




Practica de laboratorio	
Materia:	Análisis y Diseño de Sistemas
Unidad del Temario:	Documentación del proyecto
Apellido paterno, materno y Nombre(s):	Patlani Mauricio Adriana
Boleta del estudiante:	2020630637
Firma del estudiante:	
Apellido paterno, materno y Nombre(s):	Ramírez Vázquez Oscar
Boleta del estudiante:	2020630610
Firma del estudiante:	
Apellido paterno, materno y Nombre(s):	Sandoval Hernández Eduardo
Boleta del estudiante:	2020630632
Firma del estudiante:	
Grupo:	5CM5
Líder del equipo	Sandoval Hernández Eduardo
Título Oficial de Registro del Proyecto:	Aplicación web de detección de pose con la cámara web utilizando tensorflow react con ecmascript 6, babel y webpack
URL del Proyecto:	https://github.com/sonu275981/Pose-Detection-by-using-ml5.js-library

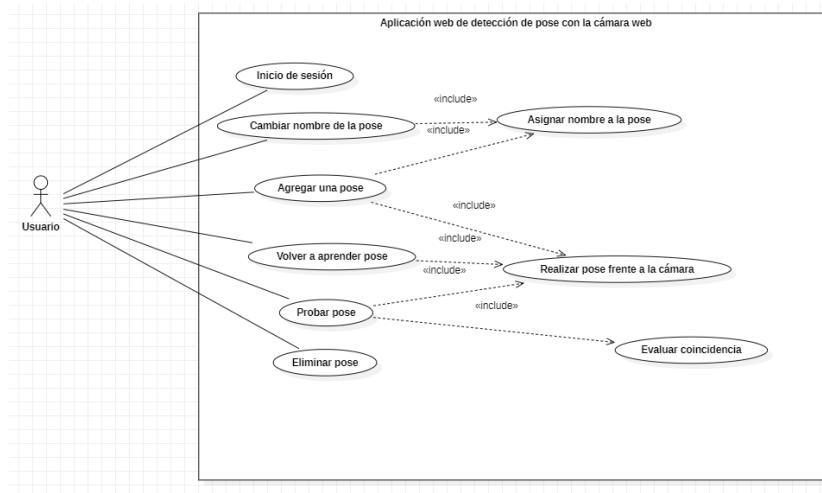
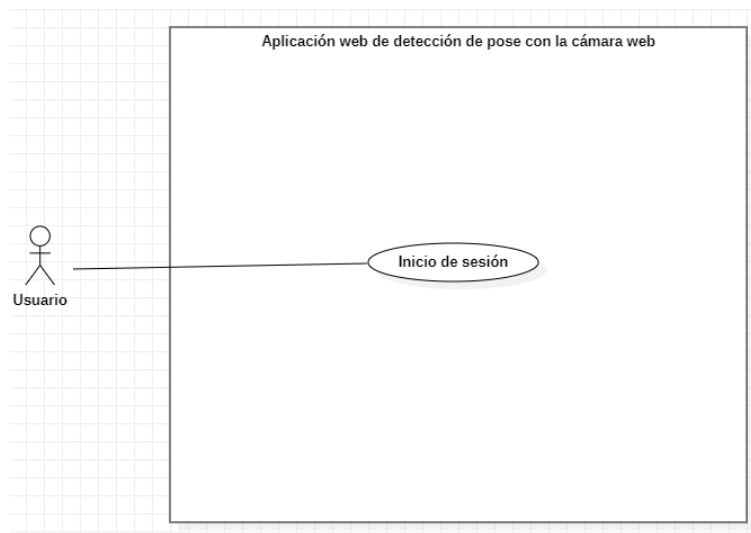
Descripción de la práctica:
El estudiante estructurara los siguientes diagramas UML del proyecto asignado.
Documentación UML solicitada del proyecto asignado con StarUML:
<ul style="list-style-type: none"> • Documento de la metodología seleccionada • Documento de los requerimientos funcionales y no funcionales • Documento de Diagramas de caso de uso • Documento de Tablas de descripción de casos de uso • Documento de Diagramas de clase • Documento de Diagramas de secuencia • Documento de Diagrama de estados • Documento de Diagrama de actividades

- Documento de Diagrama de componentes
- Documento de Diagrama de paquetes

Diagramas UML: Diagramas de caso de uso, Tablas des Descripción de casos de uso, Diagramas de clase, Diagramas de secuencia, Diagrama de estados, Diagrama de actividades, Diagrama de componentes, Diagrama de paquetes

- Diagramas de caso de uso:

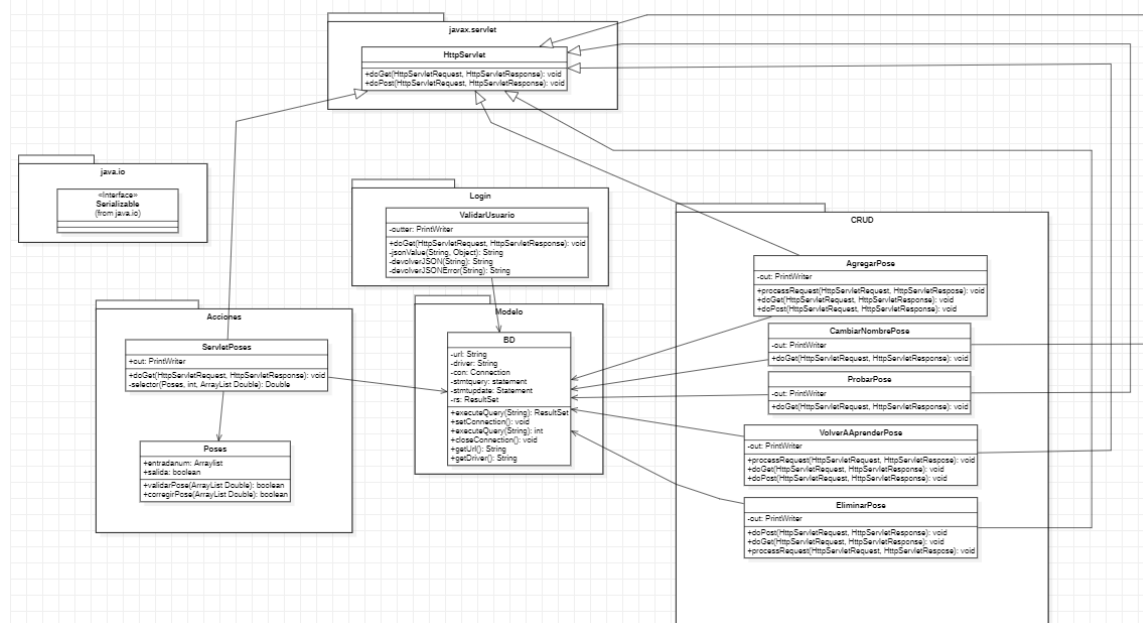
Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementará. Simplemente muestran los pasos que el actor sigue para realizar una tarea.



- Diagramas de clase:

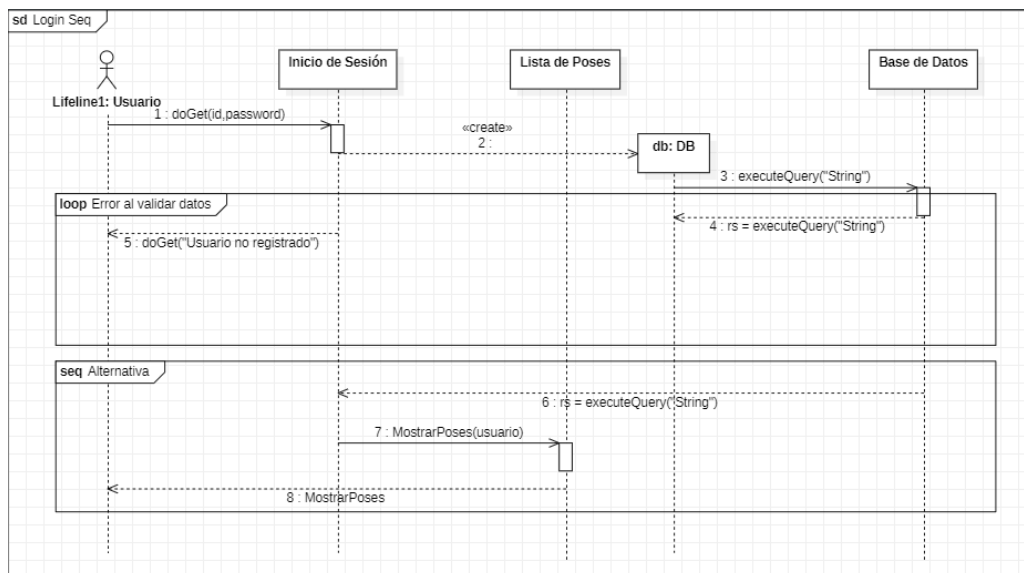
Se crea un diagrama de clases para diseñar explícitamente la estructura de una aplicación modelando sus respectivas clases, atributos, métodos y la relación entre ellos. Este diagrama se puede utilizar para describir lo que hay en el sistema de una manera que sea fácil de entender su estructura y dé una idea general de la aplicación.

