



UNIVERSIDAD DE CUENCA

FACULTAD DE INGENIERÍA

ESCUELA DE INFORMÁTICA

“METODOLOGÍA DE ENSEÑANZA CONSTRUCTIVISTA PARA GRUPOS NUMEROSOS A NIVEL UNIVERSITARIO CON SOPORTE EN HERRAMIENTAS WEB Y DISPOSITIVOS MÓVILES”

Tesis previa a la obtención
del título de Ingeniero de
Sistemas

AUTORES:

Pedro Andrés Guerrero Pozo
Lissette Maritza Muñoz Guillén
Tania Jhomara Palacios Crespo

DIRECTOR:

Jorge Mauricio Espinoza Mejía

Cuenca- Ecuador

Octubre 2012



UNIVERSIDAD DE CUENCA

RESUMEN

El uso masivo de los dispositivos móviles por parte de jóvenes estudiantes que cursan una carrera universitaria en la Facultad de Ingeniería de la Universidad de Cuenca, los convierte en una opción interesante como herramienta de distribución y acceso a contenidos de aprendizaje a nivel de educación superior.

El presente proyecto está enfocado en modernizar el proceso de enseñanza-aprendizaje a través del modelo **Constructivista** en grupos numerosos de estudiantes, mediante el uso de dispositivos móviles; estableciendo medidas para minimizar los impactos negativos que pudieran presentarse en este tipo de grupos, aprovechando las ventajas que puede brindar el uso de las herramientas antes mencionadas.

El primer capítulo es una visión introductoria en donde se justifica y describe brevemente el alcance del proyecto. El segundo capítulo es una investigación para determinar la plataforma móvil más adecuada para el proyecto. En el tercer capítulo se definen las características de la metodología constructivista y los objetos de aprendizaje. En el cuarto capítulo se listan los detalles del prototipo de aprendizaje que se pretende implementar. En el quinto capítulo, se presenta el desarrollo de la aplicación que da soporte al aprendizaje móvil. En el sexto capítulo se describe la generación del contenido de la materia piloto Cálculo Diferencial y finalmente, en el séptimo capítulo, se detallan las conclusiones y recomendaciones del presente trabajo de investigación.

PALABRAS CLAVE: Constructivismo, Android, m-learning, móvil, aprendizaje.



UNIVERSIDAD DE CUENCA

TABLA DE CONTENIDOS

AGRADECIMIENTOS	12
DEDICATORIA.....	13
CAPITULO 1:	14
1.1 IDENTIFICACIÓN DEL PROBLEMA.....	15
1.2 JUSTIFICACIÓN	15
1.3 ALCANCE	16
1.4 OBJETIVO GENERAL.....	16
1.5 OBJETIVOS ESPECÍFICOS	16
CAPITULO 2:	17
2.1 INTRODUCCIÓN	18
2.2 CRITERIOS DE COMPARACIÓN.....	19
2.3 SISTEMA OPERATIVO ANDROID	23
2.4 SISTEMA OPERATIVO iOS.....	35
2.5 iOS Y ANDROID EN UN CONTEXTO EDUCATIVO.....	45
2.6 COMPARACION ENTRE LOS SISTEMAS OPERATIVOS iOS Y ANDROID.....	48
2.7 RESUMEN DEL CAPITULO.....	63
CAPITULO 3:	66
3.1 INTRODUCCIÓN	67
3.2 METODOLOGÍA DE APRENDIZAJE CONSTRUCTIVISTA	68
3.3 OBJETOS DE APRENDIZAJE	80
3.4 R RESUMEN DEL CAPITULO	84
CAPITULO 4:	85
4.1 INTRODUCCIÓN	86
4.2 DESCRIPCIÓN DEL PROTOTIPO.....	87
4.3 REQUISITOS FUNCIONALES.....	87
4.4 REQUISITOS NO FUNCIONALES.....	87
4.5 ANÁLISIS DEL PROTOTIPO	88
4.6 RESUMEN DEL CAPITULO.....	98
CAPITULO 5:	99
5.1 INTRODUCCIÓN	100
5.2 DESCRIPCIÓN DE LA APLICACIÓN.....	100
5.3 DISEÑO DE LA APLICACIÓN.....	101
5.4 CONEXIÓN ANDROID-MOODLE	106
5.5 PERSISTENCIA DE DATOS.....	114
5.6 OPERACIONES EN SEGUNDO PLANO	118
5.7 RESUMEN DEL CAPITULO.....	119



UNIVERSIDAD DE CUENCA

CAPITULO 6:	120
6.1 INTRODUCCIÓN	121
6.2 DESCRIPCIÓN GENERAL DE LOS CONTENIDOS	121
6.3 DISEÑO DE LOS CONTENIDOS Y LOS ODAS	122
6.4 MAPA DE PROCESOS PARA LOS CONTENIDOS	123
6.5 DISEÑO DE LOS OBJETOS DE APRENDIZAJE	129
6.6 HERRAMIENTAS UTILIZADAS	133
6.7 CREACIÓN DE LOS CONTENIDOS	139
6.8 IMPLEMENTACIÓN EN LA PLATAFORMA MOODLE	147
6.9 RESUMEN DEL CAPITULO	152
CAPITULO 7:	152
7.1 INTRODUCCIÓN	154
7.2 CONCLUSIONES	154
7.3 RECOMENDACIONES	155
7.4 REFERENCIAS	156
ANEXO A: ENCUESTA SOBRE EL USO DE DISPOSITIVOS MÓVILES.	159
ANEXO B: MANUAL DE INSTRUCCIONES DE LA APLICACIÓN CLIENTE DE ANDROID PARA MOODLE EVIAM	163

INDICE DE FIGURAS

Figura 2.3-A Arquitectura del Sistema Operativo Móvil Android	24
Figura 2.4-A Arquitectura del sistema iOS	36
Figura 4.5-A Diagrama de caso de uso Ingresar a la aplicación	93
Figura 4.5-B Diagrama caso de uso Acceder a los contenidos de la materia	95
Figura 4.5-C Diagrama caso de uso Mantenimiento del contenido de la materia	96
Figura 4.5-D Resumen de casos de uso	97
Figura 5.3-A Arquitectura General del Sistema	102
Figura 5.3-B Clases que componen el Modelo de la Aplicación Móvil	103
Figura 5.3-C Clases que conforman la Vista de la Aplicación Móvil	104
Figura 5.3-D Clases que conforman el Controlador de la Aplicación Móvil	105
Figura 5.4-A Proceso de Autenticación OAuth 2.0	108
Figura 5.4-B Credenciales previamente almacenadas en Moodle	109
Figura 5.4-C Solicitud de Inicio de sesión de Moodle	109
Figura 5.4-D Solicitud de Autorización	109
Figura 5.4-E Funciones del servicio Web EviamWS	110
Figura 5.5-A Diagrama de Base de Datos de la Aplicación	115
Figura 5.5-B Opción de habilitar la caché	117



UNIVERSIDAD DE CUENCA

Figura 5.5-C Opción de tiempo de validez de la caché	118
Figura 5.6-A Mensaje de error en Android	118
Figura 5.6-B Operaciones en segundo plano	119
Figura 6.3-A Esquema de los contenidos de Cálculo Diferencial- Cap.1 y Cap.2	123
Figura 6.4-A Mapa de procesos del Cap. 1 de Cálculo Diferencial	126
Figura 6.4-B Mapa de procesos del Cap. 2 de Cálculo Diferencial	128
Figura 6.5-A Objeto de Aprendizaje 1 (Cap. 1 – Parte 1).....	130
Figura 6.5-B Objeto de Aprendizaje 2 (Cap. 1 – Parte 2).....	131
Figura 6.5-C Objeto de Aprendizaje 3 (Cap. 2 – Parte 1).....	132
Figura 6.5-D Objeto de Aprendizaje 4 (Cap. 2 – Parte 2).....	133
Figura 6.6-A Estructura de la herramienta ExeLearning	134
Figura 6.6-B Principales elementos de la herramienta ExeLearning.....	135
Figura 6.6-C Estructura de la herramienta Geogebra	136
Figura 6.6-D Estructura de la herramienta Illustrator.....	137
Figura 6.6-E Estructura de la herramienta Flash.....	139
Figura 6.7-A Objetivos y Pre-Conocimiento de un ODA.....	140
Figura 6.7-B Esquema de cada sección.....	140
Figura 6.7-C Conceptos y notaciones matemáticas	141
Figura 6.7-D Conceptos y notaciones matemáticas	142
Figura 6.7-E Inclusión de Tablas.....	142
Figura 6.7-F Ejemplos desarrollados paso a paso	143
Figura 6.7-G Imagen perteneciente a un link seleccionado	144
Figura 6.7-H Explicación de Ejercicios	144
Figura 6.7-I Solución Gráfica de Ejercicios	145
Figura 6.7-J Ejemplo de animación en formato .Gif	145
Figura 6.7-K Modelo de autoevaluación – Texto	146
Figura 6.7-L Modelo de autoevaluación- Gráficas.....	147
Figura 6.8-A Edición de un curso de Cálculo Diferencial en Moodle.....	148
Figura 6.8-B Añadir una actividad o recurso en Moodle.....	148
Figura 6.8-C Agregar un paquete SCORM dentro de un curso.....	149
Figura 6.8-D Configuraciones adicionales para agregar un paquete SCORM.....	149
Figura 6.8-E Visualización de los paquetes SCORM	150
Figura 6.8-F Manipulación de un objeto SCORM (orden jerárquico).....	151
Figura 6.8-G Manipulación de un objeto SCORM (orden no jerárquico)	151



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

Fundada en 1867

Yo, Tania Jhomara Palacios Crespo, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniera de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.


Tania Jhomara Palacios Crespo
010415561-9

Cuenca Patrimonio Cultural de la Humanidad. Resolución de la UNESCO del 1 de diciembre de 1999

Av. 12 de Abril, Ciudadela Universitaria, Teléfono: 405 1000, Ext.: 1311, 1312, 1316

e-mail cdjbv@ucuenca.edu.ec casilla No. 1103

Cuenca - Ecuador



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

Fundada en 1867

Yo, Tania Jhomara Palacios Crespo, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Tania Jhomara Palacios Crespo
CI: 010415561-9

Cuenca Patrimonio Cultural de la Humanidad. Resolución de la UNESCO del 1 de diciembre de 1999

Av. 12 de Abril, Ciudadela Universitaria, Teléfono: 405 1000, Ext.: 1311, 1312, 1316
e-mail cdjbv@ucuenca.edu.ec casilla No. 1103

Cuenca - Ecuador



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

Fundada en 1867

Yo, Lissette Maritza Muñoz Guillén, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniera de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Lissette Maritza Muñoz Guillén
010583261-2

Cuenca Patrimonio Cultural de la Humanidad. Resolución de la UNESCO del 1 de diciembre de 1999

Av. 12 de Abril, Ciudadela Universitaria, Teléfono: 405 1000, Ext.: 1311, 1312, 1316

e-mail cdjbv@ucuenca.edu.ec casilla No. 1103

Cuenca - Ecuador



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

Fundada en 1867

Yo, Lissette Maritza Muñoz Guillén, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.

Lissette Maritza Muñoz Guillén
CI: 010583261-2

Cuenca Patrimonio Cultural de la Humanidad. Resolución de la UNESCO del 1 de diciembre de 1999

Av. 12 de Abril, Ciudadela Universitaria, Teléfono: 405 1000, Ext.: 1311, 1312, 1316

e-mail cdjbv@ucuenca.edu.ec casilla No. 1103

Cuenca - Ecuador



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

Fundada en 1867

Yo, Pedro Andrés Guerrero Pozo, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.


Pedro Andrés Guerrero Pozo
010548836-5



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

Fundada en 1867

Yo, Pedro Andrés Guerrero Pozo, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor/a.


Pedro Andrés Guerrero Pozo

Ci: 010548836-5



UNIVERSIDAD DE CUENCA

AGRADECIMIENTOS

Al culminar exitosamente nuestro trabajo de investigación queremos expresar nuestro más sincero agradecimiento a:

- *Nuestro director de tesis, Ing. Mauricio Espinoza, por su acertada guía, paciencia y dedicación durante el desarrollo del presente trabajo de investigación.*
- *Los miembros del proyecto, Ing. Patricio Guerrero, Ing. Pablo Vanegas, Ing. Magali Mejía, Ing. Elina Ávila, Lcda. Rosita Ávila, por su apoyo, predisposición y colaboración para llevar adelante este proyecto.*
- *El Diseñador Ángel Guapisaca, por su contribución en el diseño de la parte gráfica de nuestra tesis.*
- *Los Ingenieros Carlos Romero, Guillermo Martínez y Adrián Tola, por su desinteresada ayuda en la generación y revisión de los contenidos de la materia piloto Cálculo Diferencial.*



UNIVERSIDAD DE CUENCA

DEDICATORIA

A nuestros padres, amigos, maestros y familiares por todo su apoyo y enseñanzas a lo largo de nuestra carrera universitaria.



UNIVERSIDAD DE CUENCA

CAPITULO 1: **“Marco Contextual”**



UNIVERSIDAD DE CUENCA

1.1 IDENTIFICACIÓN DEL PROBLEMA

La gran adopción por parte de los estudiantes de los dispositivos móviles y las herramientas Web las convierten en un medio interesante para la distribución y presentación de contenidos educativos, sin embargo, actualmente en la Facultad de Ingeniería de la Universidad de Cuenca no se aprovecha completamente el potencial de dichas herramientas. El presente proyecto busca implementar una estructura basada en el modelo constructorista que permita modernizar el proceso de enseñanza-aprendizaje en grupos numerosos de estudiantes y establecer medidas para minimizar los impactos negativos que pudieran presentarse en este tipo de grupos, aprovechando las ventajas que puede brindar el uso de las herramientas antes mencionadas.

1.2 JUSTIFICACIÓN

La Facultad de Ingeniería de la Universidad de Cuenca ha iniciado un proceso para disminuir las clases repetitivas a través de la conformación de grupos numerosos de estudiantes, inicialmente de los primeros años e incrementalmente en años superiores, medida que permitiría mejorar la eficiencia de la utilización de los recursos de la Facultad pero que puede generar un impacto negativo en los logros del aprendizaje de los estudiantes. En este contexto es imperiosa la necesidad de estudiar e investigar metodologías que permitan manejar estos grupos numerosos con el soporte de procesos de enseñanza-aprendizaje adecuados. En base a lo expuesto es imprescindible buscar y aplicar metodologías que incrementen el rendimiento académico de los estudiantes a pesar de que éstos formen parte de grupos numerosos de aprendizaje, siendo oportuno contar con herramientas, las cuales los estudiantes estén más familiarizados, que les permita obtener un marco tecnológico de aprendizaje que facilite el acceso dinámico a herramientas de enseñanza-aprendizaje. Con ello el estudiante dispondrá de material suficiente y adecuado para que pueda llevar adelante tareas de auto-aprendizaje bajo la guía del profesor.

Tomando en cuenta la familiaridad de la gente joven con los dispositivos electrónicos móviles, éstos podrían convertirse en un medio adecuado para: 1) diseñar e implementar material electrónico que facilite el aprendizaje de los conceptos fundamentales de una materia determinada, 2) desarrollar herramientas que faciliten una comunicación en línea como soporte para llevar adelante trabajo cooperativo entre los estudiantes, y 3) establecer mecanismos que permitan difundir en forma clara y precisa información sobre la relevancia de las materias en el marco de los logros del aprendizaje que se pretenden alcanzar al final de una carrera.



UNIVERSIDAD DE CUENCA

1.3 ALCANCE

Tomando en cuenta los puntos anteriores, se ha definido el siguiente alcance:

- Estudio y selección de las características del modelo de enseñanza constructorista aplicables al proyecto.
- Estudio y análisis de las tecnologías para dispositivos móviles disponibles en el mercado nacional y selección de la plataforma adecuada para el proyecto.
- Revisión de experiencias en el desarrollo de herramientas móviles aplicadas en educación y de las características que éstas poseen para dar soporte a metodologías constructoristas.
- Diseño de la materia piloto (cálculo diferencial), contenidos y la forma de presentar dicho contenido a los estudiantes, los cuales estarán basados en la metodología constructorista. Se encuentra fuera del alcance la forma de evaluación, talleres y cronogramas.
- Desarrollo de los contenidos de aprendizaje de los dos primeros capítulos de la materia piloto.
- Diseño, análisis y desarrollo de un primer prototipo de aprendizaje para la materia piloto sobre la plataforma Moodle, con soporte en dispositivos móviles.
- Pruebas a nivel del prototipo, está fuera del alcance lo que corresponde a pruebas de amigabilidad, rendimiento e impacto de la metodología en los estudiantes.

1.4 OBJETIVO GENERAL

Implementar un prototipo de enseñanza basada en una metodología constructorista a grupos numerosos de estudiantes.

1.5 OBJETIVOS ESPECÍFICOS

Una vez conocido el objetivo general del proyecto, a continuación detallamos los objetivos específicos a cumplir:

1. Analizar y evaluar la tecnología móvil como elemento de apoyo a los procesos de enseñanza-aprendizaje.
2. Diseñar los contenidos, estructura y la forma en que se imparte la materia piloto cálculo diferencial bajo un enfoque constructorista.
3. Desarrollar una aplicación móvil que permita el acceso a los contenidos de la materia piloto de los primeros años de las carreras de la Facultad de Ingeniería.



UNIVERSIDAD DE CUENCA

CAPITULO 2:

**“Investigación y comparación entre los SO móviles iOS y Android:
determinación de la plataforma tecnológica más adecuada para el
Proyecto.”**



UNIVERSIDAD DE CUENCA

2.1 INTRODUCCIÓN

La tecnología es la base sobre la cual se asienta el desarrollo y evolución de las herramientas informáticas que pretenden dar soporte a la implementación de una metodología de enseñanza-aprendizaje. Escoger la plataforma tecnológica apropiada, marcará la diferencia entre su rápida adopción y difusión en la comunidad educativa o un indeseable costo de reingeniería que implica realizar un cambio sobre la marcha. Por ello, es importante analizar y seleccionar la mejor alternativa, basándose en criterios técnicos, así como también de uso, familiaridad y penetración de dicha tecnología en el grupo objetivo del proyecto; en este caso, jóvenes adultos que atraviesan una carrera universitaria.

Una alternativa que va ganando fuerza en la actualidad es el uso de teléfonos móviles como herramientas de aprendizaje. La característica personal del teléfono móvil, su presencia constante junto o alrededor del usuario, los tipos de comunicación que permite y, ante todo, el valor de identidad que implica para los jóvenes adultos en sus relaciones sociales, han convertido a este dispositivo en parte de su vida cotidiana y, el manejo de estos dispositivos, en una destreza natural e intuitiva. Según el reporte anual de estadísticas de Tecnologías de Información y Comunicación del Ecuador presentado por el INEC (Instituto Nacional de Estadísticas y Censos) correspondiente al año 2011, el 78,8 % de los hogares posee telefonía celular frente al 24,7% que posee computadora de escritorio y el 9,8% de hogares que posee computador portátil; lo que demuestra el enorme potencial de los teléfonos móviles como herramientas de aprendizaje colaborativo, al ser relativamente accesibles por la mayoría de los estudiantes.

A este potencial, se debe sumar la convergencia de tecnología entre diferentes dispositivos que combinan la funcionalidad de un ordenador personal con la versatilidad de un teléfono móvil. Estos dispositivos denominados "Smartphones", cuentan con su propio sistema operativo y permiten la instalación de aplicaciones para incrementar el procesamiento de datos y la conectividad. Sin embargo, estas aplicaciones dependen de la plataforma sobre la cual se desarrollarán; por esta razón, el presente capítulo, realiza una comparación de dos de los sistemas operativos móviles que actualmente lideran el mercado para determinar el que se ajusta mejor a las necesidades que se pretenden cubrir: iOS de Apple y Android de Google.

Objetivos del capítulo:

Los objetivos que se pretenden alcanzar son:

- Adquirir conocimientos básicos sobre los sistemas operativos móviles iOS de Apple y Android de Google, mediante revisión de literatura para una mejor comprensión de sus características.



UNIVERSIDAD DE CUENCA

- Comparar los dos sistemas operativos móviles, mediante la evaluación de criterios definidos para determinar el que mejor se adapta a las necesidades del proyecto.

2.2 CRITERIOS DE COMPARACIÓN

En esta sección se definen los criterios que serán evaluados para realizar la comparación entre los dos sistemas operativos móviles. Han sido agrupados en tres categorías: criterios técnicos, criterios de accesibilidad y criterios exclusivamente enmarcados en el contexto del proyecto.

2.2.1 Criterios técnicos

Los criterios de evaluación técnicos analizan las características tecnológicas y funcionalidades que proveen cada uno de los sistemas operativos móviles, así como también las capacidades que ofrecen las herramientas de desarrollo de cada plataforma. A continuación, se listan los factores evaluados con una breve definición de cada uno de ellos.

2.2.1.1 Arquitectura

Hace referencia a una vista general del sistema, el mismo que incluye sus componentes principales: la conducta de esos componentes, según se la perciba desde el resto del sistema y la forma en que los componentes interactúan y se coordinan para alcanzar la misión del sistema (Paul Clements, 1996). Con este criterio, se pretende evaluar si la plataforma permite crear software fácilmente analizable, modificable, corregible; así como el nivel de interacción con otros sistemas informáticos, su escalabilidad, etc.

2.2.1.2 Framework de desarrollo

Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

Los frameworks suelen incluir:

- Soporte de programas.
- Librerías.
- Lenguaje de scripting.
- Software para desarrollar y unir diferentes componentes de un proyecto de desarrollo.

Se evaluará el nivel de facilidad que brinda el framework de cada plataforma para el desarrollo de software y en qué medida ayudará a evitar los detalles de



UNIVERSIDAD DE CUENCA

bajo nivel, permitiendo concentrar más tiempo y esfuerzo en identificar los requerimientos de las aplicaciones móviles.

En el análisis del framework se pondrá especial énfasis en lo siguiente:

2.2.1.2.a SDK

El SDK (Software Development Kit - Kit de Desarrollo de Software) consiste en un conjunto de herramientas y programas de desarrollo, que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consola de videojuego, sistema operativo o similar. Se analizará, tanto el nivel de dificultad en la instalación, como la utilización de los recursos del SDK de cada una de las plataformas evaluadas.

2.2.1.2.b Lenguaje de Programación

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de los equipos, en este caso de los dispositivos móviles. Se evaluarán las fortalezas y debilidades de los lenguajes que se utilizan para desarrollar aplicaciones en Android y en iOS: Java y Objective-C, en ese orden.

2.2.1.2.c IDE

El IDE (Integrated Development Environment – Entorno de Desarrollo Integrado) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Se evaluarán las características principales de los IDE que se utilizan comúnmente para desarrollar aplicaciones tanto para Android como para iOS, Eclipse y Xcode, respectivamente.

2.2.1.2.d Librerías

Las librerías son un conjunto de subprogramas utilizados para desarrollar software. Contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de estos. El uso de las librerías permite que el código y los datos se compartan y puedan modificarse de forma modular. Se evaluará las funcionalidades que ofrecen las librerías incluidas en el framework de desarrollo, así como también la capacidad de incluir librerías desarrolladas por terceros.

2.2.1.2.e Tipos de aplicaciones

Se evaluarán los tipos de aplicaciones que se pueden desarrollar sobre cada una de las plataformas, ya sea para la Web o aplicaciones nativas de los diferentes sistemas operativos móviles.



UNIVERSIDAD DE CUENCA

2.2.1.3 Requisitos de software

Los requisitos de software son las características que debe cumplir el software instalado en una computadora para soportar y/o ejecutar una aplicación o un dispositivo específico.

Los requisitos de software pueden ser:

- Requisitos de sistema operativo.
- Requisitos de aplicaciones específicas instaladas.
- Requisitos de ciertas aplicaciones no instaladas en el mismo sistema.
- Requisitos de determinadas configuraciones en el sistema operativo o en ciertas aplicaciones.

En este caso se evaluarán los requisitos de software que se deben satisfacer para soportar el entorno de desarrollo de aplicaciones para iOS y Android. Se tendrá en cuenta también, el software necesario para realizar simulaciones y pruebas.

2.2.1.4 Requisitos de hardware

Los requisitos de hardware hacen referencia a las características que debe cumplir el hardware de un sistema para poder soportar y/o ejecutar una aplicación o conexión con otros dispositivos.

Los requisitos de hardware pueden ser:

- Requisitos de placa madre y de CPU.
- Requisitos de memoria.
- Requisitos de espacio disponible en disco duro.
- Requisitos de monitor y adaptador de video.
- Requisitos de unidades de CD, DVD, ratón, impresora, módem, tarjeta de red, puertos disponibles.

Se evaluarán los requisitos de hardware que se deben satisfacer para poder desarrollar aplicaciones tanto para Android como para iOS; así como también se tendrá en cuenta el hardware necesario para realizar pruebas.

2.2.2 Criterios de Accesibilidad

Los criterios de accesibilidad analizan la experiencia de usuario que brinda cada una de las plataformas móviles evaluadas, cuyo resultado es la generación de una percepción positiva o negativa de las mismas. Son criterios subjetivos que evalúan la usabilidad, familiaridad y predilección de los usuarios por una determinada plataforma; así como también, la facilidad de los



UNIVERSIDAD DE CUENCA

desarrolladores para acceder a la documentación y a las herramientas necesarias para crear sus aplicaciones y distribuirlas.

Los factores a ser evaluados bajo criterios de accesibilidad son los siguientes:

2.2.2.1 Términos y condiciones de licencia

Se evaluarán las condiciones de licencia que se deben cumplir para desarrollar y publicar aplicaciones tanto para Android como para iOS. Se tendrá en cuenta el proceso para adquirir licencias, en caso de ser necesario, y el costo de las mismas.

2.2.2.2 Documentación y recursos

Se estudiará la disponibilidad de documentación técnica, ejemplos de código, tutoriales, etc.

Bajo este criterio se debe poner énfasis en los costos de acceso a dichos recursos y a la calidad de los mismos.

2.2.2.3 Usabilidad

Usabilidad es el grado de eficacia, eficiencia y satisfacción con la que usuarios específicos pueden lograr objetivos específicos, en contextos de uso específicos (ISO 9241-11, 1998). Con este criterio, se pretende comparar que tan intuitivas y agradables resultan las aplicaciones desarrolladas para Android, en comparación con las aplicaciones desarrolladas para iOS.

2.2.3 Criterios exclusivamente enmarcados en el contexto del proyecto

Los criterios exclusivamente enmarcados en el contexto del proyecto permiten comparar los dos sistemas operativos móviles específicamente como plataformas de soporte para un proyecto de enseñanza aprendizaje. Los factores que se evaluarán bajo estos criterios son los siguientes:

2.2.3.1 Disponibilidad de equipos

Es necesario tomar en cuenta la predilección de una u otra plataforma dentro del grupo objetivo del proyecto; es decir, con este criterio se pretende anticipar qué sistema operativo móvil y qué tipo de dispositivos son los preferidos por los jóvenes estudiantes.

2.2.3.2 Antecedentes

Con este criterio se pretende determinar si existen experiencias previas en otras universidades de implementación de proyectos similares, para observar qué plataforma decidieron usar como base para el desarrollo del proyecto y su justificación para hacerlo.



UNIVERSIDAD DE CUENCA

Una vez definidos los criterios de evaluación, a continuación se explican las características de los sistemas operativos evaluados.

2.3 SISTEMA OPERATIVO ANDROID

Google junto con la Open Handset Alliance¹ realizan el lanzamiento del sistema operativo móvil Android a finales del año 2007 (Di Marzio, 2008). Android es una pila de software para dispositivos móviles que incluye un sistema operativo, middleware² y algunas aplicaciones clave como una agenda y un navegador Web (Android Developers, 2012a). Android, como sistema, es un sistema operativo basado en Java que corre sobre el kernel 2.6 de Linux (Di Marzio, 2008). Permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java.

2.3.1 Arquitectura

Como se puede observar en la **Figura 2.3-A**, la arquitectura interna de la plataforma Android está básicamente formada por capas. Cada capa utiliza los servicios provistos por las capas que se encuentran por debajo de ella (Burnette, 2009).

Partiendo desde la capa inferior, las capas que conforman la arquitectura del sistema Android son:

- Kernel de Linux
- Librerías
- Tiempo de Ejecución de Android
- Framework de Aplicaciones
- Aplicaciones y Widgets³

¹ La Open Handset Alliance (OHA) es una alianza comercial de 78 compañías para desarrollar estándares abiertos para dispositivos móviles. Algunos miembros son Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia y Wind River Systems

² Middleware es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos

³ En informática, un widget es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.



UNIVERSIDAD DE CUENCA

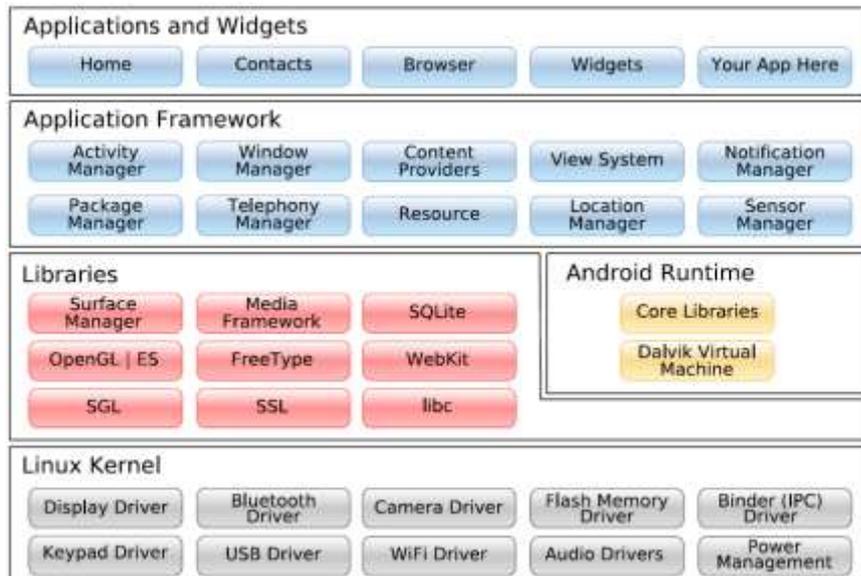


Figura 2.3-7 Arquitectura del Sistema Operativo móvil Android

2.3.1.1 Kernel de Linux

Linux provee una capa de abstracción de hardware para Android, lo cual permite una gran portabilidad dentro de un gran número de plataformas de hardware para este sistema operativo. Android utiliza Linux para realizar la gestión de memoria, la gestión de procesos, networking⁴, y otros servicios del sistema operativo (Burnette, 2009).

2.3.1.2 Librerías nativas

La capa que se sitúa por encima del kernel contiene las librerías nativas del sistema Android, las mismas que están escritas en su totalidad en lenguaje C o C++, compiladas para la arquitectura de hardware en particular utilizada por el móvil, y preinstaladas por el proveedor del mismo.

Algunas de las librerías nativas más importantes de Android incluyen lo siguiente (Burnette, 2009):

- **Surface Manager:** Gestión del acceso a la pantalla, creación de efectos y transiciones.
- **Media códecs:** Reproducción de imágenes, audio y video en diferentes formatos.
- **Base de datos SQL:** Android incluye el ligero motor de base de datos SQLite para persistencia de datos de las aplicaciones.

⁴Networking es un término utilizado para referirse a las redes de telecomunicaciones en general y a las conexiones entre ellas.



UNIVERSIDAD DE CUENCA

- **Motor de navegador Web:** Android utiliza la librería Webkit para una rápida visualización de contenido HTML.
- **Gráficos 2D y 3D:** Con Android se pueden combinar elementos en dos o tres dimensiones en una misma interfaz de usuario.

Estas librerías no son aplicaciones que corren por sí mismas, existen únicamente para ser llamadas por aplicaciones de un nivel más alto. Desde la versión 1.5 de Android es posible desarrollar librerías nativas usando el NDK (Native Development Toolkit).

2.3.1.3 Tiempo de Ejecución de Android

Esta capa también se encuentra situada sobre el kernel e incluye la máquina virtual Dalvik y las librerías del núcleo de Java. La máquina virtual Dalvik es la implementación de Java creada por Google, optimizada para dispositivos móviles. Todo el código escrito para Android estará en Java y correrá sobre esta máquina virtual (Burnette, 2009).

Dalvik tiene dos importantes diferencias con respecto a la versión tradicional de Java:

- La máquina virtual Dalvik ejecuta archivos con extensión .dex que son convertidos en tiempo de compilación a partir de los archivos estándar .class y .jar. Estos archivos .dex son más compactos y eficientes que los archivos .class, lo cual es importante debido a las limitaciones de memoria de los dispositivos móviles.
- Las librerías del núcleo de Java que vienen con Android son diferentes a las librerías tanto de la Edición Estándar de Java (Java SE) como de la Edición Móvil de Java (Java ME).

2.3.1.4 Framework de Aplicaciones

Esta capa se sitúa por encima de las librerías nativas y del tiempo de ejecución. Provee los bloques de construcción de alto nivel que se utilizan para desarrollar las aplicaciones. El framework viene preinstalado con Android; sin embargo, es posible extenderlo de ser necesario.

Las partes más importantes del Framework de Aplicaciones son las siguientes (Burnette, 2009):

- **Gestor de Actividades:** Controla el ciclo de vida de las aplicaciones.
- **Proveedores de Contenido:** Estos objetos encapsulan datos que requieren ser compartidos entre las aplicaciones, como por ejemplo: la información de los contactos.



UNIVERSIDAD DE CUENCA

- **Gestor de Recursos:** Recursos son textos, mapas de bits, o cualquier pedazo de información que no sea código que la aplicación necesita.
- **Gestor de Ubicación:** Controla funciones de localización para aplicaciones que requieran conocer la ubicación del dispositivo, ya sea mediante el GPS o información de la celda celular donde se encuentra.
- **Gestor de Notificaciones:** Controla eventos como la llegada de mensajes, alertas, etc.

2.3.1.5 Aplicaciones y Widgets

La capa superior en la arquitectura del sistema Android es la capa de Aplicaciones y Widgets. Esta capa es la única que los usuarios finales pueden ver. Las aplicaciones son los programas que ocupan toda la pantalla y que interactúan con el usuario. Los widgets sólo operan en un pequeño rectángulo de la pantalla principal.

Algunas de las aplicaciones estándar que se incluyen en los teléfonos con el sistema operativo Android son (Reto Meier, 2009):

- Un cliente de correo electrónico compatible pero no limitado a funcionar con Gmail.
- Una aplicación de gestión de SMS⁵.
- Usa suite completa de manejo de información personal que incluye un calendario y una lista de contactos, ambos sincronizados con los servicios en línea de Google.
- Una aplicación móvil con todas las funciones de Google Maps Street View, incluyendo buscador de negocios, indicaciones para llegar, vista de satélite, y las condiciones del tráfico.
- Un navegador Web basado en WebKit⁶.
- Un cliente de mensajería instantánea.
- Un reproductor de música y un visor de imágenes.
- El cliente de Android Market para descargar aplicaciones de terceros.

⁵ Short Message Service (SMS). Servicio de mensajería para teléfonos celulares. Permite enviar a un celular un mensaje de hasta 160 caracteres.

⁶ WebKit es un framework para aplicaciones que funciona como base para el navegador Web Safari, Google Chrome, Epiphany, Midori entre otros. Está basado originalmente en el motor de renderizado KHTML del navegador Web del proyecto KDE, Konqueror.



UNIVERSIDAD DE CUENCA

2.3.2 Framework de desarrollo Android

Una característica importante para los desarrolladores de aplicaciones Android es que tienen acceso completo a los mismos APIs del framework, usados por las aplicaciones nativas. De esta manera, el framework de desarrollo de Android ofrece un acceso ilimitado a todas las características del dispositivo, como el GPS, el marcador telefónico, etc. Esta capacidad de acceder a todo el hardware del dispositivo permite a los desarrolladores crear aplicaciones con niveles de interactividad muy altos que igualen o mejoren las prestaciones de las aplicaciones nativas del sistema.

2.3.2.1 Android SDK

El SDK de Android proporciona un conjunto de herramientas necesarias para desarrollar aplicaciones para el sistema operativo Android (Eero Maasalmi y Panu Pitkänen, 2011). Contiene todas las librerías de código Java necesarias para crear aplicaciones que se ejecutan en concreto en la plataforma Android. El SDK y las herramientas están disponibles gratuitamente. Los desarrolladores pueden descargarlo desde el sitio Web de Android después de haber aceptado los términos del Contrato de Licencia del Kit de Desarrollo de Software (Shane Conder y Lauren Darcey, 2011).

En la descarga del SDK de Android se incluye (Reto Meier, 2009):

- **Las API de Android:** El núcleo del SDK de Android son las API que proporcionan al desarrollador el acceso a la pila de Android. Son las mismas API utilizadas por Google para crear aplicaciones nativas para Android.
- **Herramientas de desarrollo:** el SDK incluye varias herramientas que permiten al desarrollador, compilar y depurar sus aplicaciones.
- **El emulador Android:** El emulador de Android es un emulador de dispositivo Android totalmente interactivo. Usando este emulador, se puede percibir cómo se ven y se comportan las aplicaciones en un verdadero dispositivo Android.
- **Documentación completa:** El SDK incluye una amplia información de referencia a nivel de código que detalla lo que está incluido en cada paquete y en cada clase, así como también información de cómo usarlo. Además de la documentación de código, la documentación del SDK da una explicación detallada de los fundamentos detrás del desarrollo de aplicaciones Android.
- **Código de ejemplo:** El SDK de Android incluye una selección de aplicaciones de ejemplo que demuestran algunas de las posibilidades



UNIVERSIDAD DE CUENCA

para desarrollar en Android, así como programas sencillos que demuestran las funciones individuales de las API del SDK.

2.3.2.2 Lenguaje de programación

Los programas para el sistema operativo Android están escritos en su mayoría en el lenguaje de programación Java y utilizan los API de las librerías de Java Standard Edition (Burnette, 2009). Las técnicas, sintaxis y gramática que utilizan los desarrolladores para crear aplicaciones con Java se traducen y son aplicables directamente para desarrollar aplicaciones para Android (Reto Meier, 2009). Android usa un compilador estándar de Java para compilar el código fuente en bytecodes⁷ regulares y luego traduce esos bytecodes en instrucciones Dalvik, lo cual permite que todo el lenguaje Java sea compatible, no sólo un subconjunto del mismo (Burnette, 2009).

Usar Java como lenguaje de desarrollo para aplicaciones Android es una ventaja. Java es un lenguaje mundialmente extendido. Es posible contar con una gran cantidad de documentación fácilmente accesible, lo cual reduce la curva de aprendizaje para los desarrolladores que desean crear aplicaciones para Android.

2.3.2.3 IDE

Las aplicaciones Android, al ser escritas en código Java, permiten utilizar cualquier IDE que soporte el desarrollo de aplicaciones en dicho lenguaje de programación. De hecho, es tal la flexibilidad, que es igualmente posible crear aplicaciones Android sin utilizar IDE alguno. Sin embargo, Google y la Open Handset Alliance recomiendan desarrollar las aplicaciones Android con un IDE en especial: Eclipse (DiMarzio, 2008). Al instalar el plugin Android Development Toolkit (ADT) en Eclipse se agregan algunas características al IDE, entre las cuales se destacan (Reto Meier, 2009):

- **Asistente de Proyectos Android:** simplifica la creación de nuevos proyectos e incluye una plantilla base para aplicaciones Android.
- **Editores de diseño:** facilita el crear, editar y validar los recursos de XML
- **Construcción automatizada de proyectos de Android:** que comprende la conversión del código fuente en archivos ejecutables Android (.dex), el empaquetado de los archivos ejecutables en archivos de paquetes (.apk), y la instalación de los paquetes en máquinas virtuales Dalvik.

⁷El bytecode es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina.



UNIVERSIDAD DE CUENCA

- **Emulador de Android:** permite realizar pruebas de las aplicaciones e incluye el control de la apariencia del emulador, la configuración de conexión de red, y la capacidad de simular las llamadas entrantes y mensajes SMS.
- **Servicio de Monitoreo de depuración Dalvik (DDMS):** que incluye el reenvío de puertos, visualización de hilos, detalles del proceso, y captura de pantalla.
- **Acceso al sistema de archivos del emulador o del dispositivo:** lo que permite navegar por el árbol de carpetas y transferir archivos.
- **Depuración en tiempo de ejecución:** para establecer puntos de interrupción y ver las pilas de llamadas.
- **Visualización de todos los registros y salidas:** provenientes tanto de Android como de la máquina virtual Dalvik.

2.3.2.4 Librerías

Android admite un subconjunto relativamente grande de las librerías de la Edición Estándar de Java (Java SE). Obviamente este subconjunto de librerías se encuentra adaptado a un ambiente de desarrollo para aplicaciones móviles. Algunas librerías fueron omitidas, ya que simplemente no son requeridas (por ejemplo librerías de impresión), y otras fueron sustituidas debido a una mejor API disponible, específicamente para Android (por ejemplo, las interfaces de usuario) (Burnette, 2009).

Algunos de los paquetes de librerías más importantes de la versión estándar de Java soportados por Android son:

- **java.awt.font:** Algunas constantes únicas de y tipos de letra.
- **java.io:** Entrada/Salida de flujos de datos en pantalla y archivos.
- **java.lang:** Soporte para el lenguaje y sus excepciones.
- **java.math:** Manejo de números grandes, redondeo y precisión.
- **java.net:** Entrada/Salida de red, URLs, sockets.
- **java.security:** Autorización, certificados, llaves públicas.
- **java.sql:** Interfaces para bases de datos.
- **java.text:** Formato, lenguaje natural, cotejo.
- **java.util:** Listas, mapas, conjuntos, arreglos, colecciones.
- **javax.crypto:** Cifrado, llaves públicas.
- **javax.microedition.khronos:** Gráficos Open GL⁸.

