

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 4



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

**Versión 2.0 de la aplicación móvil para la plataforma educativa
XAUCE ZERA**

Autor: Raimary Sanjudo Cabrera

Tutor(es): Ing. Sandy Nuñez Padrón

Ing. Yaritza Bárbara González Ramírez

Ing. Guillermo González Jiménez

La Habana, junio del 2018

“Año 60 de la Revolución”

Declaración de Autoría

Declaro ser el único autor del presente Trabajo de Diploma y autorizo a la Universidad de las Ciencias Informáticas a realizar uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2018.

Firma del autor:

Raimary Sanjudo Cabrera.

Firma de los tutores:

Ing. Sandy Nuñez Padrón

Ing. Yaritza Bárbara González Ramírez

Ing. Guillermo González Jiménez

Agradecimientos

A mi familia por su cariño, en especial a mi madre, mi padre y mi hermano, por ser los principales educadores de mi vida.

A mi novio Alex por ser mi compañero y amigo en estos 5 años de universidad.

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

A los profesores, que me dedicaron parte de su tiempo para ayudarme y aconsejarme.

A mis amigos que me apoyaron siempre, y fueron mis hermanos durante estos 5 años y estuvieron cuando de veras los necesité.

A todas las personas que de una forma u otra contribuyeron a que hiciera realidad este gran sueño, aunque sus nombres no aparezcan textualmente, sí están en mis agradecimientos diarios.

A todos, gracias.

Dedicatoria

A mis padres porque sin ellos nada de esto hubiera sido posible si no fuera por su dedicación e interés por mi superación y por todo el sacrificio que han tenido que hacer para que hoy finalmente pudiera ser Ingeniera.

Disímiles son las herramientas y aplicaciones desarrolladas con el objetivo de apoyar el proceso de enseñanza-aprendizaje destacándose las plataformas educativas, las cuales son muy utilizadas actualmente en cada una de las instituciones. Las tecnologías móviles por su parte, ofrecen nuevas oportunidades para el aprendizaje y su utilización en la educación ha ofrecido varias ventajas tanto para los profesores como para los estudiantes. Dado el auge del uso de los dispositivos móviles, con el paso del tiempo, se ha incrementado la cantidad de aplicaciones educativas desarrolladas para estos, lo cual ha posibilitado el nacimiento de un nuevo concepto: “aprendizaje móvil (m-learning)”. Actualmente el centro de Tecnologías para la Formación FORTES cuenta con la Plataforma educativa XAUCE ZERA para guiar el proceso de enseñanza-aprendizaje en instituciones de cualquier nivel de enseñanza. Esta última, cuenta con una versión móvil que no soporta funcionalidades tales como el proceso de notificaciones, el proceso de calificaciones, así como la realización de tareas y la interacción con los cursos de manera off-line. La presente investigación tiene como objetivo desarrollar la versión 2.0 de la aplicación móvil de la Plataforma educativa XAUCE ZERA para la interacción de forma off-line con los contenidos de la plataforma desde un dispositivo móvil con sistema operativo Android. Se describe la problemática inicial que dio origen a la solución, la cual fue desarrollada mediante la metodología AUP-UCI, se realizaron los flujos de Modelado, Implementación y Prueba, indicados por la metodología elegida, obteniéndose un sistema que cumple con los requerimientos establecidos.

Palabras Clave: aplicación móvil, off-line, XAUCE ZERA.

Índice de Contenido

Introducción	1
Capítulo 1. Fundamentación teórica	6
1 Introducción.....	6
1.1. Conceptos y definiciones asociados al problema de investigación	6
1.1.1 Dispositivos móviles	6
1.1.2 Android.....	7
1.1.2.1 Arquitectura Android	7
1.1.3 M-Learning	8
1.2 Versión 1.0 de la aplicación móvil de la Plataforma educativa XAUCE ZERA	9
1.3 Estudio de aplicaciones similares.....	10
1.4 Ambiente de desarrollo	11
1.4.1 Tecnologías en el desarrollo de aplicaciones para el sistema operativo Android	11
1.4.2 Lenguajes de desarrollo	13
1.4.3 Herramientas de Desarrollo de Software	14
1.5 Sistema Gestor de Base de Datos.....	15
1.6 Metodologías de desarrollo de software	16
1.6.1 Metodologías tradicionales.....	17
1.6.2 Metodologías de desarrollo de software ágiles.....	17
1.6.2.1 Metodología AUP	17
1.7 Conclusiones parciales	19
Capítulo 2. Propuesta de solución	17
Introducción.....	17
2.1-Descripción de la propuesta de solución	17
2.2-Modelo de dominio	17
2.3-Especificación de requisitos de software	18
2.3.1 Requisitos funcionales	19
2.3.2 Requisitos no funcionales	21
2.4-Descripción de requisitos funcionales.....	22
2.5-Patrón Arquitectónico	24

Índice de Contenido

2.6-Patrones de diseño	25
2.6.1-Patrones GRASP	25
2.6.2-Patrones GOF	26
2.7-Modelo de diseño	26
2.8- Diagrama de despliegue.....	27
2.9 Diseño de la Base de Datos	28
2.10-Conclusiones parciales.....	29
Capítulo 3. Implementación y Prueba	30
Introducción.....	30
3.1 Modelo de implementación.....	30
3.1.1 Estándar de codificación.....	30
3.1.2 Diagrama de componente (DCOM)	31
3.2 Pruebas de software.....	32
3.2.1 Pruebas de sistema	33
3.2.2 Pruebas de integración.....	37
3.2.3 Pruebas de aceptación	37
3.3 Conclusiones del capítulo	38
Conclusiones	36
Bibliografía.....	38

Índice de Tablas

Tabla 1: Fases de AUP y AUP-UCI.....	18
Tabla 2: HU Listar tareas de forma off-line.....	23
Tabla 3: Descripción de la tabla Activity.....	29
Tabla 4: Caso de prueba perteneciente al RF Listar Notificaciones	35
Tabla 5: Resultados obtenidos en las pruebas.....	36
Tabla 6: Clasificación de las No Conformidades (NC) por tipo de error.....	36
Tabla 7: HU Mostrar actividad de forma off-line	47
Tabla 8: HU Responder la actividad de forma off-line	48
Tabla 9: HU Buscar actividades de forma off-line.....	49
Tabla 10: HU Listar notificaciones.....	51
Tabla 11: HU Visualizar notificación.....	53
Tabla 12: HU Eliminar notificación	54
Tabla 13: HU Filtrar notificaciones	55
Tabla 14: HU Buscar notificación	57
Tabla 15: HU Marcar notificación como no leída	58
Tabla 16: HU Marcar notificación como leída.....	60
Tabla 17: HU Cerrar notificación	62
Tabla 18: HU Comentar en el foro	63
Tabla 19: HU Visualizar comentarios del foro	65
Tabla 20: HU Eliminar comentarios del foro.....	66
Tabla 21: HU Listar temas de los foros	67
Tabla 22: HU Listar foros	68
Tabla 23: HU Buscar foros.....	69
Tabla 24: HU Descargar un curso.....	71
Tabla 25: HU Mostrar contenidos de un curso de manera off-line.....	72
Tabla 26: HU Cambiar idioma	74
Tabla 27: Caso de prueba perteneciente al RF Visualizar Notificaciones.....	76
Tabla 28: Caso de prueba perteneciente al RF Eliminar Notificación.....	77
Tabla 29: Caso de prueba perteneciente al RF Buscar Notificación.....	78
Tabla 30: Caso de prueba perteneciente al RF Marcar Notificación como no leída.....	79
Tabla 31: Caso de prueba perteneciente al RF Marcar Notificación como leída.	80
Tabla 32: Caso de prueba perteneciente al RF Comentar en el foro.....	81
Tabla 33: Caso de prueba perteneciente al RF Visualizar Comentarios del Foro.....	82
Tabla 34: Caso de prueba perteneciente al RF Eliminar Comentario del Foro.....	82
Tabla 35: Caso de prueba perteneciente al RF Buscar Foro.....	84
Tabla 36: Caso de prueba perteneciente al RF Listar Foro.....	84

Índice de Tablas

Tabla 37: Caso de prueba Listar tareas de forma off-line.....	85
Tabla 38: Caso de prueba Mostrar tareas de forma off-line.	85
Tabla 39: Caso de prueba Responder tareas de forma off-line.	86
Tabla 40: Caso de prueba Buscar tareas de forma off-line.	87

Índice de Figuras

Fig. 1 Diagrama del modelo de dominio	18
Fig. 2 MVC para Android.....	25
Fig. 3 Diagrama de clases del diseño del Paquete curso	27
Fig. 4 Diagrama de Despliegue.....	28
Fig. 5 Diagrama entidad-relación	28
Fig. 6 Diagrama de Componentes del Sistema	32
Fig. 7 Comportamiento de las no conformidades en cada iteración	36
Fig. 8 Diagrama de Clases del Diseño del Paquete Foros	75
Fig. 9 Diagrama de Clases de Diseño del Paquete Notificaciones	76

Introducción

El proceso de enseñanza-aprendizaje desde sus comienzos hasta la actualidad ha sido de los procesos naturales de la humanidad, que más transformaciones ha sufrido. Es innegable que estas transformaciones han tenido mayor impacto gracias a la introducción de las Tecnologías de la Información y la Comunicación (TIC) en dicho proceso, permitiendo conseguir nuevas realidades comunicativas, girando en torno a tres medios básicos: la informática, la microelectrónica y las telecomunicaciones. (1)

A lo largo de este proceso las TIC han evolucionado en sí mismas, jugando un papel relevante dando surgimiento a novedosas formas de comunicación que han revolucionado las diferentes modalidades de enseñanza. Actualmente se habla de un mundo digital y se dice que los jóvenes son nativos digitales, desde que nacen tienen un contacto directo con la tecnología formando esta última, parte cotidiana de su vida y por ende aprender a través de ellas es visto de manera natural. De ahí que dotar las aulas con la tecnología necesaria para que los estudiantes aprendan desde temprana edad, es una idea que cobra cada vez mayor fuerza en el mundo. (2)

Disímiles son las herramientas y aplicaciones desarrolladas con el objetivo de apoyar el proceso de enseñanza-aprendizaje destacándose las plataformas educativas, las cuales son muy utilizadas para llevar a cabo la educación presencial y semi-presencial, así como para proveer soporte a una reinventada educación totalmente a distancia. Las tecnologías móviles por su parte ofrecen nuevas oportunidades para el aprendizaje, extendiéndolo más allá de la tradicional "aula de clases". Estos últimos, con el paso del tiempo, han aumentado su nivel de utilidad y funcionalidad, pues hoy en día, se pretende lograr independizarse de los medios computacionales tradicionales. Estos dispositivos se pueden definir como micro-ordenadores que son lo suficientemente ligeros como para ser transportados por una persona, y que disponen de la capacidad de batería suficiente como para poder funcionar de forma autónoma. (3)

La influencia de la utilización de estos dispositivos en la educación ha ofrecido varias ventajas, muchas de las cuales requieren replantear metodologías y estándares; principalmente en la forma en que los profesores se comunican con sus alumnos. Algunas de estas ventajas son (4):

- Permite la comunicación en tiempo real entre alumnos, profesores y directivos: Como es el caso del uso de redes sociales y aplicaciones de mensajería instantánea entre alumnos 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: No es necesario que alumnos 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: Sea por medio de realizar investigaciones, aprender y utilizar nuevas tecnologías, crear documentos y compartirlos.

Dado el auge y las múltiples ventajas del uso de los dispositivos móviles, con el paso del tiempo, se ha incrementado la cantidad de aplicaciones educativas desarrolladas para estos, jugando así un rol importante en la formación y elevación del nivel cultural de los usuarios, lo cual ha posibilitado el nacimiento de un nuevo concepto: “aprendizaje móvil (*m-learning*)”. El *m-learning* se basa fundamentalmente en el aprovechamiento de las tecnologías móviles como base del proceso de aprendizaje. Por tanto, es un proceso de enseñanza- aprendizaje que tiene lugar en distintos contextos (virtuales o físicos) y/o haciendo uso de tecnologías móviles. (5)

El término “tecnología móvil” se vincula al ámbito de las comunicaciones móviles y describe las capacidades de comunicación electrónica de forma no cableada o fija entre puntos remotos y en movimiento. Las tecnologías móviles propician que el usuario-estudiante no precise estar en un lugar predeterminado para aprender y constituyen un paso hacia el aprendizaje en cualquier momento y en cualquier lugar. Aunque algunos autores sostienen que el *m-learning* o aprendizaje móvil es una forma de *e-learning*, y no constituye una nueva forma de aprendizaje en sí misma, lo cierto es que esta modalidad de aprendizaje en la que intervienen tecnologías móviles supone un progreso en los sistemas de aprendizaje *online*. (5)

Lo anteriormente dicho se puede avalar teniendo en cuenta algunas de las características tecnológicas asociadas al *m-learning* (5):

- **Portabilidad**, debido al pequeño tamaño de los dispositivos.
- **Inmediatez y conectividad** mediante redes inalámbricas.
- **Ubicuidad**, ya que se libera el aprendizaje de barreras espaciales o temporales.
- **Adaptabilidad** de servicios, aplicaciones e interfaces a las necesidades del usuario.

En Cuba, con la llegada de las zonas *wifi* se ha abierto la puerta para la aplicación de este método de enseñanza; cada día aumenta exponencialmente la cantidad de usuarios interesados en la adquisición de esta importante vía de comunicación. En el país, además, se han desarrollado un gran número de herramientas y aplicaciones con el fin de apoyar la educación en los diferentes niveles de enseñanza. En cuanto al desarrollo de software en Cuba, la Universidad de las Ciencias Informáticas (UCI) juega un papel fundamental contando con 14 centros de producción con diferentes líneas de investigación. Es

válido destacar el Centro de Tecnologías para la Formación (FORTES), el cual se encarga de desarrollar tecnologías que permitan ofrecer servicios y productos para la implementación de soluciones de formación.

Uno de los principales productos desarrollados por este centro es la Plataforma educativa XAUCE ZERA, la cual es capaz de guiar el proceso de enseñanza-aprendizaje en instituciones de cualquier nivel de enseñanza a partir de un conjunto poderoso de herramientas centradas en los aprendices y ambientes de aprendizaje colaborativo, que le dan poder, tanto a la enseñanza como al aprendizaje. Esta última cuenta actualmente con una versión para dispositivos móviles debido a que la portabilidad, la movilidad y el hecho de que en muchos de los casos estos dispositivos se han vuelto imprescindibles para los usuarios; fueron aspectos importantes que se tuvieron en cuenta para que el desarrollo del *m-learning* y que constituyera una estrategia a explotar por la plataforma.

A pesar de los múltiples beneficios que traería el uso de dicha aplicación de la Plataforma educativa XAUCE ZERA para dispositivos móviles en la educación, se puede apreciar que la aplicación no soporta elementos tales como el proceso de notificaciones, ya sea del sistema en general o de un curso en específico; la realización de tareas de manera *off-line*; la participación en los foros de forma *on-line* y la interacción con los cursos de forma *off-line*. Además, la interfaz de la versión actual no es la más acertada a las exigencias de los usuarios. Por las razones antes mencionadas, se hace necesario suplir las deficiencias detectadas en dicha aplicación para que la misma pueda ser utilizada como parte del proceso de enseñanza-aprendizaje.

Luego de analizar la problemática anteriormente descrita se identifica como **problema de la investigación**: ¿Cómo interactuar con los contenidos de la Plataforma educativa XAUCE ZERA de forma *off-line* desde un dispositivo móvil con sistema operativo Android?

La investigación en curso enmarca como **objeto de estudio**: Aplicaciones para dispositivos móviles.

Campo de acción: Aplicación para dispositivos móviles vinculada a la Plataforma educativa XAUCE ZERA.

Para dar solución al problema de la investigación se define como **objetivo general**: Desarrollar la versión 2.0 de la aplicación móvil de la Plataforma educativa XAUCE ZERA para la interacción de forma *off-line* con los contenidos de la plataforma desde un dispositivo móvil con sistema operativo Android. Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

- Analizar la versión 1.0 de la aplicación móvil de la Plataforma educativa XAUCE ZERA para la identificación de sus limitaciones y deficiencias que impiden la interacción de forma *off-line* con los contenidos de la plataforma.

- Identificar las funcionalidades necesarias para la interacción de forma off-line con los contenidos de la Plataforma educativa XAUCE ZERA desde un dispositivo móvil con sistema operativo Android a partir del estudio de sistemas similares y de las exigencias del cliente.
- Implementar la propuesta de solución para la interacción de forma off-line con los contenidos de la Plataforma educativa XAUCE ZERA desde un dispositivo móvil con sistema operativo Android a partir de la generación de los artefactos propuestos por la metodología de desarrollo de software seleccionada.
- Garantizar la correcta interacción de forma off-line con los contenidos de la Plataforma educativa XAUCE ZERA desde la versión 2.0 de su aplicación móvil.

Posibles resultados:

Versión 2.0 de la aplicación para dispositivos móviles de la Plataforma educativa XAUCE ZERA con su correspondiente documentación y dando solución a las necesidades de los usuarios que emplean dicha plataforma a través de dispositivos móviles con sistema operativo Android.

Para llevar a cabo la investigación se emplearon diferentes métodos científicos tanto del nivel teórico como del empírico, los cuales se enuncian a continuación:

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.
- **Analítico-sintético:** Se empleó en el estudio de las tecnologías que se emplearán para el desarrollo de la propuesta y en el análisis de la documentación relacionada con el objeto de estudio, así como en la revisión bibliográfica.
- **Análisis documental:** Fue utilizado en la consulta de la literatura especializada, para extraer la información necesaria que responda a las características distintivas del problema.

Métodos empíricos:

- **Observación:** Permitió estudiar de cerca el objeto de la investigación, las acciones, causas y consecuencias logrando conocer la esencia del problema planteado, analizando desde varios puntos de vista la propuesta de solución y otras soluciones existentes e identificando que está hecho y que falta por hacer.

La presente investigación cuenta con la siguiente estructura capitular:

Capítulo I: Fundamentación teórica: En este capítulo se hace referencia a todas las herramientas, tecnologías, lenguajes y metodologías de desarrollo seleccionadas para darle solución al problema planteado; aborda todos los elementos teóricos que sustentan la investigación: conceptos asociados al

dominio del problema, tipos de competencias, aportes de las competencias a los procesos de formación y la evaluación de las competencias.

Capítulo II: Propuesta de solución: En este capítulo se realiza una descripción de todas las características de las funcionalidades a desarrollar. En el mismo se especifican los requisitos funcionales y requisitos no funcionales, la descripción del modelo de dominio y se describen las Historias de Usuario.

Capítulo III: Implementación y pruebas: Se define el modelo de análisis, se modelan los diagramas de clases de análisis, diagrama entidad relación de la base de datos, se describen todas las tablas que darán solución al problema planteado y se realiza el diagrama de despliegue.

Capítulo 1. Fundamentación teórica

Capítulo 1. Fundamentación teórica

1 Introducción

El presente capítulo proyecta los principales conceptos asociados al desarrollo de aplicaciones para dispositivos móviles, el objeto de estudio definido en la investigación, destacándose las principales características de los elementos asociados a la temática a investigar, así como el entorno nacional e internacional donde se desarrollan. Se describen, además, las herramientas, tecnologías y metodologías a utilizar para brindar solución al problema planteado.

