



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 4

*Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas*

**Aplicación del estudiante en el Sistema Operativo Android para
el Sistema de Gestión de Clase ATcnea.**

Autor:

Adrian Arencibia Herrera

Tutores:

MSc. Yerandy Manso Guerra

Ing. Yenima Hernández Orozco

Ing. Odenys Almora Rodríguez

La Habana, junio de 2017

“Año 59 de la Revolución”

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 _____.

Adrian Arencibia Herrera

Autor

Msc. Yerandy Manso Guerra

Tutor

Ing. Yenima Hernández Orozco

Tutor

Ing. Odenys Almora Rodríguez

Tutor

DEDICATORIA

A mi Madre, por estar conmigo siempre, por enseñarme a crecer y a que si caigo debo levantarme, por apoyarme y guiarme, por ser la base que me ayudó a llegar hasta aquí.

AGRADECIMIENTOS

El primer agradecimiento de la presente investigación es para todos los profesores que de una forma u otra han intervenido en mi formación y me han sabido guiar por el camino del conocimiento.

A mis tutores por toda su labor durante el proceso de investigación.

A Yaritza Bárbara González Ramírez por toda su ayuda en esta tesis.

A mi novia Rosmery por estar a mi lado apoyándome en todos estos años.

A mis padres, a mis hermanos y a toda mi familia que aunque lejos siempre los llevo conmigo.

A todos mis amigos y en general a todo el que me han ayudado en este largo camino.

RESUMEN

El desarrollo y utilización de las Tecnologías de la Información y las Comunicaciones (TIC), en el ámbito educacional, han propiciado el surgimiento de las aulas tecnológicas. Uno de los componentes esenciales de estas aulas es el software que las gestiona. El producto ATcnea es un sistema de gestión de clase desarrollado por el Centro de Tecnologías para la Formación (FORTES) de la Facultad 4 de la Universidad de las Ciencias Informáticas (UCI), con hardware HAIER ensamblado por la Empresa Industrial para la Informática, la Electrónica y las Comunicaciones (GEDEME). ATcnea le permite al profesor aprovechar las tecnologías en función del proceso de enseñanza-aprendizaje, este aprovechamiento está limitado por la escasa interacción de los estudiantes. Por lo que se hace necesario desarrollar una aplicación que garantice la interacción de los estudiantes en un aula tecnológica con el software ATcnea. El desarrollo de la solución está guiado por la metodología AUP-UCI en el escenario 4. Para su elaboración se usan herramientas y tecnologías como: Visual Paradigm, Android Studio, Java y Android SDK. Al culminar la implementación de la aplicación se le aplicaron pruebas funcionales y de aceptación, evidenciándose que existe una correspondencia entre el objetivo y los resultados obtenidos, así como una alta satisfacción del cliente.

Palabras claves: ATcnea, aulas tecnológicas, sistema de gestión de clase, tecnologías, proceso de enseñanza-aprendizaje

SUMMARY

The development and use of Information and Communication Technologies (ICT), in the educational field, have led to the emergence of technological classrooms. One of the essential components of these classrooms is the software that manages them. The ATcnea product is a class management system developed by the Center for Technologies for Training (FORTES) of Faculty of the University of Computer Science (UCI), with HAIER hardware assembled by the Industrial Company for Informatics, Electronics and Communications (GEDEME). ATcnea allows the teacher to take advantage of technologies based on the teaching-learning process, this use is limited by the low interaction of students. Therefore it is necessary to develop an application that guarantees the interaction of the students in a technological classroom with the ATcnea software. The development of the solution is guided by the AUP-UCI methodology in scenario 4. The tools and technologies used are: Visual Paradigm, Android Studio, Java and Android SDK. At the end of the implementation of the application are applied functional tests and acceptance, evidencing that there is a correspondence between the objective and the obtained results, as well as to high customer satisfaction.

Keywords: ATcnea, technological classrooms, class management system, technologies, teaching-learning process

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Tecnologías para el apoyo del proceso enseñanza-aprendizaje.....	5
1.2.1. Aulas Tecnológicas	6
1.2.2. Objetivo de las Aulas tecnológicas	7
1.2.3. Dispositivos móviles en el proceso de enseñanza-aprendizaje.....	7
1.2.4. Sistemas gestores de clase.....	8
1.3. Metodología, tecnologías, lenguajes y herramientas	12
1.3.1. Metodologías de desarrollo de software.....	12
1.3.2. Tecnologías a emplear.....	15
1.3.3. Bibliotecas/Librerías de terceros	17
1.4. Conclusiones parciales	18
CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	20
2.1. Introducción	20
2.2. Modelo de dominio	20
2.3. Propuesta de solución.....	21
2.4. Requisitos de software.....	22
2.4.1. Requisitos funcionales	22
2.4.2. Requisitos no funcionales	23
2.5. Descripción de los usuarios	24
2.6. Historia de usuarios	24
2.7. Modelo de Diseño.....	26
2.8. Diagramas de clases del diseño	26
2.9. Diagrama de secuencia del diseño.....	27
2.10. Diagrama de despliegue	28
2.11. Modelado de datos (BD).....	28
2.12. Descripción de las tablas de la Base de Datos	29
2.13. Patrón arquitectónico	30
2.14. Patrones de diseño	31
2.14.1. Patrones GRASP	32
2.14.2. Patrones Gof	32
2.15. Conclusiones parciales.....	33
CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA	34

ÍNDICE

3.1.	Introducción	34
3.2.	Modelo de implementación	34
3.3.	Diagrama de componentes	34
3.4.	Pruebas de software	35
3.5.	Niveles de prueba.....	36
3.6.	Tipos de prueba.....	36
3.7.	Métodos de prueba.....	36
3.8.	Diseño de casos de prueba	37
3.9.	Resultados obtenidos.....	39
3.9.1.	Resultados de la prueba de Sistema.....	39
3.9.2.	Resultados de la prueba de aceptación	41
3.10.	Conclusiones del capítulo	41
CONCLUSIONES GENERALES		42
RECOMENDACIONES		43
REFERENCIAS BIBLIOGRÁFICAS		44
ANEXOS.....		49
Anexo 1: Descripción de historias de usuario		49
Anexo 2: Diagrama de clases del diseño		78
Anexo 3: Diagramas de secuencia del diseño		83
Anexo 4: Descripción de las tablas de la Base de Datos		89
Anexo 5: Diseño de casos de prueba		91
Anexo 6: Acta de aceptación		105

ÍNDICE DE TABLAS

Tabla 1 Funcionalidades de la app Mythware	9
Tabla 2 Funcionalidades de la app AB Tutor	11
Tabla 3 Descripción de usuarios	24
Tabla 4 Descripción de la HU Iniciar sesión.....	24
Tabla 5 Descripción de los atributos la tabla tb_user	29
Tabla 6 Diseño de caso de prueba de la HU SC Iniciar Sesión	37
Tabla 7.Resultados obtenidos de las pruebas por iteración.	40
Tabla 8 Crear una cuenta	49
Tabla 9 Editar perfil	51
Tabla 10 Cerrar sesión.....	54
Tabla 11 Establecer conversación con el profesor	55
Tabla 12 Permitir conversaciones de los estudiantes	56
Tabla 13 Acceder a la clase por autenticación	58
Tabla 14 Buscar clase.....	59
Tabla 15 Cerrar sesión de estudiante.....	61
Tabla 16 Desconectar estudiantes de la clase.....	62
Tabla 17 Seleccionar clase.....	63
Tabla 18 Abrir sitio web remoto	65
Tabla 19 Configurar lenguaje del dispositivo remoto	66
Tabla 20 Iniciar aplicación remota por el profesor	68
Tabla 21 Notificar estudiantes de la evaluación	69
Tabla 22 Aceptar mano levantada	70
Tabla 23 Denegar mano levantada	71
Tabla 24 Permitir manos levantadas	72
Tabla 25 Transmitir pantalla del profesor en modo completa	73
Tabla 26 Ver pantalla de los estudiantes	74
Tabla 27 Parar video	75
Tabla 28 Reproducir archivo de video	76
Tabla 29 Descripción de los atributos la tabla tb_Lesson.....	89
Tabla 30 Descripción de los atributos la tabla.....	89
Tabla 31 Descripción de los atributos la tabla tb_Message.....	89
Tabla 32 Descripción de los atributos la tabla tb_UserLesson.....	90
Tabla 33 Diseño de caso de prueba Crear una cuenta	91
Tabla 34 Diseño de caso de prueba Editar perfil	92
Tabla 35 Diseño de caso de prueba Terminar sesión	94
Tabla 36 Diseño de caso de prueba Establecer conversación con el profesor	94
Tabla 37 Diseño de caso de prueba Permitir conversaciones de los estudiantes	95
Tabla 38 Diseño de caso de prueba Acceder a la clase por autenticación	96
Tabla 39 Diseño de caso de prueba Buscar clase.....	97

ÍNDICE DE TABLAS

Tabla 40 Diseño de caso de prueba Cerrar sesión de estudiante.....	97
Tabla 41 Diseño de caso de prueba Desconectar estudiante de la clase.....	98
Tabla 42 Diseño de caso de prueba Seleccionar clase.....	98
Tabla 43 Diseño de caso de prueba Abrir sitio web remoto	99
Tabla 44 Diseño de caso de prueba Configurar lenguaje del dispositivo remoto	99
Tabla 45 Diseño de caso de prueba Iniciar aplicación remota por el profesor.....	100
Tabla 46 Diseño de caso de prueba Notificar estudiantes de la evaluación	100
Tabla 47 Diseño de caso de prueba Aceptar mano levantada	101
Tabla 48 Diseño de caso de prueba Denegar mano levantada.....	101
Tabla 49 Diseño de caso prueba Permitir manos levantadas	101
Tabla 50 Diseño de caso de prueba Transmitir pantalla modo completo	102
Tabla 51 Diseño de caso de prueba Visualizar pantalla en miniatura del estudiante	102
Tabla 52 Diseño de caso de prueba Parar video	103
Tabla 53 Diseño de caso de prueba Reproducir archivo de video	104

ÍNDICE DE FIGURAS

Figura 1 Modelo de dominio	20
Figura 2 Diagrama de clase del diseño del Paquete Autenticación	27
Figura 3 Diagrama de secuencia del diseño de la HU Iniciar sesión	27
Figura 4 Diagrama de despliegue del sistema.....	28
Figura 5 Modelo entidad-relación	29
Figura 6 Android MVC implementación.....	31
Figura 7 Diagrama de componentes del sistema.....	35
Figura 8 Grafico con información de los resultados de las pruebas	41
Figura 9 Diagrama de clases del diseño Paquete Chat.....	78
Figura 10 Diagrama de clases del diseño Paquete Clase	78
Figura 11 Diagrama de clases del diseño Paquete Control remoto	79
Figura 12 Diagrama de clases del diseño Paquete Evaluación.....	80
Figura 13 Diagrama de clases del diseño Paquete Mano	81
Figura 14 Diagrama de clases del diseño Paquete Pantalla	82
Figura 15 Diagrama de clases del diseño Paquete Video	83
Figura 16 Diagrama de secuencia Terminar sesión	83
Figura 17 Diagrama de secuencia Crear cuenta.....	84
Figura 18 Diagrama de secuencia Editar perfil	84
Figura 19 Diagrama de secuencia Acceder a la clase por autenticación	84
Figura 20 Diagrama de secuencia Seleccionar clase.....	85
Figura 21 Diagrama de secuencia Cerrar sesión del estudiante	85
Figura 22 Diagrama de secuencia Desconectar estudiante de la clase	85
Figura 23 Diagrama de secuencia Buscar clase.....	86
Figura 24 Diagrama de secuencia Notificar estudiante de la evaluación	86
Figura 25 Diagrama de secuencia Transmitir pantalla del profesor en modo completa	86
Figura 26 Diagrama de secuencia Reproducir archivo de video	86
Figura 27 Diagrama de secuencia Parar video	87
Figura 28 Diagrama de secuencia Establecer conversación con el profesor.....	87
Figura 29 Diagrama de secuencia Permitir conversaciones de los estudiantes	87
Figura 30 Diagrama de secuencia Permitir manos levantadas	87
Figura 31 Diagrama de secuencia Aceptar manos levantadas	88
Figura 32 Diagrama de secuencia Denegar mano levantada.....	88
Figura 33 Diagrama de secuencia Notifica a estudiantes de la evaluación	88
Figura 34 Diagrama de secuencia Abrir sitio web remoto	88
Figura 35 Diagrama de secuencia Configurar lenguaje del dispositivo remoto	88
Figura 36 Acta de Aceptación.....	105

INTRODUCCIÓN

INTRODUCCIÓN

“La aplicación de los dispositivos móviles en el aula es un tema de gran trascendencia en el sistema educativo actual, debido a la controversia que suscita entre los docentes. Mientras que algunos centros tienen totalmente prohibido su uso así como su posesión en el recinto escolar, en otros colegios se empieza a aceptar esta circunstancia, intentando nutrirse de la misma introduciéndola en lo posible en los currículos de las asignaturas” (1).

Existen múltiples ventajas de la utilización de estos dispositivos en el área de la educación, algunas de estas son (2):

- Permite la comunicación en tiempo real entre estudiantes, profesores y directivos: Como es el caso del uso de redes sociales y aplicaciones de mensajería instantánea entre estudiantes y profesores, lo cual agiliza el proceso de comunicación y reduce tiempos.
- Agiliza y vuelve más eficiente la distribución de contenidos y materiales: Por ejemplo, las aplicaciones y servicios que permiten almacenar información en la nube y compartirla con otras personas.
- Elimina la barrera geográfica en el aprendizaje: Ya no es necesario que estudiantes y maestros estén todo el tiempo en un salón de clases; además, que permite que el aprendizaje llegue a lugares remotos.
- Promueve que los estudiantes sean más activos durante el proceso de aprendizaje: Ya sea por medio de realizar investigaciones, aprender y utilizar nuevas tecnologías, crear documentos y compartirlos.

La realidad actual muestra un gran auge de las tecnologías, y la educación no es la excepción. Muchos son los centros que están adquiriendo material informático tales como, pizarras digitales regidas por un computador o tabletas electrónicas. Existen experiencias sobre el acercamiento del estudiante a la tecnología en las primeras etapas de la educación. Esto se evidencia en el trabajo con tabletas para la adquisición de competencias lingüísticas o matemáticas lo cual resulta un incentivo para el estudiante y con buenas repercusiones en el rendimiento escolar (1).

Para aprovechar las facilidades de los dispositivos móviles en la educación surgen las aulas tecnológicas. Las cuales cuentan con los recursos necesarios para posibilitarle al estudiante el acceso a la información, la comunicación que proporciona la red y los sistemas de presentación multimedia disponibles en estos tiempos. La informática, Internet, el vídeo, la televisión, y los medios para audiciones y proyecciones son necesidades que tanto el estudiantado como el profesorado

INTRODUCCIÓN

deben tener a su alcance en todo momento. Cada vez más la mayoría de las actividades que se realizan en la escuela se apoyan en el uso estas tecnologías. (3)

El objetivo de las aulas tecnológicas es la creación de un ambiente colaborativo, que propicia la introducción de tecnologías como medios, parte de la didáctica y 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, información y formas de colaboración, motivará la participación y profundización en los temas objeto de estudio (3).

A escala mundial existen disimiles sistemas de gestión de clase que constituyen un componente esencial de las aulas tecnológicas ya que permiten la interacción entre educador, educando y los componentes tecnológicos. Estos sistemas son propios de cada proceso de enseñanza y presentan estándares internacionales que les permite ser comercializados.

En el año 2016 comenzó en el Centro de Tecnologías para la Formación (FORTES) de la Facultad 4 de la UCI, el desarrollo de un software para la gestión de clases en el aula con el empleo de las TIC, generalmente conocido como aula tecnológica. En este caso, el software se ha denominado ATcnea y se ha desarrollado para su funcionamiento sobre tecnología HAIER, ensamblada por la empresa GEDEME y con la colaboración de soluciones libres brindadas por el centro CESOL de la Facultad 1 de la UCI.

Actualmente el ATcnea se encuentra en proceso de desarrollo y carece de recursos que posibiliten la interacción y comunicación de los estudiantes con el profesor a través del software. A petición de la dirección del Centro FORTES, en aras de emplear la variedad tecnológica que provee GEDEME, se decide implementar una solución en el SO Android.

Como vía de solución a la dificultad definida en el contexto anterior se plantea el siguiente **problema de investigación**: ¿Cómo lograr la interacción del estudiante con el software ATcnea desde un dispositivo con el SO Android?

Se ha determinado como **objeto de estudio** de la presente investigación para resolver el problema ilustrado: Desarrollo de aplicaciones para el SO Android vinculadas a los sistemas de gestión de clase. Se enmarca como **campo de acción**: Aplicaciones para el estudiante en el SO Android vinculadas a los sistemas de gestión de clase. El **objetivo general** de esta investigación es: Desarrollar una aplicación para el SO Android que permita la interacción de los estudiantes con el software ATcnea.

Para darle respuesta al objetivo general propuesto se definen los siguientes **objetivos específicos**:

INTRODUCCIÓN

- Elaborar el marco teórico de la investigación donde se relacionen los principales referentes relativos al desarrollo de aplicaciones para el SO Android vinculadas a los sistemas de gestión de clase.
- Realizar el diseño e implementación de una aplicación para darle soporte al software ATcnea en dispositivos móviles con SO Android que contenga características de sistemas similares.
- Realizar pruebas a la propuesta de solución desarrollada para comprobar el correcto funcionamiento de la misma.

Para dar cumplimiento a los objetivos específicos se planifican las siguientes **tareas de investigación**:

- Análisis de las diferentes tecnologías para el desarrollo de aplicaciones Android para aulas inteligentes.
- Realización de un estudio acerca de las tecnologías utilizadas en desarrollo de aplicaciones Android para aulas tecnológicas.
- Realización de un estudio acerca de las metodologías de desarrollo de software existentes.
- Selección de las metodologías y tecnologías a usar para darle cumplimiento al objetivo propuesto.
- Levantamiento de los requisitos funcionales y no funcionales de la solución propuesta.
- Realización del diseño de la aplicación propuesta.
- Generación de los principales artefactos de la metodología de desarrollo seleccionada.
- Implementación de la aplicación haciendo uso de las tecnologías y herramientas seleccionadas.
- Realización de pruebas a la aplicación

Los **métodos investigativos** que se utilizan para la presente investigación son:

Métodos teóricos:

Análisis Histórico-Lógico: En la presente investigación se utilizó este método para realizar el estudio del estado del arte, para investigar acerca de otras aplicaciones o soluciones similares y de los lenguajes y metodologías de desarrollo de software existentes.

Modelación: Se empleó para el modelado de los diagramas de clases, el diseño de la interfaz gráfica y el modelado del flujo de datos de la aplicación con el objetivo de lograr una mejor comprensión del flujo de desarrollo de la aplicación.

Métodos empíricos:

INTRODUCCIÓN

Observación: Se utilizó para caracterizar las soluciones similares, teniendo en cuenta sus características y deficiencias de las mismas.

Posibles resultados

- Una aplicación que contribuya a la interacción de los estudiantes con el software ATcnea desde dispositivos Android con su correspondiente documentación.

Para una mejor comprensión de la investigación, se definió una **estructura capitular** que aporte cierto grado de organización y facilite el estudio del documento. Los capítulos que lo conforman son los siguientes:

Capítulo 1: Fundamentación teórica.

Este capítulo abarca los principales fundamentos y enfoques teóricos sobre las aulas tecnológicas así como también sobre los sistemas de gestión de clase.

Capítulo 2: Descripción de la solución propuesta.

El capítulo contiene la descripción de todos los aspectos a tener en cuenta para la confección de la propuesta de solución. Además se presenta el modelo de dominio y se especifican los requisitos que debe cumplir el sistema.

Capítulo 3: Implementación y prueba

Este apartado contiene detalles del proceso de implementación de la propuesta de solución. También se incluyen pormenores del proceso de pruebas, definiéndose los tipos y niveles de pruebas con el fin de liberar un software de calidad que cumpla los objetivos propuestos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se precisan un conjunto de conceptos y fundamentos que constituyen el marco teórico, relacionado con el objeto de estudio definido en la investigación. Se destacan características de los elementos asociados al tema a investigar, así como la situación del mismo tanto en el ámbito nacional como el internacional. Se exponen la metodología y las principales herramientas que sirven de apoyo para la realización de la solución al problema planteado.

1.2. Tecnologías para el apoyo del proceso enseñanza-aprendizaje

Las nuevas tecnologías poseen características que las convierten en herramientas poderosas a utilizar en el proceso de aprendizaje de los estudiantes: inmaterialidad, interactividad, elevados parámetros de calidad de imagen y sonido, instantaneidad, digitalización, interconexión, diversidad e innovación. (4)

La inmaterialidad proporciona información y la capacidad de construir mensajes sin necesidad de vínculos externos. Ofrece a los estudiantes la oportunidad de construir conocimiento sin espacios o materiales que se encuentren físicamente en su entorno.

Las nuevas tecnologías ofrecen la capacidad de interacción entre los estudiantes, donde no sólo elaboran mensajes (actividad también realizable con otras tecnologías más tradicionales), sino que además pueden decidir la secuencia de información por seguir, establecer el ritmo, cantidad y profundización de la información que desea, y elegir el tipo de código con el que quiere establecer relaciones con la información. (4)

Los elevados parámetros de calidad de imagen y sonido no tratan sólo de manejar información de manera más rápida y transportarla a lugares alejados, sino también de que la calidad y confiabilidad de la información sea elevada. Los sonidos y las imágenes son herramientas que fomentan la creatividad de los estudiantes, estimulan su aprendizaje al crear riqueza en el contexto impartido. (4)

La información se recibe en las mejores condiciones técnicas posibles y en el menor tiempo permitido, preferentemente en tiempo real, por medio de la instantaneidad. (4)

La digitalización consiste en transformar la información codificada analógicamente en códigos numéricos, que permiten la manipulación y la distribución más fácilmente. Por medio de la

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

digitalización, los estudiantes tienen acceso al material de clase y a un sinnúmero de obras y libros de texto, sin necesidad de cargar con ellos físicamente, de forma virtual, pueden encontrar cualquier material de apoyo que necesiten. (4)

A través de la interconexión, se forma una nueva red de comunicación de manera que se refuercen mutuamente, y eso lleva a un impacto mayor que el de las tecnologías utilizadas individualmente. Permite la conexión constante entre los estudiantes y su profesor, para así crear una red colaborativa, donde no existen barreras de tiempo ni espacio. La diversidad es una característica de las nuevas tecnologías que debe entenderse desde una doble posición: primeramente, que en lugar de encontrar tecnologías unitarias, se tienen tecnologías que giran en torno a algunas de las características citadas; y en segundo lugar, existe una diversidad de funciones que las tecnologías pueden desempeñar, desde las que transmiten información exclusivamente, como los videodiscos, hasta aquellas que permiten la interacción entre los usuarios, como la videoconferencia. (4)

La integración de las tecnologías de la computación con la telecomunicación se llama convergencia digital, y permite el uso simultáneo de herramientas de voz, textos, datos e imágenes, por medios electrónicos, que convergen en un mismo canal, a través de diferentes tecnologías. (5)

La antes mencionada convergencia digital ha tocado todas las puertas de la sociedad moderna. Producto del uso de los diferentes canales de comunicación y de las actuales herramientas tecnológicas en el recinto escolar, más específicamente en los salones de clases surgen las aulas tecnológicas también denominadas como: aulas inteligentes, aulas interactivas o aulas digitales.

1.2.1. Aulas Tecnológicas

Este término de conjunto con sus otras denominaciones ha sido conceptualizado por diferentes autores.

- Segovia (5) define: Comunidad de aprendizaje, cuyo objetivo principal es el desarrollo de la inteligencia y de los valores de los estudiantes, que planifican, realizan y regulan su propio trabajo, bajo la mediación de los profesores, por medio de métodos didácticos diversificados y tareas auténticas, evaluados por estudiantes y profesores, en un espacio multiuso abierto, tecnológicamente equipado y organizado según los principios de la calidad total en la gestión.
- Blesa (6) concreta: Aulas en las que se encuentren los recursos tecnológicos necesarios para posibilitar al alumnado el acceso a la información y a la comunicación que nos proporciona la red y los sistemas de presentación multimedia disponibles en estos tiempos. La informática,

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Internet, el vídeo, la televisión, y los medios para audiciones y proyecciones son necesidades que tanto el alumnado como el profesorado deben tener a su alcance en todo momento.

Para esta investigación se tomará como definición del término “Aula Tecnológica” lo precisado por Blesa (6), porque se adapta de manera más eficiente a los términos tratados en esta indagación.

1.2.2. Objetivo de las Aulas tecnológicas

El objetivo de las aulas tecnológicas es la creación de un ambiente colaborativo, que propicia la introducción de tecnologías como medios, parte de la didáctica y forma de enriquecimiento del contenido académico de la figura docente de que se trate y permite a profesores y educandos establecer una profunda comunicación, cuya interactividad en el intercambio de ideas e información, formas de colaboración, motivará la participación y profundización en los temas objeto de estudio. Se desarrolla en un nuevo entorno de aprendizaje, donde el estudiante podrá manejar un conjunto organizado de conocimientos que le permita ser capaz de analizar el mundo que le rodea y tomar decisiones. En un "aula tecnológica", los contenidos son diferentes de los procesos. El programa organiza ambientes, intenciones y experiencias e impulsa el desarrollo del pensamiento crítico y aplica el conocimiento. (7).

1.2.3. Dispositivos móviles en el proceso de enseñanza-aprendizaje

Las redes virtuales móviles facilitan la movilidad del conocimiento, lo que permite acceder a este en cualquier momento y lugar. Las mismas poseen características que permiten las oportunidades para un cambio de paradigma educativo en el contexto de la sociedad digital.

El enfoque metodológico y el modelo pedagógico sobre el que se sustente el uso de esta tecnología en la educación, se orienta a la participación, el diálogo, la construcción conjunta y autogestión. La educomunicación va más allá de los cambios tecnológicos acontecidos a partir del desarrollo de la Web 2.0: con nuevas o viejas tecnologías es imprescindible preguntarse sobre nuevas maneras de enseñar y aprender. Los cambios metodológicos, la búsqueda de nuevos modelos pedagógicos y las prácticas interactivas basadas en el diálogo son cuestiones que están más allá del uso de una tecnología u otra.

El término *mobile learning* (aprendizaje móvil) empieza a utilizarse a finales de la década de los años noventa, con el uso de las agendas electrónicas en el ámbito educativo. Pero mientras los dispositivos móviles evolucionaban tecnológicamente y se extendía su uso, desde el ámbito educativo se observaba como un fenómeno externo. Según se recoge en la Guía Mobile Learning de Fundación Telefónica (8): “a pesar de la ubicuidad y los tipos de aprendizaje que pueden reforzar, a menudo

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

estas tecnologías están prohibidas o ignoradas en los sistemas educativos formales". Esto representa una oportunidad perdida, ya que el potencial de estos aparatos es muy grande y seguirá creciendo.

El aprendizaje móvil puede entenderse como una evolución del *e-learning* en un contexto en el que se posibilita al alumnado el aprovechamiento de las ventajas de las tecnologías móviles como soporte al proceso de aprendizaje.

Además, de las ventajas descritas al inicio, se pueden añadir: (9)

- Facilita los procesos de comunicación horizontal y bidireccional.
- Potencia la creación y recreación del conocimiento en red, facilita la conexión de redes y el desarrollo de comunidades de enseñanza y aprendizaje.
- Motiva la creatividad e intercreatividad.
- Facilita el aprendizaje personalizado y el desarrollo de Entornos Personales de Aprendizaje (*Personal Learning Environments*, PLEs).
- Potencia el aprendizaje permanente, en un entorno virtual personal y cotidiano.
- Fortalece la alfabetización digital y mediática, competencia imprescindible en el contexto de la sociedad digital.

Las aplicaciones móviles facilitan la flexibilidad y multiplicidad de las funciones de los dispositivos móviles. El éxito de una aplicación radica en la sencillez de la aplicación, la usabilidad y accesibilidad, sumado al diseño atractivo, la disponibilidad, la diversidad temática y la adaptabilidad a las necesidades del usuario. Desde el punto de vista educativo, a estas características es necesario añadir aquellos aspectos que pueden ayudar tanto a profesorado como a alumnado a mejorar los procesos de enseñanza y aprendizaje. En este sentido, la aplicación de la tecnología móvil en cualquier nivel y contexto educativo, ya sea de educación formal o informal, debe contemplar los principios educomunicativos en su diseño pedagógico.

1.2.4. Sistemas gestores de clase

"Entre las tecnologías con mayores resultados y uso en el mundo para mejorar los estándares de aprendizaje están las llamadas Aulas Inteligentes, las cuales son el futuro inmediato en cuanto a medios técnicos desplegados con mayores potencialidades pedagógicas, según los diferentes niveles educacionales en que pueden ser utilizados" (9). Actualmente se ha desarrollado esta solución tecnológica con la finalidad de brindar una experiencia única de instrucción. Puede decirse que es una solución educativa que revoluciona el método de enseñanza-aprendizaje, que brinda una

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

experiencia única en el aula. Este innovador concepto educativo ha dado paso al desarrollo de misceláneas de sistemas que aplican este término de los cuales podemos mencionar:

Mythware

Mythware es un sistema de gestión de aula, caracterizado por su estabilidad, desde su fundación en el 2007. Se define como una herramienta de trabajo que habilita un ambiente virtual donde el docente y los estudiantes pueden colaborar e interactuar a través de la tableta en el aula, para fines educativos. Utiliza la red escolar para cumplir con este fin, el profesor crea un ambiente o aula virtual al cual los estudiantes pueden conectarse a través de una serie de tareas e instrucciones asistidas por el profesor.

Dicho sistema permite el uso adecuado de la tecnología durante las horas de clase, ayuda al profesor a enfocar el uso de la tecnología al flujo natural de la clase de una manera rápida y con simples tareas. Contribuyéndose a que la tecnología sea una verdadera herramienta de trabajo y un medio efectivo para el aprendizaje y la enseñanza (10).

