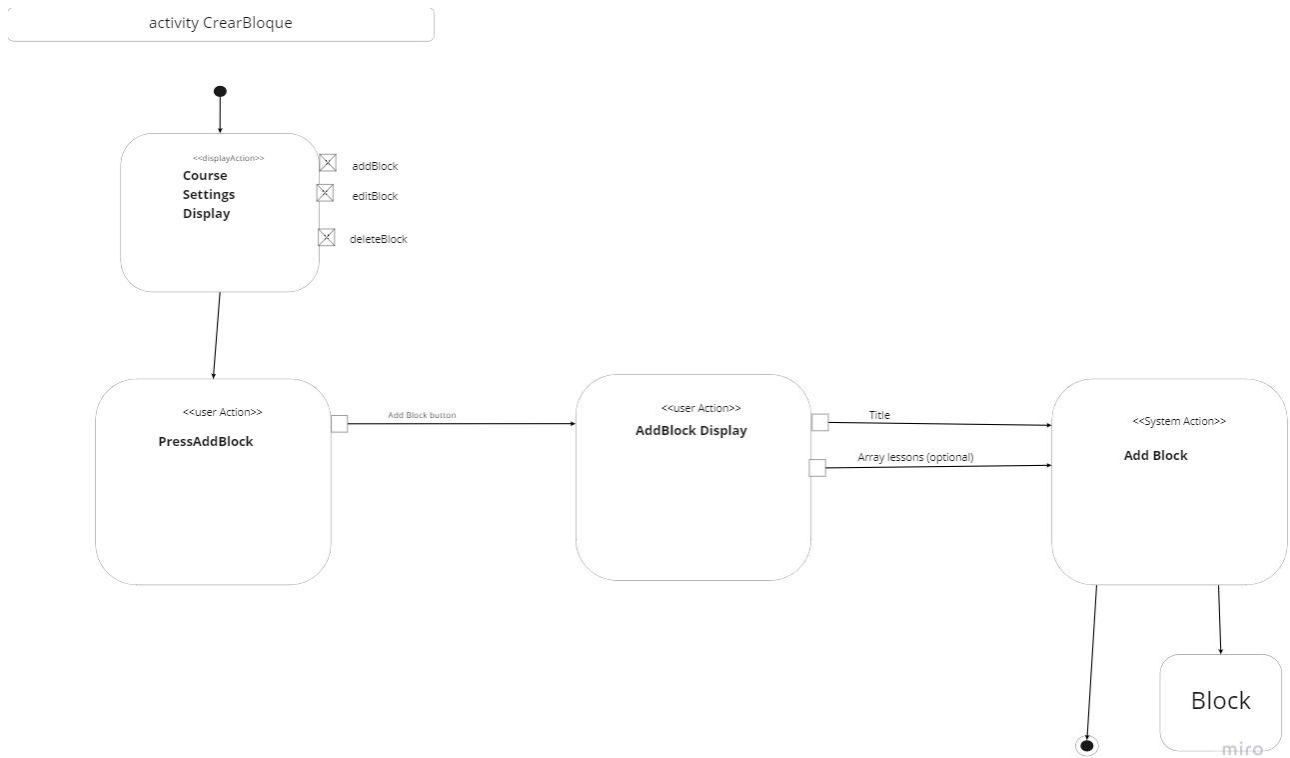


## Caso de uso: Borrar Curso

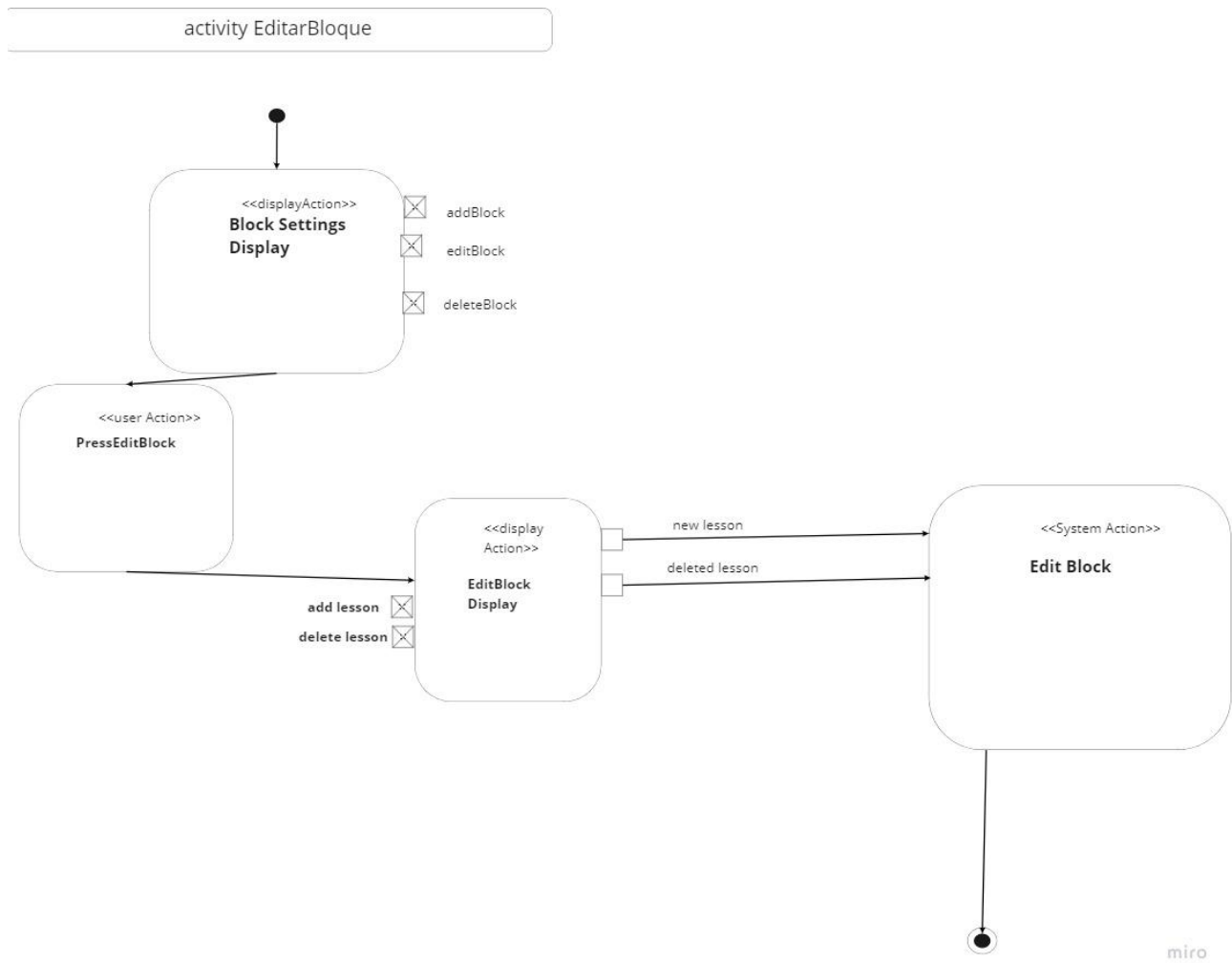


miro

## Caso de uso: CrearBloque



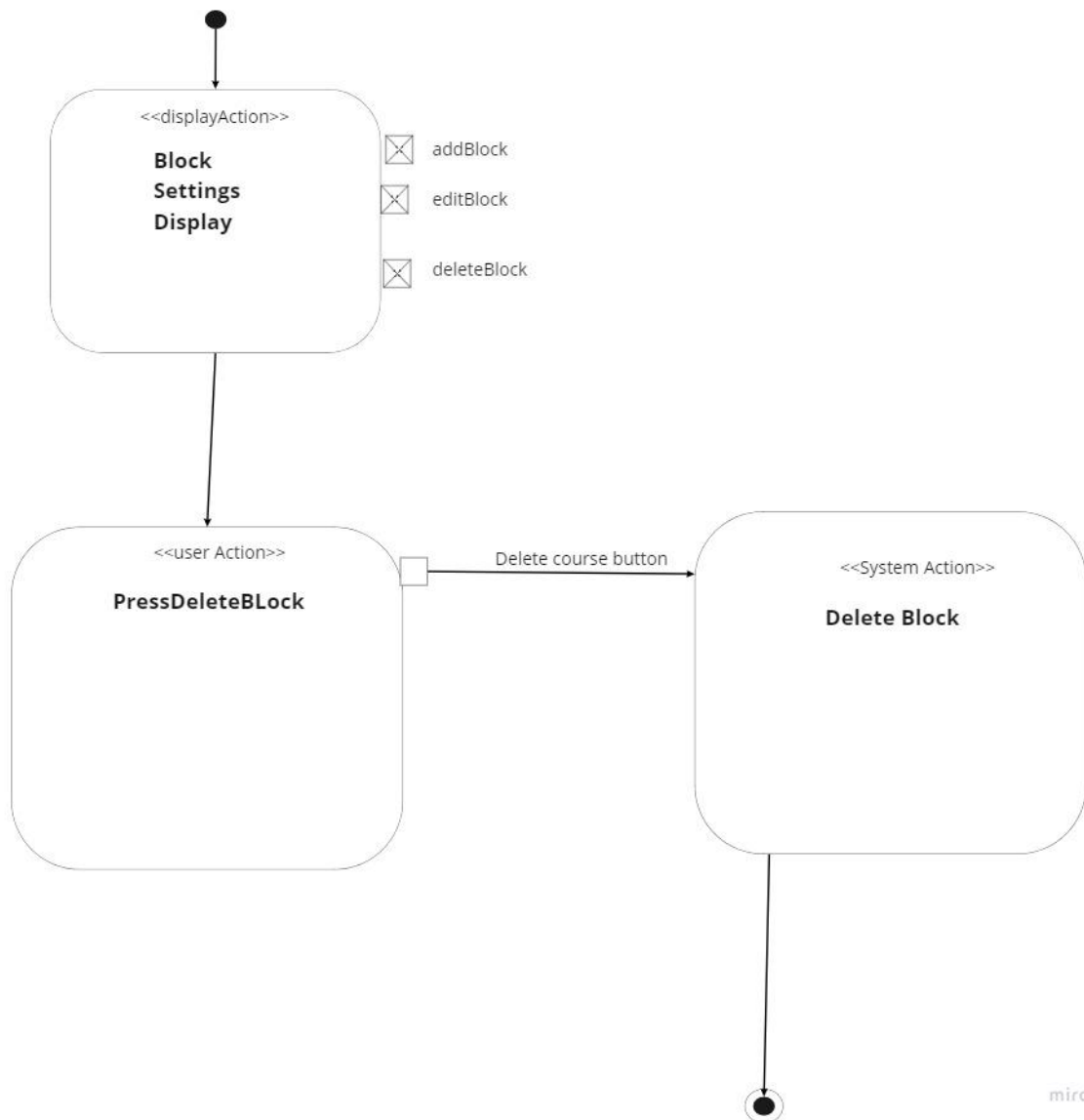
## Caso de uso: EditarBloque





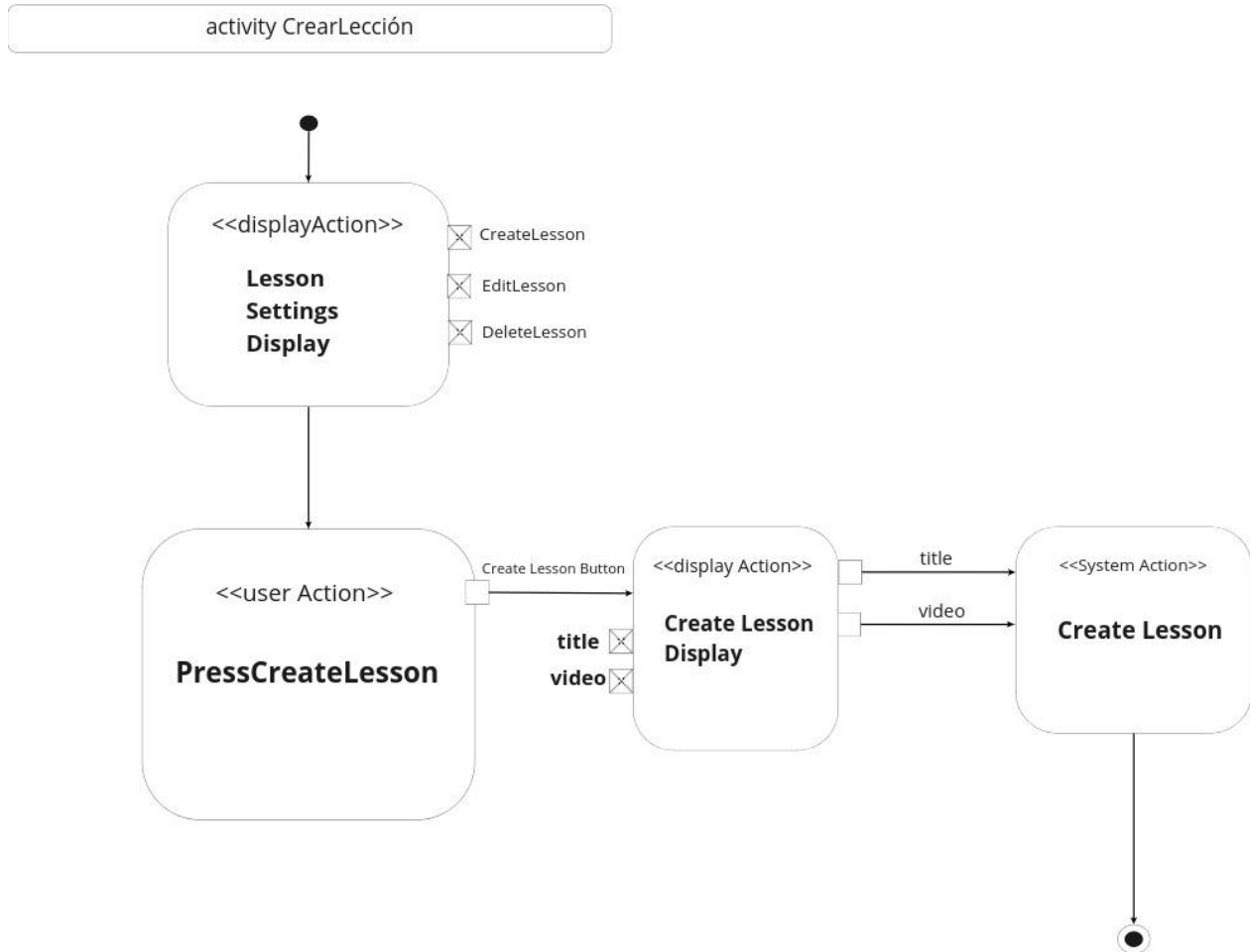