1.1. Conceptos y definiciones asociados al problema de investigación

1.1.1 Dispositivos móviles

Los dispositivos móviles se pueden definir como aquellos micro-ordenadores que son lo suficientemente ligeros como para ser transportados por una persona, y que disponen de la capacidad de batería suficiente como para poder funcionar de forma autónoma. Normalmente, son versiones limitadas en prestaciones, y por tanto en funcionalidades, de los ordenadores portátiles o de sobremesa. (6)

Por su parte Arturo Baz plantea que un dispositivo móvil es un aparato de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. De acuerdo con esta definición existen multitud de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores de Sistema de Posicionamiento Global, pasando por los teléfonos móviles o los *Tablets*. (7)

Tipos de dispositivos móviles

Existen numerosos tipos de dispositivos móviles, entre los que podemos mencionar (7):

Celulares: Los celulares son dispositivos electrónicos de comunicación, normalmente de diseño reducido y sugerente, que tiene la misma funcionalidad que cualquier teléfono de línea fija. Es portable e inalámbrico, lo que significa que no necesita de ningún tipo de cableado para llevar a cabo la conexión a la red telefónica.

Además de ser capaz de realizar llamadas como cualquier otro teléfono convencional, un celular más o menos moderno suele incorporar un conjunto de funciones adicionales, tales como mensajería instantánea, agenda y juegos, que aumentan la potencialidad de utilización de estos dispositivos. Su manejo suele ser sencillo, ya sea a través de la escritura sobre una pantalla táctil, o de un pequeño teclado.

Smartphone: Un *Smartphone* es un teléfono móvil construido sobre una plataforma informática móvil, con mayor capacidad de almacenamiento de información y mejor conectividad que un celular. El término

Capítulo 1. Fundamentación teórica

inteligente hace referencia a la capacidad de usarse como un ordenador de bolsillo, llegando incluso a remplazar a un ordenador personal en algunos casos.

Estos *Smartphone* cuentan con cámaras, Sistema de Posicionamiento Global (GPS por sus siglas en inglés) y *bluetooth*. Les dan soporte a redes *wifi*, 3G12 y 4G12 para el acceso a Internet. Permiten el almacenamiento de hasta 32 Gb de información y sus pantallas generalmente no exceden las 5 pulgadas. A algunos dispositivos se le pueden introducir tarjetas de memorias extraíbles (microSD). Los sistemas operativos más comunes que utilizan para su funcionamiento son: *Android*, *iOS*, *Symbian*, *Blackberry RIM* y *Windows Phone*.

Tablet: Un *tablet* (tableta en español) es un tipo de computadora portátil, de mayor tamaño que un teléfono inteligente o un PDA, el cual cuenta con una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o, sin necesidad de teclado físico ni *mouse*.

Técnicamente, los *tablets* se ven dotados de la mayoría de las características de los *smartphones*. La mayor diferencia está marcada por el tamaño, ya que sus pantallas miden desde 7 pulgadas, hasta más de 15 en algunos casos. Su conectividad es mayor que la de los teléfonos inteligentes, al igual que su velocidad de procesamiento. A pesar de que estos no están provistos de líneas para efectuar llamadas telefónicas, cuentan con servicios de *internet* como videollamadas y *chats* que permiten la comunicación entre varios usuarios.

1.1.2 Android

Android es un sistema operativo y una plataforma de *software*, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), *tablets*, *netbooks*, reproductores de música. Android permite programar en un entorno de trabajo (*framework*) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, *widgets*, o incluso, modificar el propio sistema operativo, dado que Android es de código libre. (8)

El sistema operativo Android es una plataforma de código abierto que admite la construcción de aplicaciones para, de manera sencilla, manejar, configurar y personalizar el dispositivo móvil; esto hace que se pueda disfrutar de una interfaz amigable en la que se puede priorizar necesidades como correo, *chat*, redes sociales, multimedia, mensajería. (9)

1.1.2.1 Arquitectura Android

La arquitectura de Android está formada por 4 capas que permiten la generación de aplicaciones. El acceso a cada capa se realiza por intermedio de librerías, en las cuales cada capa puede utilizar los elementos de la capa inferior para realizar sus funciones. Las capas de la arquitectura Android son (10):

Capítulo 1. Fundamentación teórica

- **Linux kernel:** capa de abstracción del *hardware* (hardware abstraction layer), que permite que las aplicaciones accedan a través de controladores (drivers) asumiendo la administración de los recursos del teléfono y del sistema operativo. En esta capa están disponibles los controladores para display (pantalla), keypad (teclado), camera (cámara), wifi (conexión inalámbrica), flash memory (memoria rápida), audio, binder (cobertura) y power management (administrador de energía).
- **Librerías (libraries):** bibliotecas de Android escritas en C/ C++, que se encargan de realizar la comunicación entre la capa de abstracción de *hardware* con la interfaz de programación de aplicaciones. Se destacan: surface manager (gestión gráfica), OpenGL/ES (gráficas 3D), SGL (gráficas 2D), media framework (multimedia), freetype (fuentes de mapas de bits), ssl (seguridad), SQLite, Webkit, libc (librerías C).
- **Entorno de ejecución Android (Android runtime):** no es considerada una capa, pero está formada por las siguientes librerías: Dalvik virtual machine (máquina virtual Dalvik), Core libraries (núcleo de librerías).
- **Marco de trabajo de aplicaciones (application framework):** conformado por clases y servicios que permiten las funciones básicas de los móviles y la programación de las aplicaciones. Se encuentran los siguientes elementos: activity manager (administrador de actividades), Windows manager (administrador de ventanas), content provider (compartidor de aplicaciones), views (vistas), notification manager (administrador de notificaciones), package manager (administrador de paquetes), telephony manager (administrador telefónico), resource manager (administrador de recursos), location manager (administrador de posicionamiento), applications (aplicaciones).

1.1.3 M-Learning

Muchos son los autores que han definido el *m-learning*. Primeramente, Raúl Santiago plantea que el *m-learning* o “aprendizaje electrónico móvil” es una metodología de enseñanza y aprendizaje que se vale del uso de pequeños dispositivos móviles, tales como teléfonos móviles, PDA, tabletas, PocketPC, iPod y todo otro dispositivo de mano que tenga alguna forma de conectividad inalámbrica. (11)

Lo que caracteriza entonces al *m-learning* es que se refiere a los ambientes de aprendizaje basados en la tecnología móvil, orientados a optimizar un aprendizaje muy relacionado con el *u-learning* (*Ubiquitous Learning*, o aprendizaje ubicuo), concepto que hace referencia al aprendizaje apoyado en la tecnología y que se puede realizar en cualquier momento y desde cualquier lugar. Además, este aprendizaje involucra entonces diversos conceptos o atributos que concurren para integrar ese nuevo concepto: es flexible, inmediato y ubicuo; utiliza una pantalla táctil y sensores multifunción; es portable, motivante, colaborativo y accesible; tiene conectividad a internet y acceso a apps; y es cooperativo y colaborativo, entre otros aspectos. (11)

Capítulo 1. Fundamentación teórica

Otro autor plantea, que debido al crecimiento de usuarios de internet y la disponibilidad de dispositivos móviles, se ha desarrollado un nuevo concepto, *m-learning*, el cual se basa fundamentalmente en el aprovechamiento de las tecnologías móviles como base del proceso de aprendizaje. Plantea que un 92% de los alumnos está de acuerdo con que se utilicen los dispositivos móviles con fines educativos en sus clases ya sea dentro o fuera del salón. (12)

Ventajas funcionales (12):

- Aprendizaje a cualquier hora en cualquier lugar.
- El coste de adquisición de un móvil es más barato que el de una computadora, algo que contribuye a disminuir la brecha digital.
- Los dispositivos móviles permiten estar conectados la mayor parte del tiempo.
- Favorece que los alumnos puedan compartir determinadas tareas y actividades con otros compañeros.

Desventajas (12):

- La pantalla pequeña en los dispositivos móviles dificulta la lectura y trae consigo la imposibilidad de mostrar mucha información.
- Dificultades a la hora de instalar y usar determinado software.
- Los estudiantes que no sean usuarios comunes de este tipo de dispositivos, tardarán en adaptarse o se resistirán al cambio.

1.2 Versión 1.0 de la aplicación móvil de la Plataforma educativa XAUCE ZERA

La versión 1.0 de la aplicación móvil de la Plataforma educativa XAUCE ZERA no responde a las necesidades de los usuarios debido a un grupo de características presentadas por la aplicación. En la misma se utilizan servicios simulados, lo cual impide mostrar los contenidos reales de los cursos en los que el usuario se encuentra matriculado.

También carece de funcionalidades como el proceso de notificaciones, ya sea del sistema en general o de un curso en específico, impidiéndole al usuario mantenerse actualizado con los cambios ocurridos en la plataforma. La interacción con los contenidos de los cursos de manera off-line constituye otra carencia

Capítulo 1. Fundamentación teórica

presentada por la aplicación, imposibilitando la realización de tareas y la consulta de los contenidos sin necesidad de encontrarse conectado a la red.

La participación en los fórums de forma *on-line* para el intercambio de opiniones entre los usuarios es otra funcionalidad imprescindible para los que hacen uso de la aplicación, sin embargo, no se encuentra implementada. Además, la interfaz de la versión actual no es la más acertada a las exigencias de los usuarios.

1.3 Estudio de aplicaciones similares

Moodle Mobile (MM)

Es la *app* oficial de Moodle. Es una aplicación la cual utiliza tecnologías *webs* comunes, entre ellas REST (Transferencia de Estado Representacional, siglas en inglés) como protocolo para obtener y enviar información a la plataforma Moodle. El diseño se creó usando HTML 5 y CSS 3, la interacción con el teléfono y el envasado se realiza con *PhoneGap*²⁷. Se encuentra disponible para dispositivos con los sistemas operativos *Android* y *iOS*. Esta aplicación es un reemplazo de la anterior *Moddle App*. (13)

Entre sus funcionalidades se encuentran (13):

- Subir fotos y videos a la *web* (seleccionados de la galería o tomados instantáneamente de la cámara del móvil).
- Ver la lista de participantes en los cursos en los que el usuario está inscrito.
- Ver la información de estos participantes.
- Mandar un mensaje privado a algún participante.
- Agregar una nota privada acerca de algún participante.
- Permite el trabajo *off-line* con sincronización automática.
- Agregar los participantes a la lista de contactos del teléfono.
- Acceso a la versión *web* de la plataforma.
- Acceso a los cursos de la plataforma.

Blackboard Mobile

Blackboard Mobile *Solutions* es un proyecto que brinda a estudiantes y educadores acceso a los aspectos del proceso de enseñanza-aprendizaje desde sus dispositivos móviles. Este proyecto cuenta con aplicaciones que se complementan entre sí para formar un paquete completo de funcionalidades que brindan al estudiante y al docente total acceso a los aspectos más importantes del centro de estudio al que pertenecen. *Blackboard Mobile Central* cubre los temas que respectan al *campus* escolar como mapas, direcciones, anuncios generales y rutas de *autobuses*. Por otro lado, *Blackboard Mobile Learn* se encarga de todo lo referido a la gestión del aprendizaje, brindando acceso a los elementos fundamentales de los cursos a los que el estudiante y el docente se encuentran registrados. (14)

Capítulo 1. Fundamentación teórica

De manera general ambas herramientas permiten:

- La integración con *Dropbox*²⁸.
- La realización de pruebas interactivas desde el dispositivo.
- El recibo de notificaciones de tipo *push*.
- El recibo de avisos enviados por estudiantes y docentes.
- Servicio en la nube que permite el almacenamiento de datos.
- El acceso a las evaluaciones obtenidas en clases.
- El acceso a foros y discusiones.
- El acceso a los *blogs*.
- El acceso a la información de los estudiantes que pertenecen al centro de estudio.
- El acceso a los cursos.

Las aplicaciones anteriormente estudiadas no satisfacen las necesidades de la investigación, ya que son desarrolladas a la medida de la plataforma para la que fueron realizadas, no pudiéndose utilizar en otras plataformas. Otro punto a tener en cuenta es que estas aplicaciones son de código cerrado, por lo que no se puede tomar su código fuente como base para el desarrollo de otra solución.

Sin embargo, con el estudio realizado se han identificados las siguientes funcionalidades como semejanzas entre las mismas que servirán de guía a la propuesta a desarrollar:

- Permitir la comunicación plataforma – aplicación.
- Permitir autenticarse y guardar la configuración.
- Ver calificaciones de exámenes.
- Listar los foros disponibles por materia.
- Permitir comentar en los foros.
- Recibir notificaciones sobre tareas y eventos.

1.4 Ambiente de desarrollo

1.4.1 Tecnologías en el desarrollo de aplicaciones para el sistema operativo Android

Android Software Development Kit

Un SDK (*Software Development Kit*), o kit de desarrollo de *software*, es un conjunto de herramientas que ayudan a la programación de aplicaciones para un entorno tecnológico particular. Es decir, las aplicaciones desarrolladas sobre el SDK estarán destinadas a algún SO, plataforma *hardware*, consola de videojuegos o paquete de *software* en especial. (15)

Entorno de desarrollo integrado (IDE)

El Entorno de Desarrollo Integrado (IDE) es un programa informático compuesto por herramientas de programación que pueden ser partes de aplicaciones existentes. Estos sistemas pueden manejar uno o

Capítulo 1. Fundamentación teórica

varios lenguajes de programación ejemplo Java, C# y C++. Han sido empaquetados como programas de aplicaciones, editores de código, depuradores, compiladores o constructores de interfaces gráficas (16).

Otro autor plantea que un (IDE) es una aplicación de *software*, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de *software*. Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría tienen auto-completado inteligente de código (16).

Los entornos de desarrollo integrados están diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Los IDE presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de *software*. (16)

Eclipse

Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo. No tiene en mente un lenguaje específico, sino que es un IDE genérico. Proporciona herramientas para la gestión de espacios de trabajo, escribir, desplegar, ejecutar y depurar aplicaciones. (17)

Principales características (17):

Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una preconfiguración de ventanas y editores, relacionadas entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.

Depurador de código: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente ayuda a mejorar el código.

Extensa colección de plug-ins: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros.

Android Studio

Android Studio es un nuevo entorno de desarrollo integrado para el sistema operativo Android lanzado por *Google*, diseñado para ofrecer nuevas herramientas para el desarrollo de aplicaciones y alternativa al entorno *Eclipse*, hasta ahora el IDE más utilizado. Este entorno de desarrollo gracias a su sistema de

Capítulo 1. Fundamentación teórica

emulación integrado permite ver los cambios que realizamos en nuestra aplicación en tiempo real, pudiendo además comprobar cómo se visualiza en diferentes dispositivos *Android* con distintas configuraciones y resoluciones de forma simultánea. (18)

Android Studio ofrece (18):

- Un entorno de desarrollo claro y robusto.
- Facilidad para testear el funcionamiento en otros tipos de dispositivos.
- Asistentes y plantillas para los elementos comunes de programación en Android.
- Un completo editor con muchas herramientas extra para agilizar el desarrollo de nuestras aplicaciones.

Tras el análisis realizado se decide que el IDE más adecuado para efectuar el desarrollo del sistema es el Android Studio por ser el entorno de desarrollo oficial para la plataforma Android.

1.4.2 Lenguajes de desarrollo

Java

La principal característica de Java es la de ser un lenguaje compilado e interpretado, a la vez de ser de código abierto. Todo programa en Java ha de compilarse y el código generado es interpretado por una máquina virtual. De este modo se consigue la independencia de la máquina, el código compilado se ejecuta en máquinas virtuales que si son dependientes de la plataforma. Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser conocido como un lenguaje de programación de *applets*¹⁸ que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto. (19)

Kotlin

Kotlin es un lenguaje de programación de tipado estático que corre sobre la Máquina Virtual de Java (JVM) y que también puede ser compilado a código fuente de Javascript. Además, soluciona muchos de los problemas que tiene Java sin añadir demasiada complejidad, es un lenguaje pequeño que puede usar todas las librerías que ofrece la JVM sin problemas. (20)

Principales características (21):

- Compatibilidad: Kotlin es totalmente compatible con el JDK a partir de la versión 6, esto es importante ya que garantiza que kotlin puede ejecutarse en dispositivos Android antiguos sin problemas. Kotlin además es compatible con Android Studio y con las herramientas de compilación de Android.
- Interoperabilidad: Kotlin es 100% interoperable con Java, permitiendo utilizar todas las bibliotecas existentes de Android en una aplicación de Kotlin.

Capítulo 1. Fundamentación teórica

- Tiempo de compilación: Kotlin es compatible con la compilación incremental eficiente, por lo que, aunque hay algunas sobrecargas adicionales para las compilaciones limpias, las creaciones incrementales suelen ser tan rápidas o más rápidas que con Java.

Tras el análisis realizado se decide utilizar Kotlin como lenguaje de desarrollo por ser el lenguaje oficial de Android.

XML (Extensible Markup Language)

El lenguaje de marcas extensible (XML) es un simple y flexible formato de texto, derivado del Lenguaje de Marcado de Anotaciones Generales (SGML, por sus siglas en inglés). XML es un lenguaje de etiquetado que sirve para la representación digital de documentos (y puede incluir tanto texto como fotografías, gráficos). Se creó para hacer posible el intercambio de documentos estructurados a través del Web. (22)

Algunas de las características prácticas de XML que deben tenerse en cuenta son (22):

- En principio, no necesita DTD (*Document Type Definition*). La DTD está formada por una serie de definiciones con los tipos de elementos, atributos y entidades que se consideran “válidas” dentro del documento XML. También especifica el orden en que pueden aparecer. Si no se usa DTD, XML mantiene punteros a la estructura de datos, lo que simplifica el desarrollo de la creación de la aplicación y ahorra tiempo de ejecución.
- Tipo de letra: XML distingue entre mayúsculas y minúsculas a la hora de trabajar con elementos. Si se ha especificado en la DTD el elemento <TEMARIO> y en el documento se escribe <Temario> el *parser* avisará que se está produciendo un llamamiento erróneo.
- Requiere documentos “bien formados”. En un documento XML siempre se necesitará una etiqueta de principio y otra de final para elemento.
- No se pueden enumerar los componentes de un elemento en cualquier orden, existe un orden preestablecido.

1.4.3 Herramientas de Desarrollo de Software

Herramienta Case *Visual Paradigm* para UML

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son utilizadas para automatizar o apoyar una o más fases del proceso de desarrollo de *software*, sirven de ayuda a los ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un *software*. Entre las herramientas CASE para el modelado de los artefactos se encuentra *Visual Paradigm*, de la cual a continuación se describen algunas de sus características más importantes (23).

Capítulo 1. Fundamentación teórica

Visual Paradigm para UML es una herramienta de modelado que soporta el modelado mediante UML y proporciona asistencia tanto a los analistas, ingenieros de *software* como a los desarrolladores durante todo el ciclo de vida del proyecto (24).

Características (24):

- **Dibujo:** facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- **Coherencia entre diagramas:** al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, enviando duplicados.
- **Generación de código:** permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores con la codificación del software.
- **Generación de informes:** permite generar diversos informes a partir de la información inducida en la herramienta.
- **Reutilización:** facilita la reutilización, ya que dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.