Este software permite ver y controlar la plataforma del estudiante y la del profesor, teniendo más control la plataforma del profesor.

Tabla 1 Funcionalidades de la app Mythware

	Funcionalidades	Android	Linux
Enseñanza & Aprendizaje Interactivo	Transmisión de Pantalla	x	x
	Transmisión de Voz	x	
	Demostración del Estudiante	x	x
	Película en Red	x	
	Transmisión por Cámara	x	
	Pizarra Interactiva	x	x

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

	Respuesta & Competencia	x	
Control & Apoyo	Administración & Control	x	x
	Control Web		
	Política de Aplicación	x	x
Comando Remoto	Aplicación de Ejecución	x	x
	Abrir sitio web de forma remota	x	x
	Iniciar aplicación de forma remota		x
	Apagar	x	x
Colaboración en grupo	Gestión de grupo		
	Enseñanza de grupo		
	Chat		
Gestión de Archivos	Distribución de Archivos	x	x
	Colección de Archivos	x	x
	Entrega de archivos	x	x
Otros	Iniciar sesión(Registro de estudiantes)	x	x

ITALC

ITALC es una herramienta didáctica que gestiona los procesos que se realiza un profesor en un aula. Ofrece los servicios de ver y controlar otros equipos de su red de varias maneras. Es compatible con Linux y Windows XP, Vista y 7 e incluso puede ser utilizado de forma transparente en entornos mixtos. Este sistema se caracteriza por ser un software con licencia libre o gratis.

ITALC ha sido diseñado para simular de una manera interactiva las tareas que se realizan en el aula. Por lo tanto, ofrece muchas posibilidades a los profesores tales como (11):

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Ver lo que sucede en el equipo del estudiante, utiliza el modo de vista previa y se pueden hacer capturas de pantalla.
- Tomar el control remoto del ordenador para ayudar al estudiante.
- Ver una demostración en vivo (ya sea en pantalla completa o en una ventana) de la pantalla del profesor en todas las computadoras de los estudiantes.
- Bloquear el ordenador para centrar la atención en el profesor.
- Enviar mensajes a los estudiantes en forma de texto.
- Encender, apagar y reiniciar los equipos a distancia.
- Abrir y cerrar una sesión remota.
- Ejecutar comandos / scripts.
- Educación en el hogar - la tecnología de red no se limita a una subred, por lo que los estudiantes pueden participar desde su casa a través de las lecciones de VPN - simplemente mediante la instalación del cliente con la clave pública iTALC.

AB Tutor: Gestión del aula

Ofrece funciones clave de administración del aula, como supervisión, inicio de sesión, transmisión, control remoto, cierre de sesión y bloqueo de acceso. También proporciona un potente administrador de políticas que le permite al profesor controlar las aplicaciones y sitios web a los cuales un estudiante podría acceder. El software ha sido diseñado cuidadosamente para ofrecer a los administradores de red las potentes funcionalidades de administración que requieren, al mismo tiempo que sigue siendo lo suficientemente limpio y sencillo para que los profesores puedan utilizarlo bajo la presión del aula. Este software presenta una licencia privativa y se caracteriza por ser multiplataforma (12).

Tabla 2 Funcionalidades de la app AB Tutor

Funcionalidades	AB Tutor
Gestión de asistencia	
Bloquear Monitores de estudiantes	x
Mostrar vista de estudiante	x
Colaboración	x
Libro de calificaciones	
Evaluaciones individualizadas	

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Gestión del Plan de Clases	x
Grabación de actividad en la pantalla	

Conclusiones del estudio

En cada uno de los sistemas anteriores se identificaron funcionalidades comunes tales como evaluaciones individualizadas, ver lo que sucede en el equipo del estudiante, enviar mensajes a los estudiantes y transmisión de pantalla entre otras. Estas funcionalidades son imprescindibles para un aula tecnológica por lo que cada una de ellas se tendrá en cuenta para conformar los requisitos de la propuesta de solución.

1.3. Metodología, tecnologías, lenguajes y herramientas

Para el desarrollo de la propuesta de solución no fue necesario realizar un estudio de metodologías, tecnologías y herramientas debido a que la presente investigación se rige por las especificaciones expuestas en los documentos de Arquitectura de Software versión 1.2 y el Plan de desarrollo de Software versión 1.7 del expediente de desarrollo 4.0 definido por el proyecto ATcnea. A continuación se reseñan cada uno de estos aspectos:

1.3.1. Metodologías de desarrollo de software

Durante el desarrollo de todo proyecto de software se presentan situaciones que en muchos casos se hacen difíciles de controlar. Para dar respuesta a este problema, se hicieron estudios que devinieron en la creación de las metodologías de desarrollo de software y en diversos estándares. Dichas metodologías y estándares proporcionan las guías para un desarrollo controlado y basado en soluciones exitosas, de forma tal, que gane en agilidad el proceso y en calidad el producto (13). Formalmente se pueden definir estas metodologías como un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software (14).

No obstante, cada software cuenta con necesidades y entornos diferentes, que deben tenerse en consideración. Por otra parte cada metodología tiene características propias del ambiente de desarrollo, constituyendo estos los principales aspectos que rigen el proceso de selección de la metodología adecuada. Una metodología tiene que ser usada de forma simple y efectiva, sin hacer las tareas de diseño excesivamente complejas (15).

Por las razones antes expuestas se definen dos clasificaciones muy generales en las que se encapsulan las metodologías existentes: metodologías estructuradas y metodologías orientadas a

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

objetos. Por otra parte, considerando la filosofía de desarrollo se definen dos nuevas clasificaciones comúnmente conocidas como: las metodologías ágiles y las tradicionales o pesadas.

- **Metodologías tradicionales**

Las metodologías tradicionales o pesadas cuentan con un proceso mucho más controlado. Están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo y se centran en el alcanzar una arquitectura estable. No obstante se ven altamente afectadas por los cambios, tienen altos costos de producción y mucho más trabajo de documentación (14). Este tipo de metodologías también se caracterizan por su rigidez metodológica. Un ejemplo de estas metodologías es *Rational Unified Process (RUP)*, basado en *Unified Modeling Language (UML)* para la ingeniería de sistemas y de software. (16)

- **Metodologías ágiles**

Las metodologías ágiles nacen para solventar la necesidad de un desarrollo de software rápido y con la calidad requerida. Estas son una agrupación de las prácticas tradicionales de producción de código pero llevadas al extremo, mediante las cuales se buscan entregas rápidas y oportunas hasta la entrega final del producto. Su mayor fortaleza se evidencia en la resistencia a los cambios sin un incremento radical del costo inicial de producción (17).

En la actualidad no es recomendable ejecutar un desarrollo software de una manera lenta. Las exigencias de los clientes actuales hacen crecer la necesidad de implementar soluciones rápidas y que cumplan con los requerimientos planteados, por lo que el Desarrollo Rápido de Aplicaciones (RAD por sus siglas en inglés) es una de las características que más impacto posee. Para solventar esto se deben utilizar herramientas basadas en este nuevo enfoque.

Este es el caso particular de Proceso Unificado Ágil (AUP) es una versión simplificada del Proceso Unificado de Desarrollo (RUP). Describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software, usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. AUP puede verse más simple que las versiones más clásicas del proceso unificado, mientras que desde el punto de vista ágil, puede entenderse como una versión más pesada que las metodologías ágiles (18).

La utilización de AUP ofrece numerosas ventajas al equipo de Desarrollo, entre ellas se pueden mencionar (18) :

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

El personal sabe lo que están haciendo: Las personas no van a leer documentación de procesos detalladamente, pero quieren alguna guía de alto nivel y/o capacitaciones de vez en cuando. Este producto proporciona vínculos a muchos de los detalles, si está interesado, pero no los obliga ni los impone.

Simplicidad: Todo se describe brevemente utilizando pocas páginas y no miles de ellas.

Agilidad: Permite responder de manera rápida y apropiada a los cambios.

Enfocar las actividades de alto nivel: Se centra en las actividades que realmente cuentan, no en cada cosa posible que podría pasarle al proyecto.

Independencia de Herramientas: Se puede usar el conjunto de herramientas que se desee con AUP.

- **Metodología a utilizar**

Variación de AUP para la UCI.

En la Universidad de las Ciencias Informáticas, al no existir una metodología de *software* universal, se desarrolló una variante de la antes mencionada metodología. Esta metodología fue adaptada a las características de los proyectos de la UCI exigiéndose así que el proceso sea configurable de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

Variación AUP-UCI al igual que AUP consta de tres fases las cuales son (19) :

- Inicio.
- Ejecución.
- Cierre.

AUP propone 7 disciplinas pero para la variación del ciclo de vida de los proyectos de la UCI se tienen 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Definidas de la siguiente forma:

1. Modelado de negocio.
2. Requisitos.
3. Análisis y diseño.
4. Implementación.
5. Pruebas internas.
6. Pruebas de liberación.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

7. Pruebas de aceptación.
8. Despliegue.

Además, la variación de AUP UCI posee cuatro escenarios de ellos se aplicará el escenario No 4 ya que el proyecto no modela el negocio, solo se puede modelar el sistema. De esta manera en la presente investigación la descripción de los requisitos del sistema se hará a través de Historias de usuario.

1.3.2. Tecnologías a emplear

- **Sistema Operativo Android**

Existen muchas plataformas para dispositivos móviles (IOS, Android, Windows Phone, Blackberry, Symbian, Palm, Java Mobile Edition, Linux Mobile), sin embargo, Android posee una serie de cualidades que lo hacen único. Es el primero que combina en una misma solución cualidades como: (20)

- Plataforma realmente abierta: es una plataforma de desarrollo libre y basada en Linux y código abierto. Una de sus grandes ventajas es que se puede usar y personalizar en sistemas sin pagar regalías.
- Adaptable a cualquier tipo de hardware: Android no ha sido diseñado exclusivamente para su uso en teléfonos y tabletas. Hoy en día se puede encontrar relojes, cámaras, electrodomésticos y una gran variedad de sistemas empotrados que se basa en este sistema operativo.
- Portabilidad asegurada: las aplicaciones finales son desarrolladas en Java lo que asegura que podrán ser ejecutadas en cualquier tipo de CPU, tanto presente como futuro. Esto se consigue gracias al concepto de máquina virtual.
- Arquitectura basada en componentes inspirados en Internet: por ejemplo, el diseño de la interfaz de usuario se hace en XML, permitiendo que una misma aplicación se ejecute en un móvil de pantalla reducida o en un TV.
- Filosofía de dispositivo siempre conectado a Internet.
- Gran cantidad de servicios incorporados: por ejemplo, la localización basada tanto en GPS como en redes, base de datos con SQL, reconocimiento y síntesis de voz, navegador, multimedia, etc.
- Aceptable nivel de seguridad: los programas se encuentran aislados unos de otros gracias al concepto de ejecución dentro de una caja que hereda de Linux. Además, cada aplicación

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

dispone de una serie de permisos que limitan su rango de actuación (servicios de localización, acceso a Internet, etc.).

- Optimizado para baja potencia y poca memoria: por ejemplo, Android utiliza la máquina virtual Dalvik (aunque esta fue cambiada en la versión 4.4 en favor de Android Runtime (ART)). Se trata de una implementación de Google de la máquina virtual Java optimizada para dispositivos móviles.
- Alta calidad de gráficos y sonido: gráficos vectoriales suavizados, animaciones inspiradas en flash, gráficos en 3D basados en Open Graphics Library (OpenGL) 10. Incorpora los codecs estándares más comunes de audio y video, incluyendo H.264/MPEG-4 AVC, MP3, ACC, entre otros.

Android fue seleccionado por el cliente como SO base para el dispositivo del estudiante, pero en conclusión se puede añadir que este SO ofrece una forma potente y simple de implementar poderosas aplicaciones para diferentes tipos de dispositivos y también cabe destacar la cuota de mercado alcanzada en el 2016 de un 87 % (22).

- **Android SDK**

“El Android SDK es el paquete que brinda las librerías y herramientas necesarias para construir, probar y depurar aplicaciones para Android”. A través de este, *Google* ofrece un paquete completo que incluye depurador, emulador, documentación, *demos* y el código de muchas aplicaciones. Este paquete de desarrollo no funciona por sí solo, sino que es necesario que sea importado a algún IDE (21).

- **Android Studio v2.2**

Android Studio es el Entorno de Desarrollo Integrado (IDE) oficial para la plataforma Android. Está basado en el software *IntelliJ IDEA* de JetBrains, y es publicado de forma gratuita a través de la Licencia *Apache 2.0*. Está disponible para las plataformas *Microsoft Windows*, *Mac OS X* y *GNU/Linux*. *Android Studio* está diseñado específicamente para el desarrollo de aplicaciones para en sistema operativo *Android*. (22)

Principales características (22):

- Soporte para programar aplicaciones para *Android Wear* (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- Nuevo diseño del editor con soporte para la edición de temas.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Nueva interfaz específica para el desarrollo en Android.
- Permite la importación de proyectos realizados en el entorno Eclipse.
- Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar *Mercurial*, *Git*, *Github* o *Subversion*.
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Vista previa en diferentes dispositivos y resoluciones.
- **Visual Paradigm v5.0**

Visual Paradigm para UML (*Unified Modeling Language* en inglés) es una herramienta CASE (del inglés, *Computer Aided Software Engineering*) aplicable en todo el ciclo de vida del desarrollo de software. Soporta UML, SysML (del inglés, *Systems Modeling Language*), BPMN (del inglés, *Business Process Modeling Notation*), entre otras tecnologías. Permite dibujar todos los tipos de diagramas de clases, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones (23).

Visual Paradigm también ofrece (23):

- Navegación intuitiva entre la escritura del código y su visualización.
- Potente generador de informes en formato PDF/HTML.
- Documentación automática Ad-hoc.
- Ambiente visualmente superior de modelado.
- Sincronización de código fuente en tiempo real.

1.3.3. Bibliotecas/Librerías de terceros

- **MaterialDrawer**

Proporciona una implementación muy sencilla de un cajón de navegación para aplicaciones Android. Posee un encabezado con perfiles (*AccountHeader*), un *MiniDrawer* para tabletas además proporciona *DrawerItems*, colores y temas personalizados. (24)

En la propuesta de solución será utilizada para crear el menú principal. Desde este menú el usuario podrá acceder a las diferentes secciones y funcionalidades de forma rápida y sencilla.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- **FFmpegMediaPlayer**

FFmpegMediaPlayer es una reimplementación de la clase MediaPlayer de Android. La clase proporciona una interfaz unificada para reproducir audio, video y *streams*. (25)

Protocolos compatibles: file, http, https, mms, mmsh and rtmp

Formatos compatibles (audio y video): aac, acc+, avi, flac, mp2, mp3, mp4, ogg, 3gp (...)

Sera utilizada para independizar la reproducción de multimedia en la propuesta de solución de los codecs del SO Android. Lo que permitirá tener una mayor compatibilidad con los formatos actuales de transmisión y reproducción de audio y video.

- **Otto**

Otto es un bus de eventos diseñado para desacoplar diferentes partes de su aplicación, al tiempo que les permite comunicarse de manera eficiente. (26)

En la propuesta de solución se utilizará con el objetivo de proporcionar un canal de notificaciones y eventos para las distintas funcionalidades.

- **GreenDAO**

GreenDAO es un *Object Relational Mapping* (ORM) ligero y rápido para el SO Android que mapea objetos a bases de datos SQLite. Al ser altamente optimizado para el Android, greenDAO ofrece un gran rendimiento y consume un mínimo de memoria. (27)

Sera el encargado, en la propuesta de solución, de manejar los procesos relacionados con la Base de Datos.

1.4. Conclusiones parciales

En el presente capítulo se realizó la fundamentación teórica de la investigación a través del estudio de los conceptos relacionados los sistemas gestores de aulas tecnológicas a escala mundial y sus características. Además, se efectuó la caracterización de la metodología, lenguajes, tecnologías y herramientas a emplear en el desarrollo de la solución. La investigación arrojó los siguientes resultados:

- Se concluye, luego de realizar un estudio de soluciones similares, que ninguno de los sistemas gestores de aulas tecnológicas a escala mundial responde al proceso de soberanía tecnológica que se está llevando a cabo en el país.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Se determinaron las principales funcionalidades que una aplicación destinada al estudiante debe tener y algunas que, aunque no son obligatorias, enriquecen el proceso de enseñanza aprendizaje.
- La metodología a utilizar para guiar el desarrollo de la solución fue AUP en su variante UCI dado que la misma genera una serie de actividades y artefactos necesarios para el cumplimiento de los objetivos en cada una de las fases del ciclo de vida del sistema.
- Se utiliza UML como lenguaje de modelado y Visual Paradigm para UML v8.0 como herramienta para el modelado del sistema.
- Como IDE a utilizar Android Studio v2.2.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1. Introducción

En el presente capítulo se describe la solución propuesta identificando los usuarios que interactúan con la aplicación, así como los requisitos funcionales y no funcionales. También, siguiendo la metodología AUP UCI en concordancia con el escenario escogido, se desarrollarán los artefactos pertinentes.

2.2. Modelo de dominio

Un modelo de dominio es una representación visual de las clases conceptuales significativas en un dominio (28). Se utiliza con frecuencia como fuente para el diseño de los objetos de software. Es el artefacto más importante que se crea durante el análisis orientado a objetos, debido a que propicia una mayor comprensión de los conceptos que se tratarán en la aplicación.

Diagrama de clases del Modelo de dominio

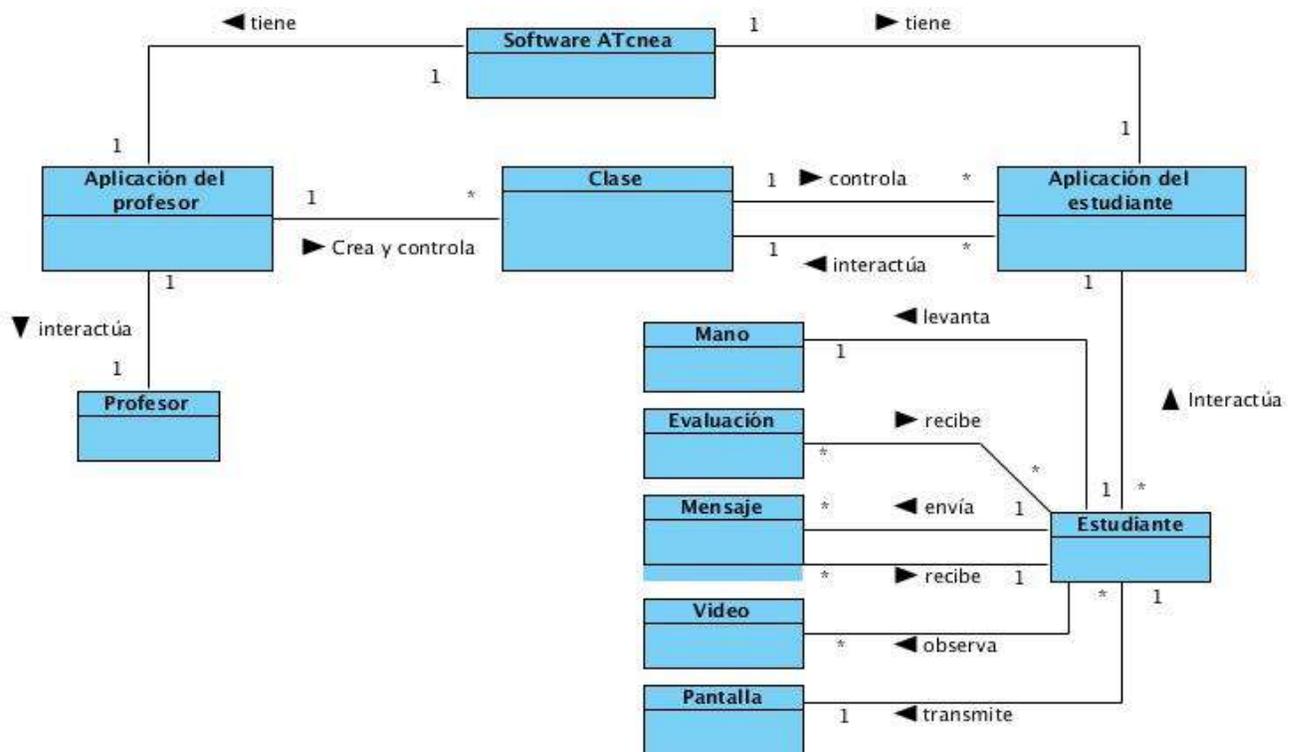


Figura 1 Modelo de dominio

Definición de las clases del modelo de dominio

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Software ATcnea: Representa el paquete de software con el que trabaja el aula tecnológica.

Profesor: Representa a los usuarios que encarnan el rol de profesor en el aula tecnológica.

Estudiante: Representa a los usuarios que encarnan el rol de estudiante en el aula tecnológica.

Aplicación del profesor: Representa el software con el que interactúa el profesor y se encarga de crear las clases para los estudiantes y de disponer de funcionalidades que a grandes rasgos permiten controlar las clases.

Aplicación del estudiante: Representa el software con el que interactúa el estudiante y se encarga de garantizar el acceso a las clases creadas por el profesor, así como a todas sus funcionalidades.

Clase: Representa la abstracción de una clase real a nivel de software.

Mano: Representa la abstracción de levantar la mano o pedir la palabra en la clase a nivel de software.

Evaluación: Representa un calificativo que se da al estudiante, por parte del profesor, para medir su desempeño en las actividades de la clase.

Mensaje: Representa los mensajes que se envían entre los usuarios del aula tecnológica utilizando el software ATcnea.

Video: Representa un audio-visual del que se auxilia el profesor para apoyar la clase.

Pantalla: Representa la funcionalidad que permite al profesor compartir su escritorio y ver el de los estudiantes.

2.3. Propuesta de solución

El sistema propuesto se corresponde con una aplicación para dispositivos móviles con SO Android. Este permitiría que los estudiantes interactúen con todos los componentes del aula tecnológica. Los estudiantes por medio de una tableta una vez autenticados podrán ver las clases que están activas y conectarse a una de ellas. También podrán observar contenido multimedia, ver la pantalla del profesor, chatear y ser evaluados. Por otra parte, el sistema propuesto le permitirá al profesor tener un control total de los dispositivos de cada estudiante para garantizar así la atención total en la clase. Los profesores pueden realizar acciones remotas como abrir y cerrar aplicaciones, abrir sitios web y bloquear el dispositivo del estudiante. Este sistema beneficiará tanto al estudiante como al profesor ya que les dará una nueva vía de comunicación permitiendo aumentar la integración de las nuevas tecnologías en las aulas.

2.4. Requisitos de software

La obtención y análisis de los requisitos constituye una de las etapas fundamentales en el proceso de ingeniería de requisitos. En esta actividad los ingenieros de software trabajan en conjunto con los clientes y usuarios finales, para determinar el dominio de la aplicación, los servicios que debe proporcionar, el rendimiento requerido para el sistema, así como las restricciones del hardware. Estos requisitos que no son más que una especificación del diseño del software, es decir una descripción abstracta del este que es la base para un diseño e implementación detallada (29).

Existen dos tipos de requisitos; funcionales y no funcionales. A continuación, se exponen cada uno de estos requisitos.

2.4.1. Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y como el sistema debe comportarse en distintas situaciones. En algunos casos estos requisitos también pueden declarar explícitamente lo que el sistema no debe hacer (30).

A continuación, se listan los requisitos funcionales identificados:

RF 1. Iniciar sesión. El sistema debe permitir al estudiante iniciar sesión en el sistema.

RF 2. Terminar sesión. El sistema debe permitir al estudiante terminar la sesión en el sistema.

RF 3. Crear una cuenta. El sistema debe permitir al estudiante crear una cuenta para acceder al sistema.

RF 4. Editar perfil. El sistema debe permitir al estudiante del sistema editar un perfil creado con anterioridad.

RF 5. Acceder a la clase por autenticación. El sistema debe permitir que los estudiantes entren a la clase luego de autenticarse correctamente.

RF 6. Seleccionar clase. El sistema debe permitir al estudiante seleccionar una clase de la lista de clases.

RF 7. Cerrar sesión de estudiante. El sistema debe permitir al estudiante desconectarse de la clase en la que se encuentra.

RF 8. Desconectar estudiante de la clase. El sistema debe permitir al profesor desde su ordenador desconectar un estudiante determinado de la clase.

RF 9. Buscar clase. El sistema debe permitir al estudiante buscar las clases creadas en el sistema.

RF 10. Notificar estudiantes de la evaluación. El sistema debe permitir al profesor emitir desde el ordenador la evaluación al estudiante mediante una notificación.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

RF 11. Transmitir pantalla del profesor en modo completa. El sistema debe permitir al profesor transmitir la pantalla de su ordenador en los dispositivos de uno o varios estudiantes en la clase, en modo de pantalla completa. En caso de los estudiantes, estos solo podrán observar la pantalla del profesor.

RF 12. Ver pantalla de los estudiantes. El sistema debe permitir al profesor ver la pantalla del dispositivo de un estudiante.

RF 13. Reproducir archivo de video. El sistema debe permitir al profesor desde su ordenador reproducir un archivo de video en su pantalla y transmitirlo hacia los dispositivos de los estudiantes conectados a la clase.

RF 14. Parar video. El sistema debe permitir al profesor desde su ordenador parar la reproducción del video que se está ejecutando en la clase.

RF 15. Establecer conversación con el profesor. El sistema debe permitir al profesor enviar y recibir mensajes a los estudiantes y viceversa.

RF 16. Permitir conversaciones de los estudiantes. El sistema debe permitir al profesor desde su ordenador bloquear o desbloquear el chat a uno o varios estudiantes.

RF 17. Permitir manos levantadas. El sistema debe permitir al profesor desde su ordenador bloquear y desbloquear la acción de levantar la mano en los dispositivos de uno o varios estudiantes.

RF 18. Aceptar mano levantada. El sistema debe permitir que el profesor desde su ordenador acepte la solicitud de atención emitida por los estudiantes (mano levantada).

RF 19. Denegar mano levantada. El sistema debe permitir al profesor denegar desde su ordenador la solicitud de atención enviada desde el dispositivo de los estudiantes.

RF 20. Iniciar aplicación remota por el profesor. El sistema debe permitir al profesor desde su ordenador iniciar remotamente una aplicación en los dispositivos de los estudiantes.

RF 21. Abrir sitio web remoto. El sistema debe permitir al profesor desde su ordenador abrir remotamente un sitio web en el dispositivo de los estudiantes

RF 22. Configurar lenguaje del dispositivo remoto. El sistema debe permitir al profesor desde su ordenador seleccionar el lenguaje del dispositivo remoto.

2.4.2. Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. En si se refiere a la fiabilidad, al tiempo de respuesta y la capacidad de almacenamiento que proporciona el sistema. A menudo son aplicados al sistema en su totalidad, aunque normalmente se aplican a características o servicios individuales del sistema (30).

A continuación, se listan los requisitos no funcionales identificados:

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Usabilidad

RNF 1. Cumplir con las pautas de diseño establecidas en la Estrategia Marcaría de la Universidad.

Confiabilidad

RNF 2. Verificar las consecuencias asociadas a los fallos del sistema.

Portabilidad

RNF 3. Permitir la instalación de la herramienta en tablets que cuenten con NovaDroid basado en Android superior a la versión 5. Se recomienda 1 GB RAM o superior, 7 pulgadas o superior de pantalla, CPU/GPU Dual Core 1.0GHz o superior.

2.5. Descripción de los usuarios

Tabla 3 Descripción de usuarios

Usuario	Descripción
Estudiante	Se refiere a los usuarios que encarnan el rol de estudiante en el aula tecnológica.
Profesor	Se refiere a los usuarios que encarnan el rol de profesor en el aula tecnológica.

2.6. Historia de usuarios

Las historias de usuario (HU) es una de las técnicas que utiliza la metodología AUP en su versión UCI para especificar los requisitos del *software*. Son tarjetas en las cuales se describen brevemente las características que el sistema debe poseer, utilizando para ello un lenguaje no técnico.

Las historias de usuario constituyen las bases para las pruebas funcionales ya que se utilizan para verificar si la propuesta desarrollada cumple con lo que especifica en ellas. A continuación, se muestra un ejemplo de una de las historias de usuario generadas en la investigación, las demás se pueden encontrar en el **Anexo 1: Descripción de historias de usuario.**

HU_Iniciar sesión.