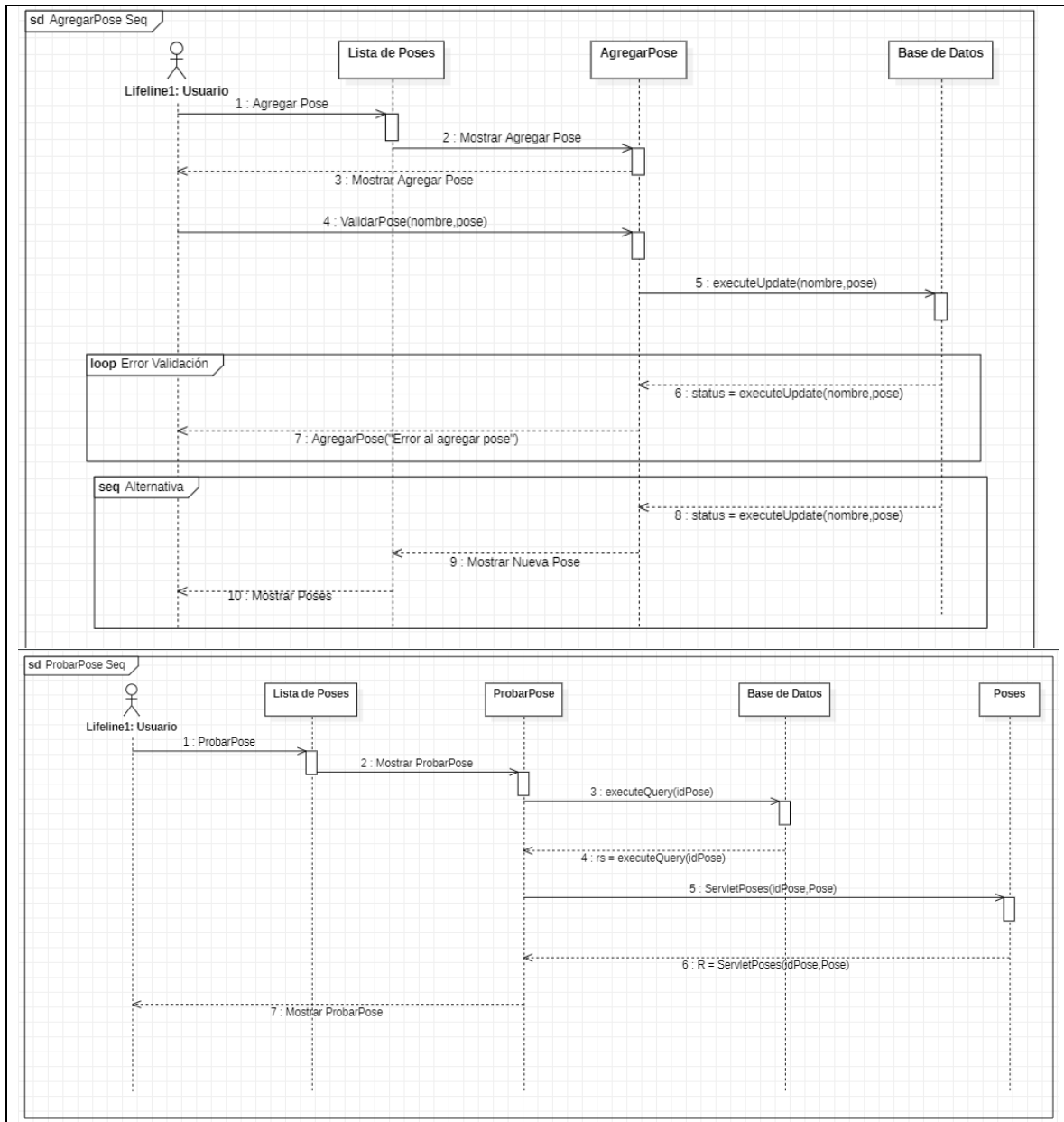
El diseño de un diagrama de clases expresa los requisitos específicos que necesita el sistema.

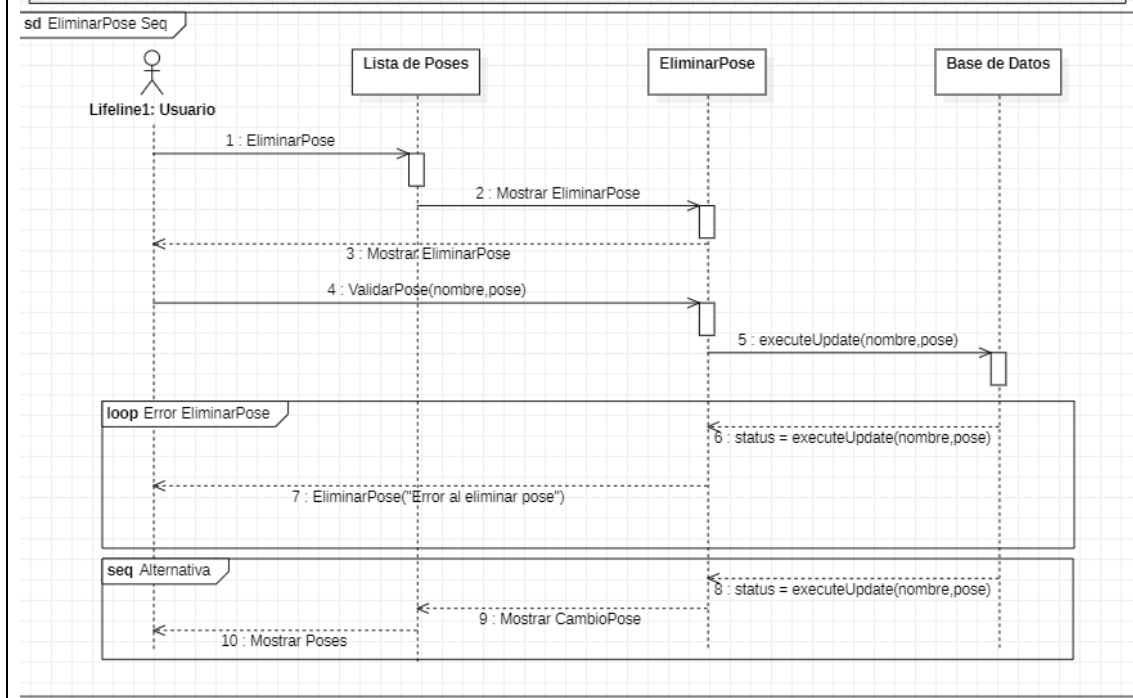
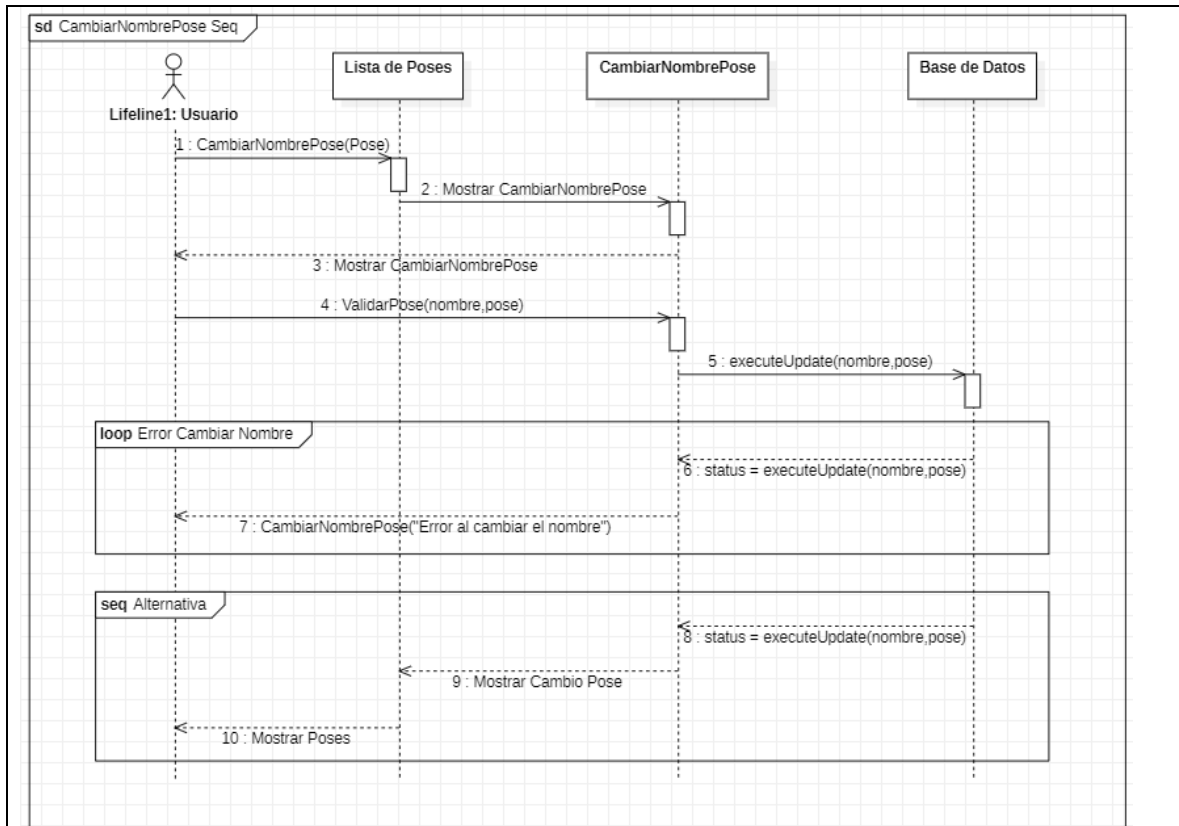