1.5 Sistema Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Management System) es una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD, (DB DataBase). (25)

MySQL

MySQL es un sistema de libre distribución y de código abierto, lo primero significa que se puede descargar libremente de Internet; lo segundo (código abierto) significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo. Esa es también la base del funcionamiento del sistema Linux, por eso MySQL se distribuye fundamentalmente para Linux, aunque también hay versiones para Windows. (26)

Principales características de MySQL (26):

- Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para los campos y columnas.
- Dispone de una Interfaz de programación de aplicaciones (API, del inglés Application Programming Interface) en gran cantidad de lenguajes: C, C++, Java, PHP, entre otros.
- Tiene un alto grado de portabilidad entre sistemas.
- Es de fácil instalación y configuración.

Capítulo 1. Fundamentación teórica

SQLite

SQLite es una librería compacta y autocontenida de código abierto y distribuida bajo dominio público que implementa un gestor de bases de datos SQL embebido, sin configuración y transaccional.

Los usuarios más conocidos que la utilizan actualmente en sus aplicaciones son: Adobe, Apple, Mozilla, Google, McAfee, Microsoft, Philips, Sun y Toshiba, entre otros.

Es adecuado su uso debido en:

- Aplicaciones autónomas que dependan de otros procesos, como podría ser un programa para llevar la contabilidad de una pequeña/mediana empresa.
- Para su uso como formato de almacenamiento para cualquier aplicación.
- En pequeños dispositivos que no disponen de servicio SQL (PDAs, Móviles, etc.) (27)

Fue escrita en lenguaje C y es de dominio público, por lo que puede ser utilizada en cualquier tipo de proyectos, sean libres o comerciales. Además, tiene características muy interesantes (28):

- Dependiendo de la plataforma, toda la biblioteca se maneja desde un único archivo (shell de comandos) de apenas 500KB.
- Se utiliza el mismo lenguaje de consulta SQL, por lo que de entrada es fácil usarlo.
- Es multiplataforma, incluso hay una gran cantidad de documentación para conectarlo con lenguajes como Java, PHP, Python, .NET, entre otros.
- Las bases de datos se guardan en un fichero con extensión “.db”.

Tras el análisis realizado se decide utilizar SQLite como Sistema Gestor de Base de Datos porque permite que la base de datos se encuentre integrada a la aplicación posibilitando las llamadas simples a subrutinas y funciones.

1.6 Metodologías de desarrollo de software

Una metodología de desarrollo de *software* es un conjunto de métodos, estas se elaboran a partir del marco definido por uno o varios modelos del ciclo de vida. Una metodología de desarrollo consiste en un conjunto de procedimientos, técnicas, herramientas y soporte documental, que deben seguirse para el desarrollo de un *software*. Estas especifican (29):

- La división de un proyecto en etapas.
- Las tareas que hay que realizar en cada etapa.
- Las salidas que se producen y cuándo.
- Las herramientas que se utilizan.
- La gestión y el control de un proyecto.

Capítulo 1. Fundamentación teórica

1.6.1 Metodologías tradicionales

Son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado, reciben el apelativo de metodologías tradicionales o pesadas. (30)

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del *software*, con el fin de conseguir un *software* más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto *de software*. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. (30)

Entre las metodologías tradicionales o pesadas podemos citar:

- RUP (Rational Unified Procces).
- MSF (Microsoft Solution Framework).
- Win-Win Spiral Model.
- Iconix.

1.6.2 Metodologías de desarrollo de software ágiles

Los métodos ágiles son formas de desarrollo iterativo que se centran en la especificación, diseño e implementación del sistema de forma incremental, implican directamente a los usuarios en el proceso de desarrollo del *software*. Lograr reducir la carga en cuanto al esfuerzo de desarrollo puede hacer posible un desarrollo de *software* más rápido. (29)

1.6.2.1 Metodología AUP

El Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*) es una versión simplificada del Proceso Racional Unificado (*Rational Unified Process*, RUP) desarrollada por Scott Ambler, que describe una aproximación al desarrollo de aplicaciones que combina conceptos propios del proceso unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad. (31)

En general, el AUP supone un enfoque intermedio entre XP (*eXtreme Programming*) y el RUP, y tiene la ventaja de ser un proceso ágil que incluye explícitamente actividades y artefactos a los que la mayoría

Capítulo 1. Fundamentación teórica

de desarrolladores ya están acostumbrados. Para la metodología AUP el ciclo de vida de un proyecto se divide en las fases de inicio, elaboración, construcción y transición. (31)

Metodología AUP en su versión UCI

Al no existir una metodología de *software* universal cualquiera puede ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos), por lo que se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, que se ha llamado Ejecución y se agrega la fase de Cierre. Para una mayor comprensión se muestra la siguiente tabla:

Tabla 1: Fases de AUP y AUP-UCI

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

Capítulo 1. Fundamentación teórica

	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.
--	--------	---------------------------------------------------------------------------------------------------------------------------------------------

Para el Modelado del dominio se plantean tres formas de encapsular los requisitos: Casos de uso del sistema (CUS), Historias de usuario (HU) y por último la Descripción de requisitos de procesos(DRP), de los cuales surgen cuatro escenarios para modelar el sistema en los proyectos quedando de la siguiente forma:

Escenario No1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

Escenario No2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

Escenario No3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

Escenario No4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Debido que en la investigación no se modela el negocio, solo se puede modelar el sistema, se escoge el escenario No4 para realizar la descripción de los requisitos.

1.7 Conclusiones parciales

El estudio y análisis del estado del arte permitió concluir que, la existencia de una nueva versión de la aplicación para el sistema operativo Android de la plataforma educativa XAUCE ZERA es necesaria, pues la anterior carece de funcionalidades imprescindibles para los intereses de los usuarios que hacen uso de la misma. El estudio de las características de las herramientas y tecnologías a utilizar fundamenta su selección, seleccionándose para el desarrollo de la solución como IDE, Android Studio en su versión 3.0, como lenguajes de programación Kotlin debido a que el mismo permite optimizar el tiempo y el ciclo de desarrollo (compilación y ejecución) y como lenguaje de etiquetado XML. Además, se selecciona como herramienta de modelado *Visual Paradigm* en su versión 8.0. Así mismo, se selecciona como metodología de desarrollo para la guía del proceso AUP en su versión UCI ya que esta es la metodología estandarizada para el desarrollo de productos de *software* en la Universidad y es la que mejor se ajusta al ciclo de vida definido para la actividad.

Capítulo 2. Propuesta de solución

Capítulo 2. Propuesta de solución

Introducción

Para continuar el desarrollo de la presente investigación es imprescindible saber las características que debe tener el sistema a desarrollar. Con este propósito, 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 en su versión UCI en concordancia con el escenario escogido, se desarrollarán los artefactos pertinentes.

2.1-Descripción de la propuesta de solución

Desarrollo de la versión 2.0 de la aplicación para dispositivos móviles con SO *Android*, que incluye la estrategia *m-learning* en la plataforma actual XAUCE ZERA, garantizando a los usuarios acceder a esta información desde cualquier lugar que se encuentren y en cualquier momento. El sistema debe permitir tener acceso a los cursos que brinda la plataforma en cada programa de estudio, aceptando el trabajo *offline* con dicho contenido, posibilitándole la realización de las tareas disponibles en los mismos.

La nueva versión admitirá incluir el acceso a los foros de cada uno de los cursos en la plataforma, permitiendo que los alumnos lean, revisen y reflexionen sobre lo expresado, lo cual favorecerá las relaciones e intercambios de carácter social promoviendo el aprendizaje colaborativo. También, se utilizarán una serie de servicios *web* creados en la plataforma, que admiten la descarga de contenido a la aplicación, para de esta manera mantenerla actualizada con los últimos cambios realizados y permitir que los usuarios puedan recibir notificaciones desde la plataforma.

Esta solución se convertirá en una aplicación que mantendrá a los usuarios informados y le permitirá gestionarse su propio conocimiento como resultado del estudio de los cursos que ofrezcan, enriqueciendo y creando así, una nueva oportunidad de enseñanza-aprendizaje, promoviendo el *m-learning* en la universidad.

2.2-Modelo de dominio

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis. (32)

A continuación, se describen los conceptos que conforman el modelo de dominio:

Usuario: usuario o persona que interactúa con la aplicación.

Capítulo 2. Propuesta de solución

Plataforma educativa XAUCE ZERA 2.0: plataforma educativa destinada a apoyar el proceso de enseñanza-aprendizaje y que les permite a los usuarios del sistema intervenir en un ambiente dinámico e interactivo.

Notificación: Tipo de alerta que se envía al usuario.

Curso: Modo de presentar el conocimiento acerca de una asignatura determinada, es la forma de mostrar el conocimiento del cual se desea que se apropie el usuario.

Actividad: Conjuntos de tareas, que contienen los cursos para medir el nivel de apropiación de los conocimientos, que pretende transmitir el curso al usuario.

Recurso: Materiales de apoyo que utilizan los cursos para hacer llegar al usuario un conocimiento dado. Estos recursos pueden ser de diferentes tipos (imágenes, vídeos y audios).

Imagen: Archivo que muestra una presentación visual, como por ejemplo una foto, un dibujo.

Tarea: Práctica de una obligación o a la realización de una actividad que debe realizarse en un tiempo limitado.

Foro: Sitio de discusión en línea asincrónico donde las personas publican mensajes alrededor de un tema, creando de esta forma un hilo de conversación.

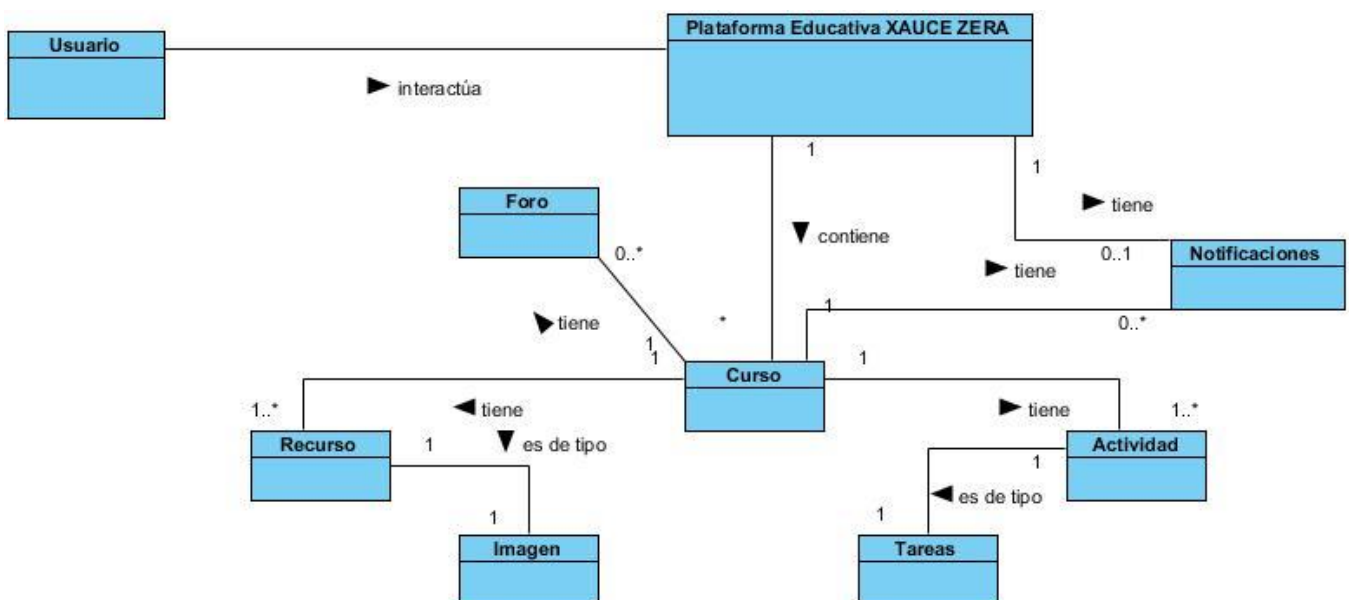


Fig. 1 Diagrama del modelo de dominio

2.3-Especificación de requisitos de software

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema

Capítulo 2. Propuesta de solución

que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. (29)

2.3.1 Requisitos funcionales

Los requisitos funcionales describen como debe comportarse el sistema ante un determinado estímulo. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones específicas. (29)

Los requisitos funcionales identificados para dar solución a la problemática planteada se describen a continuación:

No.	Requisito funcional	Descripción
RF1	Listar notificaciones	La aplicación debe mostrar un listado con las notificaciones que le han sido enviadas al usuario. En el listado se mostrarán los siguientes datos: <ul style="list-style-type: none">• Estado de la notificación.• Asunto de la notificación.
RF2	Buscar notificación	La aplicación debe permitir al usuario buscar una notificación en el listado de notificaciones teniendo en cuenta el siguiente criterio de búsqueda: <ul style="list-style-type: none">• Asunto de la notificación.
RF3	Filtrar notificaciones	La aplicación debe permitir al usuario filtrar la búsqueda en el listado de notificaciones teniendo en cuenta el siguiente criterio de búsqueda: <ul style="list-style-type: none">• Estado de la notificación.
RF4	Visualizar notificación	La aplicación debe permitir al usuario visualizar una notificación del listado de notificaciones.
RF5	Eliminar notificación	La aplicación debe permitir al usuario eliminar las notificaciones del dispositivo, pero no del sistema.
RF6	Marcar notificación como no leídas	La aplicación debe permitir al usuario marcar como no leída una notificación que fue mostrada con anterioridad.

Capítulo 2. Propuesta de solución

RF7	Marcar notificación como leídas	La aplicación debe actualizar el estado de la notificación como leída de las notificaciones visualizadas por el usuario.
RF8	Cerrar notificación	La aplicación debe permitir al usuario cerrar la notificación que le fue enviada.
RF9	Mostrar tarea de forma <i>off-line</i>	La aplicación debe permitir al usuario cuando no esté conectado a la red, visualizar las tareas que estén propuestas en el curso en el que se encuentre matriculado.
RF10	Responder tarea de forma <i>off-line</i>	La aplicación debe permitir al usuario cuando no esté conectado a la red, responder las tareas que estén propuestas en el curso en el que se encuentre matriculado.
RF11	Listar tareas de forma <i>off-line</i>	La aplicación debe mostrar al usuario un listado con las tareas que se encuentren <i>off-line</i> teniendo en cuenta los cursos en los que se encuentra matriculado. En el listado se mostrarán los siguientes datos: <ul style="list-style-type: none">• Nombre de la tarea.• Descripción de la tarea.
RF12	Buscar tareas de forma <i>off-line</i>	La aplicación debe permitir al usuario buscar una tarea en el listado de tareas teniendo en cuenta los siguientes criterios de búsqueda: <ul style="list-style-type: none">• Nombre de la tarea.• Descripción de la tarea.
RF13	Descargar curso	La aplicación debe permitir al usuario descargar un curso en el que se encuentre matriculado.
RF14	Comentar en el foro	La aplicación debe permitir al usuario comentar en un foro que se encuentre disponible en los cursos en los que se encuentra matriculado.
RF15	Visualizar comentarios del foro	La aplicación debe permitir al usuario visualizar los comentarios que se realicen en un foro que se encuentre disponible en los cursos en los que se encuentra matriculado.
RF16	Eliminar comentario del foro	La aplicación debe permitir al usuario eliminar los comentarios realizados en un foro que se encuentre

Capítulo 2. Propuesta de solución

disponible en los cursos en los que se encuentra matriculado.

RF17	Listar temas de los foros	La aplicación debe mostrar al usuario un listado de los temas de un foro que se encuentre disponible en los cursos en los que se encuentra matriculado.
RF18	Listar foros	La aplicación debe mostrar un listado con los foros que correspondan a un curso determinado. En el listado se deben mostrar el siguiente dato: <ul style="list-style-type: none">• Nombre del foro.
RF19	Buscar foros	La aplicación debe permitir al usuario buscar un foro en el listado de foros teniendo en cuenta el siguiente criterio de búsqueda: <ul style="list-style-type: none">• Nombre del foro.
RF20	Sincronizar los datos con la plataforma	La aplicación debe permitir que se puedan sincronizar los datos del servidor con los de la aplicación cuando el dispositivo esté conectado a la red.
RF21	Mostrar contenidos de un curso de manera off-line	La aplicación debe permitir al usuario visualizar el contenido del curso, el cual puede ser imágenes, texto, videos, audio y las actividades que posea el curso en el cual está matriculado.
RF22	Cambiar idioma	La aplicación debe permitir al usuario cambiar el idioma de la misma.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. (29)

A continuación, se listan los requisitos no funcionales con los que debe cumplir la solución:

Capítulo 2. Propuesta de solución

Clasificación	Requisito no funcional
Usabilidad	RNF 1. La aplicación podrá ser utilizada por usuarios con conocimientos básicos en el manejo de dispositivos móviles que posean SO Android. RNF 2. La aplicación debe ser multilinguaje (español/ingles).
Confiabilidad	RNF 3. Sólo los usuarios con una combinación de nombre de usuario y contraseña válidos podrán acceder a la información de la plataforma.
Eficiencia	RNF 4. Las peticiones a la plataforma a través de <i>Internet</i> , deben realizarse en un hilo aparte del principal, permitiendo así que el usuario nunca deje de interactuar con la aplicación.
Soporte	RNF 5. La aplicación debe adaptarse a los distintos tamaños de pantalla que presentan los dispositivos móviles en los cuales es utilizada. RNF 6. La aplicación permitirá el trabajo <i>offline</i> . RNF 7. La aplicación necesitará de una conexión a <i>Internet</i> en el momento que sea necesario para consumir o enviar datos al servidor.
Interfaz de comunicación	RNF 8. La conexión a <i>Internet</i> se realizará a través de redes inalámbricas.