## Caso de uso: BorrarBloque

activity BorrarBloque



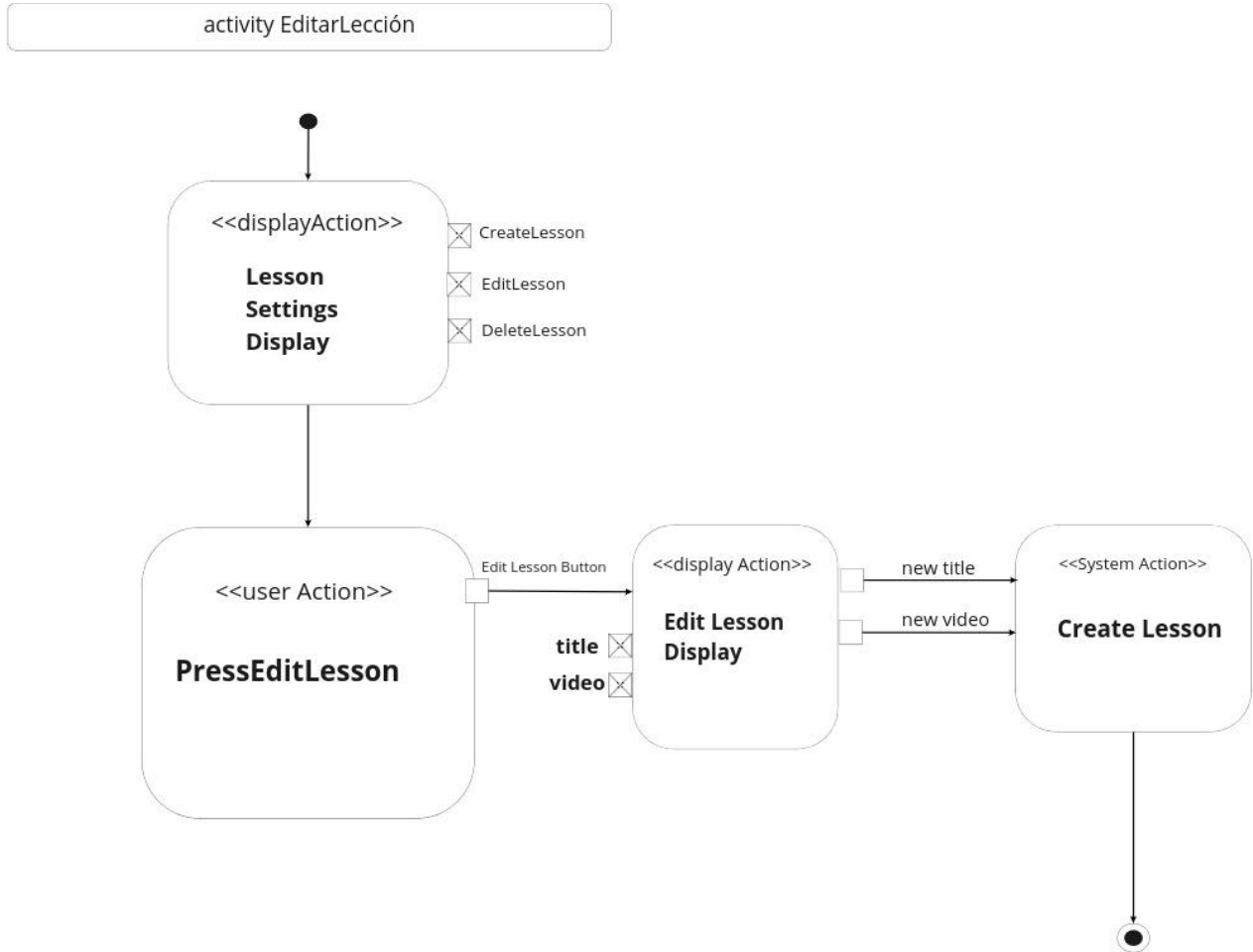
miro

## Caso de Uso: CrearLección



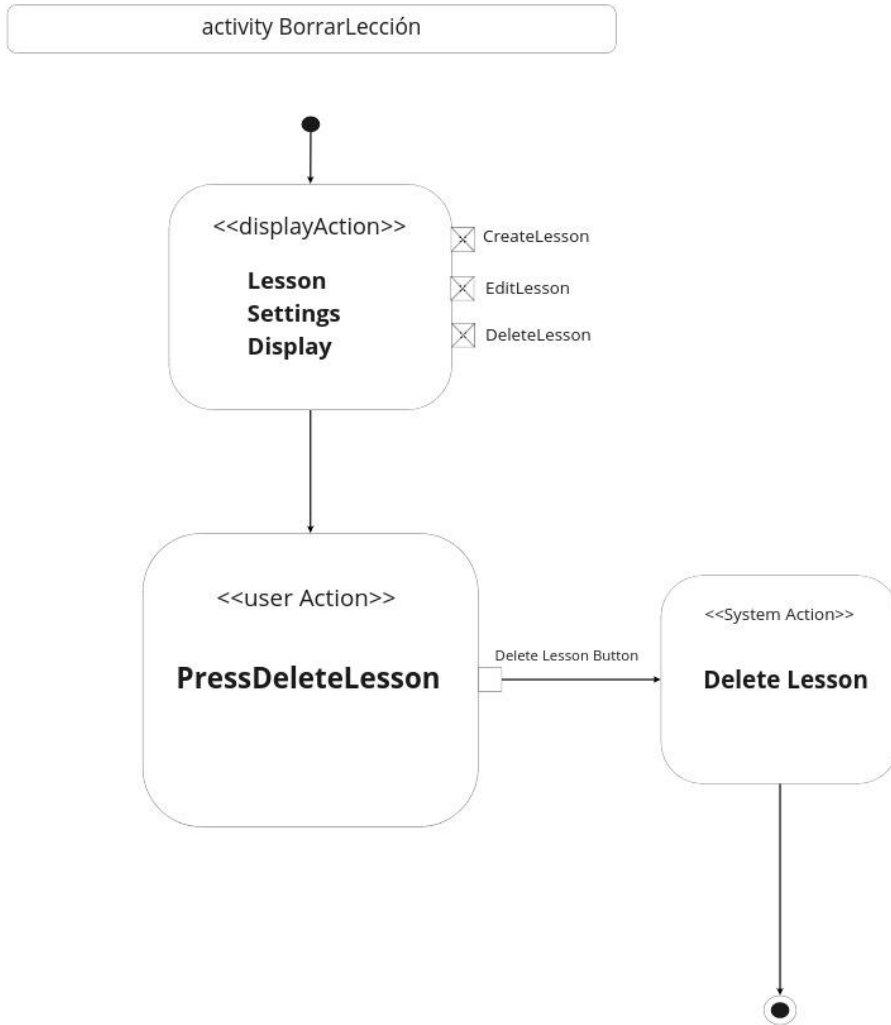
miro

## Caso de Uso: EditarLección



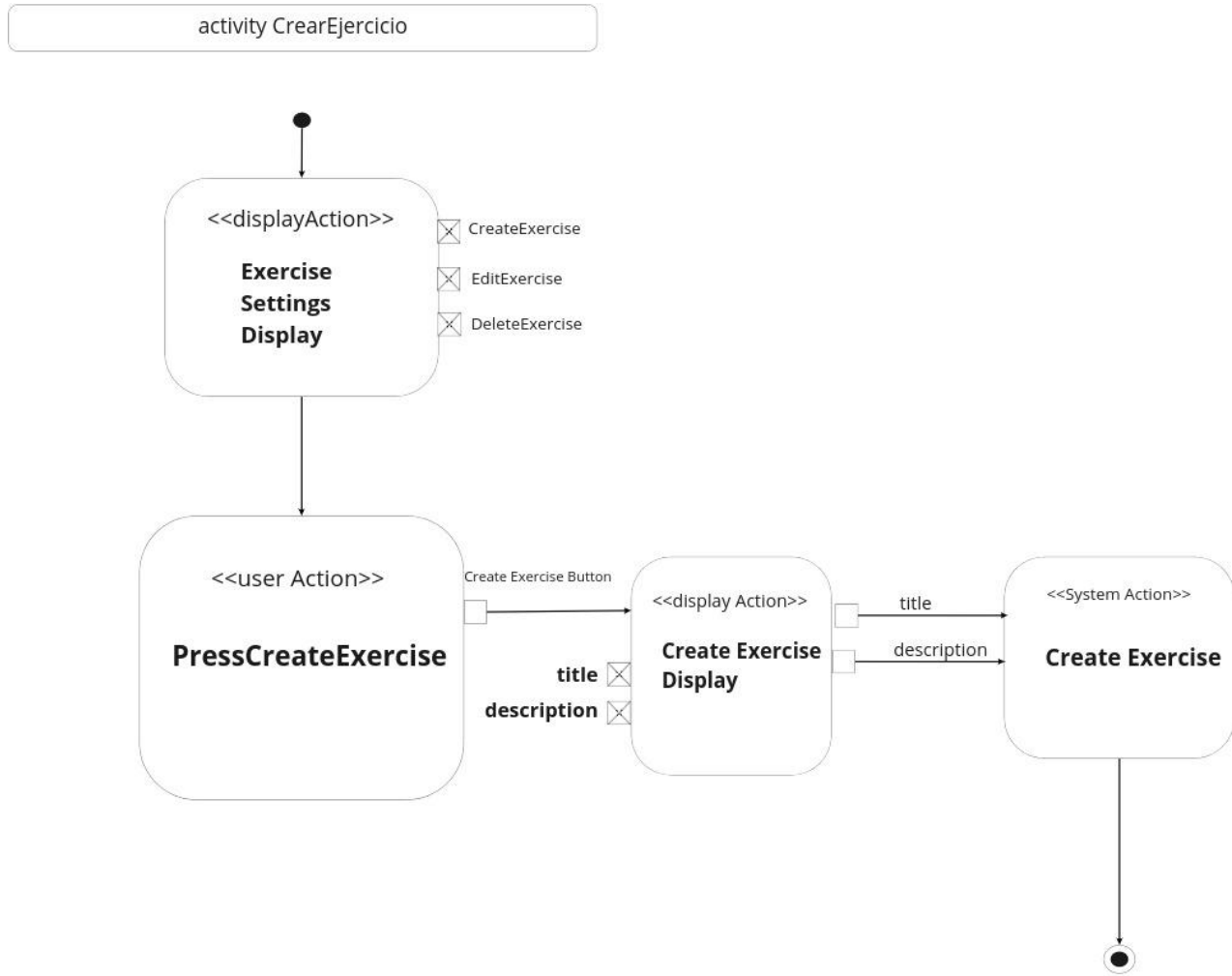
miro