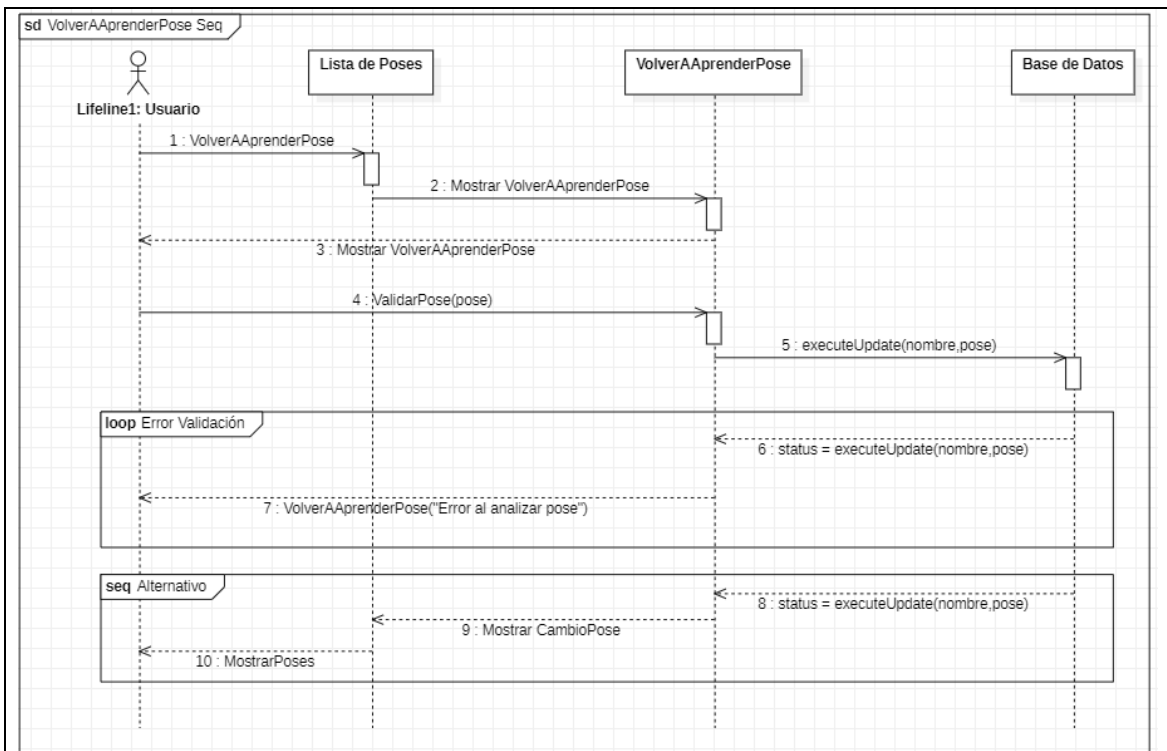
- Diagramas de secuencia:

Para el diagrama de secuencia, se crearon para mostrar y describir los procesos, como y en qué orden funcionan los objetos en la aplicación. Se crearon diagramas de secuencia para que el proyecto inicie sesión, agregar una pose, probar una pose, cambiar el nombre de una pose, eliminar una pose y volver a aprender una pose.