⁸ OpenGL (Open Graphics Library) es una especificación estándar que define una API multilinguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.



UNIVERSIDAD DE CUENCA

A parte de las librerías estándar, en Android se cuenta con algunas librerías de terceros que ofrecen funcionalidades adicionales, entre las más importantes tenemos:

- **org.apache.http**: autenticación HTTP, cookies, métodos, y protocolo.
- **org.json**: soporte para el formato ligero de intercambio de datos JSON (JavaScript Object Notation).
- **org.xml.sax**: manejo de XML.
- **org.xmlpull.v1**: manejo de XML.

Android ofrece una serie de APIs para el desarrollo de aplicaciones. Todos los dispositivos Android ofrecerán soporte al menos para estas API (Reto Meier, 2009):

- **android.util**: Es el paquete utilitario básico, contiene clases de bajo nivel como contenedores especializados, formateadores de cadenas de texto, y utilidades de análisis XML.
- **android.os**: El paquete de sistema operativo provee servicios básicos del sistema operativo como el paso de mensajes, comunicación interprocesos, funciones de reloj y depuración.
- **android.graphics**: Esta API provee las clases de bajo nivel que permiten manipular y crear gráficos y animaciones.
- **android.text**: Herramientas para procesamiento y comparación de texto.
- **android.database**: Clases de bajo nivel para el manejo de bases de datos.
- **android.content**: API utilizada para gestionar el acceso a los datos y la publicación de los mismos, provisionando servicios para manejar recursos, proveedores de contenidos y paquetes.
- **android.view**: Las vistas son las clases principales de la interfaz de usuario. Todas las interfaces de usuario son construidas usando una serie de vistas que proveen de componentes de interacción con el usuario.
- **com.google.android.maps**: Una API de alto nivel que provee acceso a controles de mapas nativos que se puede usar en las aplicaciones.
- **android.provider**: Facilita el acceso del desarrollador a ciertos proveedores de contenido estándar incluidos en todas las distribuciones Android, como por ejemplo: la base de datos de contactos.
- **android.telephony**: Permite hacer, recibir y monitorear las llamadas telefónicas, controlar el estado del teléfono y los mensajes SMS.
- **android.webkit**: Permite trabajar con contenido basado en la Web.

2.3.2.5 Tipos de aplicaciones

En Android se puede clasificar las aplicaciones en tres categorías (Reto Meier, 2009):



UNIVERSIDAD DE CUENCA

- **Actividad de primer plano:** Una aplicación que es útil solamente cuando se encuentra ejecutándose en primer plano y se encuentra suspendida cuando no es visible. Ejemplo: Juegos.
- **Servicio de segundo plano:** Una aplicación con una interacción limitada que permanece oculta, excepto cuando se realiza tareas de configuración. Ejemplo: respuesta automática de SMS.
- **Actividad intermitente:** Espera cierta interactividad, pero realiza la mayor parte de su trabajo en segundo plano. Ejemplo: reproductor de medios.

Las aplicaciones más complejas son difíciles de enmarcar dentro de una sola categoría. Generalmente la mayoría de estas aplicaciones tienen características de las tres categorías. Se debe tener en consideración al momento de crear una aplicación el nivel de interactividad que se desea con el usuario y cómo debe la aplicación realizar su trabajo para diseñarla adecuadamente.

2.3.3 Requisitos de software

Para crear aplicaciones Android, es necesario configurar el entorno de programación para el desarrollo de aplicaciones en Java. El software está disponible para su descarga sin costo.

Los sistemas operativos soportados para el desarrollo de aplicaciones Android son (Android Developers, 2012b):

- **Windows XP** (32-bits), **Windows Vista** (32-64 bits), **Windows 7** (32-64 bits)
- **Mac OS X 10.5.8** o superiores (solamente 32 bits)
- **Linux** (Ubuntu versión 8.04 o superior)

De igual manera, para desarrollar aplicaciones Android es necesario tener instalado el siguiente software en el computador (Shane Conder y Lauren Darcey, 2011):

- El **Java Development Kit (JDK)** versión 5 o 6, disponible para su descarga en: <http://java.sun.com/javase/downloads/index.jsp>.
- Un **IDE compatible con Java** (preferiblemente Eclipse 3.6.2 o superiores), junto con su plugin JDT, disponible para descargar en: <http://www.eclipse.org/downloads/>.
- El **SDK de Android**, herramientas y documentación, disponible para su descarga en: <http://developer.android.com/sdk/index.html>.



UNIVERSIDAD DE CUENCA

- El **plugin para Eclipse Android Development Tools (ADT)** (opcional), disponible para descargar a través del mecanismo de actualización de software Eclipse.

Para realizar pruebas es necesario tener instalado el **Emulador Android**. El desarrollador puede crear Dispositivos Virtuales Android (AVD) que corran sobre éste emulador y realizar pruebas como si se tratara de dispositivos Android reales.

2.3.4 Requisitos de hardware

El SDK de Android requiere almacenamiento en disco para todos los componentes que se elija instalar. La **Tabla 2.3-A** provee una idea de los requisitos necesarios de espacio en disco para el desarrollo de aplicaciones Android (Android Developers, 2012b):

Tipo de componente	Tamaño aproximado	Comentario
Herramientas SDK	35 MB	Requerido
Herramientas SDK de la plataforma	6 MB	Requerido
Plataforma Android (cada una)	150 MB	Se requiere al menos una plataforma
SDK Add-on (cada uno)	100 MB	Opcional
Driver USB	10 MB	Opcional. Sólo para Windows
Ejemplos (por plataforma)	10 MB	Opcional
Documentación fuera de línea	250 MB	Opcional

Tabla 2.3-A Requisitos de espacio en disco para componentes del SDK Android



UNIVERSIDAD DE CUENCA

2.3.5 Términos y condiciones de licencia

El framework de desarrollo de Android se distribuye bajo la Licencia de Software Apache (ASL/Apache2), la cual permite la distribución tanto libre como propietaria de derivaciones del código fuente. Los desarrolladores de aplicaciones para Android tienen la posibilidad de distribuir sus aplicaciones bajo el esquema de licenciamiento que prefieran, ya sea freeware de código abierto o aplicaciones con licencia tradicional con fines de lucro (Shane Conder y Lauren Darcey, 2011).

Los desarrolladores pueden distribuir sus aplicaciones a través de Google Play, antes conocido como Android Market. Se trata de un sitio de acceso público, donde se pueden encontrar aplicaciones tanto gratuitas como de paga. Para la publicación de una aplicación en Google Play es necesario registrarse y mantener una cuenta de desarrollador válida. El registro tiene un costo de USD \$25,00.

Para la distribución de aplicaciones pagadas, adicional al registro de usuario en Google Play se debe adquirir y mantener una cuenta de pago válida. Esta cuenta se puede configurar a través de un Procesador de pago autorizado (ejemplo Paypal), o se puede especificar la cuenta bancaria en donde se desea que se deposite el dinero. El desarrollador debe especificar el costo de la aplicación; de ese valor, el 70% será para el autor y el 30% restante para Google Play. Para efectos de reembolso, sólo los productos que puedan ofrecer al comprador una vista previa de la aplicación, pueden solicitar reembolso durante las 48 horas posteriores a la compra.

El desarrollador conserva los derechos sobre el software Android a desarrollar con el SDK, incluidos los derechos de propiedad intelectual. El desarrollador acepta que sus aplicaciones protegen los derechos de privacidad y legales de sus usuarios. Si la aplicación utiliza o tiene acceso a información personal y privada sobre el usuario (nombres de usuario, contraseñas, etc.), entonces la aplicación proporcionará una notificación adecuada de la privacidad indicando que se mantendrán los datos almacenados de forma segura (Shane Conder y Lauren Darcey, 2011).

2.3.6 Documentación y recursos

Con la descarga del SDK de Android se incluye una copia local de la documentación en la subcarpeta /docs. La documentación se encuentra en formato HTML y también es accesible en línea en la dirección: <http://developer.android.com>.

La documentación está dividida en siete secciones principales (Shane Conder y Lauren Darcey, 2011):

- La **sección de inicio** es el punto de inicio de la documentación de Android. En esta sección se encuentran los anuncios de desarrolladores y enlaces importantes a los últimos temas en el desarrollo para Android.



UNIVERSIDAD DE CUENCA

- La **sección del SDK** provee información acerca de las diferentes versiones disponibles del SDK de Android, así como información acerca del paquete de desarrollo nativo de Android (NDK)
- La **sección de guía para desarrolladores**, contiene una introducción a la plataforma Android y contiene las mejores prácticas para el diseño, desarrollo y distribución de aplicaciones Android.
- La **sección de referencia** contiene una lista de las APIs de Android con cobertura detallada de las clases específicas e interfaces.
- La **sección de recursos** provee acceso a los artículos técnicos de Android y tutoriales además de enlaces a la comunidad online de Android y a las aplicaciones de ejemplo proporcionados junto con el SDK.
- La **sección de videos** provee acceso a los videos online disponibles para el desarrollo de aplicaciones Android, incluyendo videos de la plataforma, consejos de los desarrolladores, sesiones de desarrollo en Android de la conferencia anual Google I/O, etc.
- La **sección de blog** provee acceso a los blogs online publicados por el equipo de desarrolladores Android. También se encuentran anuncios de los lanzamientos del SDK, consejos de desarrollo y noticias de eventos Android.

2.3.7 Usabilidad

Los desarrolladores Android tienen una tarea adicional al momento de diseñar sus aplicaciones para que brinden una gran experiencia de usuario: adaptar el contenido de las aplicaciones para que se ajusten a diferentes dispositivos. Esto se debe a la gran cantidad de equipos de diferentes marcas con características distintas que existen en el mercado y que utilizan el sistema operativo Android. Los desarrolladores deben tener en cuenta las distintas formas, dimensiones, resoluciones y tecnología de las pantallas de los dispositivos, sobre los cuales debe correr la aplicación manteniendo una buena relación de aspecto y sobretodo siendo atractivas para el usuario.

Por ventaja, dentro de las herramientas de desarrollo del SDK Android, el Emulador permite inicializar varios Dispositivos Virtuales Android con diferentes características para realizar pruebas. Un gran número de interfaces de usuario alternativas están disponibles en el Emulador Android para representar diferentes configuraciones, cada una con diferentes tamaños de pantalla, resoluciones, orientaciones y características de hardware para simular una variedad de dispositivos móviles (Reto Meier, 2009). De esta forma, el desarrollador puede realizar pruebas de usabilidad para medir el grado de



UNIVERSIDAD DE CUENCA

satisfacción del usuario con sus aplicaciones en diferentes ambientes, bajo diferentes características de hardware con el menor costo posible.

2.4 SISTEMA OPERATIVO iOS

El sistema operativo iOS, antes conocido como iPhone OS, es un sistema basado en UNIX construido para los dispositivos móviles de Apple: el iPhone, el iPod touch y el iPad (Eero Maasalmi y Panu Pitkänen, 2011). Fue presentado por Steve Jobs, el 9 de enero del 2007 y lanzado al mercado de los EEUU el 29 de junio del mismo año con la versión 1.0. Desde entonces, se dio el impulso para nuevas versiones y la revolución de las aplicaciones móviles comenzó (Julio Fernández, 2011). Basado en Mac OS X, iOS es considerado uno de los sistemas operativos más confiables en términos de hardware y ofrece a los desarrolladores de software un conjunto de herramientas contenidas en el SDK, además de interfaces necesarias para desarrollar, instalar, ejecutar y probar las aplicaciones nativas⁹.

2.4.1 Arquitectura

La arquitectura de iOS es similar a la que podemos encontrar en el sistema MAC OS X. En su nivel más alto, actúa como un intermediario entre el hardware y las aplicaciones que aparecen en la pantalla de los equipos. Esto brinda un nivel de abstracción que permite que las aplicaciones raramente se comuniquen directamente con el hardware del dispositivo. En realidad, las aplicaciones se comunican con una serie definida de interfaces del sistema, lo que las protege de los cambios de hardware y les permite ejecutarse sobre diferentes dispositivos con características de hardware distintas (iOS Developer, 2012).

Como podemos observar en la **Figura**, la arquitectura iOS está formada por capas. Las capas más bajas del sistema son los servicios fundamentales y las tecnologías sobre las cuales todas las aplicaciones están basadas. Las capas de más alto nivel contienen los servicios y las tecnologías más sofisticadas.

Partiendo desde la capa inferior, las capas que conforman la arquitectura del sistema iOS son:

- Núcleo del Sistema Operativo
- Servicios Principales
- Capa de Medios
- Cocoa Touch

⁹ *Aplicaciones nativas son aquellas que se instalan físicamente en un dispositivo y por ello siempre se encuentran disponibles para el usuario, ya que residen junto a otras aplicaciones del sistema.*



UNIVERSIDAD DE CUENCA

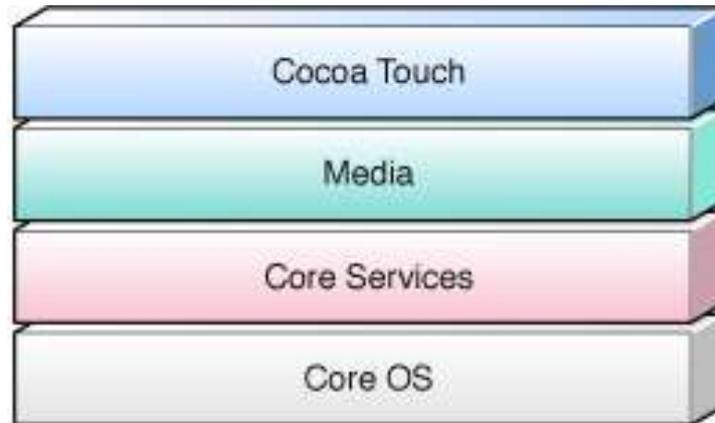


Figura 2.4-A Arquitectura del sistema iOS

2.4.1.1 Núcleo del Sistema Operativo (Core OS)

Esta capa es la base del sistema operativo iOS. Se encarga de realizar la gestión de memoria, el sistema de ficheros, conexión a la red y procesos que interactúan con el hardware. Las interfaces de esta capa están en su mayoría basadas en el lenguaje C. Las características de esta capa son raramente accedidas directamente por los desarrolladores, pero son comúnmente utilizadas por los frameworks de capas superiores.

2.4.1.2 Servicios Principales (Core Services)

Esta capa permite el acceso a los servicios fundamentales del sistema, tales como la agenda, el acceso a la base de datos, preferencias, conexión a servidores, procesamiento de URLs, etc. (iOS Developer, 2012). Tal como sucede con la capa de núcleo del sistema, la mayoría de las interfaces de esta capa también están basadas en el lenguaje de programación C.

2.4.1.3 Capa de Medios (Media Layer)

La capa de medios permite la ejecución de tareas multimedia, como la visualización de gráficos, reproducción de audio, video, y tecnologías orientadas a la creación de mejoras multimedia. La principal finalidad de esta capa es facilitar la creación de aplicaciones que se vean y suenen bien; es decir, crear una gran experiencia multimedia en un dispositivo móvil (iOS Developer, 2012).

2.4.1.4 Cocoa Touch

La capa Cocoa Touch contiene los frameworks clave para la creación de aplicaciones iOS. Esta capa define la infraestructura de las aplicaciones y brinda soporte para tecnologías clave como la multitarea, entrada de datos táctil, notificaciones y otros servicios de alto nivel del sistema. En definitiva,



UNIVERSIDAD DE CUENCA

esta capa se encarga de gestionar la interacción visual con el usuario (iOS Developer, 2012).

Entre las características clave que brinda esta capa se encuentran:

- **Storyboards:** Introducidas en iOS 5, las Storyboards han reemplazado a los archivos .nib para el diseño de las interfaces de usuario. La ventaja que tienen sobre los archivos .nib es que permiten diseñar todas las vistas del sistema: cómo trabajan juntas y las transiciones entre ellas en un solo lugar.
- **Soporte para documentos:** Introducido en iOS 5, el framework UIKit introdujo la clase UI Document para la gestión de los datos asociados con los documentos del usuario. Esto facilita la creación de aplicaciones basadas en documentos, sobre todo aplicaciones que leen o almacenan documentos en iCloud¹⁰.
- **Multitarea:** Las aplicaciones creadas con iOS SDK 4.0 o posterior (y que se ejecutan en iOS 4.0 y posteriores) no finalizan cuando el usuario presiona el botón de inicio, cambian a un contexto de ejecución en segundo plano. El soporte multitarea definido por UIKit ayuda a que la transición de la aplicación, desde y hacia el segundo plano sea mucho más sencilla.
- **Impresión:** Introducido en iOS 4.2, el soporte de impresión UIKit permite que las aplicaciones puedan enviar contenido que desean imprimir de forma inalámbrica a impresoras cercanas.
- **Protección de datos:** Introducido en iOS 4.0, la protección de datos permite a las aplicaciones que trabajan con datos confidenciales del usuario tomar ventaja de la encriptación incorporada disponible en algunos dispositivos. En iOS 5 y posteriores, la protección de datos incluye los niveles de seguridad adicionales para los archivos protegidos.
- **Servicio de notificaciones:** Introducido en iOS 4.0, el servicio de notificaciones permite a las aplicaciones generar notificaciones a nivel local, en lugar de confiar en un servidor externo. Las aplicaciones que se ejecutan en segundo plano pueden usar las notificaciones locales como una forma de llamar la atención de un usuario cuando ocurren eventos importantes.

¹⁰iCloud es un sistema de almacenamiento nube o cloud computing de Apple Inc. El servicio permite a los usuarios almacenar datos en servidores remotos para descargar en múltiples dispositivos como iPhones, iPods, iPads y las computadoras personales que funcionen con Mac OS X (Lion o más reciente) o Windows de Microsoft (Windows Vista o más reciente).



UNIVERSIDAD DE CUENCA

- **Reconocimientos de gestos:** Introducido en iOS 3.2, los reconocedores de gestos son los objetos que se conectan con las vistas y se utilizan para detectar los tipos comunes de gestos que el usuario realiza en la pantalla táctil. Permiten indicar las acciones a realizar cuando se produce un determinado gesto.
- **Soporte para intercambio de archivos:** Introducido en iOS 3.2, el soporte para intercambio de archivos permite a las aplicaciones hacer que los archivos de datos de usuario que manejan sean accesibles a través de iTunes 9.1 y posteriores. Una aplicación con soporte declarado al intercambio de archivos hace que los contenidos de su directorio /documents estén disponibles para el usuario. El usuario puede mover archivos dentro y fuera de este directorio, según sea necesario desde iTunes.
- **Servicios Peer-to-Peer:** Introducido en iOS 3.0, el framework Game Kit permite conectividad peer-to-peer a través de Bluetooth. Es posible iniciar sesiones de comunicación con dispositivos cercanos y poner en práctica muchas de las características que se encuentran en las partidas multijugador. Aunque se utiliza principalmente en los juegos, también se puede utilizar esta función en otros tipos de aplicaciones.
- **Controladores de vista estándar del sistema:** Muchos de los frameworks de la capa Cocoa Touch contienen los controladores de vista para la presentación de interfaces estándar del sistema. Estos controladores de vista son utilizados por los desarrolladores en sus aplicaciones para presentar una experiencia de usuario consistente.
- **Soporte para pantalla externa:** Introducido en iOS 3.2, el soporte para pantalla externa permite a algunos dispositivos basados en iOS, conectarse a una pantalla externa a través de un juego de cables compatible.

2.4.2 Framework de desarrollo iOS

Apple ofrece a los desarrolladores, la mayoría de sus interfaces de sistema en paquetes especiales llamados frameworks. Estos frameworks son directorios que contienen una biblioteca dinámica compartida y recursos como: archivos de encabezado, imágenes, etc. Para que el desarrollador pueda utilizar las características de los frameworks, únicamente necesita vincularlos a su proyecto y obtendrá acceso completo a sus recursos.



UNIVERSIDAD DE CUENCA

2.4.2.1 iOS SDK

El SDK de iOS contiene todas las herramientas necesarias para diseñar, crear, depurar y optimizar el software para iOS (Eero Maasalmi y Panu Pitkänen, 2011). Está disponible gratuitamente para los desarrolladores. Sin embargo, para acceder a la descarga del SDK es necesario registrarse como “Desarrollador Apple” (ver sección 4.5 Términos y condiciones de licencia).

Algunos de los componentes claves del iOS SDK incluyen (iOS Developer, 2012):

- **Herramientas XCode:** son las herramientas que soportan el desarrollo de aplicaciones iOS. Las principales herramientas XCode son:
 - **IDE XCode:** Es el entorno de desarrollo integrado (IDE) que gestiona los proyectos de las aplicaciones, permitiendo editar, compilar, correr y depurar el código. XCode es la aplicación principal que se utiliza para el desarrollo.
 - **Instrumentación:** Herramientas de análisis y depuración en tiempo de ejecución. Permite obtener información acerca del comportamiento de la aplicación en tiempo de ejecución, permitiendo identificar problemas potenciales.
- **Simulador iOS:** Es una aplicación Mac OS X que simula la pila de tecnología iOS. Permite al desarrollador probar sus aplicaciones localmente en su computador Mac como si corriera sobre un dispositivo iOS real.
- **Biblioteca del Desarrollador iOS:** Es la documentación conceptual y de referencia que enseña todo acerca de las tecnologías iOS y el proceso de desarrollo de aplicaciones.

2.4.2.2 Lenguaje de programación

Las aplicaciones escritas para iOS se desarrollan en el lenguaje Objective-C. Se trata de un lenguaje de programación orientado a objetos descendiente de C, por lo que existe mucha similitud en cuanto a sintaxis y características. Objective-C es fuertemente influenciado por Small Talk, lo que lo hace un lenguaje muy limpio, rápido y fácil de aprender. Además, Objective-C se caracteriza por su dinamismo y las múltiples decisiones que permite tomar al usuario en tiempo de compilación y ejecución.

Quizás uno de los patrones de diseño más importantes en todas las aplicaciones de iOS es la separación de la interfaz y la implementación. En el lenguaje Objective-C, una clase se define en dos archivos. La interfaz, en un archivo .h que contiene variables de instancia, firmas de métodos y



UNIVERSIDAD DE CUENCA

propiedades; y la implementación, en un archivo .m que contiene los cuerpos de los métodos (Mark Goadrich y Michael Rogers, 2011).

Desconocer completamente el lenguaje Objective-C no es una gran desventaja para el desarrollo de aplicaciones iOS. Por fortuna, se cuenta con una gran cantidad de documentación en línea como tutoriales y ejemplos de código, por lo que simplemente se requiere de conocimientos básicos acerca de la programación orientada a objetos para aprender este lenguaje.

2.4.2.3 IDE

Apple proporciona el entorno de desarrollo X-Code, que permite la creación de aplicaciones tanto para MacOSX como para todos sus dispositivos iOS (iPhone, iPodTouch y iPad). Se trata de un IDE moderno, que los desarrolladores encontrarán muy familiar si han trabajado con Eclipse, Net Beans o Visual Studio.

Entre las principales características de este IDE se destacan las siguientes:

- Incorpora **herramientas para el desarrollo**, entre ellas, el simulador de iPhone/iPad, donde se pueden probar todas las aplicaciones sin necesidad de disponer de un dispositivo de Apple.
- Soporta el lenguaje de programación **Objective-C**.
- Cuenta con un **editor visual**, el cual permite diseñar interfaces de usuario atractivas para las aplicaciones.
- Cuenta con **herramienta de análisis** que permiten optimizar y monitorizar la aplicación.
- Es un software gratuito.

Un proyecto en Xcode, es un repositorio de todos los archivos, recursos e información necesaria para construir uno o más productos de software. Contiene todos los elementos de software que se utilizarán para la construcción de las aplicaciones y mantiene correctamente la relación entre dichos elementos (Mac OS X Developer, 2011).

2.4.2.4 Librerías

Como plataforma, iOS ha demostrado popularidad entre los desarrolladores. Esto se debe a la gran cantidad de librerías disponibles que proporcionan funcionalidades muy útiles al momento de desarrollar las aplicaciones, brindando un gran apoyo y robustez al programador.



UNIVERSIDAD DE CUENCA

El sistema operativo iOS utiliza enlaces simbólicos para apuntar a la versión actual de la mayoría de las librerías, algo muy útil en el caso de las librerías dinámicas. Las versiones de las librerías pueden cambiar con la aparición de nuevas versiones de iOS. Si el software se encuentra ligado a una versión específica de una librería, ésta no siempre se encontrará disponible en el sistema del usuario.

Se debe tener en cuenta que algunas librerías especiales de las capas Core OS y Core Services, no son empaquetadas como frameworks. El sistema iOS incluye muchas librerías dinámicas en el directorio `/usr/lib` del sistema. Las librerías dinámicas compartidas se identifican por su extensión `.dylib`, mientras que los archivos de cabecera para las librerías se encuentran en el directorio `/usr/include` (iOS Developer Library, 2010).

También es posible incluir librerías de código abierto creadas por terceros en el desarrollo de las aplicaciones para iOS. Entre las más utilizadas tenemos:

- **MB Progress HUD:** Proporciona una pantalla de progreso translúcida idéntica a la aplicación oficial de Apple, con la diferencia que esta se encuentra documentada y fácil de entender. Además, ofrece opciones adicionales como indicador de progreso visual y mensaje de progreso completado.
- **ASIHttpRequest:** Simplifica la comunicación en la red y ofrece herramientas avanzadas tales como: la carga de archivos, manejo de redirección, envío de formularios y almacenamiento en caché. Útil para el manejo de HTTP en las aplicaciones.
- **JSON Framework:** En las librerías nativas de iOS no existe soporte para JSON, pero JSON Framework ofrece todo lo necesario, incluyendo un programa de análisis para convertir cadenas JSON en objetos, y un generador para crear cadenas JSON a partir de objetos.
- **Flurry:** Automatiza estadísticas del uso de la aplicación, tales como: el número de usuarios que se tiene, su actividad con la aplicación y además permite especificar el seguimiento de los eventos y los errores de registro.
- **GData client:** Librería oficial de Google que permite acceder a varios de sus servicios como: contactos, calendario, análisis, traductor, YouTube, etc. GData se encuentra completamente documentado con gran cantidad de aplicaciones de ejemplo.

2.4.2.5 Tipos de aplicaciones

El SDK de iOS permite la creación de dos tipos de aplicaciones: aplicaciones nativas y aplicaciones Web. Las aplicaciones nativas son aquellas que



UNIVERSIDAD DE CUENCA

aparecen en la pantalla principal del dispositivo y se pueden ejecutar sin conexión de red localmente en el dispositivo. Las aplicaciones Web son aquellas que están ubicadas sobre un servidor Web, se transmiten a través de la red, utilizan HTML, CSS y JavaScript y se ejecutan dentro de un navegador. (iOS Developer, 2012).

2.4.3 Requisitos de software

Para desarrollar aplicaciones iOS es indispensable contar con un equipo Mac corriendo el sistema operativo Mac OS X 10.6 Snow Leopard o superiores (Mark Goadrich y Michael Rogers, 2011). De igual manera, se necesita contar con el SDK de iOS instalado y la suite de herramientas de desarrollo de Apple XCode. Esta suite proporciona el soporte a la gestión de proyectos, edición de código, la creación de ejecutables, depuración a nivel de código fuente, gestión del repositorio de código fuente, la optimización del rendimiento, etc. En el centro de esta suite se encuentra la aplicación Xcode, que proporciona el entorno de desarrollo del código fuente (iOS Developer, 2012).

2.4.4 Requisitos de hardware

Dado que las aplicaciones tienden a ser relativamente pequeñas en tamaño, y se ejecutan en procesadores mucho más lentos, las computadoras no tienen que ser particularmente poderosas para desarrollar aplicaciones iOS (Mark Goadrich y Michael Rogers, 2011). Todo lo que se necesita para ejecutar el software es un computador Mac con procesador basado en Intel y al menos 10 GB de espacio libre en disco (Eero Maasalmi y Panu Pitkänen, 2011).

2.4.5 Términos y condiciones de licencia

En la actualidad, existen cuatro opciones de licencia para el desarrollo de aplicaciones:

- **Desarrollador Apple (Apple Developer).**-Este registro es gratuito, y permite el acceso del desarrollador a recursos de documentación y al iOS SDK, no incluye la licencia para publicar aplicaciones en el App Store. Otra limitación del registro gratuito es que únicamente se pueden correr las aplicaciones sobre el simulador. Para crear aplicaciones que pueden ejecutarse en un dispositivo real, es necesario inscribirse en el Programa de Desarrolladores de iOS.
- **Programa de Universidades iOS (iOS University Program).**- Esta opción es completamente gratuita, aunque solo aplica a determinadas universidades que estén incluidas dentro del programa. Obliga a que no existan fines de lucro, y que todo se destine a la formación de alumnos en las herramientas de Apple.



UNIVERSIDAD DE CUENCA

- **Programa de Desarrolladores de iOS (iOS Developer Program).**- Esta opción tiene un costo de 99 dólares al año, y nos ofrece adicionalmente las siguientes opciones:
 - Documentación técnica (de versiones beta)
 - Herramientas de desarrollo (de versiones beta)
 - Descarga de firmwares beta para probar las aplicaciones
 - Permite instalar aplicaciones en dispositivos Apple reales, registrándolos previamente en la Web de Apple.
 - Permite publicar aplicaciones en la AppStore y comercializarlas con beneficios del 70% sobre el precio de venta que se establezcan.
- **Programa Empresarial de Desarrolladores iOS (iOS Developer Enterprise Program).**- Esta opción tiene un costo de 299 dólares al año, y está orientada al desarrollo de aplicaciones de tipo In-house (aplicaciones corporativas privadas). Permite publicar de forma privada aplicaciones que puedan ser instaladas en los dispositivos de los diferentes empleados de una empresa.(iOS Developer, 2012)

2.4.6 Documentación y recursos

La biblioteca de desarrolladores iOS contiene la documentación, código de ejemplo, tutoriales y otra información que se necesita para escribir aplicaciones iOS. Al instalar el SDK de iOS, XCode automáticamente permite que la biblioteca de desarrolladores iOS se encuentre disponible para su uso local fuera de línea. El contenido de la biblioteca para desarrolladores iOS se actualiza con frecuencia. Sin embargo, también es posible acceder a la documentación, notas de la versión, notas técnicas, y código de ejemplo desde el Centro de Desarrolladores iOS en línea. Todos los documentos están disponibles en HTML y la mayoría también están disponibles en formato PDF (iOS Developer, 2012).

La biblioteca de desarrollador de iOS, se encuentra dividida en cinco secciones para facilitar la búsqueda por parte del usuario:

Recursos

- Notas técnicas
- Guías
- Códigos de ejemplo
- Preguntas y respuestas técnicas
- Notas de lanzamiento
- Referencias
- Artículos
- Videos
- Tareas en Xcode



UNIVERSIDAD DE CUENCA

Temas

- General
- Audio&Video
- Manejo de la información
- Gráficos y animación
- Cálculo Matemático
- Internet&Networking
- Rendimiento
- Seguridad
- Herramientas&Lenguajes
- Experiencias de los usuarios
- Frameworks
- Acelerómetro
- Herramientas de audio

Cocoa Touch

- Núcleo de audio
- Núcleo de bluetooth
- Núcleo de información
- Núcleo base
- Núcleo de gráficos
- Núcleo de imágenes
- Núcleo de texto
- Núcleo de localización
- Núcleo de video
- Núcleo de movimiento

Núcleo del Sistema Operativo

- Núcleo MIDI
- Núcleo de telefonía
- Equipo de eventos
- Accesorios externos
- Equipo de juegos
- Equipo de mapas

Multimedia

- Intérprete multimedia
- Mensajes de la interfaz de usuario
- Servicios del núcleo móvil
- OpenGL ES
- Seguridad
- Equipo de tiendas



UNIVERSIDAD DE CUENCA

- Sistema
- Twitter
- Automatización de interfaz de usuario
- Equipo de interfaz de usuario

2.4.7 Usabilidad

El nivel de interacción que permite iOS es uno de los mejores en cuanto a sistemas operativos móviles. Las aplicaciones pueden compartir información unas con otras y, además, Apple incluye un menú multitarea por lo que es sencillo para el usuario, poder ir de una aplicación a otra.

Los desarrolladores iOS tienen una ventaja al momento de diseñar sus aplicaciones para que brinden una gran experiencia de usuario. Al correr las aplicaciones exclusivamente en dispositivos Apple, el desarrollador sabe exactamente las características de dichos dispositivos. Esto sin duda evita al desarrollador la tarea de adaptar el contenido de la aplicación para diferentes tamaños de pantalla, resoluciones, tecnologías, etc. permitiéndole concentrar mayor esfuerzo en diseñar correctamente la funcionalidad de la aplicación.

2.5 iOS Y ANDROID EN UN CONTEXTO EDUCATIVO

En las secciones anteriores se presentaron los sistemas operativos iOS y Android desde una perspectiva técnica enmarcada en el desarrollo de aplicaciones en general. A continuación, se estudian ambos sistemas operativos móviles como plataformas de soporte para aplicaciones y contenido orientado a la enseñanza-aprendizaje.

2.5.1 Antecedentes de uso de tecnología móvil en la educación

El uso de tecnología móvil para la enseñanza todavía se encuentra en una fase experimental; sin embargo, existen proyectos que pretenden difundir la educación móvil como una alternativa a la formación tradicional. Países como Francia, Japón, España, Reino Unido y Estados Unidos han tomado la iniciativa, con programas que fomentan el uso de este tipo de tecnología en la educación. La mayoría de estos programas son a pequeña escala, guiados por alguna institución, y tienen por objetivo estudiar las posibilidades presentadas por los dispositivos móviles en entornos de aprendizaje.

A continuación, se listan algunas de las iniciativas principales de educación móvil que utilizan los sistemas operativos iOS y Android como plataformas tecnológicas.

2.5.1.1 iUOC

La Universidad Oberta de Catalunya (UOC) junto con la compañía de telecomunicaciones Orange Spain, realizaron el proyecto iUOC. Se trata de una aplicación de “campus móvil” cuyo objetivo es hacer al campus virtual de la



UNIVERSIDAD DE CUENCA

UOC más interactivo y sobretodo, que esté disponible para dispositivos móviles. Es uno de los proyectos referentes en España, en cuanto a educación móvil. Algunos puntos que describen el proyecto son los siguientes:

- Se desarrolló una aplicación nativa para el iPad.
- Se utilizó la tecnología RSS para proveer una conexión para la comunicación.
- Espacios para estudiantes y profesores en el campus virtual.
- Se adaptaron los materiales educativos de la UOC para el iPad.
- Los estudiantes tienen la capacidad de revisar su correo, agendas, materiales educativos, salas de clase y las opiniones de otros estudiantes.

Los profesores y estudiantes respondieron favorablemente al proyecto. Calificaron a la aplicación iUOC como amigable con el usuario e intuitiva. También tuvieron buenas críticas para el iPad como medio para obtener información, como herramienta de comunicación y como soporte interactivo para el aprendizaje; sin embargo, expresaron que el iPad no es una buena herramienta para escribir y crear contenido y que los estudiantes y profesores no conocían las posibilidades pedagógicas y técnicas completas de estos dispositivos (GSMA, 2011b).

2.5.1.2 Univ Mobile

Univ Mobile es una aplicación móvil desarrollada en Francia por estudiantes en la ciudad de Paris. Cuenta con una aplicación nativa para iPhone y también con un sitio Web accesible desde cualquier smartphone conectado a internet. El objetivo de esta aplicación es ofrecer a los estudiantes información práctica en diferentes módulos, como por ejemplo (GSMA, 2011c):

- **Mon ENT:** Búsquedas en el directorio de las universidades, consulta de horarios, fechas de exámenes, resultados de pruebas y mensajes.
- **GeoCampus:** este módulo incluye un mapa interactivo del campus y localiza servicios disponibles como restaurantes, zonas WiFi, etc.
- Además existen otros módulos como noticias, podcasts¹¹, etc.

2.5.1.3 WaPEduc 2.0

Se trata de un servicio comercial, caracterizado por ser el primero en ofrecer el servicio de aprendizaje móvil en Francia. El objetivo de este servicio es muy claro: proveer de aprendizaje para los estudiantes a cualquier hora y en cualquier lugar. Entre las características que destacan en este servicio se encuentran (GSMA, 2011c):

¹¹El podcasting consiste en la distribución de archivos multimedia (normalmente audio o vídeo, que puede incluir texto como subtítulos y notas) mediante un sistema de redifusión (RSS) que permita suscribirse y usar un programa que lo descarga para que el usuario lo escuche en el momento que quiera. No es necesario estar suscrito para descargarlos.



UNIVERSIDAD DE CUENCA

- La plataforma para teléfonos móviles ofrece acceso al estudiante a materiales del curso, cuestionarios, tutoriales, etc.
- Más de 110 lecciones disponibles.
- Permite la colaboración entre profesores y estudiantes fuera del salón de clases.
- Páginas de problemas y columnas de consejos.
- Documentos interactivos de preguntas que permiten al estudiante auto-evaluarse.

Este servicio está disponible para todos los smartphones; sin embargo, cuenta con una aplicación específica para el iPhone.

2.5.1.4 iStanford Mobile Campus App

Se trata de una de las primeras aplicaciones de tipo campus virtual para dispositivos móviles que fueron lanzadas al mercado en Estados Unidos. Se encuentra disponible tanto para iOS como para Android. Entre sus características principales se destacan (GSMA, 2011d):

- Búsqueda de cursos y matriculación en clases.
- Capacidad para ver notas y nóminas de estudiantes.
- Búsqueda de información pública de directorio.
- Búsquedas en el mapa del campus, rutas y estaciones de buses, etc.
- Información sobre eventos en el campus.
- Leer noticias del campus.

Como se ha visto, existen varios proyectos educativos alrededor del mundo que utilizan dispositivos móviles, y en su gran mayoría se inclinan por iOS para el desarrollo de las aplicaciones dirigidas a los estudiantes. Esto se debe a la gran aceptación que los jóvenes han tenido con los dispositivos iPad de Apple, los cuales superan a sus principales rivales como el PlayBook de BlackBerry y las Tablets que utilizan el sistema operativo Android de Google.

Pero esto no ocurre para todos los dispositivos, Android ha logrado una rápida consolidación y una cuota de mercado muy importante en cuanto a teléfonos móviles, lo que hace suponer que en un futuro muy cercano las aplicaciones no deberían centrarse en una sola plataforma ni en un solo grupo de dispositivos, sino estar disponibles para cualquier tecnología a la que tengan acceso los usuarios.

2.5.2 La disponibilidad de los dispositivos móviles: iOS y Android dentro de la Facultad de Ingeniería de la Universidad de Cuenca

Un criterio importante a ser tomado en cuenta dentro de un proyecto educativo que se apoye en el uso de dispositivos móviles es la disponibilidad de los mismos. Con la finalidad de conocer la preferencia de los estudiantes con respecto al uso de los dispositivos móviles en nuestro medio, se vio



UNIVERSIDAD DE CUENCA

conveniente realizar una encuesta (**ver Anexo A**) que refleje resultados fehacientes sobre el tema. La encuesta fue realizada a estudiantes del primer año de la Facultad de Ingeniería de la Universidad de Cuenca, ya que pertenecen a cursos numerosos y, posiblemente, serán idóneos para la implementación y pruebas del proyecto. Mediante la encuesta se pudo conocer: el tipo de dispositivo móvil más difundido, la predilección por una u otra plataforma móvil y si existe predisposición por parte de los estudiantes para utilizar los dispositivos móviles con fines educativos.

Del total de encuestados 69 pertenecían al sexo masculino y 09 al sexo femenino. Los resultados de dicha encuesta, nos reveló que:

- La mayoría de estudiantes poseen dispositivos móviles antiguos, que únicamente funcionan como teléfono celular, y no cuentan con las características de un smartphone.
- Existe un marcado desconocimiento por parte de los estudiantes en cuanto al sistema operativo que utiliza su dispositivo móvil, sin embargo, la plataforma preferida por la mayoría es iOS.
- La gran mayoría de estudiantes no acceden a internet utilizando su dispositivo móvil.
- El uso que la mayoría de estudiantes da a sus dispositivos móviles tienen que ver principalmente con la comunicación y el entretenimiento.
- La gran mayoría de estudiantes está dispuesto a utilizar un dispositivo móvil con fines educativos.
- Las limitaciones u obstáculos, que tendrían los estudiantes al utilizar un dispositivo móvil para estudiar alguna asignatura, tienen que ver principalmente con el costo del dispositivo y el acceso a internet.

2.6 COMPARACION ENTRE LOS SISTEMAS OPERATIVOS iOS Y ANDROID

En la siguiente sección se realiza la comparación entre los dos sistemas operativos móviles evaluados: iOS y Android. Esta comparación está basada en los criterios definidos en la sección 2 del presente documento. La evaluación de dichos criterios está enmarcada estrictamente en el marco del proyecto y sujeta a la investigación realizada sobre ambas plataformas. El objetivo al realizar esta comparación, es determinar la plataforma que se adapta mejor a las necesidades del proyecto y no pretende de ninguna forma evaluar cuál de los sistemas operativos es mejor.