## Caso de Uso: BorrarLección

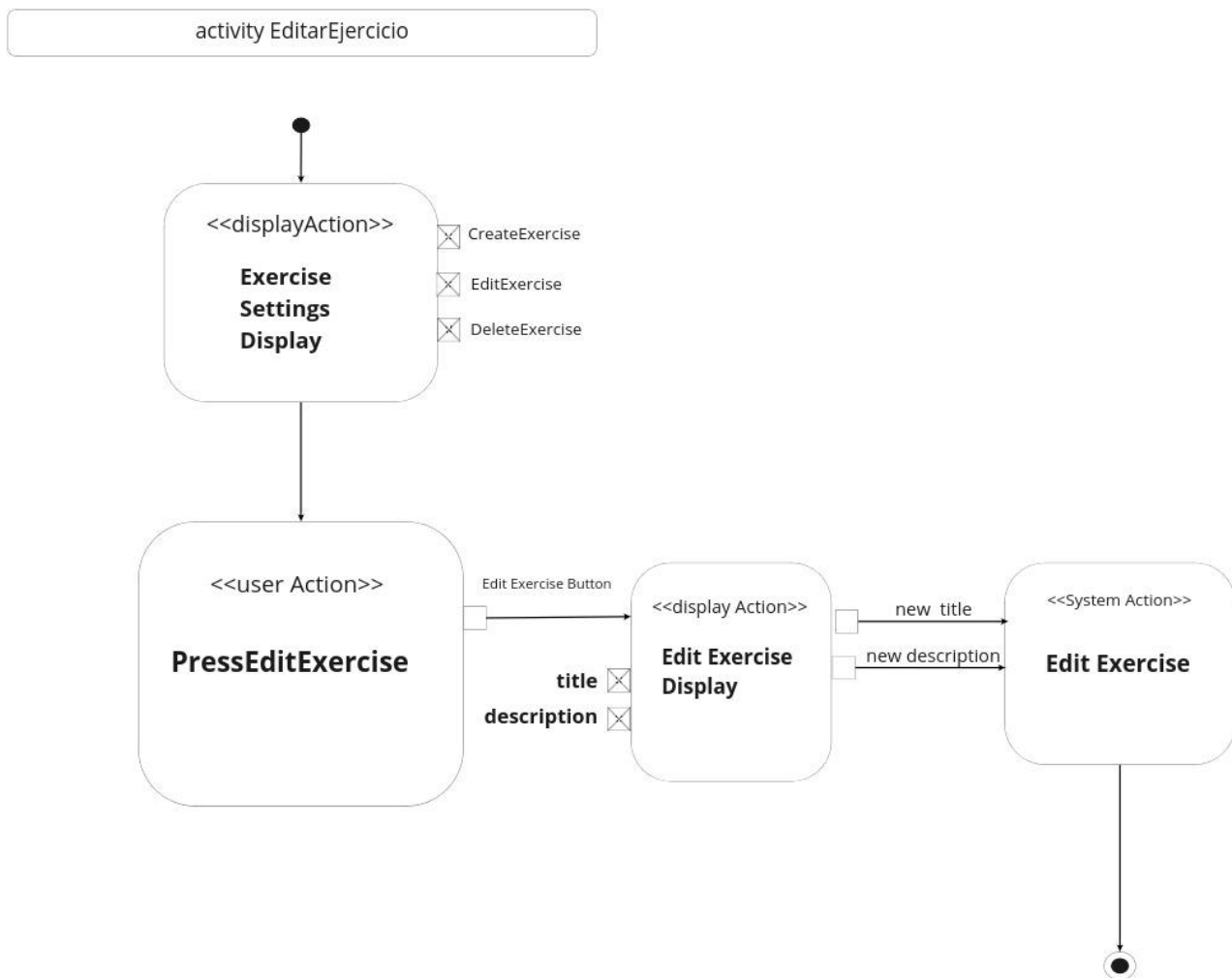


miro

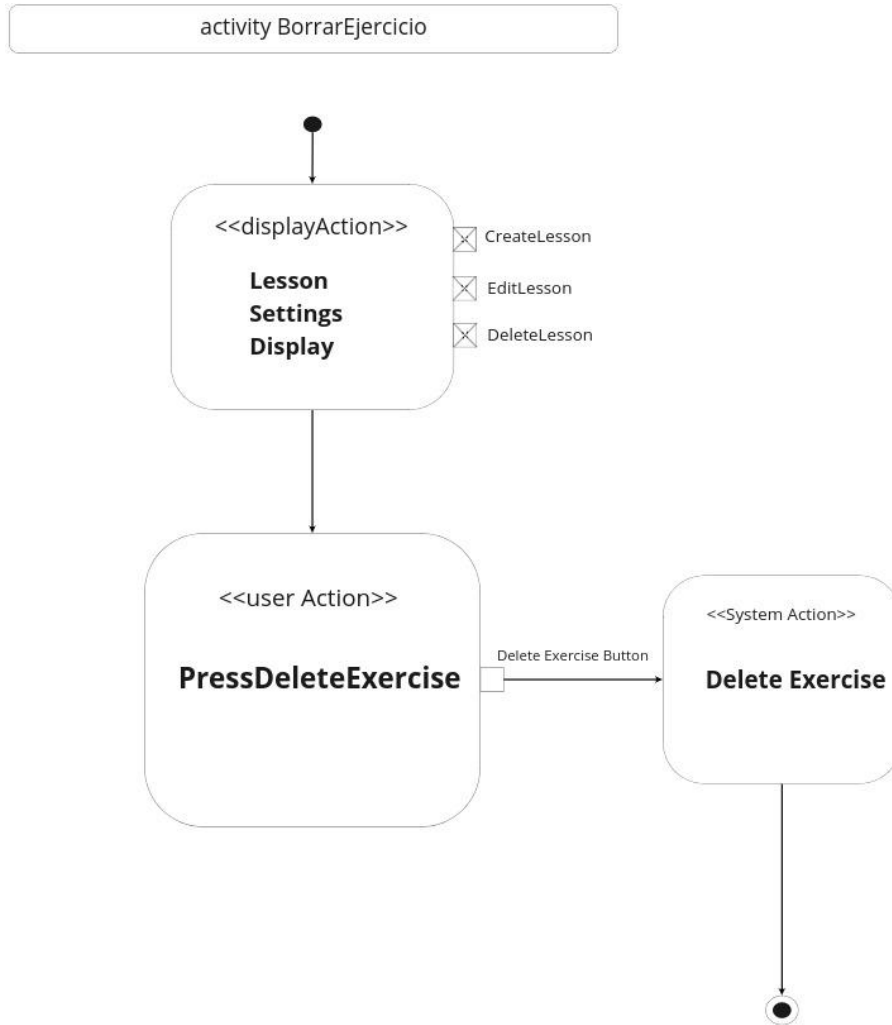
## Caso de Uso: CrearEjercicio



## Caso de Uso: EditarEjercicio

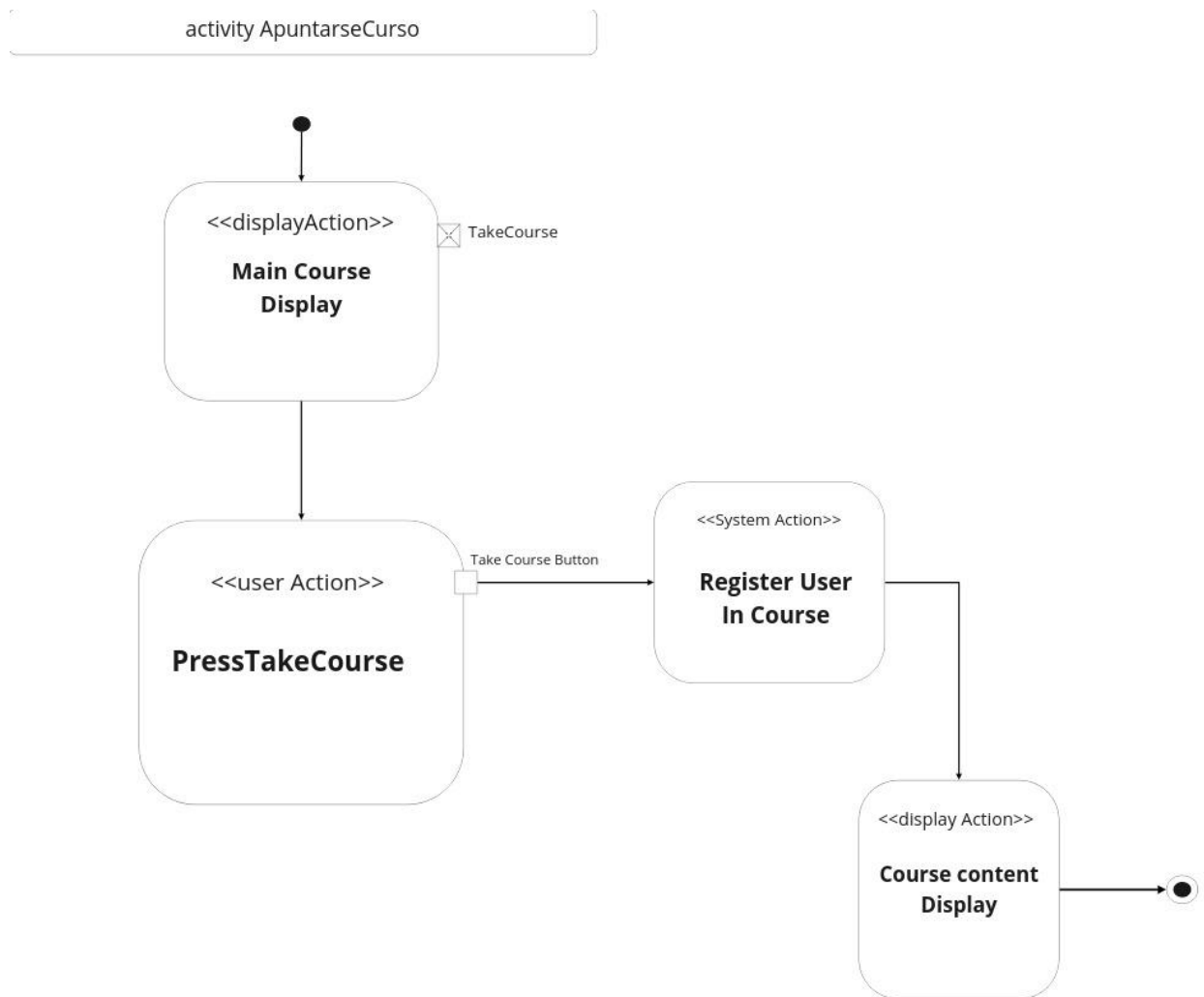


## Caso de Uso: BorrarEjercicio



miro

