



# “Aplicación Android para controlar el software ATcnea”

Trabajo de diploma para optar por el título de Ingeniero en  
Ciencias Informáticas

Autores: Maylin Torres Rodríguez

Oreste A. Nillar Cambara

Tutor (es): MSc. Iván Pérez Mallea, Prof. Auxiliar

Ing. Guillermo González Jiménez

La Habana, junio de 2017

“Año 59 de la Revolución”

# *Declaración de Autoría*

## **Declaración de Autoría**

Declaramos ser los autores del presente trabajo de diploma y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Maylin Torres Rodríguez

---

Oreste A. Nillar Cambara

---

MSc. Iván Pérez Mallea

---

Ing. Guillermo González Jiménez



*“La educación es el arma más poderosa que puedes usar para cambiar el mundo.”*

*Nelson Mandela*

## *Dedicatoria*

*Dedico el resultado de esta tesis de manera general a mi familia, en especial a mis padres por creer en mí y darme todo su apoyo y por ser mi mayor inspiración.*

*Maylin Torres Rodríguez*

*Dedico el resultado de esta tesis a mi familia, en especial a mis padres por darme todo su amor y brindarme la posibilidad de superarme.*

*Oreste A. Nillar Cambara*

## *Agradecimientos*

*A mis padres por ser mis mejores amigos, mis consejeros y mi ejemplo a seguir. Gracias por siempre estar ahí para mí.*

*A mis abuelos Nancy y Hevia por darme siempre su amor y cariño.*

*A mis hermanas, en especial mi hermana Maye y mi cuñado Airam que sé, que les hubiera gustado estar aquí a mi lado compartiendo este triunfo.*

*A mi tío Manito, mi tía Nieves y mi tío Beiler por siempre estar a mi lado apoyándome y que decir de mis primos, Lisset, Leannet, Ana Laura, que más que primos son mis hermanos.*

*A mis amigos que siempre los llevaré conmigo y serán parte de mi familia en especial: Noralis, Malidia, Jorgito, Henry, Yuliet, Bel, Keimis y a los que en el transcurso de la UCI se han ganado mi cariño y admiración.*

*A los tutores por dedicarnos parte de su tiempo para que todo este trabajo saliera con la mejor calidad posible.*

*Maylín Torres Rodríguez*

## *Agradecimientos*

*A mis padres por ser mi apoyo incondicional, mi estrella guía, los verdaderos autores de la persona en la que me he convertido.*

*A mi abuela Gloria que cariñosamente llamamos yoya, gracias por cumplir cada uno de mis caprichos.*

*A mis abuelos Abel y Anabel por expresarme en todo momento su amor y cariño.*

*A mi hermana Esther por ser la mejor hermana del mundo y darme la felicidad de ser tío.*

*A mis tíos Alexy y Adaimis por estar siempre presente, y mis primos por todas las travesuras realizadas.*

*A mi novia Naxy, por hacer mis días los más lindos en mucho tiempo y convertirse en la chica que ha robado mi corazón.*

*A mis amigos que tanto quiero y se han convertido en parte de mi familia, en especial: Raciél, Danilo, Danger, David, Daimel, Guillermo, Henry, el flaco, Humberto e Israel.*

*A los tutores por dedicarnos parte de su tiempo para que todo este trabajo saliera con la mejor calidad posible.*

*Oreste A. Nillar Cambara*

A medida que las necesidades educativas han evolucionado impactadas por la tecnología, ha aumentado la disponibilidad de recursos como software y aplicaciones para apoyar el proceso de enseñanza-aprendizaje. Un ejemplo de esto lo constituyen las Aulas Tecnológicas las cuales se presentan como una solución educativa contemporánea. Cuba no se ha quedado rezagada en este aspecto, por lo que actualmente se desarrolla en la Universidad de las Ciencias Informáticas el software ATcnea perteneciente a la primera aula tecnológica cubana haciendo uso de las tecnologías actuales. A pesar de los beneficios que el mismo brinda para el proceso de enseñanza-aprendizaje, no es capaz de permitir la movilidad del profesor por el aula para atender las individualidades de los estudiantes sin perder el control y monitoreo de la clase. El presente trabajo tiene como objetivo desarrollar una aplicación para dispositivos móviles con sistema operativo Android que permita interactuar con el software ATcnea. Para llevar a cabo la investigación se realizó el estudio de los referentes teóricos asociados a los sistemas de gestión de aulas, destacándose las principales características del software a desarrollar. Además, se hace un acercamiento a las tecnologías utilizadas para el desarrollo del mismo, así como pruebas al software con el propósito de verificar su correcto funcionamiento. Finalmente se logró una aplicación móvil para dispositivos con sistema operativo Android, que permite al profesor obtener una mayor movilidad en el aula para atender las individualidades de los estudiantes sin perder el control y monitoreo del resto de la clase.

**Palabras claves:** aula tecnológica, Android, movilidad, control y monitoreo.

## Índice

<b>Introducción.....</b>	<b>1</b>
<b>Capítulo 1: Fundamentación teórica.....</b>	<b>7</b>
1.1 Introducción.....	7
1.2 Conceptos asociados al dominio del problema.....	7
1.3 Estudio de aplicaciones similares.....	12
1.4 Sistema de gestión de aula XAUCE ATcnea.....	14
1.5 Metodología de desarrollo de software.....	16
1.6 Herramientas y tecnologías.....	19
1.6.1 Lenguaje de modelado.....	19
1.6.2 Herramienta de modelado.....	20
1.6.3 Lenguaje de programación.....	20
1.6.4 Herramienta de desarrollo.....	21
1.6.5 Herramienta de control de versiones.....	21
1.6.6 Herramientas de pruebas.....	22
1.7 Conclusiones parciales.....	22
<b>Capítulo 2: Descripción de la solución propuesta.....</b>	<b>23</b>
2.1 Introducción.....	23
2.2 Descripción del sistema.....	23
2.3 Modelo de dominio.....	24
2.3.1 Diagrama de clases del modelo del dominio.....	24
2.3.2 Definición de las clases del modelo del dominio.....	24
2.4 Especificación de los requisitos.....	25
2.4.1 Requisitos funcionales.....	25
2.4.2 Requisitos no funcionales.....	29
2.5 Historias de usuarios.....	29
2.6 Patrón arquitectónico.....	31
2.7 Patrones de diseños.....	33

2.7.2 Diagramas de clases del diseño (DCD) .....	36
2.7.3 Diagrama de secuencia del diseño (DSD).....	37
2.8 Conclusiones parciales .....	38
<b>Capítulo 3: Implementación y prueba de la solución propuesta.....</b>	<b>39</b>
3.1 Introducción.....	39
3.2 Implementación .....	39
3.2.1 Adecuaciones realizadas sobre la aplicación de escritorio de ATcnea .....	39
3.2.2 Estándares de codificación .....	40
3.3 Modelo de despliegue .....	41
3.3.1 Diagrama de despliegue (DD) .....	42
3.4 Pruebas de software .....	42
3.4.1 Pruebas unitarias .....	43
3.4.2 Pruebas funcionales .....	43
3.4.3 Pruebas de aceptación .....	45
3.4.4 Resultados de las pruebas:.....	45
3.5 Conclusiones parciales .....	50
<b>Conclusiones generales.....</b>	<b>51</b>
<b>Recomendaciones.....</b>	<b>52</b>
<b>Referencias.....</b>	<b>53</b>
<b>Anexos.....</b>	<b>56</b>

## Introducción

El proceso de enseñanza-aprendizaje desde sus comienzos hasta la actualidad ha sido, de los procesos naturales de la humanidad, que más transformaciones ha sufrido. Es innegable que estas transformaciones han tenido mayor impacto gracias a la introducción de las tecnologías en dicho proceso. Hablar de una educación moderna y de calidad sin tecnologías es algo absurdo.

La educación es la socialización de las personas a través de la enseñanza, la adquisición de un conjunto de conductas para dar respuesta, de acuerdo con las oportunidades, a las exigencias que se plantean en cada situación. Se trata de que el educando adquiera conocimientos, actitudes, destrezas, habilidades y hábitos que lo capacitan para decidir y realizar sus proyectos (1), buscando que el individuo adquiera ciertos conocimientos que son esenciales para su interacción social y para su desarrollo en el marco de una sociedad (2). En la actualidad se habla de paradigmas de la enseñanza, es común encontrar en la literatura un paradigma tradicional y un nuevo paradigma apoyado este último por las Tecnologías de la Información y las Comunicaciones (TIC).

El paradigma de enseñanza tradicional se centra en el profesor, en el contenido que este debe transmitir a sus alumnos y en el establecimiento de un ambiente que permita la trasmisión de este contenido de manera única, nueva e irrepetible para ellos, en un lugar específico y en un momento determinado. En este ambiente los alumnos deben acudir preparados de manera consciente con aquellos medios que le permitan escuchar atentamente, ejecutar las acciones que el profesor les indique, copiar y aprender a reproducir, fundamentalmente en su memoria, los nuevos contenidos que se le brinden. (3)

Con el tiempo la interacción entre los alumnos generó espacios de participación en las aulas, de esa manera la comunicación unilateral fue reemplazada por el proceso bilateral en el cual el maestro además de transmitir sus conocimientos, también recibe las ideas y el aprendizaje de los estudiantes. En estos espacios la enseñanza se produce de ambos lados (maestro-alumno) modificando el papel del docente, donde el trabajo en equipo permite el intercambio

de ideas. Este escenario propició la aparición de un paradigma centrado en el estudiante y no en el profesor, donde este último aún guía el proceso de enseñanza-aprendizaje, pero cambia su rol para gestionar aprendizajes personalizados y crear recursos en función de sus estudiantes.

El papel de las TIC en este nuevo paradigma es relevante, han dado surgimiento a novedosas formas de comunicación que han revolucionado las diferentes modalidades de enseñanza. Se han desarrollado plataformas educativas y otras herramientas con el objetivo de apoyar la educación presencial y semi-presencial, así como para dar soporte a una reinventada educación totalmente a distancia.

A lo largo de este proceso las TIC han evolucionado en sí mismas, hoy se habla de un mundo digital y se dice que los jóvenes son nativos digitales; desde que nacen tienen un contacto directo con la tecnología haciéndose esta parte cotidiana de su vida y por ende aprender a través de ellas es visto de manera natural (4). De aquí que dotar a las aulas con la tecnología para que los estudiantes aprendan desde temprana edad es una idea que cobra cada vez mayor fuerza en el mundo dando a luz a las llamadas aulas interactivas.

Las aulas interactivas o aulas digitales, son una realidad actual que constituye una solución educativa contemporánea para el método de enseñanza-aprendizaje, brindando una experiencia única en el aula. Surge producto de la aplicación de las nuevas tendencias educativas y su vinculación con las tecnologías. El objetivo de las mismas es la creación de un ambiente colaborativo, que propicia la introducción de tecnologías como medios, parte de la didáctica y es una forma de enriquecimiento del contenido académico de la figura docente de que se trate. Permite a profesores y educandos establecer una profunda comunicación, cuya interactividad en el intercambio de ideas e información y formas de colaboración, motivará la participación y profundización en los temas objeto de estudio. (5)

Cuba no se encuentra ajena a estas transformaciones educacionales y ha sido vanguardia en la aplicación de las mismas. Desde el 2014 se evalúan por parte del Ministerio de Educación (MINED) algunas variantes de aulas tecnológicas de diversos fabricantes con la intención de introducirlas en los primeros niveles de la educación. (6)

Durante el curso 2015-2016 comenzó en el Centro de Tecnologías para la Formación (FORTES) de la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI), el desarrollo de un software para la gestión de una clase en un aula con el empleo de las TIC. Este software denominado ATcnea, se ha desarrollado sobre tecnología HAIER proporcionada por la Empresa Industrial para la Informática, las Comunicaciones y la Electrónica (GEDEME) y con la colaboración de soluciones brindadas por el Centro de Soluciones Libres (CESOL) de la Facultad 1 de la UCI.

El objetivo fundamental de este proyecto es desarrollar el software de un aula tecnológica utilizando tecnologías libres. Consiste en dos aplicaciones informáticas con fines educativos, una de escritorio alojada en una laptop que utiliza el profesor y una con sistema operativo Android alojada en las tabletas de los estudiantes. Entre ambas existe una comunicación para que el profesor, desde su laptop, pueda guiar el proceso de enseñanza-aprendizaje usando los medios que proveen las aulas tecnológicas y los estudiantes desde las tabletas puedan recibir e interactuar con el contenido. En junio del 2016 se obtiene la versión beta del producto ATcnea y es presentada en la feria CUBAINDUSTRIA, donde tuvo gran aceptación y se identificaron nuevas necesidades.

El esquema pensado para el aula tecnológica XAUCE ATcnea implica al profesor en su laptop, al frente de un grupo de estudiantes que acceden e interactúan a través de sus tabletas. Este esquema desde el punto de vista de hardware limita la movilidad del profesor para atender las individualidades de los estudiantes sin perder el control y monitoreo de la clase. El profesor normalmente se encuentra en la parte delantera del aula donde está la laptop con el software y la pizarra interactiva. Si el profesor necesitara ir hasta el puesto de un estudiante en el fondo del aula para atender alguna duda o chequear los ritmos de aprendizaje, debería alejarse de la laptop y dejar desatendido el resto del control de la clase que se realiza a través de ATcnea.

Teniendo en cuenta lo antes planteado se define como **problema de la investigación**:

¿Cómo permitir una mayor movilidad del profesor para atender las individualidades de los estudiantes sin perder el control y monitoreo de la clase?

Por lo que se define como **objeto de estudio** los sistemas de gestión de aulas, enmarcado en el **campo de acción** las aplicaciones móviles para la interacción con sistemas de gestión de aulas.

Para dar solución al problema de investigación planteado se define como **objetivo general** desarrollar una aplicación para dispositivos móviles con sistema operativo Android, que brinde al profesor mayor movilidad en el aula sin perder el control y monitoreo de la clase.

A partir del objetivo general se desglosa en los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico referencial de la investigación.
- ✓ Desarrollar la solución, tanto la aplicación para móviles como las adecuaciones en el núcleo de ATcnea.
- ✓ Realizar pruebas al software como método de validación de las funcionalidades del sistema.

Para dar cumplimiento a los objetivos propuestos se define las siguientes **tareas de investigación**:

- ✓ Estudio de las soluciones similares.
- ✓ Análisis y selección de la metodología, herramientas y tecnologías.
- ✓ Identificación y especificación de los requisitos funcionales y no funcionales.
- ✓ Descripción de las funcionalidades y las características del sistema.
- ✓ Diseño de prototipo de interfaces.
- ✓ Diseño del sistema.
- ✓ Implementación de las funcionalidades del sistema.
- ✓ Realización de las pruebas unitarias, funcionales y de aceptación.

Para el cumplimiento del objetivo general planteado se utilizaron los siguientes métodos de investigación:

## **Métodos teóricos**

- ✓ Analítico-sintético: Se utiliza para el análisis de teorías, documentos y materiales relacionados con el desarrollo de aplicaciones para aulas tecnológicas arribando a conclusiones que sustenten la necesidad de la investigación.
- ✓ Análisis histórico-lógico: Se utiliza para analizar la evolución histórica de soluciones similares, para el estudio de su comportamiento y la identificación de necesidades para adecuar la propuesta a desarrollar al marco actual de la aplicación.

## **Método empírico**

- ✓ Observación: Mediante este método se tomaron experiencias del desarrollo de una clase común, acerca del desempeño del estudiante y su rol dentro de la clase para poder adaptar a la aplicación, un reflejo de su comportamiento.

El presente trabajo posee la siguiente estructura:

## **Capítulo I: Fundamentación teórica**

En el presente capítulo se realiza un análisis de los conceptos fundamentales relacionados con el objeto de estudio. Se lleva a cabo un estudio de soluciones similares existentes a nivel internacional. Además, se realiza el análisis de las distintas herramientas, tecnologías y la metodología a utilizar para el desarrollo de la aplicación.

## **Capítulo II: Descripción de la solución propuesta**

En el presente capítulo se describe la propuesta de solución del sistema y se presentaron los artefactos generados que facilitan al desarrollador entender las funcionalidades y garantizar una implementación exitosa. Se representa el diagrama del Modelo del dominio con la correspondiente descripción de cada uno de sus componentes. Se capturan los requisitos funcionales y no funcionales que debe cumplir el sistema y se conforman las historias de usuarios correspondientes a los mismos. Además, se describe la arquitectura de la aplicación, así como los patrones de diseño a aplicar.

## **Capítulo III: Implementación y prueba**

En el presente capítulo se exponen las especificaciones asociadas a la implementación de la aplicación. Se describen las pautas de codificación. Se presenta el diagrama de despliegue correspondiente a la propuesta de solución y un resumen de las pruebas realizadas al software para garantizar su correcto funcionamiento.

# Capítulo 1: Fundamentación teórica

## Capítulo 1: Fundamentación teórica

### 1.1 Introducción

En el presente capítulo se realiza un análisis de los conceptos fundamentales relacionados con el objeto de estudio. Se lleva a cabo un estudio de soluciones similares existentes a nivel internacional. Además, se realiza el análisis de las distintas herramientas, tecnologías y la metodología a utilizar para el desarrollo de la aplicación.

### 1.2 Conceptos asociados al dominio del problema

Para una mejor comprensión sobre la investigación realizada, se necesitan conocer los siguientes conceptos:

#### **Aula digital**

Aula en la que se encuentran los recursos tecnológicos necesarios para posibilitar el acceso a la información y la comunicación que nos proporciona la red y los sistemas de presentación multimedia.

Según María Josefina Vidal, las aulas digitales revolucionan el método de enseñanza-aprendizaje, brindando una experiencia única en el aula. Su principal objetivo es la creación de un ambiente colaborativo, donde la tecnología enriquece el contenido académico de cada asignatura y permite al maestro-alumno establecer una amplia comunicación interactiva, que motivará a los alumnos hacia el estudio y mejorará su nivel académico. (5)

Otros autores plantean que el aula digital es una solución educativa concebida para transformar la enseñanza y el aprendizaje. Está estructurada por un dispositivo móvil para cada estudiante, una pizarra interactiva, una laptop para el profesor y un software que permita la interacción entre todos los dispositivos mencionados. Favorece el desarrollo de capacidades y participación de los estudiantes. (7)

#### **Ambiente colaborativo**

Son procesos intencionales de un grupo para alcanzar objetivos específicos, con herramientas diseñadas para dar soporte y facilitar el trabajo, logrando la motivación y participación activa

# Capítulo 1: Fundamentación teórica

del recurso humano. Un ejemplo de esto son las tecnologías de la información y las comunicaciones. (8)

## **Aplicación móvil**

Aplicación diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo sea profesional, de ocio, educativas, o de acceso a servicios. (9)

## **Sistema de gestión de aulas**

Los sistemas de gestión de aulas son la pieza indispensable para que profesores y alumnos puedan disfrutar en conjunto de una mejor experiencia en el aula con el uso de las tecnologías. A través de estos sistemas los profesores pueden colaborar con los estudiantes, gestionar sus terminales de forma remota y monitorizar la actividad en el aula. Estos sistemas brindan como principales características:

- ✓ Permitir que los estudiantes aprovechen las nuevas tecnologías en un entorno seguro y controlado.
- ✓ Mayor motivación e interés en la clase.
- ✓ Vigilar a los estudiantes.
- ✓ Permitir al alumno ser más protagonista de su propio aprendizaje. (10)

## **Control remoto**

Los programas de acceso remoto, llamados también aplicaciones de control remoto (remote control software) son aquellas diseñadas para ejecutar funciones de otros programas o sistemas operativos sin encontrarse el usuario frente a estos. Ellos en si no ejecutan los procesos sino que instruyen a las aplicaciones originales para que los ejecuten. (11)

## **Control y monitoreo de la clase**

Son las acciones que ejecuta el profesor para cerciorarse que el proceso docente marcha según lo planificado, asegurando la atención de los estudiantes y el cumplimiento de las actividades del programa sin desatender las diferencias individuales y los ritmos de aprendizaje de los alumnos.

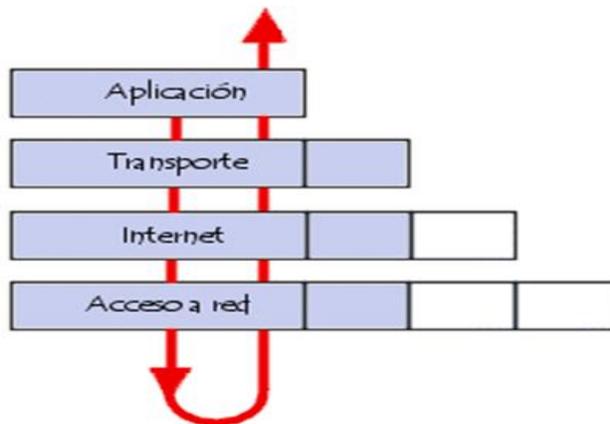
# Capítulo 1: Fundamentación teórica

## Protocolos de comunicación en redes

Las aplicaciones que intercambian información en las redes modernas utilizan diferentes protocolos, la mayoría sobre el modelo TCP/IP<sup>1</sup>. El modelo TCP/IP utiliza un enfoque compuesto por cuatro módulos o capas. A continuación se muestran las funciones de cada una:

- ✓ Capa de acceso a la red: especifica la forma en la que los datos deben enrutarse, sea cual sea el tipo de red utilizado.
  - ✓ Capa de Internet: es responsable de proporcionar el paquete de datos (datagrama).
  - ✓ Capa de transporte: brinda los datos de enrutamiento, junto con los mecanismos que permiten conocer el estado de la transmisión. Comprende a los protocolos TCP y UDP<sup>2</sup>.
  - ✓ Capa de aplicación: incorpora aplicaciones de red estándar (Telnet, SMTP, FTP, etc.).
- (12)

Durante una transmisión, los datos cruzan cada una de las capas en el nivel del equipo remitente. En cada capa, se le agrega información al paquete de datos. Esto se llama encabezado, es decir, una recopilación de información que garantiza la transmisión. En el nivel del equipo receptor, cuando se atraviesa cada capa, el encabezado se lee y después se elimina. Entonces, cuando se recibe, el mensaje se encuentra en su estado original:



**Figura1:** Transmisión de datos a través de las capas del modelo TCP/IP (13)

<sup>1</sup> **TCP/IP:** Protocolo de Control de Transmisión/Protocolo de Internet por sus siglas en inglés Transmission Control Protocol/Internet Protocol

<sup>2</sup> **UDP:** Protocolo de Datagrama de Usuario por sus siglas en inglés User Datagram Protocol

# Capítulo 1: Fundamentación teórica

## Protocolo TCP

TCP es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación, posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él. Proporcionan los datos al protocolo IP, los agrupa en datagramas listos para entregar a la capa de acceso a red. TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

Entre sus principales características están las siguientes:

- ✓ TCP es el responsable de colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
- ✓ TCP permite que el monitoreo del flujo de los datos y así evitar la saturación de la red.
- ✓ TCP permite que los datos se formen en segmentos de longitud variada para "entregarlos" al protocolo IP.
- ✓ TCP permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente.
- ✓ TCP permite comenzar y finalizar la comunicación entre aplicaciones de manera mutua y segura.

Con la utilización de este protocolo, las aplicaciones pueden comunicarse en forma segura (gracias al sistema de acuse de recibo del protocolo TCP) independientemente de las capas inferiores. Esto significa que los router (que funcionan en la capa de Internet) sólo tienen que enviar los datos en forma de datagramas, sin preocuparse con el monitoreo de datos porque esta función la cumple la capa de transporte (o más específicamente el protocolo TCP).

Durante una comunicación usando el protocolo TCP, las dos máquinas deben establecer una conexión. La máquina emisora (la que solicita la conexión) se llama cliente, y la máquina receptora se llama servidor. Por eso se dice que es un entorno Cliente-Servidor. Las máquinas de dicho entorno se comunican en modo en línea, es decir, que la comunicación se realiza en ambas direcciones.

# Capítulo 1: Fundamentación teórica

Para posibilitar la comunicación y que funcionen bien todos los controles que la acompañan, los datos se agrupan; es decir, que se agrega un encabezado a los paquetes de datos que permitirán sincronizar las transmisiones y garantizar su recepción. Otra función del TCP es la capacidad de controlar la velocidad de los datos usando su capacidad para emitir mensajes de tamaño variable. Estos mensajes se llaman segmentos.

## Protocolo UDP

El protocolo UDP es un protocolo no orientado a conexión de la capa de transporte del modelo TCP/IP. Este protocolo es muy simple ya que no proporciona detección de errores. Al igual que TCP conforma segmentos de información de longitud variable que se intercambian entre aplicaciones.

El encabezado del segmento UDP es muy simple y está compuesto de la siguiente manera:

puerto de origen (16 bits);	puerto de destino (16 bits);
longitud total (16 bits);	suma de comprobación del encabezado (16 bits);
datos (longitud variable).	

Significado de los diferentes campos:

- ✓ Puerto de origen: es el número de puerto relacionado con la aplicación del remitente del segmento UDP. Este campo representa una dirección de respuesta para el destinatario. Por lo tanto, este campo es opcional. Esto significa que si el puerto de origen no está especificado, los 16 bits de este campo se pondrán en cero. En este caso, el destinatario no podrá responder (lo cual no es estrictamente necesario, en particular para mensajes unidireccionales).
- ✓ Puerto de destino: este campo contiene el puerto correspondiente a la aplicación del equipo receptor al que se envía.

# Capítulo 1: Fundamentación teórica

- ✓ Longitud: este campo especifica la longitud total del segmento, con el encabezado incluido. Sin embargo, el encabezado tiene una longitud de 4 x 16 bits (que es 8 x 8 bits), por lo tanto la longitud del campo es necesariamente superior o igual a 8 bytes.
- ✓ Suma de comprobación: es una suma de comprobación realizada de manera tal que permita controlar la integridad del segmento.

## Principales diferencias entre UDP y TCP

UDP es un protocolo no orientado a conexión. Es decir cuando una maquina A envía paquetes a una maquina B, el flujo es unidireccional. La transferencia de datos es realizada sin haber realizado previamente una conexión con la máquina de destino (maquina B), y el destinatario recibirá los datos sin enviar una confirmación al emisor (la maquina A). Esto es debido a que la encapsulación de datos enviada por el protocolo UDP no permite transmitir la información relacionada al emisor. Por ello el destinatario no conocerá al emisor de los datos excepto su IP. Este tipo de protocolo es ideal para mensaje de difusión o para construir faros que no son más que mensajes que manda una aplicación para que otras las detecten en la red.

Contrariamente a UDP, el protocolo TCP está orientado a conexión. Cuando una máquina A envía datos a una máquina B, la máquina B es informada de la llegada de datos, y confirma su buena recepción. Aquí interviene el control CRC de datos que permite verificar la integridad de los paquetes transmitidos. De este modo, si los datos recibidos son corruptos, el protocolo TCP permite que los destinatarios soliciten al emisor que vuelvan a enviarlos. Este tipo de protocolo es utilizado para el intercambio seguro de información entre aplicaciones, donde es importante saber quién envía los datos y que estos lleguen de manera íntegra hasta su destino.

(12)

## 1.3 Estudio de aplicaciones similares

Para el desarrollo de la propuesta de solución se realizó un análisis de los sistemas similares que existen a nivel internacional, con el objetivo de definir las principales características que debe cumplir un software de este tipo. A continuación, se describe el resultado de la investigación.

## Capítulo 1: Fundamentación teórica

Se analizaron 12 sistemas de gestión de aulas:

Mythware Classroom Management Software	AB Tutor
Gradelink	LanSchool
Radix	ClassLink
NetSoporte School	XClass
Impero Education Pro	Netop Vision Pro
Pupilpad	iTALC

De estos sistemas se pudo constatar que solo **Netop Vision Pro** posee funcionalidades que permiten controlar la aplicación del profesor desde un dispositivo móvil. Esta solución se basa en extrapolar las funciones más frecuentes y solo controlar las actividades y los estudiantes en el transcurso de la clase.

Vision Pro ofrece dos software independientes, **Vision Hand** y **Vision Mobile** para acceder a la máquina del profesor y ejecutar de manera remota las tareas que se puedan realizar directamente en la misma. En la siguiente tabla se muestra una comparación de estos dos software:

<b>Características</b>	<b>Vision Mobile</b>	<b>Vision Hand</b>
Accede a la computadora del profesor.	si	si
Funciona de manera independiente o con el software de gestión de aula de Vision.	si	no
Ver la pantalla de la computadora del profesor desde un dispositivo móvil.	si	si
Controlar cualquier software en la computadora del profesor.	si	no
Acceder a un documento o cualquier recurso de la computadora del profesor.	si	no
Conectar o Desconectar el acceso a internet.	no	si

## Capítulo 1: Fundamentación teórica

Ver en miniaturas las pantallas de los estudiantes desde cualquier dispositivo con acceso a la web.	no	si
Moverse libremente por el aula.	si	si
Bloquear y desbloquear pantalla.	no	si
Bloquear y desbloquear la web.	no	si
Presentar trabajo de algún estudiante.	no	si
Sistema operativo móvil soportado.	IOS	Android
Comunicación inalámbrica a través del protocolo TCP/IP	si	si

Luego de realizar el estudio de estos software independientes se puede decir que los dos cumplen con un grupo de características imprescindibles para controlar de forma remota las funciones de gestión de aulas. La principal diferencia entre ellos está, en que la primera solo es para sistemas IOS y permite un control más profundo de la computadora del profesor. Estos dos software solo funcionan con el sistema de gestión de aulas de Netop Vision Pro.