Cada criterio será evaluado a su vez mediante subcriterios en una escala cualitativa que mide el nivel de adaptación de los mismos con respecto al proyecto. Los rangos de valores asociados con dicha escala están descritos en la **¡Error! No se encuentra el origen de la referencia..**



UNIVERSIDAD DE CUENCA

Escala de valoración	
Rango (%)	Adaptación
0-24	Nada
25-49	Poco
50-74	Parcialmente
75-100	Totalmente

Escala de valoración	
Rango (%)	Adaptación
0-24	Nada
25-49	Poco
50-74	Parcialmente
75-100	Totalmente

Escala de valoración	
Rango (%)	Adaptación
0-24	Nada
25-49	Poco
50-74	Parcialmente
75-100	Totalmente

Tabla 2.6-A Escala de Valoración

2.6.1 Arquitectura

En cuanto a arquitectura, iOS y Android tienen esquemas similares distribuidos en capas. Ambos sistemas operativos móviles están basados en UNIX, lo cual permite niveles de abstracción que protegen a las aplicaciones de cambios de hardware y permiten su ejecución en dispositivos con diferentes características y capacidades. La arquitectura bien definida de los dos sistemas operativos, agrupa los servicios de cada capa de acuerdo a su función, facilitando al desarrollador la búsqueda de recursos cuando requiere cubrir una determinada funcionalidad del sistema.



UNIVERSIDAD DE CUENCA

La **¡Error! No se encuentra el origen de la referencia.**muestra la comparación de las arquitecturas de iOS y Android en cuanto a escalabilidad, desempeño y consistencia refiriéndonos a las necesidades específicas del proyecto.

	Arquitectura			
	Escalabilidad	Desempeño	Consistencia	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

	Arquitectura			
	Escalabilidad	Desempeño	Consistencia	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

	Arquitectura			
	Escalabilidad	Desempeño	Consistencia	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Tabla 2.6-B Comparación de arquitecturas

2.6.2 Framework de desarrollo

Tanto iOS como Android permiten el desarrollo de aplicaciones en un ambiente de alto nivel. Ambos sistemas operativos cuentan con bloques de construcción o frameworks que utilizan los servicios de las capas de más bajo nivel, evitando al desarrollador centrarse en detalles tediosos como por ejemplo, la gestión de memoria. A continuación se detalla la comparación de los componentes principales del framework de desarrollo de cada plataforma.

2.6.2.1 SDK

Tanto Apple (iOS) como Google (Android) permiten la descarga gratuita del SDK, con la diferencia que Apple requiere registrarse previamente como desarrollador para acceder a dicha descarga. La instalación del SDK de Android resulta más dificultosa, debido a que el usuario debe instalar y configurar por su cuenta, y en un determinado orden, el JDK, el IDE y finalmente el SDK. En el caso de iOS esto no es necesario, ya que el paquete



UNIVERSIDAD DE CUENCA

del SDK instala y configura la suite completa con el IDE y el entorno de ejecución.

La **¡Error! No se encuentra el origen de la referencia.**muestra la comparación del SDK de cada sistema operativo en cuanto a disponibilidad y facilidad de instalación.

SDK			
	Disponibilidad	Facilidad de instalación	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Parcialmente	Totalmente

SDK			
	Disponibilidad	Facilidad de instalación	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Parcialmente	Totalmente

Tabla 2.6-C Comparación del SDK

2.6.2.2 Lenguaje de Programación

Tanto Java en el caso de Android como Objective-C en iOS, son lenguajes de programación orientados a objetos que resultan fáciles de aprender. Al ser descendientes del lenguaje C, manejan sintaxis similares aunque con diferencias en cuanto a semántica. Para empezar a programar en Java u Objective-C, simplemente es necesario estar familiarizado con el paradigma de programación orientada a objetos ya que ambos lenguajes cuentan con la documentación necesaria y códigos de ejemplo que resultan de gran ayuda para los programadores.

En cuanto a seguridad, las aplicaciones escritas en lenguajes interpretados como Java son más robustas ante problemas relacionados con el desbordamiento de buffer. En el caso de Objective-C, las solicitudes están vinculadas a las bibliotecas de C, lo que permite usar funciones de manejo de cadenas de caracteres como "strcat" que son propicias para un ataque de este tipo. Esta diferencia entre ambos lenguajes resulta importante, ya que no tomar las previsiones adecuadas puede conllevar un riesgo al momento de almacenar la información que utilizan las aplicaciones.

Finalmente, la experiencia que tienen los desarrolladores del presente proyecto con Java, es superior a la que poseen trabajando con el lenguaje Objective-C, siendo esta una consideración muy importante al momento de escoger la



UNIVERSIDAD DE CUENCA

plataforma sobre la cual se desarrollará el prototipo. Sin embargo, no se tiene experiencia previa con el lenguaje Java en aplicaciones móviles, sino únicamente en aplicaciones de escritorio.

La **¡Error! No se encuentra el origen de la referencia.** resume lo anteriormente citado, y muestra la comparación de acuerdo con las características de los lenguajes de programación utilizados para el desarrollo de aplicaciones en las dos plataformas evaluadas.

	Lenguaje de Programación				
	Facilidad de Aprendizaje	Experiencia en Desarrollo	Soporte	Seguridad	Resultado
iOS (Objective-C)	Totalmente	Poco	Totalmente	Parcialmente	Parcialmente
ANDROID (Java)	Totalmente	Parcialmente	Totalmente	Totalmente	Totalmente

Tabla 2.6-D Comparación de Lenguajes de Programación

2.6.2.3 IDE

En lo referente a entornos de desarrollo, tanto Eclipse (Android) como Xcode (iOS) cuentan con características importantes para la construcción de aplicaciones móviles. Ambos ofrecen prestaciones similares, sin embargo, la diferencia es que la instalación de Xcode incluye por defecto las herramientas del SDK requeridas para el diseño, desarrollo, depuración y pruebas de aplicaciones para iOS, mientras que en el caso de Eclipse es necesario instalar el SDK de Android posteriormente a la instalación del IDE para poder contar con dichas herramientas.

Una de las principales diferencias entre estos dos IDE tiene que ver con el diseño de interfaces de usuario. Xcode permite un diseño más intuitivo y sencillo a nivel de vistas, pero con la restricción de que no es posible acceder fácilmente al código XML que se encuentra por detrás. En Eclipse por el contrario, la mayor parte del diseño se realiza directamente manipulando código XML, lo cual puede resultar un tanto tedioso y complicado.

Otra diferencia se da al momento de agrupar archivos pertenecientes a un proyecto en diferentes carpetas. Xcode lo hace, pero únicamente de una manera lógica, es decir, no se crean realmente estas carpetas en el disco duro.



UNIVERSIDAD DE CUENCA

Lo contrario ocurre en Eclipse, donde al agrupar los archivos, estaremos realmente creando nuevas carpetas en el sistema de ficheros. Esto no representa gran diferencia en cuanto al desarrollo en sí, pero resulta importante al momento de organizar los componentes del proyecto. Finalmente, tal como ocurre con el lenguaje de programación, la experiencia de los desarrolladores del presente proyecto con Eclipse es superior a la que poseen trabajando con Xcode, lo cual es un criterio importante a ser tomado en cuenta.

La **¡Error! No se encuentra el origen de la referencia.** contiene la comparación de las características de los IDE utilizados para el desarrollo de aplicaciones en las dos plataformas evaluadas.

	IDE				
	Prestaciones	Disponibilidad	Experiencia Previa	Facilidad de instalación	Resultado
iOS (Xcode)	Totalmente	Totalmente	Poco	Totalmente	Parcialmente
ANDROID (Eclipse)	Totalmente	Totalmente	Parcialmente	Parcialmente	Parcialmente

Tabla 2.6-E Comparación de IDE

2.6.2.4 Librerías

Tanto Android como iOS cuentan con una gran cantidad de librerías incluidas en sus respectivos frameworks de desarrollo, lo cual resulta de mucha utilidad para los programadores de aplicaciones móviles. Sin embargo, las librerías de iOS son creadas de una manera más específica para funcionar en un tipo exclusivo de dispositivos (iPhone, iPad, iPod Touch). Esto resulta ventajoso para el desarrollador de aplicaciones iOS, pues existe una considerable cantidad de código hecho y probado, que puede ser reutilizado. En el caso de Android, se tiene librerías más generales que amplían las posibilidades de emplearlas de diferentes maneras, pero requieren que el programador desarrolle muchas cosas partiendo prácticamente desde cero, lo que implica utilizar más tiempo y esfuerzo.

Un punto importante a ser valorado es que tanto Android como iOS admiten la inclusión de librerías de terceros en el desarrollo de aplicaciones. Esto sin duda, permite a los desarrolladores trabajar con la tecnología que consideren más apropiada para cubrir un determinado fin y no limitarse a las librerías nativas que vienen en la plataforma.



UNIVERSIDAD DE CUENCA

La **¡Error! No se encuentra el origen de la referencia.** muestra la comparación entre las librerías de Android y iOS, tomando en cuenta la disponibilidad, funcionalidad y la posibilidad de incluir librerías de terceros.

Librerías				
	Disponibilidad	Funcionalidad	Posibilidad de incluir librerías de terceros	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Parcialmente	Totalmente	Totalmente

Librerías				
	Disponibilidad	Funcionalidad	Posibilidad de incluir librerías de terceros	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Parcialmente	Totalmente	Totalmente

Librerías				
	Disponibilidad	Funcionalidad	Posibilidad de incluir librerías de terceros	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Parcialmente	Totalmente	Totalmente

Tabla 2.6-F Comparación de Librerías



UNIVERSIDAD DE CUENCA

2.6.2.5 Tipos de aplicaciones

En el mercado existe una gran variedad de aplicaciones creadas para funcionar tanto en iOS como en Android. Esta diversidad pone en evidencia la capacidad de desarrollar cualquier tipo de aplicaciones sobre estas dos tecnologías, ya sean de carácter educativo, utilitarios o de entretenimiento. Sin embargo, se debe considerar que Android, ejecuta sus aplicaciones sobre la máquina virtual Dalvik, lo que brinda una gran portabilidad para correr en diferentes dispositivos de distintos fabricantes pero implica una penalización en cuanto al rendimiento; Por el contrario las aplicaciones iOS se ejecutan directamente sobre el sistema operativo, ofreciendo una respuesta teóricamente más rápida, pero las aplicaciones se encuentran restringidas a correr exclusivamente en dispositivos Apple.

Otro factor importante a ser considerado es la fragmentación que existe en la plataforma Android debido al gran número de fabricantes de dispositivos que utilizan este sistema operativo. Esto genera el riesgo de que algunas aplicaciones no funcionen correctamente en todos los dispositivos Android, a pesar de cumplir con los requerimientos básicos de desarrollo. Como consecuencia, los desarrolladores pueden llegar a perder tiempo valioso intentando que las aplicaciones funcionen para todos los dispositivos, en lugar de invertir más tiempo en la esencia misma del producto. Por último, tanto Android como iOS permiten la creación de aplicaciones nativas y para la Web, lo cual brinda flexibilidad a los desarrolladores al momento de escoger la forma en la que se distribuirán sus aplicaciones.

La comparación de los tipos de aplicaciones que se pueden desarrollar tanto para iOS como para Android se muestra en la **¡Error! No se encuentra el origen de la referencia..**



UNIVERSIDAD DE CUENCA

Tipos de Aplicaciones				
	Portabilidad	Soporte para aplicaciones Nativas	Soporte para aplicaciones Web	Resultado
iOS	Poco	Totalmente	Totalmente	Parcialmente
ANDROID	Parcialmente	Totalmente	Totalmente	Totalmente

Tipos de Aplicaciones				
	Portabilidad	Soporte para aplicaciones Nativas	Soporte para aplicaciones Web	Resultado
iOS	Poco	Totalmente	Totalmente	Parcialmente
ANDROID	Parcialmente	Totalmente	Totalmente	Totalmente

Tipos de Aplicaciones				
	Portabilidad	Soporte para aplicaciones Nativas	Soporte para aplicaciones Web	Resultado
iOS	Poco	Totalmente	Totalmente	Parcialmente
ANDROID	Parcialmente	Totalmente	Totalmente	Totalmente

Tabla 2.6-G Comparación de los tipos de aplicaciones

2.6.3 Requisitos de software

El SDK tanto de Android como de iOS, contiene el software necesario para el desarrollo de aplicaciones móviles y el emulador del dispositivo para realizar pruebas. Los inconvenientes que se pueden presentar en cuanto a requisitos de software tienen que ver con la disponibilidad del sistema operativo. En el caso de Android es posible desarrollar aplicaciones tanto en Windows como en Linux y Mac, pero para iOS la única plataforma que soporta el framework de desarrollo es Mac OS X 10.6 Snow Leopard o superiores. Para el presente proyecto se cuenta con el software necesario para desarrollar en ambas plataformas.

La **¡Error! No se encuentra el origen de la referencia.** resume la comparación en cuanto a requisitos de software para el desarrollo de aplicaciones iOS y Android en el presente proyecto.



UNIVERSIDAD DE CUENCA

Requisitos de Software				
	Disponibilidad del SO	Facilidad para configurar el software de desarrollo	Disponibilidad de software para simulación/pruebas	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Requisitos de Software				
	Disponibilidad del SO	Facilidad para configurar el software de desarrollo	Disponibilidad de software para simulación/pruebas	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Requisitos de Software				
	Disponibilidad del SO	Facilidad para configurar el software de desarrollo	Disponibilidad de software para simulación/pruebas	Resultado
iOS	Totalmente	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Tabla 2.6-H Comparación de requisitos de software

2.6.4 Requisitos de hardware

En el caso de Android, es posible desarrollar aplicaciones utilizando cualquier PC o computador portátil que cumpla los requisitos de hardware para ejecutar los sistemas operativos Windows, Linux o Mac. Para desarrollar aplicaciones en iOS es indispensable contar con computadores Apple que cumplan los requisitos de hardware para ejecutar el sistema MAC OS X Snow Leopard o superiores. Para el desarrollo del presente proyecto contamos con los equipos de hardware necesarios para desarrollar aplicaciones en las dos plataformas, así como también equipos portátiles para realizar pruebas.

La **¡Error! No se encuentra el origen de la referencia.** resume la comparación en cuanto a requisitos de hardware para el desarrollo de aplicaciones iOS y Android en el presente proyecto.



UNIVERSIDAD DE CUENCA

Requisitos de Hardware			
	Disponibilidad de equipos para el desarrollo	Disponibilidad de equipos para pruebas	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente

Requisitos de Hardware			
	Disponibilidad de equipos para el desarrollo	Disponibilidad de equipos para pruebas	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente

Requisitos de Hardware			
	Disponibilidad de equipos para el desarrollo	Disponibilidad de equipos para pruebas	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente

Tabla 2.6-I Comparación de requisitos de hardware

2.6.5 Términos y condiciones de licencia

El acceso a la descarga del SDK tanto para iOS como para Android es gratuito. Esto permite que el desarrollo de aplicaciones en ambas plataformas se pueda realizar libremente, es decir, sin la necesidad de realizar ningún pago. Sin embargo, esto no ocurre al momento de distribuir las aplicaciones, ya que en ambos casos se requiere pagar una cuota de registro.

Para publicar aplicaciones Android, el desarrollador necesita registrarse en el servicio Google Play usando una cuenta Google. Para ello debe crearse un perfil de desarrollador, pagar una cuota de registro de USD \$25,00 (13/04/2012) y aceptar los términos del servicio. Una vez culminado el registro, el desarrollador puede subir la aplicación a los servidores de Google Play, poner una descripción de la misma y finalmente publicarla para que pueda ser vista, descargada y calificada por los usuarios.

En el caso de iOS, para que los desarrolladores tengan la posibilidad de publicar sus aplicaciones necesitan ser miembros del Programa de Desarrolladores iOS o del Programa Empresarial de Desarrolladores de iOS. El pago por mantener una cuenta activa en cualquiera de estos dos programas se realiza anualmente, y en el primer caso es de USD \$99,00 y en el segundo caso USD \$299,00 (13/04/2012). Sin embargo, para que una aplicación sea



UNIVERSIDAD DE CUENCA

publicada en el App Store de Apple, primero necesita pasar por un proceso de validación, cosa que no ocurre con Android y Google Play.

En cuanto a políticas de distribución, tanto Apple (iOS) como Google (Android) permiten que los desarrolladores puedan distribuir sus aplicaciones gratuitamente o mediante paga. En el caso de aplicaciones gratuitas, el desarrollador no incurre en ningún tipo de pago para que la aplicación sea publicada. Cuando se trata de aplicaciones pagadas, el desarrollador obtiene el 70% del precio de la aplicación, el 30% restante es para el distribuidor.

La **¡Error! No se encuentra el origen de la referencia.** resume el resultado de la comparación entre los términos y condiciones de licencia de las dos plataformas evaluadas, en cuanto a costos de desarrollo y distribución de las aplicaciones.

Términos y condiciones de licencia			
	Evita costos de desarrollo	Evita costos de distribución	Resultado
iOS	Parcialmente	Poco	Parcialmente
ANDROID	Totalmente	Parcialmente	Totalmente

Términos y condiciones de licencia			
	Evita costos de desarrollo	Evita costos de distribución	Resultado
iOS	Parcialmente	Poco	Parcialmente
ANDROID	Totalmente	Parcialmente	Totalmente

Términos y condiciones de licencia			
	Evita costos de desarrollo	Evita costos de distribución	Resultado
iOS	Parcialmente	Poco	Parcialmente
ANDROID	Totalmente	Parcialmente	Totalmente

Términos y condiciones de licencia			
	Evita costos de desarrollo	Evita costos de distribución	Resultado
iOS	Parcialmente	Poco	Parcialmente
ANDROID	Totalmente	Parcialmente	Totalmente

Tabla 2.6-A Comparación de Términos y condiciones de licencia



UNIVERSIDAD DE CUENCA

2.6.6 Documentación y recursos

Tanto Android como iOS incluyen en la descarga del SDK la documentación de referencia del mismo, además de códigos de ejemplo y notas de lanzamiento. En Android, la documentación contiene las mejores prácticas para el diseño, desarrollo y distribución de aplicaciones, APIs con detalle de clases e interfaces, artículos técnicos y tutoriales. En iOS, la documentación contiene: código de ejemplo, tutoriales y recursos técnicos. Ambas plataformas cuentan con documentación de muy buena calidad, sin embargo, al ser Android un sistema abierto la documentación técnica permite llegar al detalle del código de las API propias de la plataforma, algo que no ocurre en iOS donde este tipo de información es escasa o nula.

La **¡Error! No se encuentra el origen de la referencia.** resume la comparación entre la documentación y recursos técnicos de ambas plataformas en términos de disponibilidad y calidad de los mismos.

Documentación y recursos				
	Disponibilidad de Documentación y recursos	Nivel de detalle en documentación técnica	Calidad de Documentación y recursos	Resultado
iOS	Totalmente	Parcialmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Documentación y recursos				
	Disponibilidad de Documentación y recursos	Nivel de detalle en documentación técnica	Calidad de Documentación y recursos	Resultado
iOS	Totalmente	Parcialmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Documentación y recursos				
	Disponibilidad de Documentación y recursos	Nivel de detalle en documentación técnica	Calidad de Documentación y recursos	Resultado
iOS	Totalmente	Parcialmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente	Totalmente

Tabla 2.6-K Comparación de Documentación y recursos



UNIVERSIDAD DE CUENCA

2.6.7 Usabilidad

En cuanto a términos de usabilidad, tanto iOS como Android permiten la creación de aplicaciones con interfaces agradables e intuitivas. No existen mayores diferencias entre ambas plataformas al momento de evaluar la experiencia de usuario que ofrecen, sin embargo, el hardware puede llegar a ser determinante en este sentido. Aunque es posible adquirir teléfonos Android de muy alta calidad, también se puede comprar teléfonos baratos muy mal diseñados que pueden causar una terrible experiencia de usuario. Esto no es problema en el caso de iOS, ya que los dispositivos de Apple se caracterizan por su hardware de excelente calidad y diseño.

La **¡Error! No se encuentra el origen de la referencia.** muestra la comparación entre la usabilidad de las dos plataformas evaluadas.

Usabilidad			
	Intuitivo	Agradable	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente

Usabilidad			
	Intuitivo	Agradable	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente

Usabilidad			
	Intuitivo	Agradable	Resultado
iOS	Totalmente	Totalmente	Totalmente
ANDROID	Totalmente	Totalmente	Totalmente

Tabla 2.6-L Comparación de Usabilidad

2.6.8 Disponibilidad de equipos

Para conocer la disponibilidad de equipos que utilicen iOS o Android dentro del entorno del proyecto, es decir, en la Facultad de Ingeniería de la Universidad de Cuenca, se realizó una encuesta a los estudiantes del primer año, la misma que reveló los resultados indicados en la **¡Error! No se encuentra el origen de la referencia.** en cuanto a estos dos sistemas operativos móviles evaluados.



UNIVERSIDAD DE CUENCA

Disponibilidad de equipos			
	Poseen actualmente	Planean adquirir	Resultado
iOS	19 %	34 %	Poco
ANDROID	15 %	27 %	Poco

Disponibilidad de equipos			
	Poseen actualmente	Planean adquirir	Resultado
iOS	19 %	34 %	Poco
ANDROID	15 %	27 %	Poco

Tabla 2.6-M Comparación de disponibilidad de equipos

2.6.9 Antecedentes

Tanto Android como iOS son opciones válidas como plataformas tecnológicas de soporte para proyectos de educación móvil. Sin embargo, la investigación realizada revela la existencia de muchos más proyectos que trabajan con el sistema operativo iOS. En muchos de los casos, esto se debe a la existencia de programas de apoyo por parte de Apple, que busca difundir sus productos en el contexto educativo.

La **¡Error! No se encuentra el origen de la referencia.** muestra la comparación entre los antecedentes de uso de ambas plataformas en proyectos de educación móvil.



UNIVERSIDAD DE CUENCA

	Antecedentes de uso en proyectos de educación móvil
iOS	Totalmente
ANDROID	Poco

	Antecedentes de uso en proyectos de educación móvil
iOS	Totalmente
ANDROID	Poco

	Antecedentes de uso en proyectos de educación móvil
iOS	Totalmente
ANDROID	Poco

Tabla 2.6-N Comparación de Antecedentes

2.7 RESUMEN DEL CAPITULO

Los resultados obtenidos de la evaluación de cada uno de los criterios de comparación tanto para iOS como Android se resumen en la siguiente tabla. Error! No se encuentra el origen de la referencia..



UNIVERSIDAD DE CUENCA

	RESULTADOS	
	iOS	ANDROID
Arquitectura	Totalmente	Totalmente
SDK	Totalmente	Totalmente
Lenguaje de Programación	Parcialmente	Totalmente
IDE	Parcialmente	Parcialmente
Librerías	Totalmente	Totalmente
Tipos de Aplicaciones	Parcialmente	Totalmente
Requisitos de Software	Totalmente	Totalmente
Requisitos de Hardware	Totalmente	Totalmente
Términos y condiciones de Licencia	Parcialmente	Totalmente
Documentación y Recursos	Totalmente	Totalmente
Usabilidad	Totalmente	Totalmente
Disponibilidad de equipos	Poco	Poco
Antecedentes en proyectos educativos	Totalmente	Poco

	RESULTADOS	
	iOS	ANDROID
Arquitectura	Totalmente	Totalmente
SDK	Totalmente	Totalmente
Lenguaje de Programación	Parcialmente	Totalmente
IDE	Parcialmente	Parcialmente
Librerías	Totalmente	Totalmente
Tipos de Aplicaciones	Parcialmente	Totalmente
Requisitos de Software	Totalmente	Totalmente
Requisitos de Hardware	Totalmente	Totalmente
Términos y condiciones de Licencia	Parcialmente	Totalmente
Documentación y Recursos	Totalmente	Totalmente
Usabilidad	Totalmente	Totalmente
Disponibilidad de equipos	Poco	Poco
Antecedentes en proyectos educativos	Totalmente	Poco



UNIVERSIDAD DE CUENCA

	RESULTADOS	
	iOS	ANDROID
Arquitectura	Totalmente	Totalmente
SDK	Totalmente	Totalmente
Lenguaje de Programación	Parcialmente	Totalmente
IDE	Parcialmente	Parcialmente
Librerías	Totalmente	Totalmente
Tipos de Aplicaciones	Parcialmente	Totalmente
Requisitos de Software	Totalmente	Totalmente
Requisitos de Hardware	Totalmente	Totalmente
Términos y condiciones de Licencia	Parcialmente	Totalmente
Documentación y Recursos	Totalmente	Totalmente
Usabilidad	Totalmente	Totalmente
Disponibilidad de equipos	Poco	Poco
Antecedentes en proyectos educativos	Totalmente	Poco

Tabla 2.7-A Resumen de evaluación de criterios

En conclusión, tanto iOS como Android son buenas opciones para actuar como plataforma tecnológica de soporte para el desarrollo de un proyecto educativo. Sin embargo, en el caso puntual del presente proyecto, el sistema operativo móvil que se adapta de mejor manera es Android.

Con la elección de Android como plataforma sobre la cual se desarrollará la herramienta de aprendizaje móvil culmina el Capítulo 1 de la tesis. El siguiente paso es definir las características de la metodología de enseñanza que se pretende aplicar con la implementación de este prototipo de aprendizaje en la materia piloto *Cálculo Diferencial*, este análisis se lo realiza en el Capítulo 2.



UNIVERSIDAD DE CUENCA

CAPITULO 3:

“Definición de las características principales de la metodología constructivista y los objetos de aprendizaje.”



UNIVERSIDAD DE CUENCA

3.1 INTRODUCCIÓN

El cambio tecnológico, al que se encuentra constantemente sometido nuestro entorno, ha ido revolucionando muchos aspectos de la vida del ser humano con el paso del tiempo. Las nuevas formas de comunicación, transporte, generación de energía, e inclusive entretenimiento y alimentación, son ejemplos de cómo la tecnología ha permitido la evolución de tareas consideradas como tradicionales y esenciales en el día a día de las personas. Sin embargo, este potencial tecnológico, no ha sido completamente aprovechado para mejorar la educación en centros de formación de alumnos de todo nivel, quedándose en cambios de forma que únicamente han “automatizado” el método educativo tradicionalmente adoptado.

Para conseguir una verdadera revolución, en cuanto a educación se refiere, el uso de nueva tecnología en el aula de clase debe verse apoyada por un cambio en la metodología de enseñanza. No tiene sentido brindar a los estudiantes nuevas y mejores formas de acceder a la información si únicamente se pretende que sean capaces de aceptar y reproducir los conocimientos que son transmitidos por su profesor. Por lo tanto, se vuelve indispensable orientar la educación hacia un aprendizaje centrado en el estudiante, donde sean los propios alumnos, los responsables de construir su conocimiento.

Es así como, la idea de un enfoque constructivista en la educación, viene siendo una opción bastante aceptada como metodología para sustentar un ambiente de enseñanza basado en el uso de herramientas tecnológicas. Para el constructivismo, el estudiante es el protagonista del proceso de aprendizaje a quien hay que brindar las herramientas necesarias que le permitan generar su propio conocimiento. Por esta razón, en el desarrollo del presente capítulo, se realiza una revisión de los conceptos básicos del constructivismo orientado a la educación, así como también de los objetos de aprendizaje basados en herramientas tecnológicas, que permitan cubrir las características básicas de un entorno de aprendizaje constructivista.

Objetivos del capítulo

Los objetivos que se pretenden alcanzar son:

- Identificar las principales características del constructivismo orientado a la educación, mediante revisión de literatura, para definir aquellas que son aplicables en el presente proyecto.
- Definir los objetos de aprendizaje y sus características aplicables de la metodología constructivista.



UNIVERSIDAD DE CUENCA

3.2 METODOLOGÍA DE APRENDIZAJE CONSTRUCTIVISTA

La forma de percibir el conocimiento y el proceso que se encuentra detrás de la formación del mismo es la base de la práctica educativa. Para los constructivistas, los estudiantes construyen activamente el conocimiento en sus intentos de entender el mundo que los rodea, por lo que consideran que el aprendizaje debe enfatizar el desarrollo de la comprensión y no la memorización (Elizabeth Murphy, 1997). La escuela constructivista apunta a una educación donde los estudiantes “aprenden a aprender”, en un entorno centrado en el alumno, con énfasis en el aprendizaje a través del descubrimiento, la exploración y las experiencias en el desarrollo de estrategias para resolver problemas (Aharon Aviram, 2000).

Sin embargo, pese a compartir la misma filosofía, han surgido varios tipos de constructivismo a lo largo de la historia. Ernest (1995) señala que: “existen tantas variedades de constructivismo como investigadores que lo definan”; cada uno con características propias que difieren únicamente en su forma de ver cómo se construye el conocimiento por parte de los aprendices. Finalmente, y a pesar de esta variedad de corrientes constructivistas, en lo referente a educación se distinguen dos filosofías claramente marcadas: el constructivismo Endógeno y el constructivismo Exógeno.

3.2.1 Constructivismo endógeno

Este enfoque constructivista sostiene que el proceso de construcción del conocimiento es de carácter individual. Para quienes defienden el constructivismo endógeno, este proceso se da en la mente de las personas como representaciones únicas del mundo; por tanto, consideran que el aprendizaje es un proceso interno que consiste en relacionar la nueva información con las representaciones preexistentes en la mente del individuo (José Serrano y María Pons, 2011).

El objetivo del aprendizaje para el constructivismo endógeno, es llegar a comprender la información recibida. Para este enfoque constructivista, comprender significa efectuar una interpretación personal de la realidad, y es aquí donde entran en juego los conocimientos y experiencias anteriores del individuo. El profesor debe actuar más como guía o facilitador de la tarea de aprendizaje que como director o administrador de la enseñanza. Además, el material de aprendizaje debe tener una clara organización interna u organización lógica, donde cada elemento de la información debe tener una conexión lógica o conceptual con el resto de los elementos. En definitiva, la situación de enseñanza-aprendizaje debe propiciar que el alumno relacione, de una forma sustantiva y no arbitraria, el material de aprendizaje con su propia estructura cognitiva (UNIREC, 2001).

Los cuatro principios sobre los cuales se asienta este tipo de constructivismo son (Von Glasersfeld, 1995):



UNIVERSIDAD DE CUENCA

1. “El conocimiento no se recibe pasivamente, ni a través de los sentidos, ni por medio de la comunicación, sino que es construido activamente por el sujeto cognoscente”.
2. “La función del conocimiento es adaptativa, en el sentido biológico del término, tendiente hacia el ajuste o la viabilidad”.
3. “La cognición sirve a la organización del mundo experiencial del sujeto, no al descubrimiento de una realidad ontológica objetiva”.
4. Existe una exigencia de “socialidad”, en términos de “una construcción conceptual de los otros” y, en este sentido, las otras subjetividades se construyen a partir del campo experiencial del individuo. Según esta tesis la primera interacción debe ser con la experiencia individual.

3.2.2 Constructivismo Exógeno

Este tipo de constructivismo enfatiza el medio externo como la fuente del aprendizaje. En este tipo de enfoque, el sujeto participa activamente en la interpretación del medio (Edgar Salgado, 2006) y reconstruye su propio conocimiento a través de influencias externas como experiencias personales de terceros y modelos a los que se expone en su aprendizaje diario. El alumno procesa la información recibida para organizarla y reorganizarla, de manera que refleje de forma fiel un determinado contenido.

La idea es mejorar el tratamiento y la reproducción de la información sin alterar su significado original. El papel del profesor se centra en transmitir información, pero guiando a los alumnos a un conocimiento más preciso y completo, enseñando las estrategias adecuadas y corrigiendo los conceptos erróneos. Esta forma de enseñar es ampliamente utilizada por la mayoría del profesorado, prueba de ello es que gran parte de los alumnos aprenden a extraer información relevante de un determinado contenido, siendo capaces de organizar el estudio de un tema o dar una respuesta estructurada a una "pregunta larga" de examen, tratando de realizar en todos los casos una traslación adaptada a las exigencias del momento (Grupo "Marmenor", 2011).

Con el aprendizaje Exógeno no se producen necesariamente cambios conceptuales, ni se capacita para la aplicación del conocimiento o la resolución de problemas, sino que supone un refinamiento en cuanto a la organización de la información. Por ejemplo, un alumno que ha sido entrenado en la realización de mapas conceptuales o esquemas, podrá usar estos recursos para estructurar un tema determinado o para realizar un examen organizado y coherente (Grupo "Marmenor", 2011).

3.2.3 La tendencia constructivista en la educación

Entre la disociación entre lo individual y lo social existen, en el momento actual, un conjunto de propuestas que buscan la integración de estos dos enfoques



UNIVERSIDAD DE CUENCA

constructivistas (José Serrano y María Pons, 2011). La finalidad es demostrar que “si incorporamos las perspectivas socio-cultural y lingüística al modelo cognitivo de los procesos mentales, es posible vislumbrar cómo el lenguaje y los procesos sociales del aula, constituyen las vías a través de las cuales los alumnos adquieren y retienen el conocimiento” (Nuthall, 1997). En definitiva existe un enfoque constructivista emergente que sería el resultado de la coordinación explícita de dos perspectivas teóricas: una perspectiva social consistente en una visión interaccionista de los procesos colectivos y compartidos que tienen lugar en el aula y una perspectiva psicológica, consistente en una visión constructivista cognitiva de la actividad individual de los alumnos mientras participan en esos procesos compartidos (Coob y Yakel, 1996).

3.2.4 Características de un entorno de aprendizaje constructivista

Varios autores definen diversas características que debe cumplir un entorno de aprendizaje para ser considerado como constructivista. Sin embargo, entre las características que son comunes a todos ellos podemos citar las siguientes:

3.2.4.1 Múltiples perspectivas

El constructivismo es una teoría que propone que el ambiente de aprendizaje debe sostener múltiples perspectivas o interpretaciones de la realidad (Jonassen, 1991). Esta característica constructivista se consigue ofreciendo a los estudiantes diferentes interpretaciones de un mismo problema para que de esta forma puedan percibir, de una mejor manera, su complejidad.

En un entorno de aprendizaje constructivista, el papel del profesor es organizar y presentarla información de múltiples formas: textual, gráfica, audiovisual, etc. Una vez que se ha presentado el problema de diferentes formas, debe obtener y valorar las diferentes soluciones que los estudiantes puedan dar para resolver la tarea. El objetivo es brindar al estudiante un nuevo espacio, en donde tiene a su disposición actividades innovadoras, con aspectos creativos, que pueden ser personales o de carácter colaborativo y que les permiten afianzar lo que aprenden al mismo tiempo que se divierten (Stefany Hernández, 2008).

3.2.4.2 Objetivos dirigidos por los estudiantes

Para el constructivismo los objetivos de aprendizaje que se persiguen son individuales, pero los conocimientos se ven enriquecidos con las aportaciones del grupo de trabajo. Esta característica constructivista requiere que los alumnos enfoquen sus objetivos adecuadamente, tomando en cuenta los recursos educativos que tienen a su alcance y el ambiente social que los rodea. Por lo tanto, los estudiantes dentro de un ambiente de aprendizaje constructivista, deben ser capaces de auto-dirigir su aprendizaje.

El aprendizaje autorregulado está enfocado siempre a una meta y controlado por el sujeto que aprende (Arguelles y Nagles 2007). Los alumnos que se



UNIVERSIDAD DE CUENCA

autorregulan manifiestan ser participantes activos en el proceso de aprendizaje (Pintrich y Schrauben, 1992). Sin embargo, para lograrlo necesitan ser constantes, estratégicos y mantenerse motivados hacia metas importantes (Blumenfeld y Marx, 1997; McCombs y Marzano, 1990).

Se considera que esta capacidad de autorregulación juega un papel clave en el éxito académico y en cualquier contexto vital (Nota, Soresi y Zimmeman, 2004). Por ello, es necesario que los estudiantes lleguen a la universidad con esas competencias y que el entorno les capacite para realizar un aprendizaje autónomo e independiente.

3.2.4.3 Profesores como guías

Con la metodología constructivista varían los roles tradicionales desempeñados por el profesor y el estudiante. Se pasa a nuevos entornos donde los conocimientos se desarrollan de forma conjunta y colaborativa. El alumno deja el papel de receptor total de información, mientras que el profesor pasa de emisor a ser el principal orientador, guía y facilitador de la enseñanza, donde su presencia es primordial en la selección de actividades, la organización, toma de decisiones y aplicación de métodos para alcanzar aprendizajes significativos. (Julio Cabero, 2005)

El objetivo de esta característica constructivista es que el alumno sea capaz de construir su conocimiento, con el profesor como un guía y con la libertad de explorar el ambiente tecnológico necesario para alcanzar sus objetivos de aprendizaje. Esto se puede lograr considerando los siguientes puntos:

- Especificar con claridad los propósitos del curso o lección.
- Tomar ciertas decisiones en la forma de ubicar a los alumnos en el grupo.
- Explicar con claridad a los estudiantes la tarea y la estructura de la solución buscada.
- Evaluar el nivel de logros de los alumnos y motivarlos a trabajar en grupo.

3.2.4.4 Metacognición

La metacognición se refiere al conocimiento, concientización, control y naturaleza de los procesos de aprendizaje. La importancia de este concepto para la educación radica en que todo alumno es un aprendiz que se halla constantemente ante nuevas tareas de aprendizaje; y debe ser capaz de aprender de forma autónoma y autorregulada. De esta manera el propio estudiante puede plantear, controlar y evaluar el desarrollo de las responsabilidades y tareas que debe cumplir (Sonia Osses, Sandra Jaramillo, 2008).

Para desarrollar estrategias metacognitivas en el aula se debe:



UNIVERSIDAD DE CUENCA

- Decidir cuál es la naturaleza del problema que hay que solucionar, es decir, planificar las actividades antes de enfrentarlas.
- Formar una representación mental que guíe la ejecución de las estrategias.
- Observar los procesos de la solución.

Dichas estrategias se las puede llevar a cabo utilizando distintos métodos. Entre los más usados encontramos: la identificación de las ideas principales, técnicas de subrayado, creación de resúmenes, redacción escrita, comprensión, atención, memoria, apuntes, razonamientos, solución de problemas, cuestionamientos, representaciones, etc. (Rafael Flores, 1994).

3.2.4.5 Control del alumno

En el constructivismo el aprendizaje es activo, no pasivo. Las personas aprenden cuando pueden controlar su aprendizaje y están al corriente del control que poseen, es decir construyen conocimientos por sí mismos a medida que van aprendiendo. (Julio Cabero, 2005). Una de las características de los entornos de enseñanza constructivista, es que ofrecen la posibilidad de individualizar los procedimientos y contenidos de aprendizaje de acuerdo a las necesidades e intereses del alumno, aumentando su grado de autonomía y su control respecto al ritmo y secuencias de aprendizaje.

Para que el estudiante controle su aprendizaje, es importante aumentar su interés y motivación. Esto se puede lograr a través de la hipermedia, la cual permite presentar el contenido de diferentes formas.

Entre los elementos más destacados de la hipermedia, se menciona (Conklin, 1987):

- Texto
- Gráficos
- Voz digitalizada
- Grabaciones de audio
- Imágenes
- Animaciones, vídeos
- Incluso gustos, olores y sensaciones táctiles

Con estos elementos el estudiante tiene el control de su aprendizaje, haciendo uso de la información más conveniente y necesaria para alcanzar sus objetivos.

3.2.4.6 Actividades y contextos reales

Para construir un aprendizaje de mejor calidad, se debe trascender más allá de las aulas y permitir a los alumnos resolver problemas en situaciones cotidianas. Un componente importante del constructivismo es que la educación se enfoca



UNIVERSIDAD DE CUENCA

en tareas auténticas, las cuales tienen una relevancia y utilidad en el mundo real. La enseñanza debe partir de experiencias y situaciones reales que permitan tanto su transferencia posterior como la percepción de la complejidad del mundo real.

El conocimiento se construye a medida que el aprendiz va descubriendo el sentido de sus experiencias. Es importante que el entorno de enseñanza provea de contextos reales y auténticos que fomenten la colaboración, otorgando al alumno un alto grado de control en el proceso de aprendizaje y le permita descubrir el sentido a sus logros alcanzados.

Es el propio alumno quien habrá de lograr la transferencia de lo teórico hacia lo práctico situando su aprendizaje en contextos reales. El estudiante puede ayudarse en actividades que le permitan construir significados a partir de la información que recibe, como por ejemplo: organizadores gráficos, mapas, esquemas, simulaciones, etc. Estas actividades deben reflejar situaciones reales productivas para el estudiante y que, posteriormente, le permitan enfrentar el mundo real. (Elisa Navarro, Alexandre Texeira).

3.2.4.7 Construcción del conocimiento

El enfoque constructivista destaca la construcción de nuevo conocimiento y maneras de pensar mediante la exploración y la manipulación activa de objetos e ideas, tanto abstractas como concretas (A. Cañas, K. Ford, P. Hayes, T. Reichherzer, N. Suri, J. Coffey, R. Carff, G. Hill, 1997). El concepto actual de construcción del conocimiento enfatiza a un estudiante centrado en actividades colaborativas que apuntan a la producción de nuevo conocimiento (Sergio Zúñiga Sánchez, 2004). Y es justamente, en la forma de ejecutar dichas actividades, donde puede reformarse la enseñanza mediante el uso de herramientas que involucren Tecnologías de Información y Comunicación (TIC) como por ejemplo computadoras en red y dispositivos móviles.

Para el constructivismo, es el sujeto quién construye el conocimiento a partir de su interacción con el entorno físico y social. La realidad se construye activamente mediante la aplicación y el reajuste de los esquemas cognitivos del sujeto al medio. Sin embargo, el resultado de esta construcción, dependerá del grado de desarrollo de las capacidades cognitivas propias del estudiante (Luz Zuluaga, Olga Serna, Ana Botero, Teresa Urrea, 2009).

En un entorno de aprendizaje constructivista, la primera actividad que permite la construcción del conocimiento es un estudio profundo e individual del tema. A partir de este estudio se debe realizar, entre todos los estudiantes, foros de conversación para el intercambio de ideas y la resolución de dudas que hayan surgido (Álvarez y Guasch, 2006). El docente debe intervenir como mediador del cambio conceptual de sus alumnos, ya que conocida sus ideas previas, su tarea consiste en plantear interrogantes o problemas, a partir de esas preconcepciones, de manera de incitarlos a buscar y a construir su propio aprendizaje.