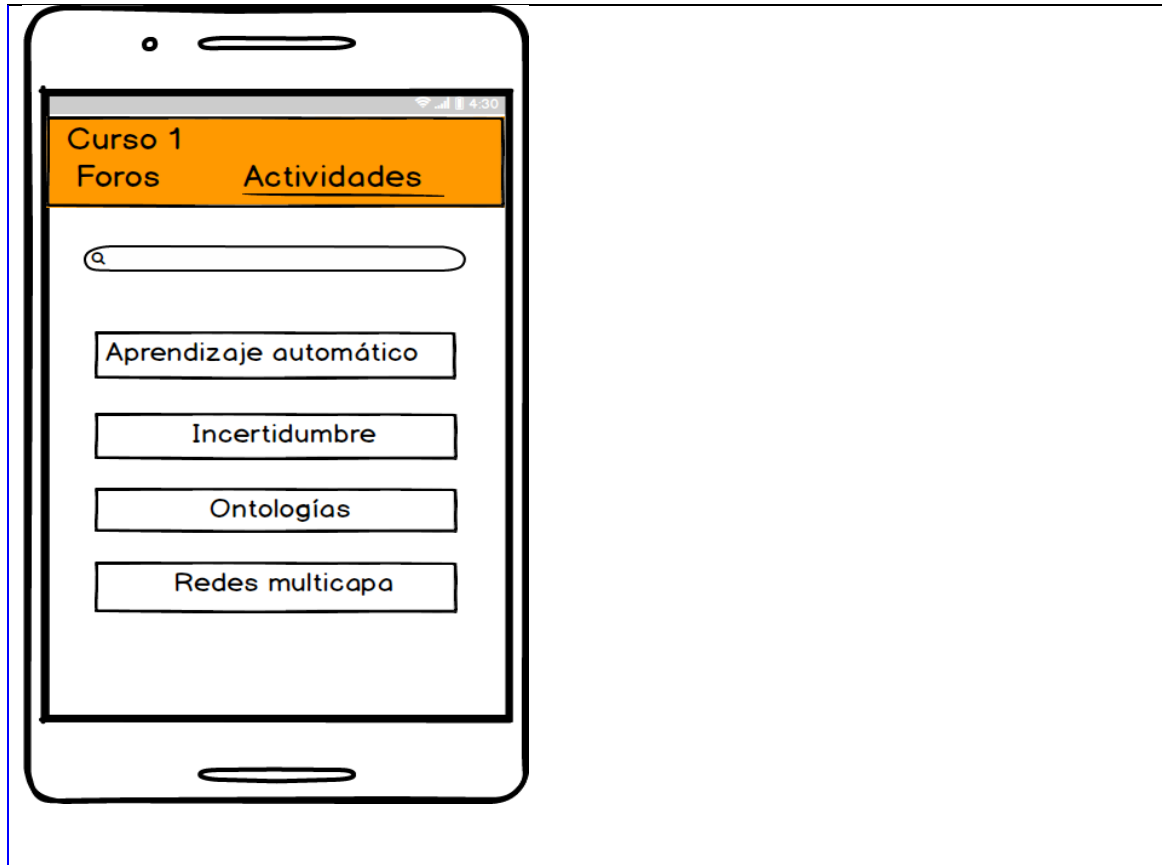
2.4-Descripción de requisitos funcionales

Teniendo en cuenta la metodología seleccionada para guiar el proceso de desarrollo y el 4to escenario, los requisitos funcionales se administran en Historias de usuarios (HU). Estas últimas son una de las técnicas que utiliza la metodología AUP en su versión UCI para describir los requisitos del *software*, las cuales describen brevemente las características que el sistema debe poseer, utilizando para ello un lenguaje no técnico. Para el desarrollo de la solución propuesta se describen un total de 22 HU (Ver Anexo), de las cuales se presenta la correspondiente al RF5: Listar actividades de forma off-line.

Capítulo 2. Propuesta de solución

Tabla 2: HU Listar tareas de forma off-line

Número: 11		Nombre del requisito: Listar tareas de forma <i>off-line</i>	
Programador: Raymari Sanjudo Cabrera		Iteración Asignada: 1era	
Prioridad: Alta		Tiempo Estimado: 4 días	
Riesgo en Desarrollo: N/A		Tiempo Real: 3 días	
Descripción:			
1- Objetivo:			
Permitir listar los datos de las actividades en el sistema			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para listar las actividades hay que:			
- Estar autenticado en el sistema con el rol X.			
- Debe existir en el sistema al menos una actividad.			
3- Comportamientos válidos y no válidos (flujo central y alternos):			
N/A			
4- Flujo de la acción a realizar:			
Cuando el usuario selecciona la opción tareas aparecen todas las tareas registradas en el sistema.			



2.5-Patrón Arquitectónico

El patrón Modelo-Vista-Controlador (MVC) surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del *software* de forma sencilla y en un reducido espacio de tiempo (33).

- Modelo: El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones (33).
- Vista: La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario (33).
- Controlador: El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo (33).

En Android el patrón arquitectónico MVC posee una particularidad en cuanto a la vista y el controlador, definiéndose estos dos como un par Controlador-Vista. En este caso particular el controlador notifica al

Capítulo 2. Propuesta de solución

modelo y el modelo notifica a todos los dependientes (pares Controlador-Vista) asociados a él. Para este par la vista contiene una instancia de su controladora y la controladora además posee una instancia de la vista.

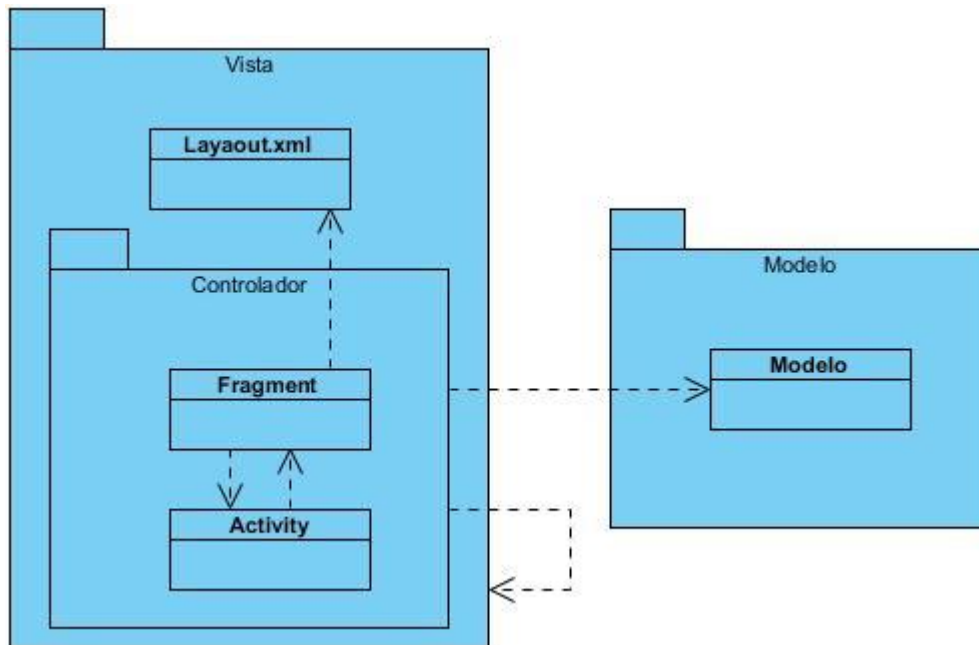


Fig. 2 MVC para Android

2.6-Patrones de diseño

Los patrones de diseño brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares (34).

De los patrones de diseño estudiados, se seleccionaron los que se considerarán necesarios en el desarrollo de la aplicación. A continuación, se muestra dicha selección en conjunto con una breve descripción del patrón.

2.6.1-Patrones GRASP

General Responsibility Assignment Software Patterns (GRASP) nos da unos principios generales para asignar responsabilidades (35).

- **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 (35). Se utilizó en clases controladoras como `LoginViewModel` y `CursoViewModel`.
- **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

Capítulo 2. Propuesta de solución

conectarse al objeto creado en alguna situación (35). Se utilizó este patrón en clases como NetworkModule y PreferenceModule.

- **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 (35). Se utilizó en clases como CursoRepository y NotificationRepository.

2.6.2-Patrones GOF

- **Singleton** (instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto. (35) Se utilizó en clases como ServiceUtil y Permissions.

2.7-Modelo de diseño

Representa todas las clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos, constituyendo la entrada principal a las actividades de la fase de implementación. (36)

A continuación, se describe de manera general el significado de los principales elementos presentes en los diagramas de clases del diseño para lograr una mejor comprensión de estos.

- **Paquete Vista:** Contiene las clases que muestran la información al usuario.
- **Paquete Controlador:** Contiene las clases que realizan el tratamiento de eventos.
- **Paquete Modelo:** Contiene las entidades generadas en correspondencia con las tablas de la base de datos que almacena toda la información que maneja el módulo.

Capítulo 2. Propuesta de solución

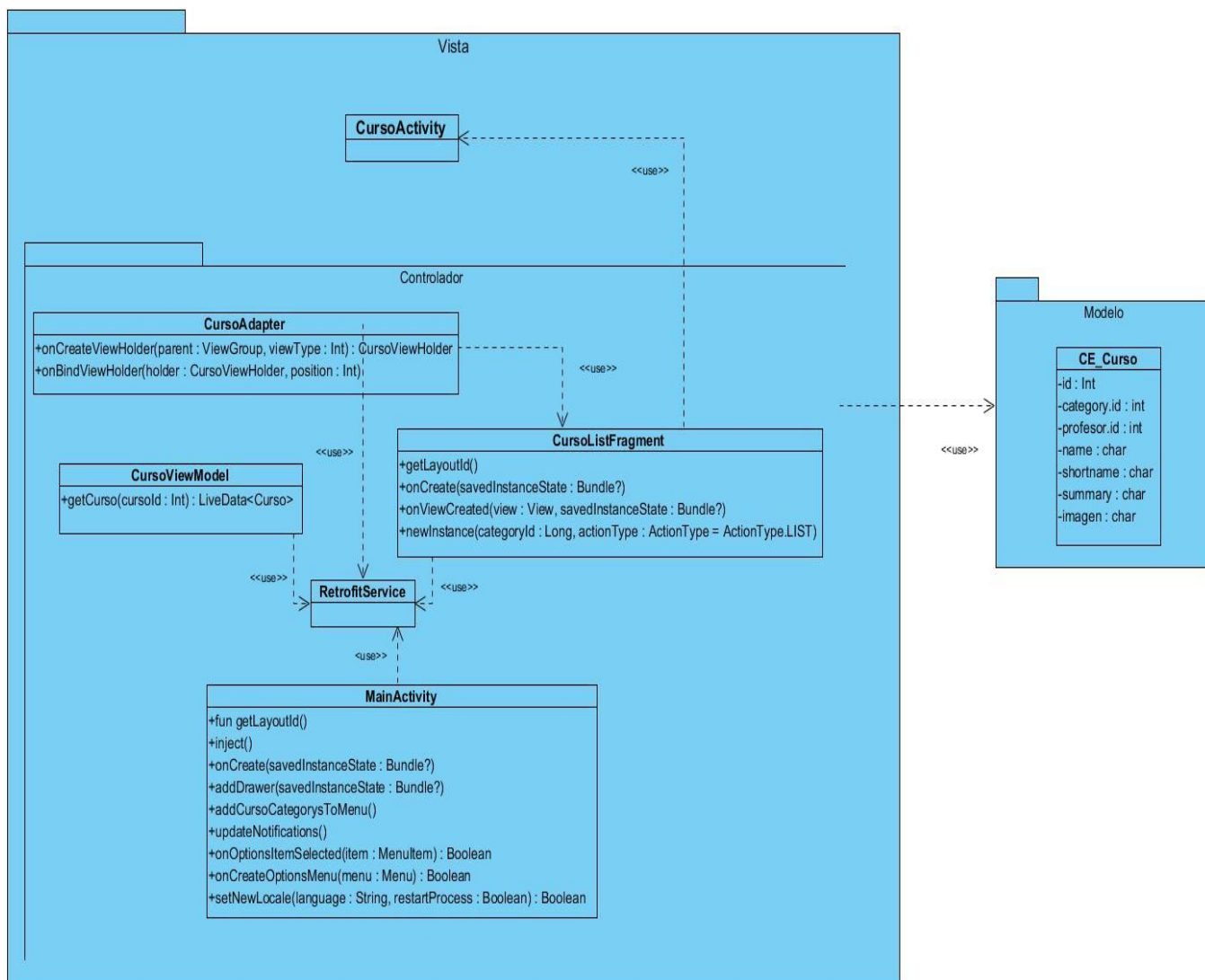


Fig. 3 Diagrama de clases del diseño del Paquete curso

2.8- Diagrama de despliegue.

El Diagrama de Despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo (37):

- Dispositivos
- Procesadores
- Memoria

Capítulo 2. Propuesta de solución

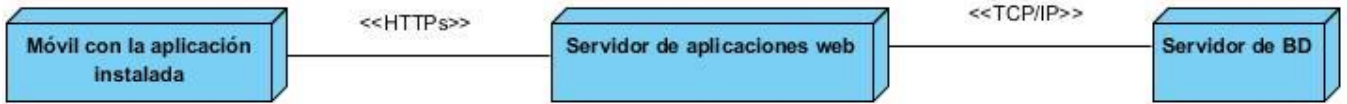


Fig. 4 Diagrama de Despliegue

2.9 Diseño de la Base de Datos

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 (38).

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.

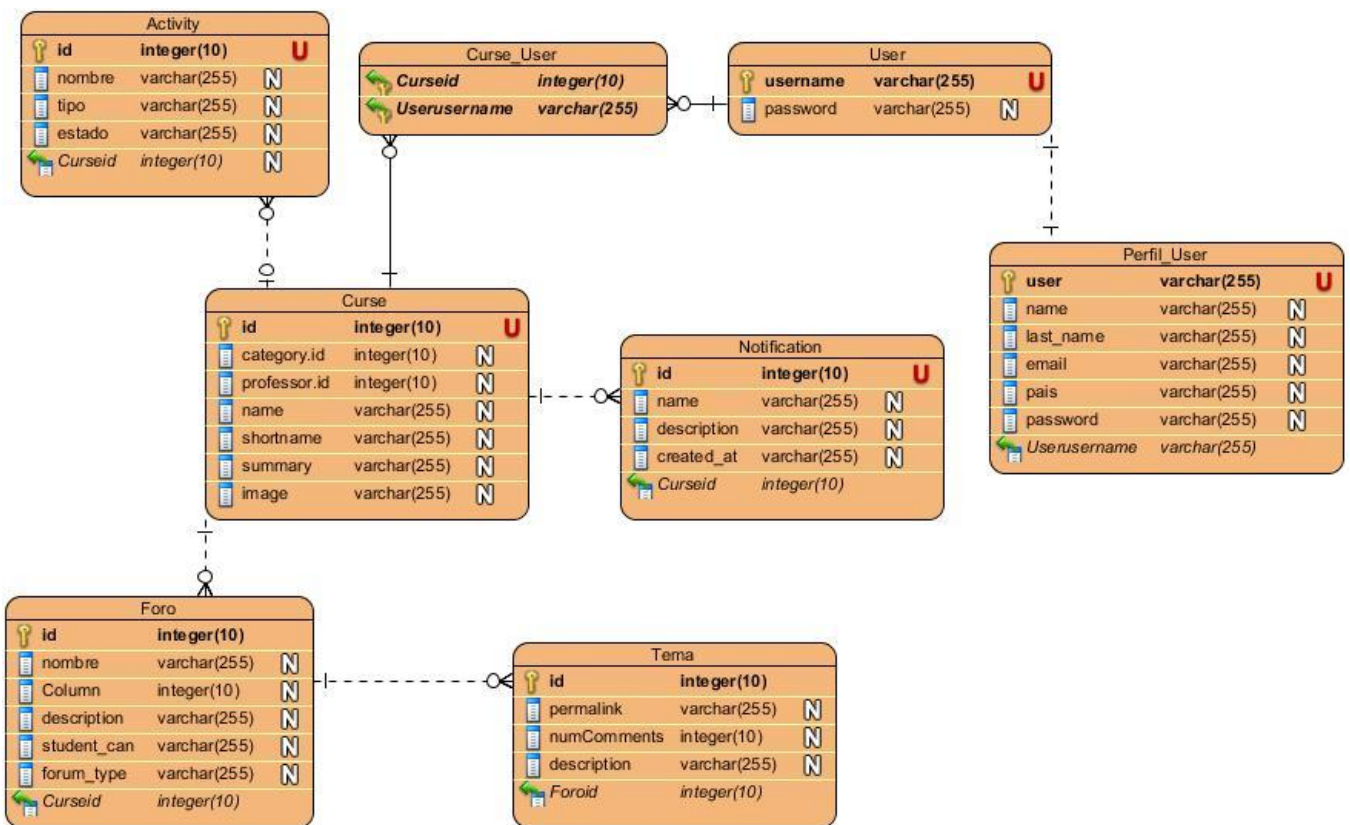


Fig. 5 Diagrama entidad-relación

En la siguiente tabla se muestra la descripción de una de las tablas utilizadas:

Capítulo 2. Propuesta de solución

Tabla 3: Descripción de la tabla Activity

Activity		
Atributo	Tipo	Descripción
Id	Integer (10)	Etiqueta única que identifica a la actividad.
Nombre	Varchar (255)	Almacena el nombre de la actividad.
Tipo	Varchar (255)	Almacena el tipo de actividad.
Estado	Varchar (255)	Almacena el estado de la actividad.

2.10-Conclusiones parciales

-La identificación de los requisitos funcionales y no funcionales del sistema, y su administración mediante historias de usuarios, sirvió de guía para la implementación de las distintas funcionalidades del módulo propuesto.

-La selección de los patrones arquitectónicos y de diseño aportó sencillez al proceso de mantenimiento y permitirá la obtención de un módulo con componentes reutilizables.

-Los artefactos generados durante el análisis y diseño de la solución contribuyeron a un mejor entendimiento del sistema para dar paso a la implementación de la solución propuesta.

Capítulo 3. Implementación y Prueba

Capítulo 3. Implementación y Prueba

Introducción

En el siguiente capítulo se desarrollará la implementación de la propuesta de solución, teniéndose como precedente, los artefactos generados y la descripción de cada uno de los requisitos. Además, se llevará a cabo el proceso de prueba a la solución desarrollada con el objetivo de alcanzar la calidad de la misma. Por último, se muestra un resumen de los resultados obtenidos en cada una de las pruebas realizadas.

3.1 Modelo de implementación

El modelo de implementación garantiza la implementación de la propuesta de solución, basándose en su arquitectura, sus componentes, de acuerdo a su mecanismo de estructuración, y los lenguajes de programación viables.

3.1.1 Estándar de codificación

Se entiende **como estándar de código** a un conjunto de convenciones establecidas de antemano (denominaciones, formatos) para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y además varían en cobertura, algunos son más extensos que otros. Pero hay algunos puntos que todo estándar debería cubrir (39):

- Comentarios: ayudan a entender y revisar más fácilmente el código
- Convenio de denominaciones: proporciona una guía clara de cómo se deben elegir los nombres de clases, funciones y variables
- Legibilidad: provee facilidad de lectura y buena estructuración visual que acompaña la lógica del problema

Las convenciones de código son importantes para los programadores por un gran número de razones (40):

- El 80% del coste del código de un programa va a su mantenimiento.
- Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
- Si distribuyes tu código fuente como un producto, necesitas asegurarte de que está bien hecho y presentado como cualquier otro producto.

Capítulo 3. Implementación y Prueba

Se establece una serie de parámetros para el trabajo de codificación (40):

Convenciones de nombres

Las convenciones de nombres hacen los programas más entendibles haciéndolos más fácil de leer. También pueden dar información sobre la función de un identificador, por ejemplo, cuando es una constante, un paquete, o una clase, que puede ser útil para entender el código.

Clases

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas. Los nombres de las interfaces y los fragments siguen la misma regla que las clases.

Interfaces

Los nombres de las interfaces siguen la misma regla que las clases.

Los métodos

Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.

Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula, deben ser cortos, pero con significado excepto para variables índices temporales. Las palabras restantes que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres guion bajo "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje.