Una aplicación que funcione como control remoto del sistema de gestión de aula, debe al menos permitir al profesor moverse libremente por el aula, bloquear y desbloquear pantalla, controlar cualquier software, acceder a un documento o cualquier recurso de su computadora y ver en miniaturas las pantallas de los estudiantes desde cualquier dispositivo con acceso a la web, así como bloquearlas. Características estas presentes en los software de Netop Vision Pro. (13)

Ninguno de los sistemas estudiados implementa todas las funcionalidades del sistema de gestión de aula para el cual fueron diseñados. De manera directa ninguna de sus características es aplicable a la solución que se pretende desarrollar, pero sirven como referencia para comprender como funciona un sistema de este tipo.

### 1.4 Sistema de gestión de aula XAUCE ATcnea

El aula tecnológica XAUCE ATcnea como solución integral comprende dos software desarrollados por la UCI, los medios tecnológicos producidos por la empresa HAIER y el mobiliario aportado por GEDEME.

## *Capítulo 1: Fundamentación teórica*

XAUCE ATcnea adapta la enseñanza al manejo y utilización de los recursos tecnológicos para que el aprendizaje de los estudiantes, se realice de una manera más efectiva y mucho más rápida y ha sido pensada para proporcionar el mejor ambiente de estudio. Constituye una solución educativa amigable, fácil de operar y de colaboración para el proceso de enseñanza y aprendizaje, brindando una experiencia única en el aula a partir del aprovechamiento de las TIC.

Permite la comunicación, interactividad, dinamismo en la presentación de contenidos, atención diferenciada a los alumnos con distintos estilos de aprendizaje, un ambiente colaborativo, el uso de recursos multimedia, métodos didácticos diversificados y tareas auténticas, así como prácticas dinámicas y efectivas.

Entre las principales características de ATcnea se tienen:

- ✓ Transferencia de archivos entre el estudiante y el profesor.
- ✓ Pasar asistencia de forma automática (cuando el estudiante se conecta a la clase automáticamente se registra como presente).
- ✓ Mensajería instantánea.
- ✓ Gestión de la clase.
- ✓ Control de las terminales de los estudiantes, de manera remota, por parte del profesor (abrir sitio web remoto, iniciar aplicación remota por el profesor, recorrer aplicaciones instaladas remotas).
- ✓ Evaluación (evaluar a los estudiantes y seleccionar a qué estudiantes evaluar).
- ✓ Gestionar ejercicios, exámenes y preguntas interactivas.
- ✓ Variadas tipologías de ejercicios.
- ✓ Selección de las escalas de evaluación definidas en el Ministerio de Educación (MINED) y el Ministerio de Educación Superior (MES), ambos de Cuba, para la calificación de las actividades.
- ✓ Aceptar, bloquear, denegar y visualizar mano levantada.
- ✓ Bloquear, desbloquear y visualizar en miniatura la pantalla de la tableta del estudiante.

# Capítulo 1: Fundamentación teórica

- ✓ Streaming<sup>3</sup> de video y audio.
- ✓ Creación de grupos dentro de la clase para poder realizar atención diferenciada a los estudiantes.
- ✓ Reportes de asistencia y evaluaciones.

## 1.5 Metodología de desarrollo de software

Una metodología de desarrollo en ingeniería de software, es un conjunto de herramientas, técnicas, procedimientos y soporte documental encaminados a estructurar, planificar y controlar el proceso de desarrollo de forma organizada y lógica, que tienen como objetivo apoyar a los desarrolladores en la creación de un nuevo software. (14)

Las metodologías de desarrollo de software se clasifican en dos grandes grupos, las tradicionales (Proceso Unificado de Rational) y las ágiles (Programación Extrema, Proceso Unificado Ágil). Las metodologías tradicionales se basan en la idea de que el éxito del producto se puede lograr si se tiene todo correctamente documentado, mientras que, las ágiles defienden la idea de que el proceso de desarrollo de software, se centra en el software como tal y no en la documentación alrededor de este, dándole una mayor importancia a la programación que a la documentación, aunque no lo obvia por completo, solo se toma en cuenta la documentación necesaria y de forma muy sencilla.

En la presente investigación se utilizó la metodología definida por el centro FORTES para el desarrollo de sus proyectos. La misma consiste en una versión desarrollada por la UCI basada en la metodología de desarrollo de software AUP<sup>4</sup>. A continuación, se muestran las características fundamentales de la metodología de desarrollo AUP y su variante para la UCI.

### Metodología de desarrollo de software AUP

La metodología de desarrollo de software AUP es una versión simplificada de la metodología de desarrollo RUP<sup>5</sup>. Describe de una manera fácil y simple de entender la forma de desarrollar

---

<sup>3</sup> **Streaming:** Transmisión

<sup>4</sup> **AUP:** Proceso Unificado Ágil por sus siglas en inglés Ágil Unified Process

<sup>5</sup> **RUP:** Proceso Unificado Racional por sus siglas en inglés Rational Unified Process

# Capítulo 1: Fundamentación teórica

aplicaciones de software de negocio empleando técnicas ágiles y conceptos que se mantienen válidos en RUP. Aplica técnicas ágiles que incluyen:

- ✓ Desarrollo dirigido por pruebas.
- ✓ Modelado ágil.
- ✓ Gestión de cambios ágil.
- ✓ Refactorización de base de datos para mejorar la productividad.

AUP establece cuatro fases que transcurren de manera consecutiva:

- ✓ Inicio: Se obtiene una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- ✓ Elaboración: El equipo de desarrollo profundiza en la comprensión de los requisitos del sistema y en validar la arquitectura.
- ✓ Construcción: El sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- ✓ Transición: El sistema se lleva a entornos donde se somete a pruebas de validación y aceptación; finalmente se despliega en los sistemas de producción.

Además, define 7 disciplinas:

- ✓ Modelo: El objetivo de esta disciplina es entender el negocio de la organización, el problema de dominio que se abordan en el proyecto, y determinar una solución viable para resolver el problema de dominio. Agrupa los flujos de trabajos de Modelado de negocio, Requisitos y Análisis y Diseño.
- ✓ Implementación: El objetivo de esta disciplina es transformar su (s) modelo (s) en código ejecutable y realizar un nivel básico de las pruebas, en particular, la unidad de pruebas.
- ✓ Prueba: El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificar que se cumplan los requisitos.
- ✓ Despliegue: El objetivo de esta disciplina es la prestación y ejecución del sistema y que el mismo este a disposición de los usuarios finales.

## *Capítulo 1: Fundamentación teórica*

- ✓ Gestión de configuración: El objetivo de esta disciplina es la gestión de acceso a herramientas de su proyecto. Esto incluye no sólo el seguimiento de las versiones con el tiempo, sino también el control y gestión del cambio para ellos.
- ✓ Gestión de proyectos: El objetivo de esta disciplina es dirigir las actividades que se lleva a cabo en el proyecto. Esto incluye la gestión de riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, etc.), coordinación con el personal y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.
- ✓ Entorno: El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea el adecuado, la orientación (normas y directrices), y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

### **Metodología de desarrollo de software AUP variante UCI.**

La UCI desarrolló una versión de la metodología de desarrollo de software AUP, con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre. Se decide mantener las 7 disciplinas, manteniéndose algunas de las propuestas por AUP y unificándose o arreglándose las otras. Para una mejor comprensión se muestra como Anexo 1 las Tablas 1.1 y 1.2.

En el modelado de negocio se proponen tres variantes a utilizar en los proyectos: Caso de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) o Modelo Conceptual (MC) y existen tres formas de encapsular los requisitos: Caso de Uso del Sistema (CUS), Historia de Usuario (HU) y Descripción de Requisitos por Procesos (DRP). A partir de esto surgen cuatro escenarios para modelar el sistema en los proyectos, manteniéndose en dos de ellos el MC, quedando de la siguiente forma:

**Escenario No 1:** Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

# Capítulo 1: Fundamentación teórica

**Escenario No 2:** Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

**Escenario No 3:** Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

**Escenario No 4:** Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Teniendo en cuenta las características antes mencionadas, la necesidad de usar una metodología que responda con facilidad a los cambios continuos siguiendo las políticas de desarrollo de software de la universidad, se decide utilizar la variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Identificándose el cuarto escenario para encapsular los requisitos debido a que en el proyecto no se modela el negocio, además el cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos.

## 1.6 Herramientas y tecnologías

El proceso de desarrollo de software se apoya en el uso de diferentes herramientas y tecnologías, las cuales, unidas a la metodología seleccionada, conforman el ambiente de desarrollo de un sistema. A continuación se describen las características de las herramientas y tecnologías seleccionadas para desarrollar la solución teniendo en cuenta las utilizadas por el proyecto: Desarrollo del producto Aula Tecnológica (ATcnea) para GEDEME.

### 1.6.1 Lenguaje de modelado

**UML**<sup>6</sup> es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, como las cosas concretas. Sus principales funciones son:

- ✓ Visualizar: Permite representar cualquier sistema de forma gráfica de manera que cualquier persona lo pueda entender.

---

<sup>6</sup> **UML:** Lenguaje Unificado de Modelado por sus siglas en inglés Unified Modeling Language.

# Capítulo 1: Fundamentación teórica

- ✓ Especificar: Facilita especificar las características de un sistema.
- ✓ Construir: Mediante los modelos permite construir el sistema.
- ✓ Documentar: Documenta los elementos gráficos lo que se puede utilizar como documentación, permitiendo una futura revisión. (15)

Se utilizó para el desarrollo de la propuesta de solución la versión 2.0.

## 1.6.2 Herramienta de modelado

**Visual Paradigm** es una herramienta CASE<sup>7</sup> que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Soporta, entre otros, el lenguaje de modelado UML. Permite el análisis, diseño, codificación, prueba y despliegue de aplicaciones. Está disponible para Windows y Linux. (16) Entre sus características se encuentra la capacidad de realizar ingeniería directa e inversa, generación de código, generación de bases de datos, la incorporación de varios idiomas y una licencia gratuita y comercial, así como un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. (17) Se utilizó para el desarrollo de la propuesta de solución la versión 8.0.

## 1.6.3 Lenguaje de programación

**JAVA** es un lenguaje de programación con elementos de C, C++ y otros lenguajes, que está unido a un gran conjunto de bibliotecas con un soporte de ejecución estándar y una máquina virtual. Constituye un lenguaje de propósito general, de alto nivel, y orientado a objetos puro, en el sentido de que no hay ninguna variable, función o constante que no esté dentro de una clase. Java es un lenguaje dinámico debido a que las clases son cargadas en el momento en que son necesitadas, ya sea del sistema de archivos local o desde algún sitio de la red. (18) Se utilizó para el desarrollo de la propuesta de solución la versión 2.0

---

<sup>7</sup> **CASE**: Ingeniería de Software Asistida por Computación por sus siglas en inglés Computer Aided Software

# Capítulo 1: Fundamentación teórica

## 1.6.4 Herramienta de desarrollo

**NetBeans IDE**<sup>8</sup> de código abierto dedicado a proveer un sólido desarrollo de software, dirigido fundamentalmente a las necesidades de los desarrolladores y los usuarios; dotándolos de una herramienta para el desarrollo rápido y fácil de productos de software. NetBeans es una herramienta modular de desarrollo que incluye un avanzado editor multilenguaje y un detector de errores. Además cuenta con una amplia documentación y una gran comunidad de usuarios. (19) Se utilizó para el desarrollo de la propuesta de solución la versión 8.3.

**Android Studio** es el IDE oficial para Android, basado en el software IntelliJ IDEA de JetBrains. Entre sus características principales se encuentra la renderización en tiempo real; la incorporación de Gradle como compilador, ofreciendo múltiples variantes para la construcción y generación de APKs<sup>9</sup>; la existencia de plantillas predeterminadas para crear diseños comunes de Android y otros componentes; herramientas Lint para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones (20). Ofrece además un editor visual y un editor de temas para las aplicaciones, así como refactorización específica de Android y arreglos rápidos, consola de desarrollador, consejos de optimización, ayuda para la traducción y estadísticas de uso. (21) Se utilizó para el desarrollo de la propuesta de solución la versión 2.2.

## 1.6.5 Herramienta de control de versiones

**Git** es un sistema de control de versiones diseñado por Linux Torvalds que tiene como objetivo la eficiencia y la confiabilidad en el mantenimiento de versiones de las aplicaciones. Es un sistema de control distribuido de versiones con un énfasis en la velocidad, la integridad de datos y el soporte para flujos de trabajo distribuidos. Una de sus principales características es que el código puede almacenarse de forma pública, aunque también se puede hacer de forma privada. Se utilizó para el desarrollo de la propuesta de solución la versión 2.7.4.

---

<sup>8</sup> **IDE:** Entorno de Desarrollo Integrado por sus siglas en inglés Integrated Development Environment.

<sup>9</sup> **APK:** Aplicación instalable en dispositivos móviles con sistema operativo Android por sus siglas en inglés Android Package.

# Capítulo 1: Fundamentación teórica

## 1.6.6 Herramientas de pruebas

**JUnit** es un marco de trabajo creado por Erich Gamma y Kent Becket para realizar pruebas unitarias a aplicaciones que empleen el lenguaje de programación Java. Incluye formas de ver los resultados de las pruebas ya sea en forma de texto, gráfico o como tarea. Incluido a través de plugins en los principales IDEs como son NetBeans, Eclipse y Android Studio. De código abierto, puede integrarse con Maven, emplea anotaciones y condiciones de aceptación para facilitar el trabajo. (22) Se utilizó para el desarrollo de la propuesta de solución la versión 4.12.

## 1.7 Conclusiones parciales

A partir del desarrollo del presente capítulo se arribó a las siguientes conclusiones:

- El estudio de los conceptos básicos asociados al objeto de estudio posibilitó la adquisición de una mayor comprensión del problema a resolver.
- El estudio de las soluciones existentes a nivel internacional arrojó como resultado que las aplicaciones estudiadas no brindan una solución completa al problema a resolver, debido fundamentalmente a que están diseñadas para trabajar con un sistema de gestión de aula específico.
- Las diferentes herramientas y tecnologías, así como de la metodología a utilizar, permitió profundizar los conocimientos necesarios para el desarrollo de la solución a la problemática planteada.

## *Capítulo 2: Descripción de la solución propuesta*

### **Capítulo 2: Descripción de la solución propuesta**

#### **2.1 Introducción**

En el presente capítulo se describe la propuesta de solución del sistema y se presentaron los artefactos generados que facilitan al desarrollador entender las funcionalidades y garantizar una implementación exitosa. Se representa el diagrama del modelo del dominio con la correspondiente descripción de cada uno de sus componentes. Se capturan los requisitos funcionales y no funcionales que debe cumplir el sistema y se conforman las historias de usuarios correspondientes a los mismos. Además, se describe la arquitectura de la aplicación, así como los patrones de diseño a aplicar.

#### **2.2 Descripción del sistema**

El presente trabajo propone como solución desarrollar una aplicación para dispositivos móviles con sistema operativo Android que brinde al profesor mayor movilidad en el aula sin perder el control y monitoreo de la clase.

La herramienta desarrollada se conecta a la aplicación ATcnea utilizando el protocolo TCP. Cuenta con una sola sesión la cual está destinada al profesor que impartirá la clase. Una vez autenticado, el sistema brinda una serie de funcionalidades mediante la cual podrá mantener el control y monitoreo de la clase desde cualquier punto del aula.

La solución propuesta implementa la mayor cantidad de funcionalidades del sistema de gestión de aula. Las funcionalidades bloquear chat y bloquear pantalla le permitirá al profesor lograr la atención absoluta de los estudiantes. Podrá reproducir videos, transmitir diapositivas, abrir sitio web remoto, entre otras funcionalidades que se utilizarán para el desempeño de las actividades durante la clase. Además se podrán enviar tareas, preguntas interactivas y exámenes los cuales permitirán evaluar el desempeño de los estudiantes de la clase, todo desde la tableta del profesor sin necesidad de acceder a la laptop de manera directa.

## Capítulo 2: Descripción de la solución propuesta

### 2.3 Modelo de dominio

El modelo de dominio es utilizado por el analista como un medio para comprender el sistema debido a que es una especie de representación esquemática de los conceptos y elementos de la vida real que serán usados en el mismo. Se crea con el fin de representar los conceptos claves del dominio del problema, las propiedades más importantes y las relaciones entre los conceptos facilitando una mejor comunicación entre desarrolladores y clientes al establecer un lenguaje común para el entendimiento del mismo. (23)

#### 2.3.1 Diagrama de clases del modelo del dominio

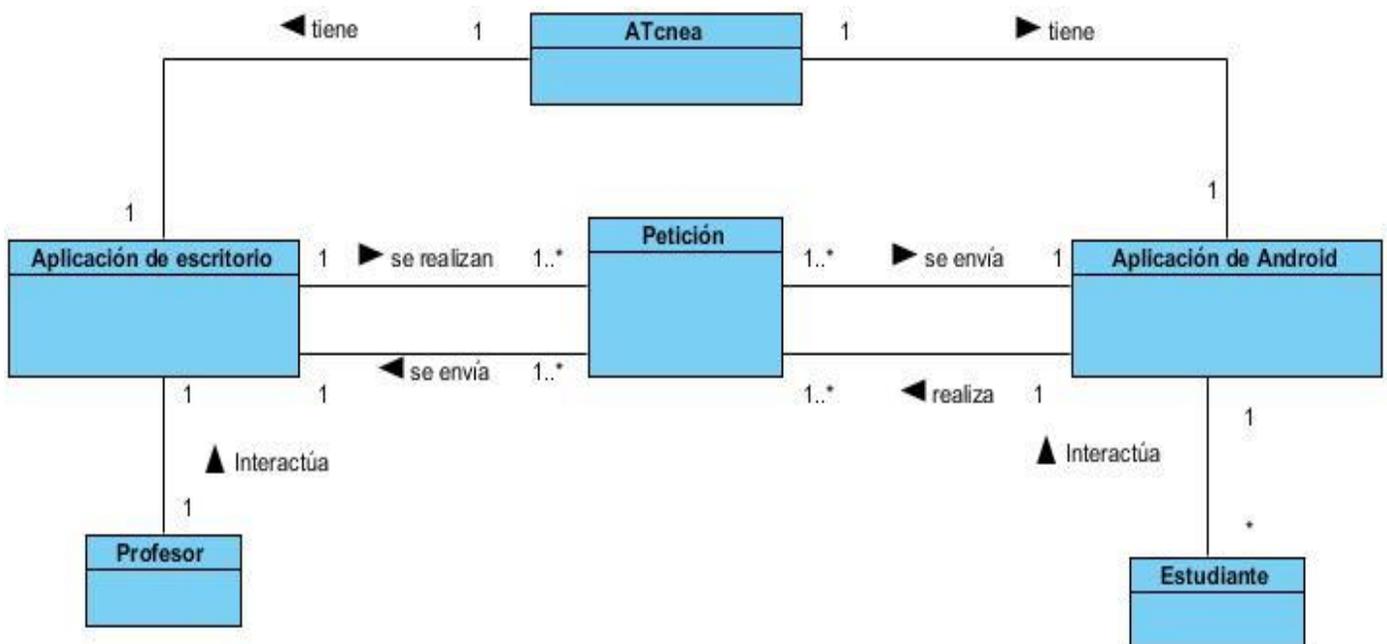


Figura 2: Diagrama de clases del modelo de dominio

#### 2.3.2 Definición de las clases del modelo del dominio

A continuación, se muestra la definición de cada una de las clases del modelo de dominio:

**ATcnea:** Identifica al aula tecnológica cubana XAUCE ATcnea.

**Aplicación de escritorio:** Identifica el software alojado en la laptop del profesor en donde se encuentra el contenido de la clase.

## *Capítulo 2: Descripción de la solución propuesta*

**Aplicación de Android:** Identifica el software alojado en las tabletas de los estudiantes donde se recibe el contenido impartido por el profesor.

**Profesor:** Identifica al usuario que interactúa con la laptop para impartir el contenido.

**Estudiante:** Identifica al usuario que interactúa con la tableta para recibir el contenido.

**Petición:** Identifica a las diferentes peticiones que se pueden realizar desde la aplicación de escritorio y las que el profesor puede recibir a través de las tabletas de los estudiantes.

### 2.4 Especificación de los requisitos

El levantamiento de los requisitos constituye una de las etapas más importantes en el proceso de ingeniería de requisitos. Para lograr el éxito durante el desarrollo de esta etapa, se realiza un trabajo en conjunto entre los participantes y los desarrolladores con el objetivo de identificar el problema, proponer elementos de solución, negociar diferentes enfoques y especificar un conjunto preliminar de requisitos para la solución. (24)

#### 2.4.1 Requisitos funcionales

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera que éste debe reaccionar a entradas particulares y cómo se debe comportar en situaciones particulares (25). Seguidamente se muestran los requisitos funcionales definidos para la siguiente investigación los cuales van estar estructurados por paquetes:

##### **Paquete Conexión**

**RF1 Iniciar sesión:** Permite al profesor desde su terminal iniciar sesión en el sistema.

**RF2 Cerrar sesión:** Permite al profesor desde su terminal cerrar sesión en el sistema.

##### **Paquete Clase**

**RF3 Buscar clase:** Permite al profesor desde su terminal buscar las clases activas.

**RF4 Seleccionar clase:** Permite al profesor desde su terminal seleccionar una de las clases activas.

**RF5 Permitir la solicitud de estudiantes para conectarse a la clase:** Permite al profesor desde su terminal aceptar o denegar la solicitud de un estudiante para conectarse a la clase.

## *Capítulo 2: Descripción de la solución propuesta*

**RF6 Impedir desconexión de la clase:** Permite al profesor desde su terminal impedir la desconexión de los estudiantes de la clase.

**RF7 Desconectar estudiante de la clase:** Permite al profesor desde su terminal desconectar los estudiantes seleccionados de la clase.

### **Paquete Archivo**

**RF8 Permitir solicitud de envío de archivo del estudiante:** Permite al profesor desde su terminal aceptar o denegar la solicitud de envío de archivo de los estudiantes.

### **Paquete Chat**

**RF9 Bloquear chat de los estudiantes:** Permite al profesor desde su terminal bloquear el chat a uno o varios estudiantes.

**RF10 Desbloquear chat de los estudiantes:** Permite al profesor desde su terminal desbloquear el chat a uno o varios estudiantes.

**RF11 Establecer conversación con los estudiantes:** Permitir al profesor desde su terminal enviar o recibir mensajes de los estudiantes.

### **Paquete Pantalla**

**RF12 Bloquear pantalla:** Permite al profesor desde su terminal bloquear la pantalla del estudiante.

**RF13 Desbloquear pantalla:** Permite al profesor desde su terminal desbloquear pantalla del estudiante.

**RF14 Visualizar la pantalla de un estudiante en el terminal del profesor:** Permite al profesor desde su terminal visualizar la pantalla de un estudiante en su terminal.

**RF15 Grabar pantalla del estudiante:** Permite al profesor desde su terminal grabar la pantalla del estudiante en su terminal.

**RF16 Terminar transmisión de la pantalla del estudiante:** Permite al profesor desde su terminal suspender la transmisión de la pantalla del estudiante en su terminal.

**RF17 Compartir pantalla del profesor en modo completo:** Permite al profesor desde su terminal transmitir su pantalla en los terminales de los estudiantes.

## *Capítulo 2: Descripción de la solución propuesta*

### **Paquete Control Remoto**

**RF19 Ver listado de aplicaciones remotas:** Permite al profesor desde su terminal visualizar el listado de aplicaciones.

**RF20 Ver listado de sitios web:** Permite al profesor desde su terminal visualizar el listado de sitios web.

**RF21 Abrir sitio web remoto:** Permite al profesor desde su terminal abrir de forma remota un sitio web en los terminales de los estudiantes.

**RF22 Agregar sitio web:** Permite al profesor desde su terminal agregar un sitio web al listado de sitios web.

**RF23 Eliminar sitio web:** Permite al profesor desde su terminal eliminar un sitio web del listado de sitios web.

**RF24 Agregar aplicación:** Permite al profesor desde su terminal agregar una aplicación al listado de aplicaciones

**RF25 Abrir aplicación remota:** Permite al profesor desde su terminal abrir una aplicación de forma remota en los terminales de los estudiantes.

**RF26 Visualizar aplicaciones activas:** Permite al profesor desde su terminal visualizar las aplicaciones activas en los terminales de los estudiantes.

**RF27 Cerrar aplicaciones activas remotas:** Permite al profesor desde su terminal cerrar las aplicaciones que se encuentran ejecutándose en el terminal de los estudiantes.

### **Paquete Diapositiva**

**RF28 Pasar diapositiva hacia adelante:** Permite al profesor desde su terminal desplazar hacia adelante la diapositiva.

**RF29 Pasar diapositiva hacia atrás:** Permite al profesor desde su terminal desplazar hacia atrás la diapositiva.

**RF30 Salir de la diapositiva:** Permite al profesor desde su terminal cerrar la diapositiva.

### **Paquete Evaluación**

**RF31 Evaluar estudiantes:** Permite al profesor desde su terminal evaluar a uno o varios estudiantes previamente seleccionados.

## *Capítulo 2: Descripción de la solución propuesta*

**RF32 Notificar estudiantes de la evaluación:** Permite al profesor desde su terminal emitir una evaluación al estudiante mediante una notificación.

### **Paquete Examen**

**RF33 Ver listado de exámenes:** Permite al profesor desde su terminal visualizar el listado de exámenes.

**RF34 Enviar examen a los estudiantes:** Permite enviar un examen a los estudiantes previamente seleccionados.

### **Paquete Mano**

**RF35 Permitir manos levantadas:** Permite al profesor desde su terminal aceptar o denegar la solicitud de levantar mano emitida por los estudiantes.

**RF36 Bloquear manos levantadas:** Permite al profesor desde su terminal bloquear la acción de levantar la mano en los terminales de uno o varios estudiantes.

**RF37 Desbloquear mano levantada:** Permite al profesor desde su terminal desbloquear la acción de levantar la mano en los terminales de uno o varios estudiantes.

**RF38 Visualizar mano levantada:** Permite al profesor desde su terminal visualizar el nombre de los estudiantes que seleccionaron la opción Mano levantada.

### **Paquete Pregunta Interactiva**

**RF39 Ver listado de preguntas interactivas:** Permite al profesor desde su terminal visualizar la lista de preguntas interactivas.

**RF40 Enviar pregunta interactiva:** Permite al profesor enviar la pregunta interactiva.

### **Paquete Video**

**RF41 Reproducir archivo de video:** Permite al profesor desde su terminal reproducir el archivo de video que ha sido pausado por el profesor.

**RF42 Pausar video:** Permite al profesor desde su terminal pausar la reproducción del video que se está ejecutando en la clase.

**RF43 Detener video:** Permite al profesor desde su terminal detener el video.

### **Paquete Tarea**

**RF44 Ver listado de tarea:** Permite al profesor desde su terminal visualizar las tareas.

## *Capítulo 2: Descripción de la solución propuesta*

**RF45 Enviar tarea:** Permite al profesor desde su ordenador enviar una tarea a uno o varios estudiantes.

### 2.4.2 Requisitos no funcionales

Los requisitos no funcionales, como su nombre sugiere, son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las restricciones de los servicios o funciones ofrecidas por el sistema. Normalmente solo se aplican a características o servicios individuales del sistema. (25) A continuación se muestran los requisitos no funcionales definidos para la investigación:

✓ **Confiabilidad:**

**RNF 1:** Al sistema solo tendrán acceso los usuarios registrados como profesor, la información estará restringida por una contraseña.

**RNF 2:** El sistema debe verificar las consecuencias asociadas a los fallos del sistema.

✓ **Portabilidad:**

**RNF 3:** La instalación de la herramienta debe ser en una tableta que cuente con Android 4,0 o superior. Se recomienda 1 GB RAM o superior, 7 pulgadas o superior de pantalla, CPU/GPU Dual Core 1.0GHz o superior.

✓ **Interfaz de usuario:**

**RNF 4:** Cumplir con las pautas de diseño establecidas en la estrategia marcaría de la universidad.

### 2.5 Historias de usuarios

Las historias de usuario (HU), son una técnica utilizada para especificar los requisitos del software. Son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Se elabora una por cada funcionalidad. El tiempo de desarrollo ideal para una HU es entre una semana y tres semanas y permiten responder rápidamente a los requerimientos cambiantes. (26)

## Capítulo 2: Descripción de la solución propuesta

A continuación, se presenta la HU correspondiente al requisito funcional Iniciar Sesión.