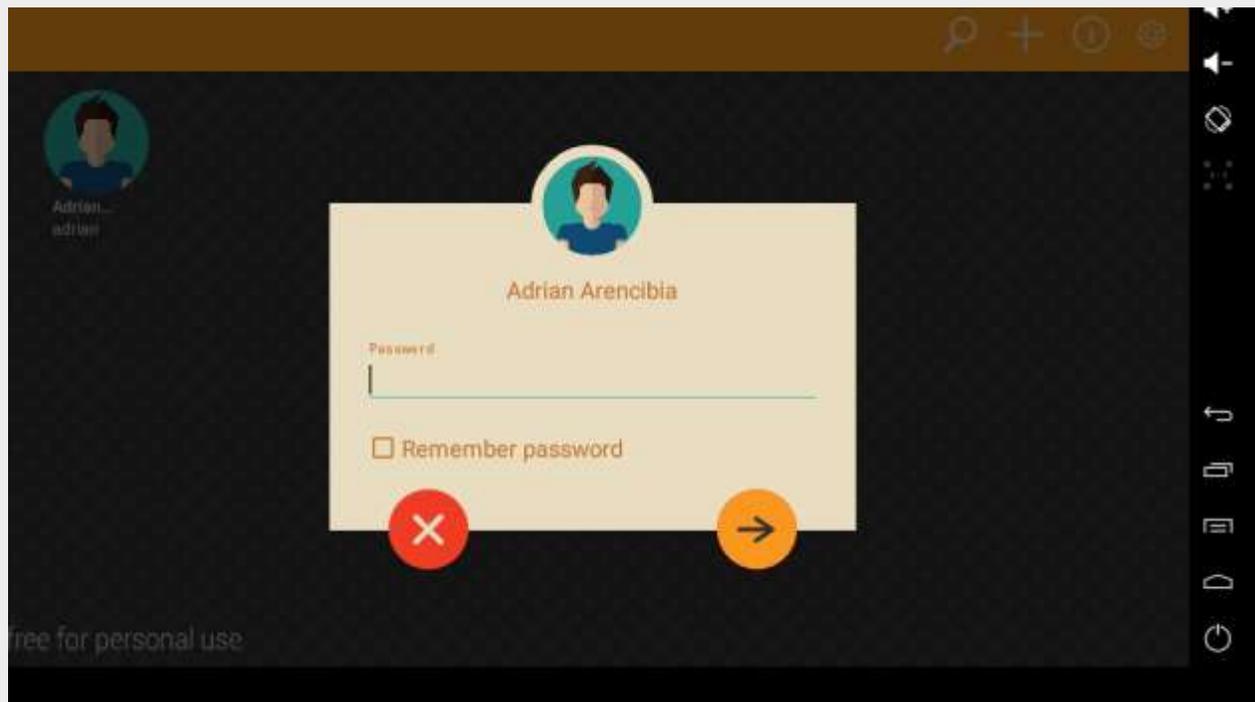
Tabla 4 Descripción de la HU Iniciar sesión

Número: N/A	Nombre del requisito: Iniciar sesión
--------------------	---

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 8 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción	
1- Objetivo: Permitir a los usuarios iniciar sesión en el sistema.	
2- Acciones para lograr el objetivo (precondiciones y datos): - El usuario debe tener una cuenta creada dentro del sistema (usuario y contraseña).	
3- Comportamientos válidos y no válidos (flujo central y alternos): Campo contraseña es obligatorios. <ul style="list-style-type: none">• 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	
4- Flujo de la acción a realizar: - Cuando el usuario selecciona el usuario del listado de usuario el sistema permite introducir el siguiente dato obligatorio: <ul style="list-style-type: none">• (*) Contraseña Además, debe permitir seleccionar una de las siguientes opciones: <ul style="list-style-type: none">• Aceptar• Cancelar Al seleccionar la opción Aceptar el sistema entra al perfil del estudiante y permite Buscar las clases en el sistema. Al seleccionar la opción Cancelar el sistema cierra la interfaz para especificar el dato y regresa al listado de usuarios. En caso que el usuario no introduzca el dato o introduzca un valor incorrecto, el sistema muestra un mensaje de información especificando el error.	
Observaciones: El sistema debe visualizar la contraseña insertada en forma encriptada, esta acción solo se realizará antes de presionar la opción Aceptar.	

Prototipo de interfaz:



2.7. Modelo de Diseño

En esta etapa se modela el sistema y se encuentra la forma de que soporte todos los requisitos, incluyendo los requisitos no funcionales y cualquier otra restricción. El modelo de diseño crea un punto de partida para las actividades de implementación subsiguientes. Permite descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. Da una forma al sistema mientras que intenta preservar la estructura definida por el modelo de análisis (14).

2.8. Diagramas de clases del diseño

Contiene debido a la adaptación del modelo de diseño a la implementación, mayor cantidad de detalles que el diagrama de análisis entre los que podemos citar: clases, atributos, operaciones, subsistemas y relaciones (14) .A continuación se muestra el diagrama de clase del diseño correspondiente al Paquete Autenticación, los demás se pueden encontrar en el **Anexo 2: Diagrama de clases del diseño**.

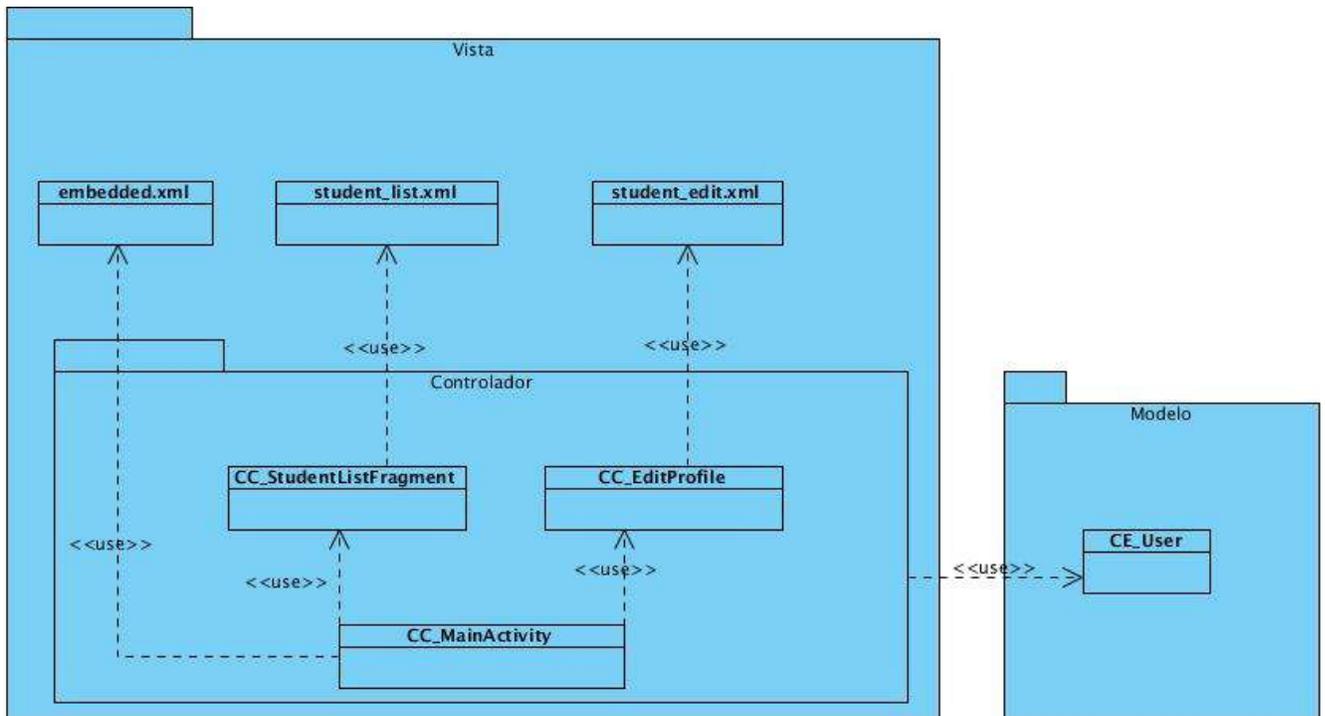


Figura 2 Diagrama de clase del diseño del Paquete Autenticación

2.9. Diagrama de secuencia del diseño

Los diagramas de secuencia muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas. El nombre del mensaje debería indicar una operación del objeto que recibe la invocación o de una interfaz que el objeto proporciona (14). A continuación se presenta el diagrama de secuencia del diseño correspondiente al requisito Iniciar sesión, los demás se pueden encontrar en el **Anexo 3: Diagramas de secuencia del diseño**.

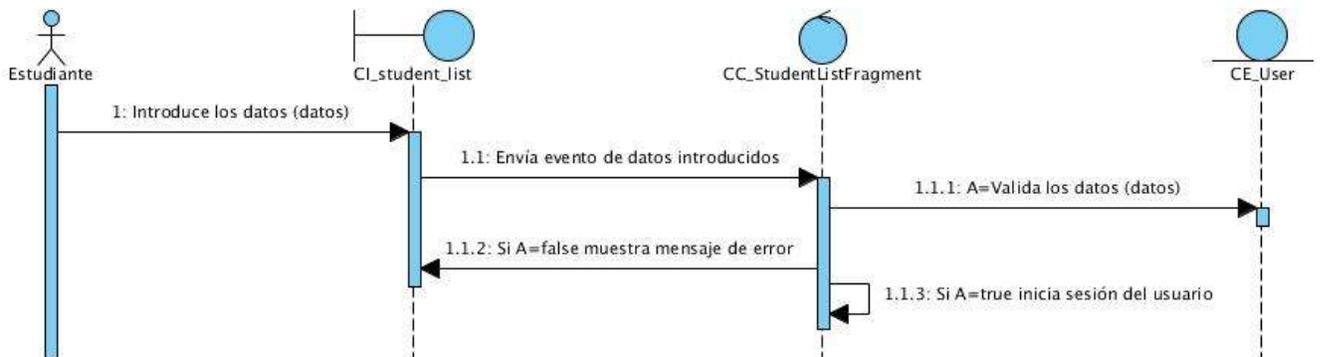


Figura 3 Diagrama de secuencia del diseño de la HU Iniciar sesión

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.10. Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema. 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 (14) .

A continuación, se presenta el Diagrama de despliegue propuesto para el sistema. En el mismo se muestra que la Tableta del estudiante se comunica con la PC del profesor mediante los protocolos:

- UDP (Protocolo de datagrama de usuario) que es un protocolo no orientado a conexión de la capa de transporte del modelo TCP/IP. (31)
- TCP (Protocolo de Control de Transmisión) que es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. Haciendo uso de este, 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 routers (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). (32)

UDP es utilizado para el proceso de encontrar las Clases activas y TCP para el resto de la comunicación entre estos dispositivos.

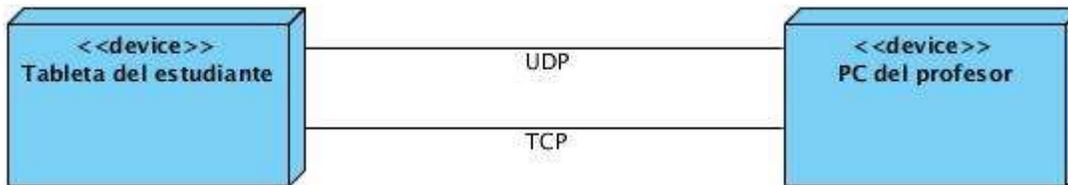


Figura 4 Diagrama de despliegue del sistema

2.11. Modelado de datos (BD)

El diagrama entidad-relación es uno de los modelos más usados para diseñar bases de datos, este modelo se encuentra basado en dos conceptos fundamentales: entidades, que representan objetos sobre los cuales se desea guardar información y las relaciones, que constituyen las relaciones entre las entidades (33).

A continuación, se presenta el modelo de datos que contiene las entidades que serán utilizadas por las funcionalidades a desarrollar y las relaciones entre ellas, las mismas representan las tablas en la base de datos.

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

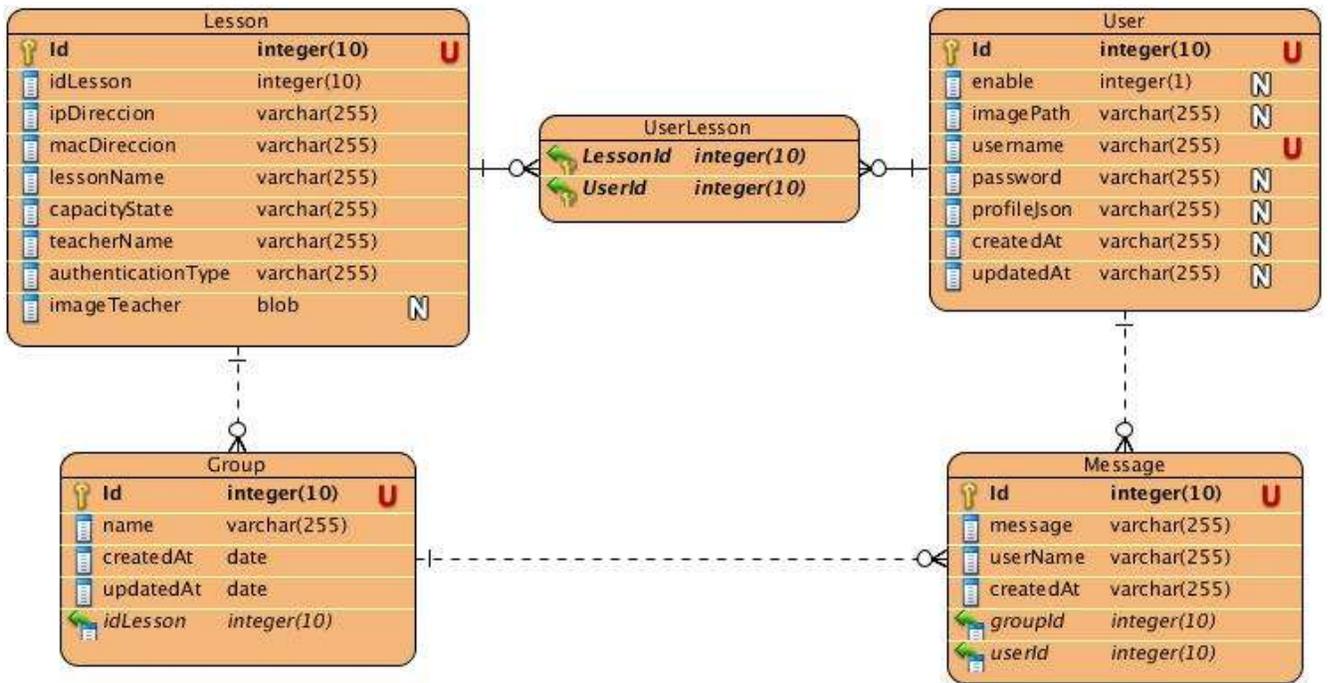


Figura 5 Modelo entidad-relación

2.12. Descripción de las tablas de la Base de Datos

A continuación, se muestra la descripción de la tabla `tb_user` creada para almacenar los datos de los estudiantes que se registran en la aplicación. Las descripciones de las restantes tablas se muestran en el **Anexo 4: Descripción de las tablas de la Base de Datos**.

Tabla 5 Descripción de los atributos la tabla `tb_user`

tb_User		
Descripción: almacena los datos pertenecientes a la entidad estudiante.		
Atributo	Tipo	Descripción
id	integer (10)	Campo que contiene el identificador del usuario
enable	integer(1)	Campo que indica si está activo o no el usuario
imagePath	varchar(255)	Campo que almacena la ruta de la imagen
username	varchar(255)	Campo que almacena el nombre de usuario

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

password	varchar(255)	Campo que almacena la contraseña del usuario
profileJson	varchar(255)	Campo que almacena información adicional del usuario
createAt	date	Campo que contiene la fecha de creación del usuario
updateAt	date	Campo que contiene la fecha de actualización del usuario
savePasssword	integer(1)	Campo que indica, si se salvó o no la contraseña

2.13. Patrón arquitectónico

Un patrón arquitectónico se representa como una estructura de datos y componentes del programa que se requieren para construir un sistema computacional. Para su implementación se considera el estilo de arquitectura que adoptará el sistema, la estructura y las propiedades de los componentes que lo constituyen y las interrelaciones que ocurren entre sus componentes arquitectónicos. Resumiendo, según Pressman “la arquitectura es la estructura de organización de los componentes de un programa (módulos), la forma en la que éstos interactúan y la estructura de datos que utilizan” (34)

La arquitectura del *software* por otra parte, es la estructura o las estructuras del sistema, lo que comprende a los componentes del *software*, sus propiedades externas visibles y las relaciones entre ellos. En la actualidad existe una amplia gama de arquitecturas, pero para el desarrollo de la propuesta de solución se utilizó la arquitectura Modelo-Vista-Controlador (MVC). Esta arquitectura cuya principal bondad consiste en separar los datos de una aplicación (Modelo), la interfaz de usuario (Vista) y la lógica de negocios (Controlador) en tres componentes distintos que se relacionarán para al final tener como resultado la aplicación.

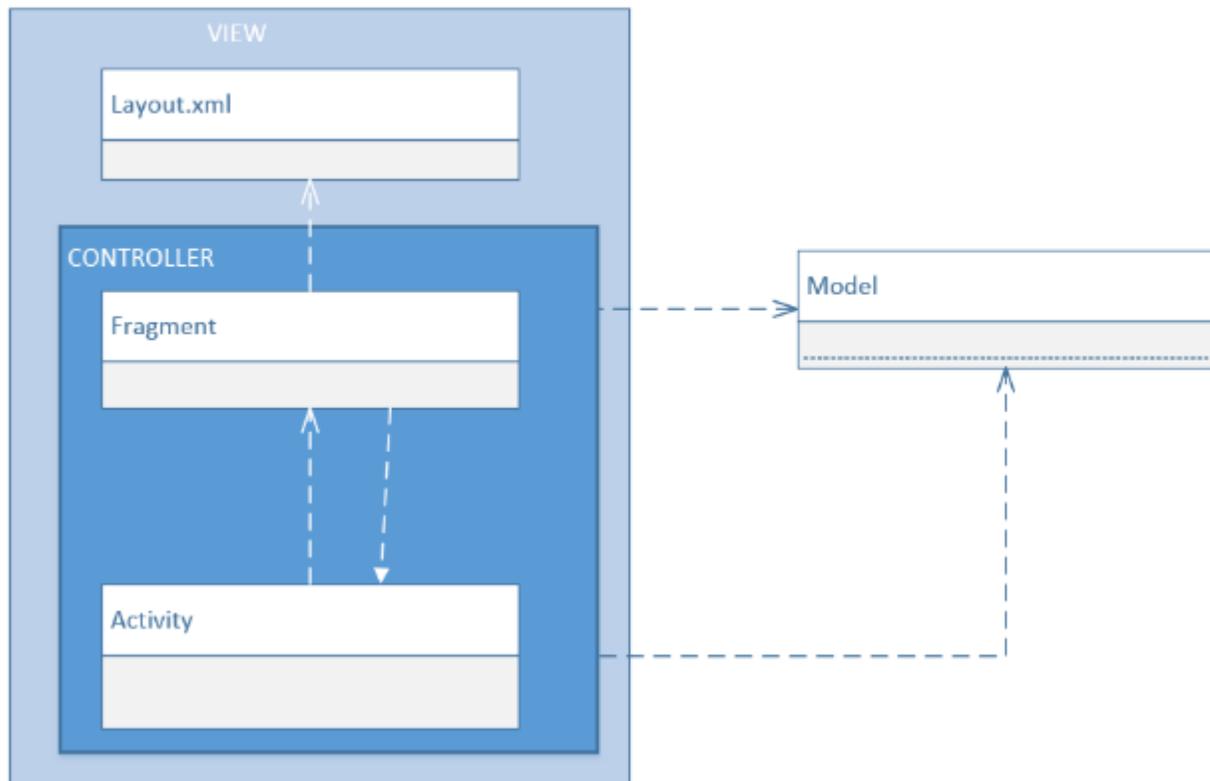


Figura 6 Android MVC implementación

A continuación, se explican los componentes:

- **Modelo:** Se refiere a las representaciones que se construyen basadas en la información con la que operará en la aplicación. Un ejemplo dentro de la propuesta de solución es la clase User.
- **Vista:** No es más que la interfaz con la que va a interactuar el usuario. En Android, las interfaces se construyen en XML y un ejemplo dentro de la propuesta de solución es la vista student_login.xml.
- **Controlador:** Es donde se implementa la lógica de las distintas funcionalidades y a grandes rasgos se puede decir que funciona de intermediario entre la vista y el modelo. Un ejemplo dentro de la propuesta de solución es la clase MainActivity.

2.14. Patrones de diseño

Un patrón de diseño es una pareja de problema o solución con un nombre y es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. Los patrones no se

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

proponen descubrir ni expresar nuevos principios de la ingeniería del software. Al contrario, estos intentan codificar el conocimiento, las expresiones y los principios ya existentes (28).

2.14.1. Patrones GRASP

Lo esencial de un diseño de objeto lo constituye el diseño de las interacciones de objetos y la asignación de responsabilidades. Las decisiones que se tomen pueden influir profundamente en la extensibilidad, claridad y mantenimiento del sistema de software de objetos, además en el grado y calidad de los componentes reutilizables, por esta razón, durante el diseño se deben realizar los casos de usos con objetos basados en los patrones GRASP (35).

A continuación, se describen los patrones GRASP utilizados durante el desarrollo:

Controlador: Proporciona guías acerca de las opciones generalmente aceptadas y adecuadas para manejar eventos. Es conveniente utilizar la misma clase controlador para todos los eventos del sistema de un requisito, de manera que es posible manejar la información acerca del estado del caso de uso en el controlador. Se utilizó en clases controladoras como *ChatController*.

Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Se utilizó este patrón en clases como *MainApp* para la creación de los objetos.

Experto: Se utiliza con frecuencia en la asignación de responsabilidades, es un principio de guía básico que se utiliza continuamente en el diseño de objetos. Expresa la intuición común de que los objetos hacen las cosas relacionadas con la información que tienen. Se utilizó en clases como *ChatFragment*.

2.14.2. Patrones Gof

Describen 23 patrones de diseño comúnmente utilizados y de gran aplicabilidad en problemas de diseño usando modelado UML. Se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento (35).

Los patrones creacionales abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados o inicializados. (Abstract Factory, Factory Method, Prototype y Singleton).

Los patrones estructurales se ocupan de cómo las clases y objetos se combinan para formar grandes estructuras y proporcionar nuevas funcionalidades. (Adapter, Decorator, Fachada, y Proxy).

CAPÍTULO 2. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Los patrones de comportamiento están relacionados con los algoritmos y la asignación de responsabilidades entre los objetos. Son utilizados para organizar, manejar y combinar comportamientos. (Chain of Responsibility, Interpreter, Observer, Template Method).

A continuación, se describen los patrones Gof más significativos utilizados durante el desarrollo:

Dependency injection: inyección de dependencia consiste en suministrar objetos a una clase en lugar de ser la propia clase quien cree el objeto. Es utilizado en varias partes del diseño realizado como es el caso de la clase MainApp, en esta se maneja un objeto de tipo ServerObject utilizado para definir el servidor actual, este objeto es inyectado a esta clase, similar ocurre en otras clases implementadas.

Singleton: está diseñado para restringir la creación de objetos pertenecientes a una clase, forzando a que solo se pueda crear una única instancia, de ahí su nombre singleton (instancia única). En la implementación realizada se utiliza en la clase MainApp.

2.15. Conclusiones parciales

En el presente capítulo se llevó a cabo el análisis y diseño del sistema a partir de la metodología de trabajo AUP en su variante UCI arrojando los siguientes resultados:

- La realización del modelo de dominio permitió obtener una visión detallada de las relaciones entre los procesos del sistema.
- La disciplina de requisitos posibilitó la identificación y descripción de todos los requisitos funcionales y no funcionales a incorporar en la solución. Se identificaron 22 requisitos funcionales y 3 no funcionales.
- La disciplina de diseño permitió la confección de los artefactos de correspondientes donde se obtiene: la arquitectura del sistema utilizando el patrón MVC para aplicaciones Android; el diseño de los requisitos del sistema; los diagramas de clases del diseño muestran las clases y sus relaciones para llevar a cabo el desarrollo de la solución, los diagramas de secuencia que evidencia el flujo de cada requisito y el diagrama de despliegue que muestra el entorno físico en que muestra en el sistema.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

3.1. Introducción

En este capítulo se documenta el proceso de implementación de los elementos identificados durante la elaboración de capítulo anterior. Para dar cumplimiento a este propósito, se modela el diagrama de componentes. Además, se incluyen los resultados de las pruebas y las validaciones realizadas a la aplicación desarrollada lo cual permitió que los componentes desarrollados cumplan con los requisitos establecidos.

3.2. Modelo de implementación

El modelo de implementación describe cómo los elementos del diseño se implementan en componentes. También describe la organización de los componentes según los mecanismos de estructuración y modularización disponibles en el entorno de desarrollo, el lenguaje de programación utilizado, y la dependencia entre componentes. Como parte del correcto desarrollo del modelo de implementación se obtiene el diagrama de componente que a continuación se presentan.

3.3. Diagrama de componentes

“Los diagramas de componentes describen la descomposición física de los elementos de un sistema (modulo, base de datos, programa ejecutable, etc.) y sus relaciones. Muestran las opciones de realización incluyendo código fuente y ejecutable, los cuales pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros.” (36)

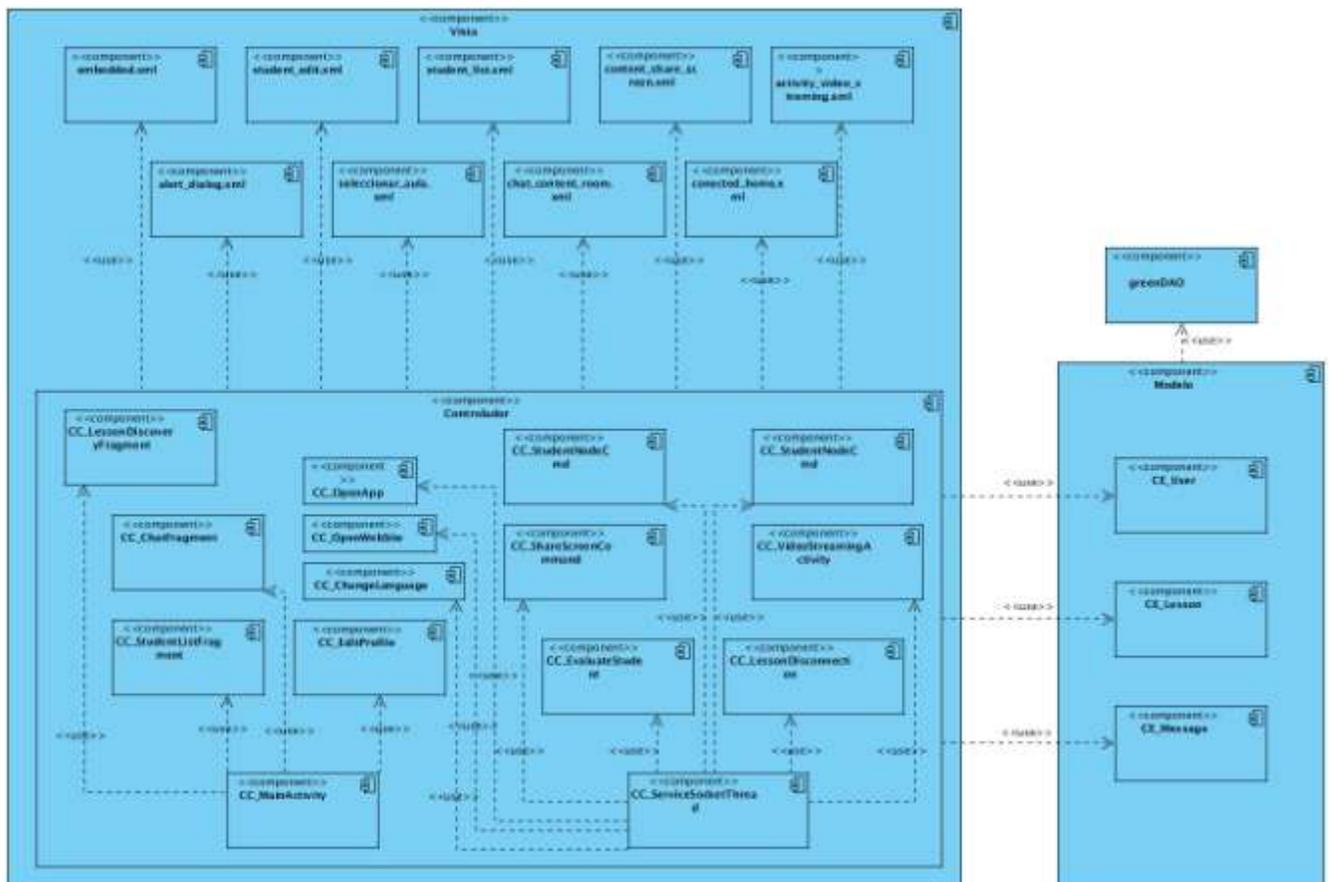


Figura 7 Diagrama de componentes del sistema

3.4. Pruebas de software

“La fase de pruebas del sistema tiene como objetivo verificar el sistema software para comprobar si este cumple sus requisitos. Dentro de esta fase pueden desarrollarse varios tipos distintos de pruebas en función de los objetivos de las mismas. Algunos tipos son pruebas funcionales, pruebas de usabilidad, pruebas de rendimiento, pruebas de seguridad, etc.” (36)

Las pruebas realizadas a todo producto de software constituyen un elemento vital para validar el correcto funcionamiento del mismo. Permiten a los desarrolladores definir las pautas para identificar, documentar y corregir un conjunto de no conformidades antes de que el producto sea liberado. Garantizándose que el producto final funcione de acuerdo a los requerimientos por los cuales fue diseñado e implementado, de manera correcta.

3.5. Niveles de prueba

Los niveles de prueba son diferentes formas de verificar y validar un producto de software, estos pueden ser de diferentes tipos y se utilizan indistintamente según el software desarrollado. A continuación, se distinguen los siguientes niveles de prueba empleados para la identificación de los errores en el sistema propuesto.

Prueba de sistema: Son las pruebas que se realizan cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. De manera general, este nivel de prueba es preparado y ejecutado por un grupo independiente al desarrollador, y consiste en validar que el software cumpla con los requerimientos especificados por el cliente. (36)

Prueba de aceptación: son realizadas por el usuario final en lugar del responsable del desarrollo del sistema. Una prueba de aceptación puede ir desde un informal paso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. Además, se llevan a cabo con el objetivo de que el cliente valide todos los requisitos que debe poseer el sistema. (37)

3.6. Tipos de prueba

Pruebas funcionales: son aquellas que tienen por objetivo probar y validar que el sistema cumpla con todos los requisitos especificados.

Pruebas de usabilidad: consisten en documentar la interacción de los usuarios con el sistema, a la vez que estos llevan a cabo las tareas para las cuales fue diseñado el software.

3.7. Métodos de prueba

Las técnicas de prueba proporcionan distintos criterios para generar casos de prueba que provoquen fallos en los programas. Estas técnicas se agrupan en: (38).

- **Pruebas de caja blanca o estructural**, también conocidas como pruebas estructurales o pruebas de caja transparente se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Su correcto uso da la posibilidad de obtener casos de prueba que garantizan, que se examinen de manera detallada los procedimientos y caminos lógicos del sistema para llegar a la conclusión de si el estado real del mismo coincide con el esperado.
- **Pruebas de caja negra o funcionales**, también conocidas como pruebas de caja opaca, funcionales, de entrada/salida o inducidas por los datos están especialmente indicadas a

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

centrarse en los requisitos funcionales del software. Se llevan a cabo sobre la interfaz del software buscando errores en cada una de las funcionalidades. Con la aplicación de estas se trata de demostrar que las funciones del software son completamente operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado.