3.1.2 Diagrama de componente (DCOM)

El Diagrama de componente se usa para modelar la estructura del *software*, incluyendo las dependencias entre los componentes, los componentes de código binario, y los componentes ejecutables. Además, muestra las interacciones y relaciones de los componentes de un modelo. (41)

Según varios autores se entiende como componente a una clase de uso específico, que puede ser implementada desde un entorno de desarrollo, ya sea de código fuente, binario o ejecutable, dichos componentes poseen tipo y están unidos mediante relaciones de dependencia. (41)

Capítulo 3. Implementación y Prueba

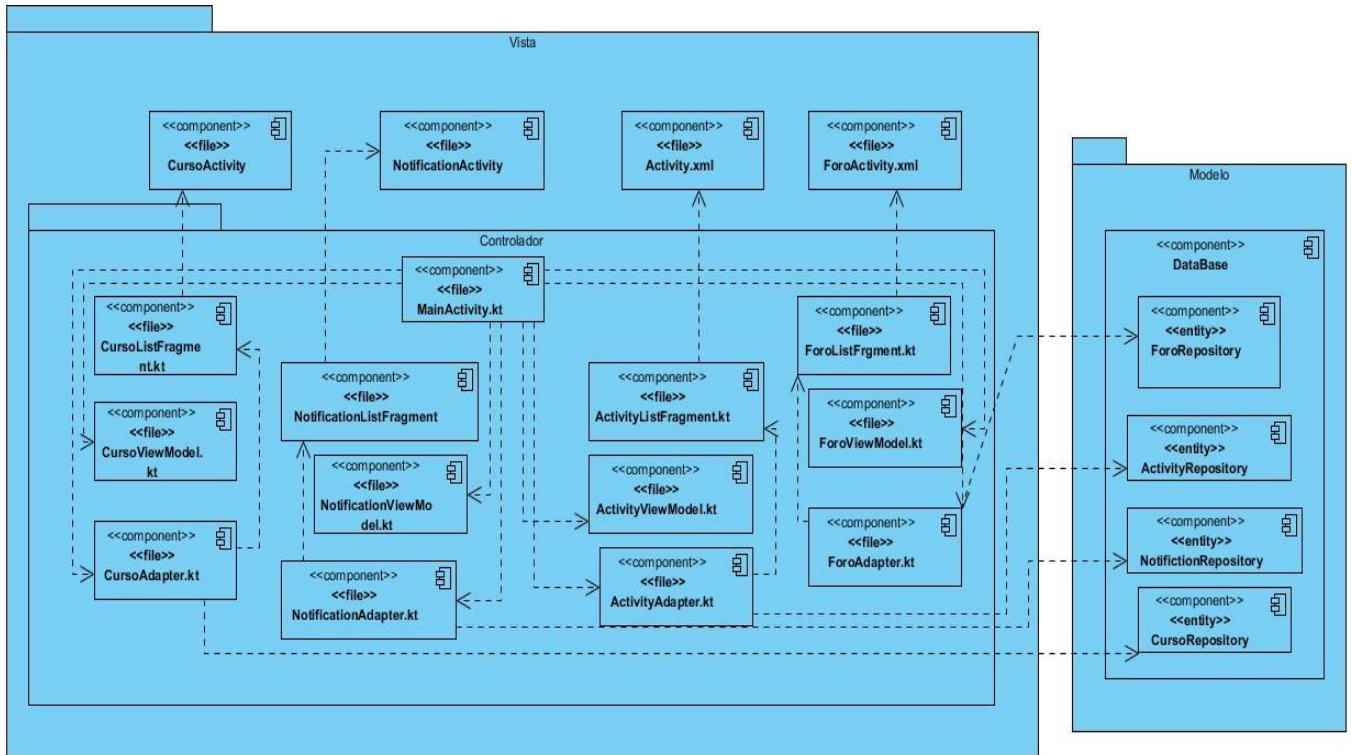


Fig. 6 Diagrama de Componentes del Sistema

3.2 Pruebas de software

Las pruebas de *software* son un proceso iterativo que implican ejercer una implementación del software con datos de prueba. Durante su aplicación se examinan las salidas del software para comprobar que funciona tal y como se quiere. Las pruebas son una técnica dinámica de verificación y validación. (42)

Una estrategia para pruebas de *software* proporciona una hoja de ruta que describe los pasos que se realizarán como parte de las pruebas, cuándo se planifican estos pasos y cuánto esfuerzo, tiempo y recursos se necesitarán. Por lo tanto, cualquier estrategia de prueba incluye los niveles a que se realizarán las pruebas, el tipo de prueba a ser ejecutada y los métodos que se emplearán. Se especifican además las técnicas de prueba y herramientas que se usarán, así como los casos de prueba diseñados para lograr los objetivos (42).

Según la metodología AUP, el objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.

Capítulo 3. Implementación y Prueba

Niveles de pruebas

Cuando se le van a aplicar pruebas a un software, se tienen en cuenta una serie de objetivos en diferentes escenarios y niveles de trabajo, debido a que las pruebas son agrupadas por niveles que se encuentran en distintas etapas del proceso de desarrollo. (42)

3.2.1 Pruebas de sistema

Las pruebas de sistema buscan discrepancias entre el programa y sus requisitos, enfocándose en los errores hechos durante la fase de implementación. Estas últimas pueden incluir pruebas basadas en especificaciones de requisitos, procesos de negocio, casos de uso u otras descripciones de texto de alto nivel o modelos de comportamiento de sistema, interacciones con el sistema operativo y recursos del sistema. (42)

El éxito de una prueba depende de descubrir un error no detectado hasta entonces y su objetivo es diseñar casos de prueba que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo.

Las pruebas de sistema de los requisitos funcionales empiezan utilizando las técnicas basadas en la especificación (técnicas de caja negra) más apropiadas para el aspecto del sistema a probar. (42)

Para el desarrollo de las pruebas se hace uso de los Diseños de Casos de Pruebas elaborados con el objetivo de realizar dichas pruebas.

Métodos de prueba

Cualquier producto sometido a ingeniería puede probarse de dos formas: 1) al conocer la función específica que se asignó a un producto para su realización, pueden llevarse a cabo pruebas que demuestren que cada función es completamente operativa mientras al mismo tiempo se buscan errores en cada función, 2) al conocer el funcionamiento interno de un producto, pueden realizarse pruebas para garantizar que “todos los engranes embonan”; es decir, que las operaciones internas se realizan de acuerdo con las especificaciones y que todos los componentes internos se revisaron de manera adecuada. El primer enfoque de pruebas considera una visión externa y se llama prueba de caja negra. El segundo requiere una visión interna y se denomina prueba de caja blanca. (42)

Para la validación de la propuesta de solución será aplicado el método de caja negra, debido a que este permitirá corregir problemas en la interfaz del usuario y se comprobará que la propuesta de solución realiza las funciones requeridas por el usuario.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software. (42)

Capítulo 3. Implementación y Prueba

Estas pruebas intentan encontrar errores en las categorías siguientes (42):

- Funciones incorrectas o faltantes
- Errores de interfaz
- Errores en las estructuras de datos o en el acceso a bases de datos externas
- Errores de comportamiento o rendimiento
- Errores de inicialización y terminación.

Partición por equivalencia

La partición de equivalencia es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el *software* y descubrir de forma inmediata una clase de errores. La partición equivalente se basa en la definición de casos de pruebas que descubran errores, lo reduce el número de clases de prueba que hay que desarrollar. (42)

Diseños de Casos de pruebas

El diseño de casos de prueba es una parte de las pruebas de componentes y sistemas en las que se diseñan las entradas y salidas esperadas, para probar el sistema. El objetivo de este proceso es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el *software* satisface sus requerimientos. (42)

Los casos de prueba se diseñaron según las funcionalidades descritas en las HU. La intención que se persigue con estos artefactos es lograr una comprensión específica de las condiciones que la solución debe cumplir. Cada descripción de casos de pruebas recoge la especificación de una historia de usuario, dividida en secciones y escenarios, detallando las funcionalidades descritas en ella y describiendo cada variable.

A continuación, se muestra el caso de prueba perteneciente al requisito funcional **Listar notificaciones:**

Capítulo 3. Implementación y Prueba

Tabla 4: Caso de prueba perteneciente al RF Listar Notificaciones

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Listar notificaciones	El usuario selecciona la opción Notificaciones del menú de opciones.	El sistema muestra el listado de notificaciones y de cada una de ella muestra: <ul style="list-style-type: none">• Nombre• Descripción Además permite Buscar una notificación en el listado y marcar dichas notificaciones como no leídas.	Menú/Notificaciones

Resultados obtenidos en las pruebas

A la Versión 2.0 de la aplicación móvil para la plataforma educativa XAUCE ZERA se le realizó la verificación del cumplimiento de los requisitos funcionales establecidos mediante uso de las Pruebas de Caja Negra, teniendo en cuenta la técnica de partición por equivalencia, la cual se basa en la definición de casos de pruebas, que tienen como objetivo detectar la mayor cantidad de no conformidades posibles en las funcionalidades del sistema.

Se realizó una primera iteración a los 16 requisitos funcionales que se encontraban implementados, luego se efectuó 3 iteraciones más, a los 22 requisitos correspondientes ya todos implementados, como se muestra en la siguiente tabla:

Capítulo 3. Implementación y Prueba

Tabla 5: Resultados obtenidos en las pruebas

Iteraciones	Cantidad de casos de pruebas	No conformidades detectadas			
		Alta	Media	Baja	Total
1	16	5	7	11	23
2	22	7	5	14	26
3	22	0	2	5	7
4	22	0	0	0	0

Tabla 6: Clasificación de las No Conformidades (NC) por tipo de error.

Clasificación de la NC	Tipo de NC	Descripción
Funcionalidad	Alta	El resultado Mostrado al realizar una acción no es el esperado.
Errores de Idioma	Alta/Media	Errores relativos al uso de dos o más idiomas en la aplicación.
Ortografía	Baja	Existen palabras con errores ortográficos, por ejemplo cambios de consonantes o vocales repetidas, abreviaturas mal empleadas, palabras sin tildar, utilización incorrecta de mayúsculas y minúsculas.

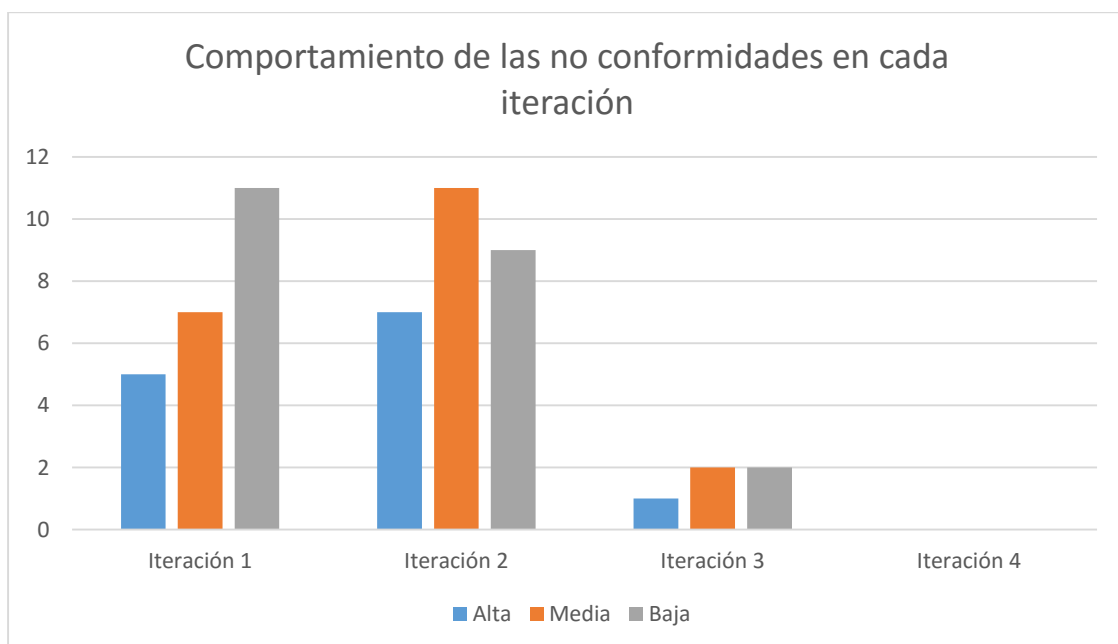


Fig. 7 Comportamiento de las no conformidades en cada iteración

Capítulo 3. Implementación y Prueba

3.2.2 Pruebas de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño. Los componentes a integrar pueden ser: componentes comerciales, componentes reutilizables que han sido adaptados a un sistema particular, o componentes nuevos desarrollados. (42)

Estas pruebas valoran si los componentes individuales trabajan en conjunto tal y como se espera de ellos o lo que es lo mismo, se prueba el funcionamiento de los diferentes módulos del sistema una vez unidos o agrupados en elementos mayores, verificando el comportamiento de los mismos frente a las comunicaciones que se produzcan entre ellos. El objetivo es la localización de errores de interfaces y comprobar el correcto funcionamiento conjunto de los componentes. (43)

Entre las tendencias a utilizar podemos mencionar la integración no incremental donde todos los componentes se combinan a la vez. Todo el programa se prueba como un todo. Otra tendencia a utilizar constituye la integración incremental donde el programa se construye y se prueba en pequeños incrementos, así los errores son más fáciles de aislar y corregir; las interfaces tienen más posibilidades de probarse por completo; y puede aplicarse un enfoque de prueba sistemático. (42)

Resultados de las pruebas de integración

En la presente investigación se realizaron las pruebas de integración de la aplicación móvil a la Plataforma educativa XAUCE ZERA utilizando la integración no incremental, comprobando en una primera iteración el correcto funcionamiento de cada uno de los paquetes de requisitos definidos, y se decide añadir una nueva funcionalidad que le permita diferenciar al usuario los cursos en los que se encuentre matriculado del resto de cursos disponibles en la plataforma. En una segunda iteración se obtuvo resultados favorables, integrándose correctamente la aplicación móvil a la plataforma.

3.2.3 Pruebas de aceptación

Las pruebas de aceptación son importantes dado que significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente. Se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra. (42)

Resultados de las pruebas de aceptación

Las pruebas de aceptación fueron realizadas por el jefe de proyecto y la analista del centro obteniéndose como resultado gran satisfacción por parte del cliente, como evidencia de cumplimiento de las mismas se firma el acta de aceptación del producto.

Capítulo 3. Implementación y Prueba

3.3 Conclusiones del capítulo

Los artefactos generados durante la disciplina de implementación permitieron entender la relación de dependencia y por último el desarrollo final de la solución. Además, se aplicaron pruebas de sistema y de aceptación, permitiendo determinar y erradicar las deficiencias encontradas, obteniendo finalmente una solución con un alto nivel de calidad.

Conclusiones

Una vez finalizada la investigación se puede arribar a las siguientes conclusiones:

- La versión 2.0 de la aplicación móvil para la Plataforma educativa XAUCE ZERA garantiza la interacción de forma off-line con los contenidos de la plataforma desde un dispositivo móvil con sistema operativo Android.
- La utilización de REST como protocolo para intercambiar datos con la capa de servicios web y SQLite como sistema gestor de base de datos permite la sincronización y persistencia en un dispositivo móvil con Sistema Operativo Android de los contenidos de la Plataforma educativa XAUCE ZERA para la interacción con ellos de forma off-line.

Recomendaciones

- Extender las funcionalidades de la aplicación para que los usuarios con rol profesor puedan crear y/o editar un curso.
- Adicionar nuevas funcionalidades a la aplicación como la realización de cuestionarios que contienen los cursos para medir el nivel de apropiación de los conocimientos, que pretende transmitir el curso al usuario.

Bibliografía

1. Soto, Carlos Ferro. VENTAJAS DEL USO DE LAS TICs EN EL PROCESO DE ENSEÑANZAAPRENDIZAJE. España : EDUTEC, 2009.
2. Ceceñas Torrero, Paula Elvira, Martínez Leyva, Diana y Campos Martínez, Karla Rocío. ¿Qué son las TIC'S. México : Red Durango de Investigadores Educativos, 2014. ISBN: 978-607-9063-25-2.
3. Tardáguila, César. Dispositivos Móviles y Multimedia. 2009.
4. Herrera, Susana I. y Fennema, Marta. Tecnologías Móviles Aplicadas a la Educación Superior. Argentina : s.n., 2011.
5. Area, Manuel. E-Learning: enseñar y aprender en espacios. . 2016.
6. Moro, César Tardáguila. Dispositivos Móviles. 2006.
7. Baez, Arturo y Ferreira, Irene. Dispositivos móviles. 2011.
8. Manuel Báez, Álvaro Borrego, Jorge Cordero. Introducción a Android. s.l. : E.M.E. Editorial ©. . ISBN: 978-84-96285-39-5..
9. Vanegas, Carlos Alberto. DESARROLLO DE APLICACIONES SOBRE ANDROID. [En línea]. 2012. .
10. Vanegas, Carlos Alberto. DESARROLLO DE APLICACIONES SOBRE ANDROID. Vol. 9. s.l. : Universidad Distrital Francisco José de Caldas, 2014.
11. Raúl Santiago, Susana Trbaldo, Mercedes Kamijo, Álvaro Fernández. Mobile learning: nuevas realidades en el aula. 2015.
12. Franco, Román Alberto Zamarripa. M-learning: El aprendizaje a traves de la tecnologia movil, desde la perspectiva de los alumnos de la educacion superior. s.l. : Instituto de Estudios Superiores de Tamaulipas, 2015.
13. Montañó, Enrique Castillo. CLIENTE MODDLE PARA ANDROID. [En línea] 2014.
14. Castellanos, José González. DESARROLLO DE UNA APLICACIÓN PARA MÓVILES PARA LA PLATAFORMA EDUCATIVA ZERA. [En línea] 2013.
15. ¿Qué es un Kit de desarrollo de software(SDK)? [En línea] 2013. [http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/..](http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/) [En línea] 2017.