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Iniciar sesión
<b>Programador:</b> Maylin Torres Rodríguez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 8 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A
<p><b>Descripción</b></p> <p><b>1- Objetivo:</b> Permitir a los usuarios iniciar sesión en el sistema.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> El terminal del profesor debe estar conectado a la laptop del profesor. El usuario debe tener una cuenta creada dentro del sistema (contraseña). Debe haber seleccionado previamente la clase.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b> El campo contraseña es obligatorio. Contraseña: Campo de texto que identifica la clave secreta de los usuarios para autenticarse en el sistema. Admite los caracteres alfanuméricos: guión bajo (_), arroba (@), punto (.), asteriscos (*), mayúsculas y minúsculas. El tamaño límite es de 5 a 7 caracteres. Ejemplo: ATcne4*C8</p> <p><b>4- Flujo de la acción a realizar:</b> Cuando el usuario seleccione la clase a la que se desea conectar, el sistema debe permitir al profesor identificarse, permitiendo introducir el campo obligatorio:</p> <p><b>(*) Contraseña</b> Una vez que el usuario se autentique el sistema debe mostrar la interfaz correspondiente al listado de los estudiantes conectados a la clase y las diferentes funcionalidades que le brinda el sistema.</p>	

## Capítulo 2: Descripción de la solución propuesta

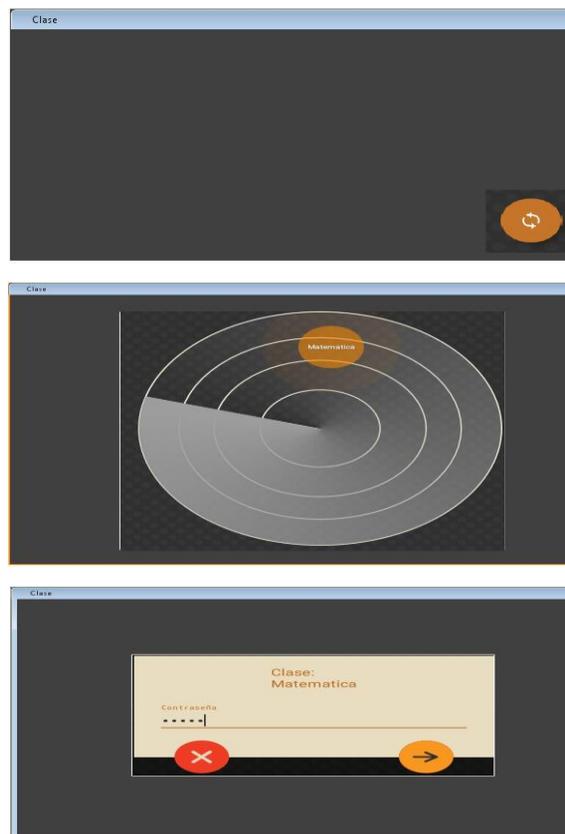
En caso que el usuario no introduzca correctamente la contraseña el sistema le permitirá escanear nuevamente las clases activas.

### Observaciones:

-El sistema debe visualizar la contraseña insertada en forma encriptada, esta acción solo se realizará antes de presionar la opción Iniciar Sesión.

-La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos.

### Prototipo de interfaz:



### 2.6 Patrón arquitectónico

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una

## *Capítulo 2: Descripción de la solución propuesta*

descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. (24)

Para el desarrollo la propuesta de solución se utilizó el patrón arquitectónico Modelo-Vista-Controlador (MVC) teniendo en cuenta que es empleada cada vez más en la implementación de aplicaciones Android.

### **Patrón Modelo Vista Controlador (MVC)**

El patrón Modelo-Vista-Controlador (MVC), es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario. Permite no mezclar lenguajes de programación en el mismo código. Divide las aplicaciones en tres niveles que a continuación se explicarán cómo fueron utilizados en el sistema (27).

Esta arquitectura separa presentación e interacción de los datos del sistema. El sistema se estructura en tres componentes lógicos que interactúan entre sí. El componente Modelo almacena datos/estado en clase de objeto complejo o simple de datos primarios, no almacena los datos en sí, sino que recupera los datos de forma remota. El componente Vista define y gestiona como se presentan esos datos al usuario y el componente Controlador dirige la interacción del usuario y pasa estas interacciones a la Vista y el Modelo. Permite que los datos cambien de manera independiente de su representación y viceversa. (28)

## Capítulo 2: Descripción de la solución propuesta

A continuación se muestra una imagen donde se representa cada uno de los componentes del MVC.

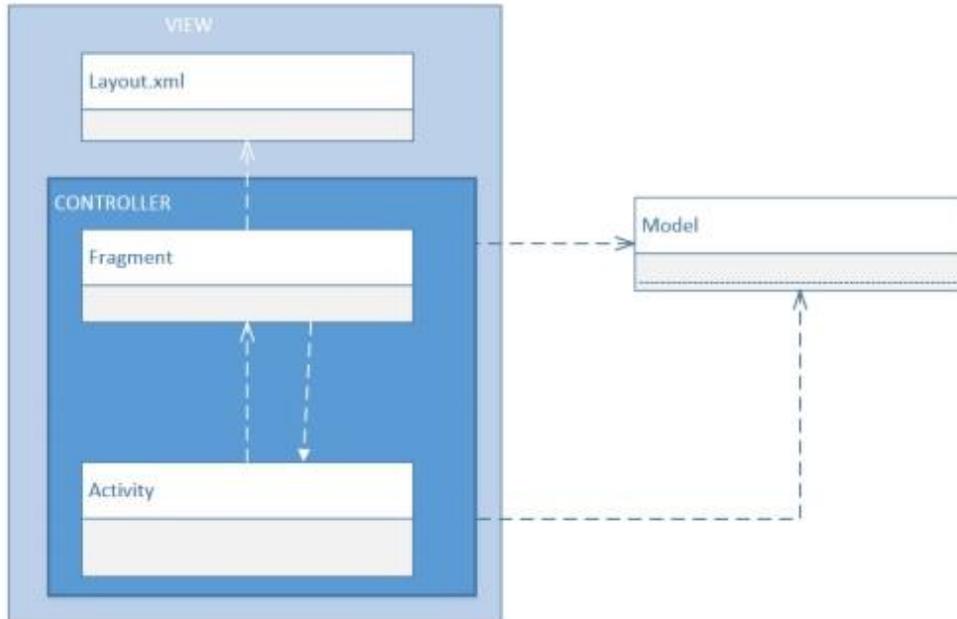


Figura 3: Android MVC. (28)

### 2.7 Patrones de diseños

Un patrón de diseño es una descripción de la comunicación entre objetos y clases, personalizada para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades (29). Se presentan como pares de problema-solución con nombre, sugiriendo aspectos relacionados con la asignación de responsabilidades.

Los patrones de diseño se caracterizan por:

- ✓ Representar soluciones técnicas a problemas concretos.
- ✓ Propiciar la reutilización.
- ✓ Representar problemas frecuentes.

## *Capítulo 2: Descripción de la solución propuesta*

### **Patrones GRASP**

Los patrones GRASP<sup>10</sup> describen los principios fundamentales de la asignación de responsabilidades a objetos (30). El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar un software de manera eficaz.

**Experto:** Se utiliza más que cualquier otro al asignar responsabilidades, es un principio básico que suele utilizarse en el diseño orientado a objetos (31). Consiste en la asignación de una responsabilidad a la clase que cuenta con la información necesaria para llevarla a cabo. El uso de este patrón da pie a un bajo acoplamiento y una alta cohesión, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. El cumplimiento de una responsabilidad requiere a menudo información distribuida en varias clases de objetos.

**Bajo Acoplamiento:** El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (31). El bajo acoplamiento soporta el diseño de clases más independientes y reutilizables, lo cual reduce el impacto de los cambios y acrecienta la oportunidad de una mayor productividad. Este patrón fue utilizado en las clases Quest, Notify, User.

**Alta Cohesión:** En la perspectiva del diseño orientado a objetos, la cohesión (o más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme, clases con responsabilidades moderadas en un área funcional que colaboran con las otras para llevar a cabo las tareas (31). Este patrón fue utilizado en las clases Quest, Notify, User.

---

<sup>10</sup> **GRASP:** Patrones Generales de Software para Asignar Responsabilidades por sus siglas en inglés General Responsibility Assignment Software Patterns.

## Capítulo 2: Descripción de la solución propuesta

**Controlador:** Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Consiste en asignar la responsabilidad del manejo de los eventos de un sistema a una única clase (31). Este patrón fue utilizado en la clase ServerUtil.

**Creador:** Este patrón como su nombre lo indica es el que crea, el guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un objeto de la clase A solamente cuando

- ✓ B contiene a.
- ✓ B es una agregación (o composición) de A.
- ✓ B almacena a.
- ✓ B tiene los datos de inicialización de A (datos que requiere su constructor).
- ✓ B usa a.

A la hora de crear objetos se deben tener en cuenta las características de la clase (32). Este patrón fue utilizado en la clase ServerUtil.

```
public class ServerUtil {  
    ...  
    public void Inicialite(){  
        taskList=new ArrayList<>();  
        studentAdapter=new StudentAdapter();  
        webAdapter=new WebAdapter();  
        webSendAdapter=new WebSendAdapter();  
        ...  
    }  
    ...  
}
```

Figura 4: Patrón Creador

## Capítulo 2: Descripción de la solución propuesta

### Patrones GOF

Los patrones GOF<sup>11</sup> son alternativas de solución a problemas conocidos pero son mucho más específicas las situaciones en las que se aplican. Se clasifican en creacionales, estructurales y de comportamiento (29).

**Singleton:** Está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella (29). Este patrón fue utilizado en la clase ServerUtil.

```
public class ServerUtil {
    private static ServerUtil instance;
    public static ServerUtil getInstance() {
        if(instance==null) {
            instance = new ServerUtil();
        }
        return instance;
    }
    ...
}
```

Figura 5: Patrón Singleton

### 2.7.2 Diagramas de clases del diseño (DCD)

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones incluyendo herencia, agregación y asociación. Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (33).

---

<sup>11</sup> **GOF:** por sus siglas en inglés Gang of Four. Traducido al español La pandilla de los cuatro.

## Capítulo 2: Descripción de la solución propuesta

A continuación se muestra el diagrama de clase del diseño correspondiente a la HU Iniciar Sesión. Los restantes diagramas se muestran en el Anexo #4.

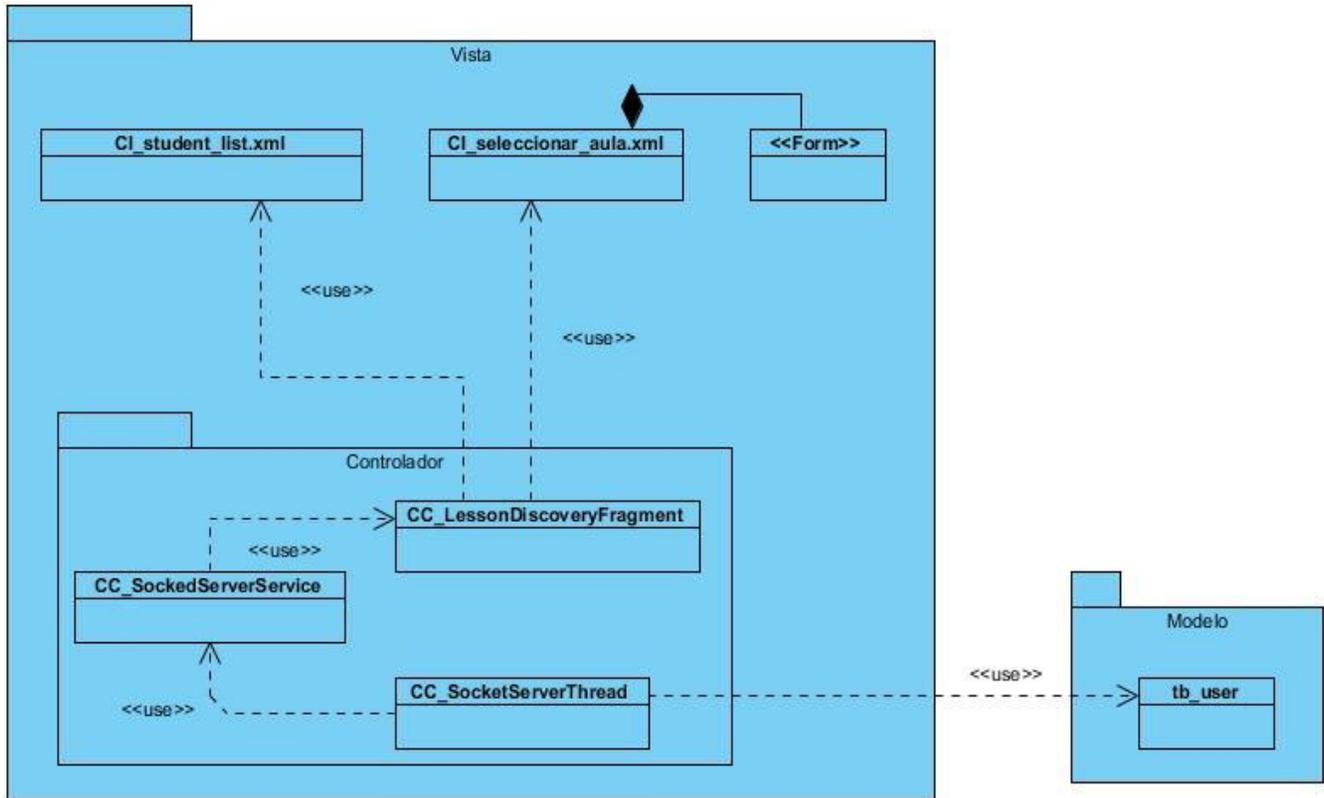


Figura 6: DCD\_Iniciar\_Sesión

### 2.7.3 Diagrama de secuencia del diseño (DSD)

El diagrama de secuencia muestra cómo se pasa de un objeto a otro a medida que se ejecuta la HU relacionándose a través de mensajes. A continuación se muestra el diagrama de secuencia correspondiente a la HU Iniciar Sesión. Los restantes diagramas se muestran en el Anexo #3.

## Capítulo 2: Descripción de la solución propuesta

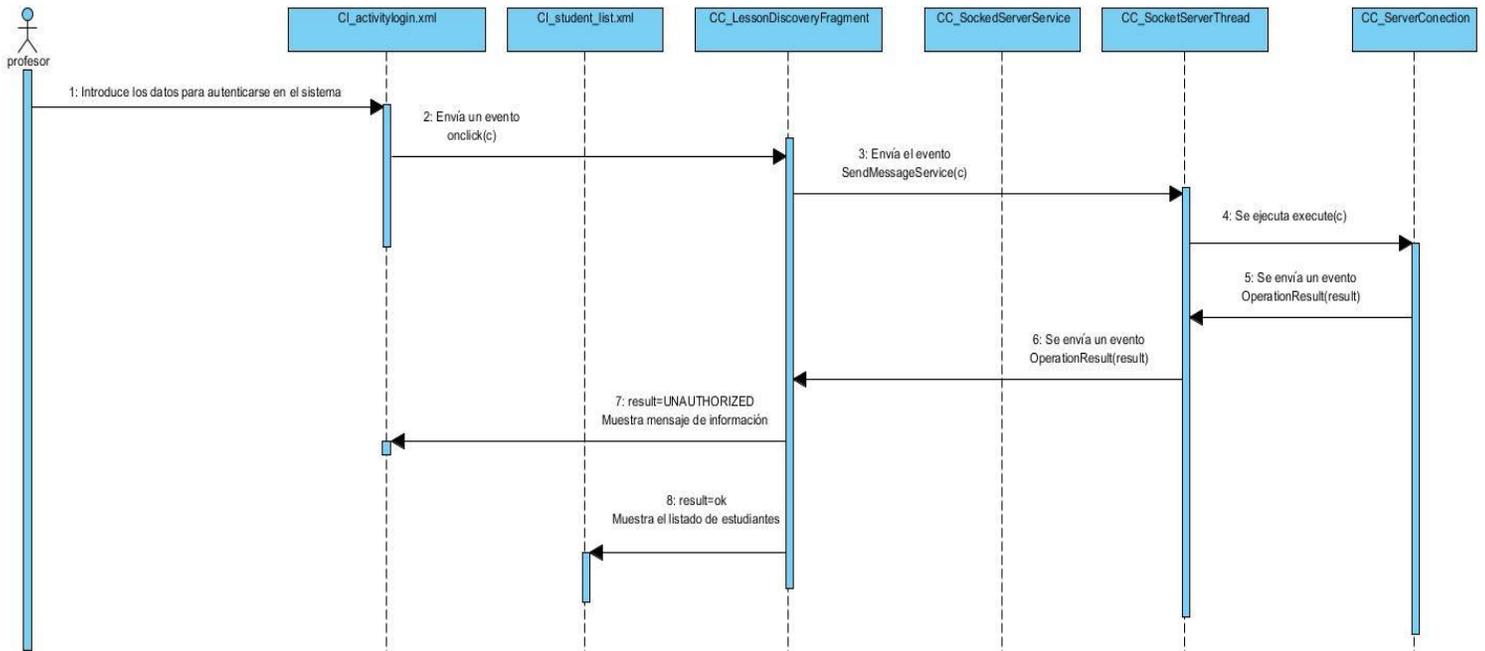


Figura 7: DSD Iniciar\_Sesión

### 2.8 Conclusiones parciales

A partir del desarrollo del presente capítulo se arribó a las siguientes conclusiones:

- Se identificaron los requerimientos de la aplicación para un total de 45 requisitos funcionales y 4 no funcionales los cuales fueron agrupados en 45 historias de usuario. Se realizó la descripción textual de cada uno de ellos, lo cual permitió comprender mejor las funcionalidades de la aplicación que se desea desarrollar.
- Se describe el patrón arquitectónico modelo vista controlador que presenta la propuesta de solución, teniendo en cuenta el uso de Android como sistema operativo.
- Con la realización del diagrama de clases del diseño se obtuvo una visión más exacta de la aplicación en términos de implementación.

# Capítulo 3: Implementación y prueba

## Capítulo 3: Implementación y prueba de la solución propuesta

### 3.1 Introducción

En el presente capítulo se exponen las especificaciones asociadas a la implementación de la aplicación. Se describen las pautas de codificación. Se presenta el diagrama de despliegue correspondiente a la propuesta de solución y un resumen de las pruebas realizadas al software para garantizar su correcto funcionamiento.

### 3.2 Implementación

La codificación de la solución propuesta tiene lugar una vez que se han definido las historias de usuario y se ha concluido el diseño de la aplicación. Está encaminada a desarrollar de forma iterativa e incremental un producto completo listo para el despliegue, obteniendo versiones útiles de forma rápida, las que paulatinamente completan el desarrollo de la aplicación.

#### 3.2.1 Adecuaciones realizadas sobre la aplicación de escritorio de ATcnea

Para el funcionamiento de la aplicación de control de ATcnea fue necesario realizar cambios en los componentes internos de la aplicación destinada al profesor. La mayoría de las funcionalidades controladas desde la tableta del profesor necesitan una contraparte desde el lado de ATcnea que reciba la instrucción y ejecute la funcionalidad, a la vez que notifiquen los cambios a la tableta para así actualizar la interfaz.

A todas las funcionalidades modificadas, se les agrego un código de compatibilidad que sin cambiar la estructura del proyecto ajustara los nuevos comportamientos de las funcionalidades. A continuación, se describen las partes del software del profesor donde se realizaron los cambios.

Componente para la gestión de las peticiones y acciones de las manos de los estudiantes

- ✓ Evento de escucha para sincronizar las peticiones de los estudiantes con la aplicación de control.

## Capítulo 3: Implementación y prueba

- ✓ Funcionalidad de gestionar los eventos de mano en la aplicación de control, para de esta forma controlar dicha funcionalidad.
- ✓ Las acciones de bloquear y desbloquear se gestionan por la aplicación de control.

### Componente para la gestión de la pantalla

- ✓ Acción de visualizar las pantallas compartidas de los estudiantes y de cancelar dicho evento.
- ✓ Acción de compartir la pantalla del profesor, así como de cancelar dicho evento.
- ✓ Control desde la aplicación de control de grabar la pantalla compartida.

### Componente para la gestión del envío de tareas, exámenes y preguntas interactivas

- ✓ Acción de visualizar el listado de las tareas, exámenes y preguntas interactivas.
- ✓ Envío desde la aplicación de control para enviar las tareas, los exámenes y las preguntas interactivas.

### Componente para la gestión de control sobre los videos y las diapositivas

- ✓ Control desde la aplicación de control de reproducir el video.
- ✓ Control desde la aplicación de control de pausar el video.
- ✓ Control desde la aplicación de control de detener el video.
- ✓ Control desde la aplicación de control de pasar hacia adelante y hacia atrás las diapositivas.

### 3.2.2 Estándares de codificación

Durante el proceso de implementación de un software es considerado una buena práctica realizar la codificación del mismo siguiendo estándares que guíen este proceso. Los estándares de codificación permiten un mejor entendimiento del código por parte de todos los miembros del equipo de desarrollo y en consecuencia hace que el código sea fácil de mantener. El uso de estándares de codificación trae consigo los siguientes beneficios:

- ✓ Facilita el mantenimiento de una aplicación.
- ✓ Permite que cualquier programador entienda y pueda mantener la aplicación.

## Capítulo 3: Implementación y prueba

- ✓ Mejoran la legibilidad del código, al mismo tiempo que permiten su rápida comprensión (34).

En la propuesta de solución, se tendrán en cuenta las siguientes convenciones definidas por el equipo de desarrollo:

- ✓ No se emplearán caracteres especiales (@, #, \$, %, ^, &, \* u otros) para la nomenclatura.
- ✓ El nombre de las clases utilizan el estándar UpperCamelCase, CamelCase o PascalCase, las cuales comienzan con la primera letra mayúscula y las demás en minúscula. En caso de ser un nombre compuesto se escribe el segundo nombre seguido del primero, también con la primera letra mayúscula y el resto minúscula.
- ✓ Cada atributo debe tener encima un comentario que lo identifique.
- ✓ Los métodos deben estar nombrados en inglés.
- ✓ Los bloques de código siempre deben estar encerrados por llaves (incluso si solo constan de una línea).
- ✓ El nombre de los métodos utilizan el estándar LowerCamelCase, CamelCase o DromedaryCase. Los cuales comienzan con la primera letra minúscula. En caso de ser un nombre compuesto se escribe el segundo nombre seguido del primero con la primera letra mayúscula y el resto en minúscula.
- ✓ No se usarán nombres de variables que coincidan con palabras reservadas.

### 3.3 Modelo de despliegue

El modelo de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse elementos de software. Se utiliza como entrada principal en las actividades de diseño e implementación, debido a que la distribución del sistema tiene una influencia principal en su diseño. (35)

## Capítulo 3: Implementación y prueba

### 3.3.1 Diagrama de despliegue (DD)

Un diagrama de despliegue muestra las relaciones físicas que se establecen entre componentes de software y hardware dentro de un sistema de cómputo determinado. Se modela a través de un conjunto de nodos y las relaciones existentes entre ellos. Cada nodo en este tipo de diagrama representa un tipo de unidad computacional, en la mayoría de los casos de tipo hardware. (36)

El diagrama que se presenta a continuación representa la distribución física del sistema Android para controlar ATcnea a través de nodos, está compuesto por la tableta del profesor, donde la comunicación entre ella y la laptop del profesor se realiza a través del protocolo TCP. Además se utiliza el protocolo UDP para realizar un escaneo de las clases que se encuentren activas y poder acceder a una de ellas.

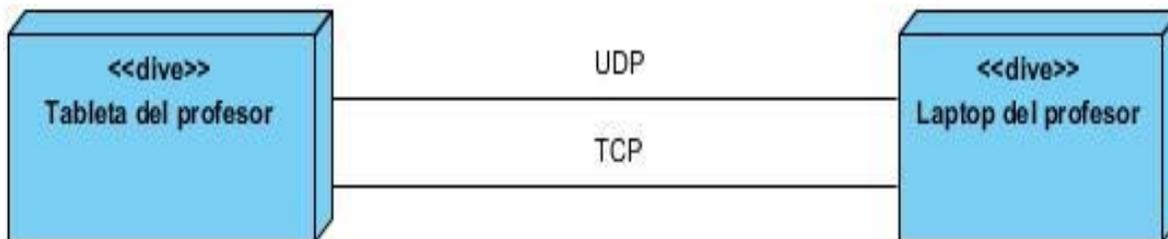


Figura 8: Diagrama de Despliegue de la propuesta de solución

### 3.4 Pruebas de software

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación. Una vez generado el código fuente es necesario probar el software para descubrir y corregir la mayor cantidad de errores posibles antes de ser entregado. Su objetivo es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores.

Algunos objetivos de las pruebas de software son:

- ✓ La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.

## Capítulo 3: Implementación y prueba

- ✓ Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- ✓ Una prueba tiene éxito si descubre un error no detectado hasta entonces. (24)

Las pruebas se aplican con diferentes propósitos y generalmente en distintos escenarios de trabajo. Con el objetivo de verificar el comportamiento y funcionamiento del sistema se decidió realizar pruebas unitarias, pruebas funcionales y pruebas de aceptación.

### 3.4.1 Pruebas unitarias

Las pruebas unitarias son el proceso de probar componentes del programa tales como métodos o clases de objetos. Las funciones o los métodos individuales son el tipo más simple de componente. Las pruebas deben llamarse para dichas rutinas con diferentes parámetros de entrada (25).

Para llevarla a cabo esta prueba el método aplicado fue el de caja blanca, donde las pruebas se enfocan en la estructura de control del programa. Los casos de prueba se derivan para asegurar que todos los enunciados en el programa se ejecutaron al menos una vez durante las pruebas y que todas las condiciones lógicas se revisaron (24). Con el fin de automatizar este tipo de pruebas sobre la solución se decidió emplear la herramienta JUnit, la cual está integrada con el IDE Android Studio que se utilizó para desarrollar la aplicación.

### 3.4.2 Pruebas funcionales

Las pruebas funcionales aseguran el apropiado trabajo de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Las metas de estas pruebas son verificar la apropiada aceptación de datos y verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio (24).

Para llevar a cabo esta prueba el método empleado fue el de caja negra, el cual se centra en los requisitos funcionales del software. Es decir, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (24).

## Capítulo 3: Implementación y prueba

Como técnica se utilizó los casos de pruebas diseñadas según las funcionalidades descritas en las historias de usuario. La intención que se persigue con estos artefactos es lograr una comprensión específica de las condiciones que la solución debe cumplir. Cada planilla de casos de pruebas recoge la especificación de una historia de usuario, dividida en secciones y escenarios, detallando las funcionalidades descritas en ella y describiendo cada variable.

A continuación, se muestra el caso de prueba correspondiente a la HU Iniciar Sesión. Los casos de pruebas asociados al resto de las historias de usuarios son definidos en los anexos.

Escenario	Descripción	contraseña	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción aceptar.	Selecciona la opción aceptar.	V	Valida los datos. Muestra el listado de los estudiantes.	Seleccionar clase/iniciar sesión/aceptar.
<b>EC 1.2</b> Opción cancelar.	Selecciona la opción cancelar.	V	Regresa a la interfaz que le permite seleccionar una clase.	Seleccionar clase/iniciar sesión/cancelar.
<b>EC 1.3</b> Datos incompletos.	Existen datos incompletos.	I	Muestra un mensaje de información. Muestra un indicador sobre los campos vacíos. Regresa al EC 1.1.	Seleccionar clase/iniciar sesión/aceptar.
<b>EC 1.4</b> Datos incorrectos.	Existen datos incorrectos.	I	Muestra un mensaje de información. Muestra un indicador sobre los campos incorrectos. Regresa al EC 1.1.	Seleccionar clase/iniciar sesión/aceptar.

**Figura 8:** Caso de Prueba Iniciar Sesión

## Capítulo 3: Implementación y prueba

### 3.4.3 Pruebas de aceptación

Esta es la etapa final en el proceso de pruebas, antes de que el sistema se acepte para uso operacional. El sistema se pone a prueba con datos suministrados por el cliente del sistema, en vez de datos de prueba simulados. Las pruebas de aceptación revelan los errores y las omisiones en la definición de requerimientos del sistema, ya que los datos reales ejercitan el sistema en diferentes formas a partir de los datos de prueba. Asimismo, las pruebas de aceptación revelan problemas de requerimientos, donde las instalaciones del sistema en realidad no cumplan las necesidades del usuario o cuando sea inaceptable el rendimiento del sistema (25).