UNIVERSIDAD DE CUENCA

3.2.4.8 Aprendizaje colaborativo

Se denomina aprendizaje colaborativo al intercambio y desarrollo del conocimiento en el seno de grupos de iguales encaminados a la consecución de objetivos académicos (Quintina Martín-Moreno Cerrillo, 2004). Este tipo de aprendizaje sostiene que los alumnos aprenden mediante un proceso en el cual se proponen y comparten ideas para resolver una tarea, favoreciéndose con el diálogo la reflexión sobre las propuestas propias y las de los compañeros (B. Barros y M.F. Verdejo, 2001).

Los métodos de aprendizaje colaborativo comparten la idea de que los estudiantes trabajan juntos para aprender y son responsables del aprendizaje tanto de sus compañeros como del suyo propio (César Collazos, Luis Guerrero y Adriana Vergara, 2001). En general, es necesaria la formación de pequeños grupos de trabajo quienes, además de desarrollar habilidades sociales y de trabajo en equipo, deben cumplir con actividades académicas asociadas a la solución de problemas. Entre estas actividades se encuentran: ejecutar un análisis del problema, comprobar el nivel de comprensión del mismo, construir diagramas de flujo y organizadores gráficos, hacer estimaciones, explicar materiales escritos, formular y generar preguntas, hacer listados y predicciones, presentar información, hacer razonamientos, consignar referencias a materiales revisados con anterioridad, resolver cuestionamientos, resumir y pensar creativamente (EcuRed, 2011).

El papel del profesor, durante una sesión de trabajo en equipo, es explicar claramente la actividad, los criterios a evaluar y determinar responsabilidades en el grupo. Durante la sesión de clase debe moverse de equipo a equipo, observando las interacciones, escuchando conversaciones e interviniendo cuando sea apropiado. El profesor estará continuamente observando los equipos y haciendo sugerencias acerca de cómo proceder o dónde encontrar información. (Dirección de Investigación y Desarrollo Educativo Instituto Tecnológico y de Estudios Superiores de Monterrey, 2005).

3.2.4.9 Construcciones previas del conocimiento

Los alumnos poseen una estructura cognitiva previa, elaborada a partir de su experiencia diaria, que le sirve para explicar y predecir lo que ocurre en su entorno y en su día a día (Mauricio Compiani, 1998). El interés de la concepción constructivista, por los esquemas en los que se hallan organizados los conocimientos previos de los alumnos, es indagar su incidencia directa en los procesos de enseñanza y aprendizaje que se llevan a cabo en el aula (Coll, Martín, Mauri, Miras, 1994).

En un entorno de aprendizaje constructivista, organizar la enseñanza desde los conocimientos que ya poseen los alumnos es importante. Esto se debe a que, frente a una nueva información o a un nuevo material, los alumnos ponen en juego conocimientos anteriores, a partir de los cuales, interpretan los nuevos contenidos (José López Recacha, 2009). Esta característica constructivista, en



UNIVERSIDAD DE CUENCA

definitiva, pretende evitar que el alumno se convierta en un ente memorista y que, por el contrario, sea capaz de elaborar su propio conocimiento a partir de la deducción y el razonamiento lógico.

Para cumplir con este objetivo el alumno dispone de un conjunto de instrumentos, estrategias y habilidades generales que ha ido adquiriendo en distintos contextos a lo largo de su desarrollo. Instrumentos como el lenguaje (oral y escrito), la representación gráfica y numérica; habilidades como subrayar, tomar apuntes o resumir; y estrategias generales para buscar y organizar información, para repasar, o para escribir reflexivamente sobre un tema, son algunos ejemplos de este conjunto de recursos de tipo general que pueden formar parte, en una u otra medida, del repertorio inicial del alumno y con los que cuenta (o no) para afrontar el aprendizaje de nuevo contenido (José López Recacha, 2009).

3.2.4.10 Resolución de problemas

La resolución de problemas es un proceso mental que consiste en descubrir, analizar y resolver problemas utilizando diversas estrategias. Muchos libros de texto dedican una fracción significativa de su espacio al planteamiento de problemas y ejercicios. De hecho existen manuales especializados e incluso colecciones y series editoriales dedicadas íntegramente a la resolución de problemas en diversas áreas (Iván Sánchez Soto, 2008).

Resolver problemas y ejercicios constituye una actividad de aprendizaje importante en la propuesta constructivista. En un entorno de aprendizaje basado en esta metodología, se debe partir de los problemas, los ejemplos o de los proyectos o problemas y, mediante ellos, se llega a la información y a elaborar los conceptos adecuados (Jonass en, 2000). Para lograr este objetivo, primero se presenta un problema, se identifican las ideas previas, los contenidos a trabajar en el curso, las necesidades de aprendizaje, se busca la información necesaria y finalmente se regresa al problema, por medio de la resolución de una secuencia de problemas más pequeños que lleva a la solución del gran problema (Sánchez, 2007).

La innovación en esta propuesta implica cambiar el papel de alumno y el docente. El primero se vuelve más activo y responsable de su propio aprendizaje, resolviendo los problemas. El docente crea instancias de aprendizaje y de interacción en el aula. El objetivo es que el alumno construya conocimiento a través de la búsqueda de información, investigación, solución de actividades de aprendizaje y la solución de ejercicios (Iván Sánchez Soto, 2008).

Con el fin de resolver correctamente un problema, es importante seguir una serie de pasos (Kendra Cherry, 2012):

- Identificar el problema
- Definir del problema



UNIVERSIDAD DE CUENCA

- Desarrollar una estrategia para resolver el problema: el enfoque utilizado variará dependiendo de la situación y preferencias únicas del individuo.
- Organizar la información: antes de llegar a una solución, se debe organizar primero la información disponible. Cuanta más información está disponible, mejor preparados estarán los alumnos para llegar a una solución precisa.
- Monitorear el progreso
- Evaluar los resultados: es importante para evaluar los resultados determinar cuál es la mejor solución para el problema.

3.2.4.11 Consideración de errores

Los errores de los estudiantes han sido empleados tradicionalmente por los investigadores y profesores como una herramienta valiosa para el diagnóstico de dificultades de aprendizaje, para estudiar el proceso de aprendizaje, para planificar programas de estudios y diseñar material didáctico correspondiente. Para el constructivismo, sin embargo, los errores de los alumnos tienen una importancia que va más allá de la simple valoración; y tienen un significado y sentido propio como parte misma del aprendizaje significativo.

Para el constructivismo, los errores son parte del proceso de adaptación que define el aprendizaje. Cuando se cometen errores, los constructivistas ven en ellos oportunidades potenciales y ricas para desarrollar el conocimiento y el aprendizaje. Los errores de los alumnos no son vistos como fracasos, no se ven peyorativamente, sino se perciben como parte natural del proceso de aprendizaje y una oportunidad para que el alumno aprenda a partir de ellos (Jérôme Proulx, 2006).

En un entorno de aprendizaje constructivista, los errores tienen el potencial para ayudar a evitar y predecir posibles “callejones sin salida” en el futuro. Los alumnos pueden beneficiarse de ellos y entender mejor los conceptos trabajados en el aula, aclarando sus dudas cuando han incurrido en un error. La tarea del profesor, en este sentido, es mostrarle al alumno las diferencias entre un error y una respuesta adecuada en contextos específicos, haciendo uso de herramientas y recursos, reflexionando sobre la situación y realizando la exploración del problema (Borasi Raffaella, 1998).

3.2.4.12 Exploración

El método de exploración consiste en la presentación de un problema con una o varias soluciones. Los estudiantes tratan las formas sugeridas por el maestro y anotan los resultados para llegar a conclusiones. Las tareas del profesor en este aspecto son investigar y diagnosticar tanto destrezas como actitudes. Las actividades del estudiante son: procesar información, reconocer su capacidad para pensar, utilizar conocimientos previos, utilizar la metacognición, reconocer sus fortalezas, superar sus limitaciones y fortalecer su autoestima (Nilda Santiago Lebrón, 2004).



UNIVERSIDAD DE CUENCA

El conjunto de técnicas empleadas en la exploración de las ideas y concepciones de los alumnos abarca una amplia gama, desde el uso de cuestionarios hasta la observación directa de sus actividades en el aula. Los cuestionarios son, en muchas ocasiones, los instrumentos más utilizados por los profesores para explorar el conocimiento de sus alumnos sobre un tema. En la evaluación, el docente puede usar la exploración a través de preguntas formuladas por el profesor durante la clase: se suele utilizar para estimular el nivel de comprensión de los alumnos sobre algo que se está revisando y con base en ello proporcionar de manera oportuna algún tipo de ayuda requerida (María González, 2006).

3.2.4.13 Aprendizaje cognitivo

El término aprendizaje cognitivo proviene del trabajo de Collins, Brown, y Newman (1989). Se refiere al proceso mediante el cual un maestro que domina una habilidad en particular enseña dicha destreza a un aprendiz. Aplicando el aprendizaje cognitivo, el profesor muestra a los estudiantes cómo se desarrolla un proceso o una actividad, y da las razones por las cuales sucede de esa manera (Collins, 1991). Esta característica apoya la integración de la educación académica y vocacional, para que los estudiantes construyan su propia comprensión de los estándares académicos e internalicen los procesos de pensamiento utilizados para hacerlo (Aziz Ghefaili, 2003).

El aprendizaje cognitivo trata de involucrar a los estudiantes en escenarios del mundo real, en los que actúan e interactúan, para lograr resultados útiles y/o aplicables (Brill, Kim, y Galloway, 2001). La tecnología provee diversas alternativas útiles que ayudan a establecer ambientes de aprendizaje cognitivo efectivos. Las aplicaciones hipermedia y la realidad virtual permiten emular problemas auténticos del mundo real en el aula de clases. El uso de videos, sistemas expertos, demostraciones y simulaciones, permiten a los estudiantes observar el trabajo y desempeño de científicos y expertos en el tema que es objeto de su estudio. Las animaciones e imágenes permiten visualizar procesos que no son observables a simple vista (Ejemplo: fotosíntesis) (Aziz Ghefaili, 2003). De esta manera, la aplicación de la tecnología en entornos de aprendizaje cognitivo permite diseñar e implementar modelos de enseñanza interactivos, flexibles y basados en los principios del constructivismo, que mejoren la capacidad de los estudiantes de formar sus propias representaciones del conocimiento.

3.2.4.14 Interrelación de conceptos

Los conceptos son construcciones o imágenes mentales, por medio de las cuales comprendemos las experiencias que emergen de la interacción con nuestro entorno. En un ambiente de aprendizaje constructivista, relacionar los nuevos conceptos con aquellos que fueron aprendidos anteriormente, resulta ser una práctica fundamental que facilita la internalización de nuevo conocimiento a partir del conocimiento que actualmente posee el estudiante.



UNIVERSIDAD DE CUENCA

Existen varias herramientas que facilitan a los profesores interrelacionar conceptos en un entorno constructivista. Entre las más comunes podemos citar (Federico Maglio, 1999):

- **Mapas semánticos:** Son gráficos que ayudan a los alumnos a ver cómo las palabras se relacionan entre sí. Se construye a partir de un concepto central y se vuelcan otros que tienen con él una cierta relación. Se parte de un concepto y los alumnos lo van completando con sus conocimientos previos fomentando el debate, la discusión, el intercambio de ideas, el aprendizaje de nuevas palabras, etc. A partir de allí se pueden construir las redes y mapas conceptuales.
- **Red conceptual:** Es un esquema que establece la relación entre los conceptos o palabras claves (generalmente sustantivos). A partir de los mapas semánticos, los alumnos deben realizar los dos pasos siguientes: **a)** establecer las posibles relaciones entre las palabras (conceptos) y, **b)** ordenar los conceptos en una red definitiva.
- **Mapa conceptual:** Es una jerarquización de conceptos desde los más generales a los más específicos. Un concepto es una idea, una imagen acerca de lo que se habla. De cada uno surgen las derivaciones hacia otros conceptos menos abarcadores.

3.2.4.15 Diferentes puntos de vista

En la metodología constructivista, el aprendizaje enfatiza el proceso y no el producto. Lo que resulta realmente importante es cómo el estudiante llega a una respuesta determinada y no a la recuperación de una “solución objetivamente verdadera” (Elizabeth Murphy, 1997). La construcción propia del conocimiento por parte del estudiante, en lugar de la aceptación de una única verdad que debe ser aceptada y reproducida, es consistente con la tendencia del constructivismo de privilegiar las múltiples representaciones, perspectivas y realidades del conocimiento.

El papel del maestro es organizar la información en torno a grupos conceptuales de problemas, preguntas y situaciones discrepantes, que permitan observar de diferentes formas un mismo problema. Esto lo realizan con el fin de captar el interés del estudiante, ayudándolo a desarrollar nuevos conocimientos y conectarlos con su aprendizaje previo. Las actividades están centradas en el alumno y a los estudiantes se les anima a hacer sus propias preguntas, llevar a cabo sus propios experimentos, hacer sus propias analogías y llegar a sus propias conclusiones (Susan Hanley, 1994).

3.2.4.16 Andamiaje cognitivo (“Scaffolding”)

El término andamiaje cognitivo (“scaffolding”) proviene de los trabajos de Wood, Bruner y Ross (1976). Esta característica del constructivismo tiene que ver con la forma en la que los profesores interactúan con los estudiantes



UNIVERSIDAD DE CUENCA

mientras completan una determinada actividad. En un entorno de aprendizaje constructivista, es de gran importancia, que el alumno sea capaz de completar las tareas tanto como le sea posible sin asistencia del profesor. El maestro sólo trata de ayudar al estudiante con las tareas que están más allá de su capacidad actual. Los errores por parte de los estudiantes son comunes, pero con la retroalimentación y sugerencias del profesor, el estudiante logra completar satisfactoriamente la tarea o meta asignada. Una vez que el alumno domina la tarea, el profesor inicia la supresión gradual de los “andamios”, es decir, permite al estudiante trabajar de manera independiente (Lipscomb, Swanson, y West, 2004).

El andamiaje cognitivo busca la construcción de nuevo conocimiento a partir del conocimiento actual de los estudiantes (Benson, 1997). Existen diferentes herramientas que pueden ser utilizadas por los profesores con el fin de cumplir este objetivo. Entre las más comunes se encuentran: dividir la tarea en partes más pequeñas y manejables, narrar los procesos de pensamiento mientras se completa una tarea y el aprendizaje cooperativo, que promueve el trabajo en equipo y el diálogo entre los estudiantes. Otras técnicas podrían incluir la activación de conocimientos previos, dando consejos, estrategias, pautas y procedimientos que ayuden al estudiante a resolver un nuevo problema utilizando experiencias que haya tenido con problemas anteriores.

El entorno de aprendizaje debe mantener al alumno motivado en la búsqueda de nuevo conocimiento manteniendo un nivel de dificultad adecuado. Habilidades o tareas demasiado lejos del alcance de los estudiantes pueden llevarlos a su nivel de frustración, mientras que tareas demasiado simples pueden causar el mismo efecto (Lipscomb, Swanson, y West, 2004).

3.2.4.17 Evaluación auténtica

En un entorno de aprendizaje constructivista, la evaluación no es vista como una actividad final sino más bien un proceso continuo que ayuda al alumno a seguir aprendiendo (Holt y Willard-Holt, 2000). Los constructivistas creen que la evaluación debe ser utilizada como una herramienta para mejorar tanto el aprendizaje del estudiante como también la comprensión del profesor sobre el entendimiento actual del estudiante. No se debe utilizar como una herramienta de rendición de cuentas que pueda llegar a desmotivar a los alumnos y que los pueda llevar a renunciar (North Central Regional Educational Laboratory, 2011).

Brooks y Brooks (1993) describen cómo debe ser la evaluación en un entorno de aprendizaje constructivista. En lugar de reprender al estudiante cuando no da con la respuesta exacta que se busca, el maestro intenta descifrar el entendimiento actual del estudiante sobre el tema tratando de reforzar aquellos conceptos que no se encuentran claros (North Central Regional Educational Laboratory, 2011).



UNIVERSIDAD DE CUENCA

Una herramienta apropiada para la evaluación en un entorno constructivista es el uso de cuestionarios de auto evaluación al final del contenido. El objetivo de estos cuestionarios es valorar qué tanto ha comprendido el estudiante sobre el tema tratado, e identificar los puntos en los cuales requiere poner mayor atención. De esta forma el profesor puede tasar el grado de comprensión de cada estudiante y puede dar sugerencias o explicar más detalladamente los conceptos en los cuales se tenga mayor dificultad.

3.2.4.18 Fuentes primarias de información

Se denominan fuentes de información a diversos tipos de documentos que contienen datos útiles para satisfacer una demanda de información o conocimiento (Universidad de la Salle, 2002). Como fuentes de información primarias se hace referencia a aquellas que contienen información original, que no ha sido filtrada, interpretada o evaluada por nadie más. Son producto de una investigación o de una actividad eminentemente creativa. (María Silvestrini y Jacqueline Vargas, 2008).

En un entorno de aprendizaje constructivista, el profesor se convierte en la primera fuente de información pero no en la única. El constructivismo enfatiza la necesidad de proveer a los estudiantes el acceso a diversas fuentes primarias de información como por ejemplo: libros, revistas científicas, monografías, informes de investigación, etc. El objetivo es incentivar la indagación en los estudiantes para que sean ellos quienes generen sus propias fuentes de información a través de resúmenes, fichas nemotécnicas, etc.

3.3 OBJETOS DE APRENDIZAJE

Los objetos de aprendizaje son elementos de un nuevo tipo de instrucción fundamentado en un paradigma computacional (Wiley, 2000). Los objetos de aprendizaje pueden ser utilizados y re-utilizados en el aprendizaje con el apoyo de la tecnología que incluyen: sistemas de entrenamiento, ambientes de enseñanza interactivos, sistemas inteligentes, etc. Esta tendencia se ha convertido en la más importante a nivel mundial respecto a la producción de contenidos educativos, tanto como apoyo adicional a la educación dentro del aula, como en materia prima esencial para la educación basada en tecnologías de la información y comunicaciones (Virtual Educa Brasil, 2007).

3.3.1 Características y Funcionalidad

Como se mencionó inicialmente, una de las principales características de los objetos de aprendizaje es su reutilización. Sin embargo esta no es la única, estos objetos cuentan con la posibilidad de ser actualizados, combinados, separados, referenciados y sistematizados (Lorenzo García, 2005).

Todo objeto de aprendizaje está compuesto de dos partes, por un lado el contenido del objeto y por otro lado la etiqueta que lo describe, como por ejemplo: palabras claves relativas al contenido, objetivos, nivel, prerrequisitos, evaluación, autor, fecha, lenguaje, versión, etc. Al crear un objeto de



UNIVERSIDAD DE CUENCA

aprendizaje, se debe tener en cuenta su doble funcionalidad, que consiste por un lado en ser una parte a integrar en una unidad mayor es decir en uno o más cursos, y por otro lado de ser un elemento independiente con sus propias características (GITA, 2012).

3.3.2 Clasificación

Los objetos de aprendizaje se pueden clasificar en función de: su combinación y su uso pedagógico

De acuerdo a la combinación de objetos, se definen cinco tipos de objetos de aprendizaje:

- **Fundamentales.**-Recurso digital individual, que no está combinado con ningún otro. Por ejemplo, un documento de texto sobre un sólo tópico.
- **Combinados-cerrados.**-Un conjunto de recursos digitales combinados por el autor en el momento del diseño del objeto de aprendizaje. Estos recursos digitales no pueden ser accedidos individualmente, sino en conjunto. Por ejemplo, un video clip que combina audio y video, no se puede acceder de forma independiente el audio o el video.
- **Combinados-Abiertos.**-Un conjunto de recursos digitales combinados por una computadora en respuesta a una solicitud. Estos también pueden ser reusados de forma individual. Por ejemplo, una página Web generada dinámicamente que combina audio, texto, video.
- **Generación de Presentaciones.**-Un objeto de aprendizaje de presentación generativa tiene una alta reusabilidad intra-contextual (Pueden ser usados una y otra vez en contextos similares), pero tienen una baja reusabilidad extra-contextual (no pueden ser usados en dominios para los cuales no fueron generados).
- **Generación Instruccional.**- Es concebido para evaluar la capacidad de un aprendiz y de recordar una serie de pasos.

De acuerdo al uso pedagógico, se definen cuatro tipos de objetos de aprendizaje (Reynel Agudelo Mondragon y Nestor Almid García, 2010):

- **Objetos de Instrucción.**- Su función es dar apoyo al aprendizaje y se divide en seis tipos distintos:
 - ❖ **Objetos de Lección.**- Combinan textos, imágenes, videos, animación, preguntas y ejercicios para crear aprendizaje interactivo.
 - ❖ **Objetos Workshop.**- Los Workshop son eventos de aprendizaje en los cuales un experto interactúa con los aprendices.



UNIVERSIDAD DE CUENCA

- ❖ **Objetos Seminario.-** Los seminarios son eventos en los cuales expertos hablan directamente a los aprendices usando una combinación de audio, video, presentaciones en diapositivas e intercambio de mensajes.
- ❖ **Objetos Artículos.-** Corresponden a objetos basados en breves textos que pueden corresponder a material de estudio con gráficos, tablas, etc.
- ❖ **Objetos White Papers.-** Son objetos basados en textos, pero con información detallada sobre tópicos complejos.
- ❖ **Objetos Casos de Estudio.-** Son objetos basados en textos, correspondiente a análisis en profundidad de una implementación de un producto de software, experiencias pedagógicas, etc.
- **Objetos de Colaboración.-** Son objetos que se desarrollan para la comunicación en ambientes de aprendizaje colaborativo. Algunos ejemplos son: chats, foros, reuniones on-line.
- **Objetos de Práctica.-** Son objetos destinados principalmente al autoaprendizaje, con una alta interacción del aprendiz.
 - ❖ **Simulación Conceptual.-** Este tipo de objetos (también conocido como de ejercicios interactivos) ayudan a los aprendices a relacionar conceptos a través de ejercicios prácticos.
- **Objetos de Evaluación.-** Son los objetos que tienen como función indicar el nivel de conocimiento que tiene un aprendiz.
 - ❖ **Pre-evaluación.-** Son objetos destinados a medir el nivel de conocimiento que tiene un aprendiz antes de comenzar el proceso de aprendizaje.
 - ❖ **Evaluación de Proficiencia.-** Estos objetos sirven para medir si un aprendiz ha asimilado determinado contenido que permita deducir una habilidad. Por ejemplo, si un aprendiz obtiene una determinada puntuación en un test, se puede considerar que ha cumplido los objetivos en el camino del aprendizaje y está listo para realizar una determinada tarea o asumir un determinado rol.

3.3.3 Especificaciones para su desarrollo

A continuación se destacan algunas especificaciones que todo desarrollador debe tener en cuenta durante el proceso de generación del objeto de aprendizaje (GITA, 2012):

- Adecuados formatos en la presentación de la información, es decir fácil de leer y comprender.
- Eliminación de las referencias externas al propio objeto, debe incluir toda la información necesaria para el usuario o aprendiz.
- Consistencia en el uso del lenguaje, meditada elección de la terminología evitando la utilización de sinónimos que den lugar a confusiones.



UNIVERSIDAD DE CUENCA

- Lenguaje apropiado para una gran audiencia huyendo de la excesiva especialización.
- Eliminación de grandes textos que dificulten la lectura en pantalla, por lo tanto la presentación de la información deberá estar preparada para el dispositivo en el que será mostrada.

3.3.4 Objetos de aprendizaje para la Web

La Web es considerada un gran repositorio de objetos de aprendizaje y de gran utilidad si dichos objetos se encuentran seleccionados e identificados correctamente. Se debe considerar los aspectos pedagógicos y didácticos que permitan que el estudiante, al momento de consultar estos objetos, adquiera un aprendizaje y logre capacitarse correctamente. Uno de los aspectos más importantes es el contenido de la materia, se debe cumplir con las características de auto-contenido y secuencialidad, para lo que se recomienda hacer uso de texto para la información teórica, presentación de videos que complementen la lectura, casos de estudio donde se ilustren como se desarrollan los espacios de realidad virtual y ejemplos adicionales. Con lo mencionado se puede armar un ejemplo secuencial de aprendizaje de la siguiente manera: Primero se consulta el contenido teórico, luego se puede ver un video que proporcione más información sobre el mismo tema pero en otro formato, seguidamente se puede incluir una simulación que represente la realidad de lo que se está enseñando y finalmente el alumno puede resolver sus propios ejercicios que le ayuden a evaluar su aprendizaje. De esta manera estratégica el alumno lograra sus objetivos y competencia (Pilar Gómez, Fernando Vázquez).

3.3.5 Objetos de aprendizaje en dispositivos móviles

La integración de objetos de aprendizaje con los dispositivos móviles, es actualmente una de las principales ayudas en los procesos de enseñanza-aprendizaje. En el ambiente virtual se busca que el alumno siga siendo el principal constructor de su conocimiento, y los dispositivos móviles un recurso adicional que le ofrecen flexibilidad en el tiempo, espacio y lugar. Esta necesidad de integrar objetos de aprendizaje en la tecnología móvil, parte como una necesidad de los propios estudiantes, que vieron la importancia de poder acceder y visualizar los contenidos de sus materias a través de un recurso que lo manejan todo el tiempo, como su teléfono celular.

Se indicó previamente los diferentes tipos de objetos de aprendizaje que se pueden aplicar en la educación, sin embargo no todos ellos pueden ser utilizados para un dispositivo móvil, lo mas recomendado para este tipo de tecnología son: la voz, los gráficos y las animaciones para las diferentes instrucciones, tomando en cuenta que el principal objetivo es promover el aprendizaje de campo. Cabe señalar, que el diseño de actividades no se encuentra solo en la forma de como se imparten o entregan las actividades, no



UNIVERSIDAD DE CUENCA

consiste únicamente en cambiar texto por voz, sino en lograr los objetivos del aprendizaje, se debe tener en cuenta el contenido que se va a transmitir y de acuerdo a ello la estrategia que se va a utilizar (Soledad Ramírez, 2007).

Como hemos visto, los Objetos de Aprendizaje son de gran importancia, ya que permiten potencializar la educación. Ofrecen a profesores y alumnos la posibilidad de tener contenidos educativos reutilizables, incorporando información de diversos recursos. En la actualidad el uso de videos, imágenes, audio, texto y aplicaciones, ayudan a la elaboración de planes de estudio más flexibles, didácticos y que se adapten a las necesidades de los estudiantes.

Con esta propuesta de integrar Objetos de Aprendizaje al prototipo planteado en el presente proyecto, se pretende motivar a los docentes en incursionar con en esta nueva temática educativa, permitiéndoles mejorar su proceso de enseñanza. De igual manera a los estudiantes, ya que podrán contar con un material interactivo, muy bien estructurado y con contenidos significativos que faciliten un mejor entendimiento y un aprendizaje de calidad.

3.4 RESUMEN DEL CAPITULO

Se han identificado y comprendido claramente las principales características tanto de los objetos de aprendizaje como las del modelo de enseñanza constructivista. Dichas características resultan aplicables dentro de la plataforma educativa que se pretende implementar con el desarrollo del presente proyecto.

El cubrir dichas características, a través de la generación de objetos de aprendizaje y la forma mediante la cual el estudiante tenga acceso a ellos, es la siguiente tarea a cubrir con el análisis y diseño del prototipo de enseñanza de la materia piloto que será desarrollado en el Capítulo 3.



UNIVERSIDAD DE CUENCA

CAPITULO 4:

“Análisis del prototipo de aprendizaje para la materia piloto Cálculo Diferencial.”



UNIVERSIDAD DE CUENCA

4.1 INTRODUCCIÓN

Uno de los aspectos más importantes para el desarrollo de cualquier sistema o prototipo es la creación del análisis y diseño del mismo. Definir de manera correcta las interfaces y funcionalidades de un sistema, así como los actores involucrados, garantiza que se construya un producto de calidad, que satisfaga las necesidades de los usuarios y aumente la probabilidad de éxito. Por tanto, es de vital importancia durante la fase de análisis el desarrollo de diferentes actividades que describa de manera clara y general las funcionalidades del sistema, identifique las características esenciales del mismo y los problemas que puedan presentarse en su desarrollo.

Lo que se pretende conseguir con la implementación de este prototipo de aprendizaje es que los alumnos accedan mediante sus dispositivos móviles o a través de la Web al contenido de la materia piloto *Cálculo Diferencial*, sus objetivos y el desarrollo de los capítulos de una manera intuitiva y colaborativa, que facilite el aprendizaje por parte del estudiante. Además se debe tomar en cuenta que, dentro del modelo de enseñanza constructivista, es el propio estudiante quien evalúa su conocimiento y con ello busca alcanzar sus objetivos, por lo tanto, la plataforma de aprendizaje en sí debe permitir al alumno un avance ordenado del contenido y de acuerdo a su nivel de entendimiento.

En el presente capítulo se realizará un análisis del prototipo de aprendizaje a ser implementado. Para lo cual primero se definirá el mismo, luego se realizará un proceso de clasificación e interpretación de los hechos, además de un diagnóstico del problema y la presentación de toda la información necesaria para el desarrollo y mejoras de dicho prototipo. Seguidamente, se determinará el conjunto de características y procesos necesarios para resolver el problema planteado y construir la solución adecuada. Todo esto con la finalidad de que en posteriores capítulos se puedan tomar decisiones de diseño adecuadas que tengan como respaldo diagramas y modelado de objetos y procesos que permitirán una visión más estructurada y organizada de los componentes del prototipo.

Objetivos del Capítulo

Los objetivos que se pretenden alcanzar son:

- Determinar las funcionalidades y los requerimientos del prototipo, mediante entrevistas y análisis con expertos en la materia piloto y participantes del proyecto.
- Realizar los diferentes diagramas de Casos de Uso con su respectiva especificación y lista de actores que permitan crear una idea general de la interacción entre el usuario y la aplicación.



UNIVERSIDAD DE CUENCA

4.2 DESCRIPCIÓN DEL PROTOTIPO

El prototipo de aprendizaje a implementar tiene como principal objetivo presentar los contenidos de la materia piloto *Cálculo Diferencial* a través de una plataforma tecnológica que permita el acceso del estudiante a través de la Web y/o un dispositivo móvil. El prototipo debe presentar estos contenidos de una manera atractiva e interesante para el alumno, con la capacidad de mantener un seguimiento continuo de su aprendizaje. Los estudiantes podrán visualizar el contenido de la asignatura, el mismo que estará compuesto pero no se limita a: los objetivos del curso, teoría, ejemplos y ejercicios para resolver. Cada uno de estos ítems serán presentados al estudiante a través recursos multimedia como: texto, imágenes, gráficos, videos, simulaciones, etc.

El prototipo deberá permitir el despliegue de las diferentes tareas que componen la materia piloto de una manera jerárquica y respetando el orden de los contenidos. Con ello se pretende asegurar que el alumno no se salte ninguna fase en su proceso de aprendizaje y se logren mejores resultados. Tomando en cuenta todos estos detalles del prototipo, a continuación se presenta el análisis y el diseño previos al desarrollo del mismo.

4.3 REQUISITOS FUNCIONALES

El prototipo a desarrollar debe cumplir los siguientes requisitos funcionales:

- La plataforma de enseñanza permitirá el acceso de los estudiantes a los contenidos de la materia piloto tanto en la Web como a través de un dispositivo móvil.
- El acceso a la plataforma desde el móvil deberá realizarse a través de una aplicación nativa Android.
- La plataforma deberá validar las credenciales del usuario para acceder a los contenidos, tanto en la Web como a través del móvil.

4.4 REQUISITOS NO FUNCIONALES

Además de las funcionalidades del prototipo se requieren otros aspectos igual de importantes para cumplir con las expectativas de los usuarios finales:

- Explotar la cooperación y la colaboración entre estudiantes como relaciones que fortalezcan la construcción del conocimiento.
- La actualización de contenidos se realizarán de acuerdo a lo establecido por las autoridades de la Facultad.
- Presentar los objetivos de la materia al igual que el cronograma de actividades.



UNIVERSIDAD DE CUENCA

- Mostrar la teoría referente a cada capítulo.
- Visualizar ejercicios resueltos como ejemplos.
- Resolver ejercicios como parte de la evaluación de cada capítulo.
- Evaluaciones a los estudiantes al final de cada tema o capítulo.

4.5 ANÁLISIS DEL PROTOTIPO

Este proceso pretende incluir todos los detalles relevantes de la aplicación y del uso de la misma, además de los casos de uso necesarios para identificar las acciones que llevarán a cabo los diferentes usuarios.

4.5.1 Glosario

El glosario recoge el vocabulario propio del dominio del sistema y que, dependiendo del proyecto, pueden ser términos muy especializados. Además puede usarse para definir un diccionario informal de tipos de datos. El núcleo de este producto es la definición de los diferentes conceptos en modo de diccionario, y normalmente por orden alfabético. Su propósito es hacer que los interesados en el proyecto conozcan la definición de términos y tipos de datos informales que se van a usar en el proyecto.

4.5.1.1 Definiciones

A continuación se presentan algunas definiciones de conceptos que aparecerán dentro del análisis y diseño del prototipo.

4.5.1.1.a Actor

Es un elemento que interactúa en uno o más procesos. Puede ser una persona, un objeto, una empresa, un sistema, etc.

4.5.1.1.b Caso de uso

Representa un conjunto de eventos que ocurren cuando un actor usa un sistema para completar un proceso.

4.5.1.1.c Inclusión

Un caso de uso puede incluir otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido.

4.5.1.1.d Extensión

Un caso de uso puede extender a otro para especificar diferentes variantes del mismo caso de uso. Implica que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias.



UNIVERSIDAD DE CUENCA

4.5.1.1.e Apache

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Se trata de un servidor Web HTTP de código abierto multiplataforma que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.

4.5.1.1.f eXe Learning

Es un programa de edición de sitios Web educativos de código abierto.

4.5.1.1.g Geogebra

Es un "procesador geométrico" y un "procesador algebraico" que permite la generación de graficas de funciones matemáticas.

4.5.1.1.h PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos.

4.5.1.1.i Servicio Web

Es un sistema de software diseñado para soportar interacción maquina a máquina sobre una red. En general los servicios Web son APIs Web que pueden ser accedidas en una red como internet y ejecutadas en un sistema de hosting remoto.

4.5.2 Estereotipos UML

A continuación en la

SIMBOLO	NOMBRE
	Actor
	Caso de uso
	Inclusión
	Extensión

Tabla 4.5-Ase listan los estereotipos usados dentro de la etapa de análisis del proyecto.



UNIVERSIDAD DE CUENCA

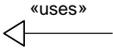
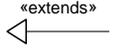
SIMBOLO	NOMBRE
	Actor
	Caso de uso
	Inclusión
	Extensión

Tabla 4.5-A Estereotipos UML

4.5.3 Lista de Riesgos

La identificación temprana de los riesgos que un proyecto enfrenta, permitirá que el equipo de trabajo pueda monitorearlo de forma adecuada y en caso de su ocurrencia, que se definan los mejores mecanismos de mitigación.

A continuación se detalla la lista de riesgos a presentarse en el desarrollo del prototipo

4.5.3.1 <Moodle>

Descripción	No se tiene un gran conocimiento sobre la plataforma Moodle
Impacto	Alto
Probabilidad de ocurrencia	Alta
Estado	identificado
Estrategia de mitigación	Revisar la documentación de Moodle en línea
Plan de contingencia	Obtener documentación de expertos que manejan Moodle y han desarrollado código sobre la plataforma.

4.5.3.2 <Android>

Descripción	No se cuenta con experiencia en el desarrollo de aplicaciones en Android
--------------------	--



UNIVERSIDAD DE CUENCA

Impacto	Alto
Probabilidad de ocurrencia	Alta
Estado	identificado
Estrategia de mitigación	Participar de capacitaciones en Android
Plan de contingencia	Se deberá capacitar al equipo técnico en desarrollo de aplicaciones Android

4.5.3.3 <Generación de contenidos para Moodle>

Descripción	Se tiene desconocimiento sobre herramientas de generación de contenido para Moodle
Impacto	Alto
Probabilidad de ocurrencia	Alta
Estado	identificado
Estrategia de mitigación	Capacitación con la Ing. Magaly Mejía experta en el tema
Plan de contingencia	Generación de contenidos manualmente

4.5.4 Especificación de Casos de Uso

A continuación se presenta la especificación de los casos de uso del sistema:

4.5.4.1 Caso de Uso para Ingresar a la Aplicación

- **Referencias**

- Resumen de casos de uso.
- Características del Prototipo

- **Lista de Actores**

- **Alumno**

Es el usuario que posee una cuenta para ingresar al sistema y tiene acceso a todas las funcionalidades existentes en la aplicación.

- **Moodle**

Repositorio en el que se almacenara toda la información necesaria a través del consumo de un servicio Web.



UNIVERSIDAD DE CUENCA

- **Flujo Normal de eventos**

Pre condición:	- El Alumno debe tener una cuenta creada y almacenada en la base de datos del Moodle
N	Descripción del paso
o	
1	La aplicación pregunta nombre de usuario y contraseña.
2	El Alumno ingresa sus datos y pide el acceso a la aplicación
3	La aplicación hace una llamada al Web Service de autenticación del Moodle (Para el caso de la aplicación en Android) o directamente al repositorio (Para el caso de la aplicación para la Web).
4	La aplicación valida los datos del usuario que se encuentran en la base de datos del Moodle.
5	Si el usuario no existe se informa del error y vuelve al paso 1
Pos condición:	- El usuario ha ingresado correctamente a la aplicación, y tendrá acceso a sus funcionalidades.

- **Diagrama del Caso de Uso**



UNIVERSIDAD DE CUENCA

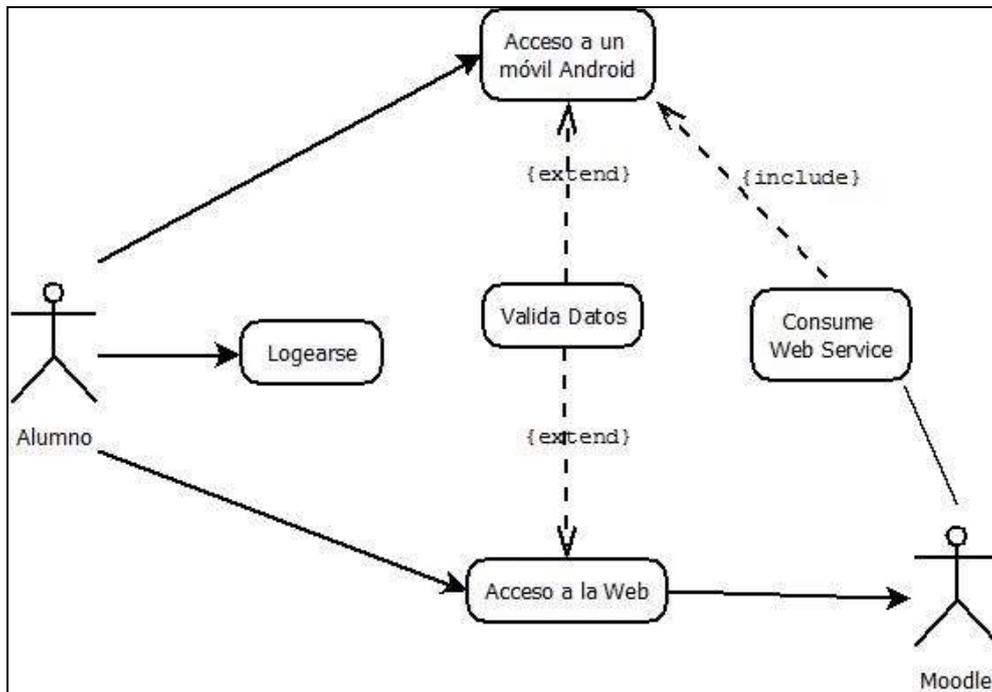


Figura 4.5-A Diagrama de caso de uso Ingresar a la aplicación

4.5.4.2 Caso de Uso para acceder a contenidos de la materia.

- **Referencias**

- Resumen de casos de uso.
- Características del Prototipo

- **Lista de Actores**

- **Alumno**

Es el usuario que ingresa a la aplicación y puede acceder a los contenidos de la materia y al desarrollo de misma como ayuda para su aprendizaje.

- **Moodle**

Repositorio en el que se almacenara toda la información necesaria.

- **Flujo Normal de eventos**

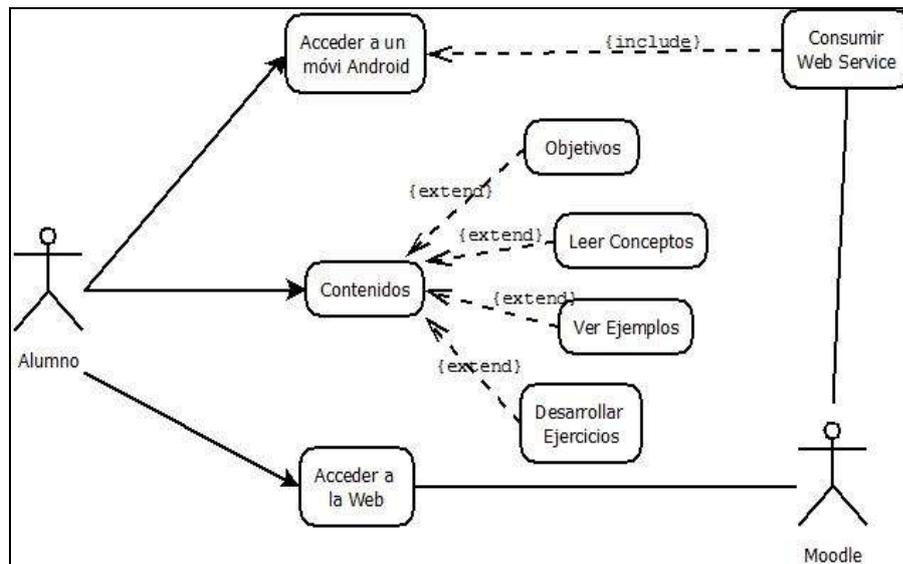
Pre condición:	- El Alumno debe haberse autenticado correctamente y haber ingresado a l aplicación.
Nº	Descripción del paso



UNIVERSIDAD DE CUENCA

Pre condición:	- El Alumno debe haberse autenticado correctamente y haber ingresado a la aplicación.
Nº	Descripción del paso
1	La aplicación consume el Servicio Web y accede al Moodle para cargar en pantalla el menú de usuario (Para el caso de la Aplicación en Android).
2	La aplicación carga en pantalla el menú de usuario directamente desde el repositorio Moodle (Para el caso de la Aplicación en la Web).
3	El alumno puede acceder a los contenidos de la materia.
4	El alumno puede leer la teoría o conceptos referentes a cada tema.
5	El alumno puede visualizar los ejercicios resueltos propuestos de acuerdo al tema.
6	El alumno puede desarrollar ejercicios a manera de autoevaluación del tema aprendido.
Pos condición:	- Se podrá visualizar el avance del alumno en la materia, y dicha información se encuentra almacenada en el repositorio de Moodle.