16. Falcón, Yuniel Suárez. [En línea] 2016. [http://deprogramacion.cubava.cu/2016/02/01/que-es-un-ide/..](http://deprogramacion.cubava.cu/2016/02/01/que-es-un-ide/)
17. Eclipse IDE. [En línea] 2014.
18. [En línea] 14 de junio de 2014. <https://android-studio.uptodown.com/windows..>
19. Tutorials, The Java™. [En línea] 2015. <http://docs.oracle.com/javase/tutorial/getStarted/TOC.html..>
20. Suarez, Daniel Diaz. Introducción a Kotlin para el desarrollo de apps en Android. [En línea] 2017.
21. Vásquez, Grover. Kotlin para Android, principales características. [En línea] 2017. <https://codigoonclick.com/kotlin-para-android/>.
22. Senso, José A. Especificaciones XML aplicadas a la Documentación. . Granada. España. : s.n.
23. Mastermagazine. Definición de CASE - Significado y definición de CASE. . [En línea] 2015. <http://www.mastermagazine.info/termino/4182.php...>
24. Visual Paradigm. . [En línea] 2015. <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/ls1y2/PracticaVP.pdf..>
25. Sistemas gestores de bases de datos 1. 2012.
26. Sánchez, Jorge. MySQL. Guía rápida. 2004.
27. Montiel, Daniel Ponsoda. Introducción a SQLite. 2012.
28. [En línea] 2012. <http://desarrollando.net/introduccion-sqlite/>.
29. Sommerville, Ian. Ingeniería de Software. . Madrid : s.n., 2005. ISBN:84-7829-074-5..
30. Acuña, Kareny Brito. SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFORMÁTICA DE LA UNIVERSIDAD DE CIENFUEGOS. Cienfuegos : Universidad de Cienfuegos “Carlos Rafael Rodríguez”, 2009.
31. Torrecilla, Pablo. El Proceso Unificado Ágil: fases y disciplinas. 2012.
32. Larman., Craig. UML y Patrones. . 2003.
33. Alvarez, Miguel Angel. Qué es MVC. . [En línea] 2014. <https://desarrolloweb.com/articulos/que-es-mvc.html..>
34. Tedeschi, Nicolás. ¿Qué es un Patrón de Diseño? . [En línea] 2015. <https://msdn.microsoft.com/es-es/library/bb972240.aspx..>

35. L.Giraldo, Gloria y F.Acevedo, Juan. Una ontología para la representación de conceptos de diseños de software. Colombia : s.n., 2011.
36. Fonseca, Miguel Enrique Verdecia y Ortiz, Osvaldo Manuel González. Visor de reportes para móviles con sistema. La Habana : s.n., 2013. .
37. Quisbert, Hugo Michel y Hualpara, Marca. ANALISIS Y DISEÑO DE SISTEMAS II.
38. Kroenke, David M. Procesamiento de bases de datos: fundamentos, diseño e implementación. s.l. : . s.l. : Pearson Educación, 2003., 2003. .
39. Mercurry. Estandares de Codificación. [En línea] 2017. [https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/..](https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/)
40. UCI, Programa de Mejora Universidad de las Ciencias Informáticas, Basado en Convenciones de Código para el lenguaje de programación JAVA™ por Scott Hommel Sun Microsystems Inc. Traducción por Alberto Molpeceres. Estándar de Codificación para JAVA. 2. 2016.
41. Sparks, Geoffrey. Una Introducción al UML. El Modelo Físico. Chile : s.n.
42. Pressman, Roger S. Ingeniería de Software. Un enfoque práctico. Séptima edición . México : s.n., 2010. ISBN: 978-607-15-0314-5.
43. Rodríguez, Nuria Gómez. Las Pruebas de Integración como Proceso de Calidad de Software en el Ámbito de las Comunicaciones. 2015.
44. <https://www.isdi.education/es/politica-de-cookies>. [En línea] 2014.
45. lopez, andres. Para Programadores. [En línea] 2018.

Anexo 1 HU de los requisitos funcionales restantes

Tabla 7: HU Mostrar actividad de forma off-line

Número: 9	Nombre del requisito: Mostrar tarea de forma <i>off-line</i> .
Programador: Raymari Sanjudo Cabrera.	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir consultar las actividades de un curso en el sistema de forma <i>off-line</i>.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para ver detalles de las actividades de un curso hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un curso X. - Debe existir al menos una actividad de un curso X. <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Cuando el usuario seleccione una actividad del listado de actividades incluidos en el sistema se mostrará la misma.</p>	



Tabla 8: HU Responder la actividad de forma off-line

Número: 10	Nombre del requisito: Responder la actividad de forma <i>off-line</i> .
:	Iteración Asignada: 1era
Programador: Raymaris Sanjudo Cabrera	
Prioridad: Media	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir responder una actividad de un curso de forma <i>off-line</i>.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para responder una actividad de un curso hay que: - Estar autenticado en el sistema con el rol X.</p>	

- Debe existir en el sistema al menos un curso.
- Debe existir en el sistema al menos una actividad.

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

N/A

4- Flujo de la acción a realizar:

Cuando el usuario seleccione una actividad, podrá responder la misma, llenando los campos que se muestran en cuestión y seleccionando la opción “enviar”, se enviará la respuesta propuesta.

Observaciones:

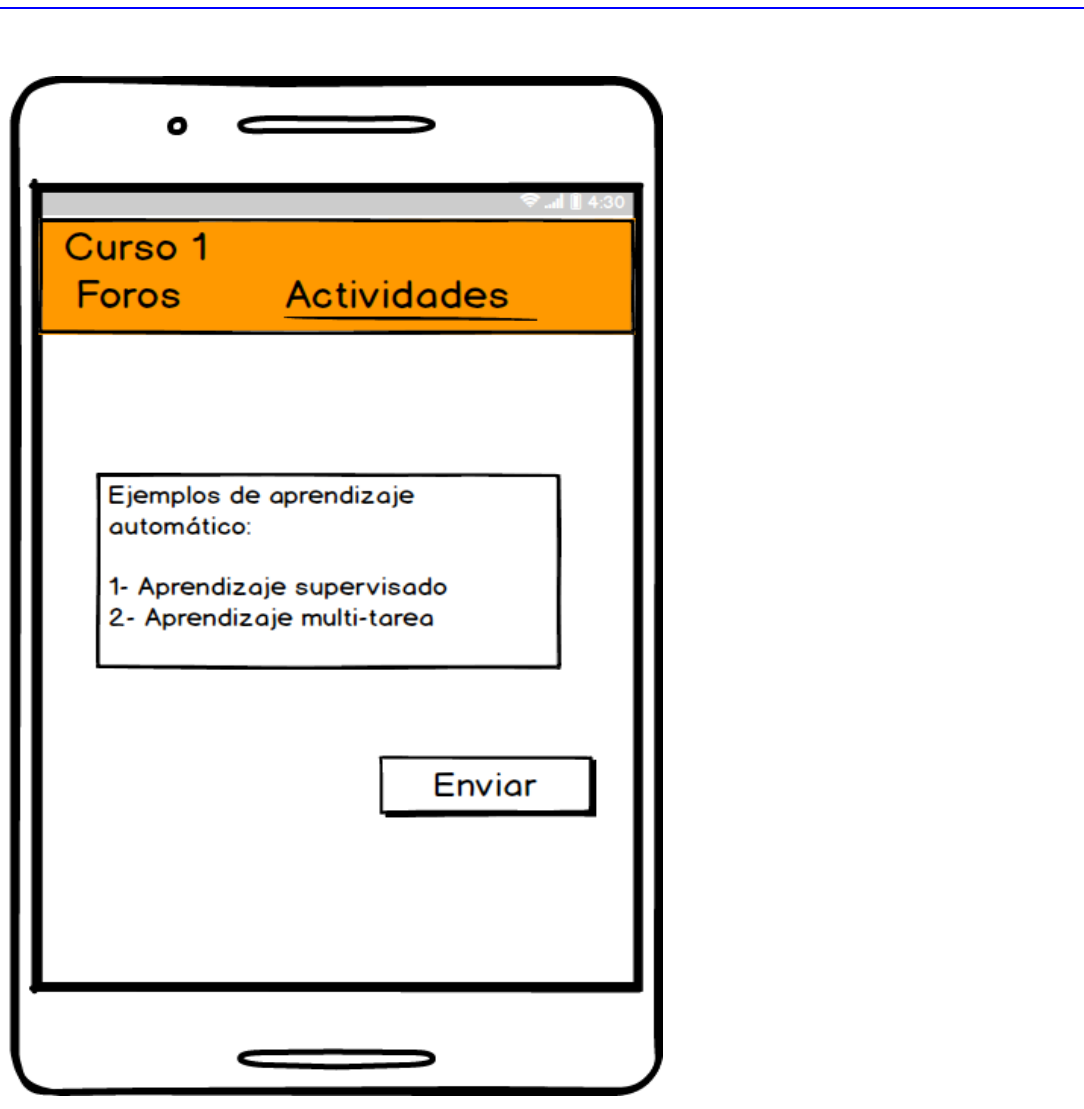


Tabla 9: HU Buscar actividades de forma off-line

Número: 12	Nombre del requisito: Buscar tareas de forma off-line

Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir buscar una actividad en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos): Para buscar en un listado de actividades hay que: <ul style="list-style-type: none">- Tener en cuenta los siguientes datos: nombre y descripción.- Estar autenticado en el sistema con el rol X.- Debe existir en el sistema al menos un curso X.- Debe existir en el sistema al menos una actividad Y de un curso X. 3- Comportamientos válidos y no válidos (flujo central y alternos): El campo búsqueda es obligatorio: nombre de la actividad a buscar. 4- Flujo de la acción a realizar: Cuando el usuario inserta un criterio de búsqueda (nombre de la actividad), el sistema muestra un listado de las actividades que cumplen con el criterio establecido, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.	



Tabla 10: HU Listar notificaciones

Número: 1	Nombre del requisito: Listar notificaciones
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 6 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir listar los datos de las notificaciones en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos):	

Para listar notificaciones hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos una notificación.

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

N/A

4- Flujo de la acción a realizar:

Cuando el usuario selecciona la opción Notificaciones, en el Menú de opciones, aparecen todas las notificaciones registradas en el sistema, dándole la posibilidad al usuario de elegir las notificaciones que desee, permitiéndole visualizar, buscar, eliminar y cambiar el estado de la notificación seleccionada.

Observaciones: las notificaciones corresponden a los avisos que le son enviados al usuario, son dependientes de las acciones que desencadene el mismo.

Prototipo:

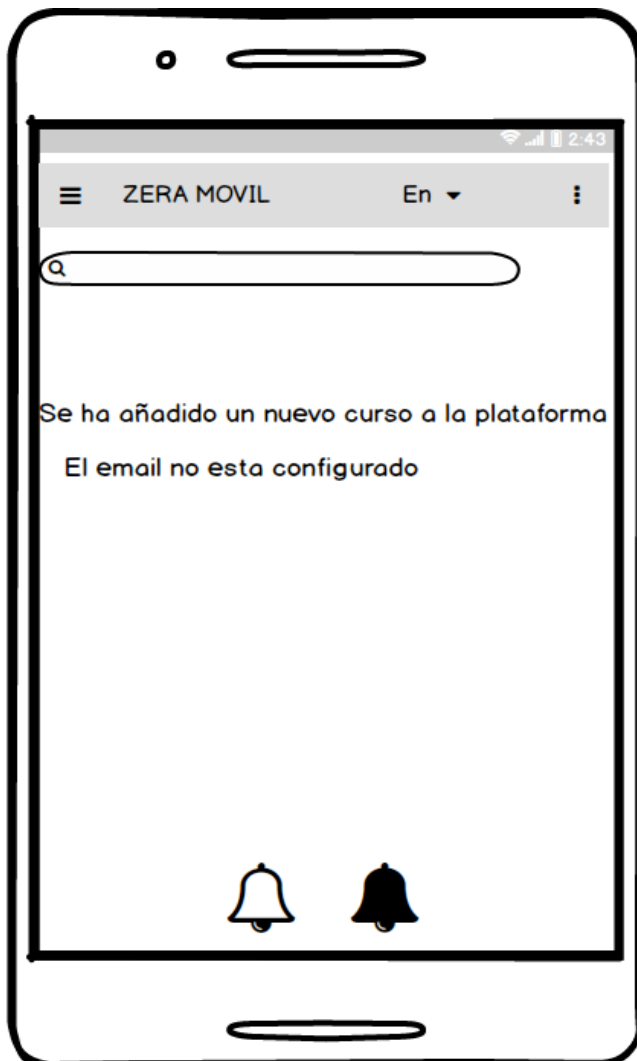


Tabla 11: HU Visualizar notificación

Número: 4	Nombre del requisito: Visualizar notificación
Programador: Raimary Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir consultar los datos de una notificación en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para ver detalles de una notificación hay que: - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos una notificación.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Inicialmente se muestra al usuario un listado con las notificaciones que han sido incluidas en el sistema. Una vez seleccionada una notificación, podrá ver sus datos: contenido de la notificación y fecha de la notificación.</p>	
Prototipo:	

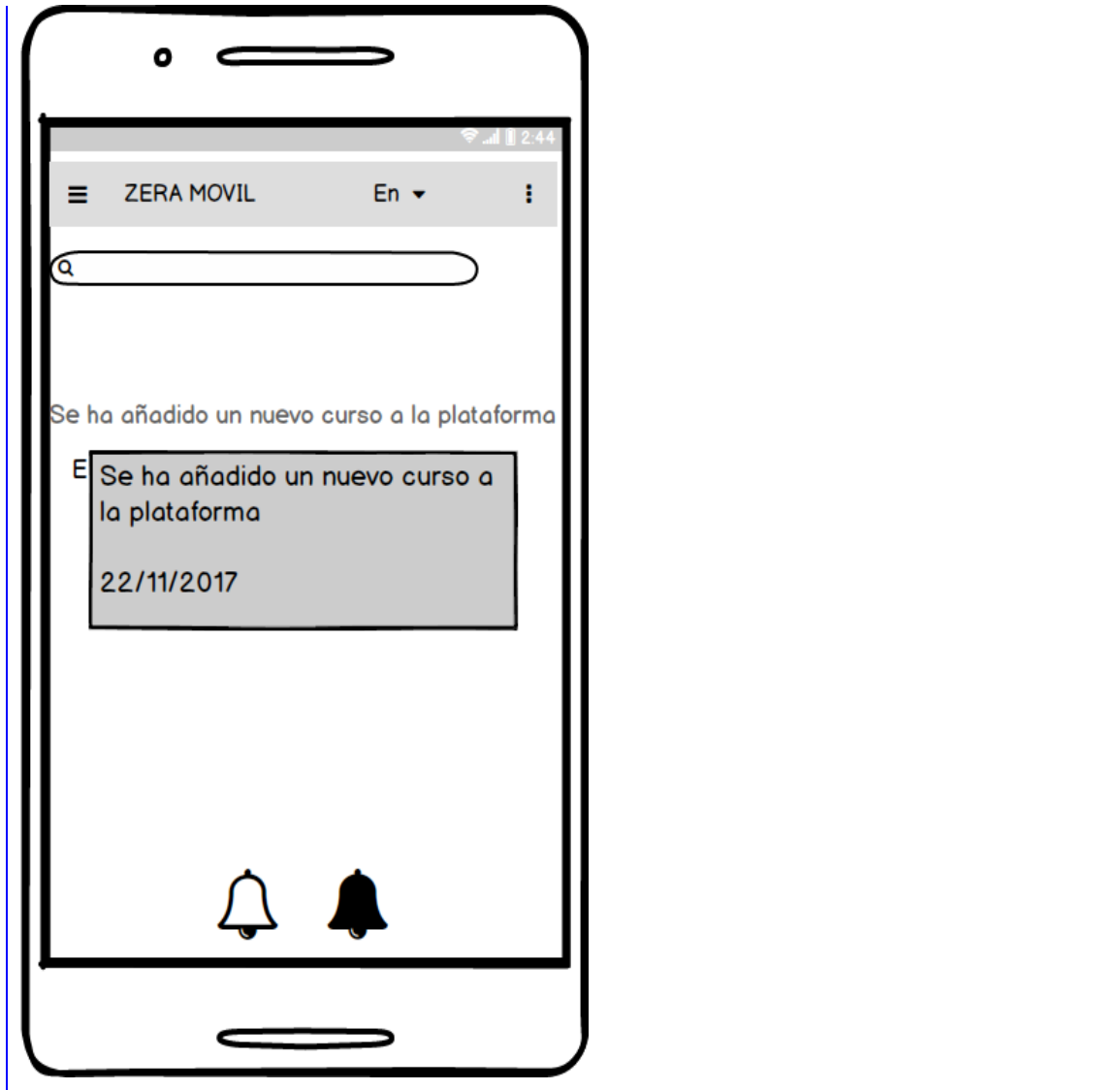


Tabla 12: HU Eliminar notificación

Número: 5	Nombre del requisito: Eliminar notificación
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 4 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir eliminar una notificación en el sistema.	

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

Para eliminar una notificación hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos una notificación.

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

N/A

4- Flujo de la acción a realizar:

Para eliminar una notificación, el sistema muestra un listado de las notificaciones que le fueron enviadas, presionando encima de una notificación se podrá eliminar la misma o cancelar la operación.

Prototipos:

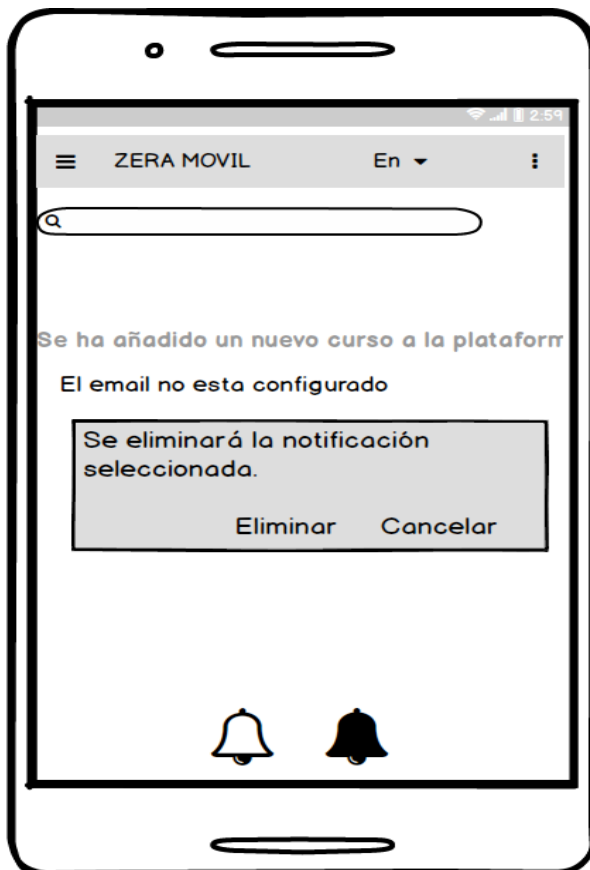


Tabla 13: HU Filtrar notificaciones

Número: 3	Nombre del requisito: Filtrar notificaciones
Programador: Raymaris Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: 2 días
<p>Descripción:</p> <p>1- Objetivo: Permitir filtrar un listado de notificaciones en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para filtrar un listado de notificaciones hay que:</p> <ul style="list-style-type: none"> - Tener en cuenta los siguientes datos: nombre y descripción. - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos una notificación. <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los siguientes campos son obligatorios: Filtrar Búsqueda: puede ser (Tipo de notificación, estado de la notificación, fecha de la notificación).</p> <p>Si se selecciona:</p> <ul style="list-style-type: none"> -Tipo de notificación: puede ser (sistema, curso). -Estado de la notificación: puede ser (leído, no leído). -Fecha de la notificación: comprendida <p>4- Flujo de la acción a realizar: Cuando el usuario inserta un criterio de búsqueda (estado de la notificación, tipo de notificación, fecha de la notificación), el sistema muestra un listado de las notificaciones que cumplen con los criterios establecidos, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.</p>	



Tabla 14: HU Buscar notificación

Número: 2	Nombre del requisito: Buscar notificación
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir buscar los datos de las notificaciones en el sistema 2- Acciones para lograr el objetivo (precondiciones y datos): Para buscar en un listado de notificaciones hay que:	

- Tener en cuenta los siguientes datos: nombre y descripción.
- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos una notificación.

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

El campo búsqueda es obligatorio: puede ser (asunto de la notificación, tipo de notificación).

4- Flujo de la acción a realizar:

Cuando el usuario inserta un criterio de búsqueda (asunto de la notificación), el sistema muestra un listado de las notificaciones que cumplen con los criterios establecidos, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.