## Caso de Uso: ApuntarseCurso

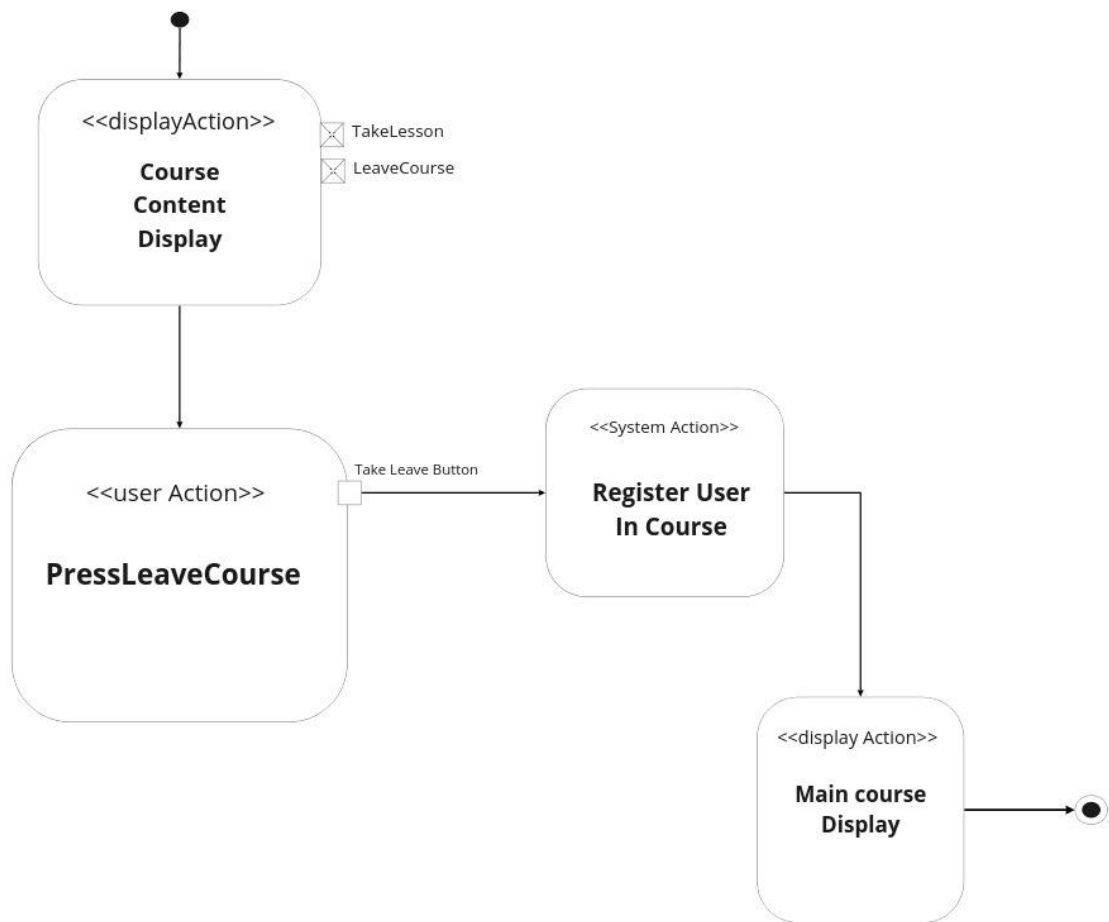


miro



## Caso de Uso: DesapuntarseCurso

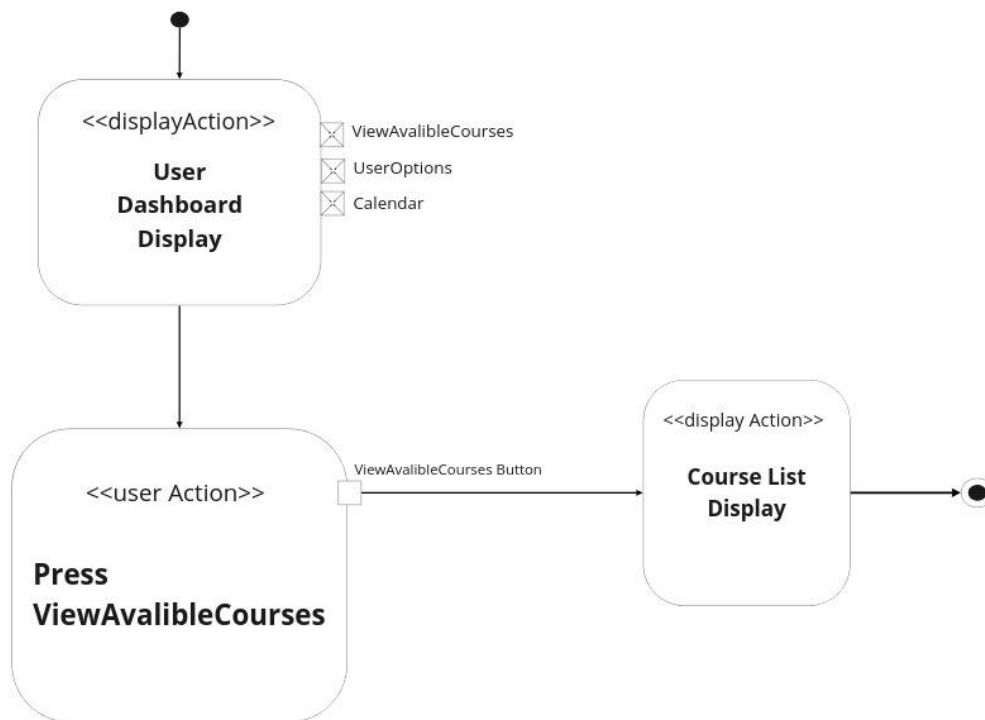
activity DesapuntarseCurso



miro

## Caso de Uso: MostrarListadoCursos

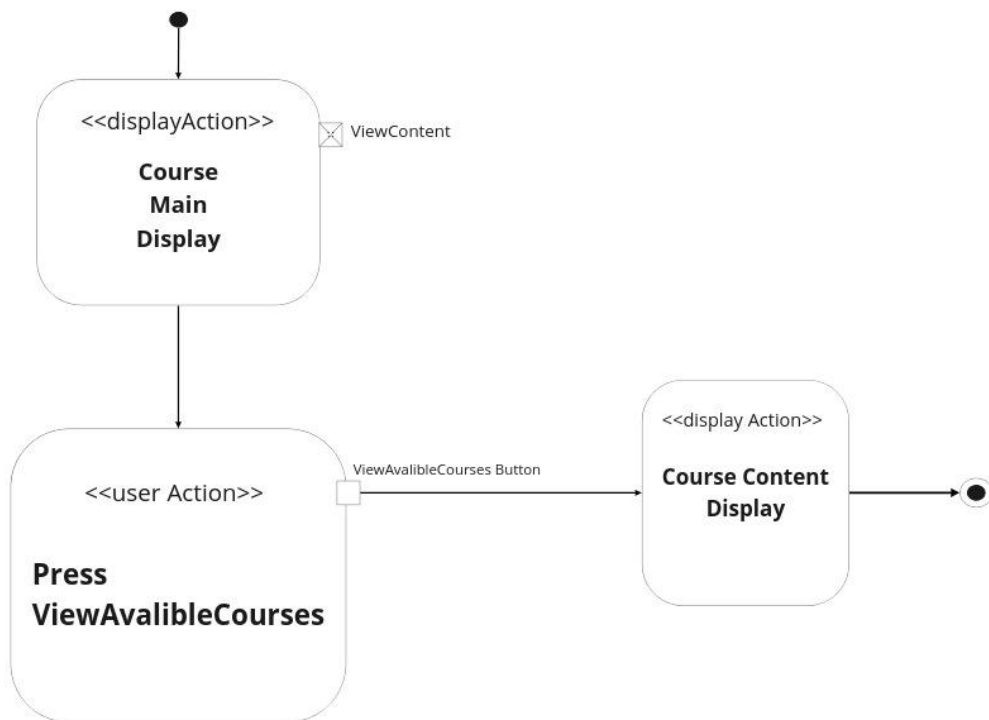