Para las pruebas de aceptación se montó una clase con todos los requerimientos, donde intervienen todos los componentes del aula tecnológica de manera que pudiera ser posible probar en acción todas las funcionalidades. Se montaron diez puestos de trabajo para estudiantes (tabletas) y para el profesor además de la laptop se utilizó una tableta con el software implementado.

### 3.4.4 Resultados de las pruebas:

Con la aplicación de las pruebas unitarias y funcionales fueron arrojados los siguientes resultados:

**Pruebas unitarias:** Permitieron comprobar el correcto funcionamiento del código, conocer la memoria consumida y el tiempo de ejecución al realizar la prueba, aunque estos resultados variaron de un dispositivo a otro. Estas pruebas se realizaron mediante la extensión Android JUnit, que es una parte del SDK<sup>12</sup> de Android el cual permite probar funciones específicas mediante clases de casos de pruebas. Estas clases proporcionan métodos auxiliares para la creación de objetos de imitación y métodos que ayudan a controlar el ciclo de vida de una función.

---

<sup>12</sup> **SDK:** kit de desarrollo de software por sus siglas en inglés software development kit

## *Capítulo 3: Implementación y prueba*

**Pruebas funcionales:** Los problemas detectados se clasificaron en: No conformidades significativas (NCS) y en No conformidades no significativas (NCNS). A continuación, se describen los aspectos que se tienen en cuenta en cada clasificación:

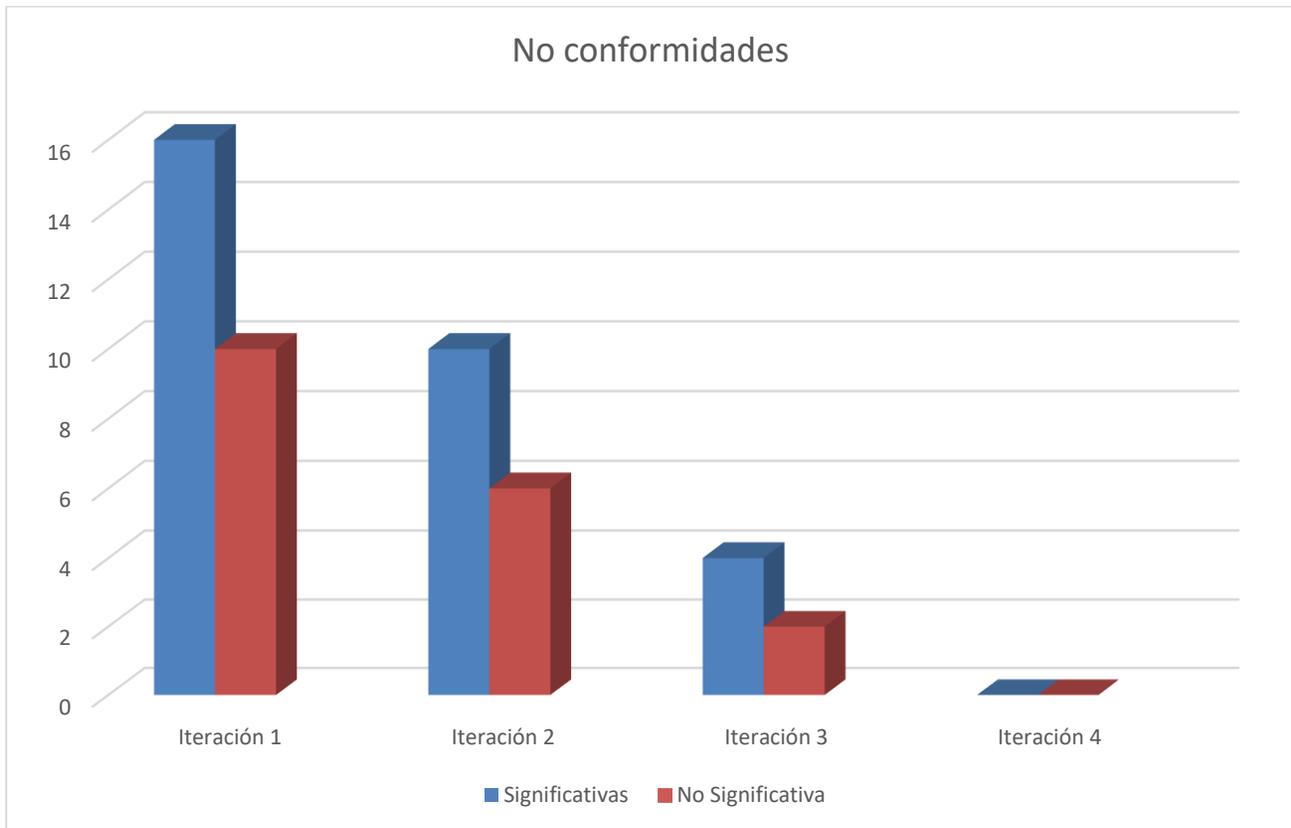
NCS: son las no conformidades en cuanto a las funcionalidades del componente: validaciones incorrectas o respuestas del sistema diferentes a lo descrito previamente en las historias de usuarios.

NCNS: son las no conformidades en cuanto al diseño de la propuesta de solución y errores ortográficos.

Las pruebas funcionales fueron realizadas en 4 iteraciones, donde se encontraron errores de interfaz y de funcionalidades incorrectas. En la primera iteración se detectaron un total de 26 no conformidades, de ellas 16 NCS y 10 NCNS. En la segunda iteración se detectaron un total de 16 no conformidades, de ellas 10 NCS y 6 NCNS. En la tercera iteración solo se detectaron un total de 6 no conformidades, de ellas 5 NCS y 1 NCNS. Además, se realizó una cuarta iteración donde no se encontraron no conformidades. En cada una de las iteraciones fueron resueltas de forma satisfactoria los errores detectados.

## Capítulo 3: Implementación y prueba

A continuación, se muestra un gráfico de los resultados obtenidos tras la realización de las pruebas funcionales, por iteraciones se observa el total de no conformidades significativas identificadas y el total de no conformidades no significativas detectadas.



**Figura 10:** Resultados de las pruebas funcionales

**Pruebas de aceptación:** Para llevar a cabo las pruebas de aceptación se hizo necesario, de a través del tipo de prueba alfa, incorporar al cliente o usuario final directamente al proceso de prueba de la aplicación. Entiéndase por prueba alfa cuando esta se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales, es decir, en un ambiente controlado, propiciando que el desarrollador pueda registrar errores y problemas de uso. (25)

## Capítulo 3: Implementación y prueba

Como parte de esta prueba se diseñó un entorno que simule la distribución de un aula convencional, con los dispositivos del aula tecnológica. Se impartió una clase previamente preparada donde se pudieran realizar las principales funcionalidades de ATcnea y con la condición de que el profesor se movería entre los puntos extremos del aula. El siguiente esquema muestra la disposición prevista donde A y B simbolizan los puntos extremos.

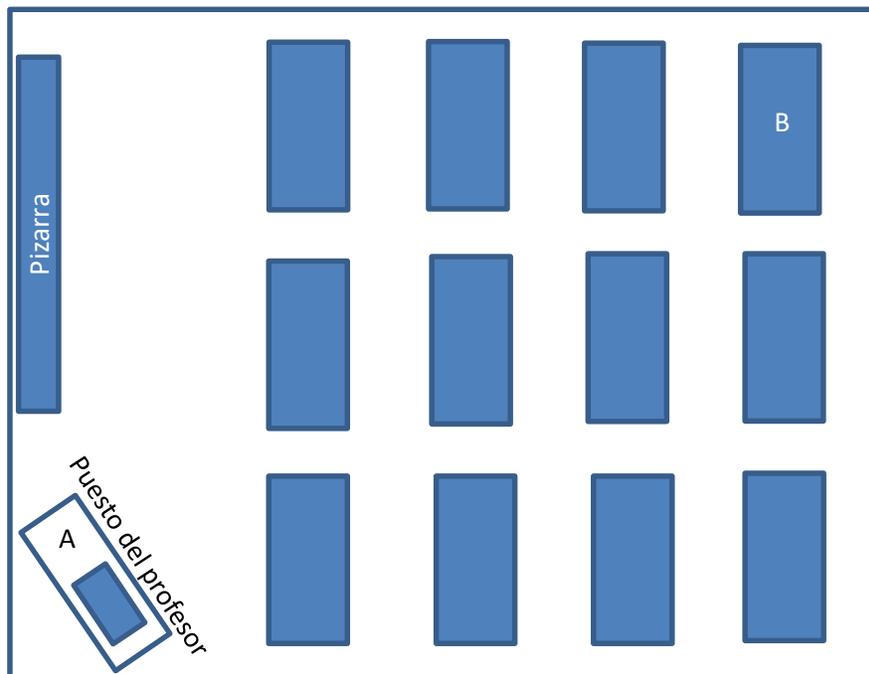


Figura 11: Disposición del aula para la prueba alfa

Para la ejecución de esta prueba de aceptación de tipo alfa se utilizaron los siguientes componentes como parte del hardware y software del aula tecnológica:

- ✓ Tableta Android del estudiante:
  - Sistema operativo Novadroid 1.0 (personalización de Android 6.0).
  - RAM 1GB.
  - CPU/GPU Dual Core 1.0GHz.
- ✓ Laptop del profesor:
  - RAM 2GB.

## *Capítulo 3: Implementación y prueba*

- CPU Core 2 E6300.
- Tarjeta de video 256Megabytes.
- Red alámbrica a 10Megabytes/100Megabytes/1000Megabytes
- Red compartida o Red conmutada y red inalámbrica a 802.11b/g/n.
- Sistema Operativo Nova 2015.
- Cable de conexión VGA (al video proyector).
- Cable de conexión USB (pantalla interactiva).
- ✓ Tableta Android del profesor:
  - Sistema operativo Novadroid 1.0 (personalización de Android 6.0).
  - RAM 1 GB.
  - CPU/GPU Dual Core 1.0GHz.
- ✓ Software ATcnea
  - Aplicación de escritorio en la Laptop del profesor
  - Aplicación para los estudiantes en sus respectivas tabletas
  - Aplicación de control en la tableta del profesor

Durante el desarrollo de la prueba el profesor pudo realizar todas las acciones que tenía prevista como parte del plan de clases. Utilizó la pizarra, las posibilidades de ATcnea desde la laptop y pudo moverse libremente por el aula ejecutando acciones desde la tableta. La clase transcurrió de manera correcta y no se apreciaron interrupciones ni demoras en las actividades debido al que el profesor estuviera alejado de la laptop atendiendo a algún estudiante en particular.

Durante esta prueba se evidenció que el profesor puede ejecutar todas las funcionalidades prevista en la aplicación de control de ATcnea desde cualquier punto del aula y monitorear las acciones de sus estudiantes al tiempo que atiende dudas o se mueve por los diferentes puestos.

## *Capítulo 3: Implementación y prueba*

### **3.5 Conclusiones parciales**

A partir del desarrollo del presente capítulo se arribaron a las siguientes conclusiones:

- La descripción del proceso de implementación de la aplicación, a través de la definición de las convenciones utilizadas para la codificación, posibilitó una mejor legibilidad del código, haciéndolo más comprensible y estandarizado.
- Las pruebas de caja blanca y caja negra permitieron comprobar el correcto funcionamiento del código de la aplicación, validar la completitud de los requisitos y determinar la aceptación por parte del cliente, respectivamente.

### **Conclusiones generales**

A raíz de los resultados obtenidos y tomando como base el cumplimiento de los objetivos trazados al iniciarla, se puede arribar a las siguientes conclusiones:

- Los métodos científicos utilizados permitieron definir los principales conceptos y teorías que sustentan la investigación.
- El análisis de la arquitectura y la aplicación de patrones de diseños permitieron comprender los aspectos relacionados con los requisitos del sistema.
- El diseño realizado permitió desarrollar una aplicación que cumpliera con las funcionalidades definidas y que dan solución al objetivo planteado.
- Las pruebas realizadas permitieron demostrar la correcta implementación de las funcionalidades.
- Se obtuvo una aplicación Android para controlar el software ATcnea que aumenta la movilidad del profesor en el aula.

### **Recomendaciones**

A partir del trabajo realizado y después de haber analizado los resultados obtenidos se recomienda:

- Desarrollar una versión de la aplicación para sistemas operativos IOS.
- Implementar funcionalidades de interacción con la pizarra.

## Referencias

1. **Touriñán Lopez Jose Manuel, Sáez Alonso.** *Revista de Investigación en Educación.*
2. **Cirigliano, Villaverde , Anibal y Villaverde, Gustavo.** *Dinámica de grupos y la educación.*
3. **Castañeda Hevia, Emilio.** El Papel de las Tecnologías de la Información y las Comunicaciones (TICs) en el Proceso de Enseñanza-Aprendizaje a Comienzos del Siglo XXI. *Preparación Pedagógica Integral para Profesores Universitarios.* La Habana : Felix Varela.
4. *Los Entornos Personales de Aprendizaje.* **Pérez Mallea, Iván.** La Habana : UCI, 2015.
5. *Aulas inteligentes, Smart classrooms.* **Vidal Ledo, MSc. María Josefina; Morales Suárez, Dra. Ileana; Rodríguez, MSc. Rosa Moraima;** 2, Ciudad de la Habana : s.n., 2014, Vol. 28.
6. *Proyección del Aula Informátizada en la Escuela Cubana.* **Ortega Cabrera, Fernández.** La Habana : Actas del Congreso de Informática en la Educación. INFOREDU, 2015.
7. **Rosario Maritza Del Carmen, Sánchez Perales Milagros.** *La mejora de los aprendizajes desde el "Aula.* 2014.
8. **Alonso.** Ambientes Colaborativos. [En línea] 2012.  
<http://ambientescolaborativo.blogspot.com/2012/10/definicion-de-ambiente-colaborativo.html>.
9. **Santiago, Raúl y Trbaldo , Susana.** *Nuevas realidades en el aula.* s.l. : OCEANO, 2015.
10. **Qué es y cómo usar un software de gestión de aula.** [En línea]  
<https://pt.slideshare.net/MERITXELLVP/qu-es-y-cmo-usar-un-software-de-gestin-de-aula-120711-8663050>.
11. **Ochoa , Sergio F y Gruia, Roman.** *Advanced Software Engineering: Expanding the Frontiers of Software.* 2012.
12. **Casad, Joe.** *TCP/IP.* [ed.] Anaya Multimedia. 2012.
13. **Netop Vision Pro.** [En línea]
14. **Stephen , Kaisler.** *Software Paradigms.* 2010.

15. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *Proceso Unificado de Desarrollo de Software*. 2010.
16. Visual Paradigm. *Visual Paradigm. Full-Featured UML Software Design Tool*. [En línea] <http://www.visual-paradigm.com>.
17. CCM. *Visual Paradigm for UML Enterprise Edition* . [En línea] <http://es.ccm.net/download/descargar-28127-visual-paradigm-for-uml-enterprise-edition..>
18. JAVA. Que es Java. [En línea] <http://www.infor.uva.es/~jmrr/tgp/java/JAVA.html>.
19. An Introduction to NetBeans. Welcome to the NetBeans. [En línea] <https://netbeans.org>.
20. Android Developers , Android Studio Overview. [En línea] <http://developer.android.com/intl/es/tools/studio/index.html>.
21. Android Developers , Android Studio and SDK Tools. *Android Developers. Download Android Studio and SDK Tools*. [En línea] <http://developer.android.com>.
22. JUnit - About. [En línea] <http://junit.org/junit4/>.
23. Hans-Erik Eriksson, Magnus Penker. *Business Modeling with UML: Business Patterns at Work*.
24. Pressman, Roger S. *Ingeniería del Software:Un Enfoque Práctico*. 7. 2010.
25. Sommerville, Iam. *Ingeniería de Software Séptima Edición*. Madrid : Pearson Educación, 2009.
26. Oracle.NetBeans. [En línea] Corporation. <https://netbeans.org/indexes.html>.
27. [En línea] [http://librosweb.es/bootstrap\\_3/](http://librosweb.es/bootstrap_3/)..
28. Lou, T. *A comparison of Android Native App Architecture MVC, MVP and MVVM*. 2016.
29. Gamma , Erich, Helm, Richard y Ralph, Johnson. *Design Patterns : Elements of Reusable Object-Oriented Software*. 2013.
30. Larman, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. s.l. : Pearson Educación, 2003.

31. —. **UML y Patrones**. La Habana : Felix Valera, 2009.
32. —. **UML y patrones**. s.l. : Pearson, 2009.
33. León, Mijaíl, VIDAL, Manuel y LINARES, Juan. Diagrama de clases. **UNIVERSIDAD BICENTENARIA DE ARAGUA VICERRECTORADO ACADÉMICO ESCUELA DE INGENIERIA TURMERO– ESTADO ARAGUA**. [En línea]  
<http://es.slideshare.net/nedowwhaw/diagrama-de-clases-16208245>.
34. Microsoft. **Revisiones de código y estándares de codificación**. [En línea]  
[https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx).
35. Jacobson, Grady Booch James Rumbaugh Ivar. **El proceso unificado de desarrollo de software**. s.l. : Pearson Eduaction S.A.
36. Fowler, Martin. **UML Distilled: A Brief Guide to the Standarad Object Modeling Language**. Second.
37. Iam, Sommerville. **Ingeniería de Software Séptima Edición**. Madrid : Pearson Educación, 2005.
38. **Proceso de pruebas para productos de software en un laboratorio de calidad**. Nuñez Jústiz Dalila, Suárez Gómez Darlene, Dapena Delgado Marta Dunia. 2014.

## Anexos

### Anexos #1:

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
Construcción		
Transición		
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

**Tabla 1.1: Fases AUP vs AUP-UCI**

Disciplinas AUP	Disciplinas Variación AUP-UCI	Objetivos Disciplinas (Variación AUP-UCI)
Modelo	Modelado de negocio	El Modelado del Negocio es la disciplina destinada a comprender los procesos de negocio de una organización. Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes : <b>1 - Casos de Uso del Negocio (CUN).</b>

		<p><b>2- Descripción de Proceso de Negocio (DPN).</b></p> <p><b>3- Modelo Conceptual (MC).</b> A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina <b>Requisitos</b>.</p>
	Requisitos	<p>El esfuerzo principal en la disciplina Requisitos es desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)], agrupados en cuatro escenarios condicionados por el Modelado de negocio.</p>
	Análisis y diseño	<p>En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Además, en esta disciplina se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis.</p>
Implementación	Implementación	<p>En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema.</p>
Prueba	Pruebas interna	<p>En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba</p>

		como: diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas.
	Pruebas de liberación	Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
	Pruebas de Aceptación	Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.
Gestión de configuración	Se cubren con las áreas de procesos PP, PMC y CM que propone CMMI-DEV v1.3. Las mismas son áreas de proceso de gestión y soporte respectivamente.	Consultar en mejoras.prod.uci.cu los libros de procesos de cada una de estas áreas.
Gestión de proyecto		
Entorno		

**Tabla 1.2 Disciplinas de AUP vs AUP-UCI**

**Anexos #2: Descripción de las Historias de usuarios (HU)**

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Cerrar sesión
<b>Programador:</b> Oreste A. Nillar	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 3 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A
<b>Descripción:</b> <b>1- Objetivo:</b> Permitir al usuario cerrar sesión en el sistema.	

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para cerrar sesión en el sistema:

- El terminal del profesor debe estar conectado a la laptop

**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Desde la interfaz principal, el sistema debe permitir al profesor cerrar la sesión accediendo al icono que representa a la opción:

- Ⓢ Cerrar sesión

Una vez seleccionada la opción el sistema debe cerrar la sesión y mostrar la interfaz que le permite seleccionar nuevamente una clase. .

**Observaciones:**

- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.

**Prototipo de interfaz:**

Número: N/A	Nombre del requisito: Seleccionar clase
Programador: Oreste A. Nillar Cambara	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 4 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías	Tiempo Real: N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al profesor seleccionar una clase.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para seleccionar una clase debe: - El terminal del profesor debe estar conectado a la laptop del profesor.</p>	

- Debe existir al menos una clase activa.

**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

El sistema debe permitir seleccionar la clase activa deseada.

Una vez seleccionada la clase el sistema mostrará el listado de los estudiantes conectados a la misma y le brindará al profesor una serie de funcionalidades que le ayudará en el desempeño de la clase que se encuentran en el panel lateral izquierdo.

**Observaciones:**

- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.

**Prototipo de interfaz:**

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Impedir desconexión de la clase
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 8 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Impedir la desconexión de los estudiantes de la clase.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para impedir la desconexión de los estudiantes a la clase:</p> <ul style="list-style-type: none"> <li>- El terminal del profesor debe estar conectado a la laptop del profesor.</li> <li>- Debe haber al menos un estudiante conectado a la clase.</li> </ul>	

<p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> <p>El sistema debe permitir impedir que uno o varios estudiantes se desconecten de la clase seleccionando la opción:</p> <ul style="list-style-type: none"> <li>❖ Impedir desconexión.</li> </ul> <p>Una vez seleccionada la opción Impedir Desconexión asociada al estudiante, el sistema inhabilita el botón desconectarse en el terminal del estudiante seleccionado.</p>
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.</li> <li>- Una vez impedida la desconexión los estudiantes seleccionados no podrán desconectarse de la clase.</li> </ul>
<p><b>Prototipo de interfaz:</b></p>

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Buscar clase
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 24 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A

<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al profesor buscar las clases que se encuentran activas.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para buscar una clase</p> <ul style="list-style-type: none"><li>- El terminal del profesor debe estar conectado a la laptop del profesor.</li><li>- Debe haber al menos una clase creada.</li></ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b> El sistema debe permitir al profesor buscar una clase activa seleccionando el ícono que representa a la opción buscar clase ubicada en la parte inferior derecha. El sistema debe mostrar los nombres de las clases activas.</p>
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"><li>- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.</li><li>-</li></ul>
<p><b>Prototipo de interfaz:</b></p>

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Permitir la solicitud de estudiantes para conectarse a la clase
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 10 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al profesor aceptar o denegar la solicitud de estudiantes para conectarse a la clase.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para aceptar la solicitud del estudiante para conectarse a la clase: - El terminal del profesor debe estar conectado a la laptop del profesor. - El estudiante debe haber enviado la solicitud para conectarse a la clase.</p> <p><b>3- Flujo de la acción a realizar:</b> El sistema notificará al profesor que un estudiante ha solicitado conectarse a la clase a través de un número encima del ícono que representa a las solicitudes, que se encuentra en el panel lateral izquierdo Una vez que el profesor accede al ícono, el sistema muestra el nombre del terminal del estudiante que realizó la solicitud. El sistema permitirá seleccionar la opción Aceptar o Denegar. Luego notificará al estudiante que su petición ha sido aceptada o denegada a través de un mensaje de información.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.</li> </ul>	
<p><b>Prototipo de interfaz:</b></p>	

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Desconectar estudiantes de la clase
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 6 horas

<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al profesor desconectar estudiantes de la clase.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para desconectar estudiantes de la clase:</p> <ul style="list-style-type: none"> <li>- El terminal del profesor debe estar conectado a la laptop del profesor.</li> <li>- Debe haber al menos un estudiante conectado a una clase.</li> <li>- Debe haber un estudiante seleccionado.</li> <li>- El sistema debe estar en la interfaz listar estudiante.</li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b> Luego de seleccionar un estudiante El sistema debe permitir seleccionar la opción:</p> <ul style="list-style-type: none"> <li>❖ Desconectar estudiante</li> </ul> <p>Una seleccionada el estudiante se desconectará automáticamente.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.</li> </ul>	
<b>Prototipo de interfaz:</b>	

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Enviar examen a los estudiantes
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 6 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia	<b>Tiempo Real:</b> 6 horas

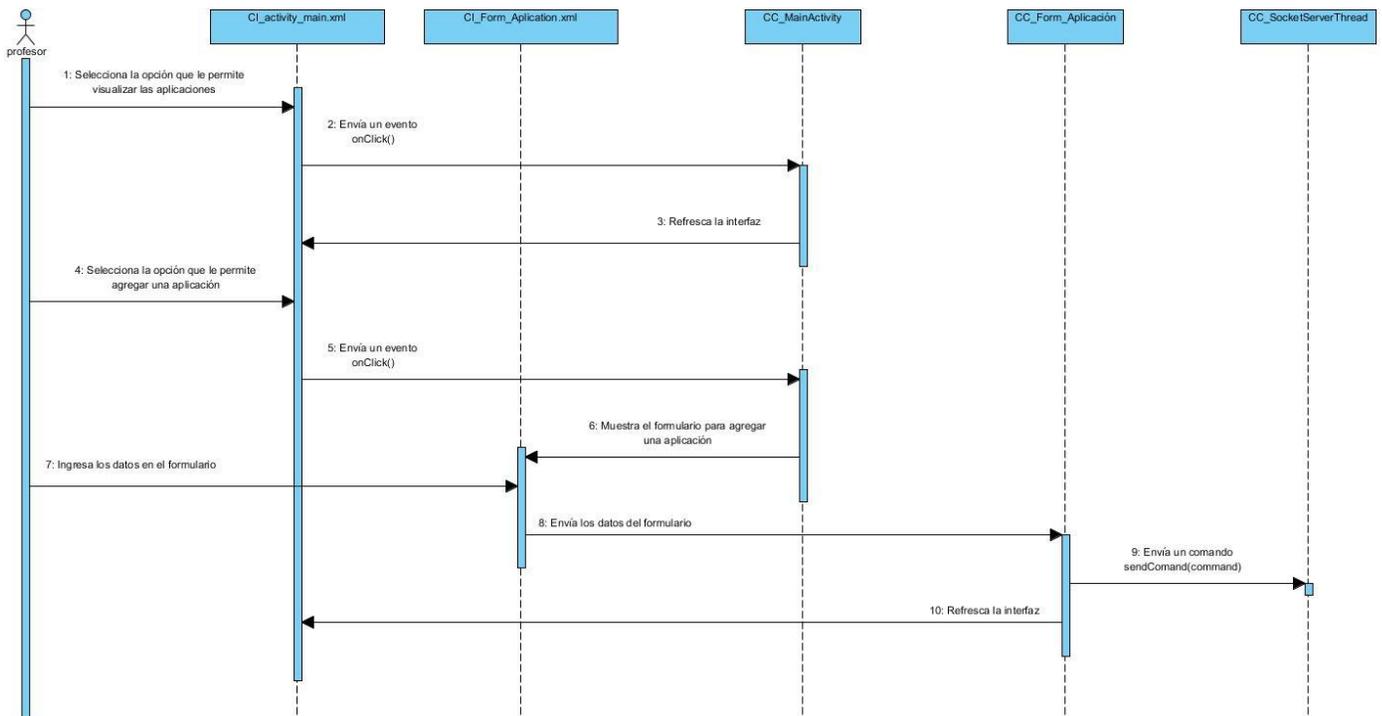
<p>con el desarrollo de las herramientas y tecnologías.</p>	
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir al profesor enviar un examen a los estudiantes</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para enviar un examen a los estudiantes:</p> <ul style="list-style-type: none"> <li>- El terminal del profesor debe estar conectado a la laptop del profesor.</li> <li>- Debe haberse creado al menos un examen.</li> <li>- Debe haber estudiantes conectados a la clase.</li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b> El sistema debe permitir al profesor seleccionar el o los estudiantes a los cuales desea enviar un examen. Una vez seleccionados los estudiantes el profesor debe seleccionar la opción Enviar examen que se encuentra en las opciones masivas en la barra lateral.</p> <p>El sistema debe permitir seleccionar el examen y enviárselo a los estudiantes seleccionados</p> <p>El sistema mostrará en el terminal de los estudiantes seleccionados el examen enviado por el profesor.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop</li> </ul>	
<p><b>Prototipo de interfaz:</b></p>	