- Diagrama del Caso de Uso





UNIVERSIDAD DE CUENCA

Figura 4.5-B Diagrama caso de uso Acceder a los contenidos de la materia

4.5.4.3 Caso de Uso para llevar un mantenimiento del contenido de la materia.

- **Referencias**

- Resumen de casos de uso.
- Características del Prototipo

- **Lista de Actores**

- **Administrador**
Es el usuario que ingresa a la aplicación con permisos de administrador y puede acceder y dar un mantenimiento a los contenidos de la materia.
- **Moodle**
Repositorio en el que se almacenara toda la información necesaria.

- **Flujo Normal de eventos**

Pre condición:	- El usuario Administrador debe estar previamente registrado en la base de datos del Moodle con permisos administrativos.
Nº	Descripción del paso
1	La aplicación pide nombre de usuario y contraseña.
2	El usuario ingresa sus datos y la aplicación accede al Moodle directamente (Aplicación en la Web) o a través del consumo de un servicio Web (Aplicación en Android)
3	La aplicación valida los datos del usuario que se encuentran en la base de datos del Moodle.
4	Si los datos son correctos el administrador ingresa a la aplicación correctamente.
5	El Administrador puede hacer un mantenimiento de los contenidos de la materia.
6	El Administrador deberá guardar los cambios realizados.



UNIVERSIDAD DE CUENCA

Pre condición:	- El usuario Administrador debe estar previamente registrado en la base de datos del Moodle con permisos administrativos.
Nº	Descripción del paso
Pos condición:	- La información creada, modificada o eliminada por parte del Administrador será almacenada en la base de datos del Moodle para su posterior visualización de los alumnos.

• Diagrama del Caso de Uso

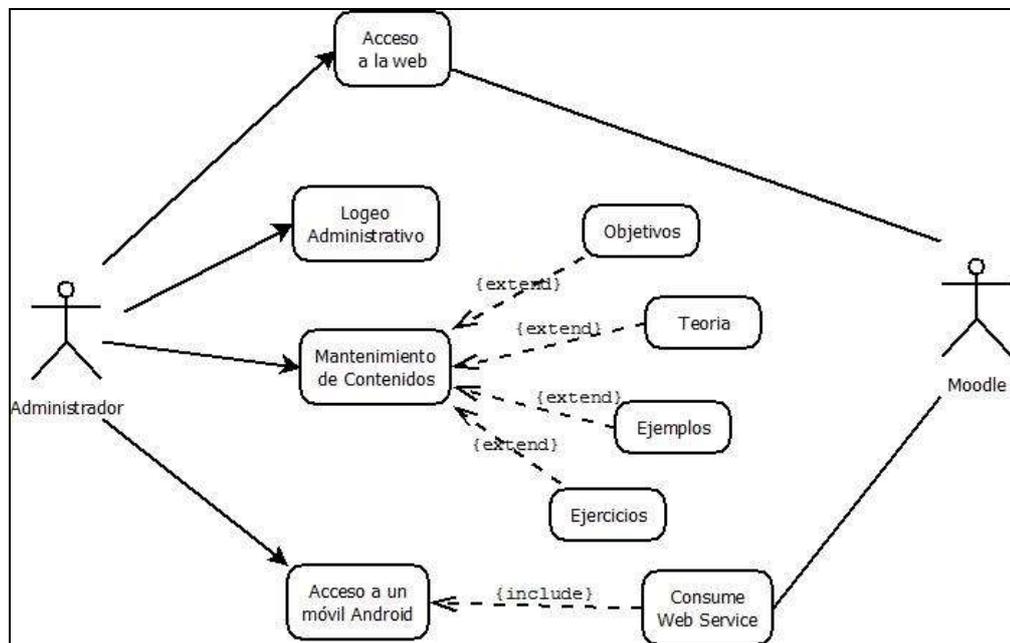


Figura 4.5-C Diagrama caso de uso Mantenimiento del contenido de la materia

4.5.5 Resumen de Casos de Uso

El resumen de casos de uso tiene por objeto resumir la funcionalidad descrita en las distintas especificaciones de casos de uso reconocidos para un software en particular. Un interesado del proyecto podrá a través de este resumen tener una visión global de cómo se está planteando la solución a nivel funcional para el software a construir.

4.5.5.1 Referencias

Detalle de los diferentes Casos de Uso

- Ingreso a la aplicación.



UNIVERSIDAD DE CUENCA

- Acceso a los contenidos de la materia.
- Mantenimiento de los contenidos de la materia.

4.5.5.2 Diagrama de Casos de Uso

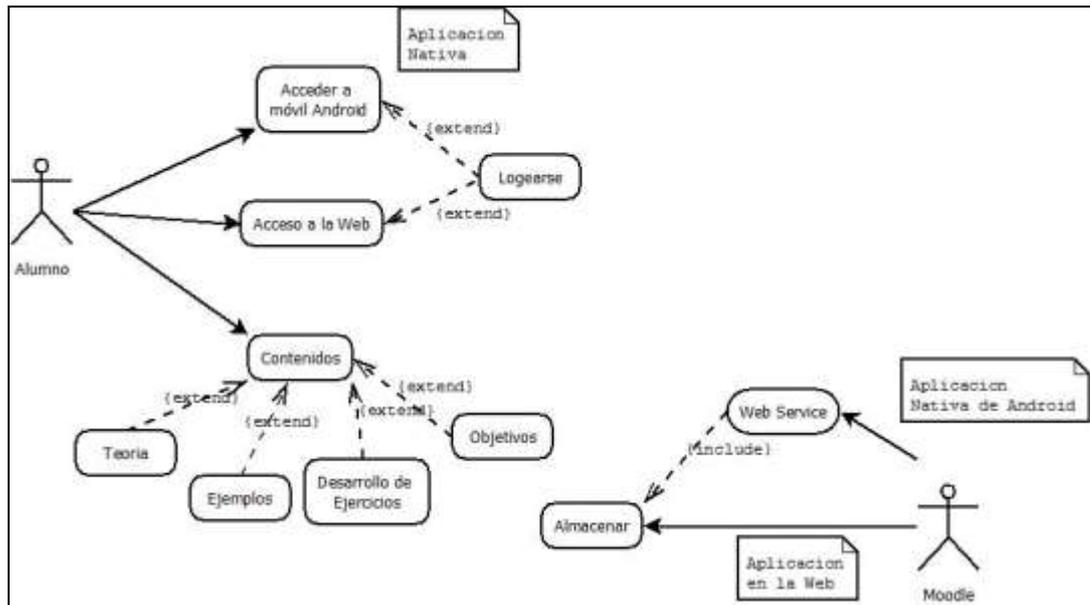


Figura 4.5-D Resumen de casos de uso

4.5.5.3 Lista de Actores

- Alumno
- Administrador
- Moodle

4.5.5.4 Lista de Casos de Uso

- Ingresar a la Aplicación.
- Acceso a los contenidos de la materia
- Mantenimiento a los contenidos de la materia

4.5.5.5 Descripción de los Casos de Uso

- *Ingresar a la Aplicación.*- Consiste en el Ingreso al Sistema del usuario, de manera que pueda acceder a las diferentes opciones de los menús.
- *Acceso a los contenidos de la materia.*- Aquí se define el acceso que tiene el alumno a las diferentes funcionalidades de la aplicación.



UNIVERSIDAD DE CUENCA

- *Mantenimiento a los contenidos de la materia.*- Consiste en acceso del Administrador a la aplicación para realizar procesos de creación, modificación y eliminación de los contenidos de la materia.

4.6 RESUMEN DEL CAPITULO

En este capítulo se ha podido indicar el proceso de análisis correspondiente al prototipo de enseñanza-aprendizaje para la materia piloto *Cálculo Diferencial*. Esta fase es considerada una de las más importantes en el proceso de creación del sistema, ya que se presenta de una manera general el funcionamiento que tendrá la aplicación en los diferentes ámbitos, así como la interacción de los usuarios en cada una de las actividades.

Concluido este capítulo se cuenta con toda la información necesaria para el proceso de diseño y desarrollo de las herramientas de aprendizaje, proceso que se detalla en los dos siguientes capítulos de una manera específica, en el capítulo 4 para la aplicación móvil y en el Capítulo 5 para la Web.



UNIVERSIDAD DE CUENCA

CAPITULO 5:

**“Diseño y desarrollo de la aplicación cliente Android para Moodle:
EVIAM.”**



UNIVERSIDAD DE CUENCA

5.1 INTRODUCCIÓN

La aparición de dispositivos portables capaces de manejar grandes cantidades de información y el acceso móvil a Internet, han facilitado la aparición de nuevas formas de educación conocidas como *MLearning* o aprendizaje móvil. *MLearning* es la adquisición de un conocimiento y habilidades a través del uso de la tecnología móvil, en cualquier lugar, en cualquier momento (Geddes 2004), permitiéndole al alumno acceder a los contenidos de aprendizaje sin importar limitaciones en cuanto a horarios y distancia. Pero, ¿es suficiente la capacidad de acceder a los contenidos sin limitaciones para garantizar una buena educación?

Obviamente la respuesta es NO, y es que se necesita de una herramienta que gestione estos contenidos y permita organizarlos, estructurarlos manteniendo un orden y una secuencia y que facilite el seguimiento del aprendizaje del alumno. Este tipo de herramientas se denominan Sistemas de Administración de Aprendizaje, conocidos como LMS del inglés *Learning Management System*. Dentro de estos sistemas LMS encontramos a Moodle, creado en 2002 por Martin Dougiamas, con más de 65.000 sitios Web registrados y más de 58 millones de usuarios¹² en todo el mundo.

Al ser Moodle el LMS utilizado por la Universidad de Cuenca, el prototipo de aprendizaje se ha basado en este sistema para ofrecer los contenidos de la materia piloto *Cálculo Diferencial*. Es por ello que, para permitir el acceso de los estudiantes a través de sus dispositivos móviles, se decidió desarrollar una aplicación nativa Android que actúe como cliente del servidor Moodle. En el presente capítulo se presentan los detalles en cuanto al diseño y desarrollo de esta aplicación móvil, la cual hemos denominado **EVIAM** por las siglas de Entorno **V**irtual de **A**prendizaje **M**óvil.

Objetivos del Capítulo

Los objetivos que se pretenden alcanzar son:

- Diseñar una aplicación móvil que permita a los usuarios del entorno Moodle conectarse al sistema desde dispositivos móviles.
- Presentar los detalles en cuanto al desarrollo y pruebas de la aplicación.

5.2 DESCRIPCIÓN DE LA APLICACIÓN

El objetivo principal de la aplicación es permitir el acceso de los estudiantes a los contenidos de aprendizaje que se encuentran almacenados en el sistema Moodle de la Universidad de Cuenca a través de un dispositivo móvil con sistema Android. La aplicación permitirá adaptar los contenidos al tamaño

¹²Estadísticas obtenidas desde la página oficial de Moodle: <http://moodle.org/stats/> (30-08-2012)



UNIVERSIDAD DE CUENCA

reducido de las pantallas de los teléfonos móviles y facilitará la navegación de los mismos, evitando, en lo posible, los costos que implican las conexiones de datos en este tipo de dispositivos mediante persistencia y almacenamiento local de datos en caché.

Los estudiantes tendrán acceso a la aplicación a través del nombre de usuario y contraseña que le fueron previamente asignados en Moodle por el Administrador del sistema, evitando la necesidad de crear nuevas cuentas de usuario o de realizar algún tipo de configuración adicional. Una vez que inicien sesión, la aplicación permitirá: visualizar la lista de cursos en los que el estudiante se encuentra matriculado, las secciones en las que se divide el curso, los contenidos de cada sección, los perfiles de los compañeros organizados por curso, la información del perfil del estudiante y la comunicación entre compañeros a través de llamadas telefónicas y mensajes de texto.

5.3 DISEÑO DE LA APLICACIÓN

La siguiente sección tiene el objetivo de presentar los detalles del diseño para la solución propuesta. Servirá de guía para el desarrollo de la aplicación.

5.3.1 Generalidades de la Aplicación

¿Es una nueva Aplicación?	Si	¿Reemplaza a alguna aplicación existente?	No
Si reemplaza, especificar a cual/es:	-		
¿Agrega funcionalidad a alguna aplicación existente?	Si	Si agrega funcionalidad, especificar a cual/es:	Moodle: acceso desde un dispositivo móvil Android.
Sectores que utilizarán la solución a desarrollar:	<ul style="list-style-type: none"> Estudiantes de los primeros años de la Facultad de Ingeniería de la Universidad de Cuenca. 		

5.3.2 Arquitectura General del Sistema

En la **Figura 5.3-A** podemos observar la arquitectura general del sistema, la cual está dividida en dos partes:

- **Moodle**, que actúa como Gestor de Contenidos.
- La **Aplicación Móvil**, en este caso la aplicación nativa Android.

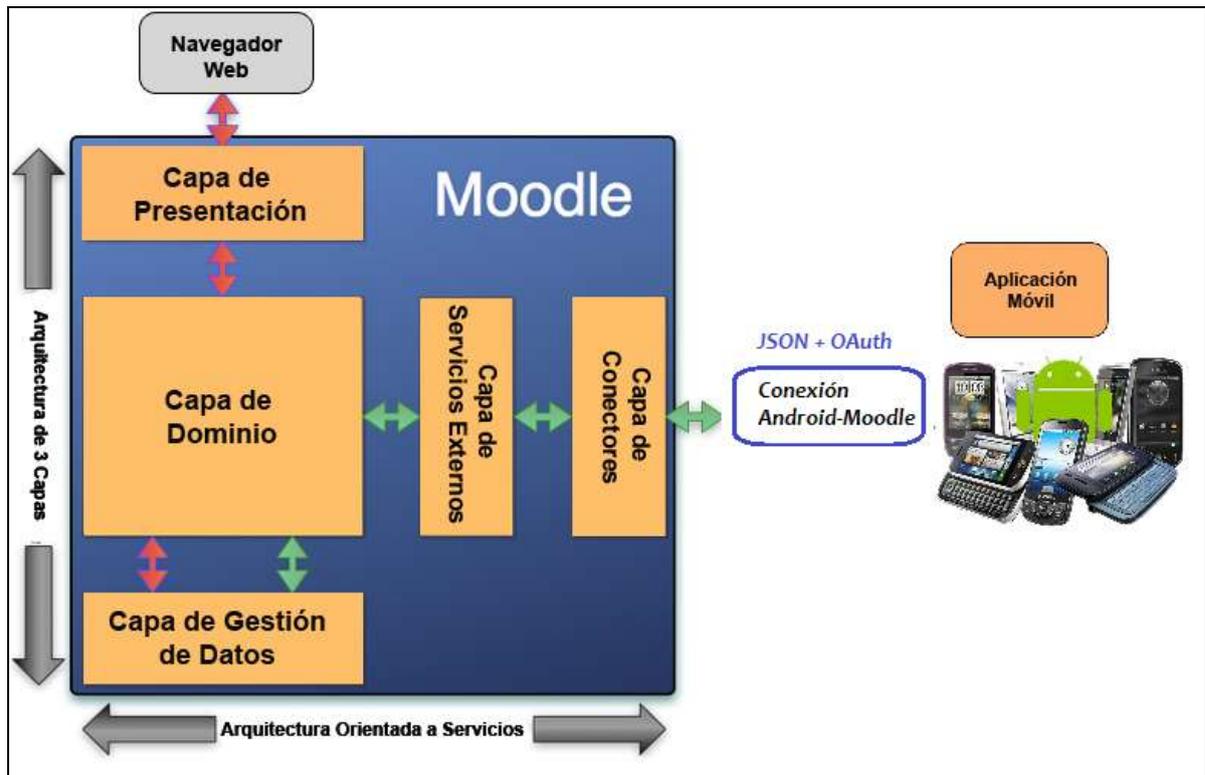


Figura 5.3-A Arquitectura General del Sistema

5.3.2.1 Moodle

Como se mencionó anteriormente, la Universidad de Cuenca utiliza Moodle como sistema LMS, por lo que el prototipo requiere que la aplicación móvil interactúe con esta plataforma. Moodle maneja dos tipos de arquitectura, una del tipo 3 capas que se encuentra representada en la **Figura 5.3-A** de manera vertical y una orientada a servicios representada de manera horizontal.

La arquitectura de 3 capas de Moodle es la que permite su funcionamiento como sistema LMS. Se compone por la **Capa de Gestión de Datos**, que se encarga de almacenar y recuperar los datos del sistema, la **Capa de Dominio** que hace de API¹³ de funciones del núcleo (core functions), las cuales proveen la lógica del sistema, interesante para ser ofrecida al exterior, y la **Capa de Presentación** que adapta los contenidos para ser presentados en los navegadores Web.

La arquitectura orientada a servicios de Moodle permite la interoperabilidad de Moodle con otras aplicaciones. Se compone de la **Capa de Servicios Externos** y sus llamados métodos externos (external methods) que el administrador puede crear para hacer visibles los servicios que Moodle ofrece

¹³API (Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.



UNIVERSIDAD DE CUENCA

al exterior, como por ejemplo las funciones para crear usuarios, visualizar el contenido de un curso determinado, la lista de alumnos inscritos en un curso, etc. y la **Capa de Conectores**, donde cada conector de forma independiente y bajo distintos protocolos posibles (ver sección 4.4 Conexión Android-Moodle) transforma los métodos externos en servicios Web.

5.3.2.2 Aplicación móvil

El cliente móvil es en sí la aplicación Android a desarrollar. Esta aplicación se comunica con el servidor Moodle a través de servicios Web y de esta forma obtiene los datos para adaptarlos y presentarlos en la pantalla del móvil. La aplicación tiene su arquitectura propia basada en el patrón Modelo Vista Controlador.

5.3.2.2.a Modelo

En la **Figura 5.3-B** podemos observar las clases que componen el modelo de la aplicación. Básicamente estas clases manejan la lógica de manipulación de los datos, ya sea mediante el uso de los servicios Web o de la caché.

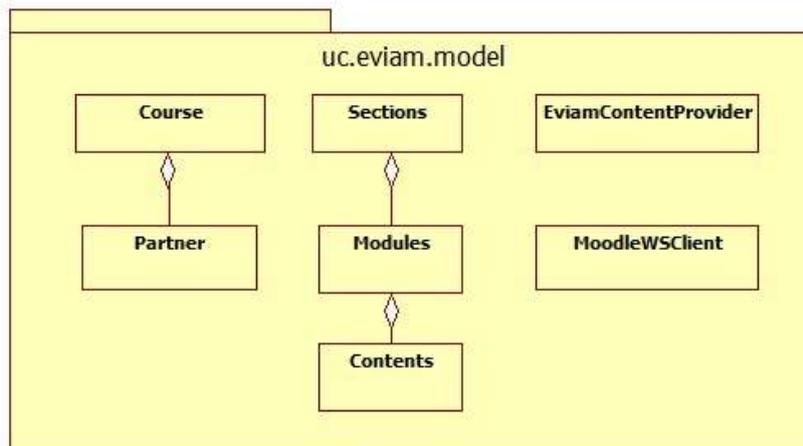


Figura 5.3-B Clases que componen el Modelo de la Aplicación Móvil

A continuación se describe brevemente la función que cumple cada clase dentro del modelo:

- **Course:** Contiene los procedimientos que permiten gestionar la información relacionada con los cursos en los que se encuentra matriculado el estudiante.
- **Partner:** Gestiona la información de los compañeros que se encuentran matriculados en el mismo curso que el usuario que inició sesión en la aplicación.
- **Sections:** Permite gestionar las secciones en las que se encuentran divididos los cursos.
- **Modules:** Gestiona la información de los módulos contenidos en el curso: documentos, foros, imágenes, actividades, etc.
- **Contents:** Maneja información adicional de los módulos como: nombres de archivos, autores, tipo de módulo, tiempo de creación y modificación.



UNIVERSIDAD DE CUENCA

- **Eviam Content Provider:** Esta clase se encarga de gestionar la base de datos interna de la aplicación para crear, actualizar, eliminar y recuperar información desde caché.
- **MoodleWSCClient:** Esta clase se encarga de gestionar las llamadas a los servicios Web para recuperar información desde el servidor Moodle.

5.3.2.2.b Vista

En la **Figura 5.3-C** podemos observar las clases que componen la vista de la aplicación. Estas clases se encargan de presentar la información al usuario mediante **Actividades de Android**, las cuales son el equivalente a los formularios en las aplicaciones de escritorio, es decir, son los contenedores sobre los cuales se colocan los elementos de la interfaz como botones, cuadros de texto, imágenes, etc.

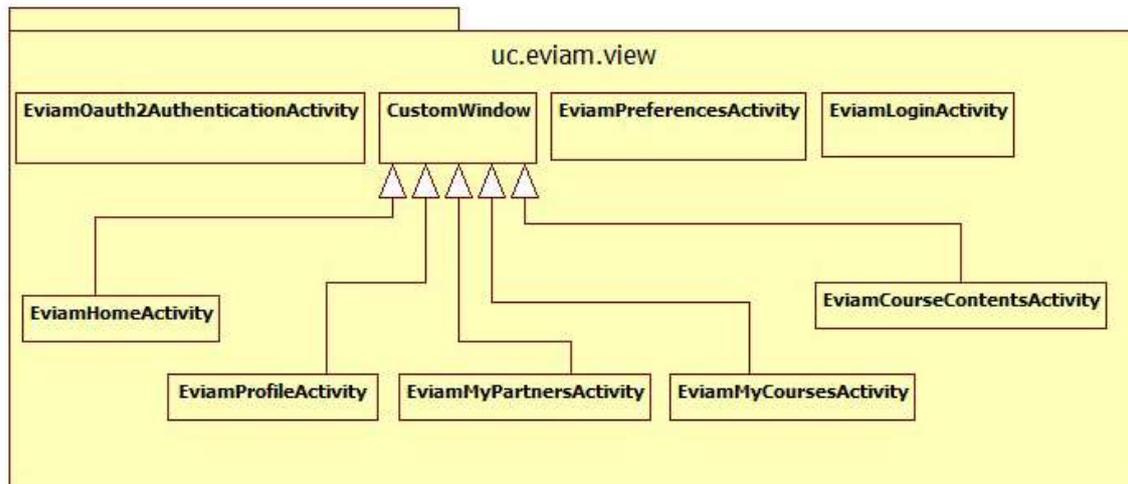


Figura 5.3-C Clases que conforman la Vista de la Aplicación Móvil

A continuación se describe brevemente la función que cumple cada clase dentro de la vista:

- **EviamOauth2AuthenticationActivity:** Esta clase se encarga del proceso de autenticación del usuario mediante el uso del protocolo OAuth2.0 (ver sección 4.4.1)
- **Custom Window:** Esta clase permite personalizar la apariencia de los controles de las actividades de Android, por ejemplo cambiar la forma de los botones, la imagen de fondo, los cuadros de texto, etc.
- **Eviam Home Activity:** Actividad que presenta al usuario la pantalla de inicio cuando finaliza el proceso de autenticación.
- **Eviam Profile Activity:** Presenta al estudiante la información de su perfil o del perfil de sus compañeros dependiendo de la opción seleccionada.
- **Eviam My Partners Activity:** Presenta la pantalla que contiene la lista de los compañeros de curso del estudiante.



UNIVERSIDAD DE CUENCA

- **Eviam My Courses Activity:** Muestra la pantalla que contiene la lista de los cursos donde se encuentra matriculado el estudiante.
- **Eviam Course Contents Activity:** Presenta la pantalla que contiene la lista de los contenidos de aprendizaje del curso que haya seleccionado el estudiante.
- **EviamPreferencesActivity:** Actividad que permite modificar las preferencias de la aplicación como activar/desactivar el uso de la caché y modificar el tiempo de validez de los datos almacenados.
- **EviamLoginActivity:** Presenta la pantalla de bienvenida de la aplicación e inicia el proceso de autenticación.

5.3.2.2.c Controlador

En la **Figura 5.3-D** observamos las clases que componen el controlador de la aplicación. Estas clases se encargan de responder a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y la vista.

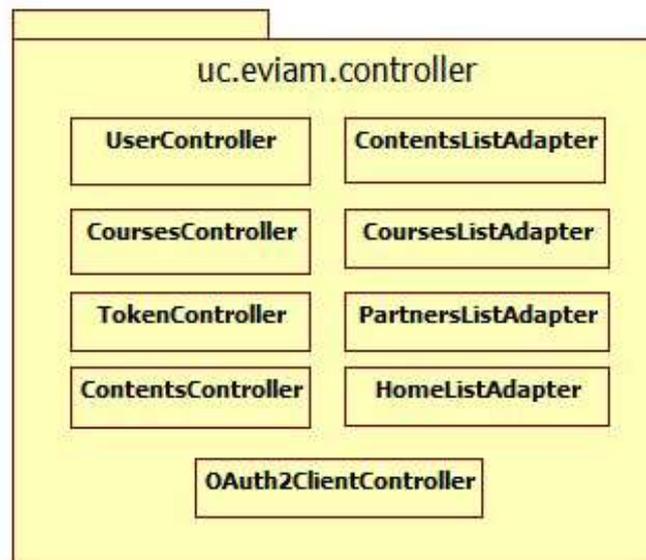


Figura 5.3-D Clases que conforman el Controlador de la Aplicación Móvil

A continuación se describe brevemente la función que cumple cada clase dentro del controlador:

- **User Controller:** Contiene los procedimientos para gestionar las solicitudes de información relacionada con los datos de usuario.
- **Courses Controller:** Permite controlar las solicitudes de información relacionada con los datos de los cursos donde se encuentra matriculado el usuario.
- **Contents Controller:** Controla las solicitudes de información relacionada con los contenidos de los cursos.



UNIVERSIDAD DE CUENCA

- **Token Controller:** Gestiona la obtención y renovación de los tokens de autenticación que utiliza el protocolo OAuth2.0 para hacer las llamadas a los servicios Web (ver sección 4.4.1).
- **OAuth2ClientController:** Gestiona los procedimientos relacionados con el protocolo de autenticación OAuth2.0 (ver sección 4.4.1).
- **Contents List Adapter:** Permite adaptar la información de los contenidos de los cursos en un arreglo¹⁴ para presentarlos como una lista.
- **Courses List Adapter:** Adapta la información de los cursos en un arreglo para presentarlos en forma de lista.
- **Partners List Adapter:** Adapta la información de los compañeros de curso en un arreglo para poder presentarlos en una lista.
- **Home List Adapter:** Clase encargada de adaptar las opciones de la lista de la pantalla de inicio.

5.4 CONEXIÓN ANDROID-MOODLE

Una vez definida la arquitectura del sistema, el siguiente punto a resolver es la conexión entre la aplicación móvil y el servidor Moodle. Como se explicó en la **sección 5.3.2.1** Moodle cuenta con una capa de conectores que hace visibles una serie de funciones a través de servicios Web bajo distintos protocolos. Los protocolos oficialmente soportados por Moodle son: **REST¹⁵, SOAP¹⁶, XML-RPC¹⁷ y AMF¹⁸.**

Obviamente el acceso a cualquier función del Moodle a través de los servicios Web requiere de autenticación por motivos de seguridad. Nativamente Moodle cuenta con dos métodos de autenticación para servicios Web: **usuario/contraseña y token de usuario.**

La autenticación por nombre de usuario y contraseña, también conocida como autenticación simple, es obsoleta y no recomendable. Como su nombre lo indica, este tipo de autenticación consiste en enviar el nombre de usuario y contraseña en cada llamada al servicio Web. Si no se cuenta con medios de encriptación adecuados, la información podría viajar en texto plano por lo que genera un enorme problema de seguridad; y aún si la información viajase encriptada, la aplicación debería mantener los datos del nombre de usuario y contraseña almacenados para realizar las llamadas a los servicios Web sin

¹⁴Un arreglo es un conjunto de datos o una estructura de datos homogéneos que se encuentran ubicados en forma consecutiva en la memoria.

¹⁵REST (Representational State Transfer) es un modelo orientado a recursos, donde cada URL es la representación de un objeto o recurso.

¹⁶SOAP (Simple Object Access Protocol) protocolo para el intercambio de mensajes que utiliza XML y es independiente de la plataforma o sistema operativo, utilizado para codificar información en los diálogos de los Servicios Web

¹⁷XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.

¹⁸AMF formato binario desarrollado por Adobe/Macromedia para interactuar con otras tecnologías del lado servidor a través de la invocación de Web Services.



UNIVERSIDAD DE CUENCA

tener que solicitar al usuario que ingrese nuevamente sus credenciales, lo que resulta peligroso ya que se trata de información delicada.

La autenticación por token de usuario puede realizarse de dos formas:

En la primera, el administrador del sistema Moodle es el encargado de generar manualmente el token de acceso. Se debe generar **un token por cada usuario y por cada servicio Web**. Es decir, un mismo usuario puede tener varios tokens de acceso (uno por cada servicio Web que utilice) y cada token funciona únicamente para el servicio Web al que fue asignado. Esto resulta ser un problema cuando se tiene un número considerable de estudiantes a quienes garantizar el acceso, ya que además de la dificultad inicial de generar y asignar tokens a cada usuario, si se necesita renovar los tokens por cualquier motivo primero deben ser revocados manualmente antes de repetir nuevamente todo el proceso de generación y asignación.

En la segunda forma de autenticación por token de usuario, el token se genera automáticamente mediante la llamada a un script de autenticación de los servicios Web. El cliente envía el nombre de usuario y contraseña a este script y recibe a cambio un token de acceso el cual utilizará para realizar las llamadas a los servicios Web. Este método evade la necesidad de enviar en cada solicitud el nombre de usuario y contraseña, pero aún debe hacerse cada vez que se requiere obtener el token de usuario, por lo que igualmente la aplicación necesita almacenar estos datos para no solicitar al usuario que los ingrese nuevamente en el caso que aún no haya cerrado su sesión y el token haya cumplido su tiempo de vencimiento.

Únicamente la primera forma de autenticación por token de usuario evita que la aplicación conozca y envíe el nombre de usuario y contraseña del cliente, ya que no necesita hacer la llamada al script de autenticación para obtener el token; aunque por obvias razones resulta ineficiente y hasta imposible generar un token por cada usuario de la Universidad de Cuenca.

Por lo expuesto anteriormente, se decidió utilizar otra forma de autenticación mediante el desarrollo de un plugin para Moodle que permita la autenticación a través del protocolo OAuth 2.0. La principal ventaja al utilizar este método en lugar de cualquiera de las formas de autenticación que ofrece nativamente Moodle, es que la aplicación no requiere conocer ni almacenar el nombre de usuario y contraseña del estudiante en ningún momento. El usuario inicia sesión normalmente en Moodle desde el navegador Web y autoriza el acceso de la aplicación a los datos requeridos. Los detalles del protocolo OAuth2.0 y su proceso de autenticación se presentan a continuación en la **sección 5.4.1**.

5.4.1 OAuth 2.0

El framework de autorización OAuth 2.0 (www.oauth.net) permite a una aplicación de terceros obtener acceso limitado a un servicio HTTP, ya sea en nombre del propietario del recurso por la orquestación de una interacción entre



UNIVERSIDAD DE CUENCA

la aprobación del propietario del recurso y el servicio HTTP, o permitiendo a la aplicación de terceros obtener acceso en su propio nombre (IETF, 2012). OAuth 2.0 define cuatro roles:

- ❖ **Propietario del recurso** Una entidad capaz de garantizar el acceso a un recurso protegido. Cuando el propietario del recurso es una persona, se lo conoce como usuario final.
- ❖ **Servidor de recursos** El servidor que aloja los recursos protegidos, capaz de aceptar y responder a las solicitudes de recursos protegidos a través de tokens de acceso.
- ❖ **Cliente** Una aplicación que hace las solicitudes de recursos protegidos en nombre del propietario y con su autorización. El término cliente no implica ninguna característica particular (por ejemplo, si la aplicación se ejecuta en un servidor, un ordenador de sobremesa, o de otro tipo dispositivos).
- ❖ **Servidor de autorización** El servidor que emite los tokens de acceso al cliente después de la autenticación exitosa del propietario del recurso y la obtención de la autorización.

En la **Figura 5.4-A** podemos observar el proceso de autenticación con OAuth 2.0.

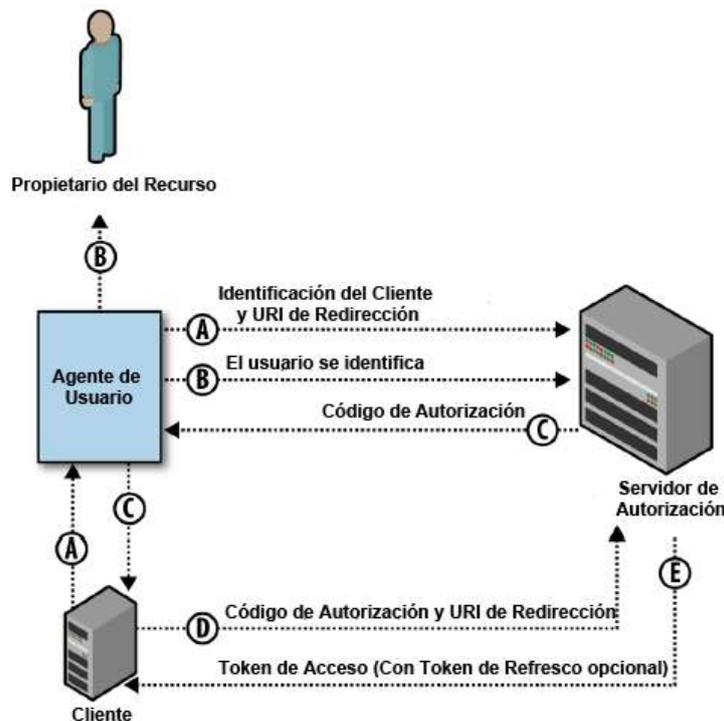


Figura 5.4-A Proceso de Autenticación OAuth 2.0



UNIVERSIDAD DE CUENCA

(A) El Cliente (en este caso la aplicación Android) se identifica ante el Servidor de Autorización enviando una identificación y una URI de redirección previamente registradas (**Figura 5.4-B**).

Lista de Clientes OAuth 2.0

Título de la Aplicación	ID del Cliente	Descripción de la aplicación
Cliente Android de Eviam	f86f8a789f08131a41b9fbf90966784b0503068f3	Aplicación nativa de Android para acceder a Eviam

[Registrar un nuevo Cliente](#)

Figura 5.4-B Credenciales previamente almacenadas en Moodle

(B) A continuación Moodle solicitará al usuario ingresar como lo hace normalmente con su nombre de usuario y contraseña (**Figura 5.4-C**).

Entorno Virtua...

Bloques Configuración

Usuarios registrados

Entre aquí usando su nombre de usuario y contraseña
(Las 'Cookies' deben estar habilitadas en su navegador) ?

Nombre de usuario

Contraseña

Recordar nombre de usuario

[¿Olvidó su nombre de usuario o contraseña?](#)

Figura 5.4-C Solicitud de Inicio de sesión de Moodle

(C) Una vez que el usuario inicia sesión en Moodle se le presenta una solicitud de autorización de la aplicación que intenta obtener los datos a través de OAuth 2.0 (**Figura 5.4-D**), Si el usuario autoriza a la aplicación, el cliente recibe un código de autorización que puede intercambiarlo por un token de acceso.

Solicitud de Autorización

Cliente Android de Eviam

Esta aplicación está solicitando permiso para acceder a sus datos.

Figura 5.4-D Solicitud de Autorización



UNIVERSIDAD DE CUENCA

(D) El Cliente envía el código de autorización recibido y la URI de Redirección al Servidor de Autorización.

(E) El Servidor intercambia el código de autorización por un token de acceso y lo envía a la URI de Redirección recibida junto con un token de refresco opcional.

Una vez que el Cliente recibe el token de acceso, lo almacena para poder realizar las llamadas a los servicios Web utilizándolo como medio de autenticación. Los tokens tienen un tiempo de validez configurable, por lo que el Cliente también recibe un token de refresco que lo utiliza para renovar el token de acceso de forma transparente para el usuario.

Para que Moodle permita utilizar un token OAuth 2.0 como identificación para acceder a los servicios Web, fue necesario desarrollar un plugin que contenga un servidor REST con autenticación OAuth 2.0. Básicamente lo que se hizo fue heredar el servidor REST nativo de Moodle y modificar el procedimiento de autenticación que manejaba para que ahora solicite el envío de un token OAuth2.0 en la llamada al servicio Web, verifique y valide este token y finalmente otorgue el acceso solicitado si cumple con todos los procesos de validación.

5.4.2 Servicios Web de Moodle

El acceso de la aplicación móvil tanto a los datos como a las funciones de Moodle se lo realiza a través de servicios Web. El administrador puede crear nuevos servicios Web en la pantalla de **Administración del sitio > Extensiones > Servicios Web** del Moodle y asignarle las funciones externas a las cuales se podrá tener acceso. Para que la aplicación móvil se comunique, se ha creado el servicio Web “EviamWS” y se le ha asignado las funciones necesarias como puede verse en la **Figura 5.4-E**.

Agregar funciones al servicio "EviamWS"			
Función	Descripción	Required capabilities	Editar
core_enrol_get_enrolled_users	Get enrolled users by course id.	moodle/user:viewdetails, moodle/user:viewhiddendetails, moodle/course:useremail, moodle/user:update, moodle/site:accessallgroups	Eliminar
core_enrol_get_users_courses	Get the list of courses where a user is enrolled in	moodle/course:viewparticipants	Eliminar
local_course_get_course_by_id	Return course details	moodle/course:view	Eliminar
local_course_get_contents	Return course contents	moodle/course:view	Eliminar
local_user_get_logged_user_id	Return current user's id	moodle/user:viewdetails, moodle/user:viewhiddendetails	Eliminar
local_user_get_user_by_id	Return user details	moodle/user:viewdetails, moodle/user:viewhiddendetails	Eliminar

Figura 5.4-E Funciones del servicio Web EviamWS



UNIVERSIDAD DE CUENCA

A continuación se describen cada una de las funciones que se han utilizado:

5.4.2.1 Función `core_enrol_get_enrolled_users`

Obtiene la lista de usuarios enrolados en un determinado curso.

PARÁMETROS:

Courseid (Requerido): un entero con el código del curso del cual se quiere obtener la lista de alumnos enrolados

options (Opcional): un arreglo de objetos con la siguiente estructura general:

```
lista de (  
objetos {  
    nombre string //nombre de la opción  
    valor string //valor de la opción  
}  
)
```

Los posibles valores para los nombres de las opciones son:

- **withcapability** (string) retorna únicamente usuarios con el permiso especificado.
- **groupid** (int) retorna solamente los usuarios pertenecientes a un grupo especificado.
- **onlyactive** (int) retorna solamente los usuarios que se encuentran enrolados y cuyas cuentas se encuentran activas.
- **userfields** ('string, string,...') retorna únicamente los valores de los campos de información del usuario especificados por este parámetro.
- **limitfrom** (int) desde qué número de registro se desea hacer la consulta.
- **limitnumber** (int) número máximo de usuarios a retornar.

RESPUESTA:

Para la aplicación móvil únicamente se necesita obtener de esta función el nombre completo y el código del alumno, por ello se utilizó la opción **userfields** especificando estos dos campos obteniendo una respuesta con la siguiente estructura:

```
lista de (  
objetos {  
    idint //código del alumno  
    fullname string //nombre completo del alumno  
}  
)
```



UNIVERSIDAD DE CUENCA

5.4.2.2 Función `core_enrol_get_users_courses`

Obtiene la lista de cursos en los que se encuentra enrolado un usuario.

PARÁMETROS:

userid(Requerido): un entero con el código del usuario del cual se quiere obtener la lista de cursos en los que se encuentra enrolado.

RESPUESTA:

La estructura general de la respuesta a esta función es la siguiente:

```
lista de (  
objetos {  
  id int //código del curso  
  shortname string //nombre corto del curso  
  fullname string //nombre complete del curso  
  enrolledusercount int //número de usuarios enrolados en este curso  
  idnumber string //número de identificación del curso  
  visible int //1 significa visible, 0 significa oculto  
}  
)
```

5.4.2.3 Función `local_course_get_course_by_id`

Obtiene la información de un determinado curso.

PARÁMETROS:

id(Requerido): un entero con el código del curso del cual se quiere obtener la información.

RESPUESTA:

La estructura general de la respuesta a esta función es la siguiente:

```
objeto {  
  id int //código del curso  
  shortname string //nombre corto del curso  
  fullname string //nombre complete del curso  
}
```

5.4.2.4 Función `local_course_get_contents`

Obtiene los contenidos de un determinado curso.