Para validar la propuesta de solución se empleará el método de caja negra con la técnica de partición por equivalencia con el objetivo comprobar la validez en las respuestas de las funcionalidades ante las acciones del usuario y la calidad de las salidas en dependencia de las entradas.

3.8. Diseño de casos de prueba

El diseño de casos de prueba consiste en probar el sistema, incluyendo los datos de entrada y los resultados esperados. Éstos se derivan de las historias de usuario y su objetivo fundamental es encontrar la mayor cantidad de defectos en las funcionalidades implementadas y mostrar que el sistema satisface las necesidades del cliente.

A continuación, se presenta el diseño del caso de prueba perteneciente al requisito *Iniciar sesión*, el resto de los artefactos de este tipo se encuentran en el **Anexo 5: Diseño de casos de prueba**.

Descripción general

Permitir a los usuarios iniciar sesión en el sistema.

Condiciones de ejecución

Para iniciar sesión en el sistema hay que:

- Tener en cuenta los siguientes datos: usuario y contraseña.
- Para el SC Iniciar sesión (estudiantes): Debe haberse creado un perfil con anterioridad.

SC Iniciar Sesión (estudiantes)

Tabla 6 Diseño de caso de prueba de la HU SC Iniciar Sesión

Escenario	Descripción	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Seleccionar perfil	Para acceder al sistema el usuario selecciona su perfil del listado	N/A	Una vez seleccionado el perfil el sistema muestra un	Listado de perfiles/Seleccionar perfil

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

	de perfiles.		formulario donde le permite introducir la contraseña para acceder en el sistema. Permite además habilitar la opción de Recordar contraseña y seleccionar una de las siguientes opciones: Aceptar y Cancelar	
EC 1.2 Opción Aceptar	Una vez especificados los datos el usuario selecciona la opción Aceptar representada por el ícono de la flecha.	V	El sistema permite que se acceda y le permite al usuario interactuar con el mismo.	Listado de perfiles/Seleccionar perfil/Especificar dato/Aceptar
EC 1.3 Opción Cancelar	El usuario selecciona la opción Cancelar representada por el ícono de la X.	N/A	El sistema muestra la lista de perfiles.	Listado de perfiles/Seleccionar perfil/Especificar dato/Cancelar
EC 1.4 Habilitar Recordar contraseña	Una vez especificado el dato el usuario habilita la opción Recordar contraseña.	V	El sistema guardará la contraseña del usuario y permitirá al usuario acceder	Listado de perfiles/Seleccionar perfil/Especificar dato/Habilitar Recordar contraseña/Aceptar

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

			al sistema sin necesidad de especificar dicho dato.	
EC 1.5 Datos incorrectos	Al insertar el dato para iniciar sesión este es incorrecto.	I	El sistema debe notificar al usuario a través de un mensaje de información. Además permitirá realizar nuevamente la acción.	Listado de perfiles/Seleccionar perfil/Especificar dato/Aceptar
		V		
EC 1.6 Datos vacíos	El usuario deja el campo vacío.	I	El sistema debe notificar al usuario a través de un mensaje de información. Además permitirá realizar nuevamente la acción.	Listado de perfiles/Seleccionar perfil/Especificar dato/Aceptar
		V		

3.9. Resultados obtenidos

3.9.1. Resultados de la prueba de Sistema

En las principales pruebas pertenecientes al nivel de sistema que se realizaron se empleó el método de caja negra sobre las interfaces gráficas del sistema. La aplicación de este método tiene como objetivo que el producto final cuente con el menor número de errores posibles. Es decir, se centra en el cumplimiento de las funcionalidades del componente que se obtiene. Las pruebas fueron realizadas por el equipo de calidad UCI haciendo uso de los diseños de casos de prueba presentados anteriormente y ejecutándose 6 iteraciones.

Las pruebas realizadas arrojaron los siguientes resultados:

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

Tabla 7. Resultados obtenidos de las pruebas por iteración.

Iteraciones	Cantidad de defectos	% Resolución	Duración de la iteración (días)
1ra iteración	37 defectos	97 % defectos resueltos – 01 No proceden(NP)	30/01/2017 al 8/02/2017 8 días
2da iteración	11 defectos	81 % defectos resueltos – 02 No proceden(NP)	10/02/2017 al 16/02/2016 5 días
3ra iteración	14 defectos	100 % defectos resueltos	16/02/2017 al 24/02/2017 4 días
4ta iteración	7 defectos	85 % defectos resueltos – 01 No proceden(NP)	24/02/2017 al 1/03/2017 3 días
5ta iteración	12 defectos	91 % defectos resueltos – 01 No proceden(NP)	1/03/2017 al 14/03/2017 10 días
6ta iteración y prueba final	6 defectos	100 % defectos resueltos	2 días

En la tabla anterior se muestra la ejecución de seis iteraciones de pruebas realizadas al sistema. Donde en cada una de estas se solucionan las no conformidades identificadas, exceptuando aquellas que no proceden. Para así obtener un producto de calidad libre de errores y satisfacer las necesidades del cliente.

A continuación, se muestra un gráfico donde se puntualiza por iteraciones el total de no conformidades detectadas, el total de aquellas que fueron resueltas y la cantidad que no proceden.

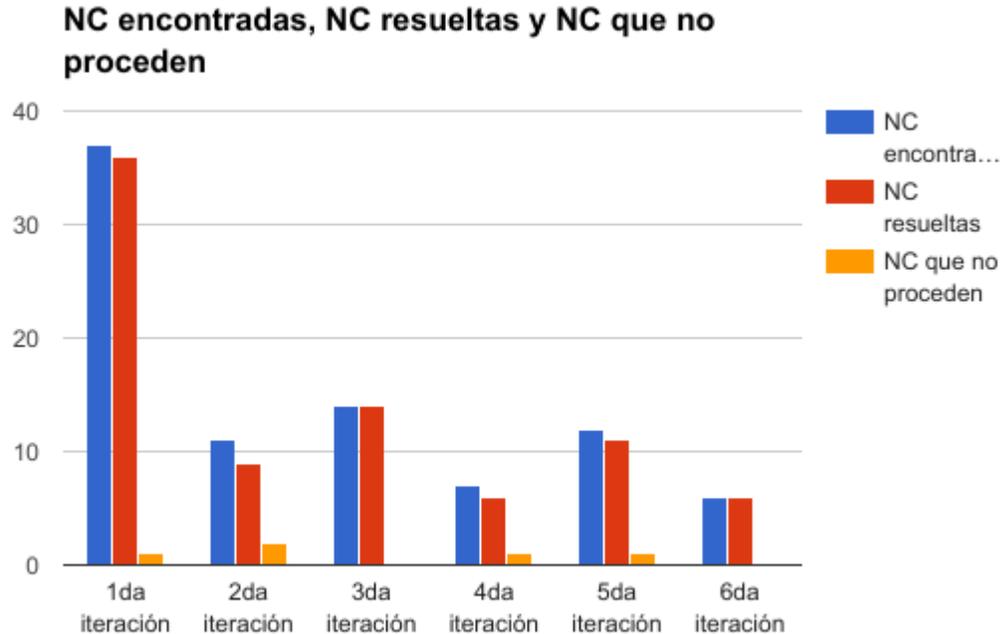


Figura 8 Grafico con información de los resultados de las pruebas

3.9.2. Resultados de la prueba de aceptación

Las pruebas de aceptación fueron realizadas por un grupo de 10 miembros de la Empresa Industrial para la Informática, la Electrónica y las Comunicaciones (GEDEME) obteniéndose como resultado gran satisfacción por parte del cliente, como evidencia de cumplimiento de las mismas se firma un acta de aceptación del producto la cual se muestra en el **Anexo 6: Acta de aceptación**.

3.10. Conclusiones del capítulo

En este capítulo se llevó a cabo las actividades de implementación y prueba atendiendo a la metodología definida en el primer capítulo. Las mismas arrojaron los siguientes resultados:

- Se obtuvo el diagrama de componentes, con el cual se logró definir la estructura general de la aplicación y el comportamiento de los componentes.
- La realización de las pruebas permitió confirmar la correcta implementación de los requisitos definidos.

CONCLUSIONES GENERALES

CONCLUSIONES GENERALES

Al inicio de esta investigación se trazaron varios objetivos y tareas que tributaban a un objetivo general, los cuales permitieron arribar a las siguientes conclusiones:

- Se determinaron las principales funcionalidades que una aplicación destinada al estudiante debe tener y algunas que, aunque no son obligatorias, enriquecen el proceso de enseñanza aprendizaje gracias al análisis de las soluciones similares.
- Se desarrolló una aplicación Android que cumple con los requerimientos del cliente apoyada en las características de los sistemas similares estudiados. Además, se garantiza que la solución desarrollada contribuye con la política de soberanía tecnológica por la que aboga el país.
- La solución cuenta con la calidad requerida, ya que fue sometida a pruebas realizadas al sistema mediante la técnica de caja negra usando partición por equivalencia. Lo que permitió validar del correcto funcionamiento del sistema propuesto, detectándose un total de 87 no conformidades en seis iteraciones. Además, se aplicó el nivel de prueba de aceptación comprobando la completa satisfacción del cliente con el producto.

RECOMENDACIONES

RECOMENDACIONES

- Continuar enriqueciendo el software ATcnea añadiéndole funcionalidades como Transmisión por Cámara, Reporte de Promedio Escolar, Noticias/Anuncios de la Escuela y Moderación Comentarios en el Chat.
- Optimizar y refactorizar la implementación de las funcionalidades existentes en el sistema.
- Añadirle internacionalización para otros importantes idiomas como portugués, ruso y chino.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

1. **Monge Bartolomé, Álvaro.** *Dispositivos móviles en la educación.* 2013.
2. **Herrera, Susana I. y Fennema, Marta C.** *Tecnologías Móviles Aplicadas a la Educación Superior.* Argentina : CACIC, 2011. 630.
3. *Aulas inteligentes/Smart classrooms.* **Vidal Ledo, María Josefina , Morales Suárez, Ileana y Rodríguez Dopico, Rosa Moraima.** 2, Cuba : Educ Med Super, 2014, Vol. 28. ISSN.
4. *El modelo construtivista con las nuevas tecnologías: aplicado en el proceso de aprendizaje.* **Hernández Requena, Stefany.** 2, RUSC. Universities and Knowledge Society Journal : s.n., 2008, Vol. 5. ISSN.
5. **Segovia, F.** *El aula inteligente: Nuevo horizonte educativo.* Madrid : s.n., 1998.
6. **Blesa, José Antonio.** Aulas autosuficientes. *EducaLAB.* [En línea] INTEF, Agosto de 2005. [Citado el: 10 de April de 2017.] <http://roble.pntic.mec.es/~jblesa/autosufi.htm>.
7. *Aulas inteligentes.* **Vidal Ledo, María Josefina , Morales Suárez, Ileana y Rodríguez Dopico, Rosa Moraima.** 2, Ciudad de la Habana : Educación Media Superior, 2014, Vol. 28. ISSN.
8. *MODELO DE INTEGRACIÓN EDUCOMUNICATIVA DE 'APPS' MÓVILES PARA LA ENSEÑANZA Y APRENDIZAJE.* **Villalonga Gómez, Cristina y Marta-Laz, Carmen.** 46, Madrid : s.n., 2015. ISSN.
9. **Esquivel González, Diana Rosa.** Aulas más «inteligentes». *Juventud Rebelde.* Edición Digital, 2017.
10. **Var , Antonio.** Prezi. [En línea] 05 de 09 de 2014. [Citado el: 28 de 02 de 2017.] <https://prezi.com/f5rnyz3ffbqr/uso-de-mythware/>.
11. **Doerffel, Tobias.** iTALC - Intelligent Teaching And Learning with Computers. [En línea] 2004. [Citado el: 28 de 02 de 2017.] <http://italc.sourceforge.net/>.
12. **ABTutor.** ab tutor. [En línea] 2017. [Citado el: 28 de 02 de 2017.] <http://www.abtutor.com/kb>.
13. **Chacón, Julio César Rueda.** *Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar J2EE.* Guatemala : s.n., 2006. s.n.
14. **Jacobson, IvarBooch, et al.** El proceso unificado de desarrollo de software/The unified software development process. *El proceso unificado de desarrollo de software/The unified software development process.* s.l. : Pearson Educación, 2000.
15. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *Proceso Unificado de Desarrollo de Software* . s.l. : Addison Wesley . s.l.
16. *Propuesta de metodología de desarrollo de software para objetos virtuales de aprendizaje.* **Castrillón, Eucario Parra.** 34, Colombia : s.n., 2011.

REFERENCIAS BIBLIOGRÁFICAS

17. **José H. Canós, Patricio Letelier M Carmen Penadés.** *Metodologías Agiles en el Desarrollo de Software*. Valencia : s.n. s.n.
18. *AUP The Agile Unified Process*. [En línea] [Citado el: 6 de diciembre de 2016.] <http://www.cc.una.ac.cr/AUP/html/philosophies.html>.
19. Metodología de desarrollo para la Actividad productiva de la UCI. [aut. libro] Programa de Mejora. Cuba : s.n., 2017.
20. **Tomás Gironés, J.** *El gran libro e Android*. 3ra edition. Barcelona, España : Alfaomega Grupo Editor, 2013.
21. **Tomás Gironés, Jesús.** *El gran libro de Android*. 2011.
22. Academia Android. *Academia Android*. [En línea] 8 de enero de 2015. [Citado el: 10 de marzo de 2017.] <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>.
23. **Targetware Informática S.A.C.** . *software.com.ar. software.com.ar*. [En línea] 2007-2017. [Citado el: 21 de enero de 2017.] <http://www.software.com.ar/p/visual-paradigm-para-uml>.
24. **MaterialDrawer**. MaterialDrawer cajón de navegación para aplicaciones Android. [En línea] MaterialDrawer, 2016. <https://github.com/mikepenz/MaterialDrawer>.
25. **FFmpegMediaPlayer**. FFmpegMediaPlayer proporciona una interfaz unificada para reproducir archivos de audio y secuencias. [En línea] 2016. <http://wseemann.github.io/FFmpegMediaPlayer/>.
26. **Otto**. Un bus de eventos mejorado con énfasis en la compatibilidad con Android. [En línea] 2016. <http://square.github.io/otto/>.
27. **GreenDAO**. GreenDAO: Android ORM for your SQLite database. [En línea] 2016. <http://greenrobot.org/greendao/>.
28. **Larman, Craig**. Modelo de dominio. *UML y Patrones*. s.l. : Prentice Hall, 2003.
29. **SOMMERVILLE, Ian**. Ingeniería del software. 7ma. Madrid : Pearson Educación, 2005.
30. —. *Ingeniería de software*. 8va. s.l. : ISBM, 2007.
31. **UDP, Protocolo**. CCM. [En línea] 2008. <http://es.ccm.net/contents/284-protocolo-udp>.
32. **TCP, Protocolo**. CCM. [En línea] 2008. <http://es.ccm.net/contents/281-protocolo-tcp>.
33. **Kroenke, David M.** *Procesamiento de bases de datos: fundamentos, diseño e implementación*. . s.l. : Pearson Educación, 2003.
34. **Pressman, Roges S.** *Ingeniería de Software. Un enfoque práctico, sexta edición*. 2011.
35. *Una ontología para la representación de conceptos de diseño de software*. **Gómez Giraldo, Gloria Lucia, ACEVEDO, Juan F y MORENO, David A** 3, s.l. : 09, 2011, Avances en Sistemas e Informática, Vol. 8. ISSN.

REFERENCIAS BIBLIOGRÁFICAS

36. **Tapia Hernández, Areli.** *MODELO DE IMPLEMENTACION.* 2014.
37. **Ruiz Tenorio, Roberto .** *Las Pruebas de Software y su Importancia en las Organizaciones.* Veracruz : s.n., 2010.
38. **Guayta , Caisa, Semblantes Chicaiza, Carmen Jimena y Lilia Vanessa.** *Implementación de Pruebas Caja Negra y Caja Blanca aplicables al Sistema Escolástico del Colegio Nacional "Primero de Abril".* Latacunga : s.n., 2011.
39. **Espinosa, Sam.** *Mapa de procesos.* s.l. : Bussines & Mgmt., 2009.
40. **Xavier Ferré Grau, María Isabel Sánchez Segura.** *Desarrollo Orientado a objetos con UML.* 2012.
41. *Desarrollo de aplicaciones sobre Android.* **Vanegas, Carlos Alberto.** 2, 2012, Vol. 9.
42. *Tendencias actuales en el uso de dispositivos móviles en educación.* **Valero, Carmen Cantillo y Redondo, Margarita Roura.** 147, 2012.
43. **Valenzuela Zambrano, B., y Pérez Villalobos.** unisabana.edu.co. *unisabana.edu.co.* [En línea] 15 de febrero de 2013. [Citado el: 6 de diciembre de 2016.] <http://educacionyeducadores.unisabana.edu.co/index.php/eye/article/view/2000/3074>.
44. **Tutorialspoint.** Angularjs- MVC Architecture. [En línea] 2016. [Citado el: 27 de febrero de 2017.] http://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm.
45. —. [En línea] 2016. [Citado el: 27 de febrero de 2017.]
46. *Dispositivos Móviles y Multimedia.* **Tardáguila, César.** 2009.
47. *VENTAJAS DEL USO DE LAS TICs EN EL PROCESO DE ENSEÑANZAAPRENDIZAJE.* **Soto, Carlos Ferro.** 29, España : EDUTEC, 2009.
48. **Sommerville, Ian.** Ingeniería de software. Séptima edición, Ian Sommerville. *Ingeniería de software.* 2009.
49. **Sarmiento, Johana.** blogspot. *blogspot.* [En línea] 12 de abril de 2013. [Citado el: 10 de marzo de 2017.] <http://umlDiagramadespliegue.blogspot.com/>.
50. **Sánchez, Tamara Rodríguez.** *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.
51. Ingeniería de software. Un enfoque práctico. [aut. libro] Roger S. Pressman. *Ingeniería de software. Un enfoque práctico.*
52. **Portillo, M. del Pilar del Saz.** *Tutorial Patrón MVC.* 2014.
53. **Polo, Juan Diego.** WWwhatsnew.com. *WWwhatsnew.com.* [En línea] 12 de febrero de 2011. [Citado el: 6 de diciembre de 2016.] <http://wwwwhatsnew.com/2011/02/12/7-plataformas-gratuitas-para-crear-cursos-de-aprendizaje-online/>.

REFERENCIAS BIBLIOGRÁFICAS

54. **Peña, Jorge Polanco.** *DESPLIEGUE Y GESTIÓN MULTITENANT DE UNA APLICACIÓN WEB DE CONTROL DE SOLICITUDES DE ACTIVIDADES.* 2016.
55. **Padrón, Sandy Nuñez.** *FICHA XAUCE ZERA 2.0.* La Habana : s.n., 2016.
56. **MSc. María J. Vidal Ledo, MSc. Xaily Gavidondo Mariño,* Lic. Alfredo Rodríguez Díaz,* Lic. Armando Cuéllar Rojas*.** Scielo Educ Med Super . *Scielo Educ Med Super* . [En línea] julio de 2015. [Citado el: 24 de noviembre de 2016.] http://scielo.sld.cu/scielo.php?pid=S0864-21412015000300024&script=sci_arttext&tlng=pt. 3.
57. **Montaño, Enrique Castillo.** *CLIENTE MODDLE PARA ANDROID.* Cádiz : s.n., 2014.
58. **Manuel Báez, Álvaro Borrego, Jorge Cordero,.** *Introducción a Android.* s.l. : E.M.E. Editorial ©, 2011.
59. **Luis Hernández, Pepe Domínguez.** *uptodown.com. uptodown.com.* [En línea] 2013. [Citado el: 6 de diciembre de 2016.] <http://www.uptodown.com/windows/buscar/plataforma-dokeos>.
60. **Lisandro Delía, Nicolás Galdamez, Pablo Thomas, Patricia Pesado.** *Un Análisis Experimental de Tipo de Aplicaciones para Dispositivos Móviles.* Universidad Nacional de La Plata. Argentina : s.n., 2012.
61. **Kremer, Harold.** Mundo Kramer's Blog. *MODELO DE ANÁLISIS. Mundo Kramer's Blog. MODELO DE ANÁLISIS.* [En línea] 23 de mayo de 2011. [Citado el: 9 de marzo de 2017.] <https://mundokramer.wordpress.com/2011/05/20/modelo-de-analisis-software/>.
62. *Plataformas tecnológicas para el entorno educativo.* **José, Sánchez Rodríguez.** 14, 2005, Vols. pp. 18-24.
63. **IZAQUITA, MARIA EUGENIA ROJAS.** *AGILIZANDO LO ÁGIL.* Bogota : s.n., 2011.
64. **Guerrero, Moreno.** 2011.
65. **Gloria L Giraldo, Juan F Acevedo.** *Una Ontología para la presentación de conceptos de diseño de software.* 2011.
66. **Gallego, Manuel Trigas.** *GESTION DE PROYECTOS INFORMÁTICOS.* 2012.
67. **Fernández, Oscar Belmonte.** *Introducción al lenguaje de programación Java. Una guía básica.* 2011.
68. **Edgardo Ruiz Carrillo, Luis Benjamín Estervel Rivera.** *La relación maestro-alumno en el contexto del aprendizaje.* México : s.n., 2014.
69. **Dianelys Pérez González, Sandy Nuñez Padrón.** *SCORMVIEW: VISOR DE PAQUETES SCORM PARA XAUCEMOVIL.* La Habana : s.n., 2015.
70. **Desarrolladores, Colectivo de.** *exelearning.net. exelearning.net.* [En línea] 2016. exelearning.net.

REFERENCIAS BIBLIOGRÁFICAS

71. *VENTAJAS E INCONVENIENTES DE LAS TICS EN EL AULA*. **Cobos, Eva María Rodríguez**. ,Nº 9, Centro Educativo de Cádiz, España : s.n., 2011, Vol. Vol 1.
72. **Castellanos, José González**. *DESARROLLO DE UNA APLICACIÓN PARA MÓVILES PARA LA PLATAFORMA EDUCATIVA ZERA*. La habana : s.n., 2013.
73. —. *Desarrollo de una aplicación para móviles para la plataforma educactiva ZERA*. 2013.
74. **Cabrera, Daimel Noda**. *Informe tecnico de Proyecto de Investigación y desarrollo*. La Habana : s.n., 2017.
75. **Balaguera, Yohn Daniel Amaya**. *Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles.Estado actual*. 2013.
76. **B, Mónica María Agudelo**. *Programa de Integración de Tecnologías a la Docencia*. 2011.
77. **autores, Colectivo de**. *Preparación Pedagógica Integral para profesores universitarios*. s.l. : Félix Valera , 2003.
78. **Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez**. *Dispositivos móviles*. 2011.
79. **Arlety González Madera, Reysel Pérez Aviléz**. *Componente para la integracion del Juez en Línea Caribeño y la plataforma educativa ZERA 2.0*. La Habana : s.n., 2016.
80. **Area, Manuel**. *E-Learning: enseñar y aprender en espacios*. 2016.
81. **Ana, Hugo Andrés y Gader, Ioana Noel**. *Desarrollo de Aplicaciones para dispositivos Móviles sobre la plataforma Android de Google*. 2011.
82. **uptodown**. *uptodown.com*. [En línea] 14 de junio de 2014. [Citado el: 12 de enero de 2017.] <https://android-studio.uptodown.com/windows>.
83. **Sparx Systems**. *Sparx Systems Pty Ltd. Sparx Systems Pty Ltd*. [En línea] 17 de noviembre de 2016. [Citado el: 23 de marzo de 2017.] <http://www.sparxsystems.com.ar/index.html>.
84. **Educación online**. *Habilidades Docentes* . [En línea] 2016. <http://hadoc.azc.uam.mx/objetivos/objetivodeaprendizaje.htm>.
85. **exelearning**. *exelearning*. [En línea] 2015. <http://exelearning.net>.
86. **Telefonía celular**. *ecured.cu. ecured.cu web site*. [En línea] 15 de septiembre de 2015. [Citado el: 20 de noviembre de 2016.] https://www.ecured.cu/Telefon%C3%ADa_celular#Influencia_en_la_sociedad.. 6.
87. **Sylvain HÉBUTERNE - Sébastien PÉROCHON**. *Android: Guía de desarrollo de aplicaciones para Smartphones y Tabletas (2a .. Android: Guía de desarrollo de aplicaciones para Smartphones y Tabletas (2a ..* [En línea] octubre de 2014. [Citado el: 20 de noviembre de 2016.] https://books.google.com.cu/books?hl=es&lr=&id=DnQkGjh2H3MC&oi=fnd&pg=PA15&dq=ventajas+d e+las+aplicaciones+android&ots=G_1xeLYAZU&sig=mrvZ-B-VD4cOblcNGKf0W-SJsmE&redir_esc=y#v=onepage&q=ventajas%20de%20las%20aplicaciones%20android&f=false.

ANEXOS

Anexo 1: Descripción de historias de usuario.

Tabla 8 Crear una cuenta

Número: N/A	Nombre del requisito: Crear una cuenta
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 8 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
<p>Descripción</p> <p>1- Objetivo: Permitir a los usuarios crear una cuenta para entrar en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): N/A</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos usuario y contraseña son obligatorios.</p> <ul style="list-style-type: none"> • Nombre: Campo de texto que debe tener una longitud menor de 15 caracteres. No admite caracteres alfanuméricos, solo texto y debe iniciar con mayúscula. Admite espacio y caracteres extraños (tildes). Ejemplo: Irina Zhailí • Apellidos: Campo de texto que debe tener una longitud menor de 20 caracteres. No admite caracteres alfanuméricos, solo texto y debe iniciar con mayúscula. Admite espacio y caracteres extraños (tildes). Ejemplo: Pérez Román. • Usuario: Identificador de los usuarios en el sistema. Campo de texto que admite los caracteres alfanuméricos: arroba (@), números y punto (.). Deben estar escritos con letra minúscula. No se permite guión bajo (_). Longitud entre 4 y 6 caracteres. Ejemplo: aTcne4 • Correo electrónico: Campo de texto que identifica el email de un usuario del sistema Ejemplo: romeo@algo.cu • 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 6 a 16 caracteres. Ejemplo: ATcne4*C8 	

- **Imagen:** Campo de selección que representa la imagen del usuario. Los formatos para las imágenes son: png, jpg, jpeg. Este campo no es de carácter obligatorio.

4- Flujo de la acción a realizar:

Cuando el usuario seleccione la opción **Crear cuenta** el sistema debe mostrar la interfaz **Crear una cuenta** permitiendo introducir los siguientes datos obligatorios:

Datos personales:

- Imagen de cuenta
- (*) Nombre
- (*) Apellidos

Datos de la cuenta:

- (*) Usuario
- (*) Correo electrónico
- (*) Contraseña
- (*) Confirmar contraseña

Además, debe permitir seleccionar las siguientes opciones:

- Cancelar
- Iniciar

El botón **Iniciar** se habilitará una vez introducidos todos los datos de carácter obligatorios.

Cuando el usuario seleccione el botón Iniciar el sistema le mostrará la interfaz **Seleccionar clase** permitiendo entrar al sistema empleando el usuario y la contraseña creado.

Si elige el botón **Cancelar** el sistema retorna a la interfaz **Iniciar sesión**.

Si los datos introducidos son incorrectos el sistema debe mostrar los siguientes mensajes destacando estos en cada uno de los campos: *“Longitud menor de 15 y que inicie con mayúscula”*, *“Longitud menor de 20 y que inicie con mayúscula”*, *“Longitud entre 4 y 6 caracteres, admite valores alfanuméricos”*, *“Longitud entre 6 y 16 caracteres”*. En caso que las contraseñas no coincidan el sistema debe mostrar el siguiente error debajo del campo nueva contraseña: *“Las contraseñas no coinciden”*.