activity MostrarListadoCursos



miro

## Caso de Uso: MostrarBloquesDeCurso

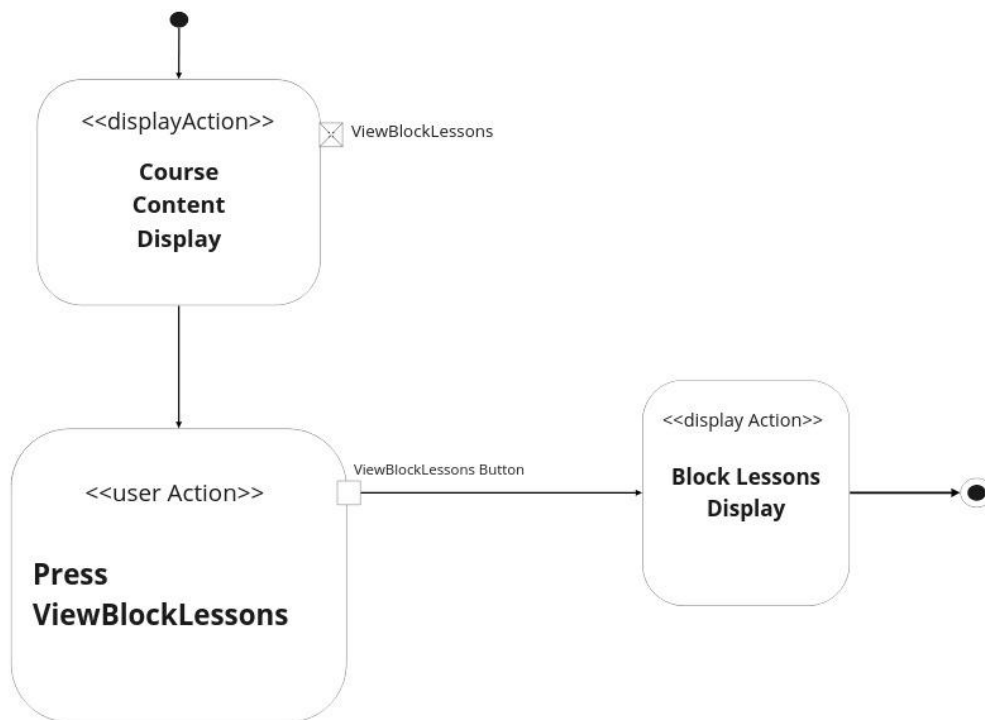
activity MostrarBloquesDeCurso



miro

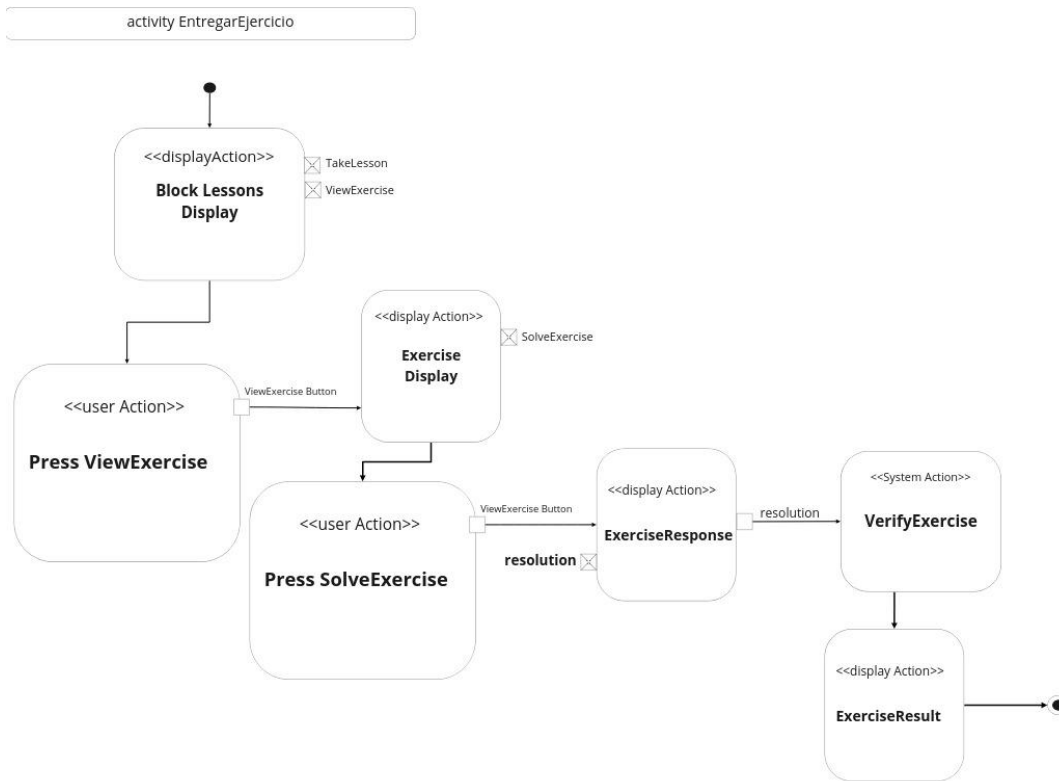
## Caso de Uso: MostrarLeccionesDeBloque