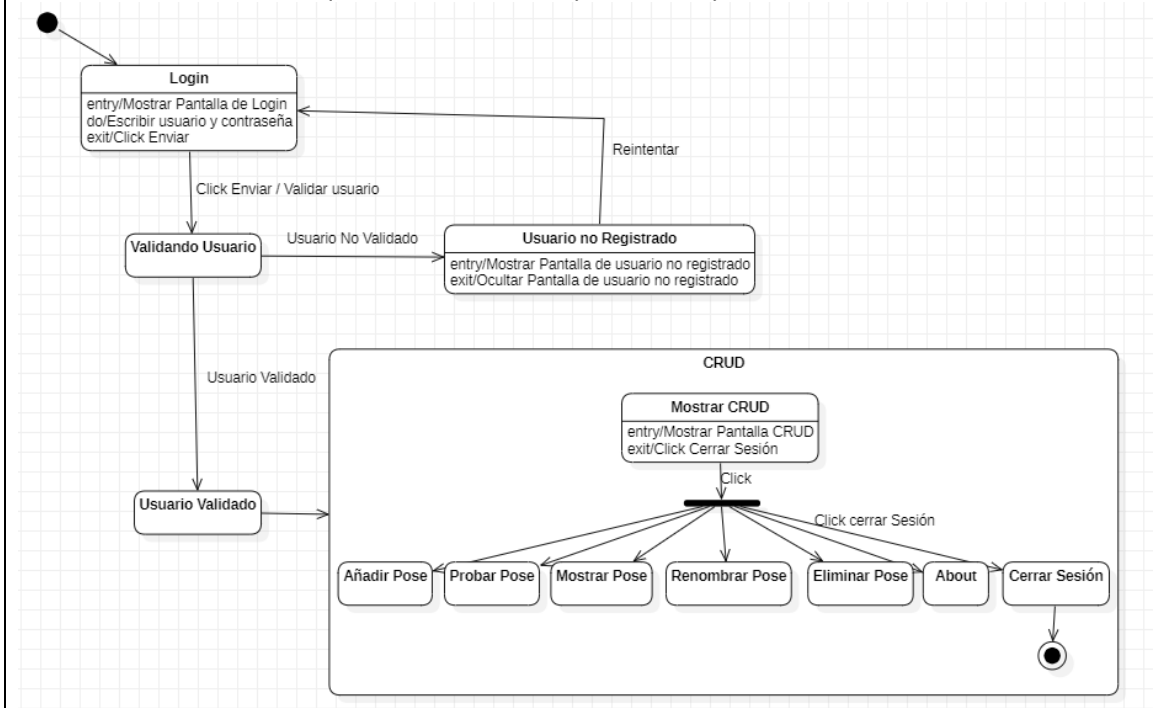


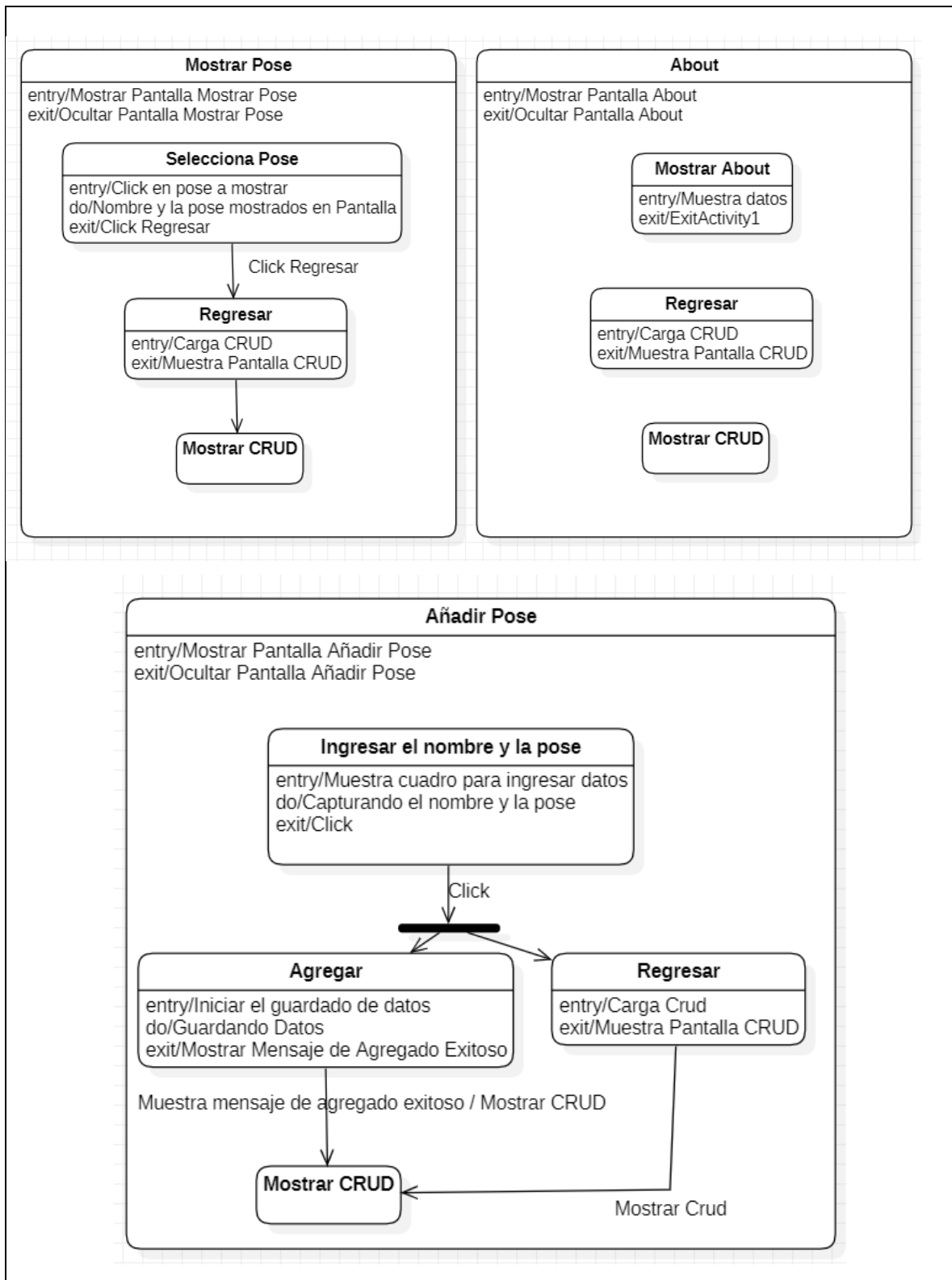


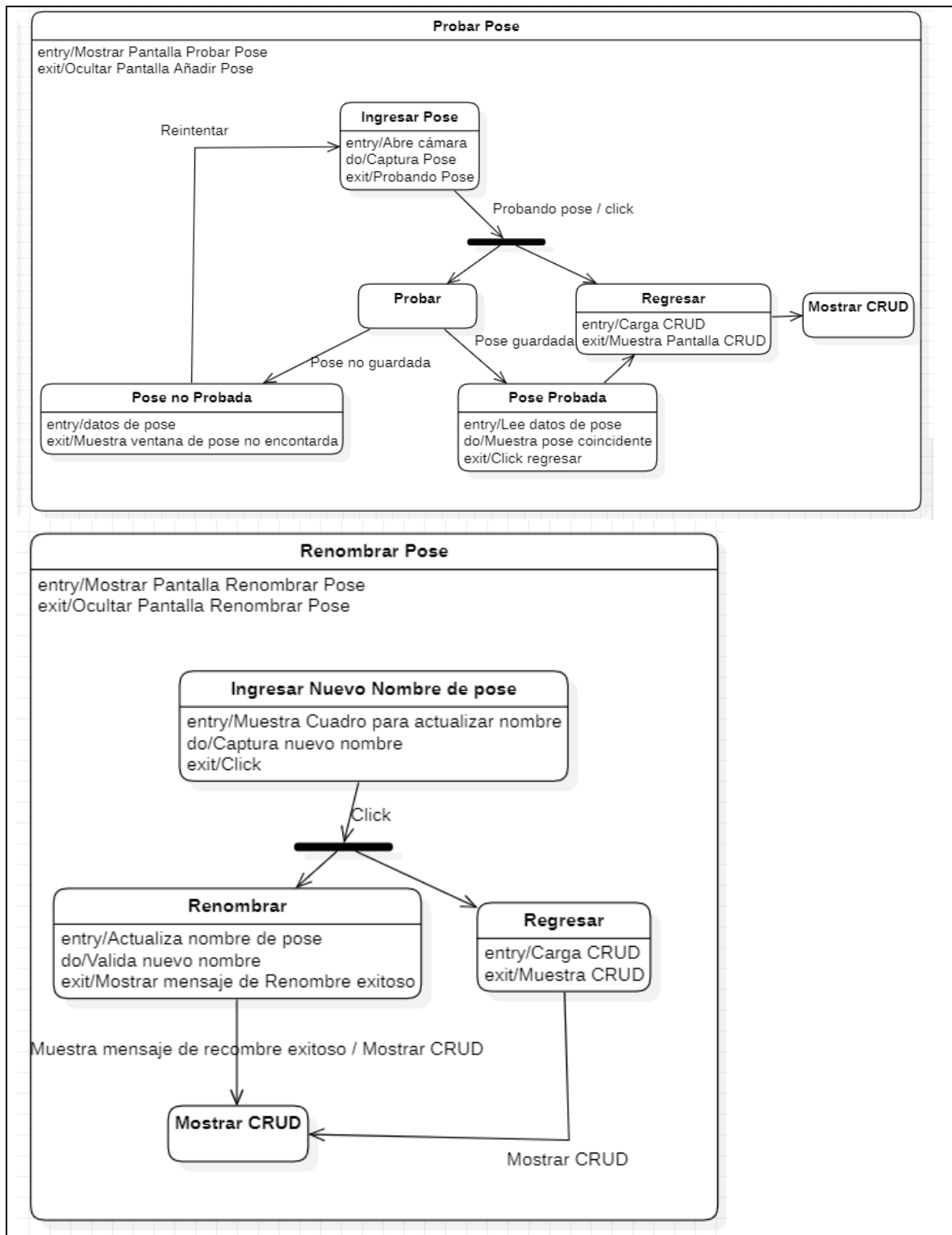


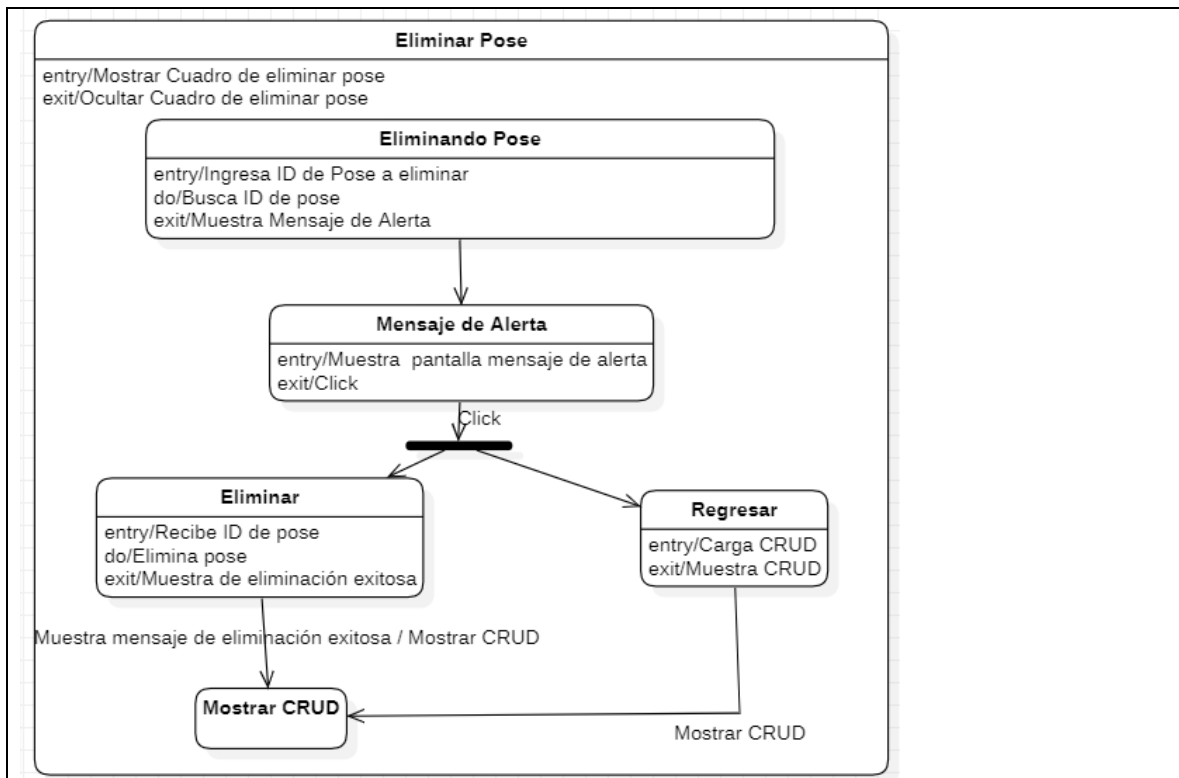
- Diagrama de estados:

Se puede usar el diagrama de estado para representar estados y transiciones, es decir, representan la serie de estados por los que pasa un objeto durante su vida útil en una aplicación en respuesta a eventos. Para una aplicación, puede observar el comportamiento de un objeto de acuerdo con acciones que ocurren en una aplicación específica.