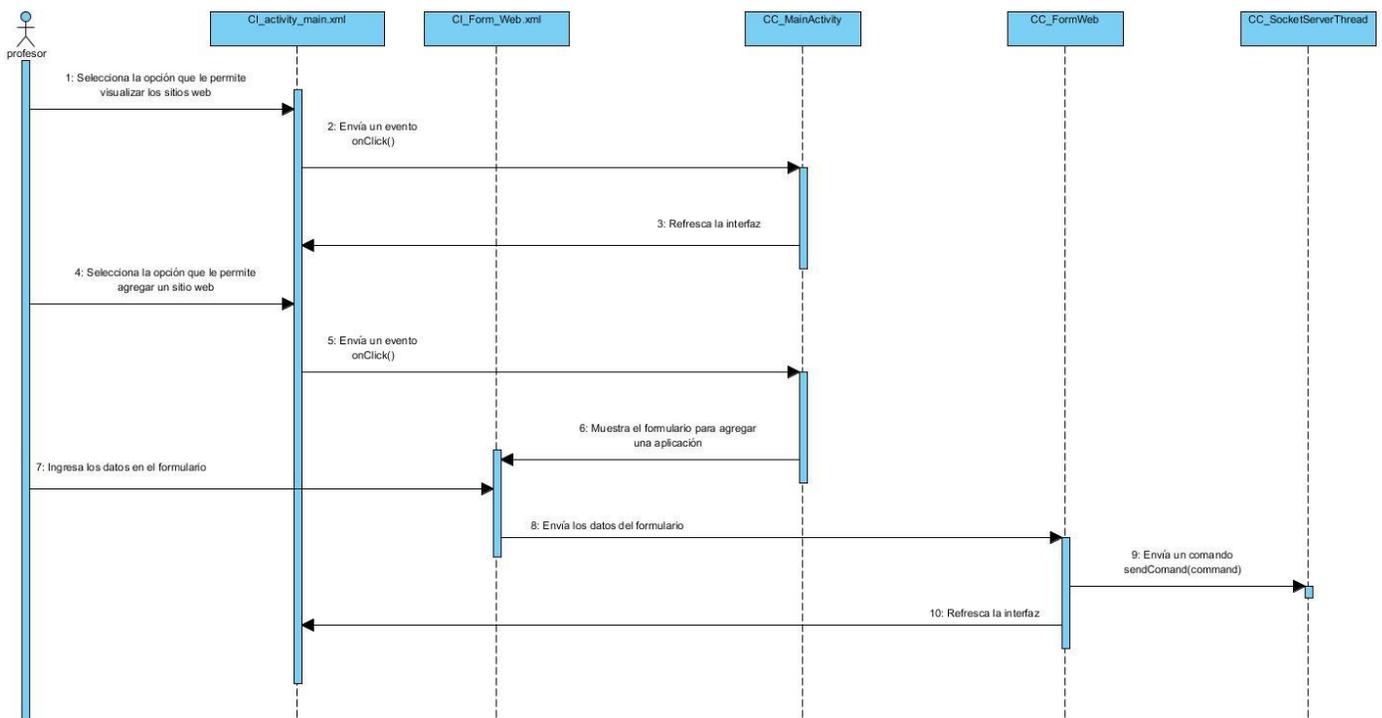
<b>Número:</b> N/A	<b>Nombre del requisito:</b> Permitir manos levantadas
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 7 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir aceptar o denegar manos levantadas.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para aceptar o denegar manos levantadas: - El terminal del profesor debe estar conectado a la laptop del profesor. - Debe haber al menos un estudiante seleccionado. - Debe existir al menos un estudiante que haya realizado la acción levantar mano.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b> El sistema notificará al profesor que un estudiante ha solicitado levantar la mano a través de un número encima del ícono que representa a las solicitudes en el sistema, que se encuentra en el panel izquierdo. Una vez que el profesor accede al ícono, el sistema muestra el nombre del terminal del estudiante que realizó la acción. El sistema permitirá seleccionar la opción Aceptar o Denegar. Una vez seleccionada una de las dos opciones se notificará al estudiante que su petición ha sido aceptada o denegada a través de un mensaje de información.</p>	
<p><b>Observaciones:</b></p> <ul style="list-style-type: none"> <li>- La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.</li> </ul>	
<p><b>Prototipo de interfaz:</b></p>	

<b>Número:</b> N/A	<b>Nombre del requisito:</b> Visualizar pantalla de un estudiante en el terminal del profesor
<b>Programador:</b> Oreste A. Nillar Cambara	<b>Iteración Asignada:</b> 1era
<b>Prioridad:</b> Media	<b>Tiempo Estimado:</b> 5 horas
<b>Riesgo en Desarrollo:</b> Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	<b>Tiempo Real:</b> N/A
<p><b>Descripción:</b></p> <p><b>1- Objetivo:</b> Permitir visualizar la pantalla de un estudiante en el terminal del profesor.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> Para ver la pantalla de un estudiante en el terminal del profesor: - Debe existir al menos un estudiante conectado a la clase. - El terminal del profesor debe estar conectado a la laptop del profesor.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b> El sistema debe permitir al profesor visualizar la pantalla del terminal del estudiante seleccionando la opción:  <input type="radio"/> Visualizar pantalla.            Una vez seleccionados los estudiantes el sistema debe permitir seleccionar la opción:  <input type="radio"/> Ver            Al seleccionar la opción Ver el sistema permite al profesor ver la pantalla del estudiante seleccionado en su laptop.</p> <p><b>Observaciones:</b> La conexión entre el terminal utilizado por el profesor y la laptop será a través del envío de comandos. Por su parte la comunicación entre los terminales de los estudiantes y el terminal del profesor será a través de la laptop.</p> <p><b>Prototipo de interfaz:</b></p>	

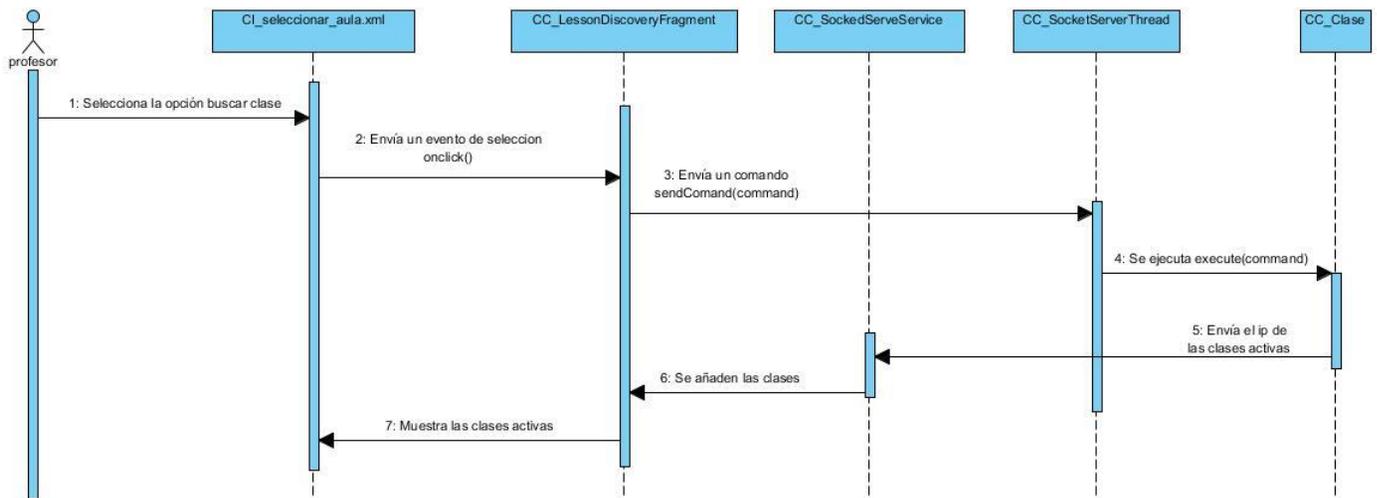
**Anexos #3: Diagramas de Secuencias del Diseño (DSD)**



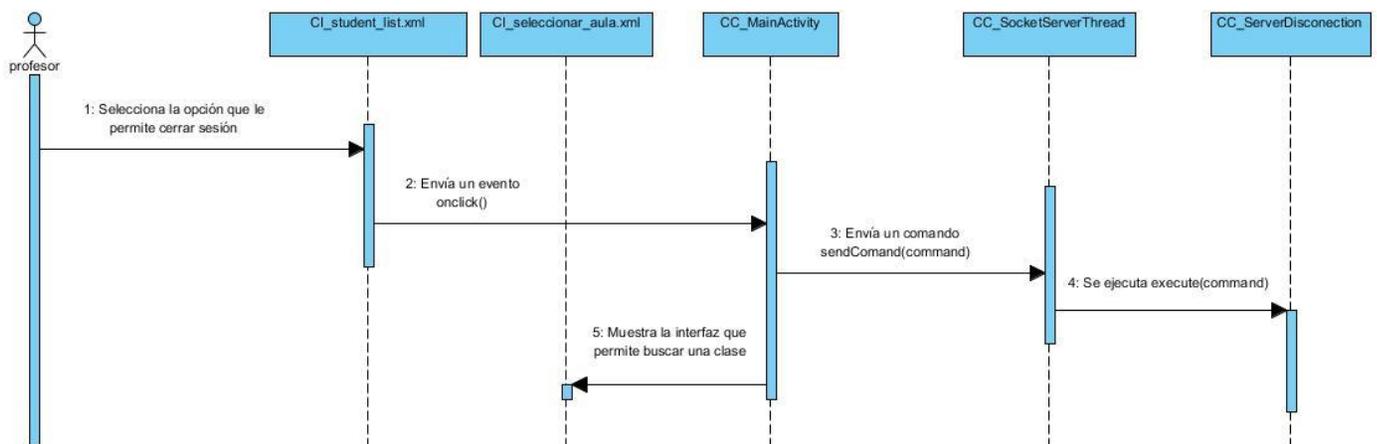
DSD Agregar\_aplicación.



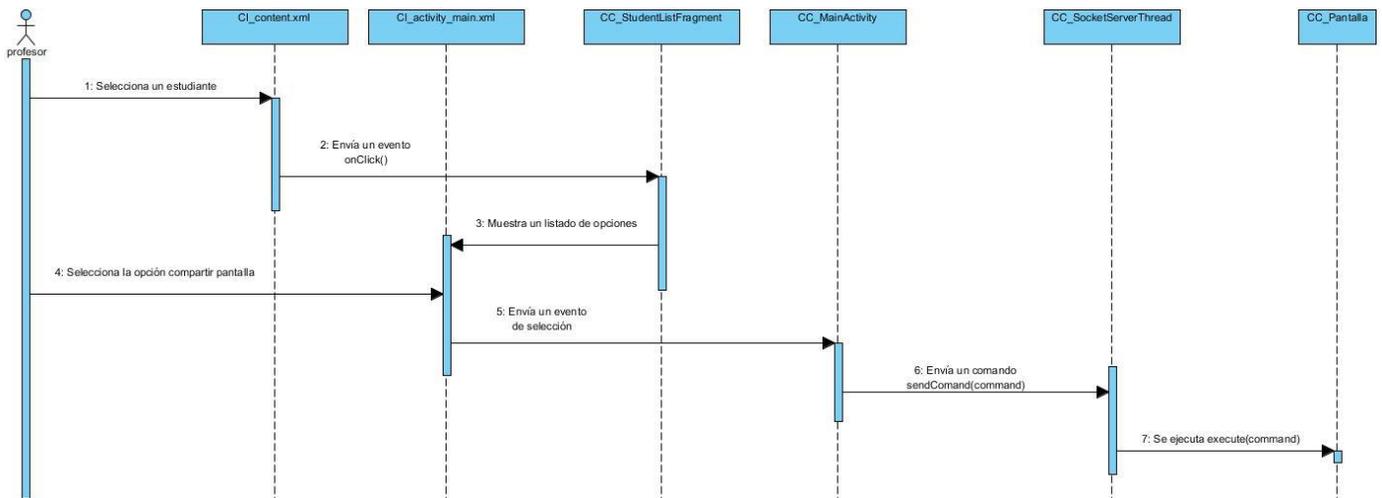
DSD Agregar\_sitio\_web.



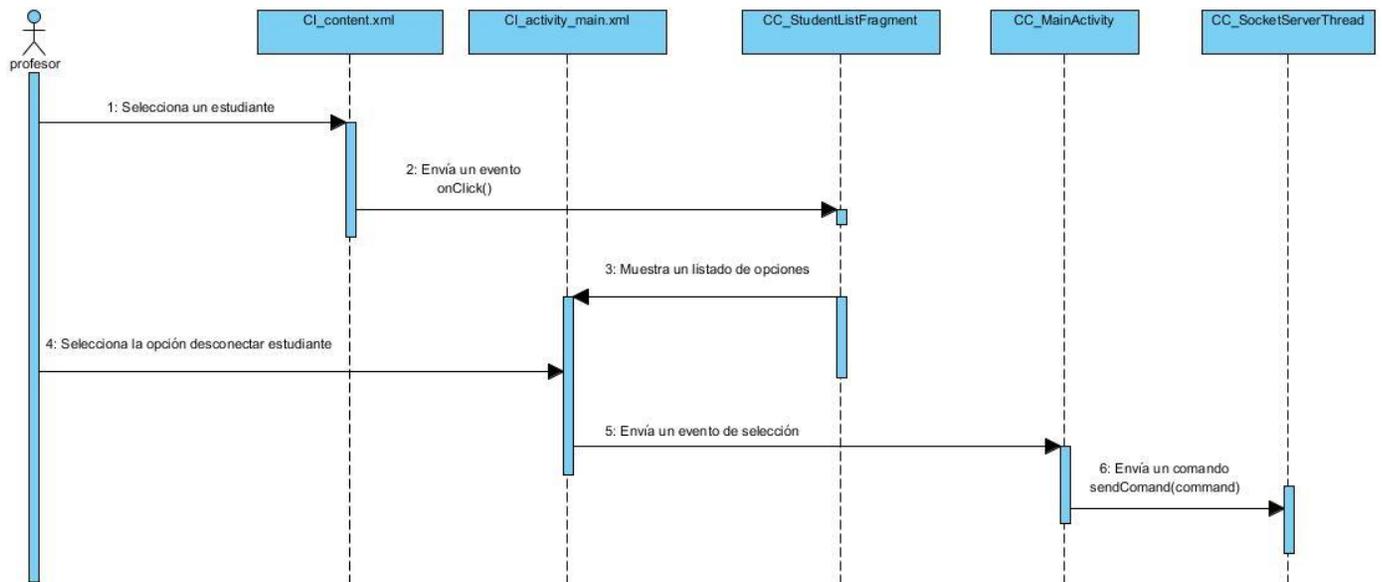
DSD Buscar\_clase.



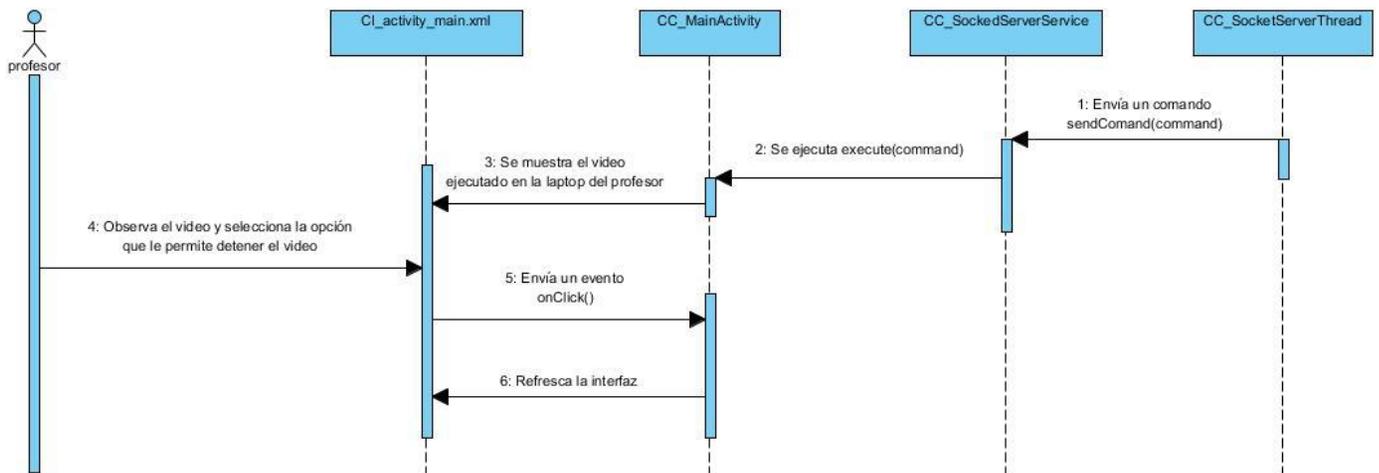
DSD Cerrar\_sesión.



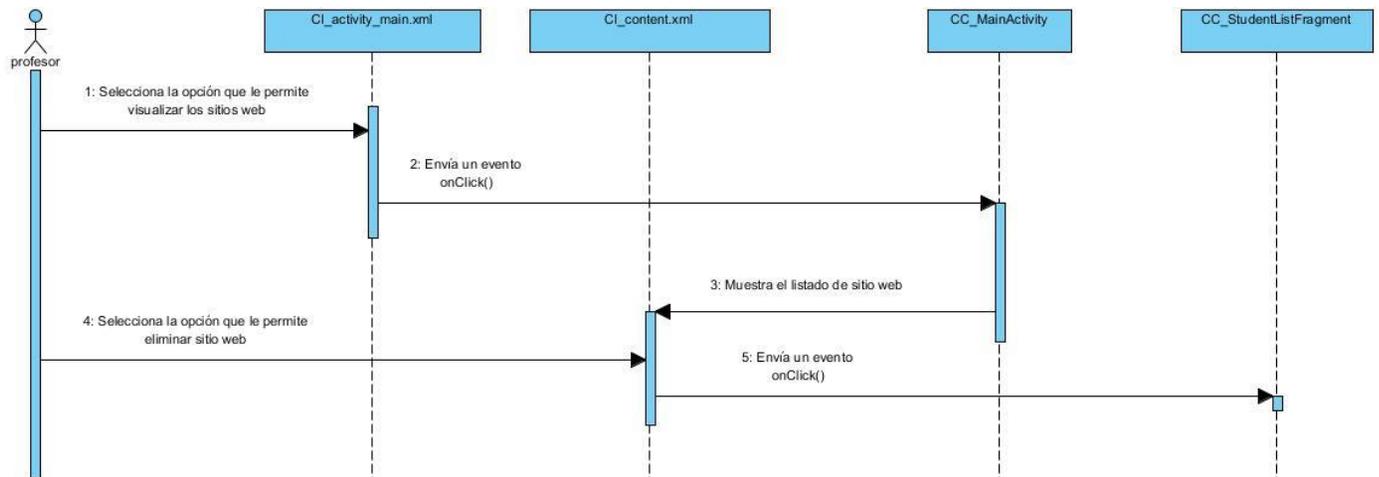
**DSD Compartir\_pantalla.**



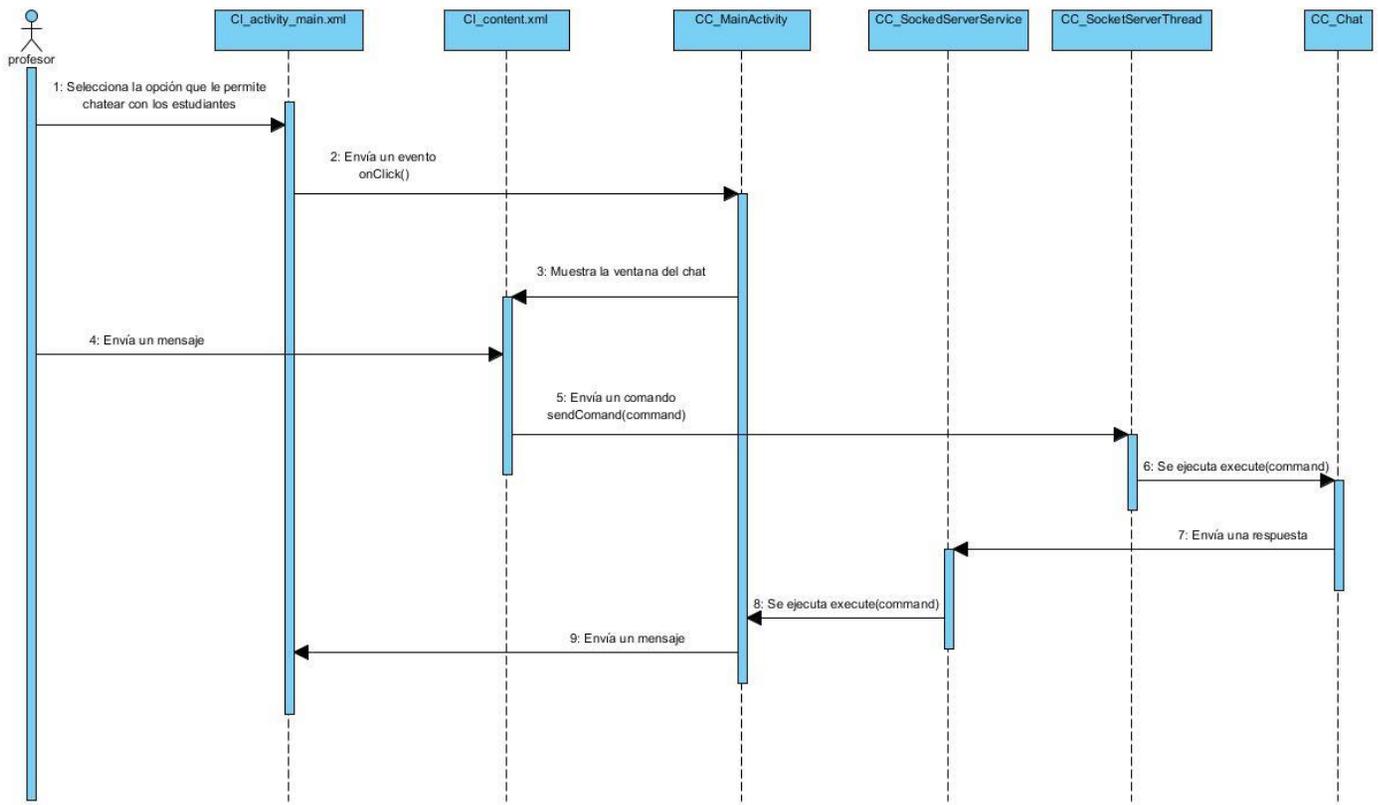
**DSD Desconectar\_estudiante.**



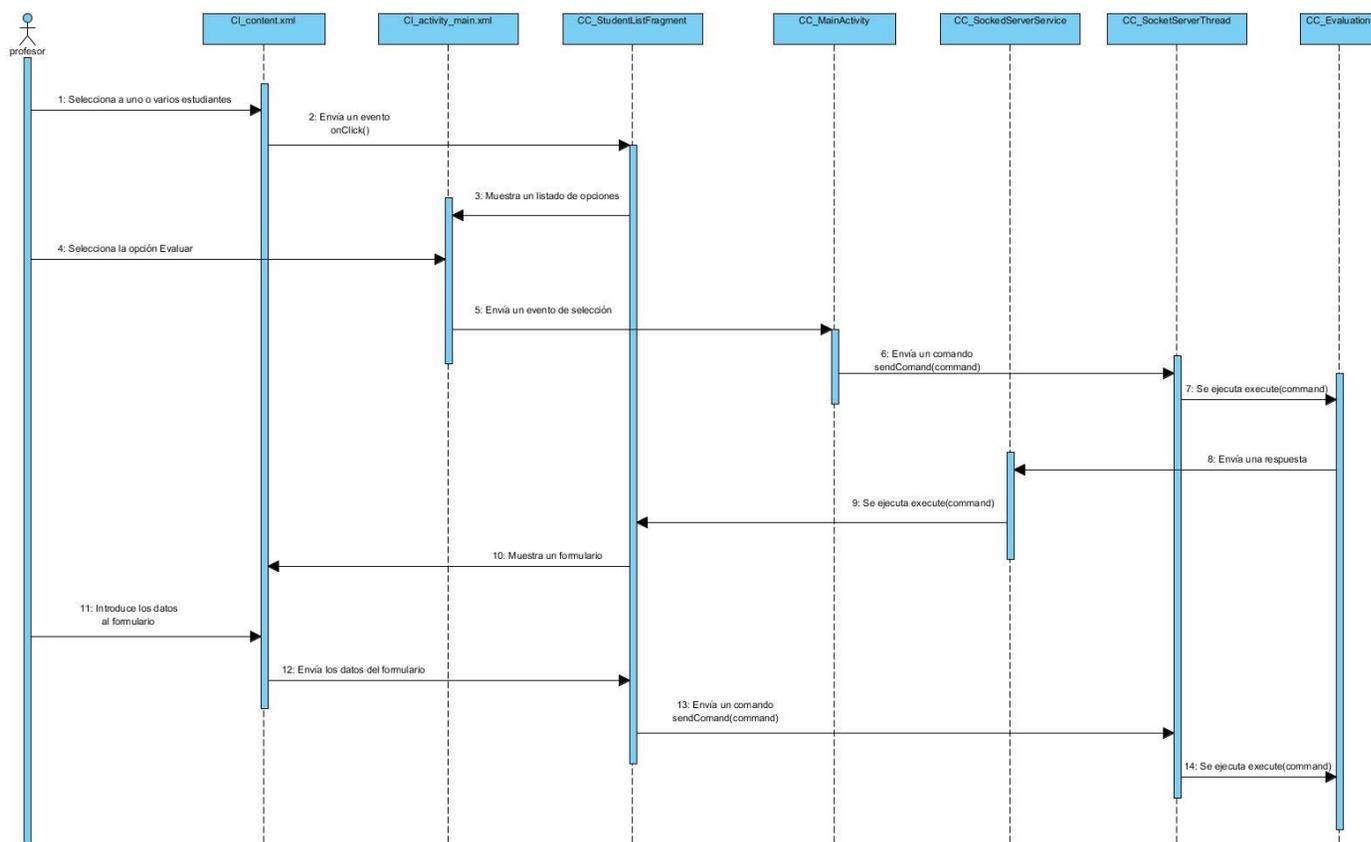
DSD Detener\_video.



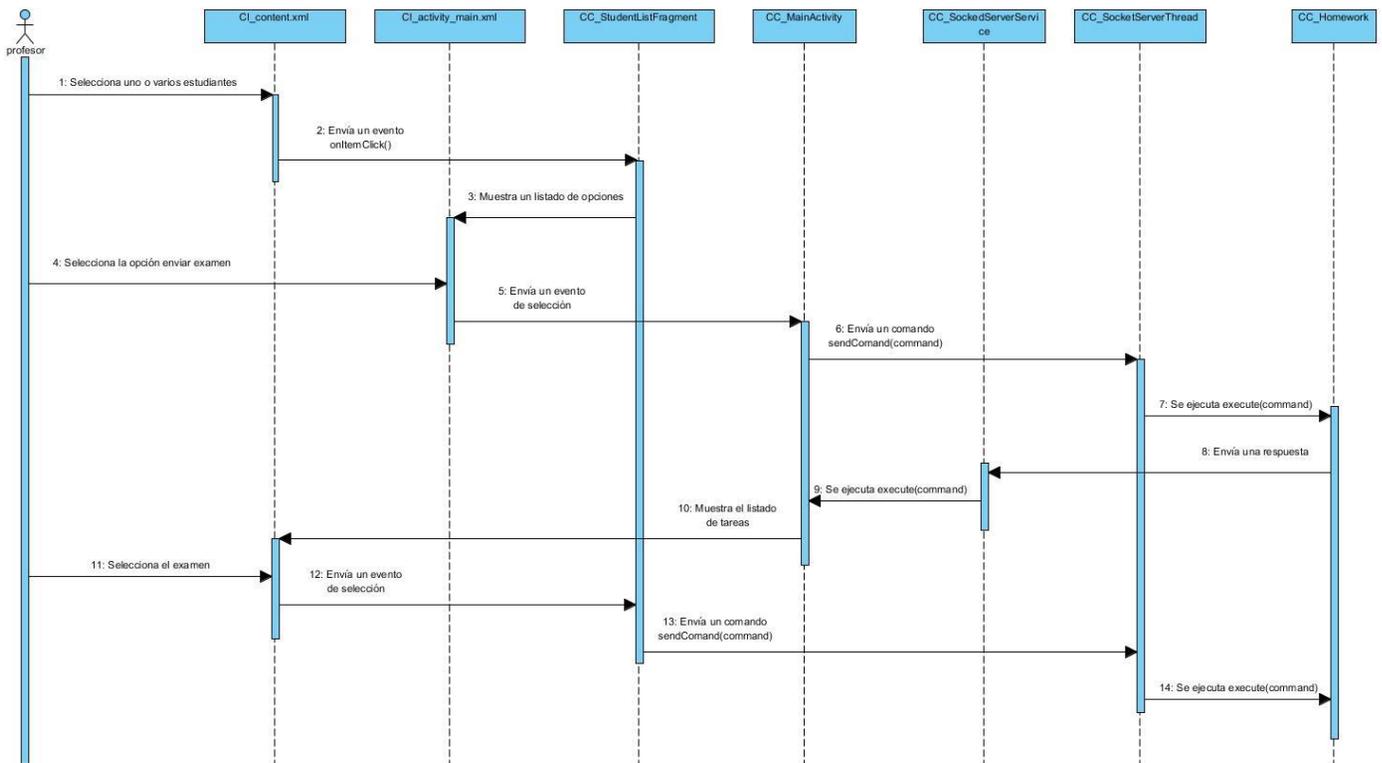
DSD Eliminar\_sitio\_web.



DSD Establecer\_conversación.