activity MostrarLeccionesDeBloque



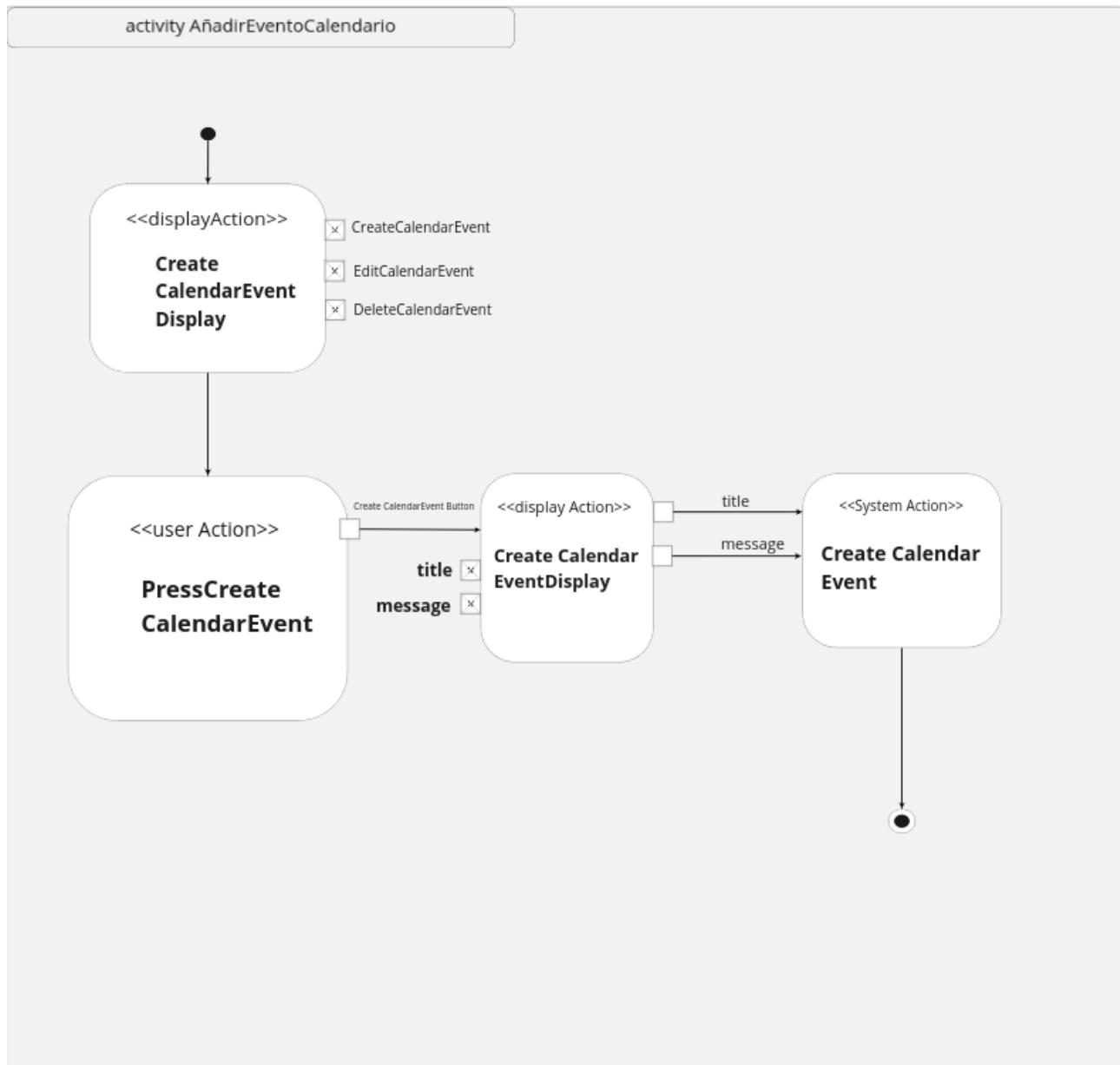
miro

## Caso de Uso: EntregarEjercicio

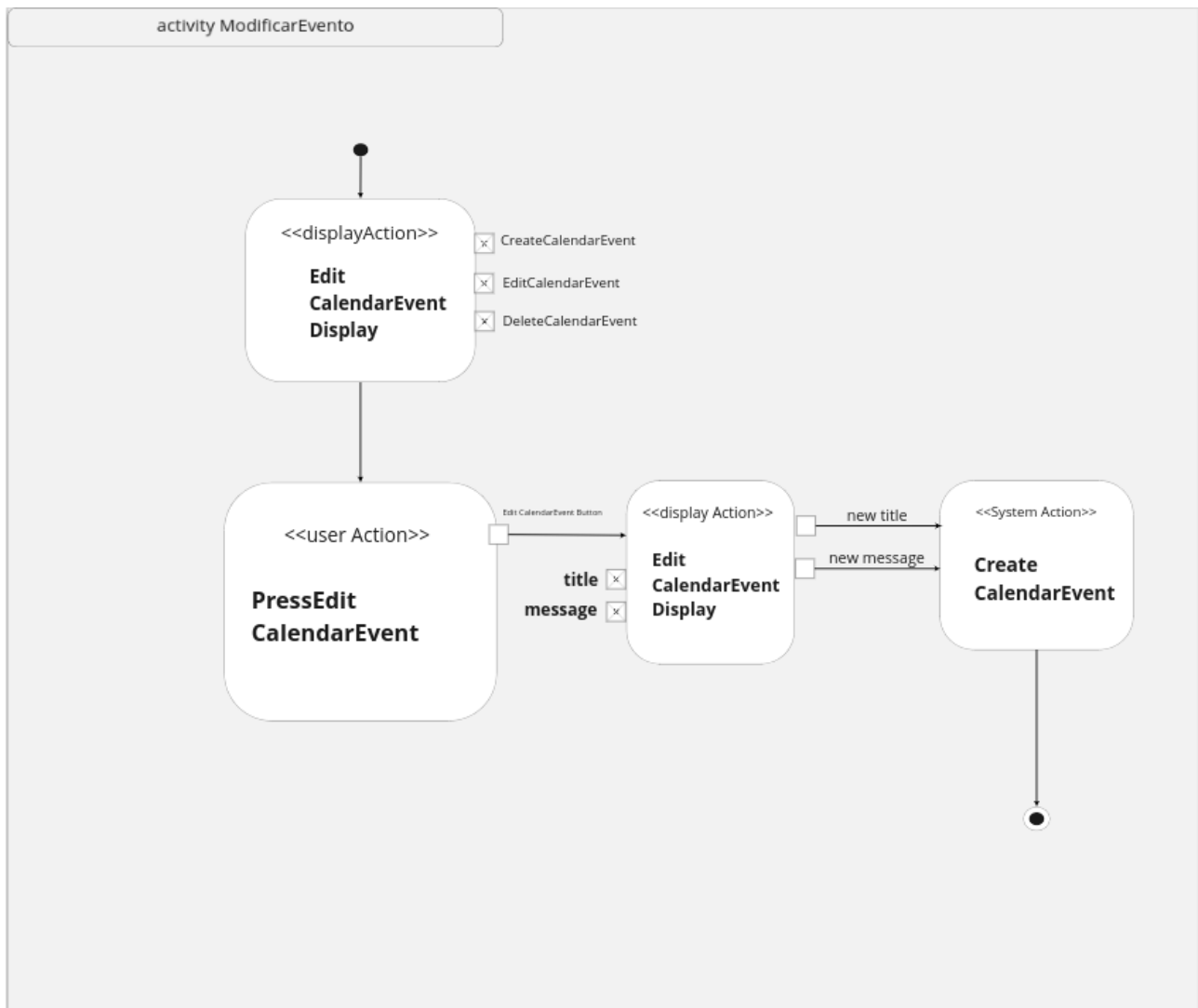


miro

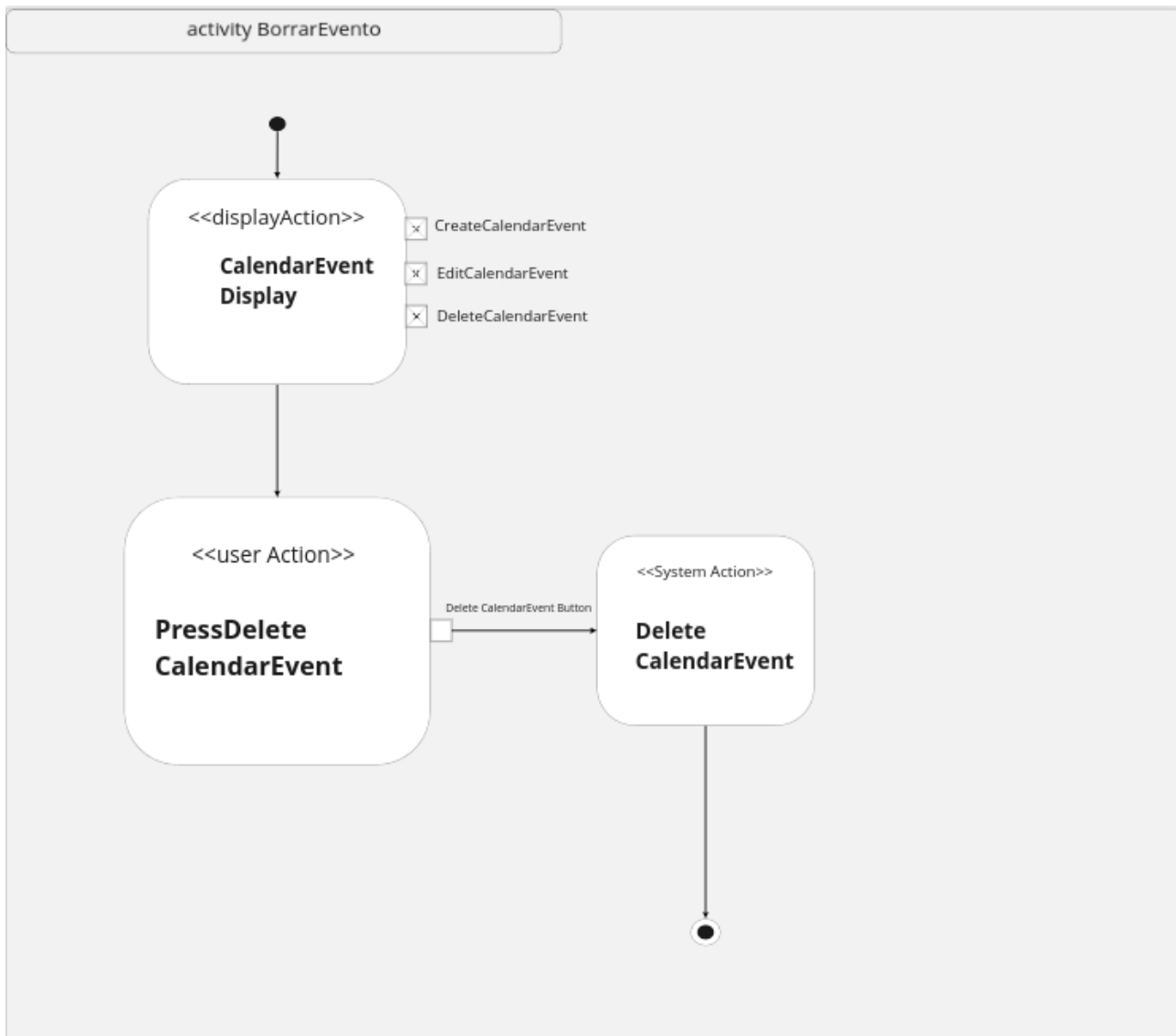
## Caso de Uso: AñadirEventoCalendario



## Caso de Uso: ModificarEvento

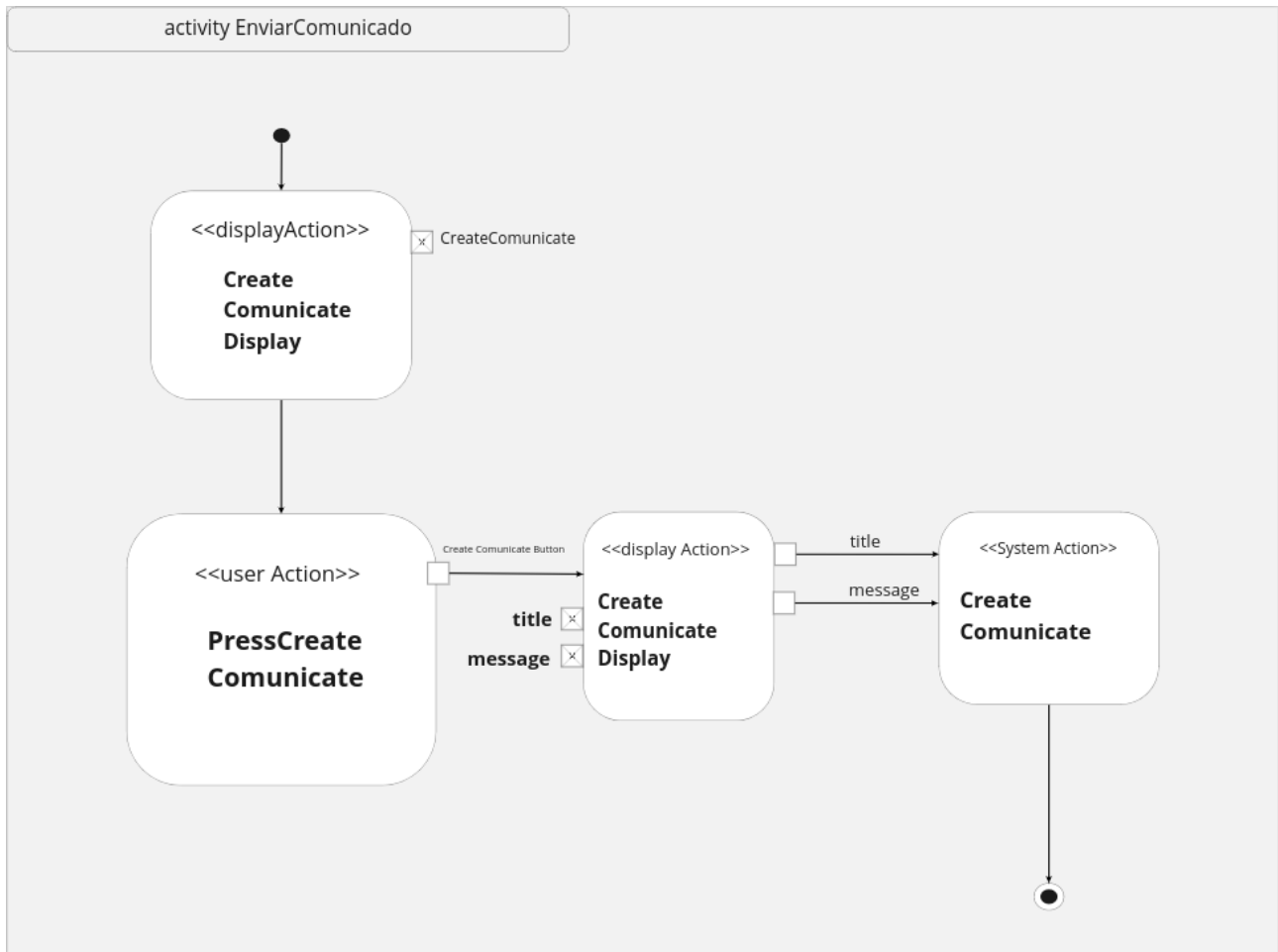


## Caso de Uso: BorrarEvento





## Caso de Uso: EnviarComunicado





# SCRUM

En este apartado hemos resumido la organización y la planificación de tareas que hemos realizado para estas dos primeras prácticas, a lo largo de tres sprints.

## Backlog

Lo primero que hacíamos, a la hora de organizar los sprints era repartir las tareas que iba a tener. Como los sprints eran de una semana, hemos intentado que el valor de la suma de los puntos de función de las tareas que se introdujesen en ese sprint no fuese superior a 30 ya que consideramos que completar 5 puntos de función al día era algo asequible. La suma total de los puntos de función de los tres sprints es de 90 puntos de función.