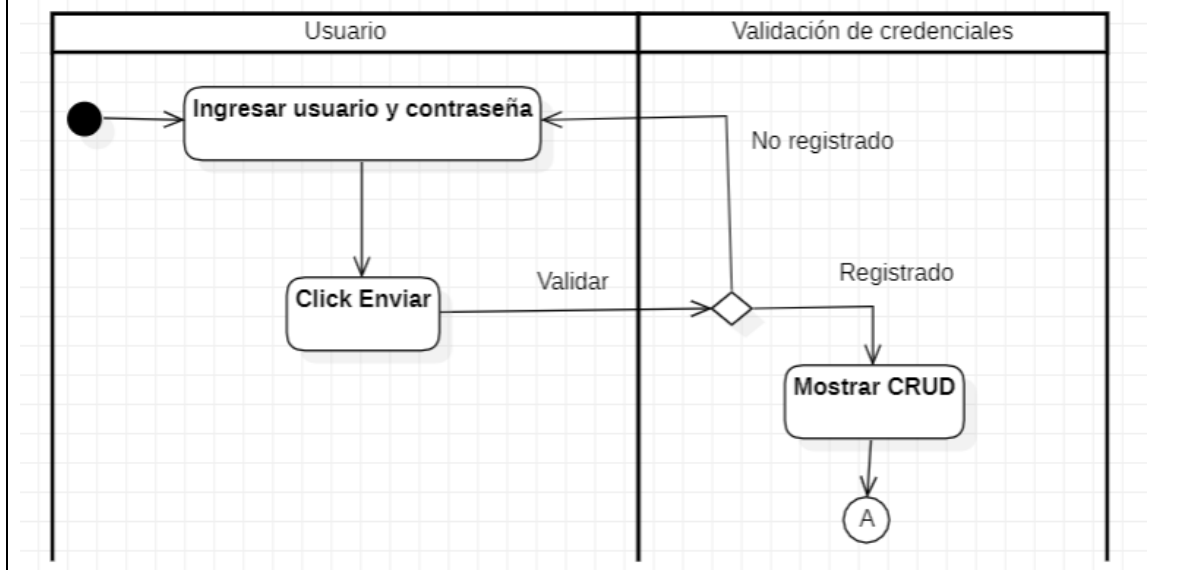


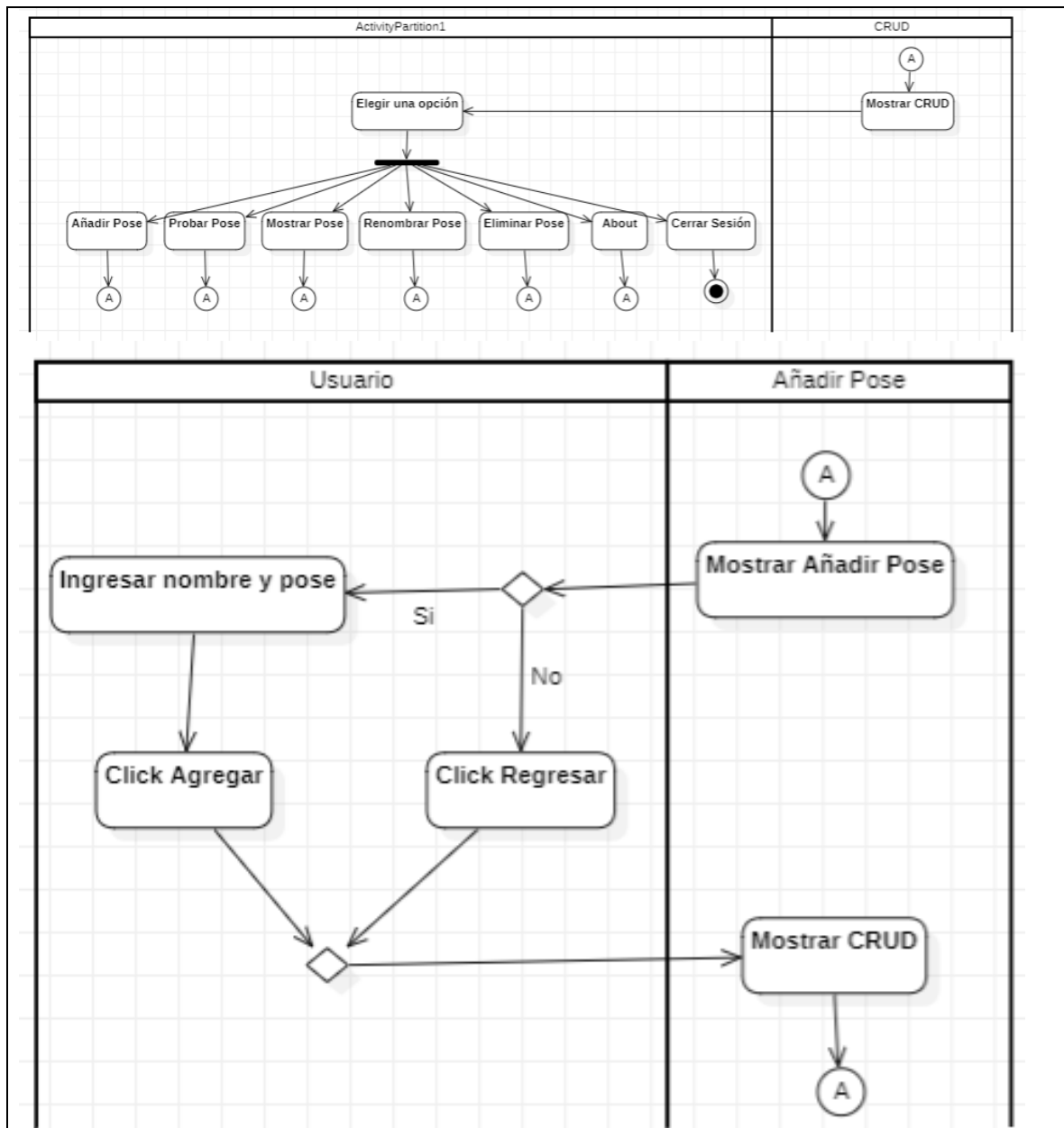


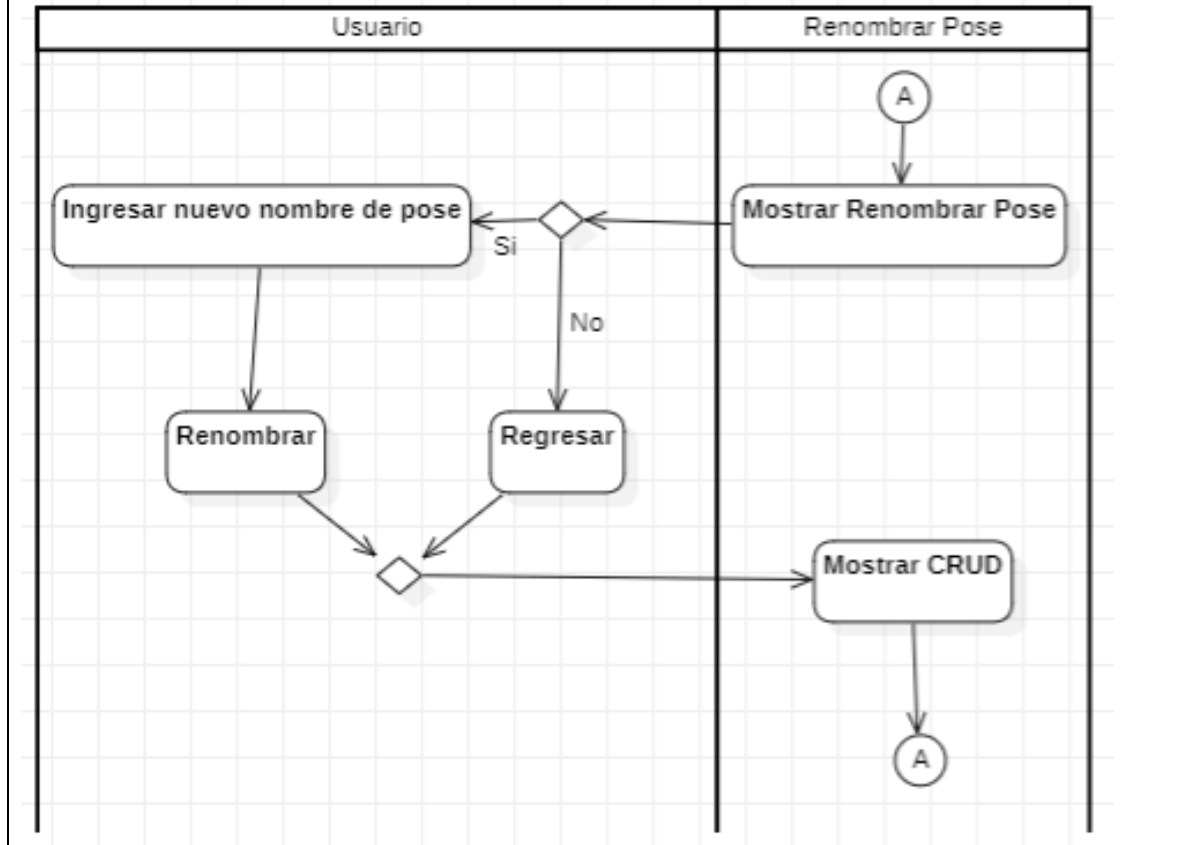
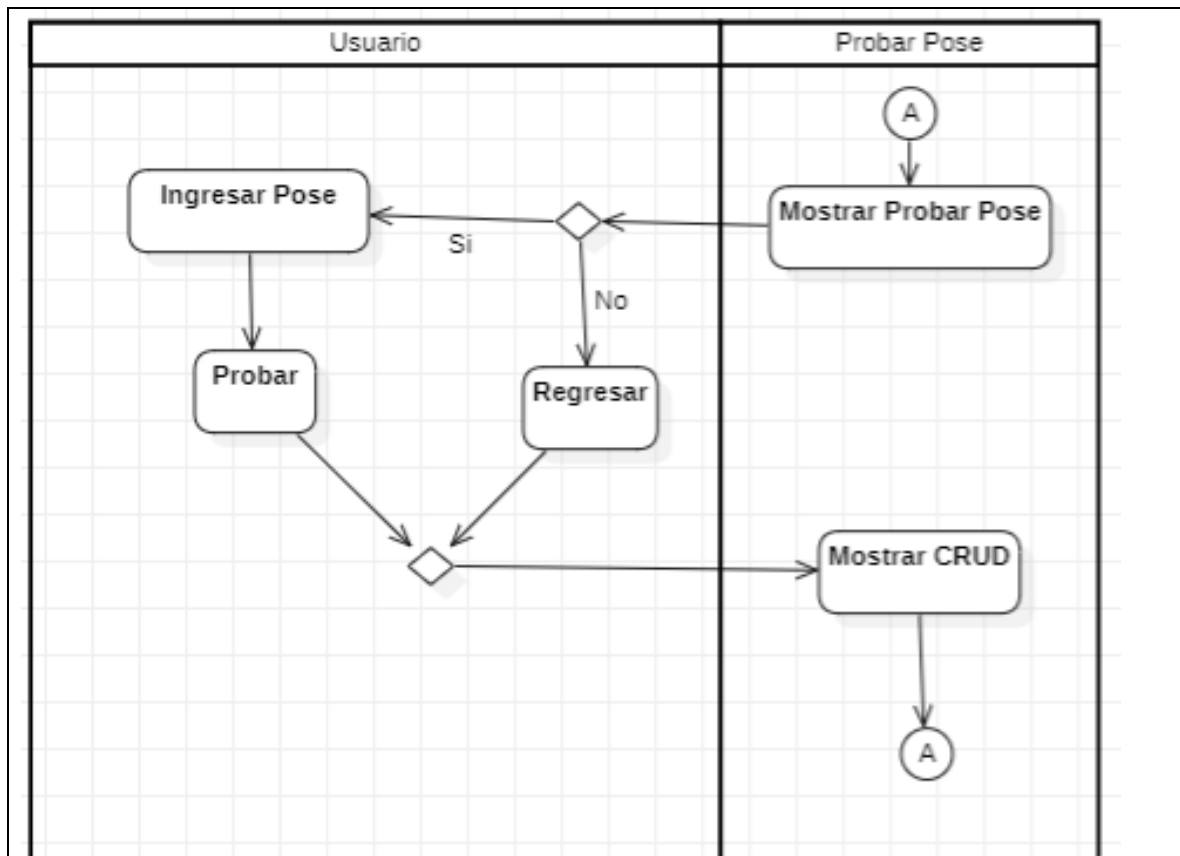