Observaciones:

Prototipo de interfaz:

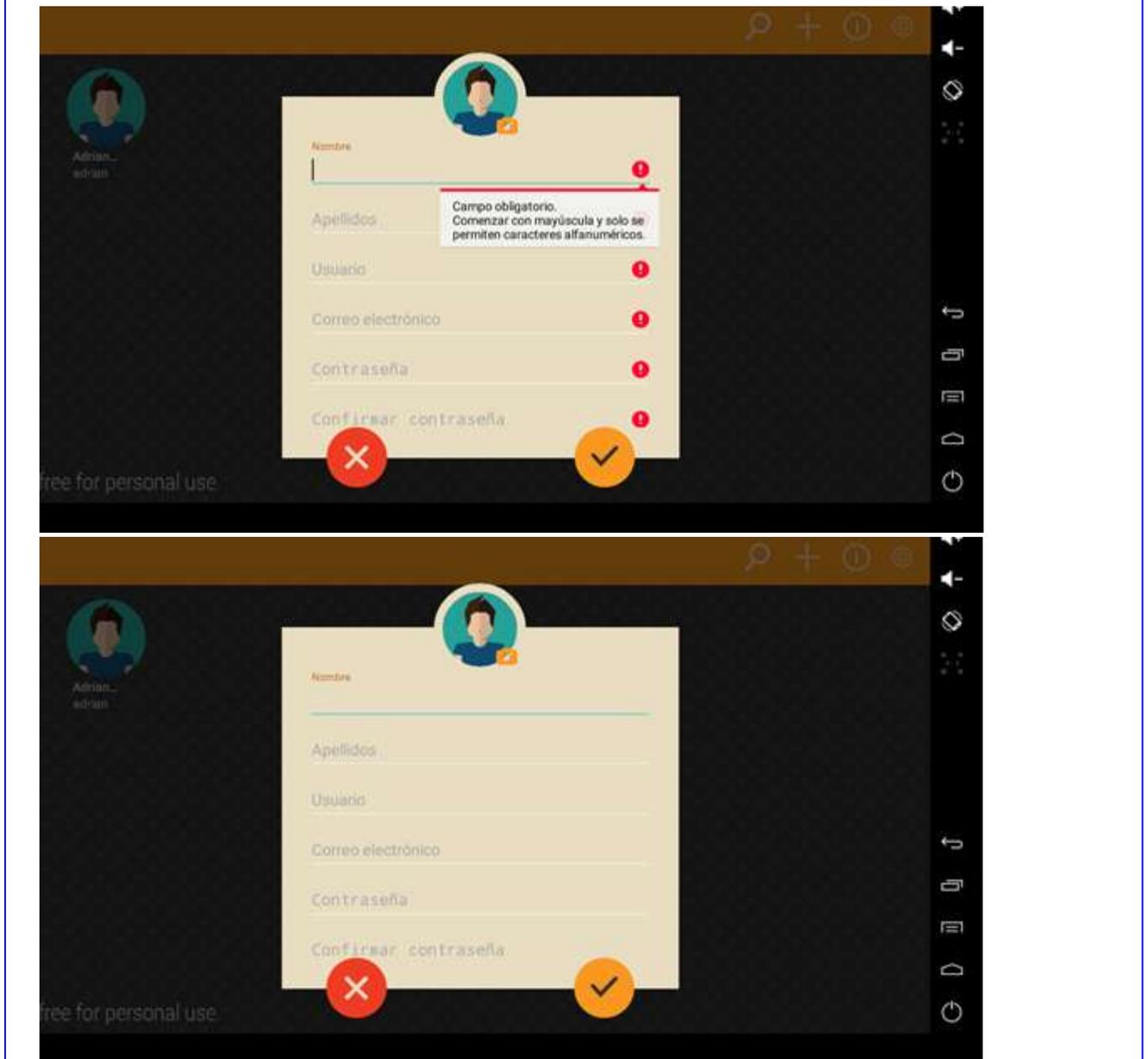


Tabla 9 Editar perfil

Número: N/A	Nombre del requisito: Editar perfil
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1

ANEXOS

Prioridad: Baja	Tiempo Estimado: 8 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción 1- Objetivo: Permitir a los usuarios editar un perfil creado con anterioridad. 2- Acciones para lograr el objetivo (precondiciones y datos): -Debe haber sido creada una cuenta con anterioridad. -Debe estar autenticado como usuario estudiante o profesor. 3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos usuario y contraseña son obligatorios. <ul style="list-style-type: none">• Nombre: Campo de texto que debe tener una longitud menor de 15 caracteres. No admite caracteres alfanuméricos, solo texto y debe iniciar con mayúscula. Admite espacio y caracteres extraños (tildes). Ejemplo: Irina Zhailí• Apellidos: Campo de texto que debe tener una longitud menor de 20 caracteres. No admite caracteres alfanuméricos, solo texto y debe iniciar con mayúscula. Admite espacio y caracteres extraños (tildes). Ejemplo: Pérez Román.• Usuario: Identificador de los usuarios en el sistema. Campo de texto que admite los caracteres alfanuméricos: arroba (@), números y punto (.). Deben estar escritos con letra minúscula. No se permite guión bajo (_). Longitud entre 4 y 6 caracteres. Ejemplo: aTcne4• 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 6 a 16 caracteres. Ejemplo: ATcne4*C8• Correo: Campo de texto que identifica el email por donde se podrá tener contacto con los usuarios. Debe tener la siguiente estructura: x@x.x• Imagen: Campo de selección que representa la imagen del usuario. Los formatos para las imágenes son: png, jpg, jpeg. Este campo no es de carácter obligatorio. 4- Flujo de la acción a realizar: El sistema debe permitir a los usuarios editar su perfil en el sistema. En caso de ser profesores se permitirá editar los siguientes campos: Datos personales:	

ANEXOS

- Imagen de cuenta
- (*) Nombre
- (*) Apellidos

Datos de la cuenta:

- (*) Usuario
- (*) Contraseña
- (*) Confirmar contraseña

En caso de ser estudiantes el sistema permitirá editar los siguientes datos :

- (*) Nombre
- (*) Apellidos
- (*) Correo
- (*) Usuario
- Imagen

El sistema debe permitir guardar los cambios realizados. Una vez guardados los cambios el sistema debe actualizar los nuevos datos de los usuarios.

Si los datos introducidos son incorrectos el sistema debe mostrar los siguientes mensajes destacando estos en cada uno de los campos: *“Longitud menor de 15 y que inicie con mayúscula”*, *“Longitud menor de 20 y que inicie con mayúscula”*, *“Longitud entre 4 y 6 caracteres, admite valores alfanuméricos”*, *“Longitud entre 6 y 16 caracteres”*. En caso que las contraseñas no coincidan el sistema debe mostrar el siguiente error debajo del campo nueva contraseña: *“Las contraseñas no coinciden”*.

Observaciones:

Prototipo de interfaz:

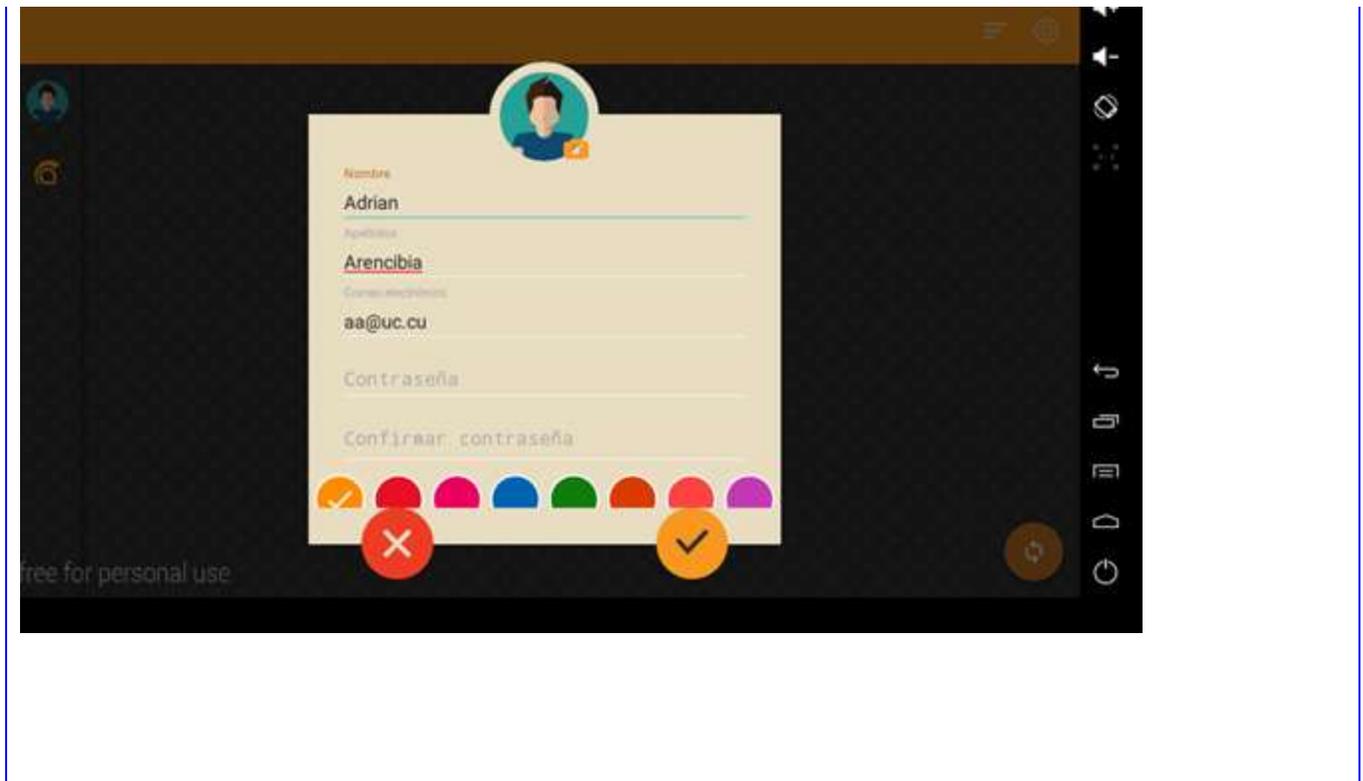


Tabla 10 Cerrar sesión

Número: N/A	Nombre del requisito: Cerrar sesión
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 3 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir a los usuarios cerrar sesión en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para cerrar sesión en el sistema: -Los usuarios deben estar autenticados en el sistema.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p>	

ANEXOS

4- Flujo de la acción a realizar:

El sistema debe permitir a los usuarios cerrar la sesión accediendo a la opción:

- Salir

Una vez seleccionada la opción el sistema terminara la sesión del usuario.

Observaciones:

A esta opción se accede una vez autenticado el usuario.

Prototipo de interfaz:

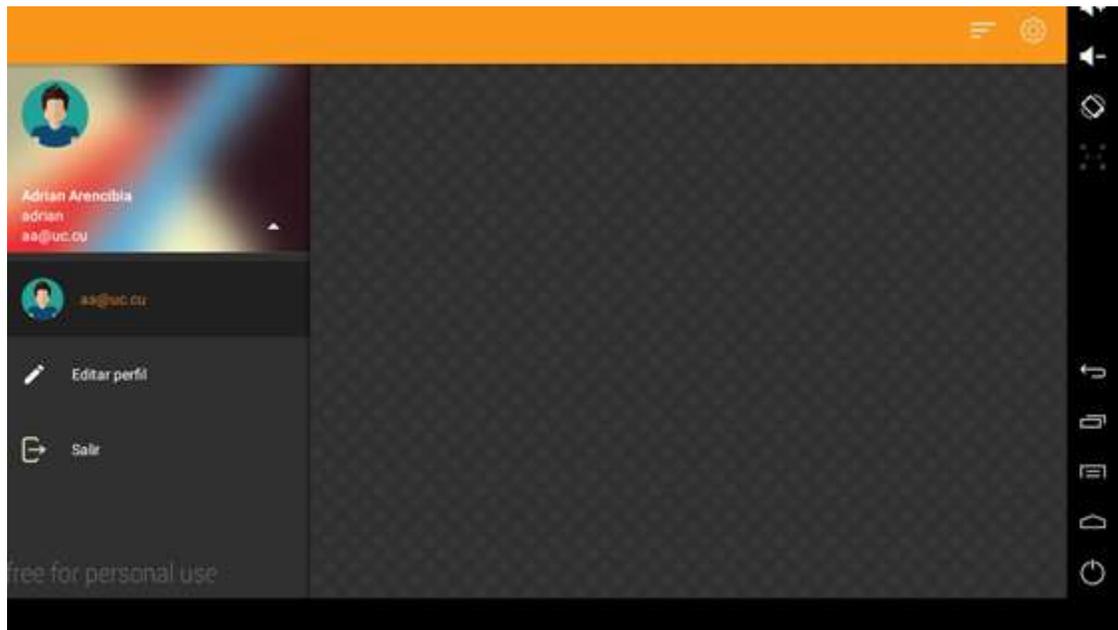


Tabla 11 Establecer conversación con el profesor

Número: N/A	Nombre del requisito: Establecer conversación con el profesor
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 8 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción:	
1- Objetivo:	
Permitir establecer conversación con el profesor.	

ANEXOS

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

Para establecer conversación con el profesor:

-Debe estar autenticado en el sistema y conectado a una clase.

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

4- Flujo de la acción a realizar:

El sistema debe permitir a los estudiantes enviar o recibir mensajes seleccionando la opción:

- Chat

Observaciones:

Prototipo de interfaz:

Tabla 12 Permitir conversaciones de los estudiantes

Número: N/A		Nombre del requisito: Permitir conversaciones de los estudiantes	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 5 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción:			
1- Objetivo: Permitir bloquear/desbloquear chat de los estudiantes.			
2- Acciones para lograr el objetivo (precondiciones y datos): Para bloquear/desbloquear chat de los estudiantes: - Debe existir al menos un estudiante conectado al chat. -Debe estar autenticado con usuario profesor.			
3- Comportamientos válidos y no válidos (flujo central y alternos):			

4- Flujo de la acción a realizar:

El sistema debe permitir al profesor acceder a la opción:

- Bloquear chat

El sistema muestra el listado de estudiantes conectados que no tienen el chat bloqueado permitiendo bloquear uno o varios estudiantes.

El sistema debe permitir seleccionar el o los estudiantes que desee bloquear. El sistema permite seleccionar la opción:

- Aceptar
- Cancelar

Al seleccionar la opción Aceptar el sistema bloquea el chat de los estudiantes seleccionados e inhabilita las acciones de los mismos en el chat.

Al seleccionar la opción Cancelar el sistema muestra la interfaz principal.

Para desbloquear el chat de un estudiante:

El sistema debe permitir al profesor acceder a la opción:

- Desbloquear chat

El sistema muestra el listado de estudiantes conectados que tienen bloqueado el chat permitiendo seleccionar uno o varios estudiantes.

El profesor selecciona el estudiante que desee y selecciona la opción:

- Desbloquear

El sistema permite seleccionar las opciones:

- Aceptar
- Cancelar

Al seleccionar la opción Aceptar el sistema habilita las acciones de los estudiantes en el chat.

Al seleccionar la opción Cancelar el sistema muestra la interfaz principal.

Observaciones:

Prototipo de interfaz:

Listado de estudiantes	Bloquear/Desbloquear
Pepe Perez	<input type="checkbox"/>

Tabla 13 Acceder a la clase por autenticación

Número: N/A	Nombre del requisito: Acceder a la clase por autenticación
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
<p>1- Objetivo: Permitir acceder a la clase por autenticación.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para acceder a una clase por autenticación: -El profesor tiene que definir con anterioridad que el estudiante debe acceder por autenticación a la clase especificada.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): El campo contraseña es obligatorio. Contraseña: campo de texto que permite cualquier carácter.</p>	

4- Flujo de la acción a realizar:

Una vez seleccionada la clase a la cual se desea acceder, el sistema debe permitir incluir el siguiente dato:

- (*) Contraseña

Una vez introducido el dato el sistema debe permitir seleccionar una de las siguientes opciones:

- Aceptar
- Cancelar

En caso de seleccionar la opción Aceptar el sistema debe permitir que el estudiante acceda a la clase especificada por el profesor.

En caso de seleccionar la opción Cancelar el sistema se mantiene en la interfaz donde se encuentra.

Observaciones:

El sistema debe permitir que los usuarios puedan introducir nuevamente los datos si se incumple alguna de las validaciones especificadas.

Prototipo de interfaz:

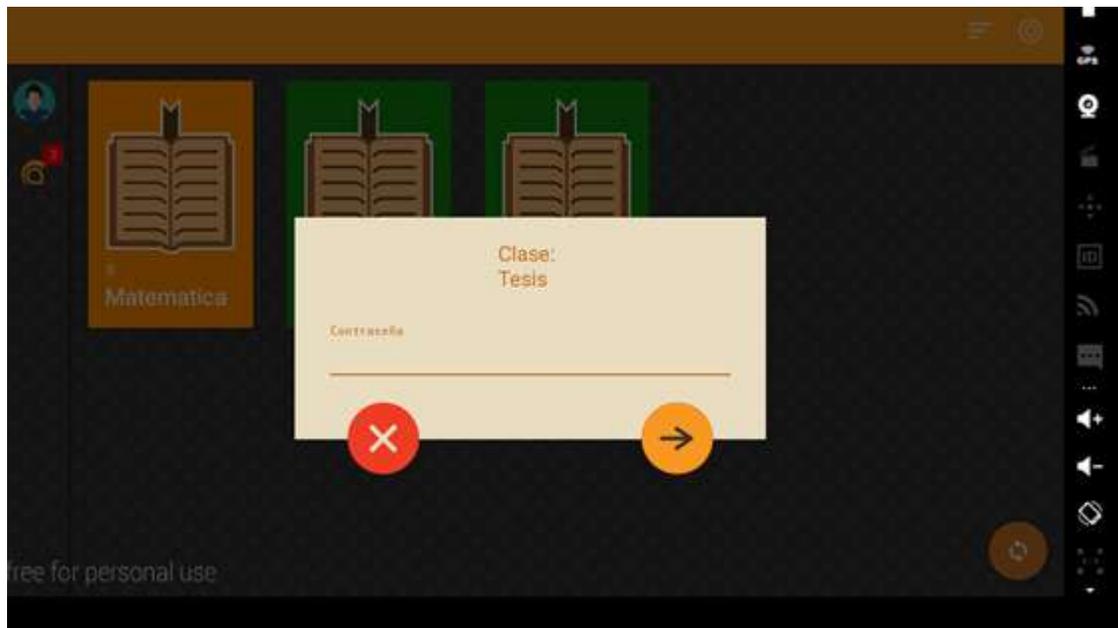
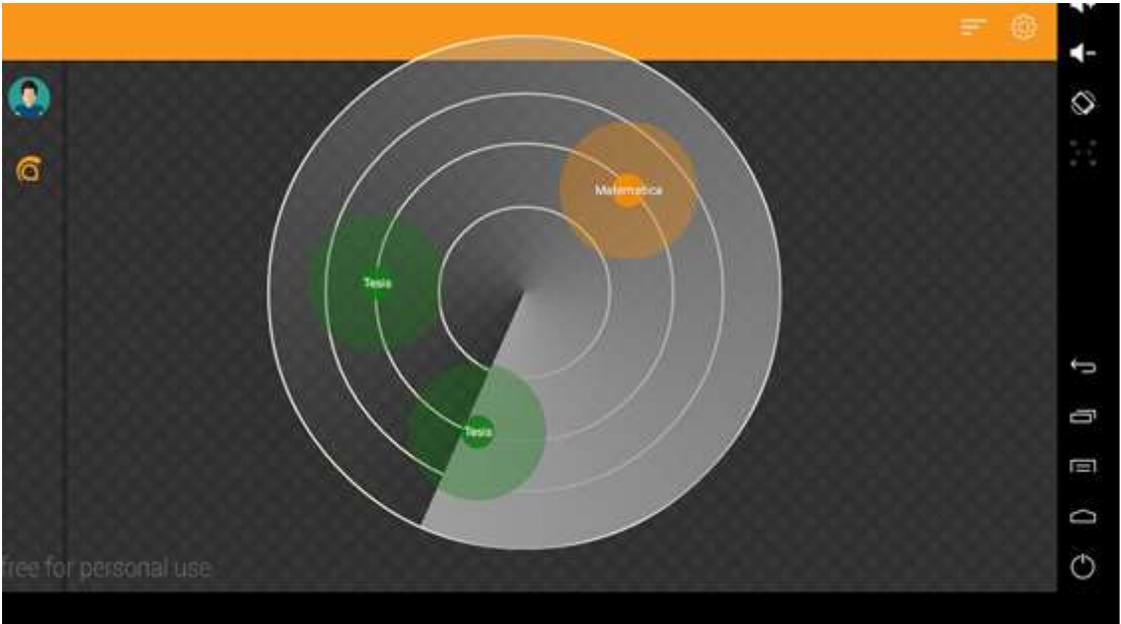


Tabla 14 Buscar clase

Número: N/A	Nombre del requisito: Buscar clase
--------------------	---

ANEXOS

Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 24 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción: 1- Objetivo: Permitir al estudiante buscar una clase creada anteriormente. 2- Acciones para lograr el objetivo (precondiciones y datos): -El usuario debe estar autentificado en el sistema. -Debe haberse al menos una clase creada. 3- Comportamientos válidos y no válidos (flujo central y alternos): N/A 4- Flujo de la acción a realizar: El sistema debe permitir al estudiante buscar las clases activas en el sistema.	
Observaciones:	
Prototipo de interfaz: 	

ANEXOS

Tabla 15 Cerrar sesión de estudiante

Número: N/A		Nombre del requisito: Cerrar sesión de estudiante	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 5 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción:			
1- Objetivo: Permitir cerrar sesión de estudiante.			
2- Acciones para lograr el objetivo (precondiciones y datos): Para cerrar sesión de un estudiante: -Debe estar autenticado en el sistema como usuario estudiante. -El estudiante debe tener la sesión abierta.			
3- Comportamientos válidos y no válidos (flujo central y alternos):			
4- Flujo de la acción a realizar: El sistema debe permitir al estudiante desconectarse de la clase en la que se encuentra, permitiendo seleccionar la siguiente opción: <ul style="list-style-type: none">• Desconectar Una vez seleccionada la opción Desconectar el sistema debe enviar una notificación al profesor informando que el estudiante desea desconectarse de la clase teniendo en cuenta la siguiente estructura: <i>El estudiante X desea desconectarse de la clase.</i> El sistema debe permitir al profesor aceptar o rechazar la notificación seleccionando para ello una de las siguientes opciones: <ul style="list-style-type: none">• Aceptar• Rechazar En caso de seleccionar la opción Aceptar, el sistema permite al estudiante desconectarse de la clase. En caso de seleccionar la opción Rechazar, el sistema no permite al estudiante desconectarse de la clase y envía una notificación al estudiante para informarle que su solicitud no ha sido aceptada a través del siguiente mensaje: <i>Su solicitud no ha sido aceptada debe continuar conectado a la clase.</i>			

ANEXOS

Observaciones:

X: Estudiante que ha seleccionado la opción Desconectar

Prototipo de interfaz:

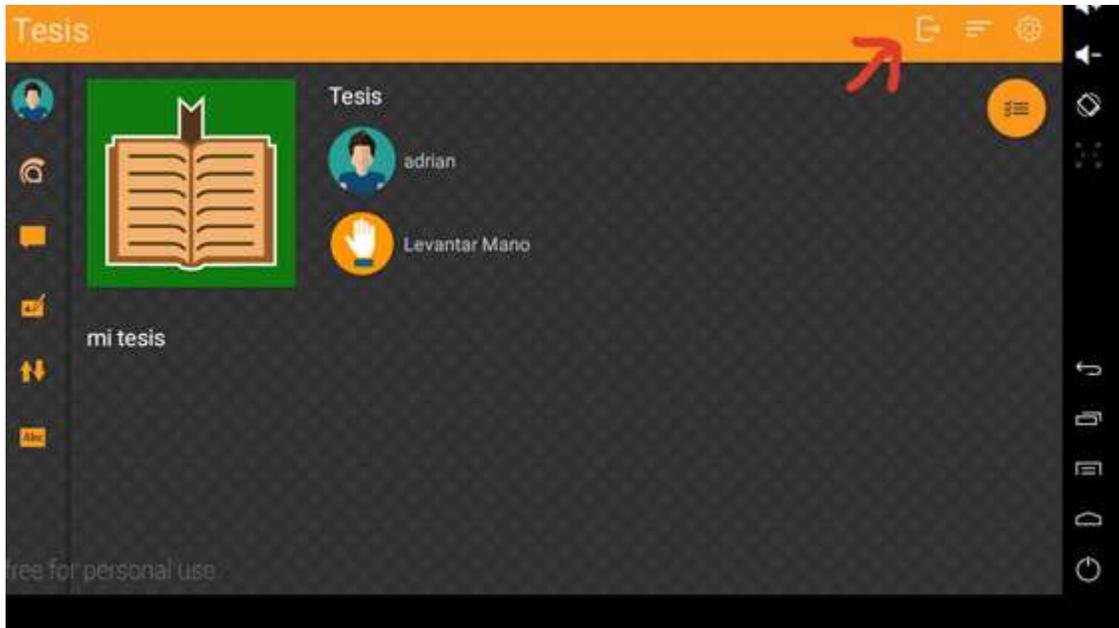


Tabla 16 Desconectar estudiantes de la clase

Número: N/A	Nombre del requisito: Desconectar estudiantes de la clase
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 6 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir al profesor desconectar estudiantes de la clase.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para desconectar estudiantes de la clase: -El usuario debe estar autenticado en el sistema como usuario profesor. -Debe haber al menos un estudiante conectado a una clase.</p>	

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

4- Flujo de la acción a realizar:

El sistema debe permitir al profesor desde su ordenador desconectar de la clase a el o los estudiantes seleccionados de la lista de estudiantes conectados, permitiendo seleccionar la opción:

- Desconectar estudiante de la clase

El sistema debe pedir confirmación para desconectar a los estudiantes a través del siguiente mensaje: *¿Está seguro que desea desconectar a los estudiantes seleccionados?*

El sistema debe permitir seleccionar una de las siguientes opciones:

- Aceptar
- Cancelar

Al seleccionar la opción Aceptar el sistema debe desconectar a los estudiantes seleccionados del sistema y a su vez debe notificar al estudiante que ha sido desconectado a través del siguiente mensaje: *Usted ha sido desconectado de la clase por el profesor.*

Al seleccionar la opción Cancelar el sistema se mantiene en la interfaz donde estaba.

Observaciones:

Prototipo de interfaz:

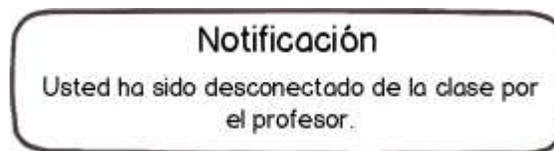


Tabla 17 Seleccionar clase

Número: N/A		Nombre del requisito: Seleccionar clase	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	

ANEXOS

Prioridad: Alta	Tiempo Estimado: 4 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción: 1- Objetivo: Permitir al estudiante seleccionar el aula a la cual desea acceder. 2- Acciones para lograr el objetivo (precondiciones y datos): Para seleccionar un aula: -Debe estar autenticado en el sistema como usuario estudiante. -Debe haber al menos un aula creada. 3- Comportamientos válidos y no válidos (flujo central y alternos): 4- Flujo de la acción a realizar: El sistema debe permitir al estudiante visualizar en su terminal el listado de aulas activas en ese momento, para ello debe seleccionar la opción: <ul style="list-style-type: none">• Aulas activas Además el sistema permite seleccionar el aula al que desea acceder y permite seleccionar la opción: <ul style="list-style-type: none">• Aceptar El sistema debe enviar una solicitud al profesor en caso que sea la primera vez que el estudiante quiera acceder al aula seleccionada a través del siguiente mensaje: <i>El estudiante X solicita incorporarse a la clase.</i> El sistema debe permitir además seleccionar una de las siguientes opciones: <ul style="list-style-type: none">• Aceptar• Rechazar Al seleccionar la opción Aceptar el sistema incorpora al estudiante al aula seleccionada. Al seleccionar la opción Rechazar el sistema envía una notificación al estudiante a través del siguiente mensaje: <i>Usted no puede conectarse al aula seleccionada.</i>	
Observaciones: X: Nombre del estudiante en cuestión	
Prototipo de interfaz:	

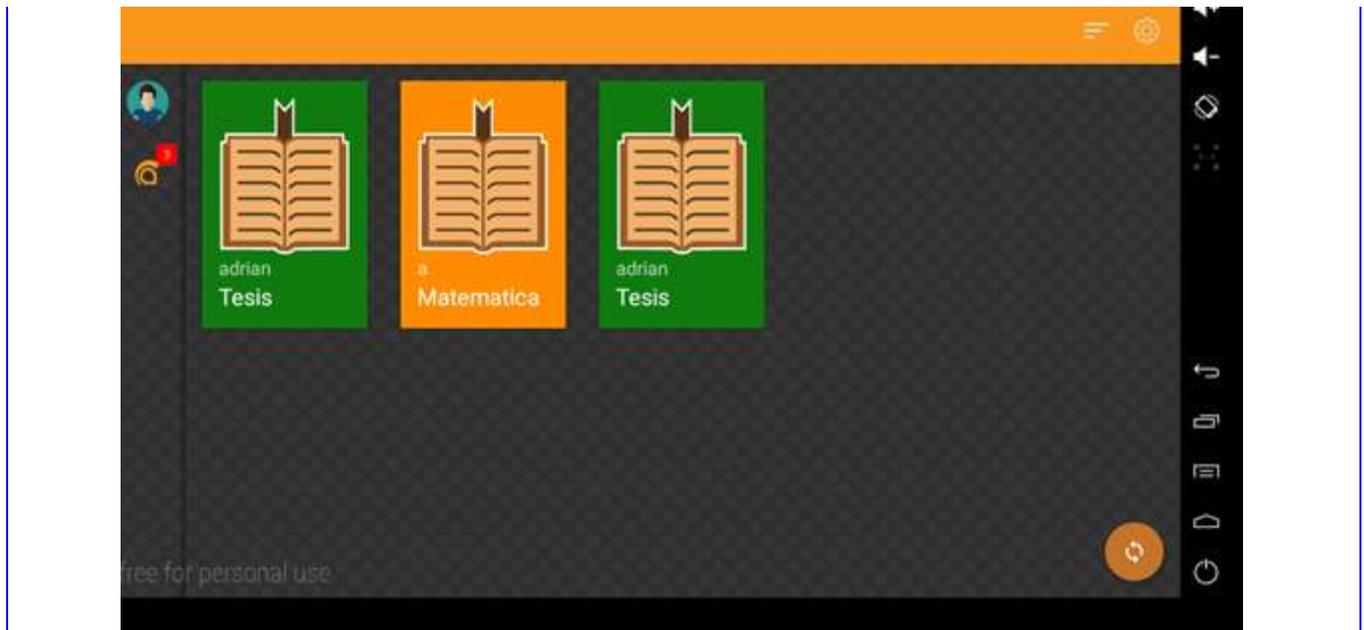


Tabla 18 Abrir sitio web remoto

Número: N/A	Nombre del requisito: Abrir sitio web remoto
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 7 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir abrir sitio web remoto.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para abrir sitio web remoto: -Debe existir al menos un estudiante conectado a la clase. -Debe estar autenticado con usuario profesor.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos):</p> <p>4- Flujo de la acción a realizar:</p>	

ANEXOS

El sistema debe permitir al profesor acceder a la opción:

- Abrir sitio web

El sistema muestra el listado de estudiantes conectados permitiendo seleccionar los que desee.

Una vez seleccionados los estudiantes el sistema debe permitir seleccionar la opción:

- Siguiente

El sistema permite al profesor introducir la URL del sitio web al que desea acceder.

Una vez introducida la URL el profesor selecciona la opción:

- Aceptar

Al seleccionar la opción Aceptar el sistema debe abrir en los terminales de los estudiantes seleccionados el sitio web especificado.