Tarea	Encargado	Sprint	Puntos
Redactar requisitos de información	Nicolás	1	2
Redactar requisitos no funcionales	Marcos Rivera	1	3
Redactar requisitos funcionales	Alex	1	3
Configuración Tableros Miro	Adrián	1	2
Diagrama Casos de Uso	David	1	3
Redacción del documento	Nicolás	1	3
Prueba de Login	Marcos Rodriguez	1	5
Demo de vídeos en APIs	Adrián	1	5
Revisión documento	Alex	1	2
Preparar sprint	Marcos Rivera	1	2
Integrar ZenHub con gitHub	Marcos Rivera	2	5
Desplegar frontend en gH-pages	Adrián	2	3
Instalación y configuración de Nodejs	Salcedo	2	3

*“La tabla x contiene el backlog de tareas”*

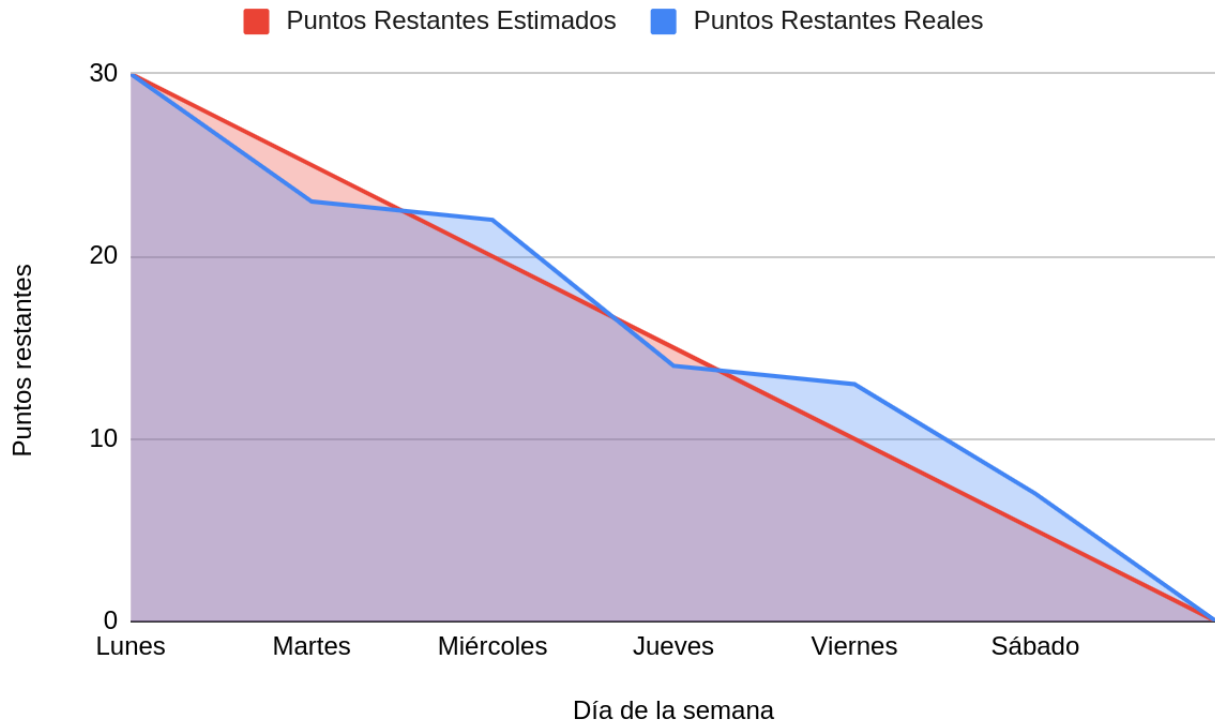
## Backlog

Tarea	Encargado	Sprint	Puntos
Instalación de Nodejs	Alex	2	2
Diseño de la interfaz	Nicolás	2	5
Preparar sprint	Marcos Rivera	2	2
Diseño del logo	Nicolás	2	2
Propuesta de arquitectura	Marcos Rodríguez	2	3
Configurar Google Drive	Alex	2	2
Investigación sobre Event Store	Adrián	2	3
Formatear documento	Alex	3	2
Dockerizar frontend	Marcos Rivera	3	3
Dockerizar backend	Adrián	3	8
Investigar stack de tecnologías	Marcos Rodriguez	3	2
Diseño paleta de colores	Nicolás	3	2
Diseño login	David	3	2
Configuración NodeWorkspaces	Adrián	3	3
Configuración del repositorio	Marcos Rodríguez	3	3
Documentación	Nicolás	3	3
Preparar sprint	Marcos Rivera	3	2

*“La tabla x contiene el backlog de tareas”*

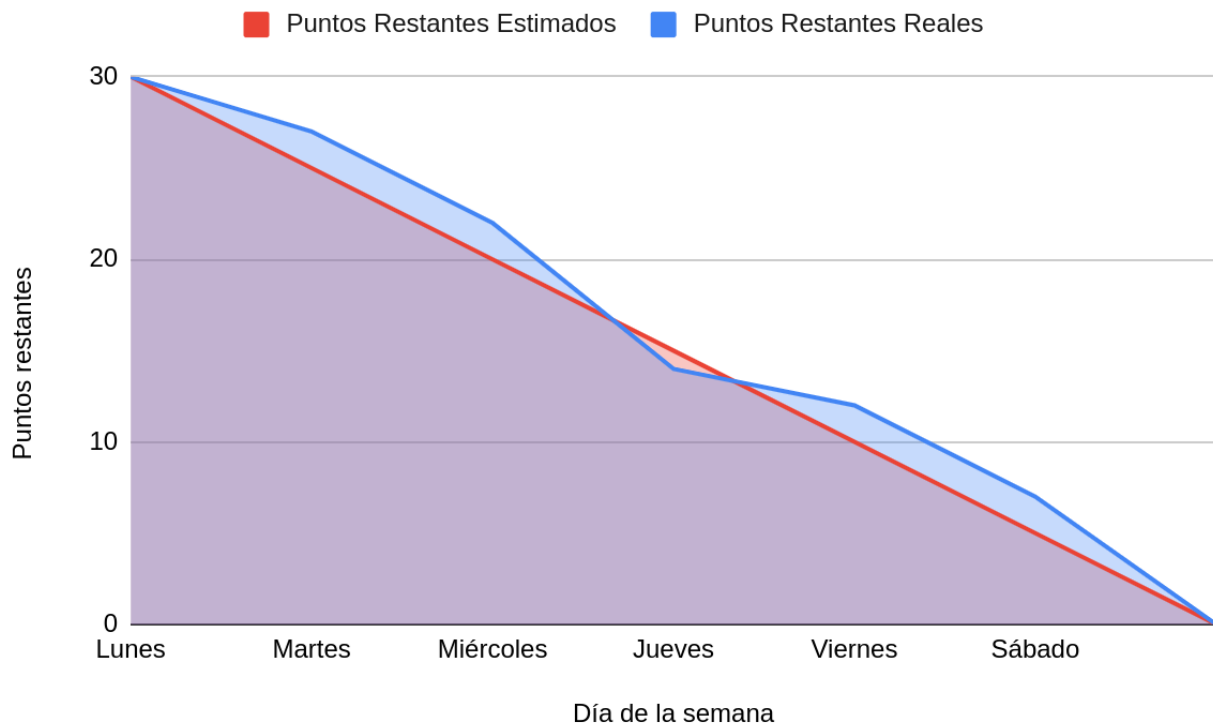
En esta página se encuentran los gráficos resultantes de las iteraciones de cada sprint:

### Sprint 1: 24-02-22 - 03/03/22



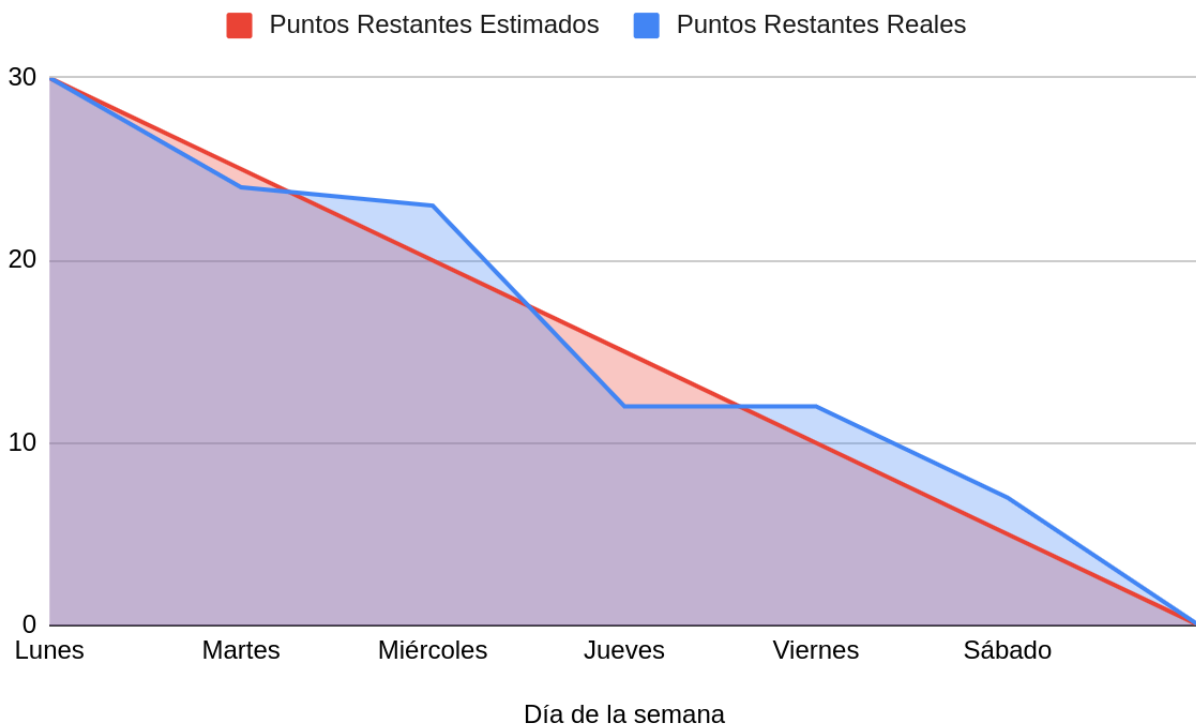
*“La imagen 8 contiene el gráfico de quemado del primer sprint”*

## Sprint 2: 03-03-22 - 10/03/22



“La imagen 9 contiene el gráfico de quemado del segundo sprint”

## Sprint 3: 10-03-22 - 17/03/22



“La imagen 10 contiene el gráfico de quemado del tercer sprint”