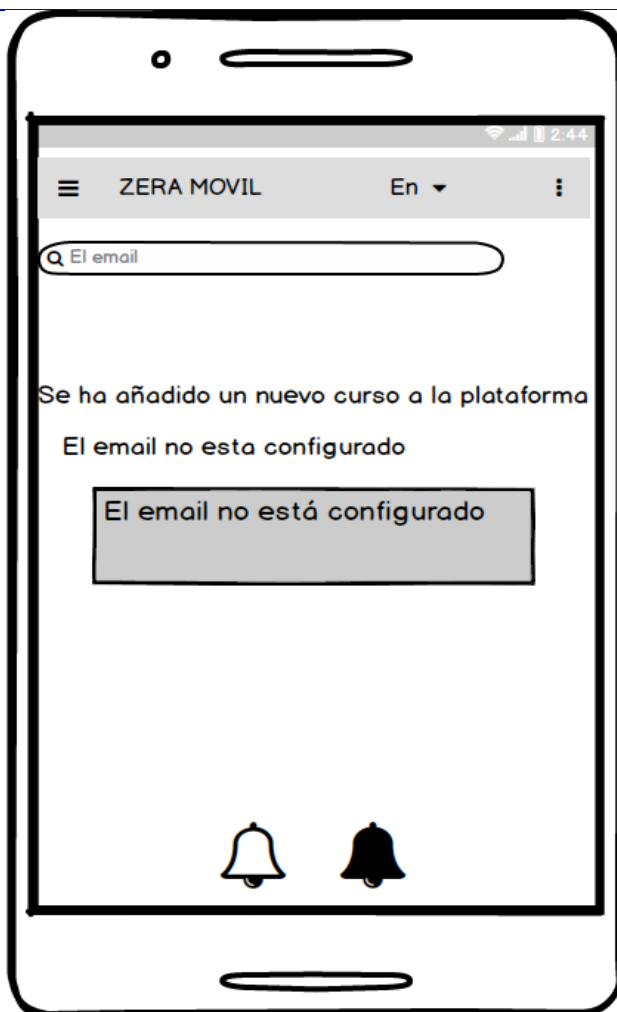


Tabla 15: HU Marcar notificación como no leída

Número: 6	Nombre del requisito: Marcar notificación como no leída.
Programador: Raymari Sanjudo Cabrera.	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 1 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir marcar una notificación como no leída de un curso o del sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para marcar una notificación como no leída hay que: - Estar autenticado en el sistema con el rol X.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Cundo el usuario marca una notificación del listado de notificaciones, ya visto previamente, se le permitirá al usuario enviar la notificación para el listado de notificaciones No Leídas, o Cancelar la operación.</p>	

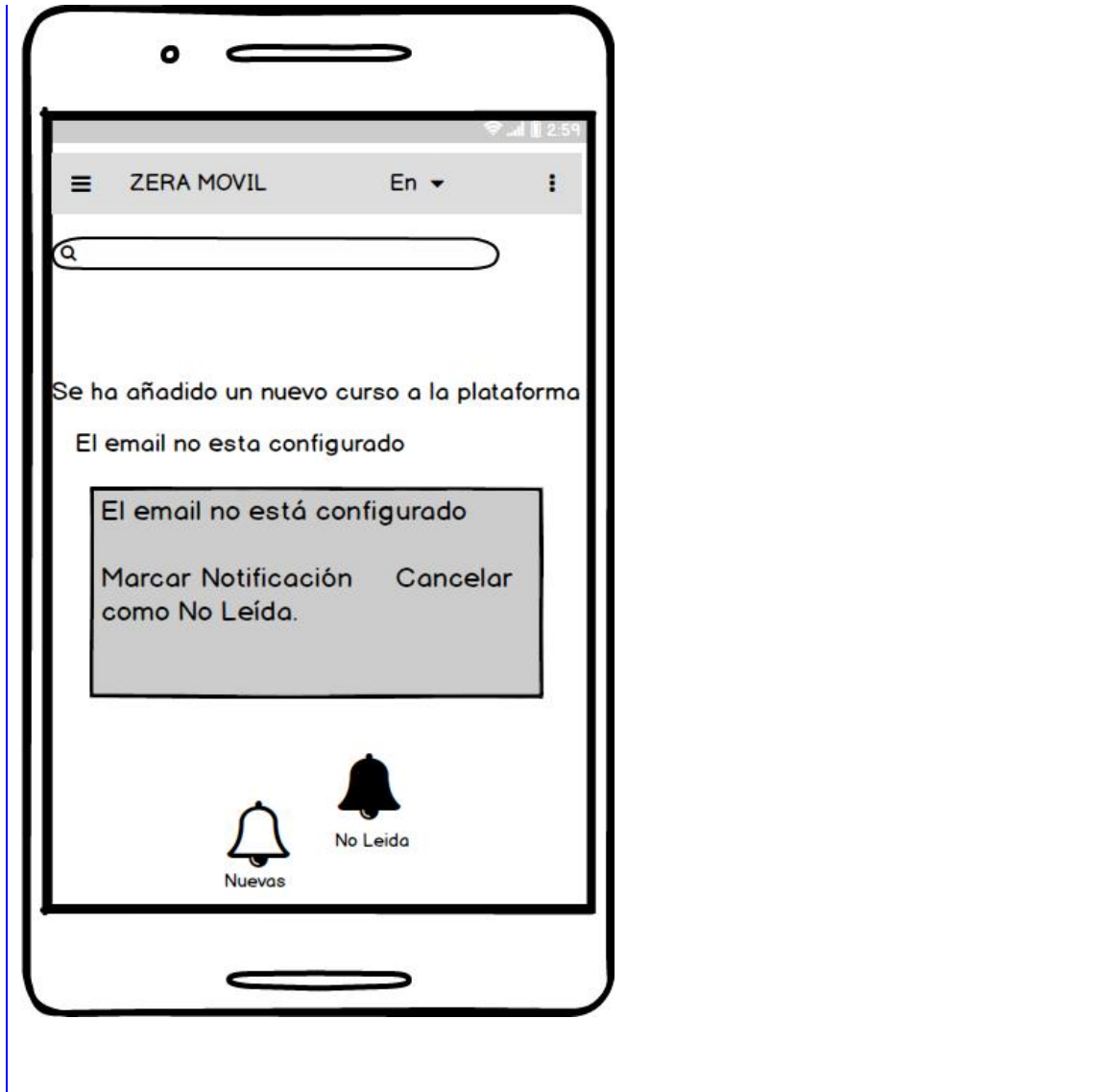


Tabla 16: HU Marcar notificación como leída

Número: 7	Nombre del requisito: Marcar notificación como leída.
Programador: Raymari Sanjudo Cabrera.	Iteración Asignada: 1era
	Tiempo Estimado: 2 días

Prioridad: Media	
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir marcar una notificación como leída de un curso o del sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para marcar una notificación como leída hay que: - Estar autenticado en el sistema con el rol X.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Cundo el usuario marca una notificación del listado de notificaciones, ya visto previamente, se le permitirá al usuario enviar la notificación para el listado de notificaciones Leídas, o Cancelar la operación.</p> 	

Tabla 17: HU Cerrar notificación

Número: 8	Nombre del requisito: Cerrar notificación.
Programador: Raymari Sanjudo Cabrera.	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 1 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir cerrar una notificación de un curso o del sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para cerrar una notificación hay que: - Estar autenticado en el sistema con el rol X.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Se muestra al usuario una notificación que se le fue enviada, presionando encima de una notificación se podrá eliminar la misma o cancelar la operación. Si selecciona la opción Eliminar se eliminará dicha notificación cerrándose la misma.</p>	



Tabla 18: HU Comentar en el foro

Número: 14	Nombre del requisito: Comentar en el foro.
Programador: Raymari Sanjudo Cabrera.	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 días

Riesgo en Desarrollo: N/A

Tiempo Real:

Descripción:

1- Objetivo:

Permitir comentar en un foro en el sistema.

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

Para comentar en un foro hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos un foro.

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

N/A

4- Flujo de la acción a realizar:

Cuando el usuario seleccione un tema X de un foro Y, aparecerá un campo de texto, permitiéndole agregar la información que desee, enviando la misma hacia la Plataforma.

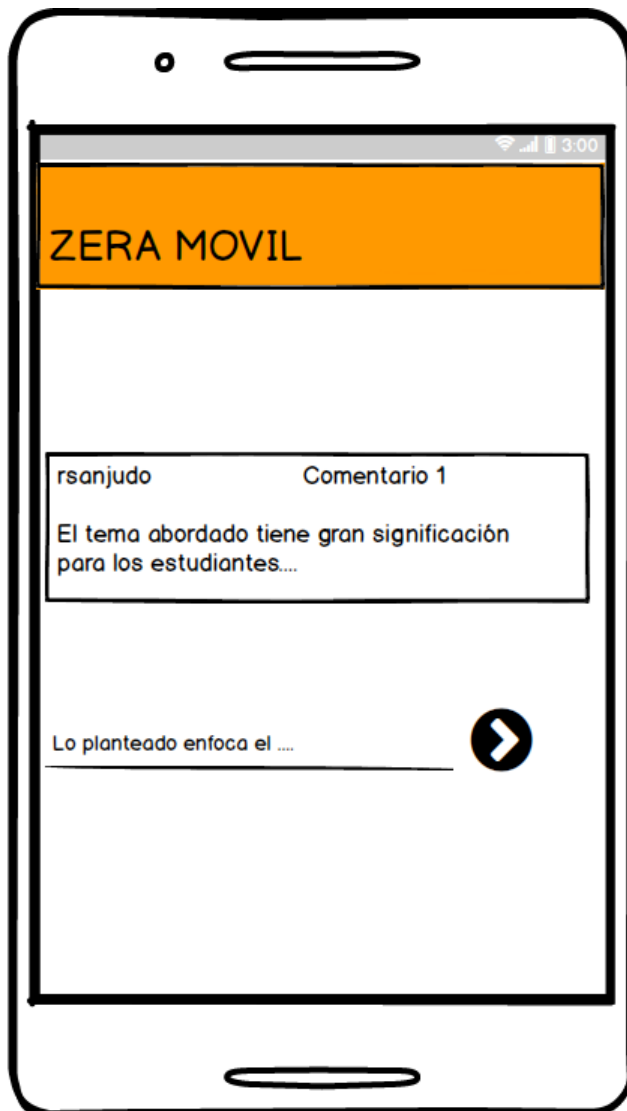


Tabla 19: HU Visualizar comentarios del foro

Número: 15	Nombre del requisito: Visualizar comentarios del foro.
Programador: Raimary Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 1 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir consultar los comentarios de un foro en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para ver detalles de un comentario hay que: - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un foro.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Inicialmente se muestra al usuario el listado de foros del curso elegido, del cual podrá ver sus comentarios seleccionando el tema del foro deseado.</p>	



Tabla 20: HU Eliminar comentarios del foro

Número: 16	Nombre del requisito: Eliminar comentarios del foro.
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción:	
1- Objetivo:	
Permitir eliminar un comentario en un foro en el sistema.	

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

Para eliminar un comentario hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos un foro.

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

N/A

4- Flujo de la acción a realizar:

Para eliminar un comentario de un foro, el sistema muestra un listado de los comentarios realizados en dicho foro, presionando un comentario se podrá eliminar el mismo, o cancelar la operación.

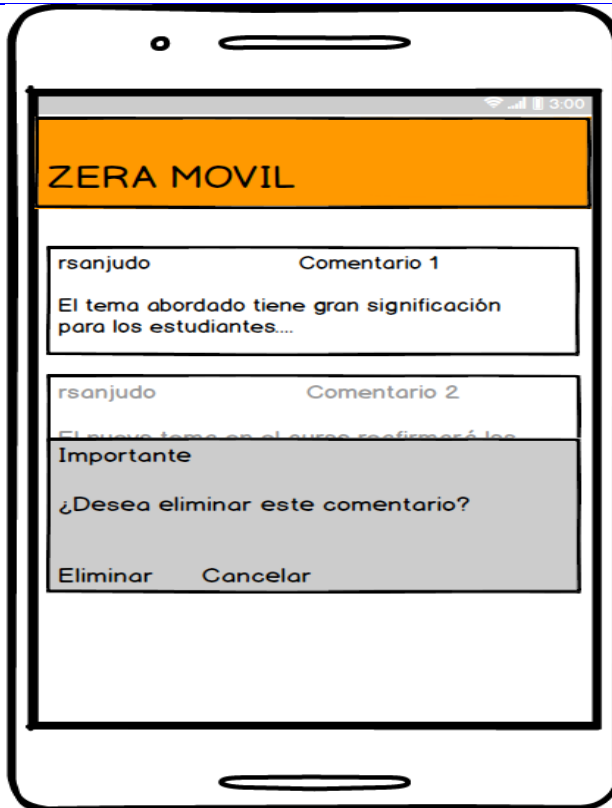


Tabla 21: HU Listar temas de los foros

Número: 17	Nombre del requisito: Listar temas de los foros.
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era

Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir listar los temas de un foro en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos): Para listar los temas hay que: - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un foro. 3- Comportamientos válidos y no válidos (flujo central y alternos): N/A 4- Flujo de la acción a realizar: Cuando el usuario selecciona la opción “Listar temas” aparecen todos los temas registrados en el sistema, dándole la posibilidad de elegir el tema que desee. Además, el usuario tiene la posibilidad de ver detalles de un tema, seleccionando la opción “ver”.	
Prototipo:	

Tabla 22: HU Listar foros

Número: 18	Nombre del requisito: Listar foros
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir listar los foros de un curso en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos):	

Para listar foros hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos un curso X.
- Debe existir al menos un foro Y de un curso X.

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

N/A

4- Flujo de la acción a realizar:

Cuando el usuario selecciona la opción “Foros” aparecen todos los foros registrados en el sistema, dándole la posibilidad de elegir los foros que desee.

Prototipo:

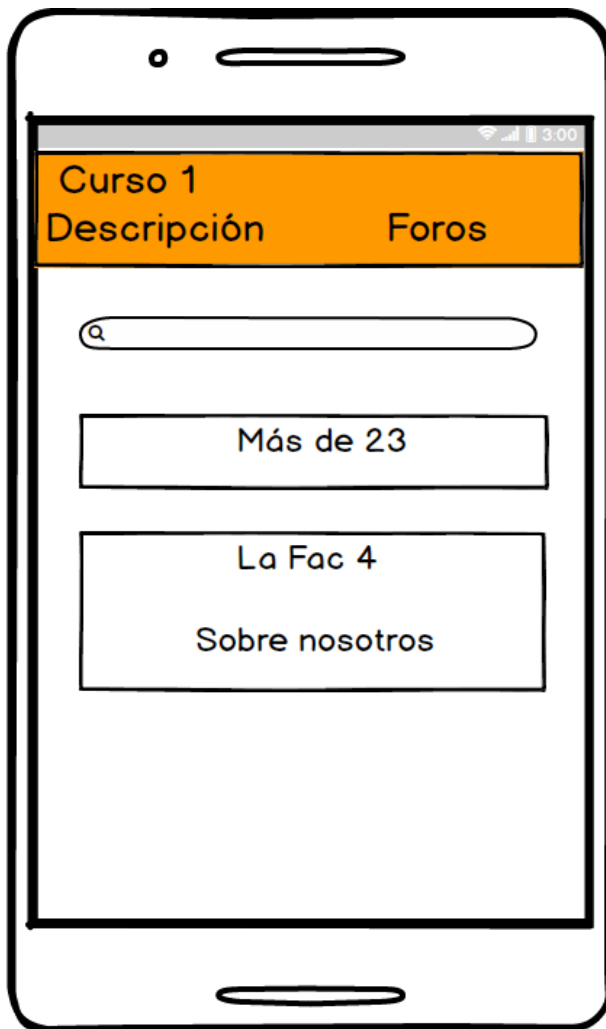


Tabla 23: HU Buscar foros

Número: 19	Nombre del requisito: Buscar foros.
Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo: Permitir buscar los foros de un curso en el sistema 2- Acciones para lograr el objetivo (precondiciones y datos): Para buscar en un listado de foros hay que: - Tener en cuenta los siguientes datos: nombre y descripción. - Estar autenticado en el sistema con el rol X. - Debe existir en el sistema al menos un curso. 3- Comportamientos válidos y no válidos (flujo central y alternos): El campo búsqueda es obligatorio: puede ser (nombre del foro). 4- Flujo de la acción a realizar: Cuando el usuario inserta un criterio de búsqueda (asunto del foro), el sistema muestra un listado de los foros que cumplen con los criterios establecidos, en caso de no encontrar ningún elemento el sistema muestra un mensaje de información.	

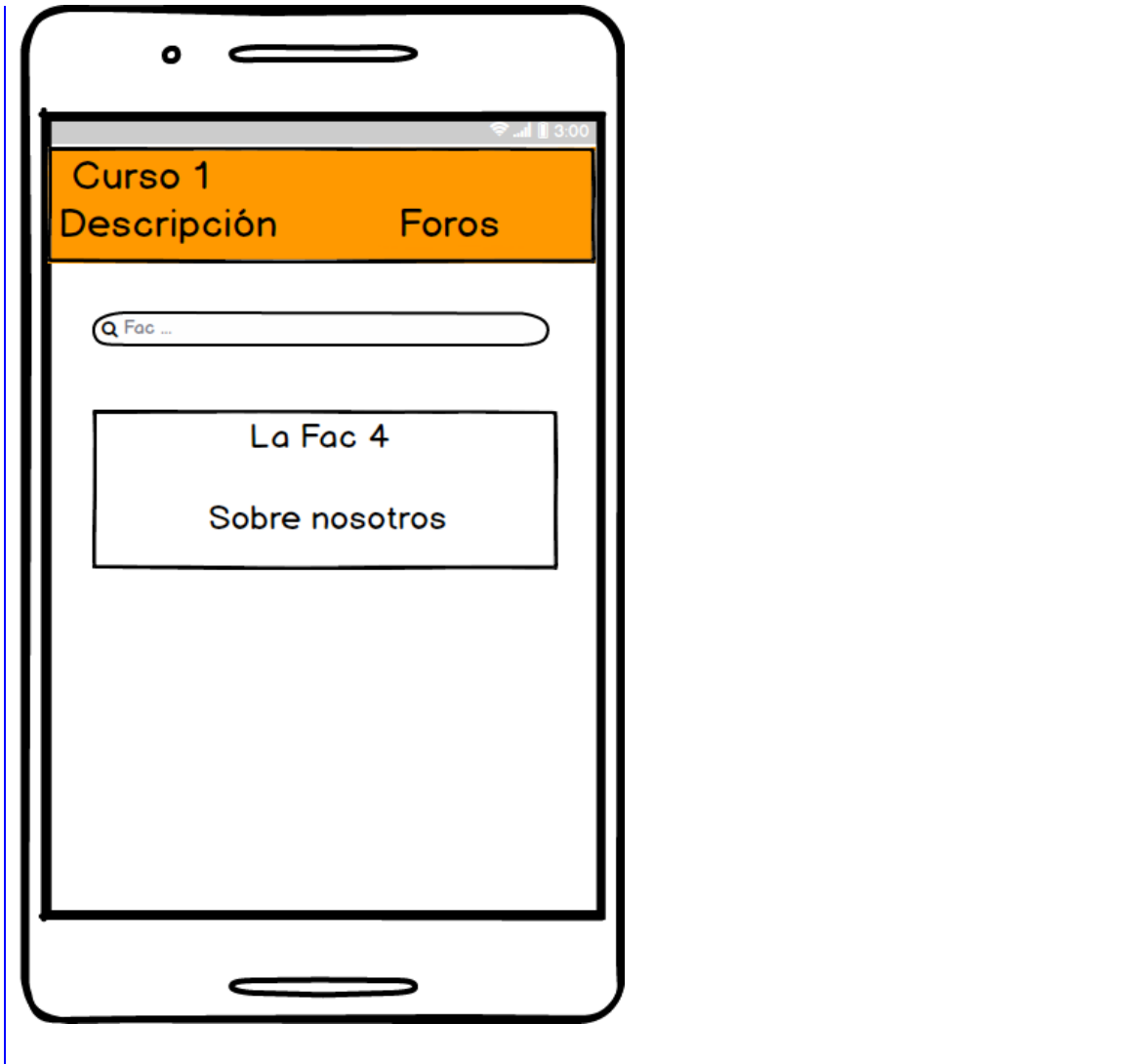


Tabla 24: HU Descargar un curso

Número: 13	Nombre del requisito: Descargar un curso
Programador: Raimary Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
Descripción: 1- Objetivo:	

Permitir descargar un curso del sistema.