Observaciones:

Prototipo de interfaz:

El prototipo de interfaz de usuario se divide en dos secciones principales:

- Listado de estudiantes:** Una ventana con el título "Listado de estudiantes" que muestra una lista de cuatro nombres "Pepe Perez". Cada nombre tiene un cuadro de verificación a su izquierda. Los segundos y últimos nombres tienen sus cuadros de verificación marcados. Debajo de la lista hay un botón rectangular con el texto "Siguiente".
- Sitio web:** Una ventana con el título "Sitio web" que muestra el texto "URL:" seguido de un campo de entrada rectangular vacío. Debajo del campo hay un botón rectangular con el texto "Aceptar".

Tabla 19 Configurar lenguaje del dispositivo remoto

Número: N/A

Nombre del requisito: Configurar lenguaje del dispositivo remoto

ANEXOS

Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 6 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción: 1- Objetivo: Permitir al profesor configurar el lenguaje del terminal de los estudiantes de forma remota. 2- Acciones para lograr el objetivo (precondiciones y datos): Para configurar el lenguaje del terminal remoto: -Estar autenticado en el sistema como usuario profesor. 3- Comportamientos válidos y no válidos (flujo central y alternos): 4- Flujo de la acción a realizar: El sistema debe permitir al profesor desde su ordenador seleccionar el lenguaje del terminal remoto seleccionando la opción: <ul style="list-style-type: none">Terminal remoto Una vez seleccionada la opción Terminal remoto el sistema debe permitir seleccionar una de las siguientes opciones: <ul style="list-style-type: none">Terminal del estudianteLenguaje del terminal del estudiante El sistema debe permitir seleccionar la opción Aceptar.	
Observaciones:	
Prototipo de interfaz: 	

ANEXOS

Tabla 20 Iniciar aplicación remota por el profesor

Número: N/A		Nombre del requisito: Iniciar aplicación remota por el profesor	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 8 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción: 1- Objetivo: Permitir iniciar aplicación remota por el profesor. 2- Acciones para lograr el objetivo (precondiciones y datos): Para iniciar una aplicación remota por el profesor: -Debe existir al menos un estudiante conectado a la clase. -Debe existir al menos una aplicación disponible para iniciar remotamente. -Debe estar autenticado con usuario profesor. 3- Comportamientos válidos y no válidos (flujo central y alternos): 4- Flujo de la acción a realizar: El sistema debe permitir al profesor acceder a la opción: <ul style="list-style-type: none">• Iniciar aplicación remota Una vez seleccionada la opción Iniciar aplicación remota el sistema muestra las aplicaciones disponibles en el terminal de los estudiantes. El profesor puede seleccionar la aplicación deseada y permite seleccionar la opción: <ul style="list-style-type: none">• Aceptar• Cancelar Al seleccionar la opción Aceptar el sistema inicia la aplicación seleccionada en los terminales de los estudiantes conectados en la clase. Al seleccionar la opción Cancelar el sistema muestra el listado de aplicaciones remotas.			
Observaciones:			

ANEXOS

Prototipo de interfaz:

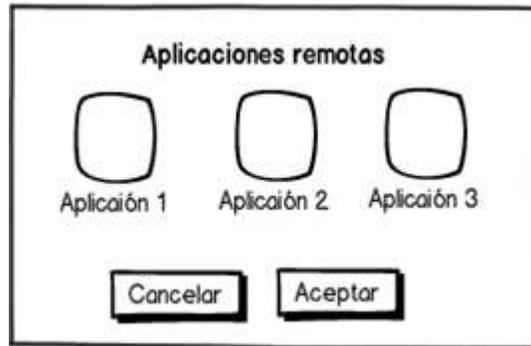


Tabla 21 Notificar estudiantes de la evaluación

Número: N/A	Nombre del requisito: Notificar estudiantes de la evaluación
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 4 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real:
Descripción:	
1- Objetivo: Permitir notificar estudiantes de la evaluación.	
2- Acciones para lograr el objetivo (precondiciones y datos): Para notificar estudiantes de la evaluación: -El profesor debe haber evaluado la respuesta del estudiante. -Los estudiantes deben estar conectados a la clase.	
3- Comportamientos válidos y no válidos (flujo central y alternos):	
4- Flujo de la acción a realizar: El sistema debe permitir al profesor emitir desde el ordenador la evaluación al estudiante mediante una notificación con los siguientes datos: <p style="text-align: center;"><i>Usted ha obtenido la calificación de:</i> Cuantitativa: X Cualitativa: Y</p>	

ANEXOS

Debe permitir, además, enviar la notificación de forma privada o colectiva.

La notificación será enviada a través de la opción:

- Mostrar evaluación a los estudiantes

Observaciones:

(X: Evaluación otorgada por el profesor)

Prototipo de interfaz:

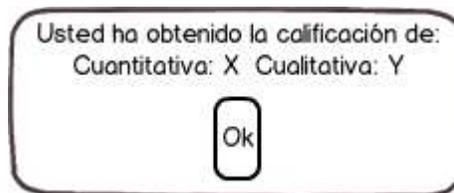


Tabla 22 Aceptar mano levantada

Número: N/A		Nombre del requisito: Aceptar mano levantada	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 7 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción:			
1- Objetivo: Permitir aceptar mano levantada.			
2- Acciones para lograr el objetivo (precondiciones y datos): Para aceptar mano levantada: - Debe estar autenticado con usuario profesor. - Debe existir al menos un estudiante que haya realizado la acción levantar mano.			
3- Comportamientos válidos y no válidos (flujo central y alternos):			
4- Flujo de la acción a realizar: Cuando un estudiante selecciona la opción Mano levantada el sistema muestra en el ordenador del profesor una ventana con el nombre de los terminales de los estudiantes que realizaron la acción permitiendo			

ANEXOS

seleccionar el estudiante que desee.

El sistema debe permitir seleccionar al estudiante y luego brinda la opción de seleccionar la opción:

- Aceptar

Una vez seleccionado la opción Aceptar el sistema debe enviar una notificación al estudiante a través del siguiente mensaje: *Usted ha sido seleccionado.*

Observaciones:

Prototipo de interfaz:

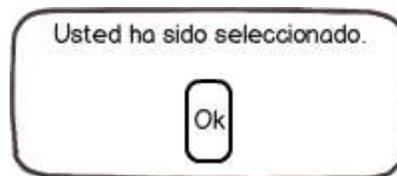


Tabla 23 Denegar mano levantada

Número: N/A		Nombre del requisito: Denegar mano levantada	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 4 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción:			
1- Objetivo: Permitir denegar mano levantada.			
2- Acciones para lograr el objetivo (precondiciones y datos): Para denegar mano levantada: - Debe existir al menos un estudiante que haya realizado la acción levantar mano. - Debe estar autenticado con usuario profesor.			
3- Comportamientos válidos y no válidos (flujo central y alternos):			
4- Flujo de la acción a realizar:			

ANEXOS

Una vez que el estudiante seleccione la opción mano levantadas el sistema muestra al profesor las terminales de los estudiantes con el ícono que representa la mano levantada.

El sistema debe permitir al profesor denegar la opción mano levantada una vez que el estudiante haya seleccionado la opción seleccionando para ello los estudiantes a quien denegará dicha acción.

Una vez seleccionados los estudiantes el sistema permitirá seleccionar la opción:

- Denegar mano levantada

Una vez seleccionada la opción Denegar mano levantada el sistema notifica al estudiante que su petición ha sido denegada a través del siguiente mensaje: *Su petición ha sido denegada.*

El sistema mostrará las terminales de los estudiantes con la acción denegada.

Observaciones:

Prototipo de interfaz:

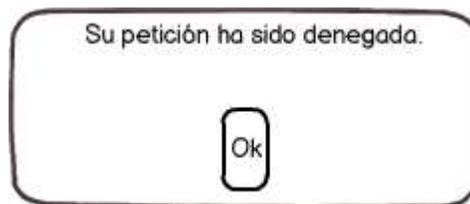


Tabla 24 Permitir manos levantadas

Número: N/A		Nombre del requisito: Permitir manos levantadas	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 6 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción:			
1- Objetivo:			
Permitir bloquear o desbloquear la funcionalidad de manos levantadas de los estudiantes.			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para bloquear o desbloquear la funcionalidad de manos levantadas de los estudiantes:			
-Debe existir al menos un estudiante conectado.			
-El sistema debe estar en la interfaz Listado de estudiantes.			
-Debe estar autenticado con usuario profesor.			

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

4- Flujo de la acción a realizar:

Para bloquear la funcionalidad de mano levantada de un estudiante:

El sistema muestra el terminal de cada estudiante con el ícono de la mano levantada. El sistema debe permitir seleccionar a que estudiantes bloqueará la acción seleccionando la opción:

- Bloquear mano levantada

Al seleccionar la opción bloquear mano levantada el sistema bloqueará todas las notificaciones de mano levantada del estudiante escogido.

Para desbloquear la acción mano levantada de un estudiante:

El sistema muestra el terminal de cada estudiante con el ícono de la mano levantada especificando que está bloqueado. El sistema debe permitir seleccionar a que estudiantes desbloqueará la acción seleccionando la opción:

- Desbloquear mano levantada

Al seleccionar la opción desbloquear mano levantada el sistema permitirá recibir la notificación de mano levantada del estudiante seleccionado.

Observaciones:

Prototipo de interfaz:

Tabla 25 Transmitir pantalla del profesor en modo completa

Número: N/A	Nombre del requisito: Transmitir pantalla del profesor en modo completa
Programador: Adrian Arencibia Herrera	Iteración Asignada: I1
Prioridad: Media	Tiempo Estimado: 10 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
Descripción:	
1- Objetivo:	
Permitir transmitir la pantalla del profesor en modo completa en los terminales de uno o varios estudiantes en la	

ANEXOS

clase.

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

Para transmitir la pantalla del profesor en modo completa en el terminal de un estudiante:

-Debe existir al menos un estudiante conectado a la clase.

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

4- Flujo de la acción a realizar:

-El profesor selecciona la opción Pantalla en modo completa y el sistema muestra un listado de los estudiantes conectados a la clase.

-El profesor selecciona uno o varios estudiantes del listado que se muestra, selecciona la opción Transmitir y el sistema permite transmitir la pantalla en modo completa del profesor en los terminales de el o los estudiantes seleccionados.

-Los estudiantes solo podrán observar la pantalla del profesor y tomarle fotos.

Observaciones:

Prototipo de interfaz:

Tabla 26 Ver pantalla de los estudiantes

Número: N/A		Nombre del requisito: Ver pantalla de los estudiantes	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Media		Tiempo Estimado: 5 horas	
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]		Tiempo Real: N/A	
Descripción:			
1- Objetivo: Permitir ver la pantalla de un estudiante.			
2- Acciones para lograr el objetivo (precondiciones y datos): Para ver la pantalla de un estudiante: - Debe existir al menos un estudiante conectado a la clase. - Debe estar autenticado con usuario profesor.			

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

4- Flujo de la acción a realizar:

El sistema debe permitir al profesor visualizar las pantallas de los terminales de los estudiantes seleccionando la opción:

- Ver pantalla de los estudiantes

El sistema debe mostrar el listado de los estudiantes conectados a la clase.

El sistema debe permitir al profesor seleccionar a los estudiantes para ver las pantallas de sus terminales.

Una vez seleccionados los estudiantes el sistema debe permitir seleccionar la opción:

- Ver

Al seleccionar la opción Ver el sistema permite al profesor ver la pantalla del estudiante seleccionado en su ordenador.

Observaciones:

Prototipo de interfaz:

Tabla 27 Parar video

Número: N/A	Nombre del requisito: Parar video.
Programador: Adrian Arencibia Herrera	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 6 horas
Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]	Tiempo Real: N/A
<p>Descripción:</p> <p>1- Objetivo: Permitir al profesor desde su ordenador parar la reproducción del video que se esta ejecutando en la clase.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para parar el video hay que: - Haber seleccionado un archivo de video de la lista de reproducción.</p>	

ANEXOS

- Estar reproduciendo un archivo de video .
- Estar autenticado con usuario profesor.

3- Flujo de la acción a realizar:

El sistema debe permitir al profesor desde su ordenador parar la reproducción del video que se esta ejecutando en la clase, seleccionando la opción:

- Parar video.

Al seleccionar la opción Para video el reproductor se detiene y la barra de reproducción se posiciona en el tiempo 00.00.00.

Observaciones:

Prototipo de interfaz:

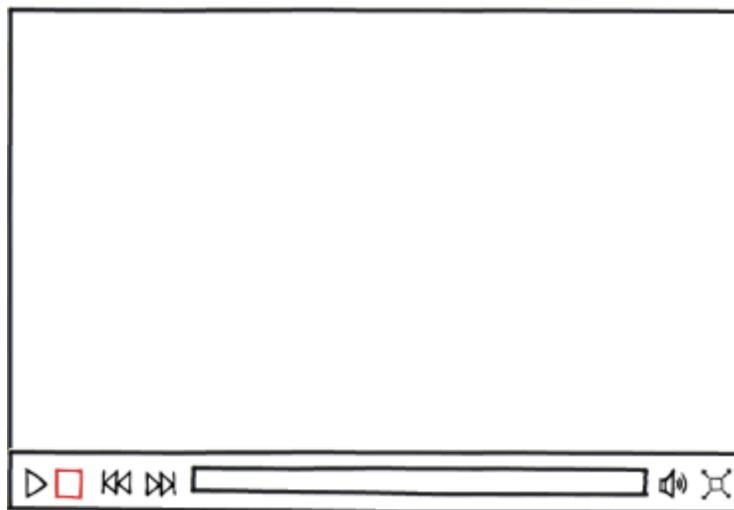


Tabla 28 Reproducir archivo de video

Número: N/A		Nombre del requisito: Reproducir archivo de video	
Programador: Adrian Arencibia Herrera		Iteración Asignada: 1	
Prioridad: Alta		Tiempo Estimado: 6 horas	

Riesgo en Desarrollo: Falta de experiencia con el desarrollo de las herramientas y tecnologías [2]

Tiempo Real :N/A

Descripción:

1- Objetivo:

Permitir al profesor desde su ordenador reproducir un archivo de video en su pantalla y transmitirlo hacia las terminales de los estudiantes conectados a la clase.

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

Para reproducir un archivo de video hay que:

- Haber seleccionado un archivo de la lista de reproducción.
- Debe estar autenticado con usuario profesor.

3- Flujo de la acción a realizar:

El sistema debe permitir al profesor desde su ordenador reproducir un archivo de video en su pantalla y transmitirlo hacia las terminales de los estudiantes conectados seleccionando la opción:

- Reproducir video

Observaciones:

Prototipo de interfaz:



Anexo 2: Diagrama de clases del diseño

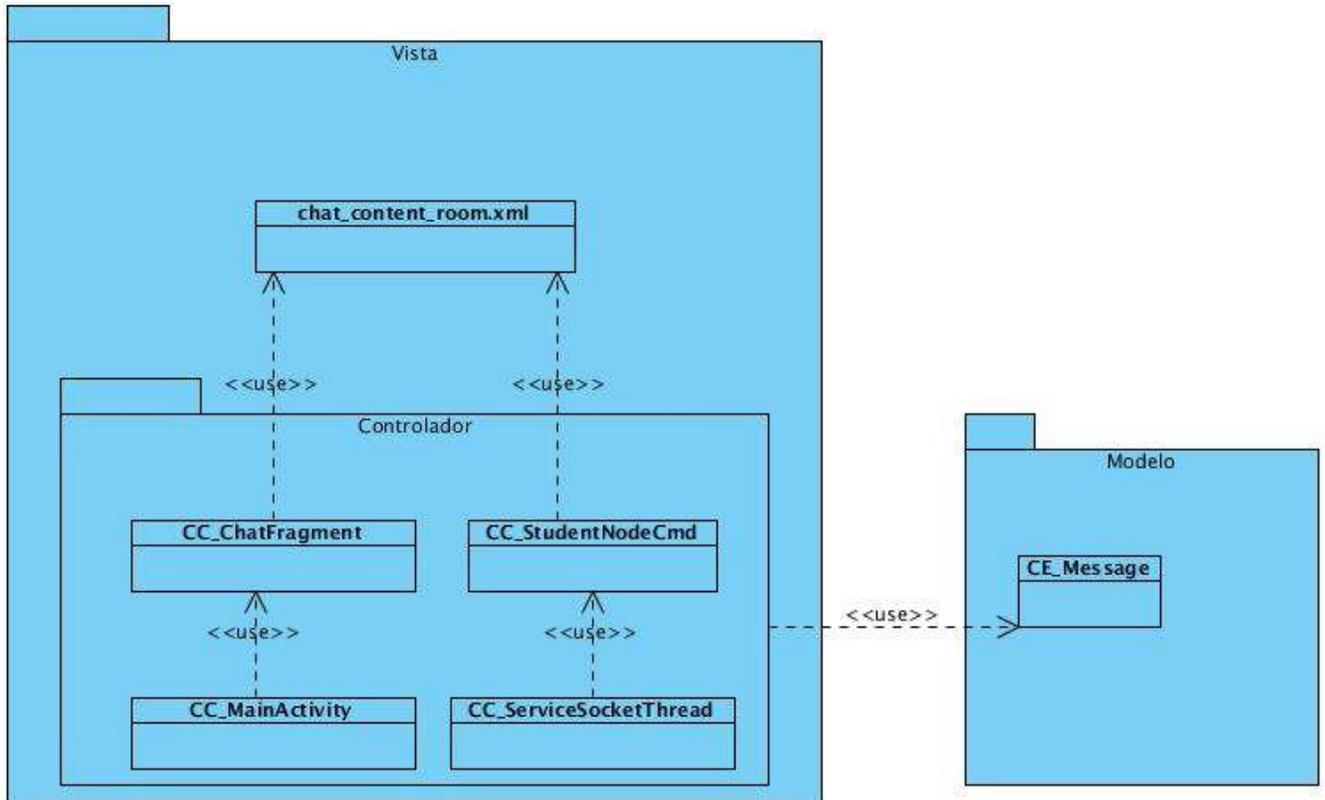


Figura 9 Diagrama de clases del diseño Paquete Chat

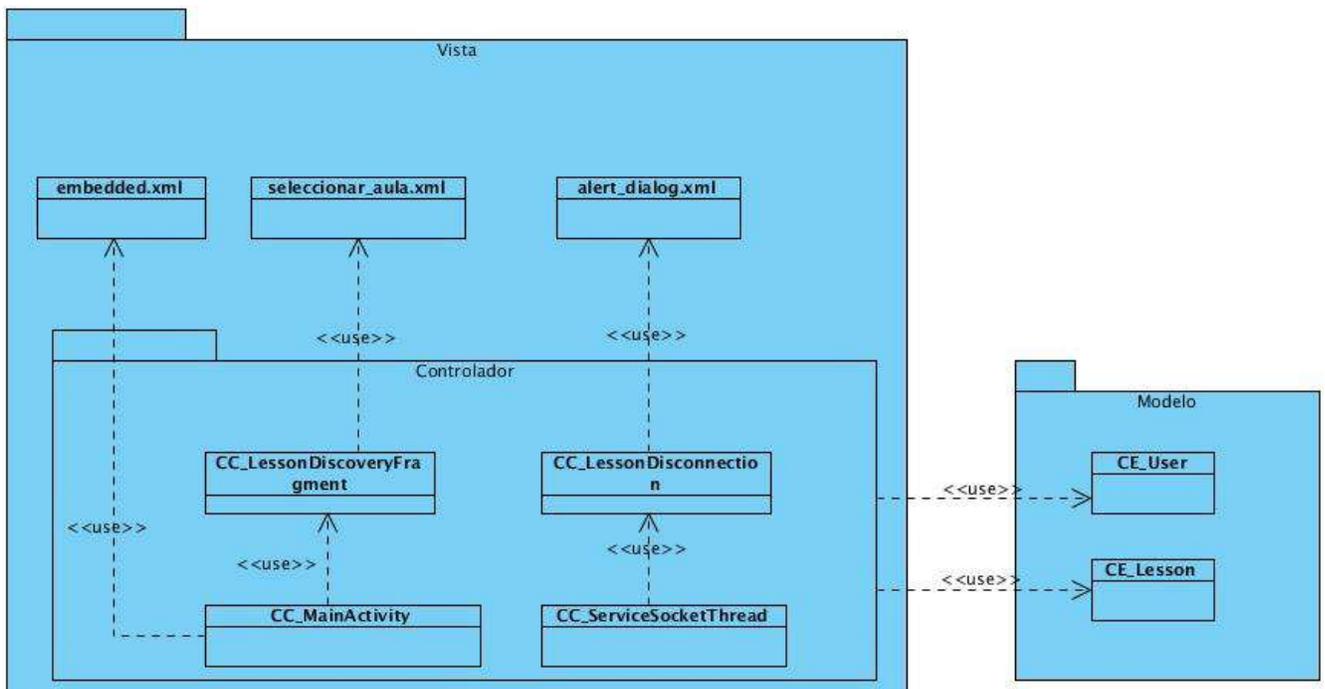


Figura 10 Diagrama de clases del diseño Paquete Clase

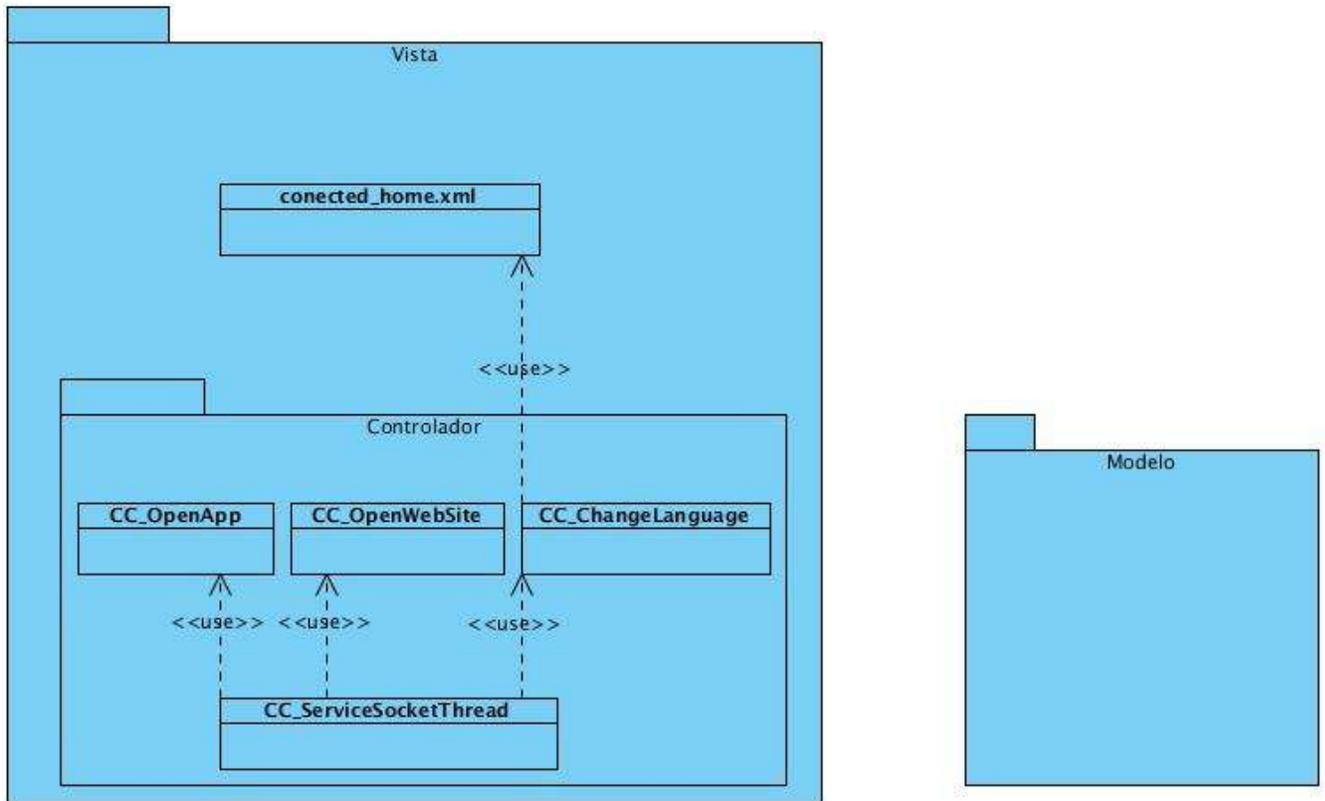


Figura 11 Diagrama de clases del diseño Paquete Control remoto

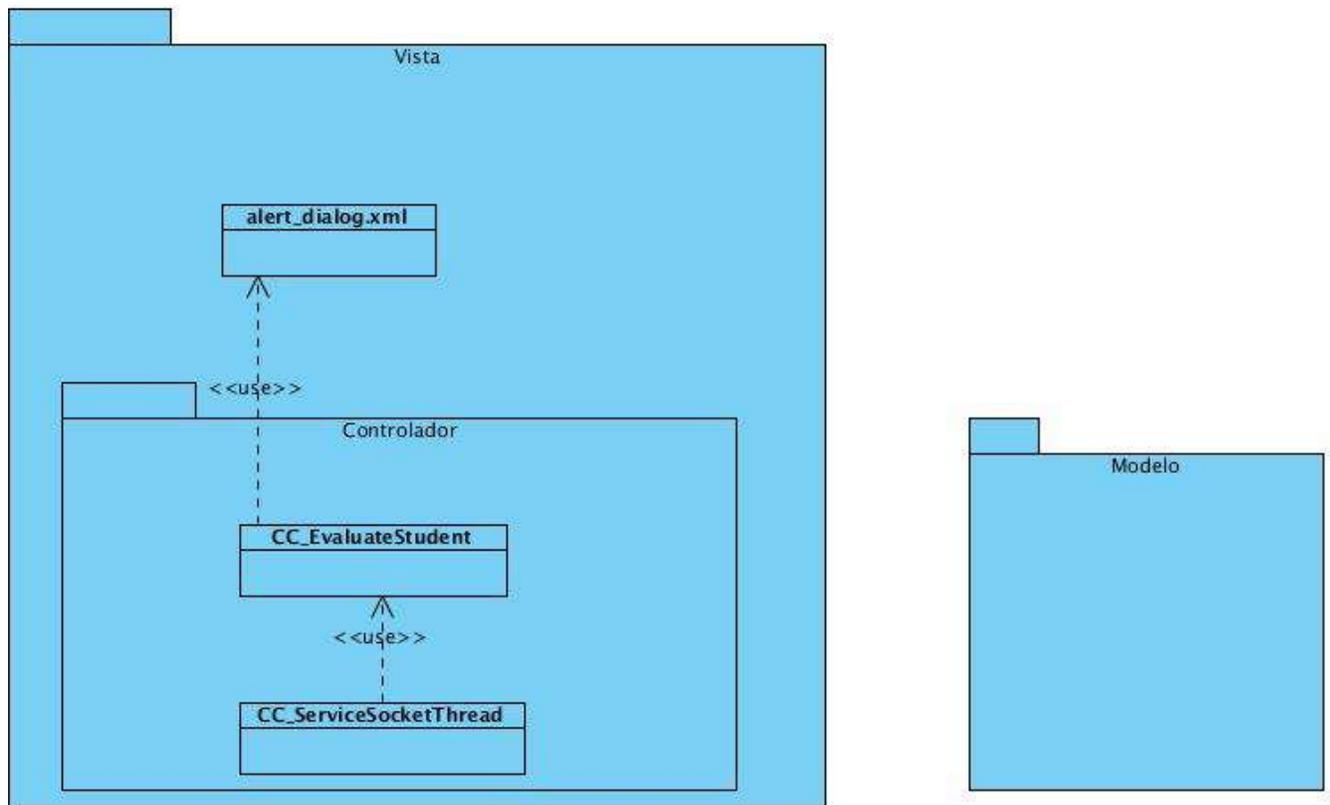


Figura 12 Diagrama de clases del diseño Paquete Evaluación

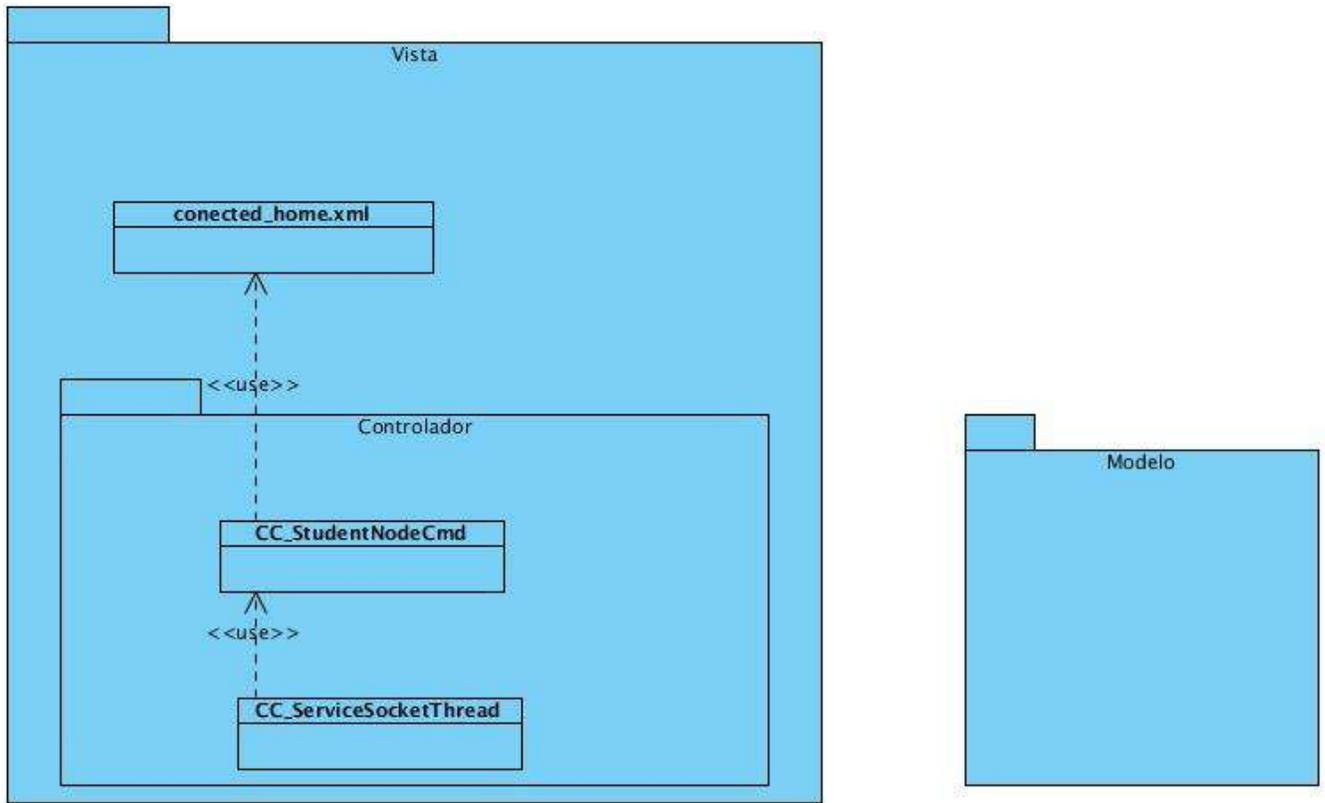


Figura 13 Diagrama de clases del diseño Paquete Mano

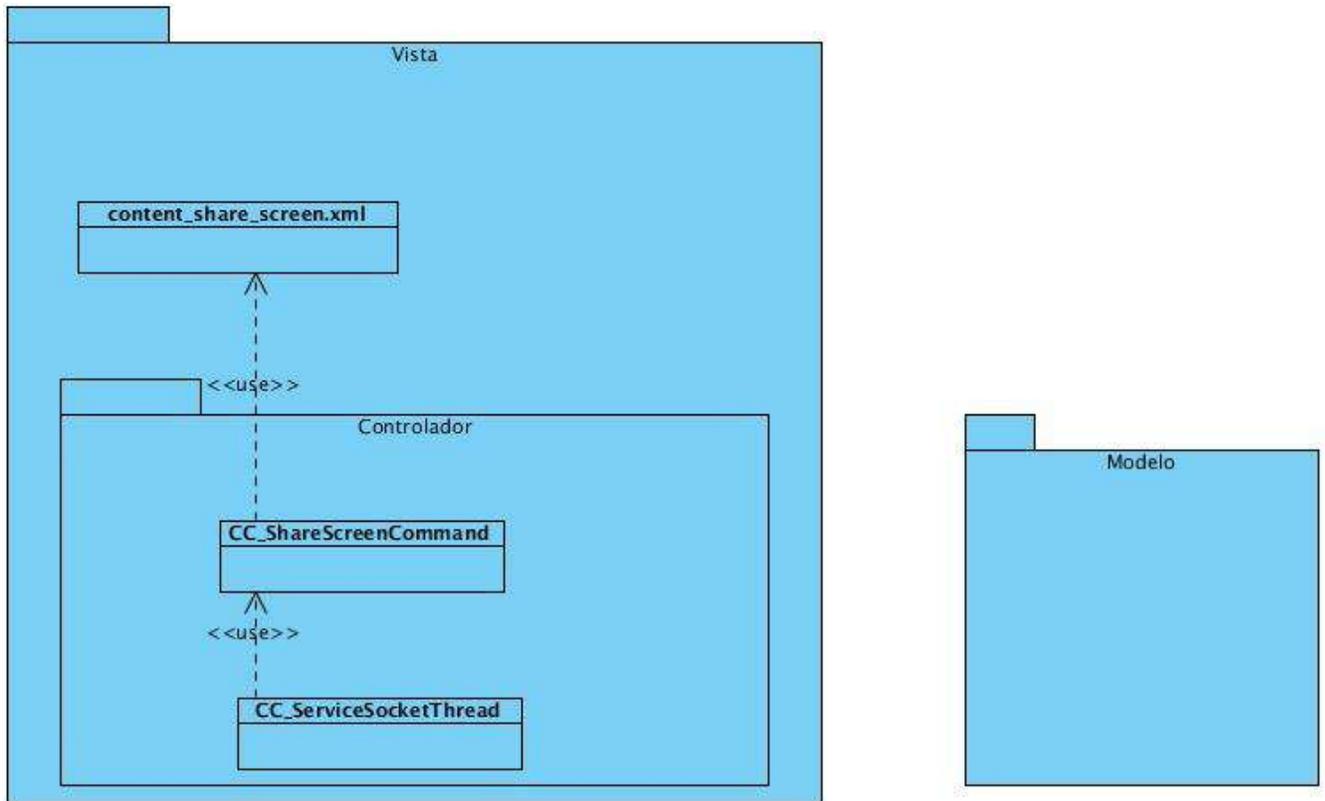


Figura 14 Diagrama de clases del diseño Paquete Pantalla

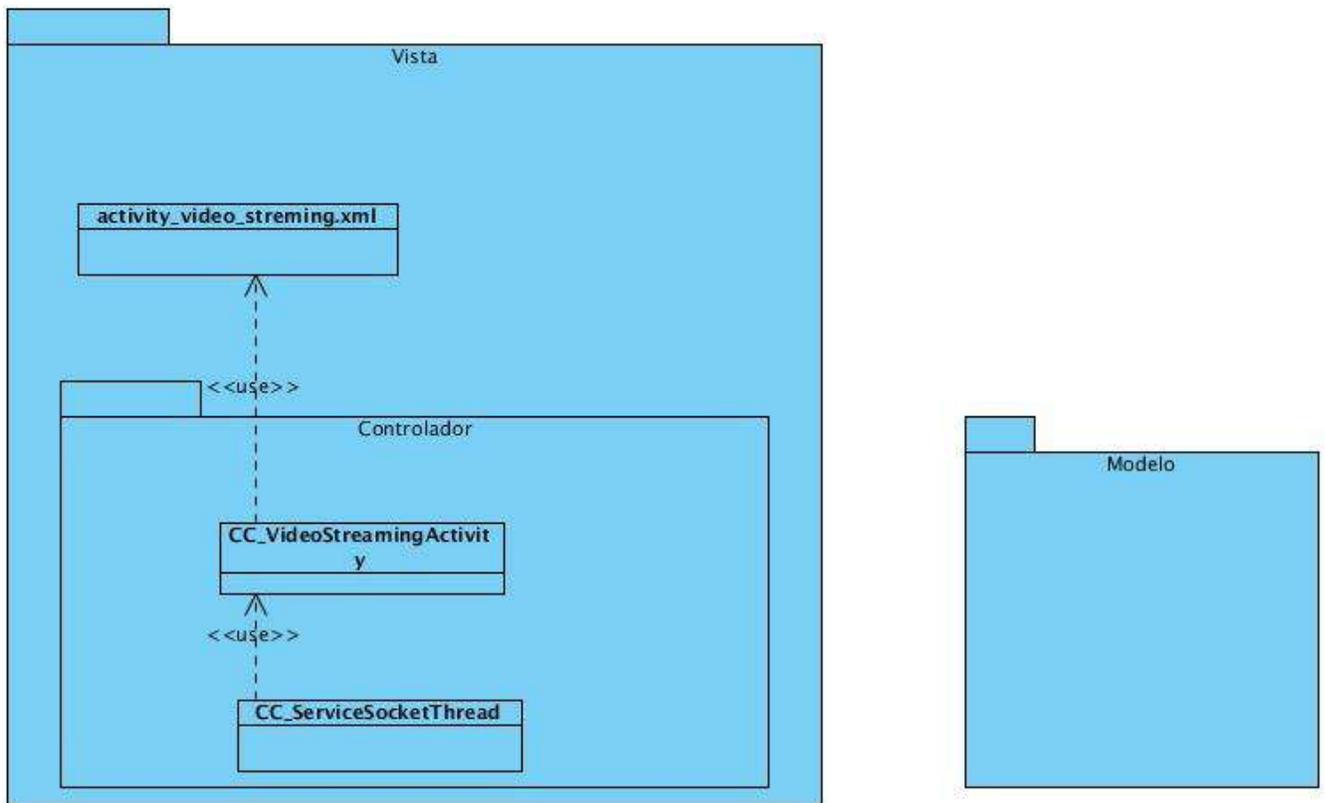


Figura 15 Diagrama de clases del diseño Paquete Video

Anexo 3: Diagramas de secuencia del diseño

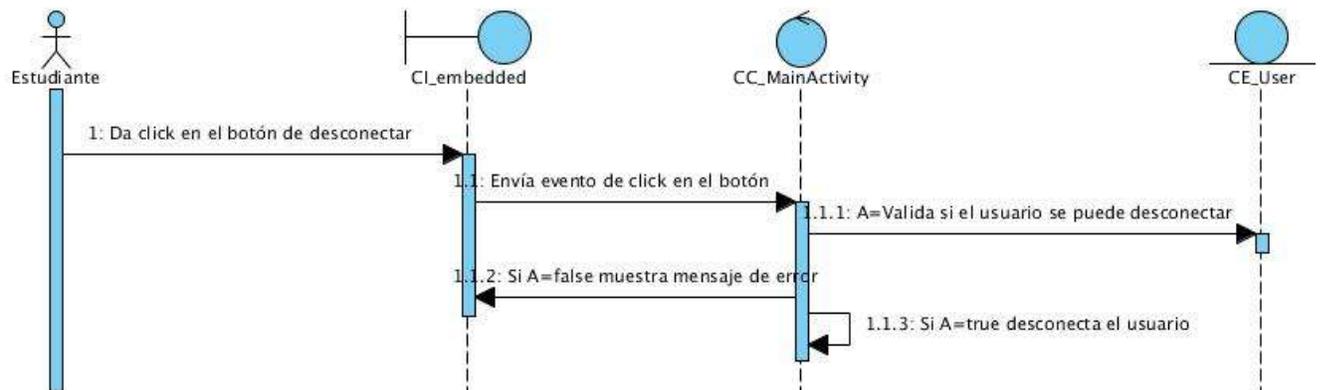


Figura 16 Diagrama de secuencia Terminar sesión

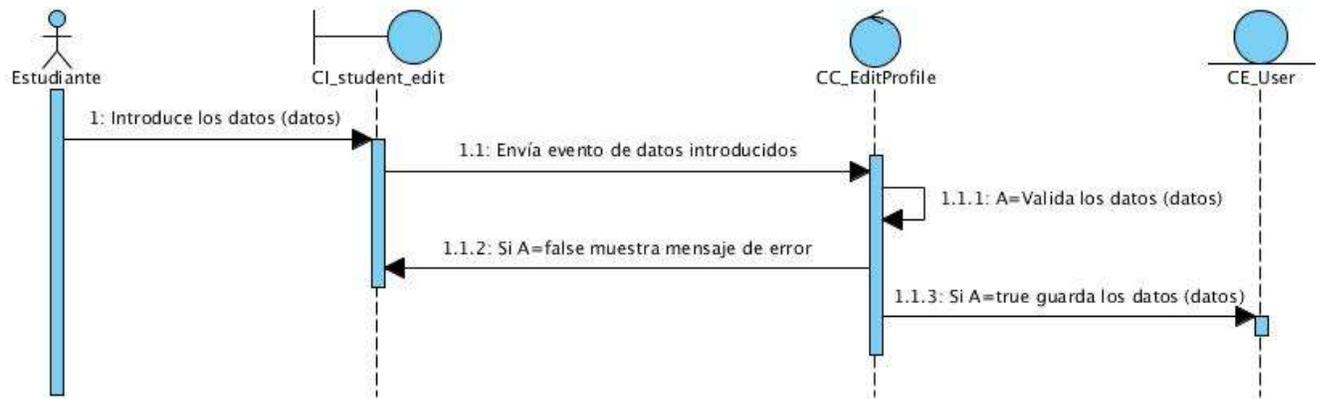


Figura 17 Diagrama de secuencia Crear cuenta

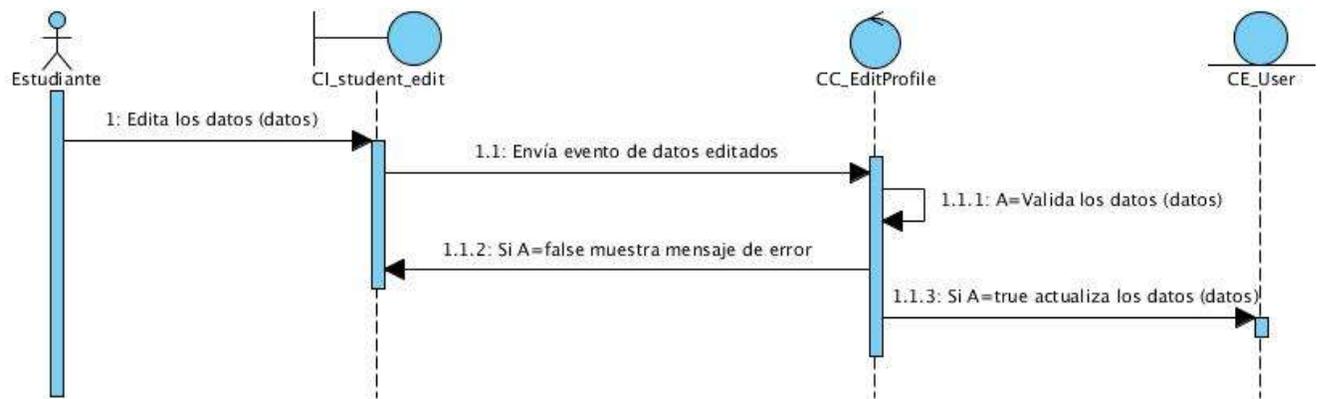


Figura 18 Diagrama de secuencia Editar perfil

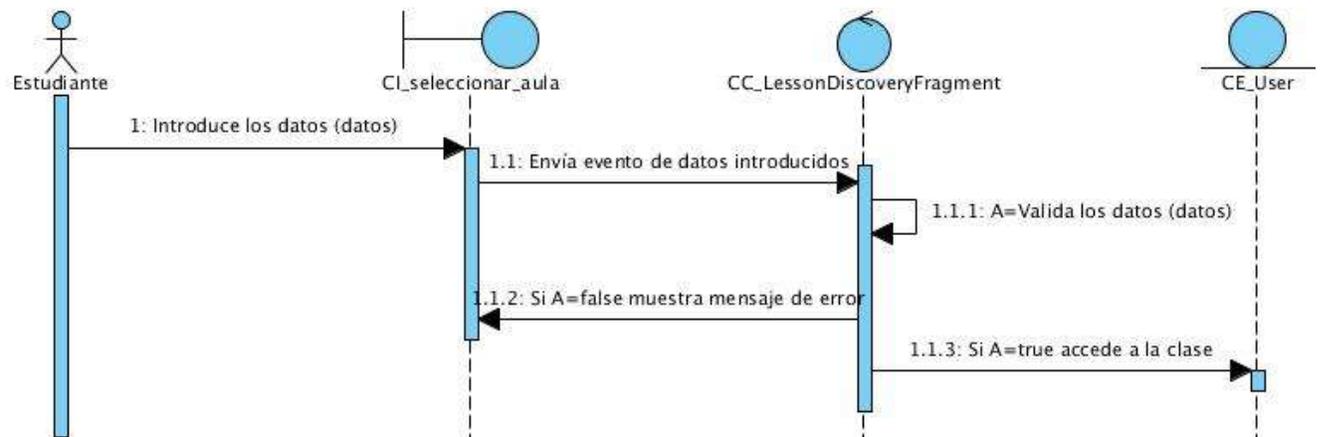


Figura 19 Diagrama de secuencia Acceder a la clase por autenticación

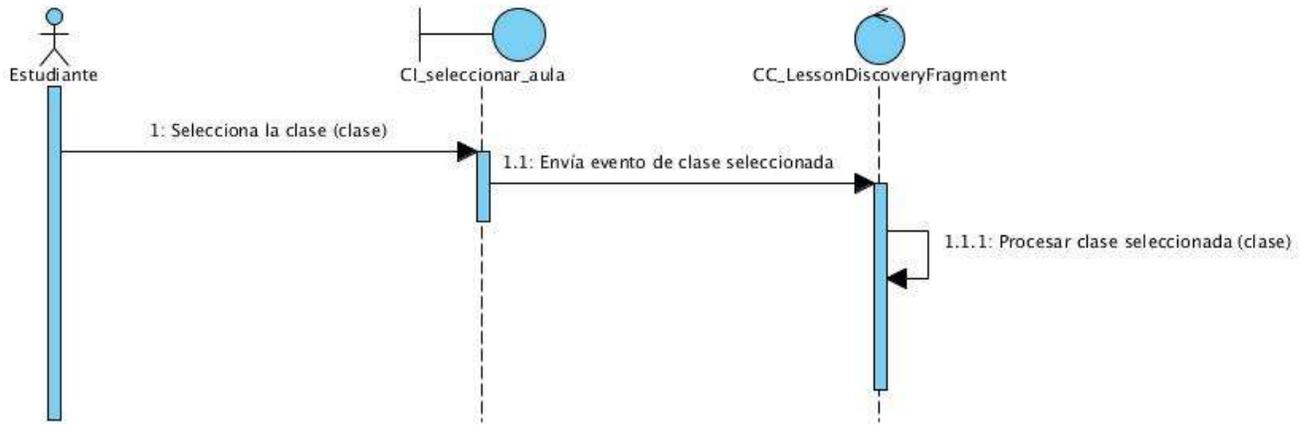


Figura 20 Diagrama de secuencia Seleccionar clase

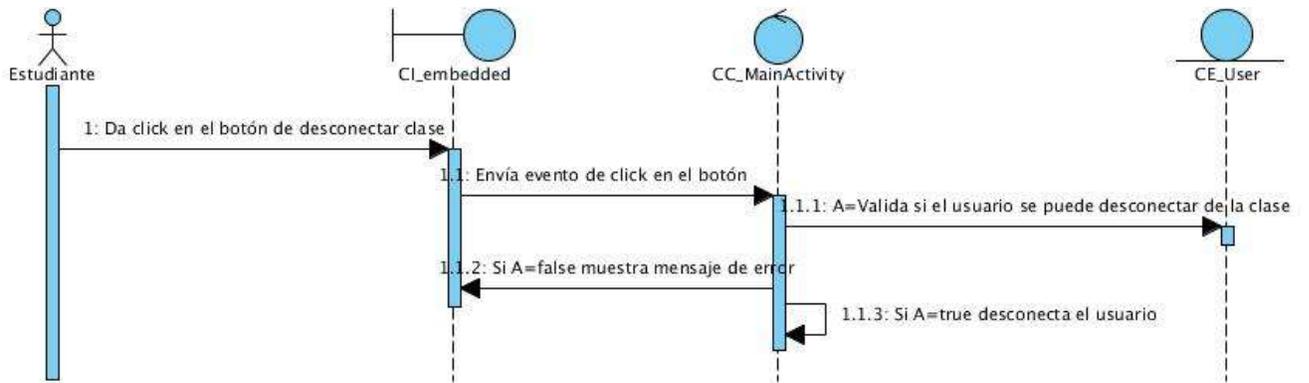


Figura 21 Diagrama de secuencia Cerrar sesión del estudiante

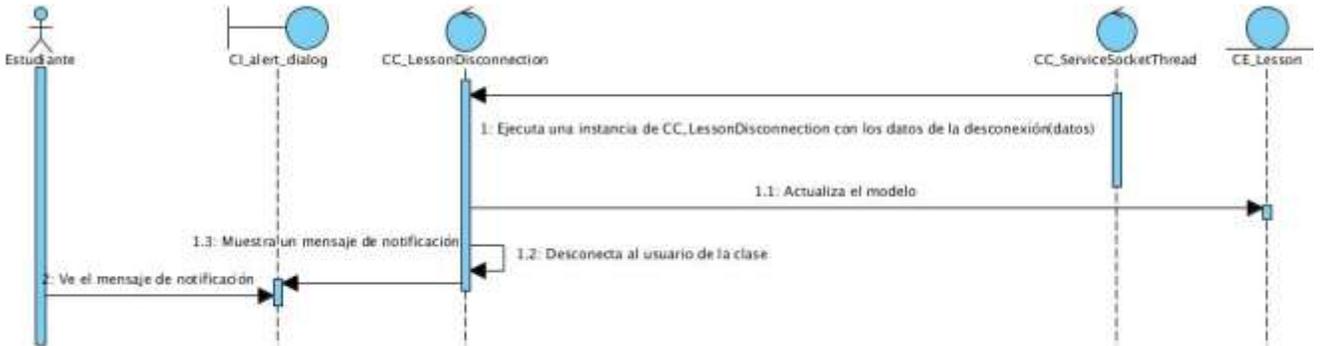


Figura 22 Diagrama de secuencia Desconectar estudiante de la clase

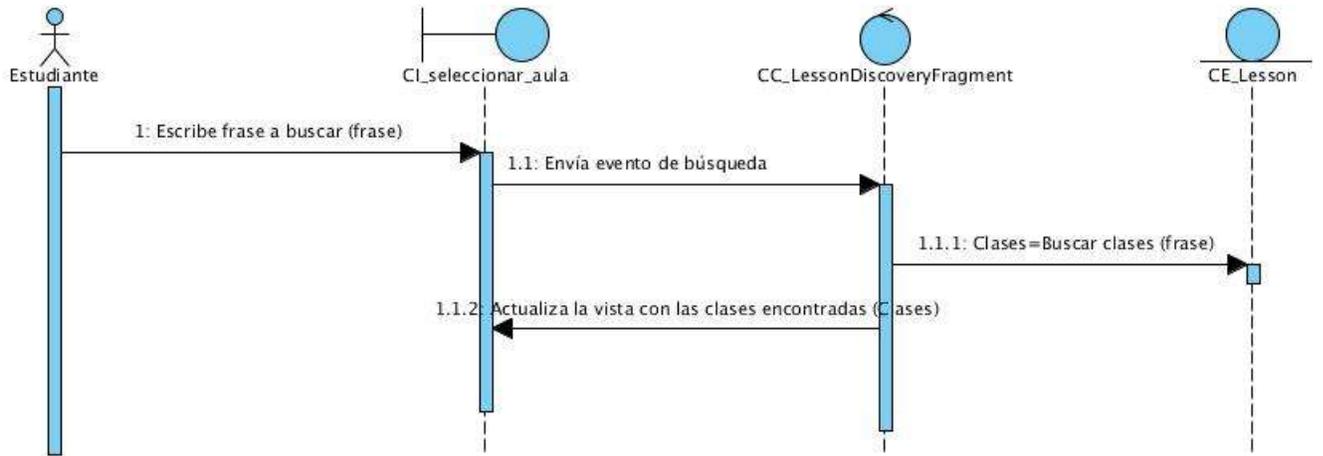


Figura 23 Diagrama de secuencia Buscar clase

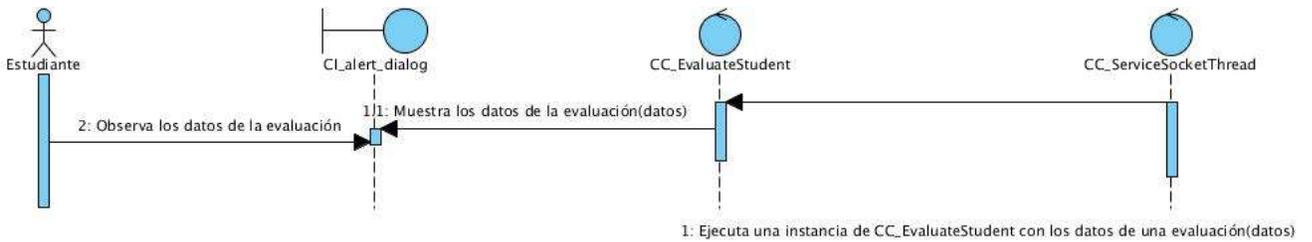


Figura 24 Diagrama de secuencia Notificar estudiante de la evaluación

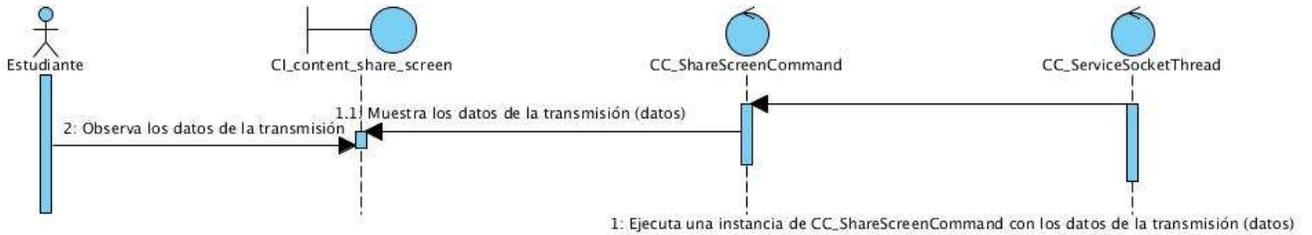


Figura 25 Diagrama de secuencia Transmitir pantalla del profesor en modo completa