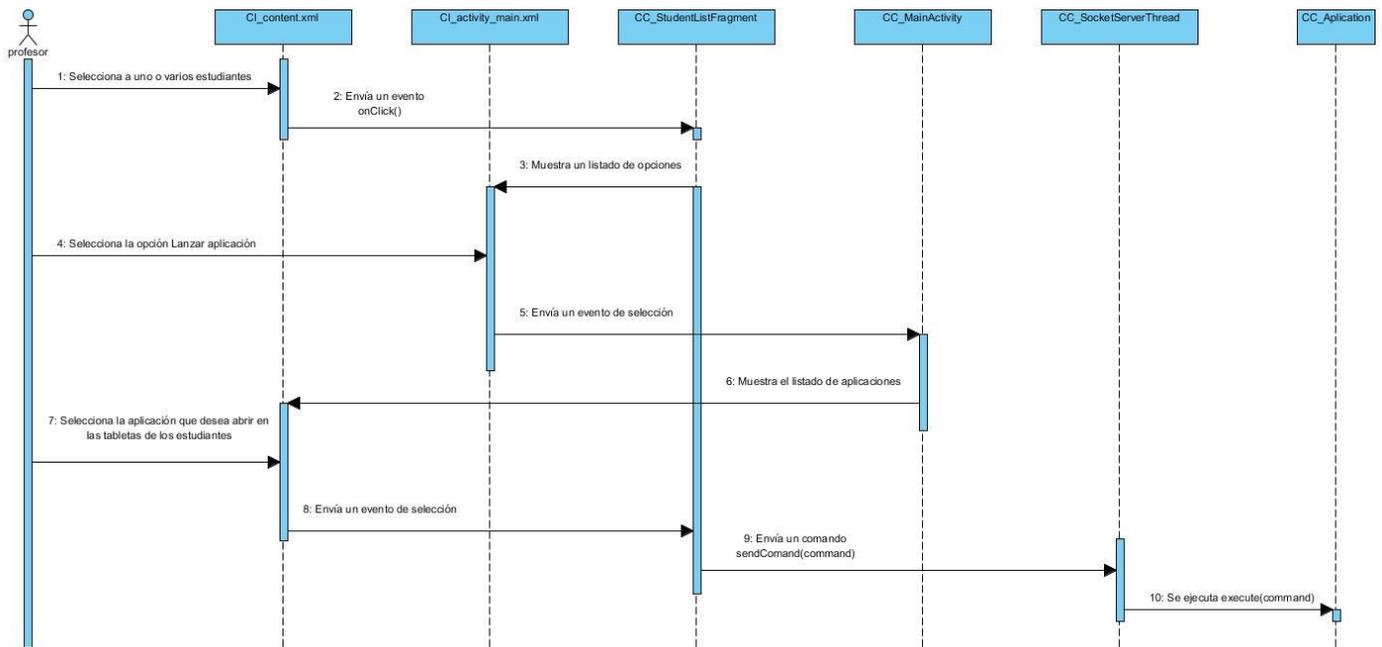


DSD Evaluación.

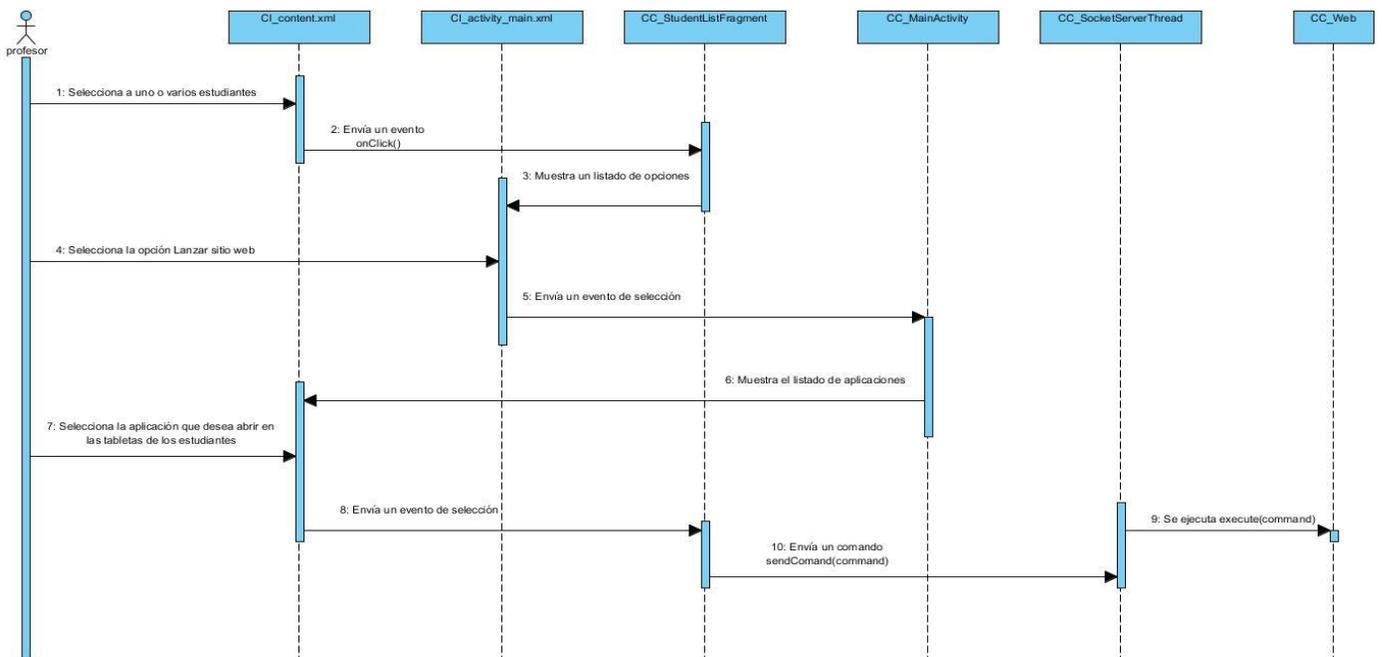


DSD Examen.<sup>13</sup>

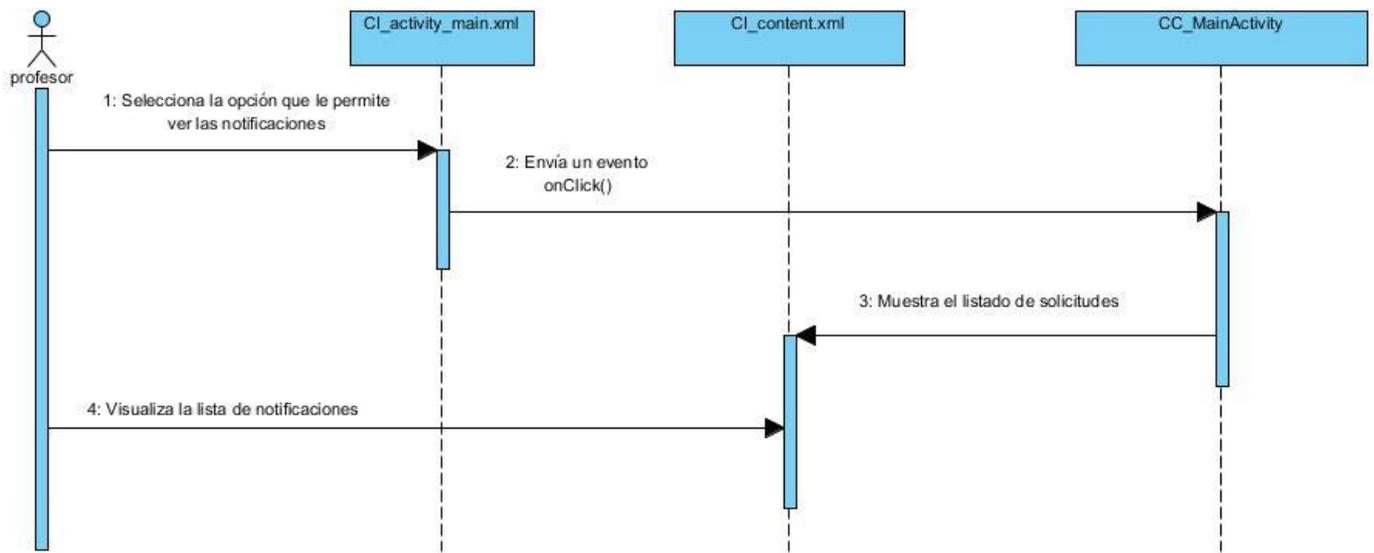
<sup>13</sup> DSD Examen: HU\_Ver listado de exámenes, HU\_Enviar examen a los estudiantes.



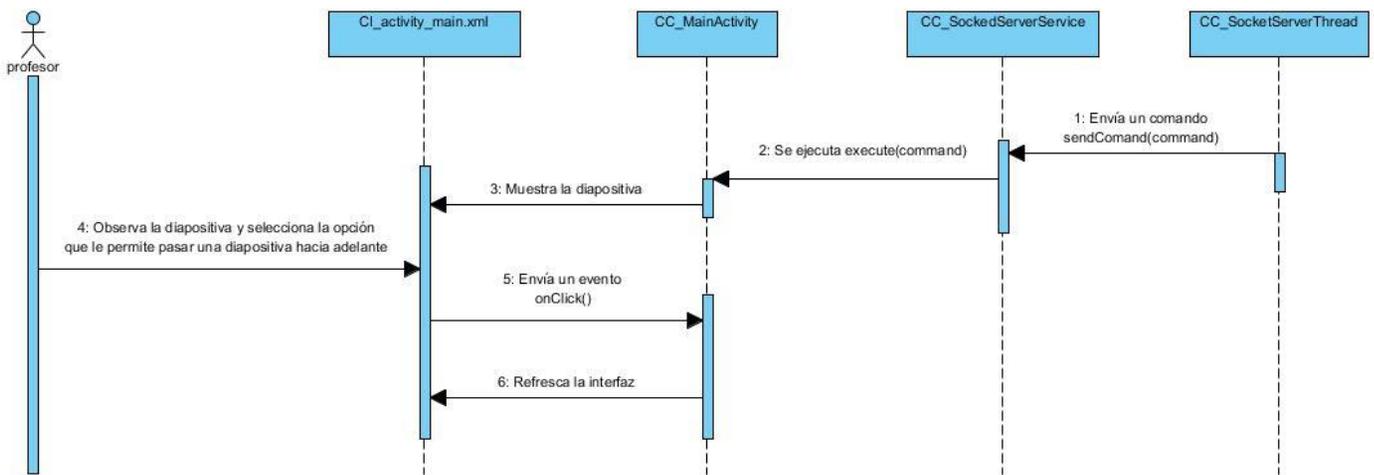
DSD Lanzar\_Aplicación.



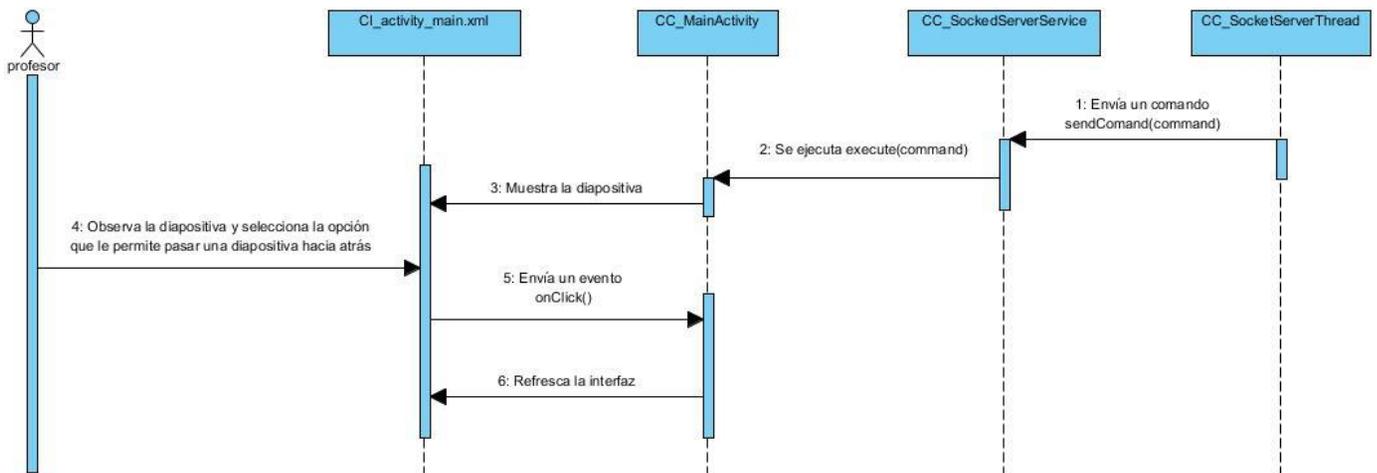
DSD Lanzar\_sitio\_web.



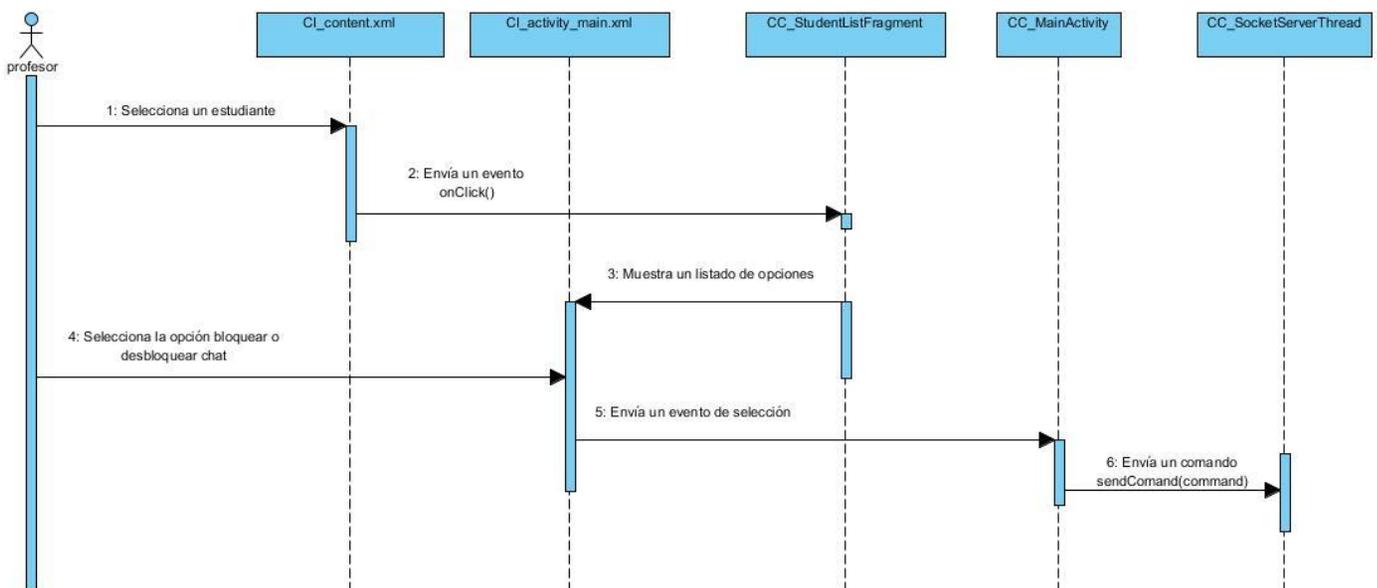
DSD Notificaciones.



DSD Pasar\_diapositiva\_hacia\_adelate.

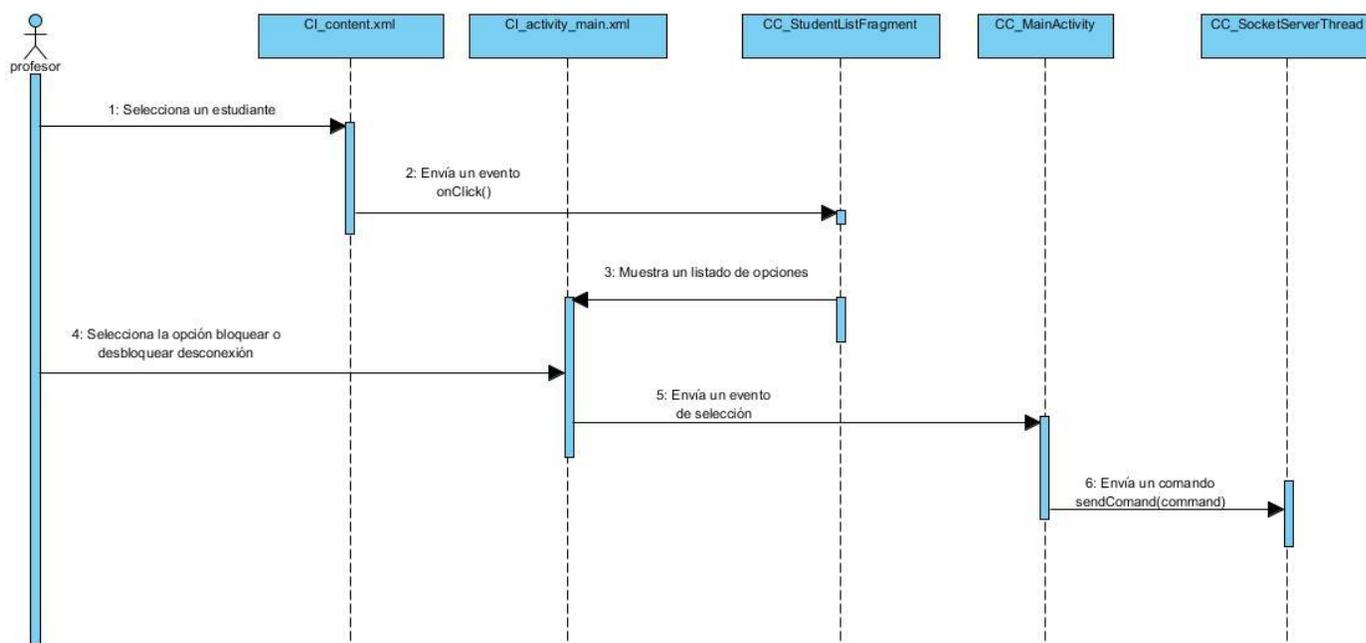


DSD Pasar\_diapositiva\_hacia\_atrás.

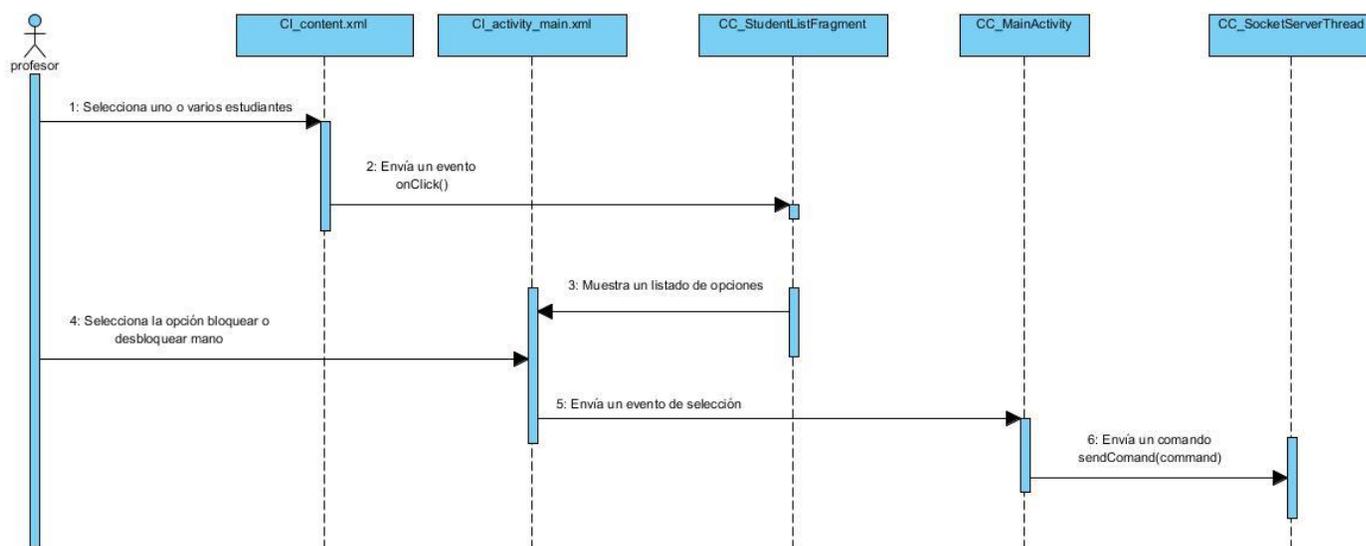


DSD Permitir\_chat.<sup>14</sup>

<sup>14</sup> DSD Permitir\_chat: HU\_Bloquear chat de los estudiantes, HU\_Desbloquear chat de los estudiantes.



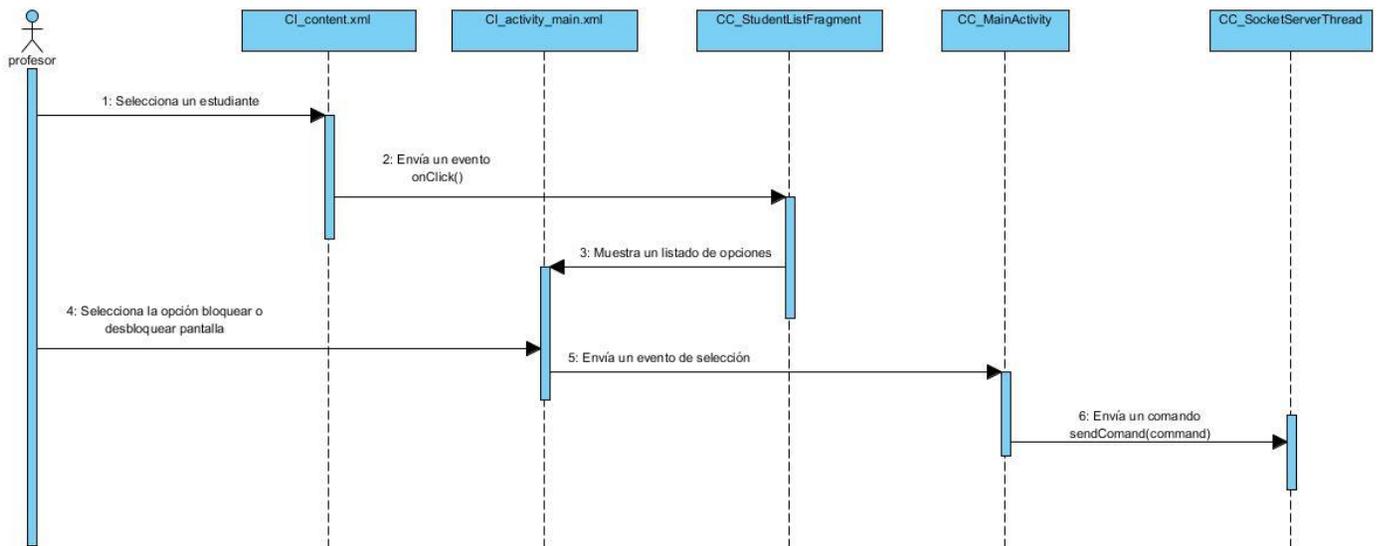
DSD Permitir\_desconexión.<sup>15</sup>



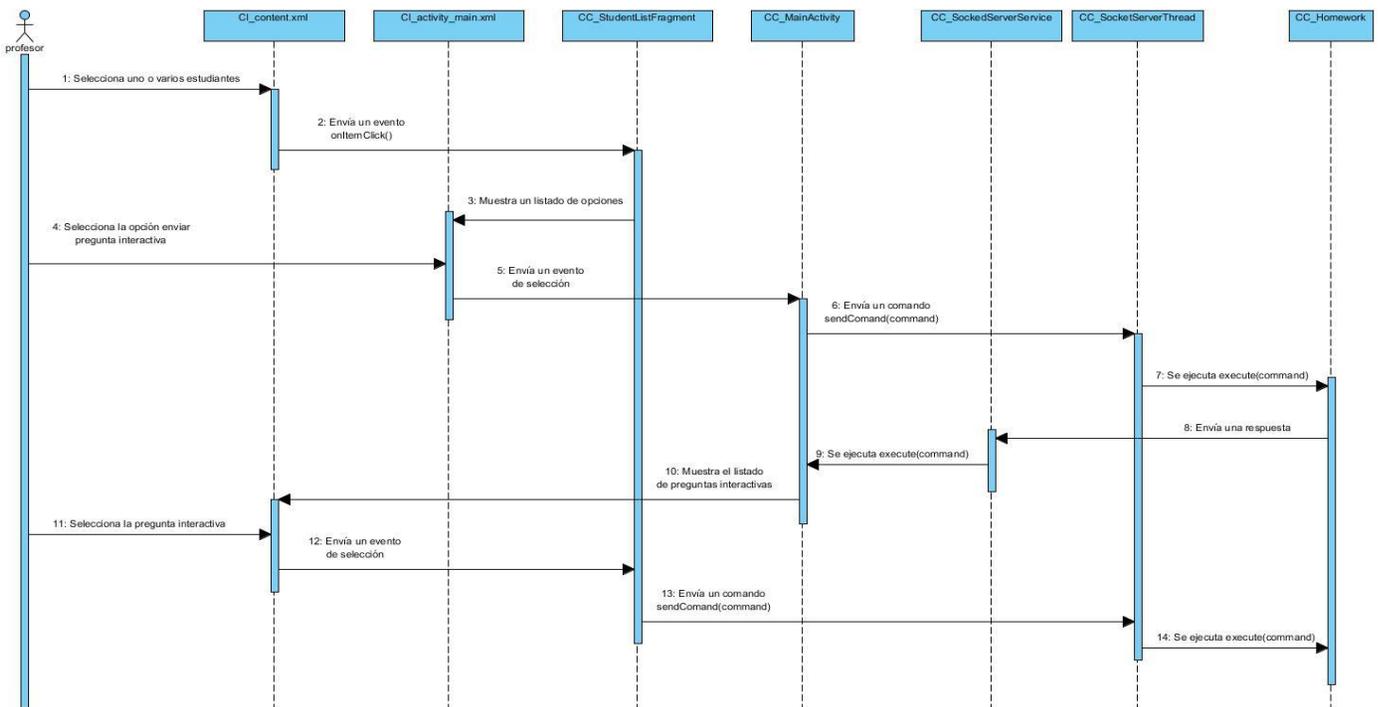
DSD Permitir\_mano.<sup>16</sup>

<sup>15</sup> DSD Permitir\_desconexión: HU\_Impedir desconexión de la clase.

<sup>16</sup> DSD Permitir\_mano: HU\_Bloquear manos levantadas, HU\_Desbloquear mano levantada.

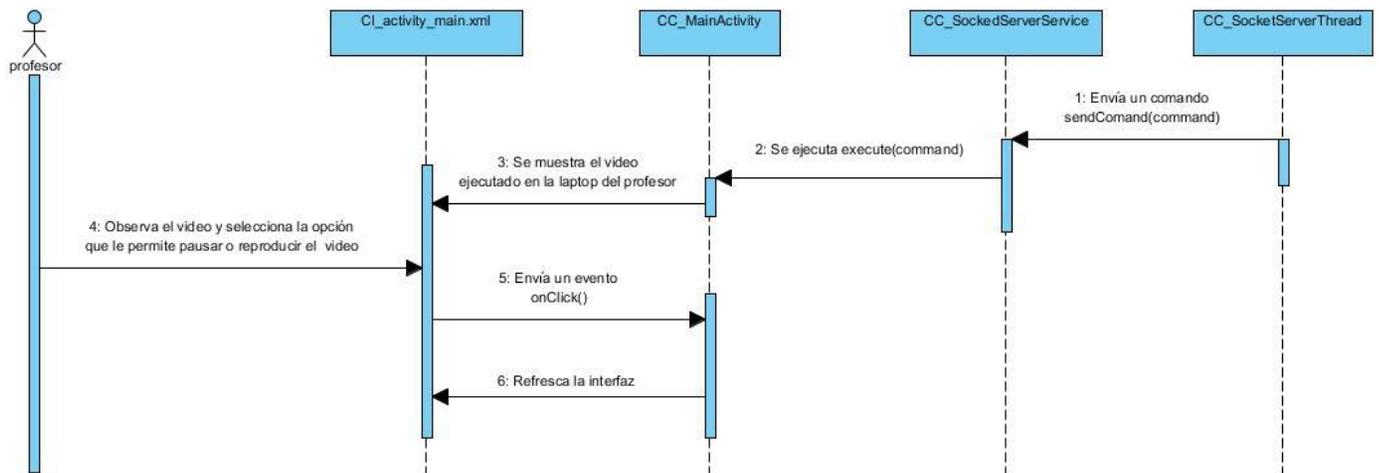


DSD Permitir\_pantalla.<sup>17</sup>

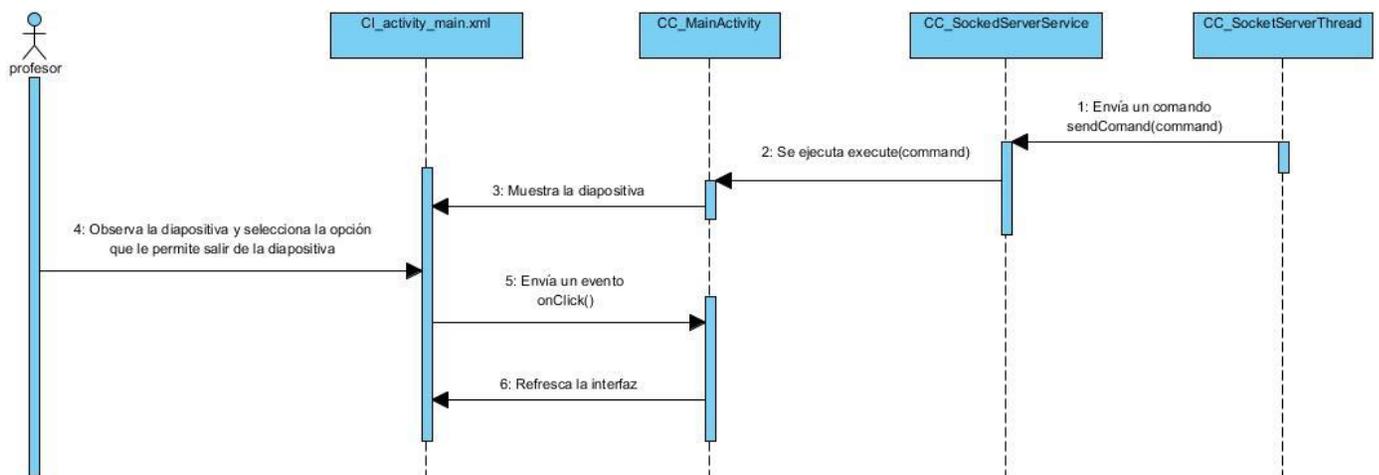


DSD Pregunta\_interactiva.