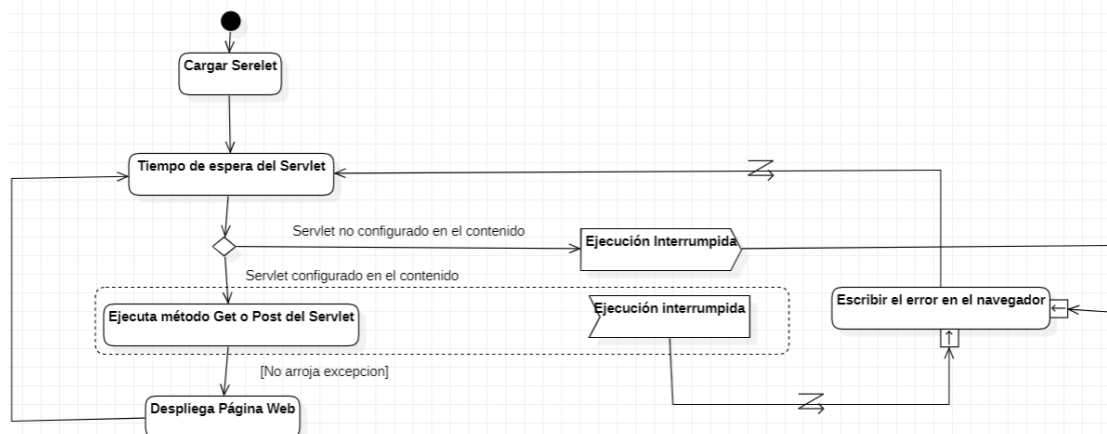
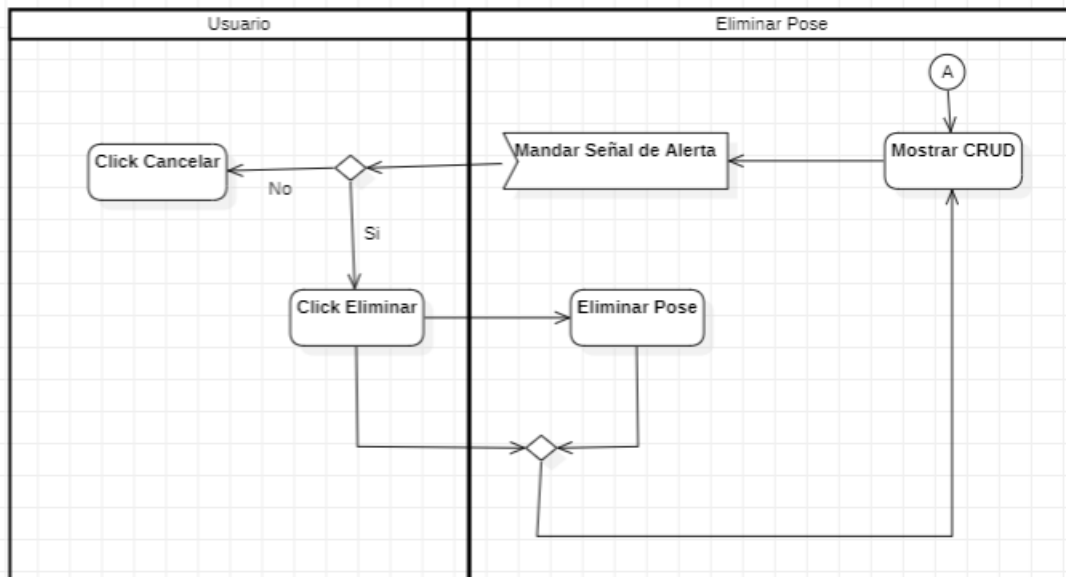
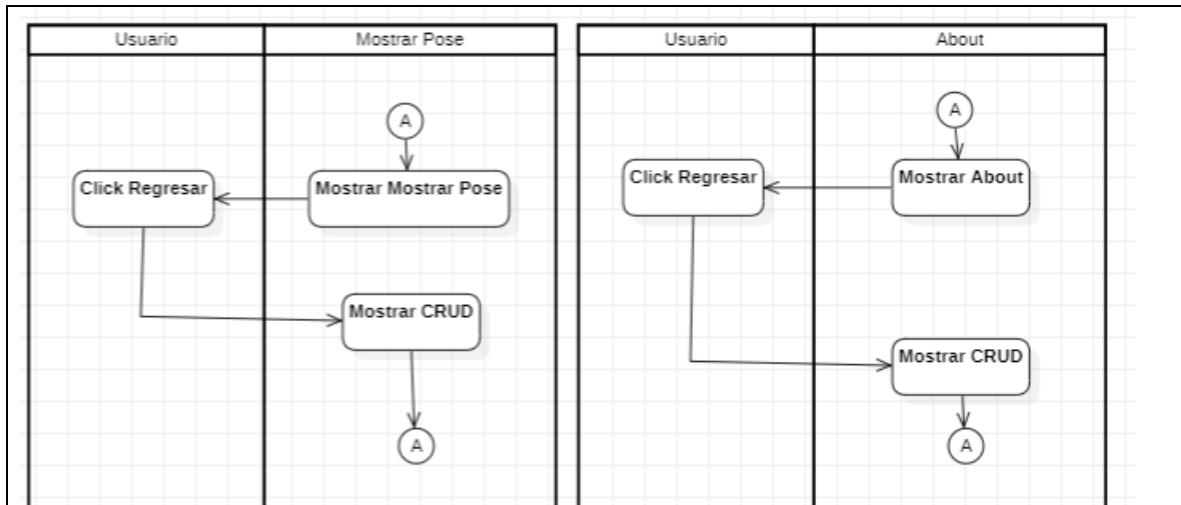
- Diagrama de actividades:

Los diagramas de actividad permiten que un equipo comprenda el mismo proceso y comportamiento. Así, el diagrama representa la lógica que tiene la aplicación y la descripción de los pasos realizados en el diagrama de casos de uso. La aplicación utilizó secciones, que se utilizan para representar o agrupar actividades realizadas por diferentes actores en un mismo hilo.



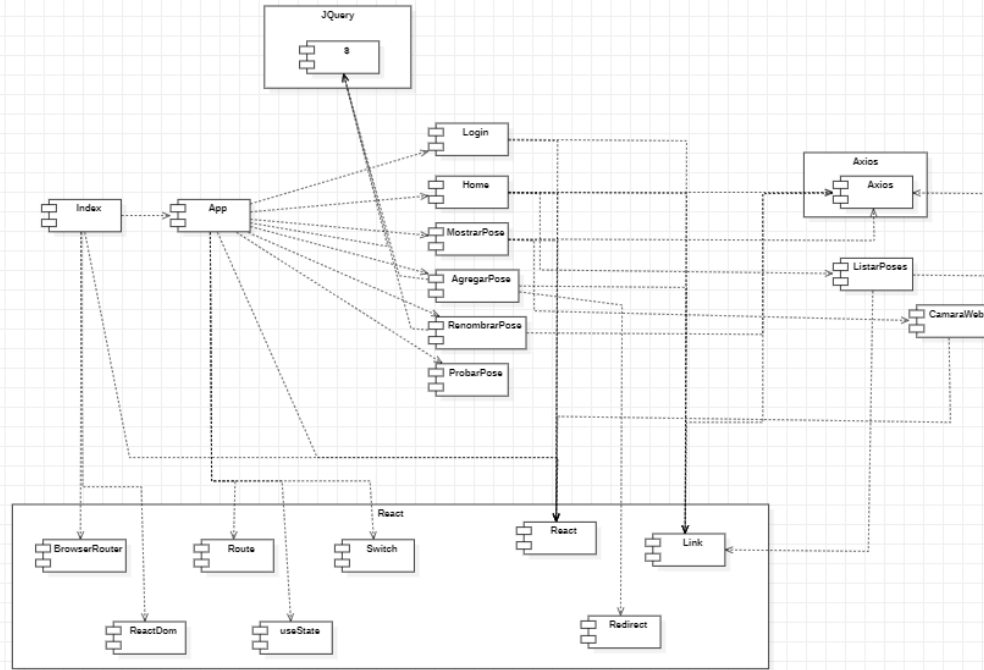






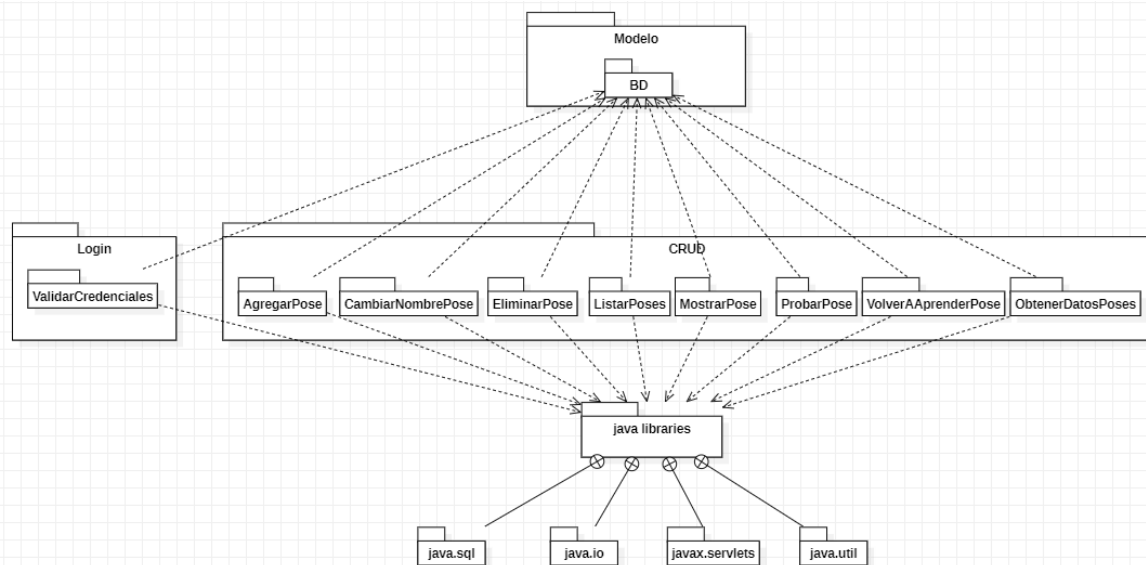
- Diagrama de componentes:

El diagrama de componentes se centra en el desarrollo de software orientado a componentes. El componente es un modulo ejecutable independiente que proporciona servicios a otros componentes o utiliza los servicios de otros componentes. El diagrama de componentes muestra la relación de cada elemento del frontend de la aplicación.



- Diagrama de paquetes:

El diagrama de paquete nos permite visualizar la distribución de clases dentro del servidor y su relación con las bibliotecas y otras clases. Se puede ver que todas las clases usan bibliotecas java y conexión a la base de datos.



Tablas de descripción de casos de uso

- Caso de uso validar credenciales

Nombre:	Validar credenciales
Descripción corta:	El sistema valida el ingreso del usuario con base a la entrada de datos y la correspondencia de estos con los registros de la base de datos.
Precondición:	Ninguna
Postcondición:	El sistema redirige al usuario a la vista principal donde mostrará las opciones de la aplicación que tiene el usuario.
Situaciones de error:	El usuario ingresa un dato erróneo.
Estado del Sistema en caso de error:	El sistema mostrará en pantalla un mensaje de error indicando que los datos ingresados son erróneos.
Actores:	Usuario
Disparadores:	El usuario ingresa a la aplicación mediante el sitio web.
Proceso estándar:	(1) Usuario ingresa al sitio web. (2) Usuario ingresa los datos de inicio de sesión. (3) El sistema valida el ingreso del usuario consultando la base de datos. (4) El sistema redirige al usuario a la vista principal.
Procesos alternativos:	(1) Usuario ingresa al sitio web. (2) Usuario ingresa datos erróneos de inicio de sesión. (3) El sistema determina que los datos de inicio de sesión no coinciden en la base de datos. (4) El sistema avisa al usuario que ha ingresado mal sus datos.