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

Para descargar un curso hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos un curso.

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

N/A

4- Flujo de la acción a realizar:

Cuando el usuario seleccione la opción “descargar” de un curso en el que se encuentre matriculado, se descargarán sus contenidos, estos pueden ser:

- foros
- actividades
- recursos

Observaciones: los cursos podrán ser descargados por el usuario facilitándole la consulta de estos en cualquier momento sin necesidad de encontrarse *on-line*.

Tabla 25: HU Mostrar contenidos de un curso de manera off-line

Número: 21	Nombre del requisito: Mostrar contenidos de un curso de manera off-line.
Programador: Raymari Sanjudo Cabrera.	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: N/A	Tiempo Real: 2 días

Descripción:**1- Objetivo:**

Permitir consultar los contenidos de un curso en el sistema.

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

Para ver detalles de los contenidos de un curso hay que:

- Estar autenticado en el sistema con el rol X.
- Debe existir en el sistema al menos un curso.

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

N/A

4- Flujo de la acción a realizar:

Inicialmente se muestra al usuario un listado con los contenidos (materiales) de un curso que han sido incluidos en el sistema. Una vez seleccionado un contenido, podrá ver sus datos seleccionando el mismo.

Observaciones: los contenidos de un curso pueden ser de varios tipos (programa del curso, actividades del curso, entre otros).

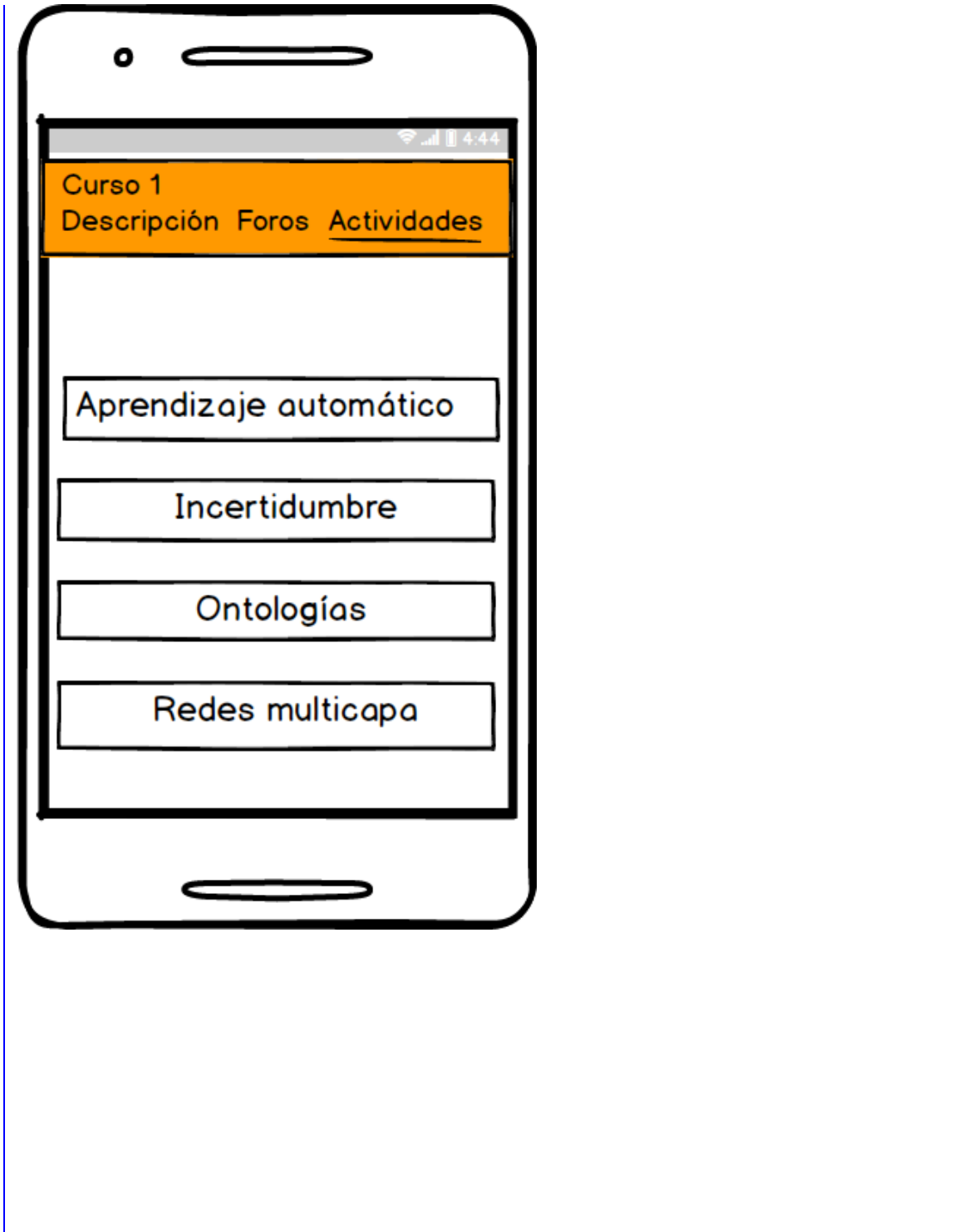


Tabla 26: HU Cambiar idioma

Número: 22	Nombre del requisito: Cambiar idioma.

Programador: Raymari Sanjudo Cabrera	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 2 días
Riesgo en Desarrollo: N/A	Tiempo Real:
<p>Descripción:</p> <p>1- Objetivo: Permitir el idioma en el sistema en el sistema</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para cambiar el idioma en el sistema hay que: - Estar autenticado en el sistema con el rol X.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): N/A</p> <p>4- Flujo de la acción a realizar: Cuando el usuario seleccione la opción "idioma", se mostrarán los idiomas disponibles en sistema, luego el usuario seleccionara el idioma que desee.</p>	

Anexo 2: Diagramas de Clases del Diseño restantes

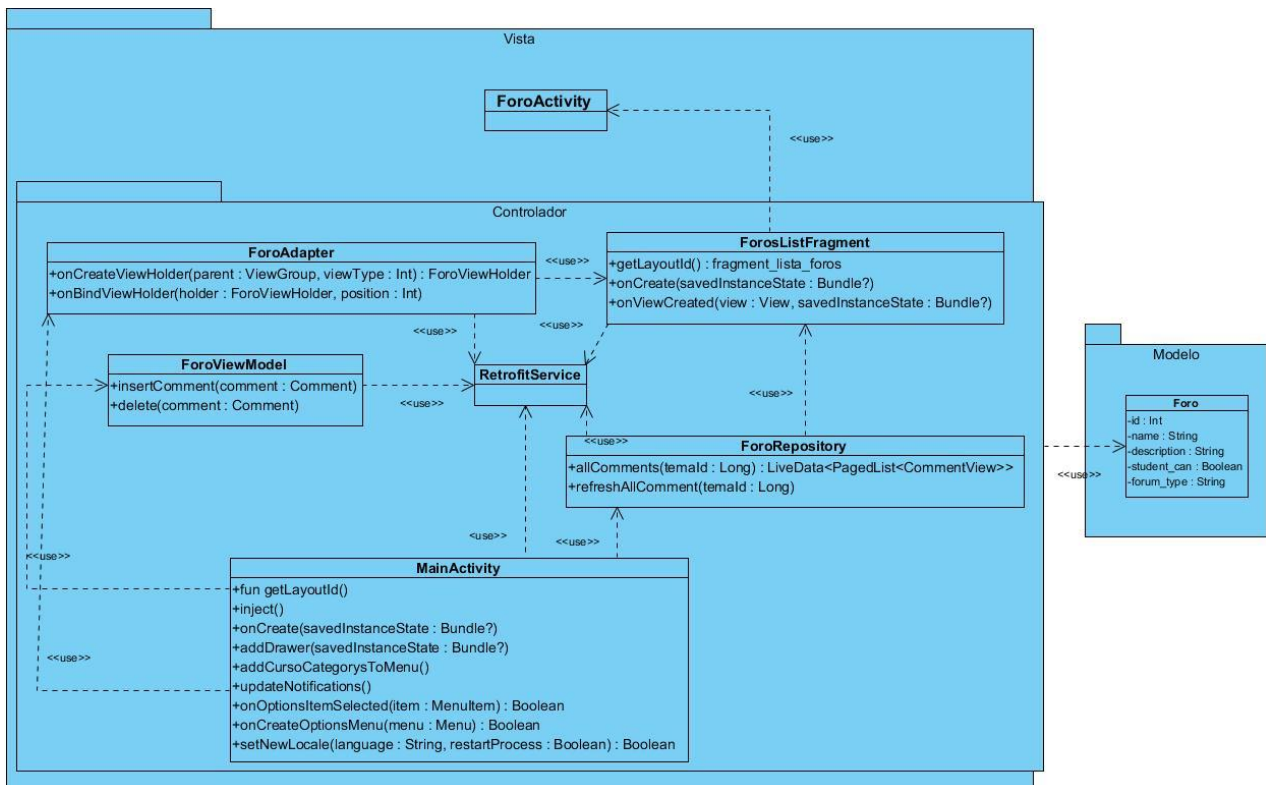


Fig. 8 Diagrama de Clases del Diseño del Paquete Foros

<p>EC 2.2</p>	<p>El usuario selecciona una notificación del listado de notificaciones.</p>	<p>El sistema muestra ella notificación seleccionada, de la misma se podrán observar los siguientes datos:</p> <ul style="list-style-type: none"> • Asunto de la notificación • Fecha de la notificación 	<p>Menú/Notificaciones/Notificación</p>
----------------------	------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------

Tabla 28: Caso de prueba perteneciente al RF Eliminar Notificación.

Escenario	Descripción	Respuesta del sistema	Flujo central
<p>EC 3.1</p>	<p>Se selecciona en el menú de opciones la opción de Notificaciones.</p>	<p>El sistema muestra un listado con todas las notificaciones que le fueron enviadas al usuario.</p>	<p>Menú/Notificaciones</p>

EC 3.2	Se selecciona la notificación que se desea eliminar.	El sistema muestra la notificación con la opción “eliminar” y la opción “cancelar”.	Menú/Notificaciones/Notificación
EC 3.3	Se selecciona la opción “Eliminar”	El sistema elimina la notificación.	Menú/Notificaciones/Eliminar
EC 3.4	Se selecciona la opción “Cancelar”	Se regresa al listado de notificaciones.	Menú/Notificaciones/Cancelar

Tabla 29: Caso de prueba perteneciente al RF Buscar Notificación.

Escenario	Descripción	Asunto de la notificación	Respuesta del sistema	Flujo central
EC 4.1	Selecciona en el menú de opciones la opción de Notificaciones.		El sistema muestra un listado con todas las notificaciones que le fueron enviadas al usuario.	Menú/Notificaciones

EC 4.2	El usuario inserta un criterio de búsqueda: asunto de la notificación.	V	El sistema muestra un listado de las notificaciones que cumplen con los criterios establecidos.	Menú/Notificaciones/ BuscarNotificación
EC 4.3	El usuario inserta un criterio de búsqueda incorrecto.	I	El sistema no muestra ninguna notificación.	Menú/Notificaciones/ BuscarNotificación

Tabla 30: Caso de prueba perteneciente al RF Marcar Notificación como no leída.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 5.1.	Se selecciona en el menú de opciones la opción de Notificaciones.	El sistema muestra un listado con todas las notificaciones que le fueron enviadas al usuario.	Menú/Notificaciones

EC 5.2	El usuario selecciona una notificación del listado de notificaciones.	El sistema muestra la notificación seleccionada con la opción "Marcar como no leída" y la opción "Cancelar"	Menú/Notificaciones/Notificación
EC 5.3	Se selecciona la opción "Marcar como no leída"	El sistema añadirá la notificación al listado de notificaciones no leídas.	Menú/Notificaciones/Notificación/ MarcarNotificaciónNoLeída

Tabla 31: Caso de prueba perteneciente al RF Marcar Notificación como leída.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 6.1	Se selecciona en el menú de opciones la opción de Notificaciones.	El sistema muestra un listado con todas las notificaciones que le fueron enviadas al usuario.	Menú/Notificaciones

EC 6.2	El usuario selecciona una notificación del listado de notificaciones.	El sistema muestra la notificación seleccionada con la opción "Marcar como leída" y la opción "Cancelar"	Menú/Notificaciones/Notificación
EC 6.3	Se selecciona la opción "Marcar como leída"	El sistema añadirá la notificación al listado de notificaciones leídas.	Menú/Notificaciones/Notificación/ MarcarNotificaciónLeída

Tabla 32: Caso de prueba perteneciente al RF Comentar en el foro.

Escenario	Descripción	Asunto de la notificación	Respuesta del sistema	Flujo central
EC 7.1	El usuario selecciona un tema de un foro de un curso en el que se encuentre matriculado.		El sistema muestra un campo de texto permitiéndole agregar la información que desee.	Menú/Curso/Foro

EC 7.2	El usuario agrega la información deseada y envía la misma a la Plataforma.	V	El sistema muestra el comentario enviado junto al nombre de usuario que realizó el mismo.	Menú/Curso/Foro/ Comentar
EC 7.3	El usuario no agrega información alguna en el campo de texto.	I	El sistema no envía ninguna información a la Plataforma.	Menú/Curso/Foro/ Comentar

Tabla 33: Caso de prueba perteneciente al RF Visualizar Comentarios del Foro.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 8.1	El usuario selecciona un tema de un foro de un curso en el que se encuentre matriculado.	El sistema muestra los comentarios de ese foro que fueron enviados por el usuario.	Menú/Curso/Foro/Comentar

Tabla 34: Caso de prueba perteneciente al RF Eliminar Comentario del Foro.

Escenario	Descripción	Respuesta	Flujo central
------------------	--------------------	------------------	----------------------

		del sistema	
EC 9.1	El usuario selecciona un tema de un foro de un curso en el que se encuentre matriculado.	El sistema muestra los comentarios de ese foro que fueron enviados por el usuario.	Menú/Curso/Foro
EC 9.2	Se selecciona el comentario que se desea eliminar.	El sistema muestra el comentario con la opción "eliminar" y la opción "cancelar".	Menú/Curso/Foro/Comentar
EC 9.3	Se selecciona la opción "Eliminar"	El sistema elimina el comentario.	Menú/Curso/Foro/Comentar/ Eliminar
EC 9.4	Se selecciona la opción "Cancelar"	Se regresa al listado de comentarios.	Menú/Curso/Foro/Comentar/ Cancelar

Tabla 35: Caso de prueba perteneciente al RF Buscar Foro.

Escenario	Descripción	Asunto de la notificación	Respuesta del sistema	Flujo central
EC 10.1	El usuario selecciona un tema de un foro de un curso en el que se encuentre matriculado.		El sistema muestra los comentarios de ese foro que fueron enviados por el usuario.	Menú/Curso/Foro
EC 10.2	El usuario inserta un criterio de búsqueda: asunto del foro.	V	El sistema muestra un listado de los foros que cumplen con los criterios establecidos.	Menú/Curso/Foro/ Buscar
EC 10.3	El usuario inserta un criterio de búsqueda incorrecto.	I	El sistema no muestra ningún foro.	Menú/Curso/Foro/ Buscar

Tabla 36: Caso de prueba perteneciente al RF Listar Foro.

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

		sistema	
EC 11.1 Listar Foro	Se selecciona la opción "Foros", dentro de un curso matriculado.	El sistema muestra un listado con todos los foros del curso correspondiente.	Menú/Curso/Foro

Tabla 37: Caso de prueba Listar tareas de forma off-line

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 12.1 Listar tareas de forma off-line.	Se selecciona la opción "Tareas", dentro de un curso matriculado.	El sistema muestra un listado con todas las tareas del curso correspondiente.	Menú/Curso/Tarea

Tabla 38: Caso de prueba Mostrar tareas de forma off-line.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 13.1 Mostrar tareas de forma off-line.	Se selecciona la opción "Tareas", dentro de un curso	El sistema muestra un listado con todas las tareas del curso correspondiente.	Menú/Curso/Tarea

	matriculado.		
EC 13.2	Se selecciona una tarea del listado de tareas.	El sistema muestra la tarea seleccionada con su correspondiente descripción.	Menú/Curso/Tarea

Tabla 39: Caso de prueba Responder tareas de forma off-line.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 14.1 Responder tareas de forma off-line.	Se selecciona la opción "Tareas", dentro de un curso matriculado.	El sistema muestra un listado con todas las tareas del curso correspondiente.	Menú/Curso/Tarea
EC 14.2	Se selecciona una tarea del listado de tareas.	El sistema muestra la tarea seleccionada con su correspondiente descripción y mostrando la opción "Cancelar" y la opción "Guardar".	Menú/Curso/Tarea

EC 14.3	Se selecciona la opción "Guardar"	El sistema guarda la tarea resuelta.	Menú/Curso/Tarea/Guardar
EC 14.4	Se selecciona la opción "Cancelar"	Se regresa al listado de tareas.	Menú/Curso/Tarea/Cancelar

Tabla 40: Caso de prueba Buscar tareas de forma off-line.

Escenario	Descripción	Nombre de la tarea	Descripción de la tarea	Respuesta del sistema	Flujo central
EC 15.1	Selecciona en el curso que se encuentre matriculado la opción "Tareas"			El sistema muestra un listado con todas las tareas que se encuentren en ese curso.	Menú/Tareas

<p>EC 15.2</p>	<p>El usuario inserta un criterio de búsqueda: nombre o descripción de la tarea.</p>	<p>V</p>		<p>El sistema muestra un listado de las tareas que cumplen con los criterios establecidos.</p>	<p>Menú/Tareas/ BuscarTareas</p>
<p>EC 15.3</p>	<p>El usuario inserta un criterio de búsqueda incorrecto.</p>	<p>I</p>		<p>El sistema no muestra ninguna tarea.</p>	<p>Menú/Tareas/ BuscarTareas</p>