PARÁMETROS:

courseid(Requerido): un entero con el código del curso del cual se quiere obtener los contenidos.

RESPUESTA:

La estructura general de la respuesta a esta función es la siguiente:



UNIVERSIDAD DE CUENCA

lista de (
objetos {

```
    id int                //código de la sección del curso
    name string           //nombre de la sección
    visible int Opcional //indica si la sección es visible
    summary string       //descripción de la sección
    summaryformat int   //formato de la descripción
    modules             //lista de módulos de la sección
    lista de (  
    objetos {  
        id int                //código de la actividad
        url string Opcional   //url de la actividad
        name string           //nombre del modulo de la actividad
        description string Opcional //descripción de la actividad
        visible int Opcional //si la actividad está visible u oculta
        modicon string       //url del icono de la actividad
        modname string       //tipo de modulo de la actividad
        modplural string     //nombre en plural del módulo
        availablefrom int Opcional //fecha de habilitación de la actividad
        availableuntil int Opcional //fecha de fin de la actividad
        indent int          //número de tabulaciones en el sitio
        contents           //contenidos de la actividad
        listade (  
        objetos {  
            type string       //un archivo, carpeta o link externo
            filename string   //nombre del archivo
            filepath string   //ruta al archivo
            filesize int      //tamaño del archivo
            fileurl string Opcional //url del archive descargable
            timecreated int   //fecha de creación del archivo
            timemodified int //fecha de última modificación del archivo
            sortorder int    //orden de la lista de archivos
            userid int       //usuario que agregó el contenido
            author string    //creador del contenido
            license string   //licencia del contenido
        }  
    }  
} )  
}
```

5.4.2.5 Función `local_user_get_logged_user_id`

Retorna el código del usuario que inició sesión.

PARÁMETROS:

Esta función no requiere parámetros.

RESPUESTA:

La estructura general de la respuesta a esta función es la siguiente:

objeto {



UNIVERSIDAD DE CUENCA

```
    id int                //código del usuario que inició sesión
}
```

5.4.2.6 Función `local_user_get_user_by_id`

Retorna los datos del perfil de un usuario determinado.

PARÁMETROS:

id(Requerido): un entero con el código del usuario del cual se desea obtener los detalles del perfil.

RESPUESTA:

La estructura general de la respuesta a esta función es la siguiente:

```
objeto {
    id int                //código del usuario
    username string      //nombre de usuario
    fullname string      //nombres completos del usuario
    email string Opcional //correo electrónico del usuario
    address string Opcional //dirección del usuario
    phone1 string Opcional //teléfono fijo
    phone2 string Opcional //teléfono celular
    city string Opcional //ciudad del usuario
    profileimageurl string Opcional //firstname
    enrolledcourses Opcional //cursos en los que está matriculado el usuario
    lista de (
    objetos {
        id int                //código del curso
        fullname string      //nombre completo del curso
        shortname string     //nombre corto del curso
    }
    )
}
```

5.5 PERSISTENCIA DE DATOS

Uno de los objetivos de la aplicación móvil es evitar la conexión permanente con el servidor Moodle para obtener datos. Esto a más de reducir la carga del servidor, ayuda a que la experiencia de usuario sea más agradable ya que los datos se cargan con mayor rapidez. Una vez que los datos son obtenidos vía servicios Web, éstos se almacenan en una caché interna, lo que permite recuperarlos localmente mientras los datos sean considerados como recientes.

La **Figura 5.5-A** muestra la estructura de datos de la caché de la aplicación.



UNIVERSIDAD DE CUENCA

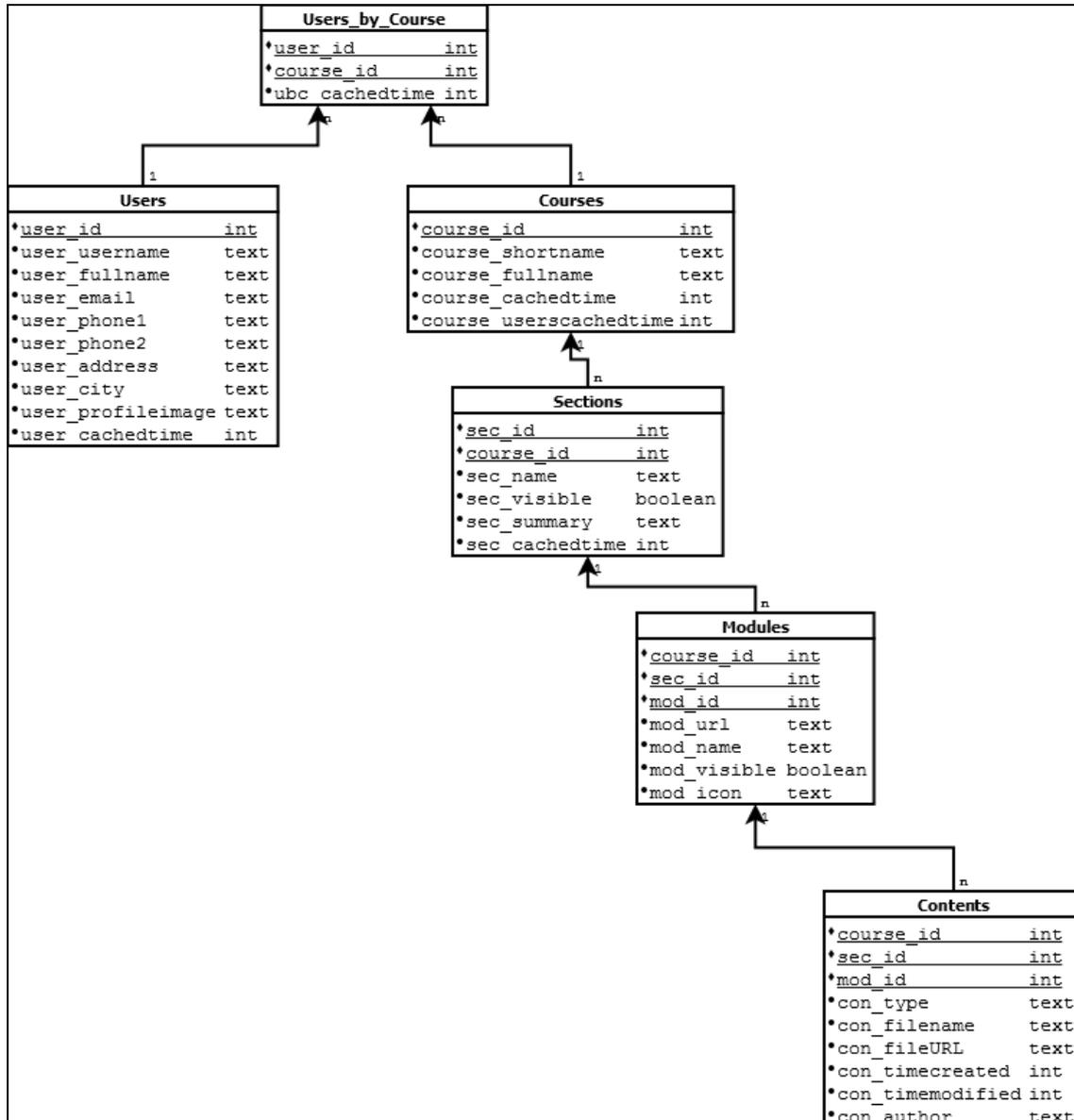


Figura 5.5-A Diagrama de Base de Datos de la Aplicación

La tabla **Users** almacena la información tanto del usuario que inició sesión como de los compañeros de curso del estudiante. Los campos que componen esta tabla son:

- ❖ **user_id:** código que permite identificar un registro único dentro de la tabla.
- ❖ **user_username:** nombre de usuario que utiliza el estudiante para iniciar sesión.
- ❖ **user_fullname:** nombres y apellidos del usuario.
- ❖ **user_email:** dirección de correo electrónico del usuario.



UNIVERSIDAD DE CUENCA

- ❖ **user_phone1**: primer teléfono de contacto del usuario.
- ❖ **user_phone2**: segundo teléfono de contacto del usuario.
- ❖ **user_address**: dirección del usuario.
- ❖ **user_city**: ciudad de nacimiento del usuario.
- ❖ **user_profileimage**: imagen de perfil del usuario.
- ❖ **user_cachedtime**: tiempo en el que los datos ingresaron a la caché.

La tabla **Courses** almacena la información de los cursos. Los campos que componen esta tabla son:

- ❖ **course_id**: código que permite identificar un registro único dentro de la tabla,
- ❖ **course_shortcode**: nombre corto utilizado en la descripción del curso.
- ❖ **course_fullname**: nombre completo del curso.
- ❖ **course_cachedtime**: tiempo en el que los datos del curso ingresaron a la caché.
- ❖ **course_userscachedtime**: tiempo en el que los datos de los compañeros de este curso ingresaron a caché.

La tabla **Users_by_Course** permite relacionar los usuarios con los cursos a los que pertenecen. Los campos que componen esta tabla son:

- ❖ **user_id**: llave foránea que apunta a un registro de la tabla Users
- ❖ **course_id**: llave foránea que apunta a un registro de la tabla Courses
- ❖ **ubc_cachedtime**: tiempo en el que los datos ingresaron a la caché.

La tabla **Sections** almacena la información de las secciones en las que se encuentran divididos los cursos. Se compone de los siguientes campos:

- ❖ **sec_id**: código que permite identificar un registro único dentro de la tabla.
- ❖ **course_id**: llave foránea que relaciona la sección con el curso al que pertenece.
- ❖ **sec_name**: nombre de la sección.
- ❖ **sec_visible**: permite identificar si la sección se encuentra visible o está oculta.
- ❖ **sec_summary**: resumen descriptivo de la sección.
- ❖ **sec_cachedtime**: tiempo en el que los datos ingresaron a la caché.

La tabla **Modules** almacena la información de los módulos o actividades que se encuentran dentro de cada sección de un curso. Se compone de los siguientes campos:

- ❖ **course_id**: llave foránea que relaciona el módulo con el curso al que pertenece.
- ❖ **sec_id**: llave foránea que relaciona el módulo con una sección determinada.
- ❖ **mod_id**: código que permite identificar un registro único dentro de la tabla.



UNIVERSIDAD DE CUENCA

- ❖ **mod_url**: URL donde se encuentra almacenado el módulo dentro de Moodle.
- ❖ **mod_name**: nombre del módulo.
- ❖ **mod_visible**: permite identificar si el módulo se encuentra visible o está oculto.
- ❖ **mod_icon**: icono que identifica el tipo de módulo.

La tabla **Contents** almacena información adicional de los módulos, sobre todo cuando se trata de archivos que fueron subidos al servidor. Se compone de los siguientes campos:

- ❖ **course_id**: llave foránea que relaciona el contenido con el curso al que pertenece.
- ❖ **sec_id**: llave foránea que relaciona el contenido con una sección determinada.
- ❖ **mod_id**: llave foránea que relaciona el contenido con el módulo al que pertenece.
- ❖ **con_type**: tipo de archivo que fue subido.
- ❖ **con_filename**: nombre del archivo que fue subido.
- ❖ **con_fileURL**: URL donde se encuentra almacenado el archivo dentro de Moodle.
- ❖ **con_timecreated**: fecha de creación del archivo.
- ❖ **con_timemodified**: fecha de última modificación del archivo.
- ❖ **con_author**: nombre del usuario que subió el archivo a Moodle.

El uso de esta caché es recomendado pero no obligatorio. El usuario tiene la capacidad de desactivar el uso de la caché, con lo cual la aplicación obtendrá siempre los datos desde el servidor. Esta opción se encuentra en el menú de preferencias de la aplicación como puede observarse en la **Figura 5.5-B**.



Figura 5.5-B Opción de habilitar la caché

De igual forma, si la caché se encuentra activada, el usuario puede escoger durante cuanto tiempo un dato se considera un acierto de caché como se observa en la **Figura 5.5-C**.



UNIVERSIDAD DE CUENCA

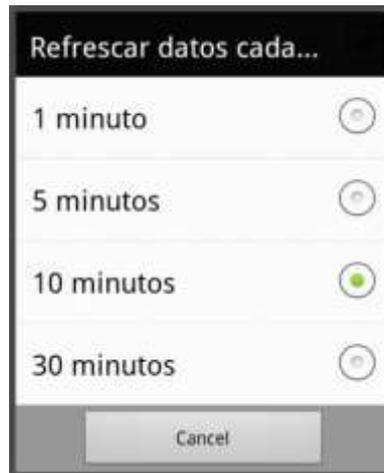


Figura 5.5-C Opción de tiempo de validez de la caché

5.6 OPERACIONES EN SEGUNDO PLANO

Android está pensado especialmente para teléfonos donde es vital que no se bloquee el terminal con operaciones costosas que impidan recibir llamadas. Por este motivo a los pocos segundos de iniciarse un acceso a la base de datos o a los servicios Web, si no se realizan en un hilo aparte, Android cierra la aplicación mostrando el mensaje de que la aplicación no responde como se puede observar en la **Figura 5.6-A**.

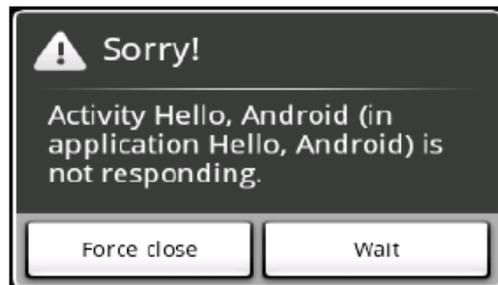


Figura 5.6-A Mensaje de error en Android

Para evitar este comportamiento no deseado se han utilizado hilos¹⁹ para realizar este tipo de operaciones. En concreto se ha gestionado todo el proceso en un hilo aparte mediante tareas asíncronas, realizando la acción y publicando los resultados posteriormente en el hilo principal. Además se presenta un diálogo de progreso mientras la tarea se encuentra ejecutándose en segundo plano como se puede observar en la **Figura 5.6-B**.

¹⁹Un hilo de ejecución o subproceso es una característica que permite a una aplicación realizar varias tareas a la vez (concurrentemente).



UNIVERSIDAD DE CUENCA



Figura 5.6-B Operaciones en segundo plano

5.7 RESUMEN DEL CAPITULO

En este capítulo se ha podido indicar el proceso de desarrollo de la aplicación móvil que brinda el acceso a los contenidos de aprendizaje que se encuentran almacenados en la Web.

En el siguiente capítulo se presentan los detalles relacionados con la creación de los contenidos de la materia piloto Cálculo Diferencial, con el uso de los objetos de aprendizaje.



UNIVERSIDAD DE CUENCA

CAPITULO 6:

“Diseño y creación de los contenidos para la materia piloto Cálculo Diferencial.”



UNIVERSIDAD DE CUENCA

6.1 INTRODUCCIÓN

La inclusión de tecnologías de información y comunicación (TIC) en el proceso de enseñanza-aprendizaje es hoy en día un complemento fundamental para el completo desempeño pedagógico de los estudiantes (Ana Schalk, 2010). La producción de material educativo digital, es una de las opciones más idóneas para proporcionar la información necesaria al alumnado y, que a la vez permita reforzar sus conocimientos adquiridos diariamente. Una temática interesante en el uso de tecnologías informáticas es la creación e implementación de Objetos de Aprendizaje (ODA), el desarrollo de este material educativo implica un trabajo conjunto de profesionales expertos en la materia o tema que se desea impartir y personal con conocimientos en diseño, herramientas multimedia, programación, entre otras.

Para la construcción de este material didáctico se debe considerar tanto factores técnicos como estratégicos que permitan de una manera dinámica y significativa la formación de los estudiantes en un área de conocimiento determinada. Es decir, tener objetivos claros de lo que se desea enseñar y enfocarlos directamente hacia el aprendizaje del alumno.

En el presente capítulo se aborda el diseño y creación del contenido de la materia piloto "*Cálculo Diferencial*", correspondiente a los dos primeros capítulos. Para cumplir con este objetivo, se tomará en consideración el análisis previo realizado en el Cap. 3 y los diferentes conceptos del modelo de enseñanza constructivista, que se han venido tratando a lo largo de esta tesis. También se detallará las herramientas utilizadas para la creación de los objetos de aprendizaje para finalmente implementar dicho material en la plataforma utilizada por la Universidad de Cuenca, Moodle.

Objetivos del Capítulo

Los objetivos que se pretenden alcanzar son:

- Determinar el conjunto de características y procesos necesarios para la construcción de los contenidos.
- Crear a través de las diferentes herramientas un material educativo, dinámico y atractivo para el estudiante, utilizando todas las decisiones de diseño.
- Subir dichos contenidos a la plataforma Moodle de manera que se encuentren disponibles para las pruebas necesarias y accesibles para la aplicación Móvil.

6.2 DESCRIPCIÓN GENERAL DE LOS CONTENIDOS

Para diseñar las actividades de aprendizaje adecuadas, primero se conoció a profundidad los contenidos de la materia y luego se plantearon objetivos claros



UNIVERSIDAD DE CUENCA

de lo que se deseaba enseñar. Para el caso de la materia piloto “*Cálculo Diferencial*”, se contó con el material correspondiente a los dos primeros capítulos de la asignatura, el cuál ha sido proporcionado por los profesores de la Facultad de Ingeniería expertos en el tema. Una vez seleccionado el material necesario, se realizó un esquema de los temas a tratar y el diseño de los objetos de aprendizaje que involucran a los diferentes temas respectivamente.

Para lograr los objetivos de aprendizaje correspondientes a la materia seleccionada, se ha visto importante considerar una estructura tal que, el alumno construya su propio conocimiento y se asegure la integración entre la parte teórica y práctica. Además, como parte primordial de los contenidos, se realizaron actividades auto evaluativas con la finalidad de que el alumno se encuentre consciente de lo aprendido en cada tema.

A pesar que el diseño de los contenidos contempla para esta tesis los dos primeros capítulos, este diseño se encuentra estructurado de tal forma que se puede extender con facilidad al resto del contenido.

6.3 DISEÑO DE LOS CONTENIDOS Y LOS ODAS

En esta sección se presentan los detalles del proceso de diseño de los contenidos y Objetos de Aprendizaje (ODA) para la materia piloto Cálculo Diferencial.

6.3.1 Referencias

- Descripción y Análisis del prototipo

6.3.2 Esquema de los contenidos

Como se mencionó en la parte inicial, el desarrollo de los contenidos de la materia piloto consta de dos capítulos. El primero que trata de funciones y aplicaciones y el segundo de límites y continuidad, cabe indicar que los capítulos escogidos para esta tesis fueron de acuerdo al orden en el que se dicta la materia y no tienen ninguna característica especial que los diferencie de los demás capítulos con respecto a la dificultad de su contenido. Un esquema general de los diferentes temas a tratar en los dos primeros capítulos, se muestra en la **Figura 6.3-A**.



UNIVERSIDAD DE CUENCA

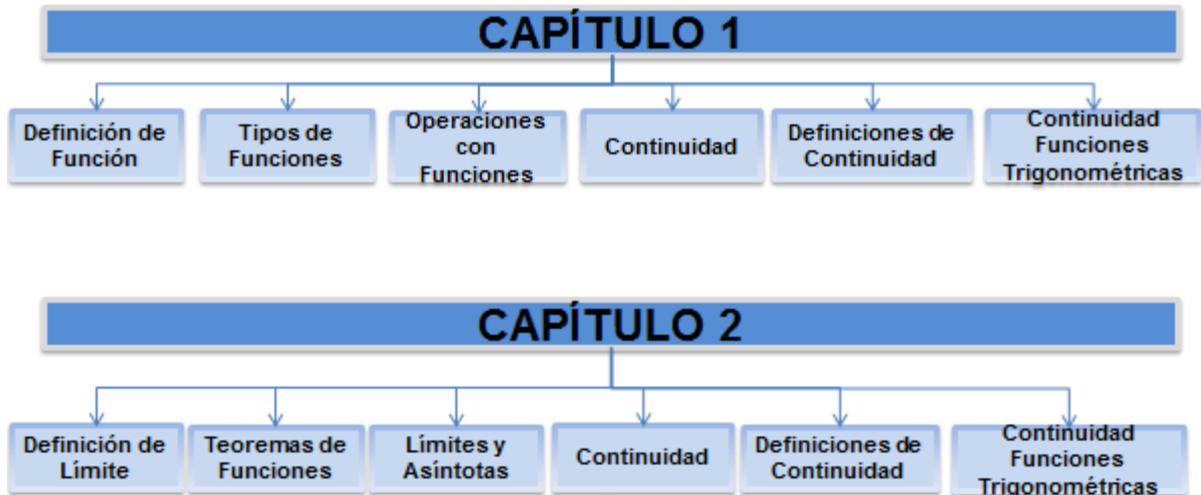


Figura 6.3-A Esquema de los contenidos de Cálculo Diferencial- Cap.1 y Cap.2

6.4 MAPA DE PROCESOS PARA LOS CONTENIDOS

Un mapa de procesos, es la representación gráfica de la estructura de procesos que conforman un sistema (Jaime Beltrán, Miguel Carmona, 2002). Este es el primer paso para adoptar un enfoque organizado y basado en procesos, para el caso del prototipo se ha elaborado un mapa de procesos que indica los diferentes procesos que realizara el prototipo, con una determinada secuencia que permita conocer todos los temas de una manera ordenada, intuitiva y que sobre todo que refuerce los conocimientos adquiridos en la clase por parte de los alumnos.

Para la elaboración del mapa de procesos y con el objetivo de facilitar la interpretación del mismo, a continuación se detallan en la tabla.



UNIVERSIDAD DE CUENCA

Representación simbólica	Significado
	<i>Inicio de un proceso.</i> - Utilizado para expresar el comienzo de un conjunto de actividades.
	<i>Actividad.</i> - Se emplea para representar una actividad o un conjunto de actividades.
	<i>Decisión.</i> - Su salida suele tener al menos dos opciones (SI - NO).
	<i>Flujo de información.</i> - Representa la secuencia en la que se ejecutan las actividades.
	<i>Final de un proceso.</i> - Utilizado para el fin de un conjunto de actividades.

Tabla 6.4-A los principales elementos utilizados (Jaime Beltrán, Miguel Carmona, 2002):

Representación simbólica	Significado
	<i>Inicio de un proceso.</i> - Utilizado para expresar el comienzo de un conjunto de actividades.
	<i>Actividad.</i> - Se emplea para representar una actividad o un conjunto de actividades.
	<i>Decisión.</i> - Su salida suele tener al menos dos opciones (SI - NO).
	<i>Flujo de información.</i> - Representa la secuencia en la que se ejecutan las actividades.



UNIVERSIDAD DE CUENCA

	<i>Final de un proceso.</i> - Utilizado para el fin de un conjunto de actividades.
---	--

Tabla 6.4-A Simbología utilizada en el mapa de procesos

Además de la simbología utilizada en el diagrama, es importante dar a conocer de manera general como funciona la secuencia, interrelación, proceso de control y ejecución eficaz de las actividades. Es necesario recalcar que la secuencia de los procesos y la relación entre ellos siguen un orden especificado por los expertos en la materia de “Cálculo Diferencial”, al igual que la incorporación de conceptos, ejercicios y demás actividades previamente aprobadas. De acuerdo al modelo de enseñanza constructivista, el alumno debe crear su propio conocimiento y autoevaluar su aprendizaje, este control ha sido integrado en el mapa de procesos mediante los elementos de **Decisión** los cuales permiten el paso hacia la siguiente actividad, siempre que el proceso anterior haya sido aprobado correctamente, en caso de no ser así las flechas de flujo de información llevaran al repaso de actividades previas. De esta manera se pretende garantizar la completa ejecución de las actividades por parte de los alumnos.

6.4.1 Mapa de procesos del Capítulo 1

Una vez conocido el concepto de mapa de procesos, la simbología empleada y su funcionamiento, a continuación en la **Figura 6.4-Ase** indica el mapa de procesos correspondiente al **Capítulo 1** de la materia *Cálculo Diferencial*, el cuál está dividido en 6 partes:

- Definición de Función
- Tipos de Funciones
- Operaciones con Funciones
- Función Composición
- Gráfica de una función
- Modelos matemáticos



UNIVERSIDAD DE CUENCA

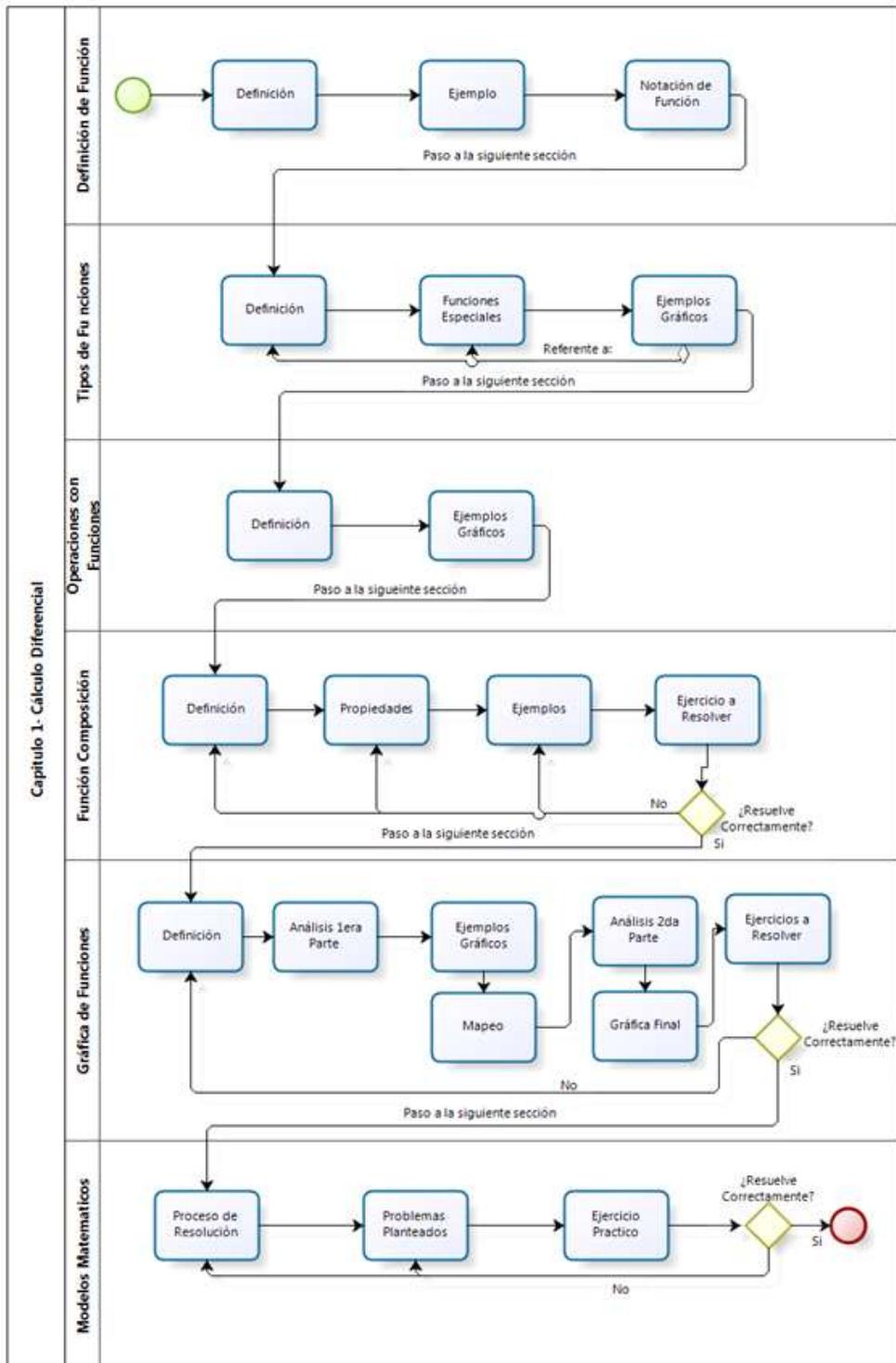


Figura 6.4-A Mapa de procesos del Cap. 1 de Cálculo Diferencial



UNIVERSIDAD DE CUENCA

Se ha diseñado de esta manera para un mejor entendimiento de los alumnos, la dirección de las flechas indica la secuencia de las actividades que a continuación se detallan:

- **Definición de Función.**- Explica el concepto de función como un objeto matemático y se refuerza la teoría con una ilustración y ejemplo del tema. También involucra la notación de funciones, útil para el entendimiento de los siguientes temas.
- **Tipos de Funciones.**- Detalla las funciones sobreyectiva, biyectiva e inyectiva. Además, se explican los tipos de funciones especiales con ejemplos gráficos de cada una.
- **Operaciones con Funciones.**- En esta sección se explican las operaciones de suma, resta, producto y división de funciones, se complementa el tema con las respectivas gráficas de ejemplo.
- **Función Composición.**- Está constituida por la definición, propiedades, ejemplos y un ejercicio práctico donde el alumno evalúe su conocimiento.
- **Gráfica de una función.**- Está compuesta por la definición y un ejemplo para reforzar el tema.
- **Funciones como modelos matemáticos.**- Consiste en definiciones y ejercicios prácticos donde se deben aplicar las funciones las funciones y ecuaciones aprendidas previamente.

Manteniendo el mismo esquema, en el siguiente punto se muestra el mapa de procesos del Capítulo 2 y su proceso de creación.

6.4.2 Mapa de procesos del Capítulo 2

Al igual que para el **Capítulo 1**, a continuación en la **Figura 6.4-B** se indica el mapa de procesos correspondientes al **Capítulo 2** de la materia *Cálculo Diferencial* el cuál está dividido en 6 partes:

- Definición de Límite
- Teoremas del Límite de Funciones
- Límites y Asíntotas
- Continuidad de Funciones
- Definiciones de Continuidad
- Continuidad de Funciones Trigonométricas

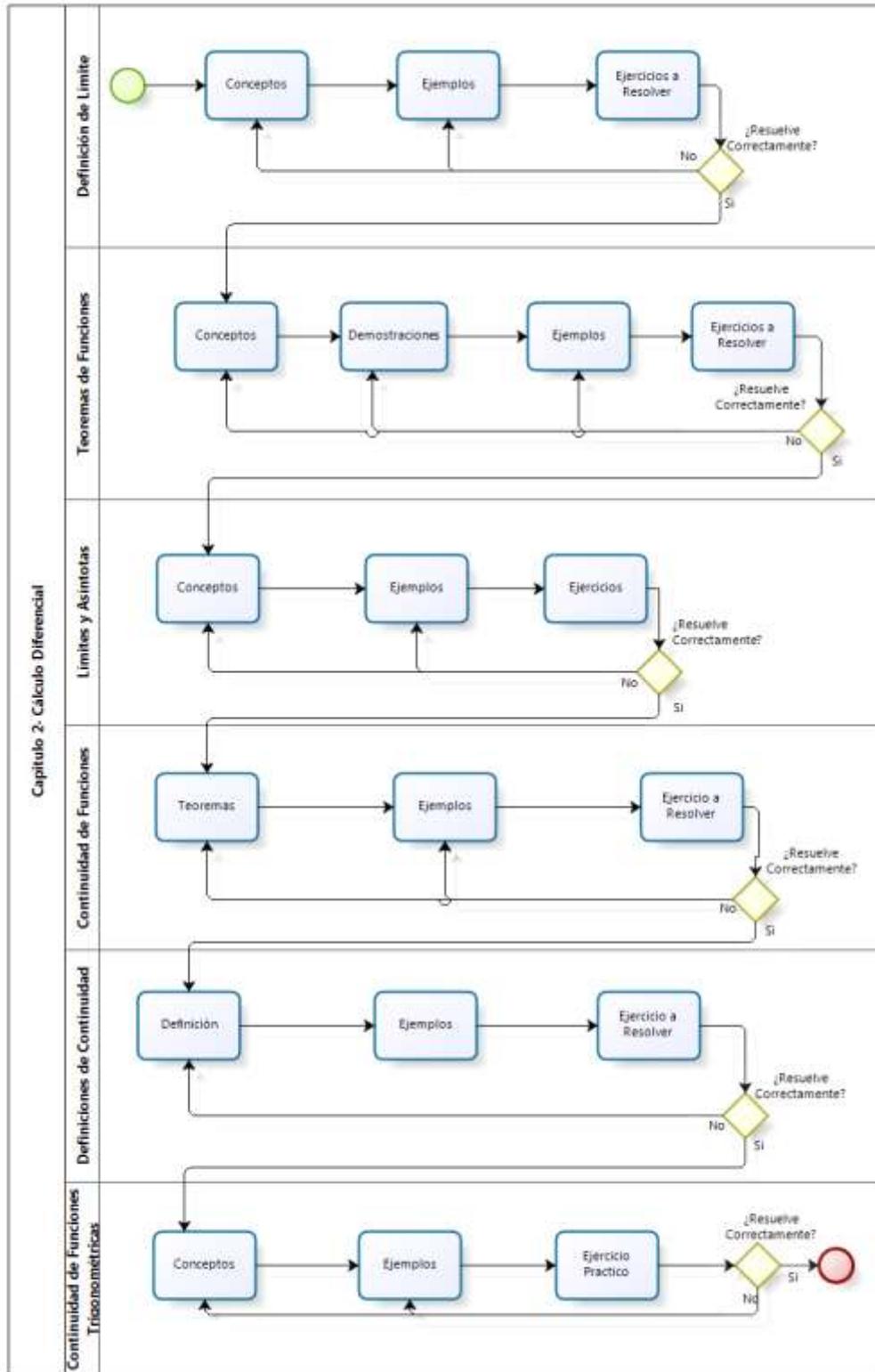


Figura 6.4-B Mapa de procesos del Cap. 2 de Cálculo Diferencial



UNIVERSIDAD DE CUENCA

Se ha diseñado de esta manera para un mejor entendimiento de los alumnos, la dirección de las flechas indica la secuencia de las actividades que a continuación se detallan:

- **Definición de Límite.-** En esta sección se explica el límite partiendo de una introducción gráfica, luego se procede a una definición formal y se refuerza con ejemplos ilustrativo.
- **Teoremas del Límite de Funciones.-** Se presenta una definición formal de cada teorema de límites de una función, demostraciones en los primeros casos y un ejemplo práctico.
- **Límites y Asíntotas.-** Detalla el procedimiento para obtener cada límite y asíntota, estos conceptos serán reforzados con ejemplos prácticos.
- **Continuidad de Funciones.-** Se presentan los teoremas con una definición formal, demostración y un ejemplo práctico de cada uno.
- **Definiciones de Continuidad.-** En esta sección se explica el concepto de continuidad en diferentes intervalos y se refuerza con un ejemplo práctico.
- **Continuidad de Funciones Trigonométricas.-** Detalla los teoremas a través de su demostración y se refuerza con un ejemplo práctico de cada teorema.

De esta manera, se ha detallado el conjunto de procesos necesarios para cumplir con todas las actividades planificadas en la enseñanza de los dos primeros capítulos de la materia piloto de "*Cálculo Diferencial*". Luego de haber creado estos mapas de procesos, se cuenta con la estructura completa de las actividades para dar paso al diseño de los objetos de aprendizaje que se trata en el siguiente punto.

6.5 DISEÑO DE LOS OBJETOS DE APRENDIZAJE

Para el prototipo se han distribuido los diferentes temas de cada capítulo en una cierto número de objetos de aprendizaje (ODA), considerando el tiempo que le toma al docente enseñar el tema y la complejidad del mismo, además para escoger la cantidad de ODA necesarios, se conto con la asesoría de la Ing. Magali Mejía (Docente de la Facultad de Ingeniería de la Universidad de Cuenca y especializada en la creación de ODA). Los objetos de aprendizaje contarán con los elementos necesarios para cada una de las actividades del contenido detalladas previamente.

El **primer objeto** de aprendizaje esta compuesto por las siguientes secciones del **Capítulo 1:**

- Definición de Función
- Tipos de Función
- Operaciones con Funciones



UNIVERSIDAD DE CUENCA

Como se observó en el diagrama de procesos de la **Figura 6.4-A**, existen un cierto número de actividades que se deben desarrollar para cada sección, las cuales requieren de algunos elementos multimedia para su explicación y entendimiento como: texto, imágenes, animaciones, videos, etc.

El detalle del primer objeto de aprendizaje lo podemos observar gráficamente en la **Figura 6.5-A**.

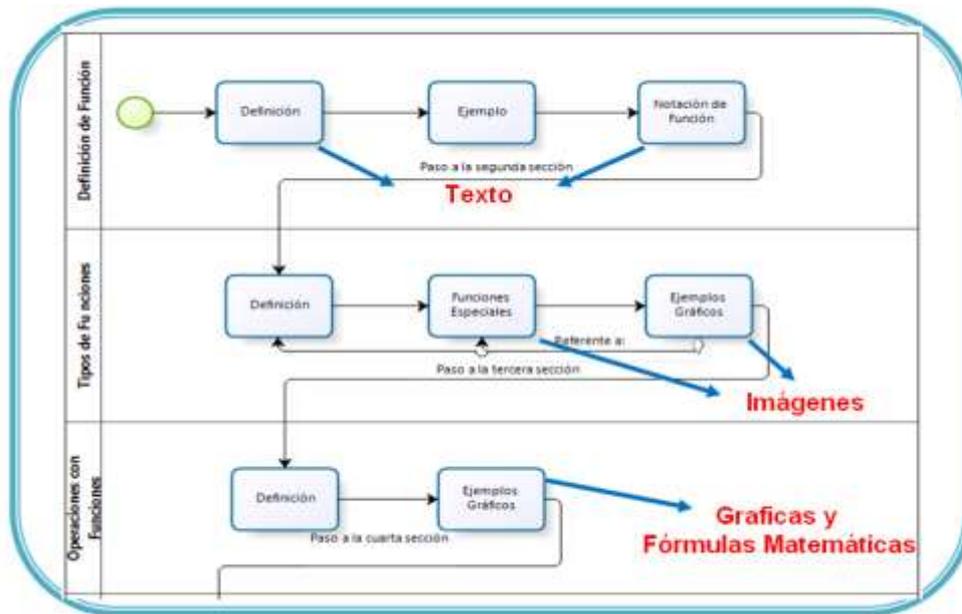


Figura 6.5-A Objeto de Aprendizaje 1 (Cap. 1 – Parte 1)

El **segundo objeto** de aprendizaje esta compuesto por las siguientes secciones del **Capítulo 1**:

- Función Composición
- Gráfica de Funciones
- Modelos Matemáticos con Funciones

El detalle del segundo objeto de aprendizaje lo podemos observar gráficamente en la **Figura 6.5-B**.

UNIVERSIDAD DE CUENCA

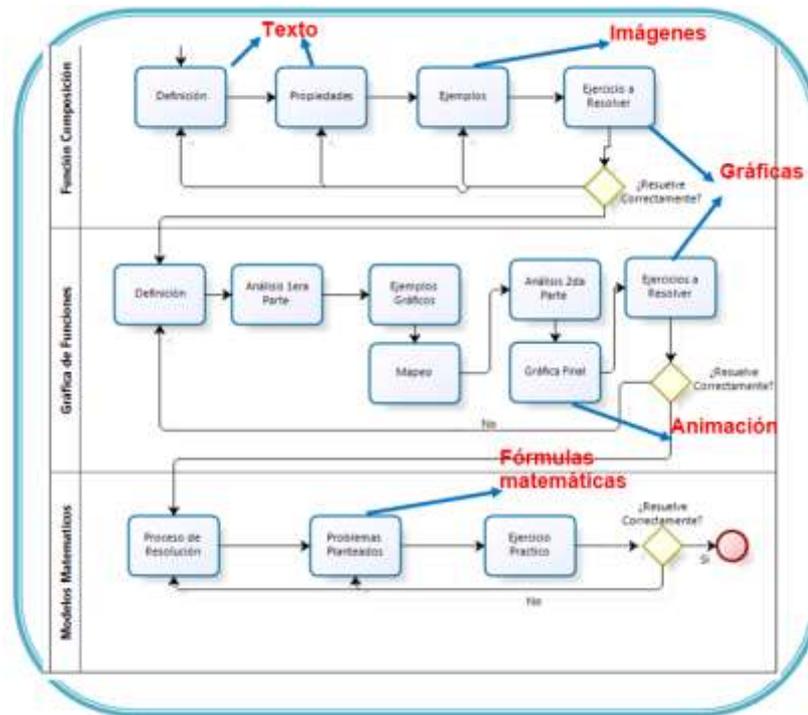


Figura 6.5-B Objeto de Aprendizaje 2 (Cap. 1 – Parte 2)

Como se ha observado en la **Figura 6.5-A** y en la **Figura 6.5-B**, se han determinado dos **ODA** para contener el **Capítulo 1**. El alumno podrá ir controlando su grado de conocimiento mediante la auto-evaluación, que para el caso del **ODA 1** existirá una sola al final de la última sección; mientras que para el **ODA 2** se incluirán pequeñas evaluaciones al final de cada sección como parte de los ejercicios prácticos a resolver.

El **tercer objeto** de aprendizaje esta compuesto por las siguientes secciones del **Capítulo 2**:

- Definición de Límite
- Teoremas del Límite de Funciones
- Límites y Asíntotas

El detalle del tercer objeto de aprendizaje lo podemos observar gráficamente en la **Figura 6.5-C**.