- Caso de uso Agregar una pose

Precondición:	El usuario debe haber iniciado sesión y no debe existir una pose con el mismo nombre en la base de datos.
Postcondición:	El sistema almacenará la nueva pose en la base de datos.
Situaciones de error:	El sistema no almacena la pose en la base de datos.
Estado del Sistema en caso de error:	El sistema desplegará un mensaje indicando que hubo un error al momento de almacenar la pose.
Actores:	Usuario
Disparadores:	El usuario selecciona la opción "Agregar una pose".
Proceso estándar:	(1) El usuario se encuentra en la vista principal. (2) El usuario selecciona la opción "Agregar una pose". (3) El sistema solicita al usuario un nombre para la nueva pose. (4) El usuario ingresa el nombre de la nueva pose. (5) El sistema solicita al usuario que realice la pose frente a la cámara para analizarla. (6) El usuario realiza la pose frente a la cámara web. (7) El sistema analiza la pose, valida que no exista una pose con el mismo nombre y almacena la nueva pose en la base de datos. (8) El sistema actualiza la vista principal.
Procesos alternativos:	(1) El usuario se encuentra en la vista principal. (2) El usuario selecciona la opción "Agregar una pose". (3) El sistema solicita al usuario un nombre para la nueva pose.

	<p>(4) El usuario ingresa el nombre de la nueva pose.</p> <p>(5) El sistema solicita al usuario que realice la pose frente a la cámara para analizarla.</p> <p>(6) El usuario realiza la pose frente a la cámara web.</p> <p>(7) El sistema determina que el nombre ya se ha usado para otra pose y/o que no se pudo almacenar la pose en la base de datos.</p> <p>(8) El sistema despliega un mensaje indicándole al usuario el error en cuestión.</p>
--	---

- Caso de uso Probar Pose

Nombre:	Probar pose
Descripción corta:	El sistema reconocerá una pose realizada por el usuario por medio de la WebCam.
Precondición:	El usuario debe haber iniciado sesión y debe existir al menos una pose en la base de datos.
Postcondición:	El sistema desplegará en pantalla el nombre de la pose que reconoce.
Situaciones de error:	El sistema no reconoce la pose o no se encuentra en la base de datos.
Estado del Sistema en caso de error:	El sistema mostrará que está analizando la pose, pero no mostrará el resultado del análisis.
Actores:	Usuario
Disparadores:	El usuario selecciona la opción "Probar pose" y realiza una pose.
Proceso estándar:	<p>(1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos.</p> <p>(2) El usuario selecciona la opción "Probar pose" de una pose.</p> <p>(3) El usuario realiza la pose frente a la cámara web.</p> <p>(4) El sistema reconoce la pose y despliega un mensaje con el nombre de dicha pose.</p>
Procesos alternativos:	<p>(1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos.</p> <p>(2) El usuario selecciona la opción "Probar pose" de una pose.</p> <p>(3) El usuario realiza la pose frente a la cámara web.</p> <p>(4) El sistema no reconoce la pose y despliega el mensaje "Analizando pose".</p>

- Renombrar Pose

Nombre:	Renombrar pose
Descripción corta:	El sistema le permite al usuario cambiar el nombre de la pose que este seleccione.
Precondición:	El usuario debe haber iniciado sesión y debe existir al menos una pose en la base de datos.
Postcondición:	El sistema mostrará el cambio en el nombre de la pose que el usuario indicó.
Situaciones de error:	El sistema no pudo cambiar el nombre de la pose.

Estado del Sistema en caso de error:	El sistema desplegará un mensaje indicando que hubo un error al momento de cambiar el nombre de la pose.
Actores:	Usuario
Disparadores:	El usuario selecciona la opción “Renombrar pose” y realiza una pose.
Proceso estándar:	<ol style="list-style-type: none"> (1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos. (2) El usuario selecciona la opción “Renombrar pose” de una pose. (3) El sistema le pide al usuario que indique el nuevo nombre para la pose. (4) El usuario ingresa el nuevo nombre. (5) El sistema valida el nombre con base a la base de datos. (6) El sistema realiza el cambio de nombre y actualiza la vista principal.
Procesos alternativos:	<ol style="list-style-type: none"> (1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos. (2) El usuario selecciona la opción “Renombrar pose” de una pose. (3) El sistema le pide al usuario que indique el nuevo nombre para la pose. (4) El usuario ingresa el nuevo nombre. (5) El sistema determina que no fue posible cambiar el nombre de la pose y despliega un mensaje en pantalla indicándoselo al usuario.

- Volver a aprender pose

Nombre:	Volver a aprender pose
Descripción corta:	El usuario le puede indicar al sistema que analice y vuelva a aprender una pose existente.
Precondición:	El usuario debe haber iniciado sesión y debe existir al menos una pose en la base de datos.
Postcondición:	El sistema actualizará la información necesaria en la base de datos para volver a aprender la pose.
Situaciones de error:	El sistema no pudo volver a aprender la pose.
Estado del Sistema en caso de error:	El sistema desplegará un mensaje indicando que hubo un error al momento de volver a aprender la pose.
Actores:	Usuario
Disparadores:	El usuario selecciona la opción “Volver a aprender pose” y realiza una pose.
Proceso estándar:	<ol style="list-style-type: none"> (1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos. (2) El usuario selecciona la opción “Volver a aprender pose” de una pose.

	<p>(3) El sistema le pide al usuario que realice la pose para que este pueda analizarla.</p> <p>(4) El sistema almacena los nuevos datos para reconocer la pose.</p> <p>(5) El sistema indica al usuario que se actualizaron las especificaciones de la pose.</p>
Procesos alternativos:	<p>(1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos.</p> <p>(2) El usuario selecciona la opción “Volver a aprender pose” de una pose.</p> <p>(3) El sistema le pide al usuario que realice la pose para que este pueda analizarla.</p> <p>(4) El sistema falla al almacenar los nuevos datos para reconocer la pose.</p> <p>(5) El sistema indica al usuario que hubo un error al momento de intentar actualizar las especificaciones de la pose.</p>

• Caso de uso eliminar pose

Nombre:	Eliminar pose
Descripción corta:	El sistema le permite al usuario eliminar una pose que se encuentre almacenada.
Precondición:	El usuario debe haber iniciado sesión y debe existir al menos una pose en la base de datos.
Postcondición:	El sistema eliminará la pose indicada de la base de datos.
Situaciones de error:	El sistema no pudo eliminar la pose indicada de la base de datos.
Estado del Sistema en caso de error:	El sistema desplegará un mensaje indicando que hubo un error al momento de intentar eliminar la pose.
Actores:	Usuario
Disparadores:	El usuario selecciona la opción “Eliminar pose” de una pose.
Proceso estándar:	<p>(1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos.</p> <p>(2) El usuario selecciona la opción “Eliminar pose” de una pose.</p> <p>(3) El sistema le pide al usuario que confirme la eliminación de la pose mediante.</p> <p>(4) El usuario confirma la eliminación de la pose.</p> <p>(5) El sistema elimina la pose de la base de datos y actualiza la vista principal.</p>
Procesos alternativos:	<p>(1) El usuario se encuentra en la vista principal y el sistema cuenta con al menos una pose en la base de datos.</p> <p>(2) El usuario selecciona la opción “Eliminar pose” de una pose.</p> <p>(3) El sistema le pide al usuario que confirme la eliminación de la pose mediante.</p> <p>(4) El usuario confirma la eliminación de la pose.</p> <p>(5) El sistema determina que no fue posible eliminar la pose y lo indica al usuario mediante un mensaje en pantalla.</p>

Metodología seleccionada

Metodología XP:

La metodología XP es un conjunto de técnicas que dan agilidad y flexibilidad en la gestión de proyectos. También es conocida como Programación Extrema (Extreme Programming) y se centra crear un producto según los requisitos exactos del cliente. De ahí, que le involucre al máximo durante el método de gestión del desarrollo del producto. [2]

Ventajas de la metodología:

- El proceso de programación es ordenado y logra un entendimiento general del código entre los desarrolladores del software.
- La tasa de errores baja en el producto final
- Constante comunicación entre todos los integrantes del equipo de desarrollo.

Desventajas de la metodología

- Esta metodología es recomendable sólo en proyectos pequeños, no se recomienda para proyectos de tipo escolar o pequeños proyectos.
- Se pueden percibir una gran cantidad de comisiones en caso de que el producto final falle o no funcione de forma satisfactoria para los clientes.

Diagrama del Ciclo de vida de la metodología



[2]

Descripción del Ciclo de vida de la metodología

El ciclo de vida de esta metodología se basa en la comunicación que tenga el equipo de desarrollo con los clientes y los usuarios, debido a que estos son los individuos que proporcionarán los parámetros de funcionalidad del proyecto, desde la manera en cómo se comportan, hasta las posibles fallas que un usuario quisiera evitar en un software basado en sus experiencias previas, es por ello que el ciclo de

vida comienza en la etapa de planificación considerando las historias de los usuarios y elaborando un plan de las posibles iteraciones que podrían llevarse a cabo a lo largo de las siguientes fases.