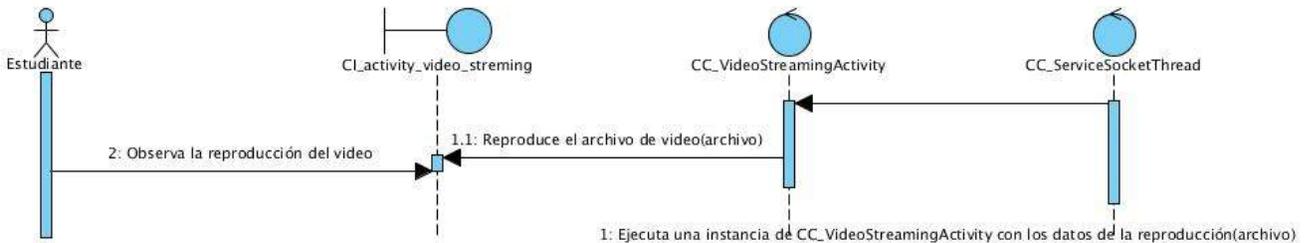


Figura 26 Diagrama de secuencia Reproducir archivo de video



Figura 27 Diagrama de secuencia Parar video

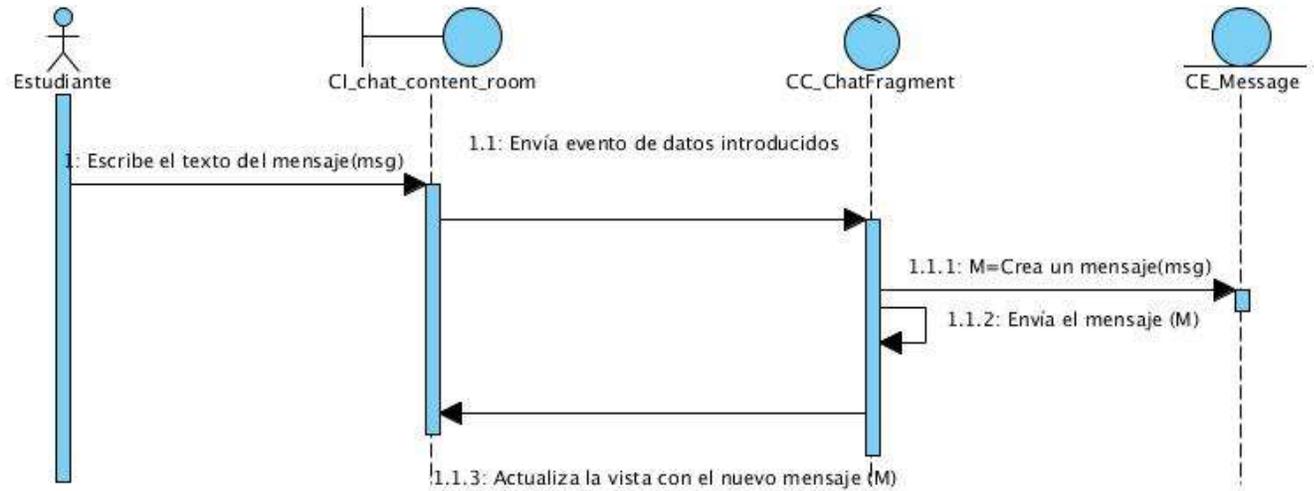


Figura 28 Diagrama de secuencia Establecer conversación con el profesor



Figura 29 Diagrama de secuencia Permitir conversaciones de los estudiantes

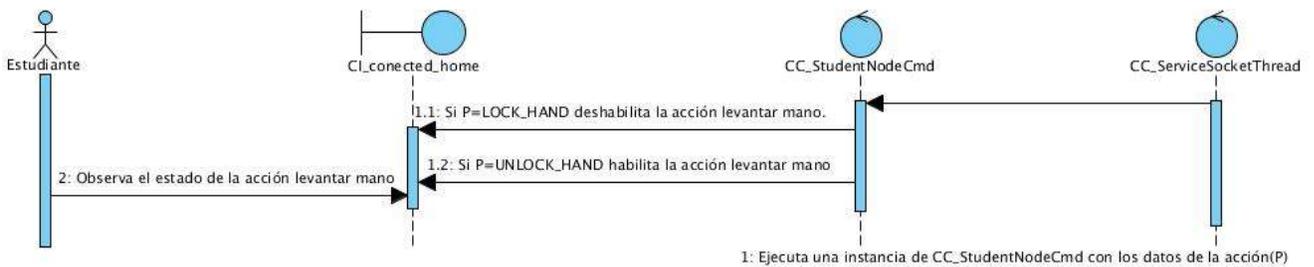


Figura 30 Diagrama de secuencia Permitir manos levantadas

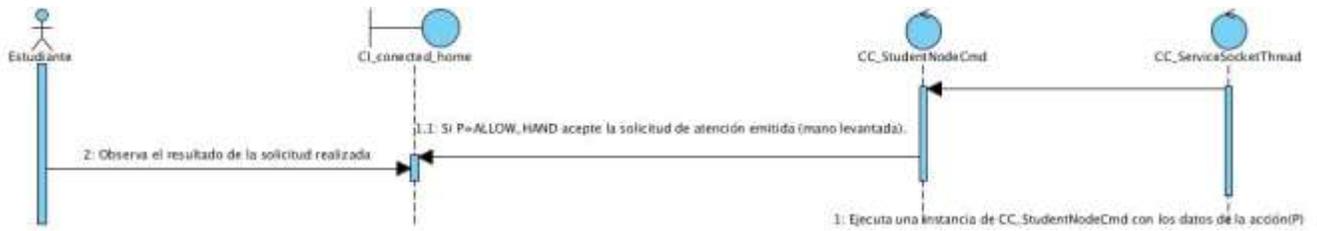


Figura 31 Diagrama de secuencia Aceptar manos levantadas



Figura 32 Diagrama de secuencia Denegar mano levantada

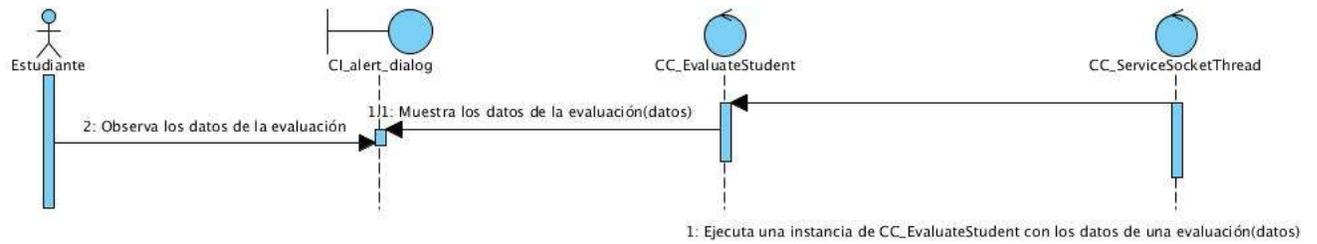


Figura 33 Diagrama de secuencia Notifica a estudiantes de la evaluación

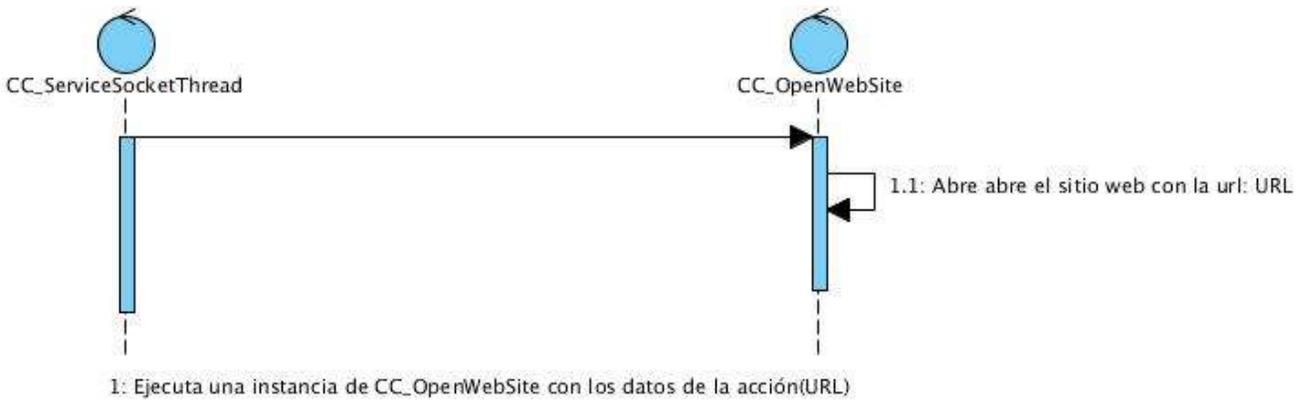


Figura 34 Diagrama de secuencia Abrir sitio web remoto

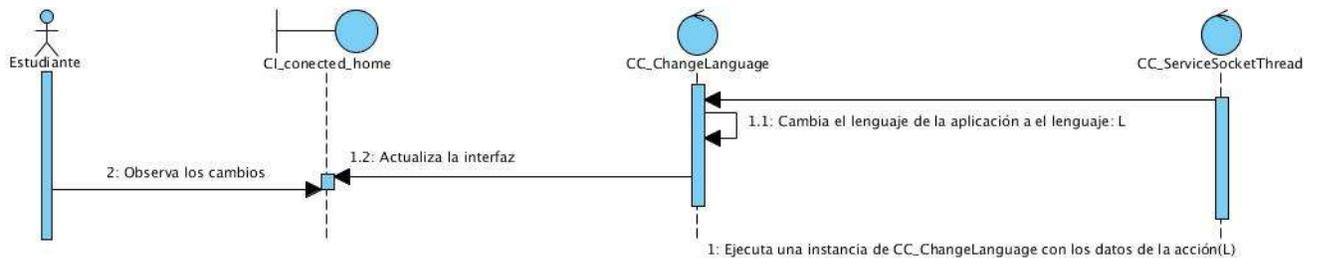


Figura 35 Diagrama de secuencia Configurar lenguaje del dispositivo remoto

Anexo 4: Descripción de las tablas de la Base de Datos

Tabla 29 Descripción de los atributos la tabla *tb_Lesson*

tb_Lesson		
Descripción: almacena los datos pertenecientes a la entidad Clase (Lesson).		
Atributo	Tipo	Descripción
id	integer (10)	Campo que contiene el identificador de la clase
idLesson	integer(10)	Campo que contiene el identificador global de la clase
ipDireccion	varchar(255)	Campo que almacena la ruta de IP
macDireccion	varchar(255)	Campo que almacena la MAC
lessonName	varchar(255)	Campo que almacena el nombre de la clase
capacityState	varchar(255)	Campo que almacena el estado de la capacidad de la clase
teacherName	varchar(255)	Campo que contiene el nombre del profesor
authenticationType	varchar(255)	Campo que contiene el tipo de autenticación
imageTeacher	blob	Campo que contiene la imagen del profesor

Tabla 30 Descripción de los atributos la tabla

tb_Group		
Descripción: almacena los datos pertenecientes a la entidad Grupo (Group).		
Atributo	Tipo	Descripción
id	integer (10)	Campo que contiene el identificador del grupo
name	varchar(255)	Campo que contiene el nombre del grupo
createdAt	date	Campo que contiene la fecha de creación del usuario
updatedAt	date	Campo que contiene la fecha de actualización del usuario
idLesson	integer (10)	Campo que almacena el id de la clase a la que pertenece

Tabla 31 Descripción de los atributos la tabla *tb_Message*

tb_Message		
Descripción: almacena los datos pertenecientes a la entidad Mensaje (Message).		
Atributo	Tipo	Descripción
id	integer (10)	Campo que contiene el identificador del mensaje

ANEXOS

message	varchar(255)	Campo que contiene el texto del mensaje
userName	varchar(255)	Campo que contiene el nombre del usuario
createdAt	date	Campo que contiene la fecha de creación del usuario
groupId	integer (10)	Campo que contiene el grupo al que pertenece el mensaje
userId	integer (10)	Campo que almacena el id del usuario al que pertenece

Tabla 32 Descripción de los atributos la tabla *tb_UserLesson*

tb_UserLesson		
Descripción: almacena los datos pertenecientes a la relación entre User y Lesson.		
Atributo	Tipo	Descripción
LessonId	integer (10)	Campo que contiene el identificador de la clase
UserId	integer (10)	Campo que contiene el identificador del usuario

Anexo 5: Diseño de casos de prueba

Tabla 33 Diseño de caso de prueba Crear una cuenta

Escenario	Descripción	Imagen	Nombre	Apellidos	Usuario	Correo	Contraseña	Confirmar contraseña	Respuesta del sistema	Flujo central
EC 1.1 Opción Adicionar usuario	El usuario selecciona la opción Adicionar usuario representada por el signo de + en la parte superior.								El sistema debe mostrar la interfaz Crear una cuenta permitiendo introducir los siguientes datos: En caso de ser estudiantes: Imagen (*) Nombre (*) Apellidos (*) Usuario (*) Correo (*) Contraseña (*) Confirmar contraseña Además, debe permitir seleccionar una de las siguientes opciones: Aceptar y Cancelar.	Entrar al sistema/Adicionar usuario
EC 1.2 Opción Aceptar	Una vez insertados los datos en caso de ser estudiante el usuario selecciona la opción Aceptar representada por un círculo con una palomita dentro.		V	V	V	V	V	V	El sistema entra al perfil del estudiante y le brinda la opción de buscar clases.	Entrar al sistema/Adicionar usuario/Aceptar
EC 1.3 Opción Cancelar	El usuario selecciona la opción Cancelar representada por una X.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	El sistema muestra la interfaz Entrar al sistema.	Entrar al sistema/Adicionar usuario/Cancelar
EC 1.4	Al usuario	I	V	V	V	V	V	V	<i>El sistema muestra al lado del</i>	Entrar al

ANEXOS

Campos incorrectos	introducir los datos existen campos incorrectos.	V	I	V	V	V	V	V	<i>campo incorrecto una alerta, al dar clic sobre la misma se muestra un mensaje de información especificando el error.</i>	sistema/Adicionar usuario/Aceptar
		V	V	I	V	V	V	V		
		V	V	V	I	V	V	V		
		V	V	V	V	I	V	V		
		V	V	V	V	V	I	V		
		V	V	V	V	V	V	I		
EC 1.5 Campos vacíos.	Al usuario introducir los datos deja campos vacíos.	I	V	V	V	V	V	V	<i>El sistema muestra el botón Aceptar deshabilitado.</i>	Entrar al sistema/Adicionar usuario/Aceptar
		V	I	V	V	V	V	V		
		V	V	I	V	V	V	V		
		V	V	V	I	V	V	V		
		V	V	V	V	I	V	V		
		V	V	V	V	V	I	V		
		V	V	V	V	V	V	I		

Tabla 34 Diseño de caso de prueba Editar perfil

Escenario	Descripción	Imagen	Nombre	Apellidos	Correo	Contraseña	Confirmar contraseña	Respuesta del sistema	Flujo central
EC 1.1 Opción Editar	El usuario selecciona la opción Editar perfil en caso de ser estudiante, en caso de ser profesor selecciona al estudiante que desea editar y selecciona la opción Editar.							El sistema debe mostrar la interfaz Editar perfil permitiendo introducir los siguientes datos: En caso de ser estudiantes: Imagen (*) Nombre (*) Apellidos (*) Correo (*) Contraseña (*) Confirmar contraseña Además, debe permitir seleccionar una de las siguientes opciones: Aceptar y Cancelar.	Clase/Perfil/Editar perfil (Android)
									Clase X/Administrar estudiantes/Seleccionar estudiantes/Editar
EC 1.2 Opción Aceptar	Una vez editados los datos los usuarios	V	V	V	V	V	V	En ambos casos el sistema actualiza la información del estudiante. En caso de ser profesor editará los datos	Clase/Perfil/Editar perfil/Aceptar (Android)

	seleccionan la opción Aceptar. En el caso de los estudiantes la opción Aceptar está representada por el ícono de la flecha.								únicamente en la clase donde se encuentra. En caso de ser estudiante editará los datos del perfil.	Clase X/Administrar estudiantes/Seleccionar estudiantes/Editar/Aceptar
EC 1.3 Opción Cancelar	Los usuarios seleccionan la opción Cancelar. En el caso de los estudiantes la opción Cancelar está representada por el ícono de la X.	N/A	N/A	N/A	N/A	N/A		N/A	El sistema muestra en caso de ser profesor la vista Administrar estudiantes . En caso de ser estudiante muestra la interfaz anterior.	Clase/Perfil/Editar perfil/Cancelar (Android)
										Clase X/Administrar estudiantes/Seleccionar estudiantes/Editar/Cancelar
EC 1.4 Campos incorrectos	Al usuario introducir los datos existen campos incorrectos.	I	V	V	V	V		V	<i>El sistema muestra en caso de ser estudiante al lado del campo incorrecto una alerta, al dar clic sobre la misma se muestra un mensaje de información especificando el error. En caso de ser profesor muestra un mensaje de información especificando el error.</i>	Clase/Perfil/Editar perfil/Aceptar (Android)
		V	I	V	V	V		V		Clase X/Administrar estudiantes/Seleccionar estudiantes/Editar/Aceptar
		V	V	V	V	I		V		
								I		
		V	V	V	V	V				
EC 1.5 Campos vacíos.	Al usuario introducir los datos deja campos vacíos.	I	V	V	V	V		V	<i>El sistema muestra en caso de ser estudiante al lado del campo vacío una alerta, al dar clic sobre la misma se muestra un mensaje de información especificando que los</i>	Clase/Perfil/Editar perfil/Aceptar (Android)
		V	I	V	V	V		V		
		V	V	I	V	V		V		
		V	V	V	I	V		V		
		V	V	V	V	I		V	Clase X/Administrar	

ANEXOS

EC 1.2 Opción Enviar mensajes	El usuario selecciona al grupo o la persona a la que desea enviar el mensaje, luego selecciona la opción Enviar o presiona el botón Enter.	V	El sistema agrega una pestaña en el área de mensajes con la nueva conversación.	Clase X/Chat/Seleccionar estudiante o grupo/Escribir mensaje/Enviar o Enter
EC 1.3 Opción Recibir mensaje	Una vez que el usuario envía el mensaje el destinatario recibirá el mismo.	V	El sistema muestra en el área del mensaje el nombre del estudiante o el profesor seguido del texto del mensaje. En el terminal de los estudiantes se mostrará en la pestaña General los mensajes enviados a todos, y en la pestaña Profesor los mensajes entre el estudiante y el profesor.	Clase X/Chat/Seleccionar estudiante o grupo/Escribir mensaje/Enviar o Enter/Recibir mensaje

Tabla 37 Diseño de caso de prueba Permitir conversaciones de los estudiantes

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Bloquear chat	El usuario selecciona el o los estudiantes a los cuales desea bloquear el chat y luego la opción Bloquear chat.	El sistema bloquea la conversación de los estudiantes seleccionados. En el terminal de los estudiantes se muestra en la pestaña de la conversación un mensaje de información y no permite a los estudiantes chatear. En el ordenador del profesor muestra en el recuadro perteneciente a los estudiantes bloqueados la opción de Desbloquear chat.	Clase X/Seleccionar estudiante/Bloquear chat
			Clase X/Listado de estudiantes panel derecho/Seleccionar estudiantes/Bloquear chat
EC 1.2 Opción Desbloquear chat	El usuario selecciona de los estudiantes que tienen el chat	El sistema desbloquea la conversación de los estudiantes seleccionados. En el terminal de los estudiantes se elimina en la pestaña de la conversación el mensaje de información y permite a los estudiantes chatear. En el ordenador del profesor muestra	Clase X/Seleccionar estudiante/Desbloquear chat

ANEXOS

bloqueado a los cuales desea desbloquear el chat y luego la opción Desbloquear chat.	en el recuadro perteneciente a los estudiantes bloqueados la opción Bloquear chat.	Clase X/Listado de estudiantes panel derecho/Seleccionar estudiantes/Desbloquear chat
--	--	---

Tabla 38 Diseño de caso de prueba Acceder a la clase por autenticación

Escenario	Descripción	Contraseña	Respuesta del sistema	Flujo central
EC 1.1 Opción Abrir clase	El usuario estudiante selecciona la clase a la que desea acceder y abre la clase.	N/A	El sistema muestra un formulario que le permite al usuario incluir el siguiente dato: (* Contraseña Una vez introducidos los datos el sistema debe permitir seleccionar una de las siguientes opciones: Aceptar (Representado por el ícono de la flecha) Cancelar (Representado por el ícono de la X)	Lista de clases/Seleccionar clase/Abrir clase
EC 1.2 Opción Aceptar	Una vez especificados los datos el usuario selecciona la opción Aceptar.	V	El sistema debe permitir que el estudiante acceda a la clase especificada por el profesor. Además muestra un mensaje de información al profesor especificando que el estudiante se ha conectado a la clase.	Lista de clases/Seleccionar clase/Abrir clase/Especificar datos/Aceptar
		Atcn34*		
EC 1.3 Opción de Cancelar.	El usuario selecciona la opción de Cancelar.	N/A	El sistema no permite al estudiante acceder a la clase y regresa a la vista anterior.	Lista de clases/Seleccionar clase/Abrir clase/Especificar datos/Cancelar
EC 1.4 Campos incorrectos	Al usuario insertar los datos para conectarse a la clase existen datos incorrectos.	I	El sistema notifica a los usuarios que hay campos incorrectos a través de un mensaje de información.	Lista de clases/Seleccionar clase/Abrir clase/Especificar datos/Aceptar
		V		
		I		

ANEXOS

EC 1.5 Campos vacíos	Al usuario insertar los datos para conectarse a la clase existen datos vacíos.	I	El sistema notifica a los usuarios que hay campos vacíos a través de un mensaje de información.	Lista de clases/Seleccionar clase/Abrir clase/Especificar datos/Aceptar
		V		
		I		

Tabla 39 Diseño de caso de prueba Buscar clase

Escenario	Descripción	Criterio	Respuesta del sistema	Flujo central
EC 1.1 Opción Buscar clase (usuario profesor)	Cuando el profesor accede al sistema se muestra el listado de clases creadas. Para buscar una clase tendrá en cuenta el siguiente criterio: Nombre de la clase.	V	El sistema muestra en la pantalla la o las clases que coincidan con el criterio especificado.	Mis clases/Insertar criterio
EC 1.2 No hay coincidencias con el criterio de búsqueda (usuario profesor)	Al insertar el criterio de búsqueda no hay coincidencias con las clases creadas.	V	El sistema mostrará la pantalla sin contenido y mostrará un mensaje de información.	Mis clases/Insertar criterio

Tabla 40 Diseño de caso de prueba Cerrar sesión de estudiante

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Cerrar sesión	El usuario selecciona la opción Salir.	El sistema debe enviar una notificación al profesor informando que el estudiante desea desconectarse de la clase siempre y cuando no esté activada la autenticación automática. El sistema debe permitir al profesor en el ícono Solicitud Aceptar o Rechazar la petición del estudiante. En caso de que esté activada la autenticación automática el sistema debe salir de la clase.	Clase X/Salir (Android)

ANEXOS

EC 1.2 Opción Aceptar	El usuario profesor en el ícono Solicitud acepta la petición del estudiante.	El sistema desconecta al estudiante de la clase.	Clase X/Solicitud/Aceptar
EC 1.3 Opción Rechazar	El usuario profesor en el ícono Solicitud rechaza la petición del estudiante.	El sistema no permite al estudiante desconectarse de la clase y envía un mensaje de información especificando que no puede desconectarse de la clase.	Clase X/Solicitud/Rechazar

Tabla 41 Diseño de caso de prueba Desconectar estudiante de la clase

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Desconectar	Una vez seleccionados el o los estudiantes a desconectar el usuario selecciona la opción Desconectar que se encuentra asociado a los mismos o la que se encuentra en las acciones masivas.	El sistema desconecta a los estudiantes seleccionados y muestra en los terminales de los mismos un mensaje de información especificando que el profesor lo ha desconectado de la clase.	Clase X/Seleccionar estudiantes/Desconectar

Tabla 42 Diseño de caso de prueba Seleccionar clase

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Seleccionar clase	El usuario selecciona la clase que desea habilitando el checkbok asociado a ella. En caso de que quiera seleccionar todas las clases debe ir habilitando el checkbook clase por clase o seleccionar el botón Seleccionar todo.	El sistema muestra el checkbok habilitado de la o las clases seleccionadas. Una vez seleccionada la o las clases el sistema debe mostrar las opciones masivas que se realizan sobre las mismas.	Mis clases/Seleccionar clases

Tabla 43 Diseño de caso de prueba Abrir sitio web remoto

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Sitio web	Una vez seleccionado el o los estudiantes a los cuales se les lanzará el sitio web el usuario selecciona la opción Sitio web.	El sistema muestra en el listado de sitios web. Permite al usuario seleccionar el sitio web a lanzar.	Clase X/Seleccionar estudiantes/Sitio web (opciones masivas)
			Clase X/Clic derecho encima del estudiante seleccionado/Control remoto/Lanzar sitio web
EC 1.2 Opción Lanzar remoto	Una vez que el usuario selecciona del listado de sitios web el sitio web que desea lanzar, selecciona la opción Lanzar remoto.	El sistema muestra el sitio web en el terminal de los estudiantes. Notifica al profesor con un mensaje de información.	Clase X/Seleccionar estudiantes/Sitio web (opciones masivas)/Seleccionar sitio web/Lanzar remoto
			Clase X/Clic derecho encima del estudiante seleccionado/Control remoto/Lanzar sitio web/Seleccionar sitio web/Lanzar remoto
EC 1.3 Opción Lanzar local	El usuario selecciona la opción Aceptar.	El sistema muestra el sitio web en el ordenador del profesor.	Clase X/Seleccionar estudiantes/Sitio web (opciones masivas)/Seleccionar sitio web/Lanzar local
			Clase X/Clic derecho encima del estudiante seleccionado/Control remoto/Lanzar sitio web/Seleccionar sitio web/Lanzar local

Tabla 44 Diseño de caso de prueba Configurar lenguaje del dispositivo remoto

Escenario	Descripción	Idioma	Respuesta del sistema	Flujo central
EC 1.1 Opción Idioma	Una vez seleccionado el estudiante, el usuario selecciona la opción Idioma que se encuentra en las acciones el lote.	N/A	El sistema debe permitir seleccionar el idioma que se mostrará en el terminal de los estudiantes. Además debe permitir seleccionar una de las siguientes opciones: Aceptar o Cancelar.	Clase X/Seleccionar estudiantes/Idioma

ANEXOS

EC 1.2 Opción Aceptar	El usuario selecciona el idioma y luego la opción Aceptar.	V	El sistema actualiza el lenguaje en el terminal remoto.	Clase X/Seleccionar estudiantes/Idioma/Aceptar
EC 1.3 Opción Cancelar	El usuario selecciona el idioma y luego la opción cancelar.	N/A	El sistema regresa a la vista anterior.	Clase X/Seleccionar estudiantes/Idioma/Cancelar

Tabla 45 Diseño de caso de prueba Iniciar aplicación remota por el profesor

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción lanzar aplicación	Una vez seleccionado el o los estudiantes a los cuales le lanzará la aplicación y la aplicación a lanzar, el usuario selecciona la opción Lanzar remoto.	El sistema lanza la aplicación en el terminal de los estudiantes seleccionados. El sistema muestra además en el ordenador del profesor un mensaje de información especificando que se ha lanzado la aplicación.	Clase X/Seleccionar estudiantes/Clic derecho/Control remoto/Lanzar aplicación/Seleccionar aplicación/Lanzar remoto Clase X/Seleccionar estudiantes/Aplicación//Seleccionar aplicación/Lanzar remoto

Tabla 46 Diseño de caso de prueba Notificar estudiantes de la evaluación

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Habilitar notificación a los estudiantes	El usuario habilita la opción Notificar a los estudiantes y luego pulsa la opción Evaluar. Si no habilita esta opción el estudiante no recibirá la notificación.	El sistema muestra en el terminal de los estudiantes un mensaje de información especificando en este la evaluación otorgada a los estudiantes y la observación realizada por el profesor en caso de que este lo especifique.	Clase X/Seleccionar estudiantes/Evaluar/Habilitar notificar al estudiante/Evaluar Clase X/Seleccionar estudiantes/Clic derecho/Evaluar/Habilitar notificar al estudiante/Evaluar

ANEXOS

Tabla 47 Diseño de caso de prueba Aceptar mano levantada

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Aceptar mano levantada	Una vez que el estudiante levanta la mano se muestra en la pestaña Solicitud una notificación que el estudiante ha levantado la mano. El usuario selecciona la opción representada por la palomita.	El sistema muestra en el terminal de los estudiantes un mensaje de información especificando que la solicitud de levantar la mano ha sido aceptada.	Clase X/Solicitud/Aceptar

Tabla 48 Diseño de caso de prueba Denegar mano levantada

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Denegar mano levantada	Una vez que el estudiante levanta la mano se muestra en la pestaña Solicitud una notificación que el estudiante ha levantado la mano. El usuario selecciona la opción representada por la X para denegar la petición.	El sistema muestra en el terminal de los estudiantes un mensaje de información especificando que la solicitud de levantar la mano ha sido denegada.	Clase X/Solicitud/Denegar

Tabla 49 Diseño de caso prueba Permitir manos levantadas

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Bloquear mano	Una vez que el usuario seleccione el o los estudiantes a los cuales bloqueará la opción Levantar mano, selecciona la opción Bloquear mano asociada al estudiante en caso de que	El sistema en el ordenador del profesor habilita la opción Desbloquear mano. El sistema mostrará en el terminal de los estudiantes la opción de levantar mano bloqueada y mostrará un mensaje de información especificando que la	Clase X/Seleccionar estudiantes/Bloquear mano Clase X/Seleccionar

ANEXOS

	sea uno y en caso de que sean más, selecciona la opción Bloquear mano que se encuentra en opciones masivas.	opción ha sido bloqueada.	estudiantes/Opciones masivas/Bloquear mano
EC 1.2 Opción Desbloquear mano	Una vez que el usuario seleccione el o los estudiantes a los cuales desbloqueará la opción Levantar mano, selecciona la opción Desbloquear mano asociada al estudiante en caso de que sea uno y en caso de que sean más, selecciona la opción Desbloquear mano que se encuentra en opciones masivas.	El sistema en el ordenador del profesor habilita la opción Bloquear mano. El sistema mostrará en el terminal de los estudiantes la opción de levantar mano desbloqueada.	Clase X/Seleccionar estudiantes/Desbloquear mano
			Clase X/Seleccionar estudiantes/Opciones masivas/Desbloquear mano

Tabla 50 Diseño de caso de prueba Transmitir pantalla modo completo

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Compartir pantalla	El usuario selecciona a los estudiantes a los cuales le compartirá la pantalla. Una vez seleccionados los estudiantes selecciona la opción Compartir pantalla. Una vez compartida la pantalla el usuario puede seleccionar otros estudiantes y de la misma forma compartir la pantalla.	El sistema muestra en el dispositivo de los estudiantes la pantalla del profesor y puede visualizar todo lo que este haga en la misma en tiempo real. Una vez compartida la pantalla los estudiantes no pueden realizar ninguna acción sobre la misma. En el ordenador del profesor se muestra sobre los estudiantes que tienen la pantalla compartida un ícono.	Clase X/Seleccionar estudiantes/Compartir pantalla
			Clase X/Clic derecho sobre el estudiante/ Transmitir/ Pantalla

Tabla 51 Diseño de caso de prueba Visualizar pantalla en miniatura del estudiante

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Visualizar pantalla	El usuario accede a la clase.	El sistema muestra en el ordenador del profesor el listado de estudiantes conectados, a través de pantallas en miniaturas. Cada pantalla mostrará un ícono con la foto del estudiante, el usuario, el estado de la batería así como las siguientes acciones: Bloquear pantalla Bloquear mano Bloquear chat Bloquear	Clase X

ANEXOS

		desconexión	
EC 1.2 Bloquear pantalla	El usuario selecciona la opción Bloquear pantalla.	Ver FORTES_DPA_DCP_Bloquear_pantalla	Clase X/Pasar el mouse sobre el estudiante/Bloquear pantalla
EC 1.3 Bloquear mano	El usuario selecciona la opción Bloquear mano.	Ver FORTES_DPA_DCP_Permitir_manos_levantadas	Clase X/Pasar el mouse sobre el estudiante/Bloquear mano
EC 1.4 Bloquear chat	El usuario selecciona la opción Bloquear chat.	Ver FORTES_DPA_DCP_Permitir_conversaciones_de_los_estudiantes	Clase X/Pasar el mouse sobre el estudiante/Bloquear chat
EC 1.5 Bloquear desconexión	El usuario selecciona la opción Bloquear desconexión.	Ver FORTES_DPA_DCP_Impedir_desconexión_de_la_clase	Clase X/Pasar el mouse sobre el estudiante/Bloquear desconexión

Tabla 52 Diseño de caso de prueba Parar video

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Detener	El usuario selecciona la opción Detener para detener la reproducción del video que se está ejecutando en la	El reproductor se detiene y muestra una imagen con el logo de la aplicación y el texto Reproductor de video.	Clase X/General/Compartir video/Seleccionar video/Reproducir/Detener

	clase.		
--	--------	--	--

Tabla 53 Diseño de caso de prueba Reproducir archivo de video

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Reproducir video	Una vez seleccionado el video a reproducir el usuario selecciona la opción Reproducir.	El sistema permite al profesor reproducir un archivo de video en su pantalla y en los dispositivos de los estudiantes conectados a la clase.	Clase X/General/Compartir video/Seleccionar video/Reproducir
EC 1.2 Transmitir video	En caso de que el profesor desee transmitir el video en el dispositivo de un estudiante específico, selecciona el mismo y luego selecciona la opción Transmitir.	El sistema permite al profesor reproducir el video en el dispositivo del estudiante seleccionado.	Clase X/Seleccionar estudiante/Clic derecho/Transmitir/Video/Seleccionar video/Reproducir

Anexo 6: Acta de aceptación

ACTA DE ACEPTACIÓN

Por este medio el **Centro de Tecnologías para la Formación**, en forma abreviada **FORTES**, en cumplimiento con lo descrito en el contrato No. 526/2014_11 y en función de la ejecución del proyecto **Desarrollo del producto ATcnea** para **GEDEME** presenta al cliente **Empresa Industrial para la Informática, Electrónica y las Comunicaciones**, en forma abreviada **GEDEME**, los siguientes productos:

___ Producto Atcnea
 ___ Manual de usuario

La Parte cliente, luego de haber revisado los productos de trabajo determina que se aceptan. Y para que así conste:

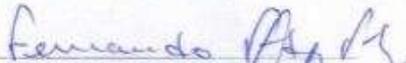
Parte Suministradora	Parte Cliente
Nombre y apellidos: Odenys Almora Rodríguez	Nombre y apellidos: René Cano Díaz
 Cargo: Jefe de proyecto	 Cargo: Director General
Representante Parte Suministradora	
Nombre y Apellidos: Yerandy Manso Guerra	
Cargo: Director de centro	
Firma: 	
Observador independiente	
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;">  <p>UCI Universidad de los Andes 1970-2014</p> </div> <div style="text-align: right;"> <p>Facultad 4 Dirección Centro Fortes</p> </div> </div>	

Figura 36 Acta de Aceptación