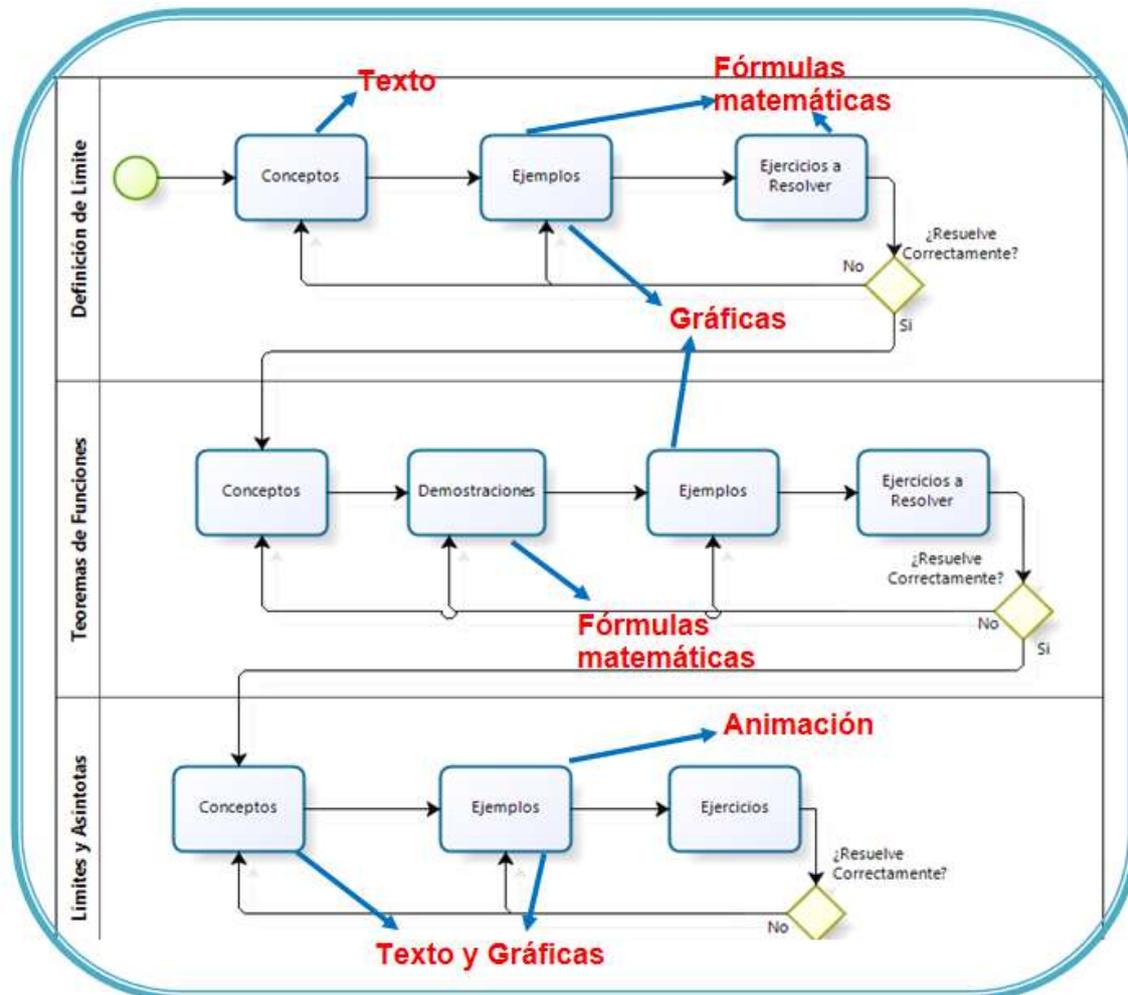


Figura 6.5-C Objeto de Aprendizaje 3 (Cap. 2 – Parte 1)

El **cuarto objeto** de aprendizaje está compuesto por las siguientes secciones del **Capítulo 2**:

- Continuidad de una Función
- Definición de Continuidad
- Continuidad de Funciones Trigonométricas

El detalle del cuarto objeto de aprendizaje lo podemos observar gráficamente en la **Figura 6.5-D**.

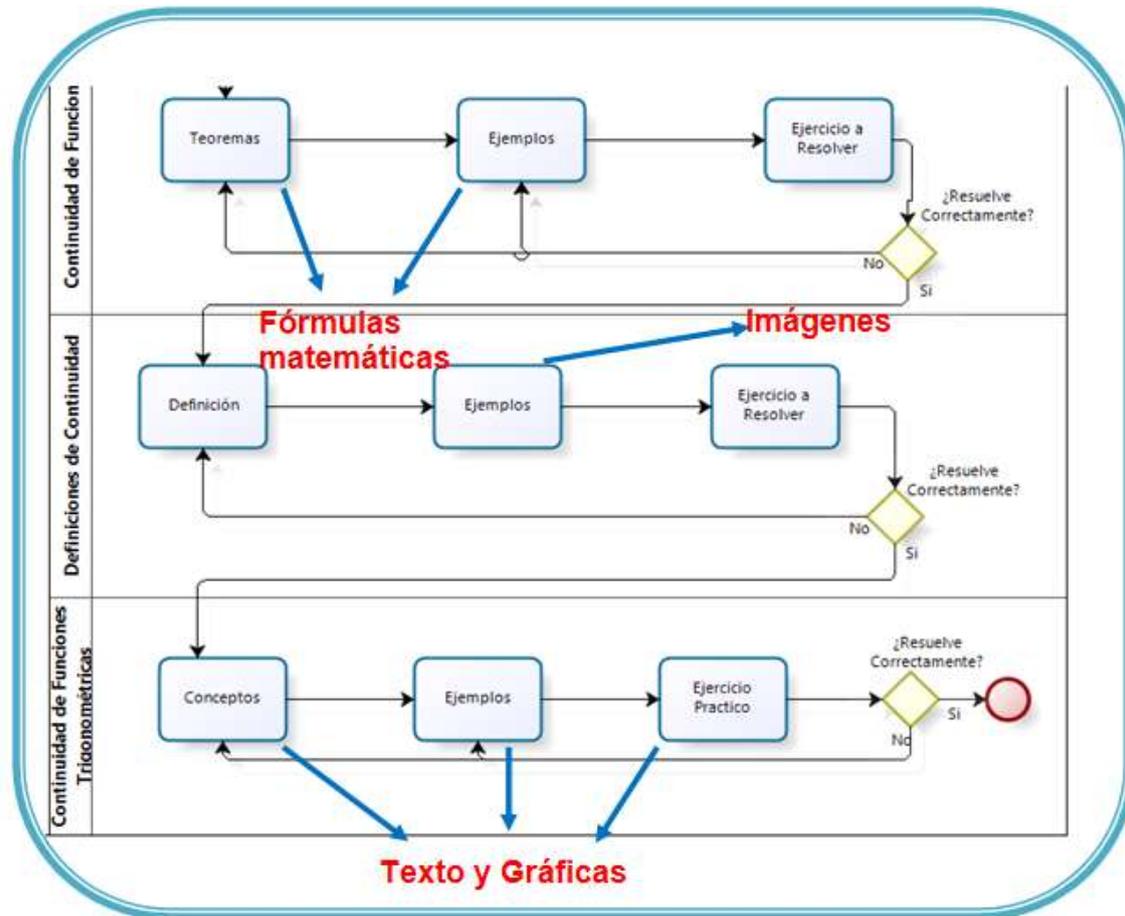


Figura 6.5-D Objeto de Aprendizaje 4 (Cap. 2 – Parte 2)

Ahora, para crear cada uno de los objetos de aprendizaje con sus respectivos elementos multimedia, se han utilizado diferentes herramientas, las cuales se detallan en el siguiente punto.

6.6 HERRAMIENTAS UTILIZADAS

En esta sección se presentan algunos conceptos generales de las herramientas utilizadas para la generación de los objetos de aprendizaje.

6.6.1 eXeLearning

Es una herramienta de autor²⁰ de código abierto para ayudar a los docentes en la creación y publicación de contenidos Web. Los recursos elaborados

²⁰Tipo de aplicaciones que permiten a sus usuarios crear sus propios proyectos multimedia con poca o nada de programación. Estas aplicaciones suelen generar los ejecutables para que los proyectos puedan ser vistos en diferentes computadoras (Priego-Montilla, 2011)



UNIVERSIDAD DE CUENCA

con **eXe** pueden exportarse en diferentes formatos: *IMS*, *SCORM 1.2*, etc. (CORE Education, 2010). Para esta tesis se ha utilizado el formato **SCORM 1.2**, ya que el material educativo de este tipo es totalmente compatible con la plataforma *Moodle* con la que se trabaja en esta tesis. Un paquete **SCORM** no solo permite que el alumno visualice los contenidos, sino la posibilidad de interactuar con ellos, además, que el **SCORM en Moodle** permite evaluar lo siguiente (Salvador Aznar, 2010):

- La cantidad de secciones que el alumno ha revisado.
- La cantidad de exámenes SCORM que el alumno ha realizado correctamente.

6.6.1.1 Estructura eXeLearning

El entorno de edición de esta herramienta se puede observar en la **Figura 6.6-Ay** su detalle se presenta a continuación:

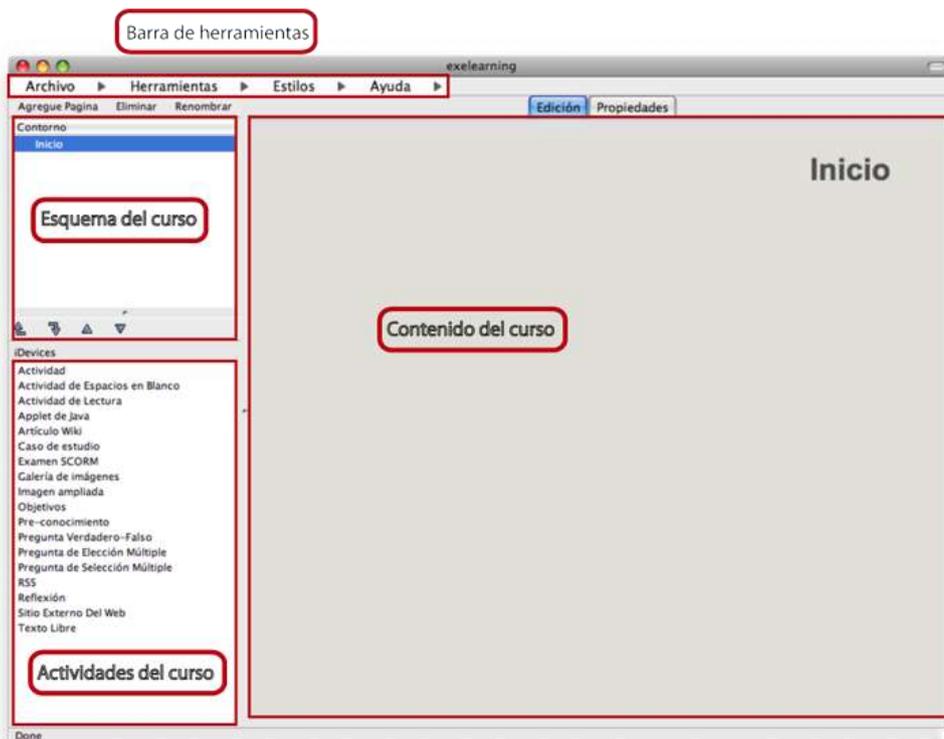


Figura 6.6-A Estructura de la herramienta ExeLearning

Detalle de la estructura indicada en la **Figura 6.6-A**

- **Esquema del curso.**- Permite crear una estructura jerárquica denominada árbol de contenidos, lo cual ayuda a una mejor organización del contenido.



UNIVERSIDAD DE CUENCA

- **Actividades del curso.-** Es un árbol de contenidos, que se encuentra constituido por elementos denominados *iDevices*.
- **Contenido del curso.-** En esta sección, se pueden editar los *iDevices* elegidos y agregar la información referente a la asignatura.

6.6.1.2 Principales elementos de eXeLearning

En la **Figura 6.6-B** se indican los principales elementos que se pueden utilizar para la creación de una actividad que haga parte del contenido.

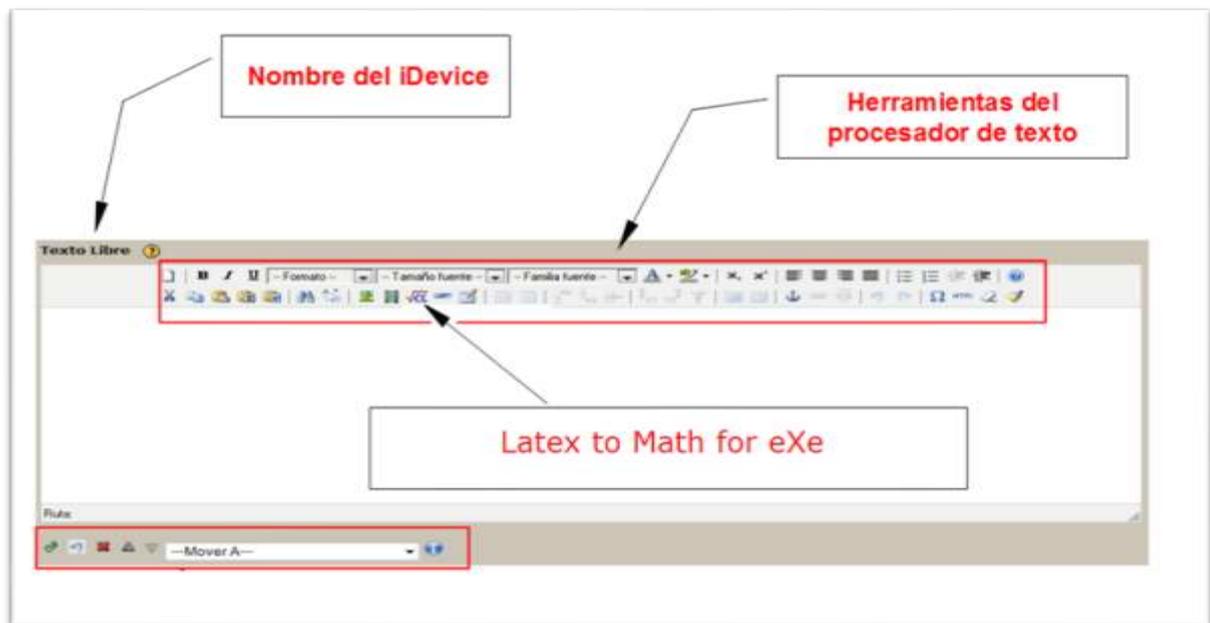


Figura 6.6-B Principales elementos de la herramienta ExeLearning

A continuación se presenta una breve descripción de estos elementos:

- **Nombre del iDevice.-** Esta etiqueta puede ser modificada en la creación del *Contenido del curso*.
- **Las Herramientas del procesador de texto.-** Permiten cambiar la apariencia del texto, agregar imágenes.
- **Latex to Math for exe.-** Herramienta utilizada para la inserción de fórmulas matemáticas, este editor lo hace a través de comandos *Latex*²¹.

²¹LaTeX es un procesador de textos que está formado mayoritariamente por órdenes (macros) construidas a partir de comandos. Es ampliamente usado en el ámbito científico debido a su flexibilidad y potencia. (Gayo, 2012)



UNIVERSIDAD DE CUENCA

6.6.2 Geogebra

GeoGebra es un software interactivo de matemática que reúne dinámicamente geometría, álgebra y cálculo (Markus Hohenwarter y Judith Hohenwarter, 2009). En la creación de los contenidos se utiliza esta herramienta para graficar las diferentes funciones de los respectivos temas.

6.6.2.1 Estructura de Geogebra

Ofrece diferentes perspectivas para la creación y vista de los objetos matemáticos, permite adaptaciones automáticas cuando se produce algún tipo de cambio en cualquiera de las vistas facilitando la utilización de esta herramienta. También permite exportar los archivos en formatos .png, hoja dinámica HTML y .gif entre otros (Markus Hohenwarter y Judith Hohenwarter, 2009). A continuación, se puede observar la **Figura 6.6-C** las diferentes perspectivas mencionadas.

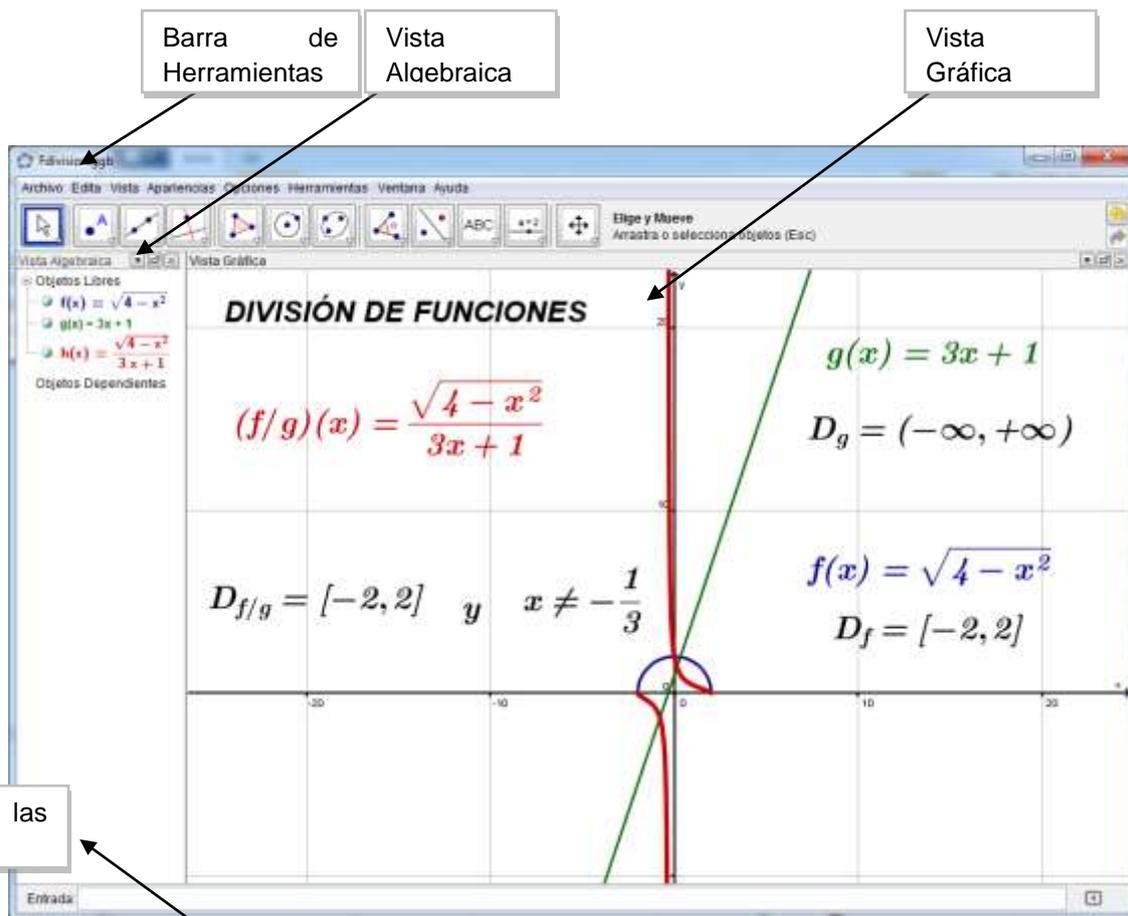


Figura 6.6-C Estructura de la herramienta Geogebra



UNIVERSIDAD DE CUENCA

El detalle de la estructura de la herramienta Geogebra se presenta en las líneas siguientes: (Markus Hohenwarter y Judith Hohenwarter, 2009)

- **Barra de Herramientas.-** Contiene varias cajas de herramientas de construcción útiles para desplazar y seleccionar objetos, además permite escoger tipos de rectas, vectores, circunferencias etc.
- **Vista Algebraica.-** Lugar donde se pueden insertar directamente las expresiones algebraicas y objetos matemáticos, los cuales aparecerán en la *Vista Gráfica*.
- **Vista Gráfica.-** Espacio en el cual se visualizan todas las construcciones geométricas, además de todos los objetos creados como: etiquetas, cuadrículas y demás elementos disponibles en el Geogebra.
- **Entrada para las fórmulas.-** Este programa ofrece una amplia gama de comandos que se pueden ingresar desde la Barra de Entrada, con la flecha que se ve en el vértice derecho inferior se despliegan muchas fórmulas e información sobre la sintaxis que se debe utilizar.

6.6.3 Illustrator

Herramienta utilizada para el diseño de imágenes, en la **Figura 6.6-D** se indica su estructura y los principales elementos a ser utilizados.

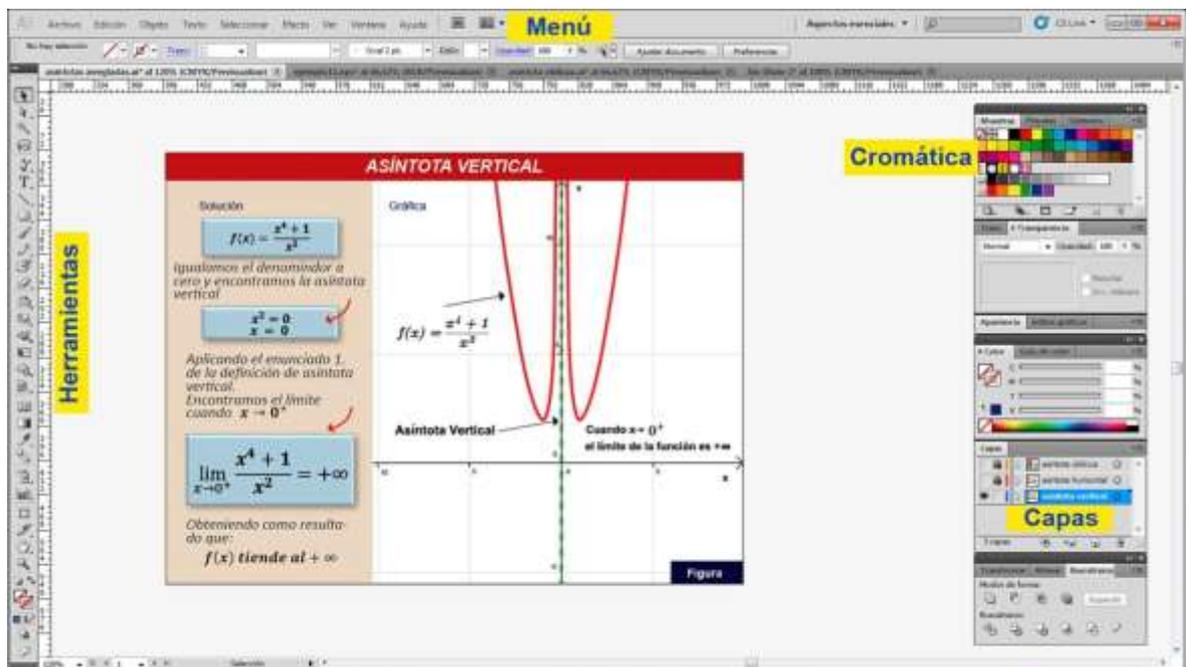


Figura 6.6-D Estructura de la herramienta Illustrator



UNIVERSIDAD DE CUENCA

Menú.- Las acciones principales consisten en importar el gráfico desde GeoGebra con la extensión .eps para redibujar con las herramientas de Adobe Illustrator. Luego de haber redibujado el archivo se exporta y se graba con las características para Web y dispositivos desde el mismo menú. (.gif o .png)

En esta barra se encuentra también una serie de filtros que sirven para dar efectos como sombreados o volumen a ciertas áreas con la finalidad de resaltar dicho elemento.

Herramientas.- Aquí encontramos las herramientas de edición, dibujo y tratamiento de formas, selección de elementos, edición de textos.

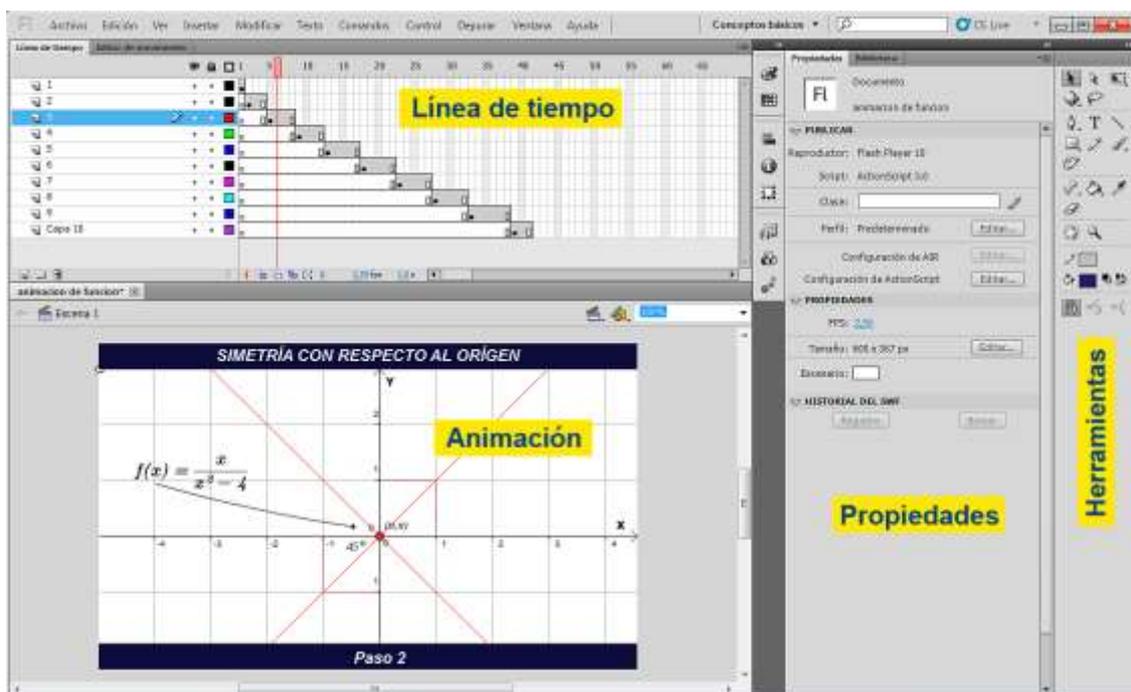
También existen herramientas para manipular las formas, escalándolas, o distorsionándolas, según sea el caso.

Cromática.- Esta paleta de colores podemos utilizar los colores que están propuestos por defecto y también nos da la posibilidad de crear nuestros propios colores a convenir. Además podemos personalizar dicha paleta y guardar para ser utilizado en caso de requerir nuevamente.

Los colores utilizamos en los fondos, en las líneas y también en los textos.

6.6.4 Flash

Herramienta utilizada para las animaciones que serán incluidas en cada sección. En la **Figura 6.6-E** se observa la estructura general de Flash, colocando mayor énfasis en las opciones que han sido utilizadas para este proyecto.





UNIVERSIDAD DE CUENCA

Figura 6.6-E Estructura de la herramienta Flash

El detalle de la estructura de la herramienta Flash se presenta a continuación:

Línea de tiempo.- En esta ventana se visualiza las diferentes capas que contienen los dibujos listos a ser animados y en la secuencia requerida previamente.

La línea de tiempo nos permite asignar a cada capa o dibujo una duración de tiempo antes de continuar con la visualización del siguiente dibujo, existen imágenes que requieren una mayor visualización, por lo tanto se le asigna más tiempo de exposición. En general con la línea de tiempo se controla la duración de toda la animación y la velocidad de reproducción.

Animación.- Aquí se editan y se visualizan todos los contenidos de las capas. Es importante que se organice a cada dibujo con su respectiva capa de acuerdo a la escena que se desee visualizar. Hay que recordar que el número de capas depende de la cantidad de pasos que tenga la animación. Para ejecutar una previa de la animación basta presionar *enter* y luego se podrán realizar los ajustes necesarios.

Propiedades.- Con esta sección se puede hacer ajustes a los diferentes gráficos, como por ejemplo al tamaño de la ventana de animación, el formato de edición como html, flash, gif, jpg etc.

Herramientas.- Las herramientas de flash tienen las mismas características que las de Adobe Ilustrador. Su uso es mínimo debido a que los gráficos vienen listos desde Adobe Ilustrador.

6.7 CREACIÓN DE LOS CONTENIDOS

Para la creación de los contenidos se basó en el esquema y diseño previo de los mismos. La herramienta utilizada para la construcción de los objetos de aprendizaje es el eXeLearning detallada en la **Sección 6.6.1**. Aquí cada **ODA** cuenta con un número de páginas definidas de acuerdo al orden jerárquico de aprendizaje de la materia.

A continuación, se presentan algunas ilustraciones que permiten apreciar como fueron creados los contenidos de la materia piloto *Cálculo Diferencial*, con sus respectivos objetos de aprendizaje y elementos multimedia.

6.7.1 ODA 1 (Capítulo 1 – Parte 1)

- **Descripción de los objetivos y pre-conocimiento.-** Al inicio de cada ODA se debe especificar los objetivos de dicha sección al igual que los requisitos de conocimiento previo. De esta manera el alumno puede conocer lo necesario para iniciar su estudio y a donde se desea llegar



UNIVERSIDAD DE CUENCA

con el aprendizaje. Un ejemplo de objetivos y pre-conocimiento se observa en la **Figura 6.7-A**.

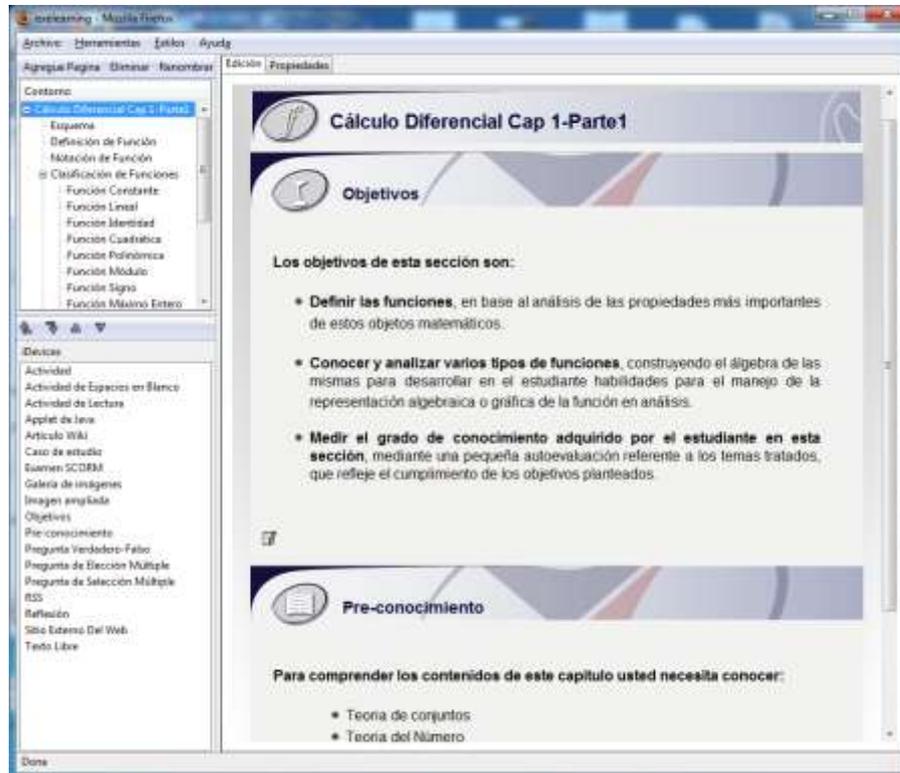


Figura 6.7-A Objetivos y Pre-Conocimiento de un ODA

- **Esquema de cada sección.-** En cada objeto de aprendizaje se debe colocar un esquema indique al alumno un resumen general de todo lo que se estudiará en la sección. Un ejemplo de esquema se presenta en la **Figura 6.7-B**.



Figura 6.7-B Esquema de cada sección

- **Conceptos y Notaciones.-** En las diferentes secciones se ha incluido texto plano y notaciones matemáticas que permitan al alumno conocer la parte teórica de la materia para luego ser aplicada a los ejercicios posteriores. Un ejemplo de conceptos y notaciones matemáticas se observa en la **Figura 6.7-C**.

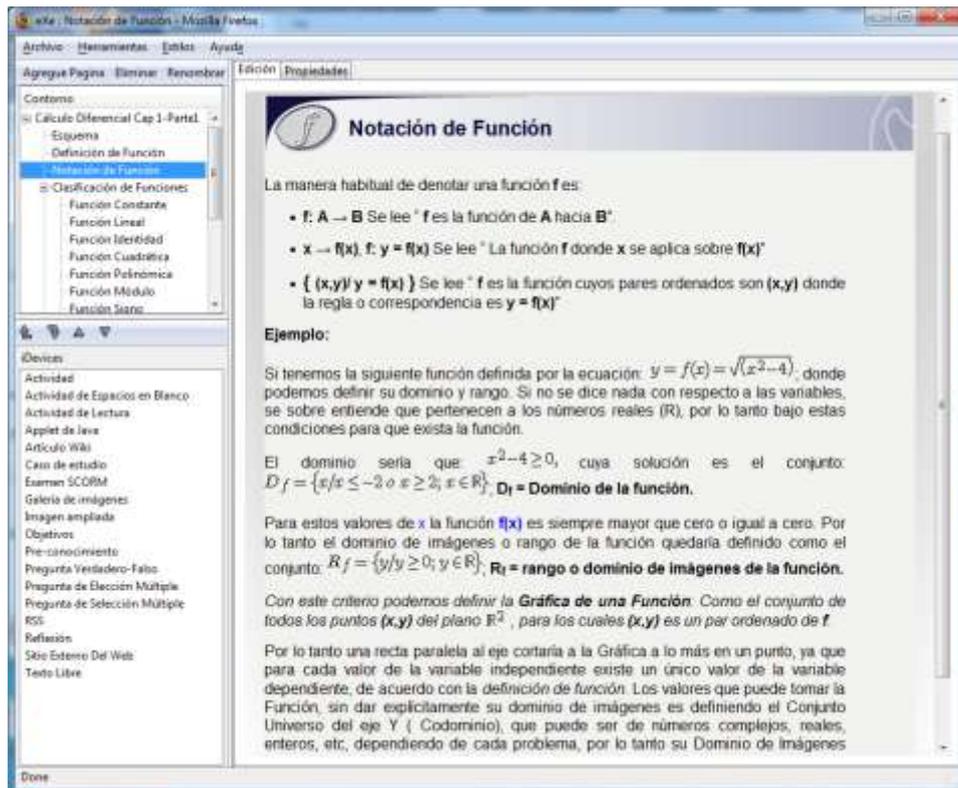


Figura 6.7-C Conceptos y notaciones matemáticas

- **Gráficas y Ejemplos.-** Para complementar las definiciones en cada uno de los temas, se ha incluido imágenes que muestran de manera gráfica los conceptos y los ejercicios de aplicación. Esto se lo puede observar en la **Figura 6.7-D**.

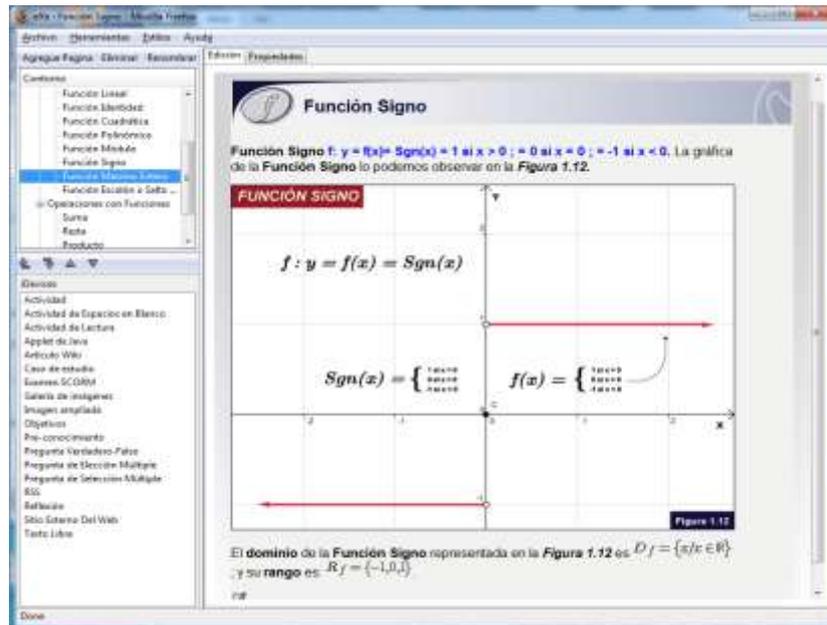


Figura 6.7-D Conceptos y notaciones matemáticas

- **Inclusión de tablas.** - Se ha visto necesario incluir algunas tablas para ciertos conceptos y ejercicios que así lo requieran. En la **Figura 6.7-E** se indica un ejemplo de inclusión de tablas.

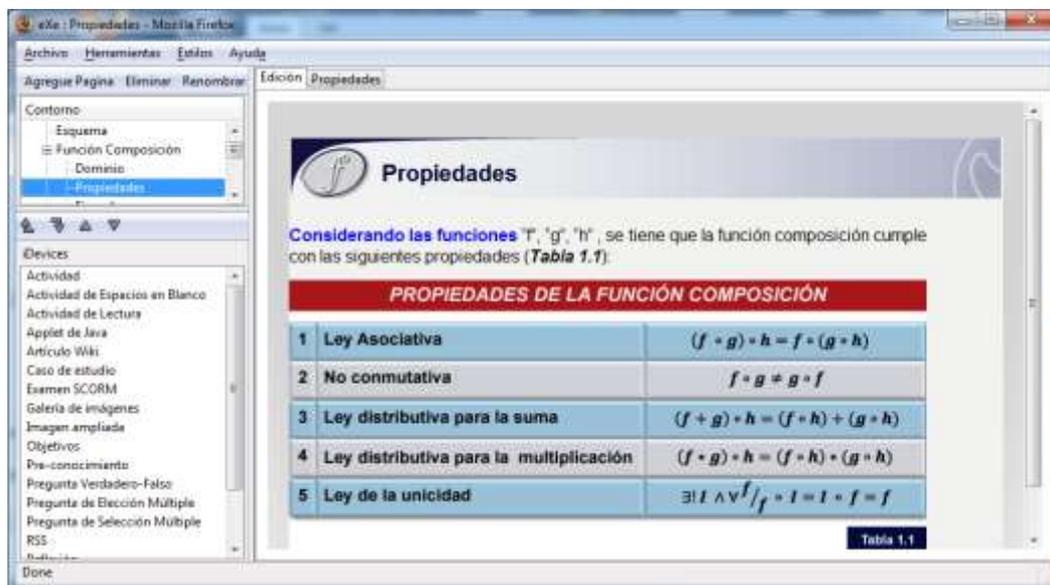


Figura 6.7-E Inclusión de Tablas

- **Ejemplos desarrollados paso a paso.** - En el caso de los ejemplos, es importante explicar su desarrollo paso a paso, de esta manera el estudiante observa la aplicación de la parte teórica y comprende de una manera general como llegar al resultado final de un ejercicio dado. En la



UNIVERSIDAD DE CUENCA

Figura 6.7-F se muestra la solución de un ejercicio con la inclusión de links hacia las respectivas gráficas.

The screenshot shows a web browser window with a sidebar on the left containing a navigation menu. The main content area displays a math problem and its solution. The problem asks for the composition of functions $f(x) = \sqrt{x}$ and $g(x) = x^2 - 1$. The solution shows the calculation of $(f \circ f)(x)$, $(g \circ g)(x)$, $(f \circ g)(x)$, and $(g \circ f)(x)$, with links to figures 1.22, 1.23, 1.24, and 1.25. A callout box points to the link for Figure 1.24.

Ejemplos

PROBLEMA 1
Si $f(x) = \sqrt{x}$ y $g(x) = x^2 - 1$.

Calcular:

a) $(f \circ f)(x)$, $(g \circ g)(x)$, $(f \circ g)(x)$ y $(g \circ f)(x)$
b) Calcular los dominios en base de g y f

NOTA: Para observar las imágenes haga clic sobre el texto de la Figura a continuación de la [→](#)

Solución:

a) **Aplicando las propiedades de la función composición, obtenemos:**
 $(f \circ f)(x) = f(f(x)) = f(\sqrt{x}) = \sqrt{\sqrt{x}} = \sqrt[4]{x} \rightarrow$ [Figura 1.22](#)
 $(g \circ g)(x) = g(g(x)) = g(x^2 - 1) = (x^2 - 1)^2 - 1 = x^4 - 2x^2 \rightarrow$ [Figura 1.23](#)
 $(f \circ g)(x) = f(g(x)) = f(x^2 - 1) = \sqrt{x^2 - 1} \rightarrow$ [Figura 1.24](#)
 $(g \circ f)(x) = g(f(x)) = g(\sqrt{x}) = (\sqrt{x})^2 - 1 = x - 1 \rightarrow$ [Figura 1.25](#)

b) **Aplicando el concepto de dominio que se ilustró anteriormente**
 $D_{(g \circ f)} = D_g \circ f = \{x / x \in (D_g \circ f)(x) \in D_g\}$, obtenemos:
Si $D_f = \{x / x \geq 0\} = [0, \infty)$ y $D_g = \{x / -\infty < x < +\infty\} = (-\infty, +\infty)$
 $D_{f \circ f} = \{x / x \in D_f \wedge f(x) \in D_f\}$

Link hacia la gráfica

Figura 6.7-F Ejemplos desarrollados paso a paso

En la Figura 6.7-G se indica la imagen de acuerdo al link seleccionado por el alumno.