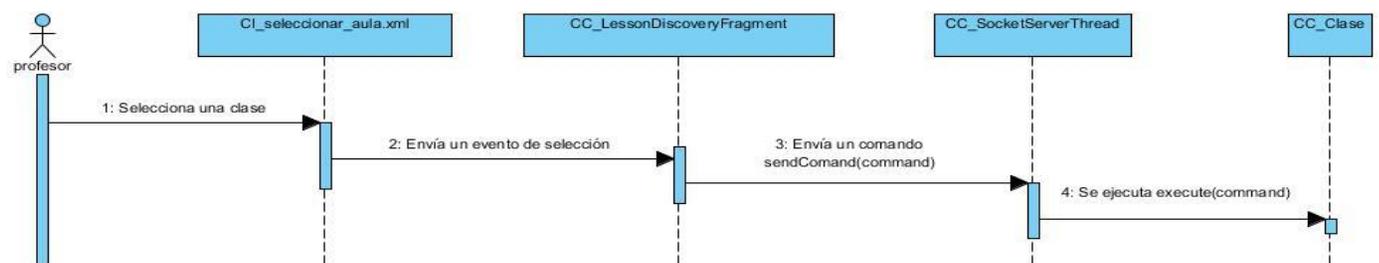
<sup>17</sup> DSD Permitir\_pantalla: HU\_Bloquear pantalla, HU\_Desbloquear pantalla.



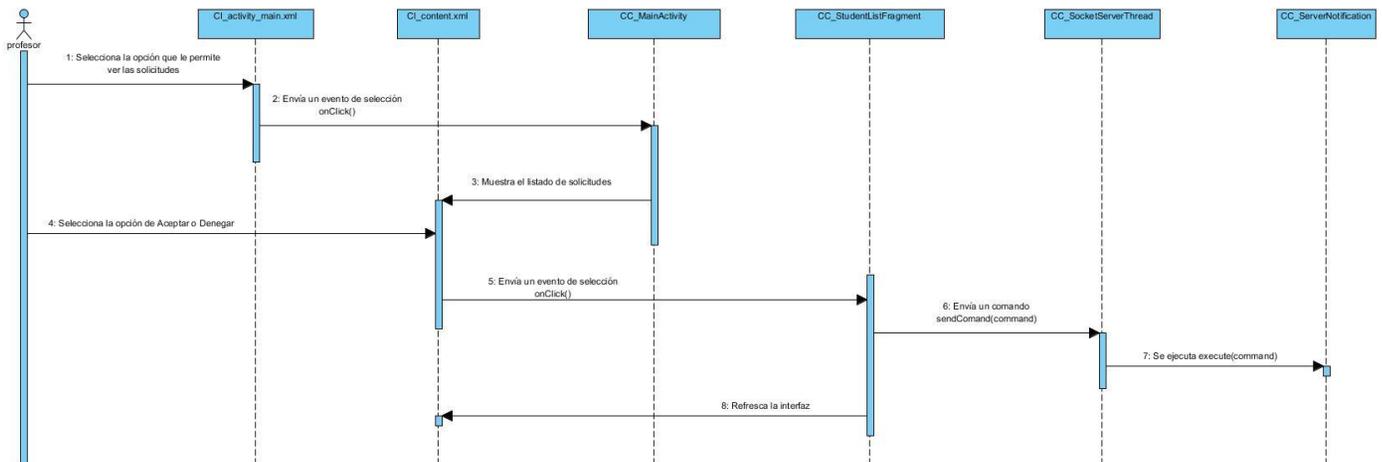
DSD Reproducir\_archivo\_de\_video.



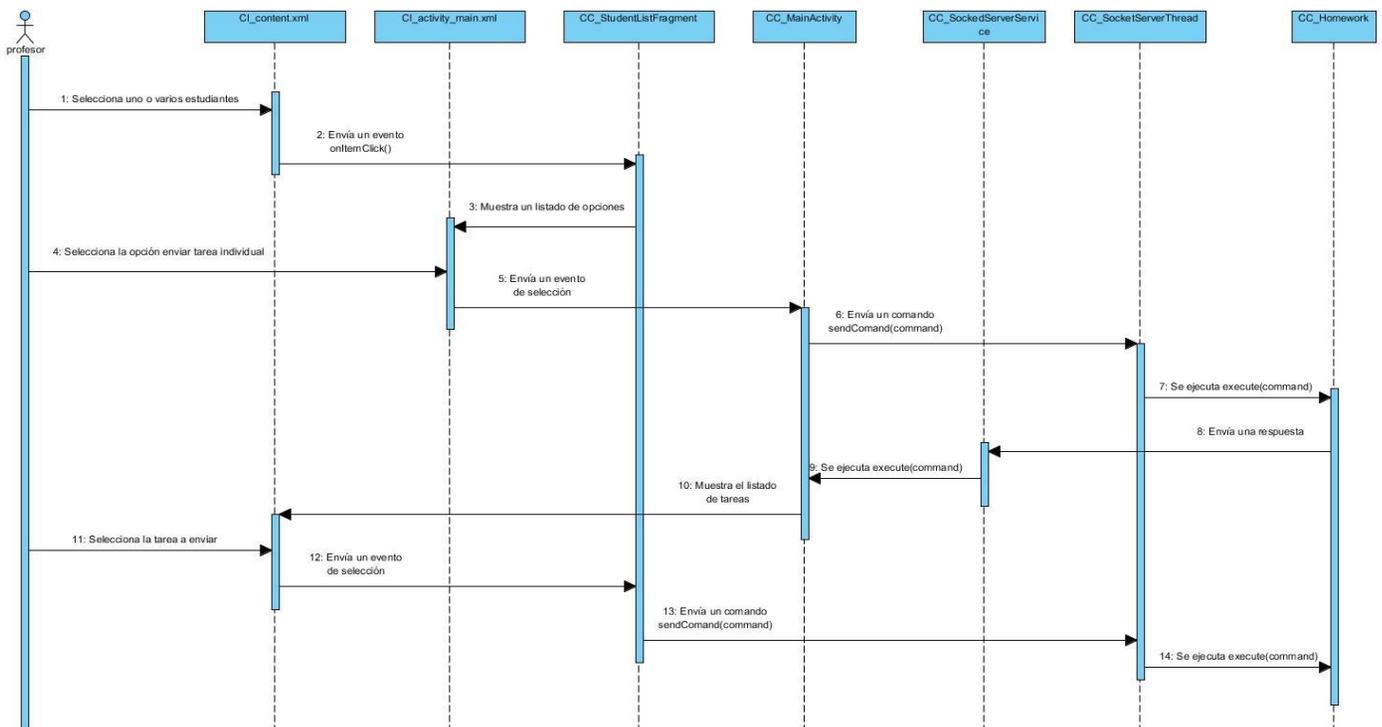
DSD Salir\_de\_la\_diapositiva.



DSD Seleccionar\_clase.

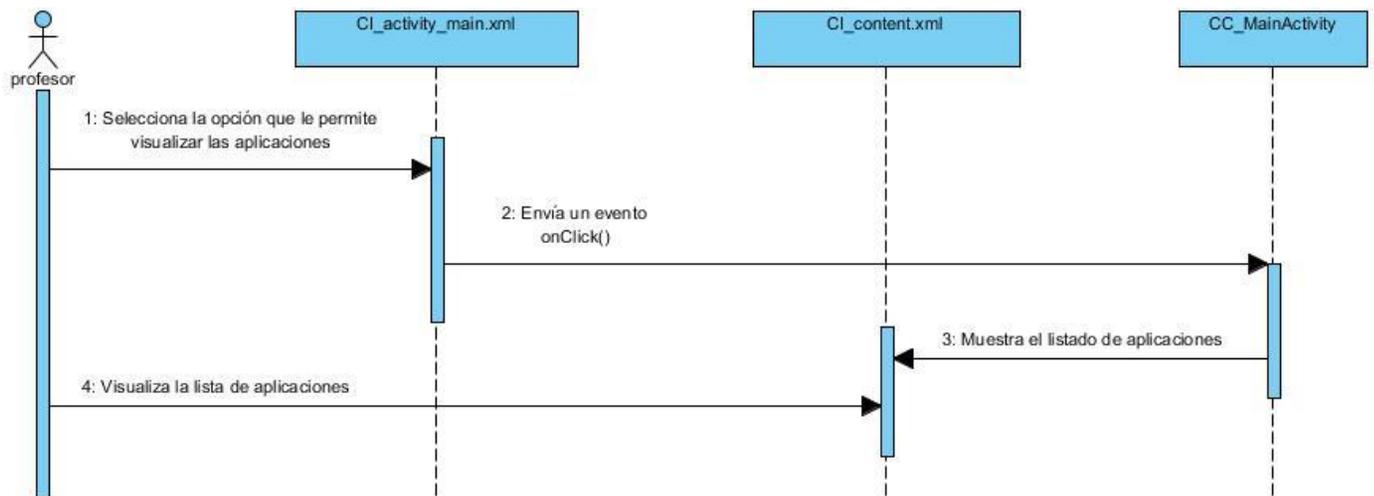


DSD Solicitudes.<sup>18</sup>

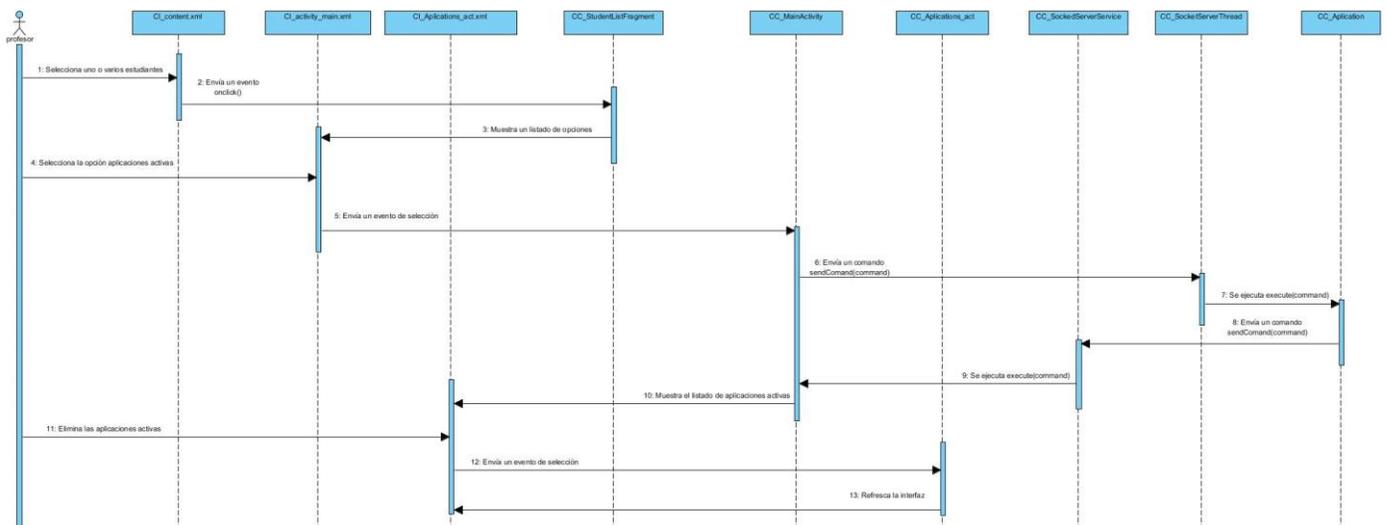


DSD Tarea.

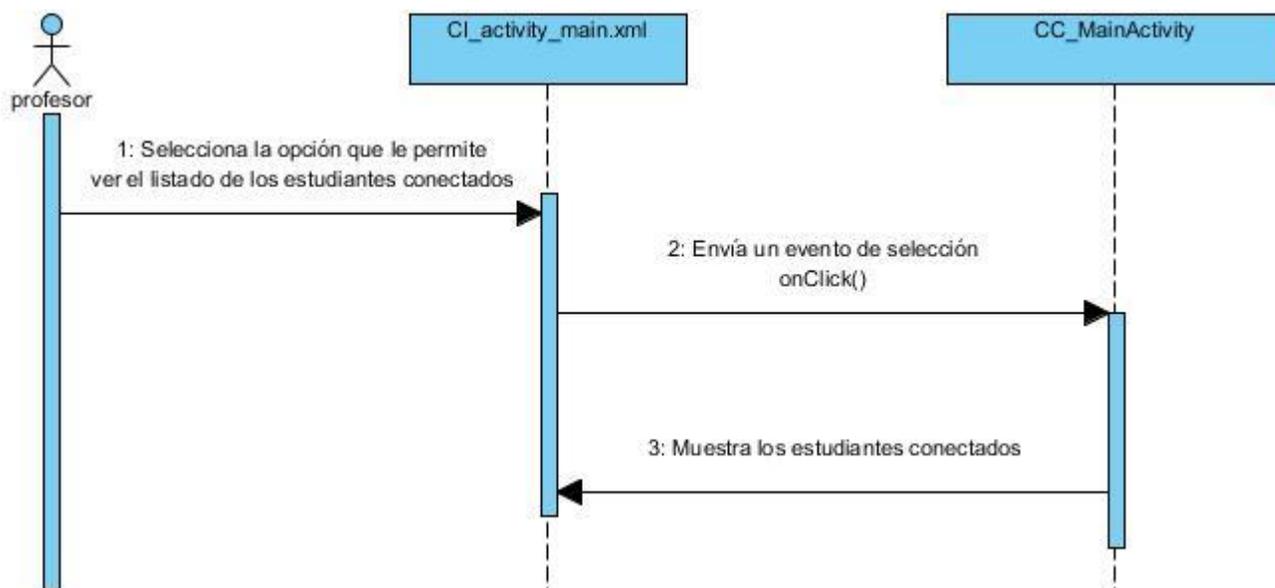
<sup>18</sup> **DSD Solicitudes:** HU\_Permitir la solicitud de estudiantes para conectarse a la clase, HU\_Permitir manos levantadas.



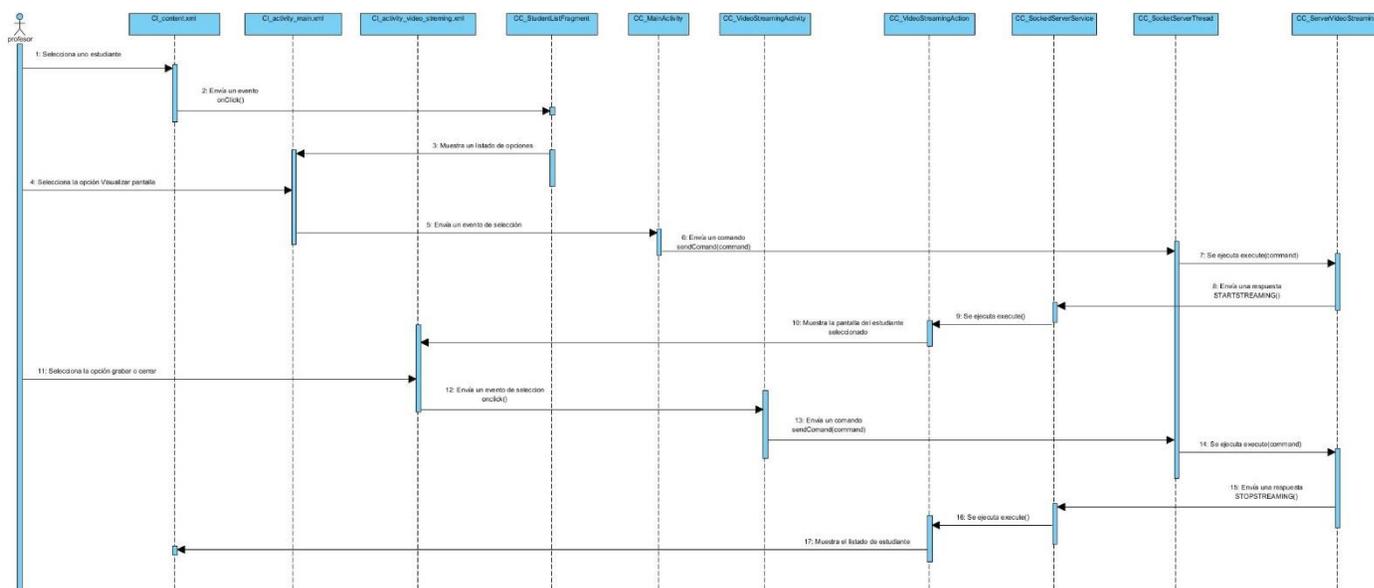
DSD Visualizar\_aplicación.



DSD Visualizar\_aplicaciones\_activas.

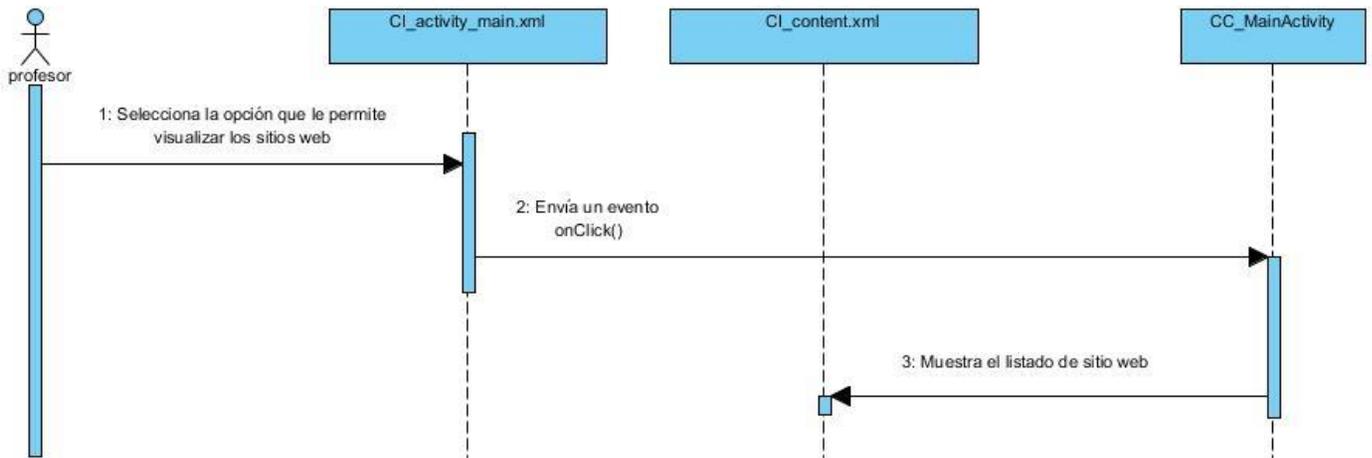


DSD Visualizar\_estudiantes.



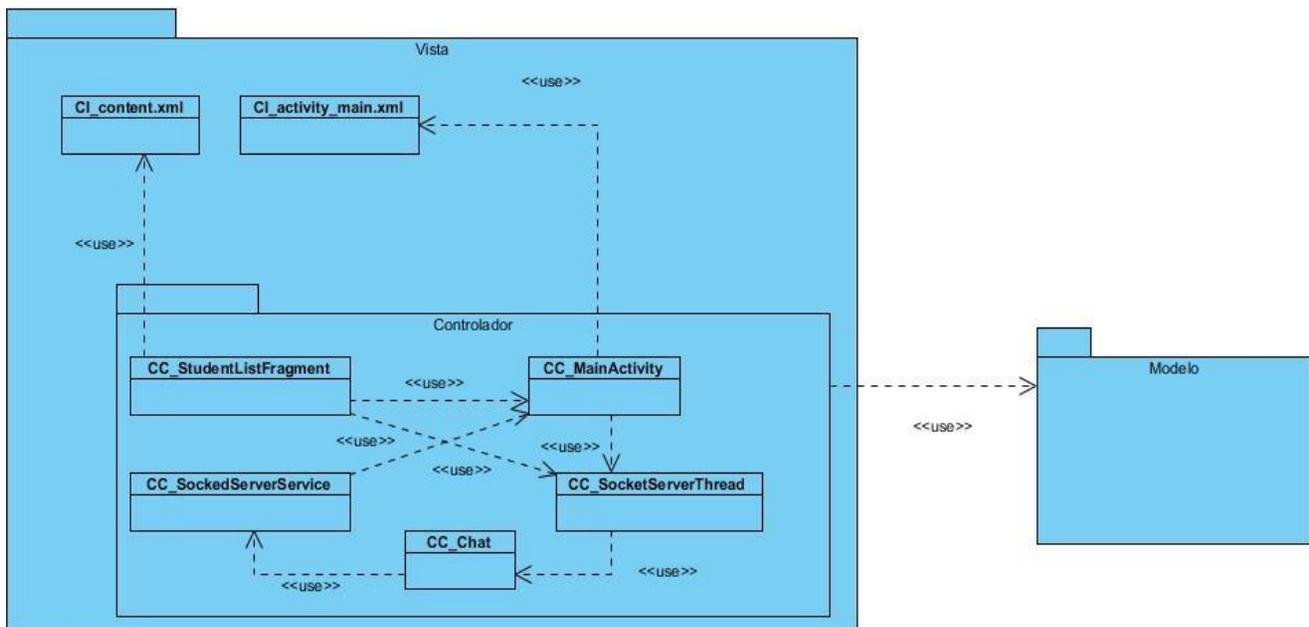
DSD Visualizar\_pantalla.<sup>19</sup>

<sup>19</sup> DSD Visualizar\_pantalla: HU\_Visualizar la pantalla de un estudiante en el terminal del profesor, HU\_Grabar pantalla del estudiante, HU\_Terminar transmisión de la pantalla del estudiante.



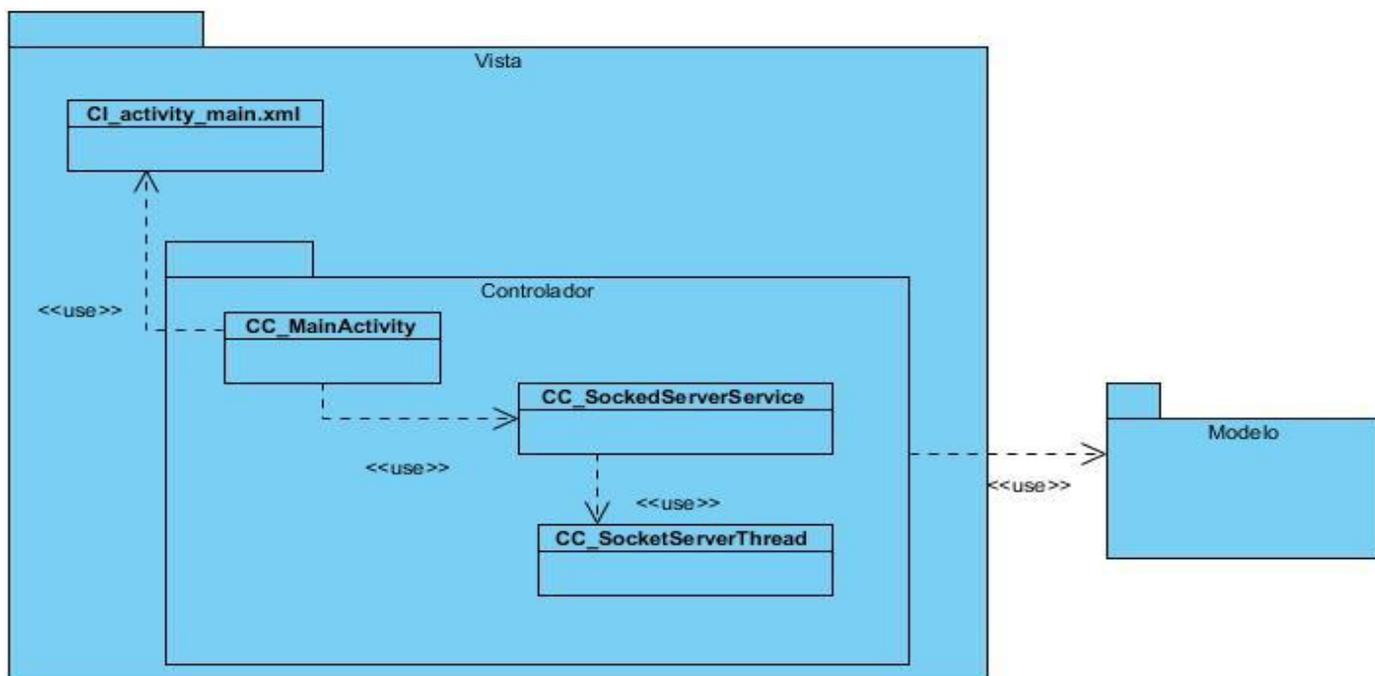
DSD Visualizar\_sitios\_web.

Anexo #4: Diagramas de Clase del diseño (DCO)

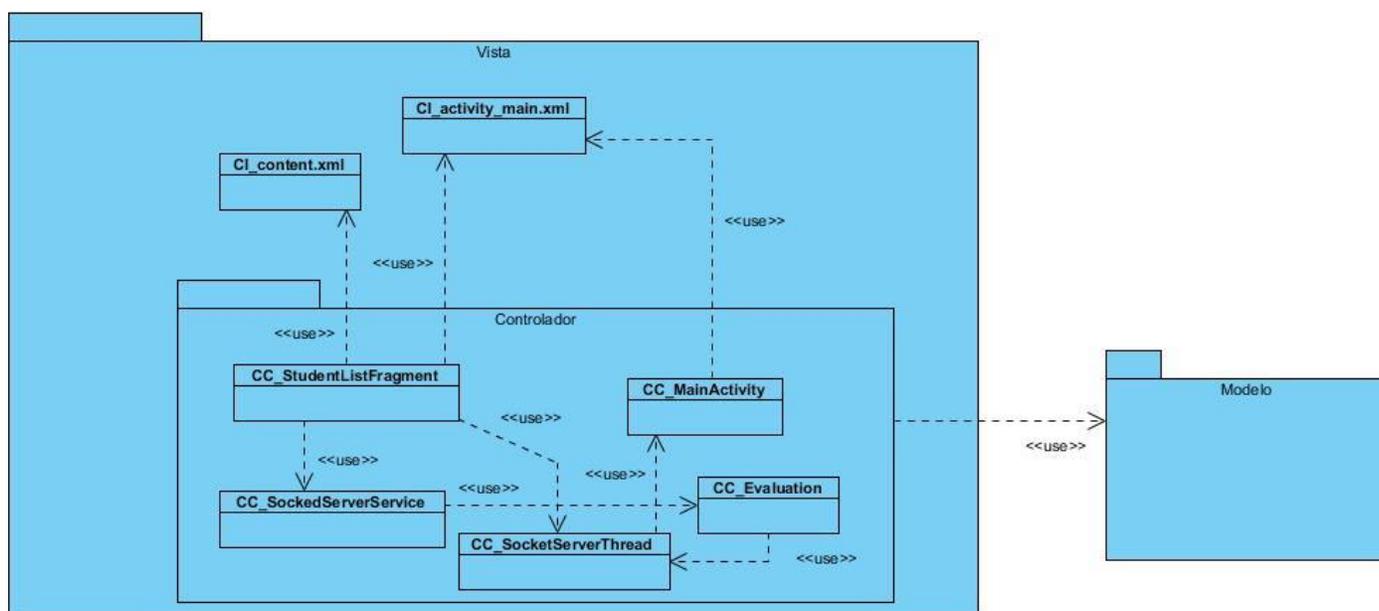


DCD Paquete\_Chat.