Principios de la metodología:

- Se requiere de una constante comunicación entre el cliente y el equipo de desarrollo durante la realización del proyecto.
- Implica una respuesta rápida a los cambios constantes solicitados por el cliente.
- Planificación abierta a un cronograma de actividades flexible.
- Se le da prioridad al desarrollo del software por encima de a cualquier documentación relacionada con el mismo.
- Se prioriza cumplir con los requisitos del cliente para la entrega final del proyecto

Roles en la metodología:

- Cliente: Define los objetivos del proyecto
- Programadores: Planificarán el proyecto con base en los requisitos de los clientes
- Testers: Encargados de pruebas, mantienen comunicación directa con los clientes
- Tracker (encargado de seguimiento): Define el control y la planificación del proyecto en función de los objetivos del cliente y los tiempos de ejecución de las labores de los programadores.
- Coach: Asesoramiento y orientación continuo para el equipo de trabajo y los clientes.
- Manager: Responsable de coordinar la comunicación entre los miembros del equipo de trabajo.

Descripción de las Fases de la metodología:

- Planificación: Fase de análisis de las historias de los usuarios y planificación de entregas e iteraciones del proyecto
- Diseño: Codificación del sistema con el objetivo de dar solución a las peticiones de los usuarios
- Codificación: Programación en pares del proyecto y sus iteraciones
- Pruebas: Pruebas unitarias basadas en la detección y corrección de errores
- Lanzamiento: Lanzamiento al público en general del proyecto.

Fase de la metodología	Tareas de la Fase de la metodología	Entregables de la Fase de la metodología
Planificación	Identifica historias de usuario, se prioriza y descompone en mini.versiones, se revisan cada dos semanas por iteración para obtener un software útil, funcional y listo para probar y lanzar	Se entrega el plan de entrega y plan de iteraciones que serán empleados para las fases siguientes considerando la información proporcionada por los usuarios.
Diseño	Se intenta trabajar con un código sencillo, (lo mínimo imprescindible para que funcione) Se obtendrá el prototipo. Además, para el	Se obtiene un primer prototipo del proyecto que aborda las especificaciones de la lista funcional.

	diseño del software orientado a objetos, se crearán tarjetas CRC (Clase-Responsabilidad-Colaboración).	
Codificación	Se programa en parejas en frente del mismo ordenador. Incluso, a veces se intercambian las parejas. Así se asegura que se realice un código más universal, con el que cualquier otro programador podría trabajar y entender. Y es que deber parecer que ha sido realizado por una única persona. Así se conseguirá una programación organizada y planificada.	Entrega de prototipos
Pruebas	Se deben realizar pruebas automáticas continuamente. Al tratarse normalmente de proyectos a corto plazo, este testeo automatizado y constante es clave. Además, el propio cliente puede hacer pruebas, proponer nuevas pruebas e ir validando las mini-versiones.	Entrega de las iteraciones con las correcciones sugeridas por los distintos miembros del equipo de desarrollo o del cliente para pasar las pruebas de aceptación que se le hagan al software y sus iteraciones.
Lanzamiento	Si hemos llegado a este punto, significa que hemos probado todas las historias de usuario o mini-versiones con éxito, ajustándonos a los requerimientos de los clientes. Tenemos un software útil y podemos incorporarlo en el producto.	Software terminado e incluido a la entrega del producto final

Requerimientos funcionales y no funcionales	
IDENTIFICADOR RF	DESCRIPCIÓN
RF1	Validar credenciales: El sistema valida las credenciales del usuario de acuerdo con la base de datos permitiéndole o negándole el acceso al sistema.
RF2	Agregar una pose: El sistema le permite al usuario agregar una nueva pose al registro.
RF3	Probar pose: El sistema analizará una pose realizada por el usuario por medio de la cámara web y verificará si el usuario la está realizando correctamente.
RF4	Renombrar pose: El sistema le permite al usuario cambiar el nombre de la pose que este seleccione.
RF5	Volver a aprender pose: El usuario le puede indicar al sistema que analice y vuelva a aprender una pose existente.
RF6	Eliminar pose: El sistema le permite al usuario eliminar una pose que se encuentre almacenada.
Tabla 1. Requerimientos Funcionales.	
IDENTIFICADOR RNF	DESCRIPCION
RNF 1	El tiempo de ejecución de la operación seleccionada por el usuario debe ser inferior a 5 segundos.
RNF 2	La interfaz debe contar con una presentación sencilla y agradable para los usuarios, además de tener un ambiente responsivo para diferentes tamaños de pantalla.
RNF 3	El sistema debe disponer de una buena documentación que permita realizar funciones de mantenimiento con el menor esfuerzo posible.
RNF 4	El sistema debe funcionar con la última versión de Google Chrome y de Mozilla Firefox
RNF 5	La interfaz debe disponer de un manual de instalación y de un manual de usuario.
Tabla 2. Requerimientos no funcionales	

Conclusiones
<ul style="list-style-type: none"> • Conclusión de la metodología seleccionada. <p>Seleccionar una metodología fue un paso que podría considerarse sencillo solo tenemos que seguir los lineamientos y estándares que esta misma nos impone, pero resulta ser uno de los pasos mas importantes del proyecto, pues esta dictara la manera en la que se va a trabajar a lo largo de todo el proyecto. La metodología que nosotros seleccionamos para el desarrollo de este proyecto fue la de programación extrema que en nuestra opinión como equipo resulto efectiva, pero bastante apresurada además de laboriosa debido a las constantes pruebas realizadas para la comprobación del correcto funcionamiento de todos los elementos del proyecto, cosa que también podría retrasar los resultados finales esperados si es que algún elemento terminaba fallando y tenias que detenernos, corregirlo y posteriormente pasar a la siguiente fase.</p>

Para concluir podemos decir que esta metodología si nos fue útil para poder terminar el proyecto a tiempo, pero no recomendaría usarla para proyectos con tiempo limitado como este.

- **Conclusión de los requerimientos funcionales y no funcionales.**

Los requerimientos funcionales podríamos describirlos como las cosas que es necesario que haga nuestra aplicación para que pueda funcionar apropiadamente y sea útil en la actividad que queremos desarrollar, por ejemplo, era indispensable para nuestra aplicación que se accediera a la base de datos pues sin ella todas las acciones que se llevan a cabo con las pose no las podríamos realizar debido a la falta de datos. Por otro lado, los requerimientos no funcionales son la descripción ideal de lo que queremos que realice nuestra aplicación, aunque muchas veces no terminemos por concluir esos aspectos al final del proyecto, un ejemplo sería que se pudiera desplegar y ejecutar en dispositivos móvil de manera correcta. Para concluir podemos decir que los requerimientos funcionales y no funcionales nos sirven para poder describir y analizar como queremos que funcione nuestra aplicación y como queremos que sea diseñada sin duda es paso muy importante a la hora de empezar un proyecto

- **Conclusión de diagramas UML.**

Los diagramas son una de las partes mas importantes que desarrollamos de todo el proyecto, cada diagrama se encarga de atacar aspectos muy puntuales del diseño y desarrollo de la aplicación , pero en conjunto se puede ver de manera grafica y explicita el como esta constituido todo el proyecto, desde los paquetes que contienen todos los archivos que componen su funcionamiento y como se relacionan explicado con el diagrama de paquetes hasta el cómo el usuario podría llegar a interactuar con las diferentes funcionalidades, pasando por cómo se desarrollan las actividades en el diagrama de secuencia, y a que niveles de acceso tendrá cada usuario en el diagrama de casos de uso, por eso consideramos que los diagramas no solo son importantes para el desarrollo de proyectos de software, más bien diríamos que son indispensables para realizar una correcto desarrollo e implementación de las actividades y funciones de dicha aplicación.

- **Conclusión de tablas de casos de uso.**

Los casos de uso nos ayudan a estructurar el como van a funcionar todos y cada uno de los procesos que tendrá incorporados nuestra aplicación, en nuestra opinión como equipo esta fase es de suma importancia debido a que con esto llevaremos a cabo el diseño del funcionamiento básico de nuestra aplicación y el cómo los usuarios podrán interactuar con ellas, así podremos asegurarnos de que tengamos un entorno de desarrollo que sea fácil de entender y saber como cada parte de el proyecto interactúa y que procesos mandar a llamar en con cada acción

- **Conclusión del manual de instalación**

Para este requerimiento, fue posible una mejor comprensión de los elementos que componen la aplicación de nuestro proyecto. El comprender cómo funciona la aplicación, los elementos que componen un funcionamiento mínimo y un funcionamiento óptimo, nos permitió visualizar diferentes escenarios en los cuales se analizó la manera más concreta de utilizar y distribuir nuestro proyecto para que pudiera ser ejecutado.

También se comprendió que, por la complejidad y temprano desarrollo del proyecto, se hizo el manual considerando que se tienen los programas y paquetes previamente instalados. Por lo que solo se vio necesario explicar la instalación de MySQL correctamente puesto que en este

proyecto se requiere que se ejecute el script de la BD por lo extenso que es al momento de la recopilación de datos que hace la red neuronal. Por lo que el hacer el manual de instalación, nos abrió el panorama más allá de lo que se tenía planeado.

- **Conclusión del manual de usuario**

El realizar un manual de usuario, nos hace conscientes a nosotros como desarrolladores que los usuarios requieren este tipo de documentación para saber y comprender el cómo se usa correctamente la aplicación y cómo se comporta respecto a diferentes escenarios. El hacer un manual de usuario, es enseñar de una forma concreta a una persona a usar y aprovechar las funcionalidades de nuestra aplicación que, aunque para nosotros como desarrolladores es sencillo e intuitivo por la interfaz realizada, debemos considerar que el manejo de algo nuevo siempre resultará desafiante, y el manual de usuario es una herramienta para que sea más sencillo y rápido el aprendizaje del manejo de la aplicación.

- **Conclusión general del proyecto.**

La realización de este proyecto a lo largo del semestre trajo con nosotros mucho aprendizaje, no solo técnico con el modelado UML o con la elaboración de toda la programación de la aplicación sino también personal al aprender a como trabajar en equipo a través del proyecto y a aprender a utilizar las diversas herramientas que nos ofrece el desarrollo de aplicación, podemos decir como equipo que este proyecto nos ha acercado más a ser los profesionales que algún día seremos y poder ofrecer nuestras capacidades y talentos al mundo laboral.