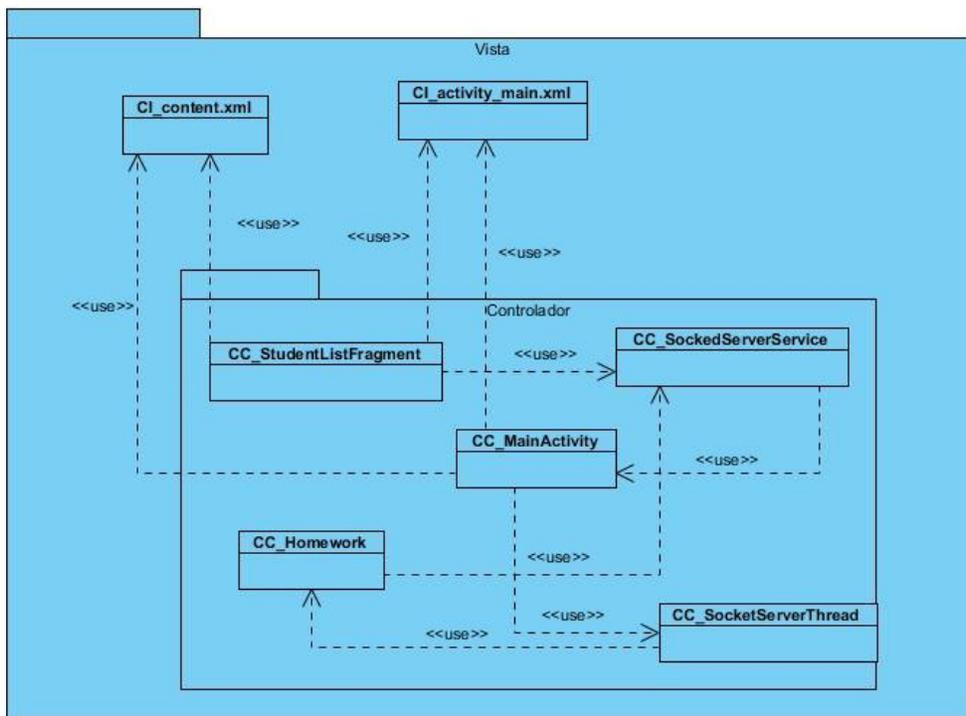




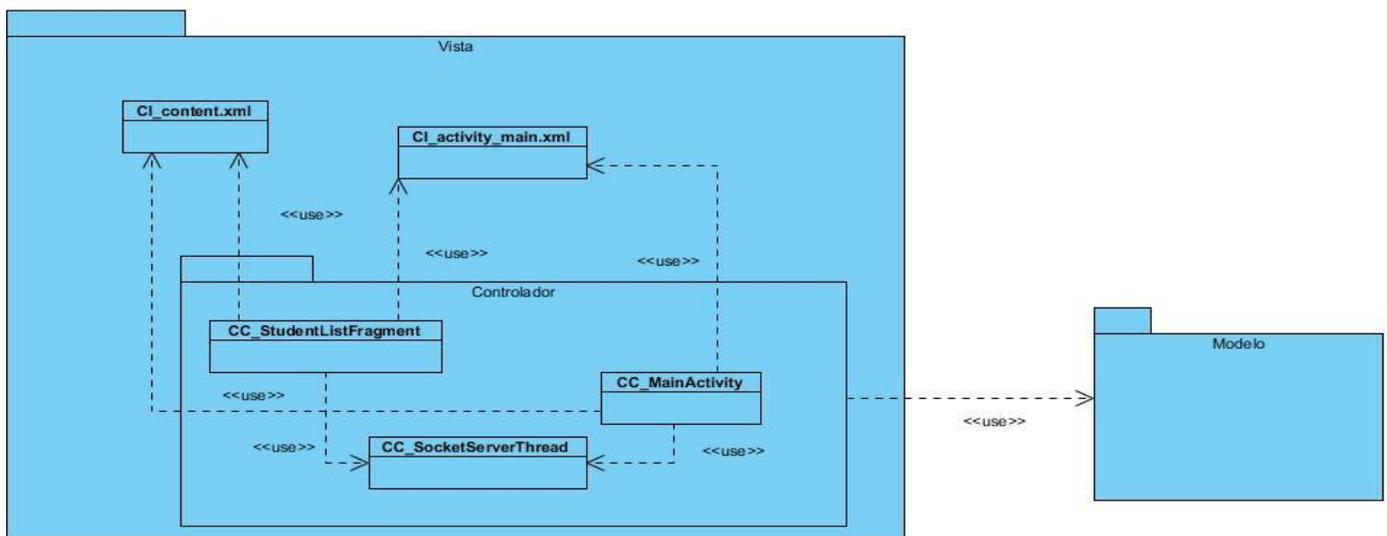
DCD Paquete\_Diapositiva.



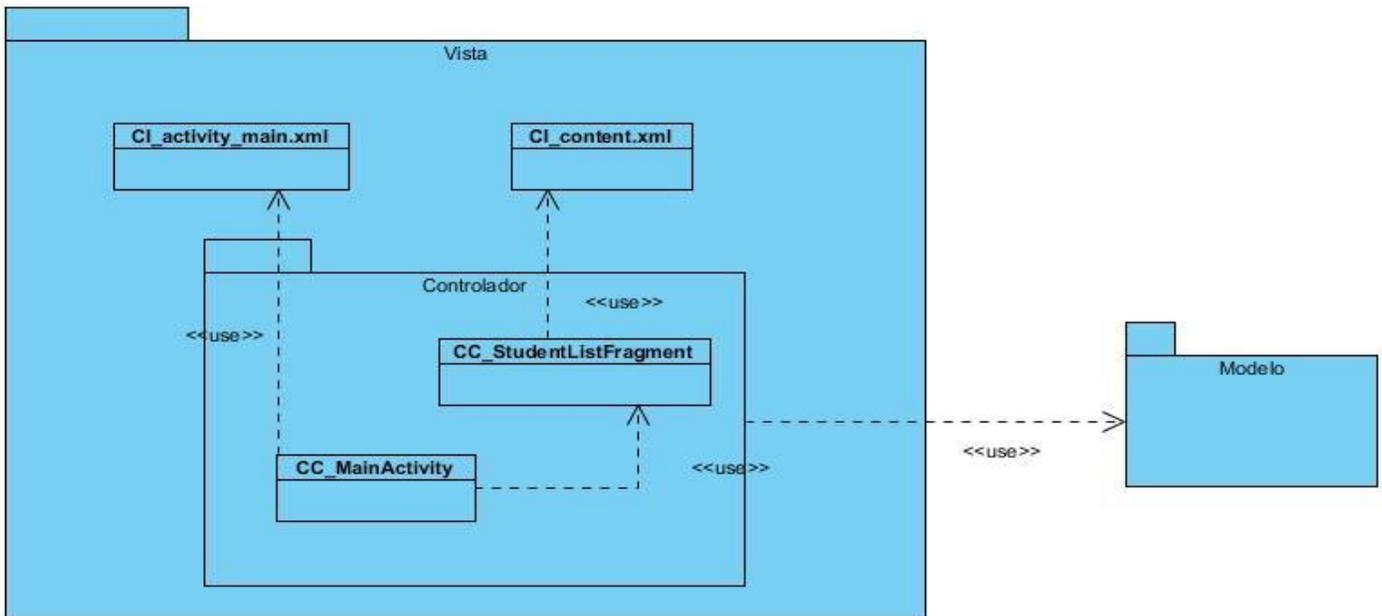
DCD Paquete\_Evaluación.



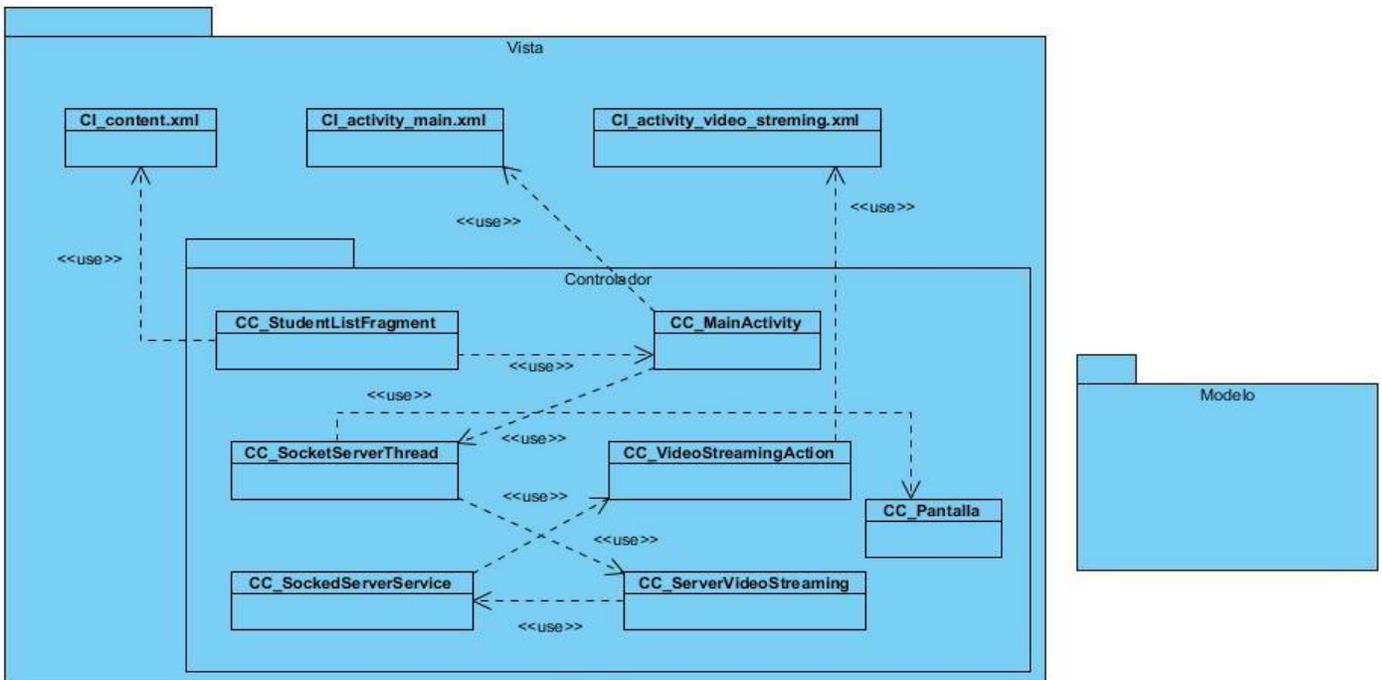
DCD Paquete\_Examen.



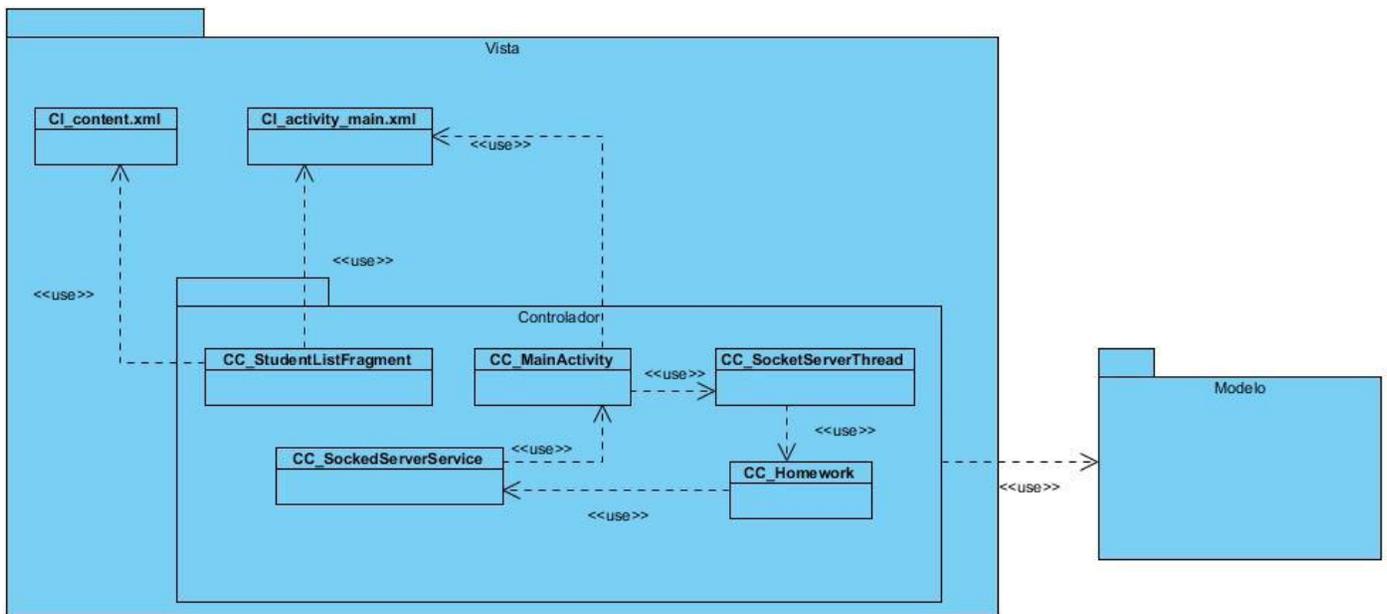
DCD Paquete\_Mano.



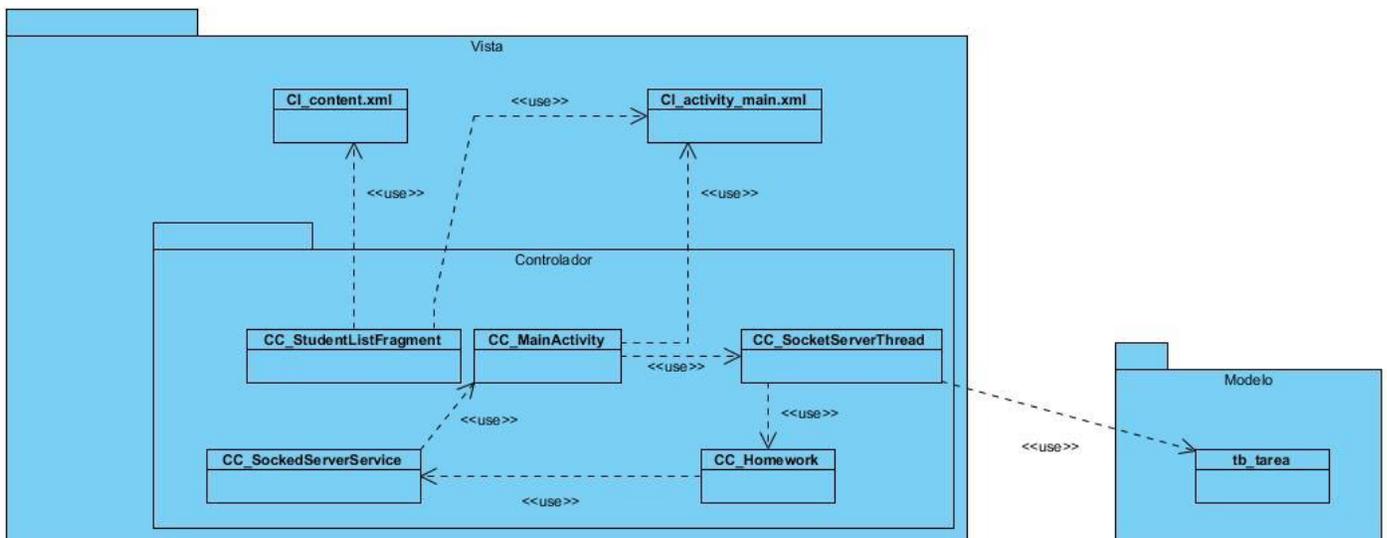
DCD Paquete\_Notificaciones.



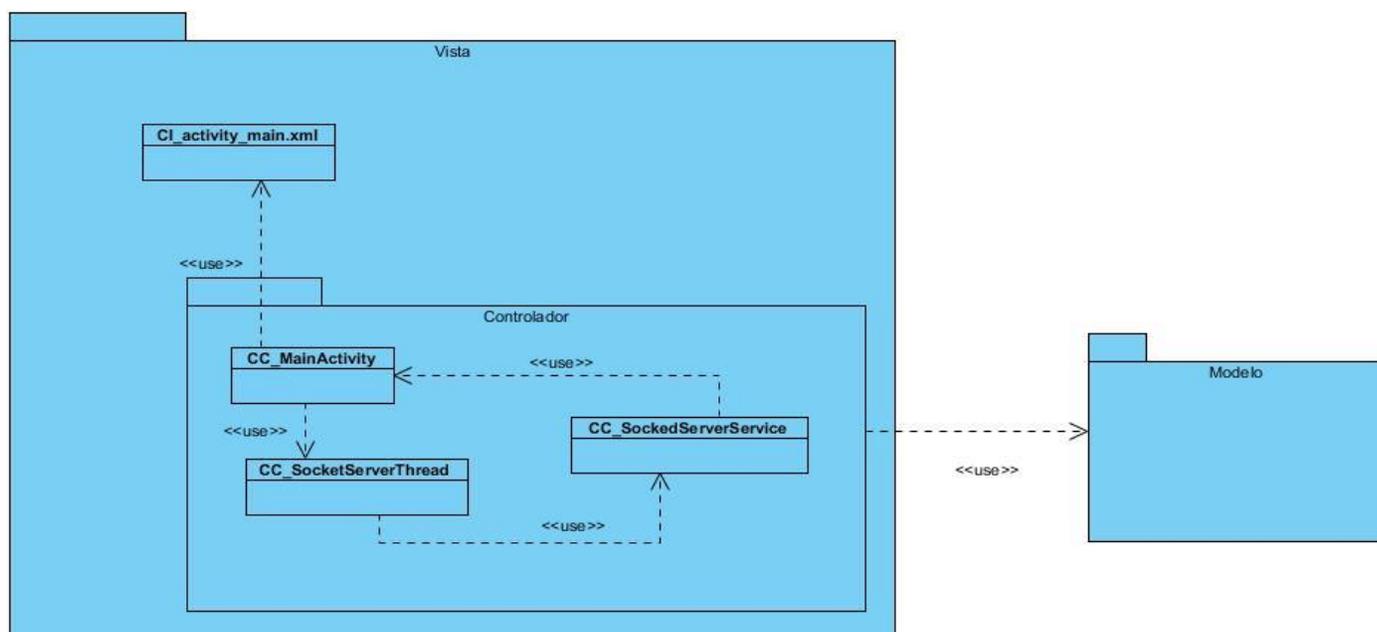
DCD Paquete\_Pantalla.



DCD Paquete\_Pregunta\_Interactiva.



DCD Paquete\_Tarea.



DCD Paquete\_Video.

Anexo #5: Casos de Pruebas (CP)

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción Cerrar sesión.	Selecciona la opción cerrar sesión	Sale de la sesión Muestra la interfaz que le permite seleccionar nuevamente una clase.	Interfaz principal/Cerrar Sesión

CP\_ Cerrar\_sesión.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción Desconectar estudiante.	Selecciona la opción desconectar estudiante	Se desconecta el estudiante de la clase.	Interfaz principal/desconectar estudiante.

CP\_ Desconectar\_estudiante.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción Buscar clase.	Selecciona la opción buscar clase	Escanea y muestra las clases activas.	Buscar clase

CP\_Buscar\_clase.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción seleccionar clase.	Selecciona la opción seleccionar clase	Muestra el formulario para autenticarse	seleccionar clase

**CP\_ Seleccionar\_clase.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción de ver el listado de solicitudes	Selecciona la opción que le permite visualizar las solicitudes	Muestra el listado de solicitudes Permite además: - Aceptar solicitud. - Denegar solicitud	Interfaz principal/Listar solicitudes
<b>EC 1.2</b> Opción de aceptar solicitud	El usuario selecciona la opción de Aceptar	Se refresca la interfaz	Interfaz principal/Listar solicitudes/aceptar
<b>EC 1.3</b> Opción de cancelar solicitud	El usuario selecciona la opción de Denegar	Se refresca la interfaz	Interfaz principal/Listar solicitudes/Denegar

**CP\_ Permitir\_Solicitudes<sup>20</sup>.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción Bloquear mano.	Selecciona la opción bloquear mano	Se bloquea la opción levantar mano del estudiante.	Interfaz principal/bloquear mano.

**CP\_ Bloquear\_mano.**

Escenario	Descripción	Respuesta del sistema	Flujo central
-----------	-------------	-----------------------	---------------

<sup>20</sup> **CP\_ Permitir\_Solicitudes:** HU\_Permitir la solicitud de estudiantes para conectarse a la clase, HU\_Permitir solicitud de envío de archivo del estudiante, HU\_Permitir manos levantadas.

<b>EC 1.1</b> Opción Bloquear pantalla.	Selecciona la opción bloquear pantalla.	Se bloquea la pantalla del estudiante.	Interfaz principal/bloquear pantalla.
---	---	--	---------------------------------------

**CP\_ Bloquear\_pantalla.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción Bloquear chat.	Selecciona la opción bloquear chat.	Se bloquea el chat del estudiante.	Interfaz principal/bloquear chat.

**CP\_ Bloquear\_chat.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción Bloquear desconexión.	Selecciona la opción bloquear desconexión	Se bloquea la opción desconexión del estudiante.	Interfaz principal/bloquear desconexión.

**CP\_ bloquear\_desconexión.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción desbloquear desconexión.	Selecciona la opción desbloquear desconexión	Se desbloquea la opción desconexión del estudiante.	Interfaz principal/desbloquear desconexión.

**CP\_ desbloquear\_desconexión.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción desbloquear mano.	Selecciona la opción desbloquear mano	Se desbloquea la opción mano levantada del estudiante.	Interfaz principal/desbloquear mano.

**CP\_ desbloquear\_mano.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción desbloquear pantalla.	Selecciona la opción desbloquear pantalla	Se desbloquea la opción pantalla del estudiante.	Interfaz principal/desbloquear pantalla.

**CP\_ desbloquear\_pantalla.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción desbloquear chat.	Selecciona la opción desbloquear chat	Se desbloquea la opción chat del estudiante.	Interfaz principal/desbloquear chat.

**CP\_ desbloquear\_chat.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción visualizar pantalla.	Selecciona la opción que le permite visualizar la pantalla de un estudiante.	Muestra la pantalla de un estudiante. Permite además: -Grabar pantalla. -Terminar grabación	Interfaz principal/Visualizar pantalla.
<b>EC 1.2</b> Opción grabar pantalla.	El usuario selecciona la opción de Grabar	Se graba la pantalla del estudiante	Interfaz principal/Visualizar pantalla/grabar.

<b>EC 1.3</b> Opción terminar grabación	El usuario selecciona la opción de Terminar	Se termina la grabación de la pantalla del estudiante	Interfaz principal/Visualizar pantalla/terminar
---	---	---	---

**CP\_ pantalla<sup>21</sup>.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción visualizar aplicaciones.	Selecciona la opción que le permite visualizar las aplicaciones.	Muestra las aplicaciones. Permite además: -Agregar aplicación.	Interfaz principal/Visualizar aplicaciones.
<b>EC 1.2</b> Opción agregar aplicación	El usuario selecciona la opción que le permite Agregar	Se agrega una aplicación al listado de aplicaciones	Interfaz principal/Visualizar aplicaciones/agregar.

**CP\_ aplicaciones.<sup>22</sup>**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción visualizar sitio web	Selecciona la opción que le permite visualizar los sitios web.	Muestra los sitios web. Permite además: -Eliminar aplicación. -Agregar aplicación.	Interfaz principal/Visualizar sitio web.
<b>EC 1.2</b> Opción eliminar sitio web	El usuario selecciona la opción Eliminar.	Se elimina el sitio web del listado de sitios web.	Interfaz principal/Visualizar sitio web/eliminar.
<b>EC 1.3</b> Opción agregar sitio web	El usuario selecciona la opción que le permite Agregar	Se agrega un sitio web al listado de sitios web.	Interfaz principal/Visualizar sitio web/agregar.

<sup>21</sup> **CP\_ pantalla:** HU\_Visualizar la pantalla de un estudiante en el terminal del profesor, HU\_Grabar pantalla del estudiante, HU\_Terminar transmisión de la pantalla del estudiante.

<sup>22</sup> **CP\_ aplicaciones:** HU\_Ver listado de aplicaciones remotas, HU\_Agregar aplicación.

CP\_sitio\_web.<sup>23</sup>

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción lanzar sitio web	Selecciona la opción que le permite lanzar un sitio web.	Muestra los sitios web. Permite además: -Lanzar un sitio web.	Interfaz principal/lanzar sitio web.
<b>EC 1.2</b> Opción lanzar sitio web	El usuario selecciona la opción Lanzar.	Se lanza el sitio web en las tabletas de los estudiantes	Interfaz principal/lanzar sitio web/lanzar.

CP\_Lanzar\_sitio\_web.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción lanzar aplicación	Selecciona la opción que le permite lanzar una aplicación.	Muestra las aplicaciones. Permite además: -Lanzar una aplicación.	Interfaz principal/lanzar aplicación.
<b>EC 1.2</b> Opción lanzar aplicación	El usuario selecciona la opción Lanzar.	Se lanza la aplicación en las tabletas de los estudiantes	Interfaz principal/lanzar aplicación/lanzar.

CP\_aplicación.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción aplicaciones activas	Selecciona la opción que le permite ver las aplicaciones activas.	Muestra las aplicaciones activas. Permite además: -Cerrar aplicación.	Interfaz principal/aplicaciones activas.

<sup>23</sup> CP\_sitio\_web: HU\_Ver listado de sitios web, HU\_Agregar sitio web, HU\_Eliminar sitio web.

<b>EC 1.2</b> Opción cerrar aplicación	El usuario selecciona la opción cerrar.	Se lanza la aplicación en las tabletas de los estudiantes	Interfaz principal/aplicaciones activas/cerrar.
--	---	---	---

**CP\_aplicaciones\_activas.**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción pasar diapositiva hacia adelante.	Selecciona la opción que le permite pasar diapositiva hacia adelante.	Se pasa la diapositiva hacia adelante	Diapositiva/pasar hacia adelante.
<b>EC 1.2</b> Opción pasar diapositiva hacia atrás.	Selecciona la opción que le permite pasar diapositiva hacia atrás.	Se pasa la diapositiva hacia atrás	Diapositiva/pasar hacia atrás.
<b>EC 1.3</b> Opción salir	El usuario selecciona la opción de salir	Se cierra la diapositiva	Diapositiva/salir.

**CP\_diapositiva.<sup>24</sup>**

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción enviar examen.	Selecciona la opción que le permite ver los exámenes.	Muestra el listado de exámenes. Permite además: -Enviar.	Interfaz principal/enviar examen.

<sup>24</sup> **CP\_diapositiva:** HU\_Pasar diapositiva hacia adelante, HU\_Pasar diapositiva hacia atrás, HU\_Salir de la diapositiva.

<b>EC 1.2</b> Opción enviar.	Se envía el examen.	Se muestra el examen en las tabletas de los estudiantes.	Interfaz principal/enviar examen/enviar.

CP\_examen.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción enviar tarea.	Selecciona la opción que le permite ver las tareas.	Muestra el listado de tareas. Permite además: -Enviar.	Interfaz principal/enviar tarea.
<b>EC 1.2</b> Opción enviar.	Se envía el examen.	Se muestra la tarea en las tabletas de los estudiantes	Interfaz principal/enviar tarea/enviar.

CP\_tarea.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción enviar pregunta interactiva	Selecciona la opción que le permite ver las preguntas interactivas	Muestra el listado de preguntas interactivas. Permite además: -Enviar.	Interfaz principal/enviar pregunta interactiva.
<b>EC 1.2</b> Opción enviar.	Se envía el examen.	Se muestra el examen en las tabletas de los estudiantes	Interfaz principal/enviar pregunta interactiva/enviar.

CP\_pregunta\_interactiva.

Escenario	Descripción	Respuesta del sistema	Flujo central
<b>EC 1.1</b> Opción reproducir video.	Selecciona la opción que le permite reproducir video	Se reproduce el video.	Video/reproducir video.

<b>EC 1.2</b> Opción pausar video.	Selecciona la opción que le permite pausar video.	Se pausa el video.	Video/reproducir video/pausar video.
<b>EC 1.3</b> Opción detener video.	Selecciona la opción detener el video.	Se detiene y se cierra el video.	Video/reproducir video/detener video.

**CP\_video.**<sup>25</sup>

---

<sup>25</sup> **CP\_video:** HU\_Reproducir archivo de video, HU\_Pausar video, HU\_Detener video.