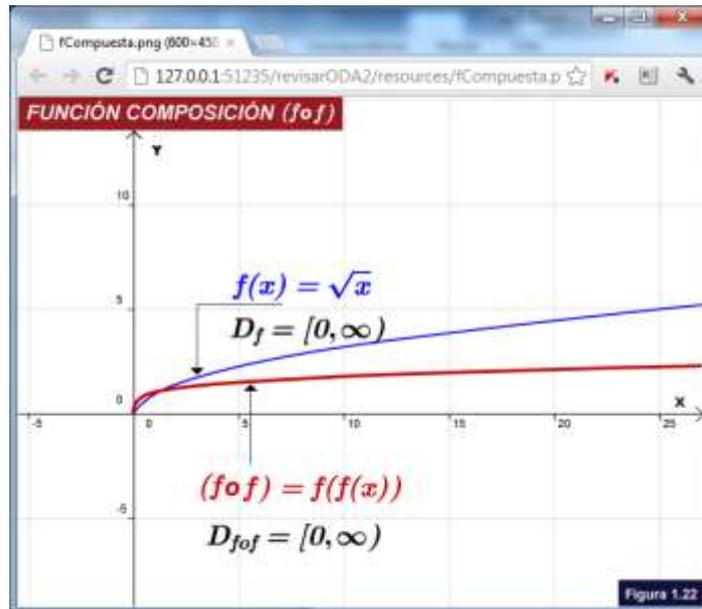
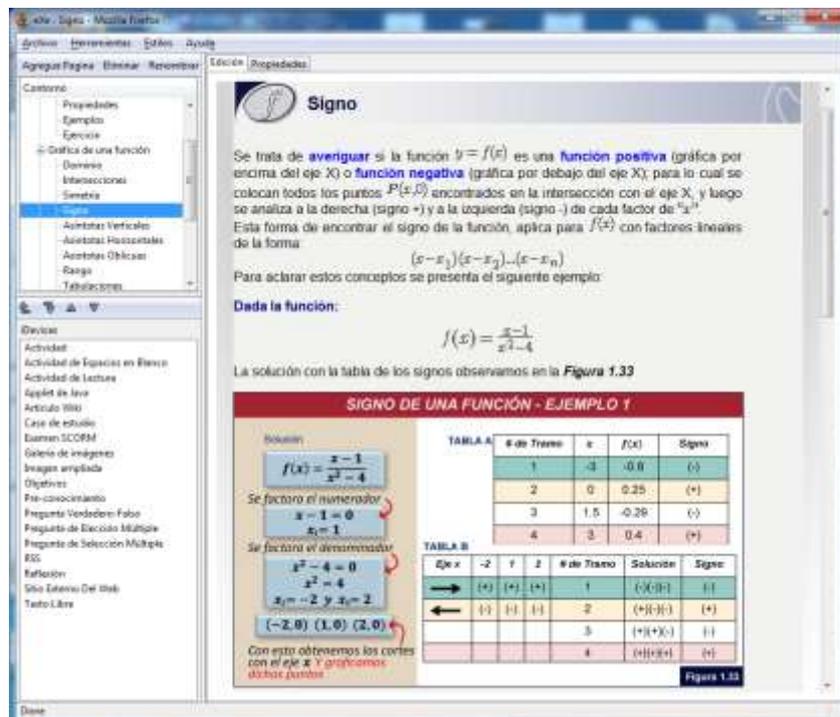


Figura 6.7-G Imagen perteneciente a un link seleccionado

- Solución de **ejercicios de una manera más dinámica y explicativa.**- Esto aplica principalmente para ejercicios complejos y que comprenden varios pasos para llegar a su solución y gráficas para visualizar resultados. Un ejemplo se observa en la **Figura 6.7-Hy** en la **Figura 6.7-I.**



Signo

Se trata de **averiguar** si la función $y = f(x)$ es una **función positiva** (gráfica por encima del eje X) o **función negativa** (gráfica por debajo del eje X); para lo cual se colocan todos los puntos $P(x, 0)$ encontrados en la intersección con el eje X , y luego se analiza a la derecha (signo $+$) y a la izquierda (signo $-$) de cada factor de $f(x)$. Esta forma de encontrar el signo de la función, aplica para $f(x)$ con factores lineales de la forma $(x-x_1)(x-x_2)...(x-x_n)$.

Para aclarar estos conceptos se presenta el siguiente ejemplo:

Dada la función:

$$f(x) = \frac{x-1}{x^2-4}$$

La solución con la tabla de los signos observamos en la **Figura 1.33**

SIGNO DE UNA FUNCIÓN - EJEMPLO 1

Solución:

Se factora el numerador:
 $x - 1 = 0$
 $x_1 = 1$

Se factora el denominador:
 $x^2 - 4 = 0$
 $x_2 = -2$ y $x_3 = 2$

Con esto obtenemos los cortes con el eje X y graficamos dicho punto.

TABLA A

# de Tramo	x	$f(x)$	Signo
1	-3	-0.6	(-)
2	0	0.25	(+)
3	1.5	-0.26	(-)
4	3	0.4	(+)

TABLA B

Eje x	-2	1	2	# de Tramo	Solución	Signo
\rightarrow	(+)	(+)	(+)	1	$(-)(-)(-)$	(-)
\leftarrow	(-)	(-)	(-)	2	$(+)(-)(-)$	(+)
				3	$(+)(+)(-)$	(-)
				4	$(+)(+)(+)$	(+)

Figura 6.7-H Explicación de Ejercicios

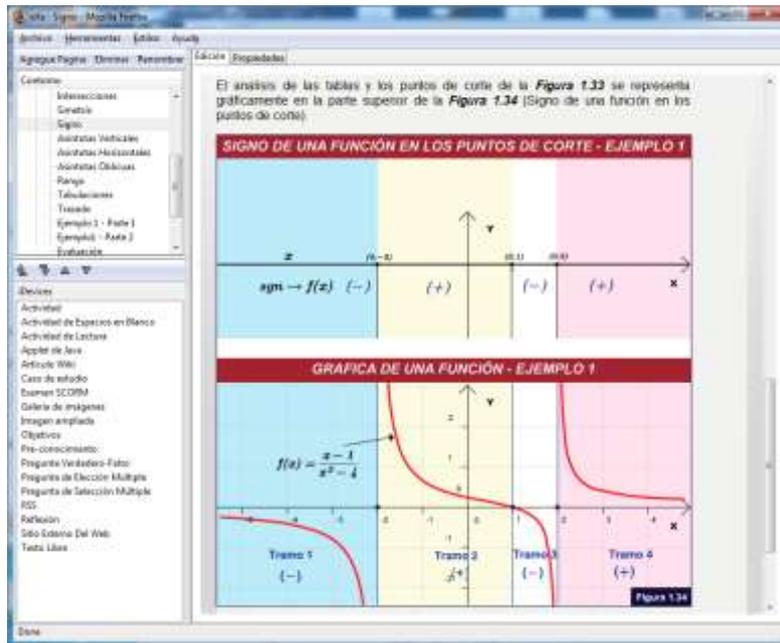


Figura 6.7-I Solución Gráfica de Ejercicios

- **Inclusión de animaciones.-** Se ha visto necesario incluir animaciones en los diferentes objetos de aprendizaje, con el objetivo de entregar al estudiante un contenido llamativo que complemente su estudio. Esto aplica especialmente para ciertos temas en los que: el texto, las fórmulas y una gráfica no son suficiente para comprender lo que se desea explicar. En la **Figura 6.7-J** se indica un ejemplo de animación creado en un formato .Gif

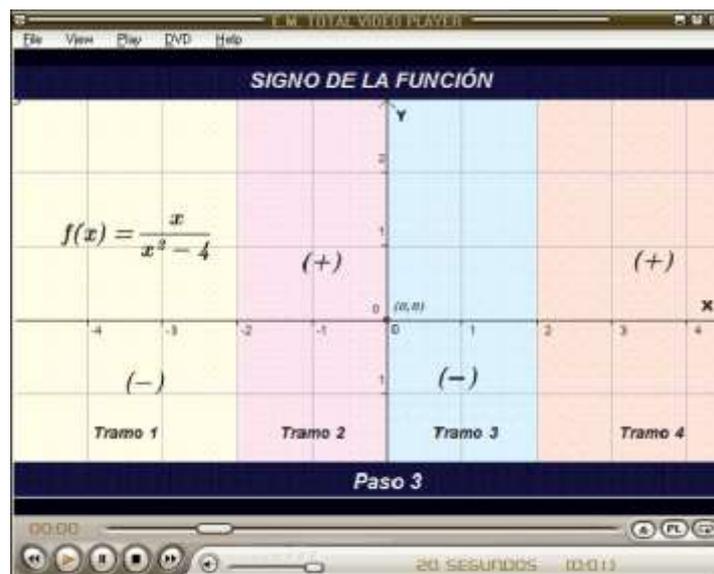


Figura 6.7-J Ejemplo de animación en formato .Gif



UNIVERSIDAD DE CUENCA

- **Autoevaluaciones con respuestas de selección múltiple**.- Como parte de los objetivos de cada sección y respetando las decisiones de diseño de los contenidos, se debe incluir una autoevaluación, de manera que el estudiante mida su grado de conocimiento en los diferentes temas aprendidos. Algunos ejemplos de autoevaluación, se pueden observar en la **Figura 6.7-K** y en la **Figura 6.7-L**.



Figura 6.7-K Modelo de autoevaluación – Texto

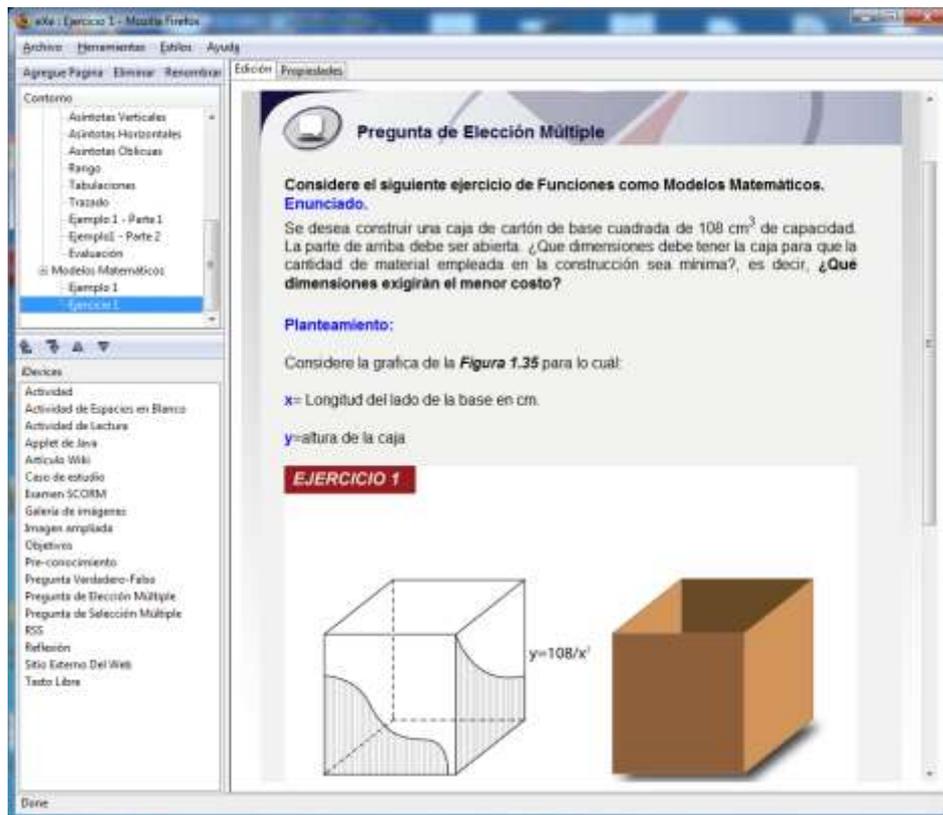


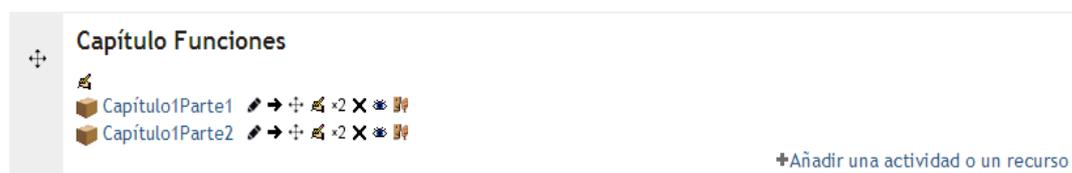
Figura 6.7-L Modelo de autoevaluación- Gráficas

6.8 IMPLEMENTACIÓN EN LA PLATAFORMA MOODLE

Una vez creados los contenidos en la herramienta ExeLearning se procede a subir en la plataforma Moodle dicho material. Para ello se deben seguir algunos pasos que se detallan a continuación:

- **Iniciar sesión y acceder al curso de Cálculo Diferencial**

En la página www.eviamnet.com, que es un dominio de prueba utilizado en esta tesis, el cual podrá ser remplazado por la dirección del sistema Moodle de la Universidad de Cuenca (evirtual.ucuenca.edu.ec), cuando se desee implementar esta tesis dentro del dominio de la Universidad. Entonces en el dominio de prueba se **inicia sesión** y luego de ello se **accede al curso** para generar los ajustes activando la edición del mismo. En la **Figura 6.8-A** se indica el acceso al curso de *Cálculo Diferencial*.





UNIVERSIDAD DE CUENCA

Figura 6.8-A Edición de un curso de Cálculo Diferencial en Moodle

- **Seleccionar y añadir una actividad o un recurso**

En la **Figura 6.8-B** se puede observar la lista de actividades que se pueden agregar a un curso, para el caso del proyecto se escogerá una **Paquete SCORM**, debido a que los contenidos creados en la herramienta ExeLearning fueron exportados de tipo *SCORM 1.2* entonces la nueva actividad a crear debe ser de este tipo.

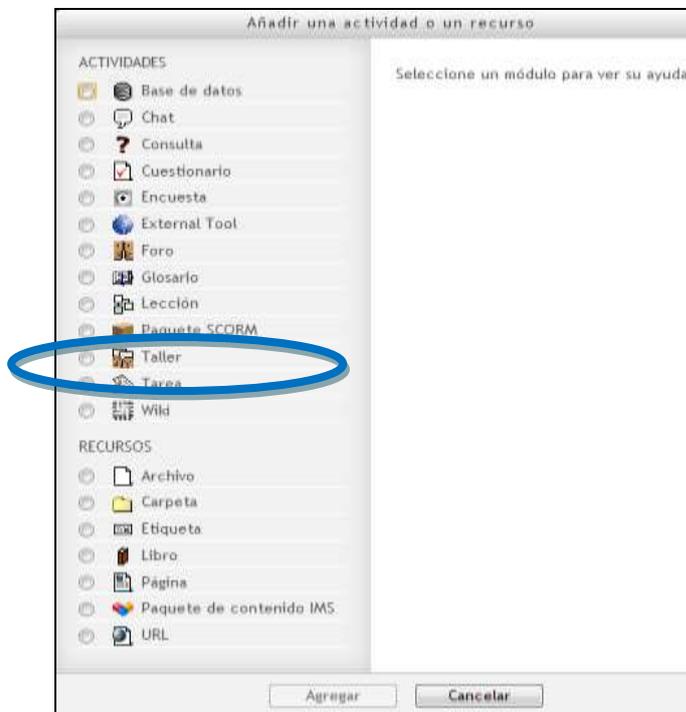


Figura 6.8-B Añadir una actividad o recurso en Moodle

- **Agregar un Paquete SCORM**

Luego de escoger el paquete SCORM se debe llenar la información necesaria para creación del mismo. Primero los datos generales como: nombre, descripción y ubicación del archivo .zip (ODA exportado desde ExeLearning). Este procedimiento se lo puede observar en la **Figura 6.8-C**.



UNIVERSIDAD DE CUENCA

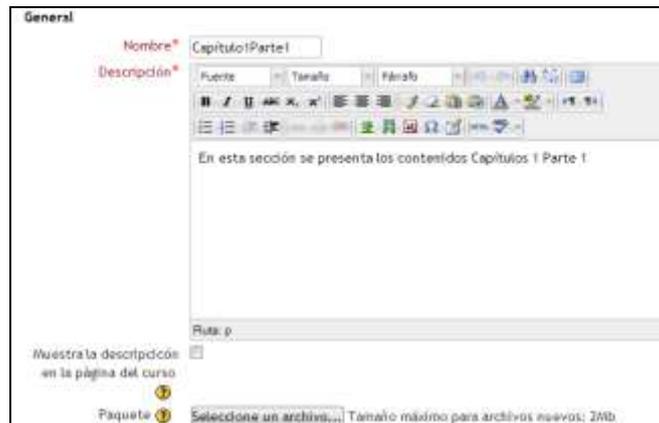


Figura 6.8-C Agregar un paquete SCORM dentro de un curso

La siguiente parte que se debe configurar es en la sección **Configuración de pantalla**. Aquí se debe indicar que los objetos aparezcan siempre en una ventana nueva y con el ancho y alto adecuado para dicha ventana. Se observa estos ajustes en la **Figura 6.8-D**.

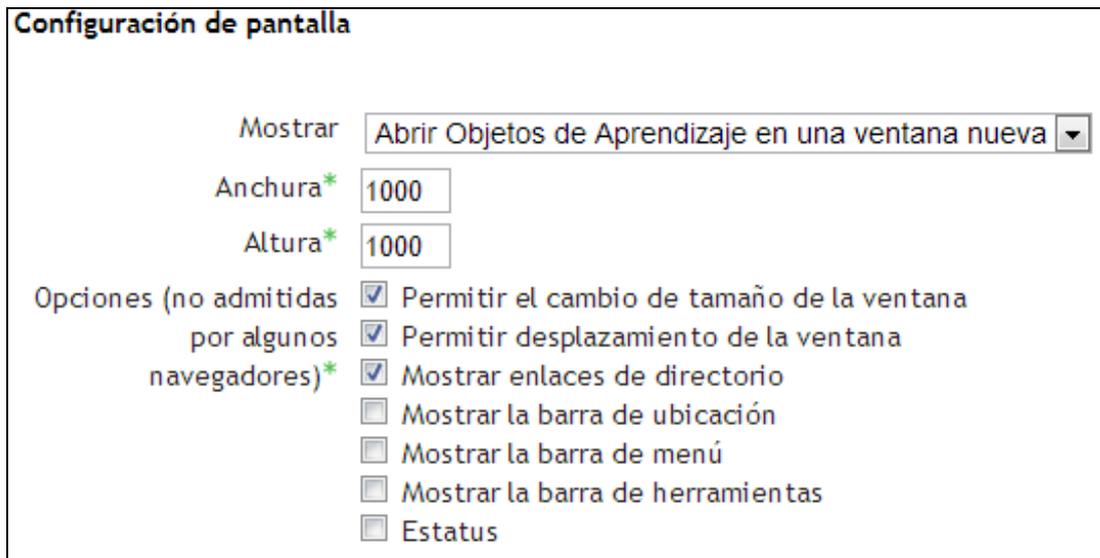


Figura 6.8-D Configuraciones adicionales para agregar un paquete SCORM

Por último se guardan los cambios y de esta manera se ha creado el paquete SCORM dentro del curso seleccionado.

- **Visualización del Paquete SCORM creado**



UNIVERSIDAD DE CUENCA

Luego de **Guardar cambios y regresar al curso**, la plataforma redirige al curso donde fue creado el paquete, en este caso **Cálculo Diferencial**. En la **Figura 6.8-E** se puede observar el **Paquete SCORM** correspondiente al **Capítulo1 - Parte1**.



Figura 6.8-E Visualización de los paquetes SCORM

- **Manipulación del Objeto SCORM**

Para ver el contenido del objeto SCORM, seleccionamos el paquete y posteriormente se muestra una ventana con cada una de las secciones que han sido creadas previamente en forma de árbol. Cada vez que el alumno ingresa a una actividad, en el árbol de contenidos ésta se marca con un check o visto, de esta manera se controla que el alumno siga con el orden de estudio de una manera jerárquica, como se observa en la **Figura 6.8-F**.

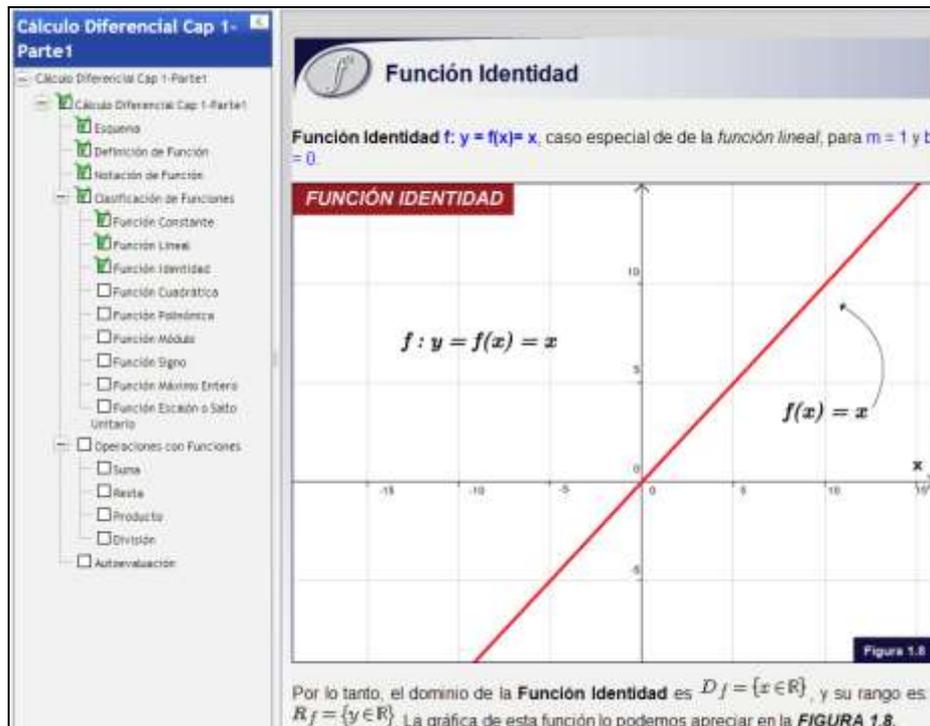


Figura 6.8-F Manipulación de un objeto SCORM (orden jerárquico)

En caso de que el alumno no siga el orden programado de los contenidos, el árbol presentará un ícono en forma de banana, esto se lo puede observar en la Figura 6.8-G.

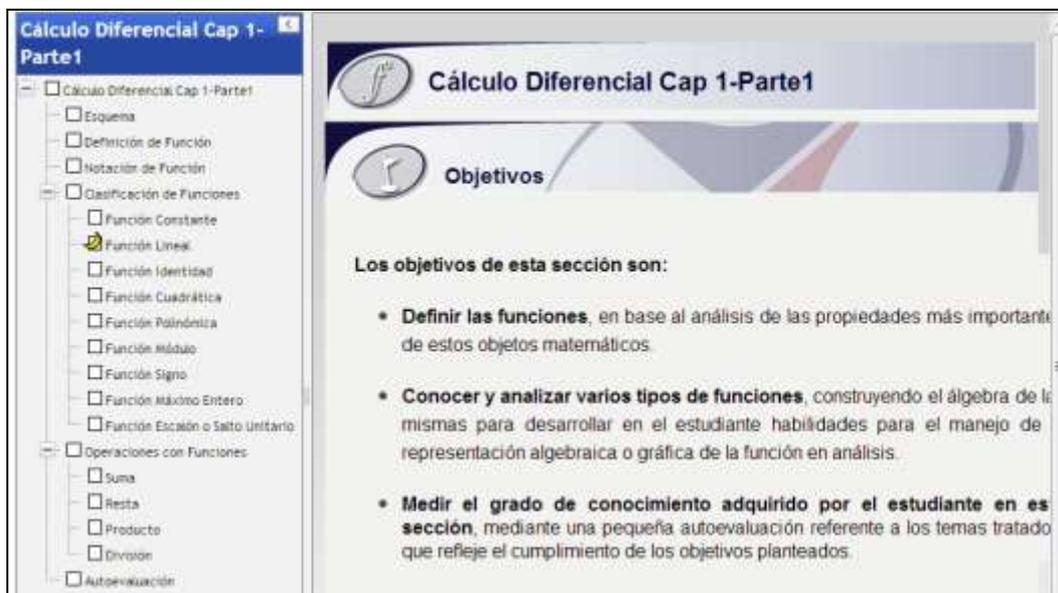


Figura 6.8-G Manipulación de un objeto SCORM (orden no jerárquico)



UNIVERSIDAD DE CUENCA

6.9 RESUMEN DEL CAPITULO

Finalmente, en este capítulo se han descrito el diseño de los contenidos, las herramientas para su creación y todo lo correspondiente a la implementación del material educativo de los capítulos uno y dos de la materia piloto *Cálculo Diferencial*. Para ello, se tomó en cuenta el análisis previo de todos los temas a tratar en el curso y luego en base a éste análisis se realizó el diseño de los contenidos dividiendo a cada uno de los capítulos en objetos de aprendizaje (ODA) dinámicos y fundados en el modelo de enseñanza constructivista, que tiene como principal objetivo entregar al alumno material didáctico, interactivo el cuál le permita crear y controlar su propio conocimiento. Además, para la creación de los contenidos se consideró la complejidad de los temas en cada uno de los capítulos y la cantidad de actividades a realizarse, esto fue muy importante ya que a través de dicho estudio se logró incluir el número de elementos multimedia necesarios para un correcto aprendizaje por parte del alumno. Como se indicó en la última sección de este capítulo los paquetes SCORM ya se encuentran disponibles en la plataforma Moodle para las pruebas necesarias y accesibles para la aplicación Móvil, cumpliendo de esta manera todos los objetivos planteados inicialmente.

CAPITULO 7:

“Conclusiones, Recomendaciones y Referencias.”



UNIVERSIDAD DE CUENCA



UNIVERSIDAD DE CUENCA

7.1 INTRODUCCIÓN

En este capítulo se presentan las conclusiones obtenidas durante el desarrollo del presente trabajo de investigación, al igual que algunas recomendaciones para futuras investigaciones sobre la implementación de la metodología constructivista con soporte en herramientas Web y dispositivos móviles para grupos numerosos de estudiantes.

7.2 CONCLUSIONES

- La metodología utilizada para desarrollar aplicaciones móviles es básicamente igual a aquella utilizada en el desarrollo de aplicaciones Web o de escritorio. La diferencia principal que se ha podido identificar es que, en la programación de aplicaciones móviles, resulta indispensable considerar las limitaciones del tamaño de la pantalla del dispositivo además de su restringida capacidad de procesamiento y almacenamiento al momento de diseñar la aplicación.
- En lo referente a los contenidos de aprendizaje generados para la materia piloto Cálculo Diferencial, se completaron de manera satisfactoria los dos capítulos contemplados dentro del alcance del presente trabajo de investigación. La inclusión de imágenes y animaciones dentro de los Objetos de Aprendizaje, facilitaron la explicación de los conceptos teóricos de la materia, permitiendo que los contenidos sean realmente muy completos y abundantes en cuanto a recursos educativos.
- Al finalizar la investigación sobre m-learning, se puede concluir que este tipo de educación es una alternativa con mucho potencial para sustituir el modelo tradicional de educación. Las herramientas Web han facilitado la difusión de contenidos educativos a los estudiantes, pero los mantiene atados a un computador; por lo que resulta importante apoyar e incentivar el desarrollo de proyectos educativos similares a este, dirigidos a crear una cultura de aprendizaje basado en herramientas tecnológicas móviles, que permitan a los estudiantes llevar consigo la educación a donde vayan, sin limitaciones de horarios.
- Se ha llegado a la conclusión de que, la metodología constructivista per se, no es completamente aplicable en la enseñanza de materias técnicas como Cálculo Diferencial, debido a la gran cantidad de conceptos abstractos que resultan difíciles de observar directamente por el estudiante. Se ha visto necesario aplicar una especie de híbrido entre el constructivismo y la educación tradicional, brindando a los estudiantes una guía con objetivos de aprendizaje claramente especificados, cuyos contenidos se encuentran jerárquicamente distribuidos, calendarizados y apuntan al cumplimiento de dichos objetivos, pero manteniendo la filosofía constructivista, es decir, centrados en el estudiante.



UNIVERSIDAD DE CUENCA

- Es necesario motivar, tanto a docentes como estudiantes, para que participen activamente de este tipo de proyectos que buscan cambiar el modelo tradicional de educación hacia un modelo centrado en el estudiante. Es importante consolidar un poco más el proyecto para que llame su atención, tratando en lo posible de integrarlos en el proceso de desarrollo del mismo.

7.3 RECOMENDACIONES

- A pesar de que el presente proyecto ha sido desarrollado exclusivamente para Android, resulta inadecuado enlazar este tipo de proyectos a una tecnología en específico, por lo que se debe procurar desarrollar esta clase de herramientas para los principales sistemas operativos móviles que existen actualmente en el mercado.
- La funcionalidad de la aplicación cliente para Android está limitada a las posibilidades que brinda la plataforma Moodle a través de los servicios Web. Sería importante profundizar en el conocimiento técnico de Moodle, para integrar nuevas y mejores funcionalidades en la aplicación móvil.
- En el desarrollo del presente proyecto se contó con la ayuda de un diseñador gráfico para crear las plantillas y las gráficas de los contenidos de la materia piloto Cálculo Diferencial. Sería recomendable mantener un estándar en la parte gráfica tanto de la aplicación como de los contenidos para evitar la contratación permanente del diseñador, para lo cual, se podría mantener la guía de diseño que se ha desarrollado en el presente trabajo de investigación.
- Actualmente el uso de la aplicación móvil está orientada a los estudiantes, sería importante integrar en el futuro funciones útiles para profesores como por ejemplo: la gestión de calificaciones, envío de tareas, informes de avance de los alumnos, etc.
- Aunque se han adaptado los recursos para el tamaño reducido de las pantallas de los teléfonos móviles, aún es posible mejorar el soporte para Tablets con el uso de fragments²².
- El presente proyecto ha contemplado únicamente la generación de contenidos de aprendizaje para la materia piloto Cálculo Diferencial. Sería recomendable realizar pruebas adicionales en otro tipo de materias, con niveles de dificultad diferentes y en otros campos de la

²²Fragment representa un comportamiento o una porción de interfaz de usuario en una actividad de Android. Se pueden combinar varios fragments en una sola actividad para construir una interfaz de usuario multi-panel y volver a utilizar un fragmento en múltiples actividades.



UNIVERSIDAD DE CUENCA

educación, para probar la efectividad del uso de este tipo de herramientas en el aprendizaje de los estudiantes.

- La forma como se realiza el manejo de los Objetos de Aprendizaje en Moodle impide adaptarlos de mejor manera en un dispositivo de pantalla pequeña como un teléfono móvil. Esta dificultad sumada al hecho de que los navegadores Web para Android no visualizan las barras de desplazamiento internas al abrir uno de estos objetos en Moodle, limitan las posibilidades en cuanto a los contenidos de aprendizaje que se pueden integrar, por lo que sería importante mejorar el soporte de Moodle para Objetos de Aprendizaje.
- Finalmente, sería recomendable la formación de un grupo que brinde soporte y continuidad a este tipo de herramientas dentro de la Facultad de Ingeniería de la Universidad de Cuenca. Este grupo podría encargarse de llevar este tipo de herramientas, no solo al resto de materias de la Facultad de Ingeniería, sino al resto de facultades de la Universidad de Cuenca y de otras universidades.

7.4 REFERENCIAS

- ISO 9241-11. (1998). Ergonomic requirements for office work with visual display terminal(VDTs). En *Guidance on usability* (pág. 11).
- Burnette, Ed. (2009). *Hello Android*.
- Clements, P. (1996). Coming Attractions in Software Architecture.
- Developers, A. (2011). *What is Android*. Recuperado el 2 de marzo de 2012, de [http:// developer.android.com/guide/basics/what-is-android.html](http://developer.android.com/guide/basics/what-is-android.html)
- Developers, A. (2012). *System Requirements*. Recuperado el 10 de marzo de 2012, de <http://developer.android.com/sdk/requirements.html>
- DiMarzio. (2008). *Android a programmer's guide*.
- Eero Maasalmi y Panu Pitkanen. (2011). Comparing Google's Android and Apple's iOS Mobile Software Development Environments.
- iOS Developer. (2012). *iOS Technology Overview*.
- Mark H. Goadrich y Michael P. Rogers. (2011). *Smart Smartphone Development: iOS versus Android*.
- Meier, R. (2009). *Professional Android Application Development*.
- Shane Conder y Layren Darcey. (2011). *Android Wireless Application Development*.
- Sonia Osses y Sandra Jaramillo. (2008). *Metacognición: Un camino para aprender a aprender*. Chile.
- Cabero Almenara Julio y Román Graván. (2005). Aplicaciones de la perspectiva cognitiva en la enseñanza a través de redes telemáticas. *Acción Pedagógica*, 6-16.
- Conklin, J. (1987). *Hypertext: An introduction and survey*.



UNIVERSIDAD DE CUENCA

- Elisa Navarro y Alexandre Texeira. (2012). *Constructivismo en la educación virtual*. Venezuela.
- Florez, R. (2012). *Hacia una pedagogía del conocimiento*. Chile.
- Ghefaili, A. (2003). Cognitive apprenticeships, technology, and the Contextualization of Learning Environments. *Journal of Educational Computing, Design & Online learning*.
- Hernández, S. (2008). El modelo constructivista con las nuevas tecnologías: aplicando el proceso de aprendizaje. *Revista de Universidad y Sociedad del conocimiento*.
- Lipscomb, L., Swanson, J., West, A. Scaffolding. (2001). *Emerging perspectives on learning, teaching and technology*. Recuperado el 2 de mayo de 2012, de http://projects.coe.uga.edu/epltt/index.php?title=Main_Page
- Maglio, F. M. (1999). *Pedagogía: Trabajar con conceptos*. Recuperado el 15 de mayo de 2012, de <http://www.fmmeducacion.com.ar/Pedagogia/trabajarconceptos.htm>
- McCombs, B.L. y Marzano, R.J. (1990). Putting the self in self-regulating learning: The self as agent in integrating will and skills.
- Murphy, E. (1997). *Constructivism: From Philosophy to Practice*. USA.
- Pintrich, P.R. (2004). The role of goal orientation in self-regulated learning.
- Bazán, C. A. (2002). *Escuela de Informática-I.S.P Modelo de Diagrama Casos de Uso*. Recuperado el 1 de junio de 2012, de <http://scribd.com/doc/17381211/001-Clase-Modelo-de-Diagrama-Casos-de-Uso-23062009>
- EcuRed. (2012). *Servidor HTTP Apache*. Recuperado el 15 de junio de 2012
- Martínez, A. (2001). *Guía a Rational Unified Process*. Recuperado el 10 de junio de 2012, de <http://www.scribd.com/doc/40218391/guia-rup>
- Rojo, J. O. (2003). *Introducción a los Sistemas Distribuidos*. Recuperado el 20 de junio de 2012, de http://augcyl.org/?page_id=231
- Group, X. P. (27 de Abril de 2007). *SOAP Version 1.2 Part 0: Primer (Second Edition)*. Recuperado el 24 de Septiembre de 2012, de <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>
- IETF. (31 de Julio de 2012). *The OAuth 2.0 Authorization Framework*. Recuperado el 24 de Septiembre de 2012, de <http://tools.ietf.org/html/draft-ietf-oauth-v2-31>
- Winer, D. (15 de Junio de 1999). *XML-RPC Specification*. Recuperado el 24 de Septiembre de 2012, de <http://xmlrpc.scripting.com/spec.html>
- Aznar, S. (2010). Generación de recursos educativos en formato estándar con eXe Learning.
- CORE Education. (2010). *eXeLearning*. Obtenido de <http://exelearning.org/>
- Gayo, J. E. (2012). *Introducción a Emacs y a Latex*.
- Hohenwarter, M. H. (2009). Documento de Ayuda de GeoGebra.



UNIVERSIDAD DE CUENCA

- Instituto Andaluz de Tecnología . (2002). *Guía para Gestión de Procesos*.
- Priego-Montilla. (2011). *Escuela TIC 2.0*.
- Quintanar, A. E. (2010). *El Impacto de las Tic en la Educación*. Brasilia.



UNIVERSIDAD DE CUENCA

ANEXO A: ENCUESTA SOBRE EL USO DE DISPOSITIVOS MÓVILES.

Con la finalidad de conocer la preferencia de los estudiantes con respecto al uso de los dispositivos móviles en nuestro medio, se vio conveniente realizar una encuesta que refleje resultados fehacientes sobre el tema. La encuesta fue orientada a los estudiantes del primer año de la Facultad de Ingeniería de la Universidad de Cuenca, ya que pertenecen a cursos numerosos, y posiblemente serán idóneos para la implementación y pruebas del proyecto.

Mediante la encuesta se pretende conocer: el tipo de dispositivo móvil más difundido, la predilección por una u otra plataforma móvil y si existe predisposición por parte de los estudiantes para utilizar los dispositivos móviles en fines educativos. Debido a que el número de estudiantes en el primer año de la Facultad de Ingeniería es relativamente grande, se ha considerado tomar una muestra de ellos, que refleje de manera apropiada la información que se desea obtener. Para garantizar que la muestra escogida es la correcta, se hará uso de fórmulas estadísticas con datos que se explican más adelante.

Cálculo del tamaño muestral

Para obtener un tamaño muestral idóneo que revele resultados seguros y con la mayor precisión posible, se debe elegir correctamente el valor asignado a los parámetros que intervendrán en el cálculo de la muestra.

Como se mencionó previamente, la población a ser estudiada consiste en los alumnos del primer año de la Facultad de Ingeniería, por lo tanto tenemos un número finito de individuos y por ello utilizaremos la siguiente fórmula estadística:

Formula estadística Nro. 1

$$n = Z_{\alpha}^2 \frac{N \cdot p \cdot q}{i^2(N - 1) + Z_{\alpha}^2 \cdot p \cdot q}$$



UNIVERSIDAD DE CUENCA

n	Tamaño de la muestra representativa que deseamos obtener
N	Tamaño de la población
Z_{α}	Valor correspondiente a la distribución de Gauss (siendo $\alpha = 0.05$ el valor escogido estándar, lo que da $Z_{0.05} = 1.96$)
i	Error de estimación el cual se prevé cometer, para este caso se utilizara el valor estándar del 10%, con lo que $i = 0.1$
p	Proporción de la variable a ser estudiada, en el caso de no conocerla se toma el valor estándar 0.5
q	1-p

Tabla Anexo A.1 Explicación de los parámetros que intervienen en la formula Nro. 1

N =	400
$Z_{\alpha} =$	1.96
i =	0.1
p =	0.5
q =	0.5

Tabla Anexo A.2 Aplicación de la formula

$$n = 1,96^2 \frac{400 * 0.5 * 0.5}{0.1^2(400 - 1) + 1.96^2 * 0.5 * 0.5}$$

$$n = 77.60$$

El total de alumnos a ser encuestados es de **78** aproximadamente.

Preguntas de la encuesta:

La siguiente encuesta tiene la finalidad de conocer, si los estudiantes disponen de un dispositivo móvil y si, además, conocen sus características.



UNIVERSIDAD DE CUENCA

1. Sexo

Respuesta	Media	Total
Hombre	 88%	69
Mujer	 12%	9
Total	 100%	78/78

Respuesta	Media	Total
Teléfono celular	 68%	57
Smartphone (Samsung, LG, Motorola, Nokia)	 18%	15
Iphone	 7%	6
Ipad	 4%	3
Ninguno	 4%	3

2. ¿Qué tipo de dispositivo móvil utiliza?

Android	 15%	12
iOS	 19%	15
Mi móvil es antiguo	 38%	31
Desconozco el tema	 28%	23

3. ¿Su dispositivo móvil utiliza alguno de estos sistemas operativos?



UNIVERSIDAD DE CUENCA

4. ¿Cuál diría usted que son los usos más frecuentes que le da a su dispositivo móvil?

Respuesta	Media	Total
Entretenimiento	28%	31
Comunicación	57%	64
Trabajo	5%	6
Educación	10%	11

5. ¿Con qué frecuencia entra a Internet con su dispositivo móvil?

Respuesta	Media	Total
Varias veces al día	18%	14
Al menos, una vez al día	8%	6
Algunas veces a la semana	10%	8
Algunas veces al mes	10%	8
No entro a Internet con el móvil	54%	42

6. ¿Planea comprar alguno de los siguientes dispositivos móviles durante el próximo año?

Respuesta	Media	Total
Smartphone (Samsung, LG, Motorola, Nokia)	21%	18
Tablet (Asus, Samsung, Acer, Dell, Archos)	6%	5
Iphone	24%	21
Ipad	10%	9
No	38%	33

7. ¿Estaría usted dispuesto a utilizar su móvil para fines educativos?



UNIVERSIDAD DE CUENCA

Respuesta	Media	Total
Sí	94%	73
No	6%	5
Total	100%	78/78

8. ¿Cuáles considera que serían las principales limitaciones u obstáculos que tendría usted al utilizar un dispositivo móvil para estudiar una asignatura?

Respuesta	Media	Total
Dificultad para manejar todas las funciones del móvil	10%	9
Desconocimiento de las funciones del móvil	9%	8
Costos	48%	45
Problemas de acceso a internet	24%	23
Otros.	10%	9

ANEXO B: MANUAL DE INSTRUCCIONES DE LA APLICACIÓN CLIENTE DE ANDROID PARA MOODLE EVIAM

Esta guía de usuario le ayudará a familiarizarse rápidamente con las funcionalidades que le ofrece la aplicación cliente de Android para Moodle **EVIAM**. Se recomienda completar totalmente la lectura de esta guía antes de empezar a utilizar la aplicación.

INICIO DE SESIÓN

Para iniciar sesión en la aplicación, presione en la pantalla de su dispositivo el ícono de la aplicación EVIAM, se le presentará la pantalla de bienvenida como se muestra en la **Figura B-1**.



UNIVERSIDAD DE CUENCA



Figura B-1 Inicio de la aplicación

Una vez en la pantalla de bienvenida, presionar el botón **Login**. Esto hará que la aplicación abra el navegador Web predeterminado en su dispositivo y se conecte a **eviamnet.com** donde deberá ingresar sus credenciales como se muestra en la **Figura B-2**.



Figura B-2 Conexión de la aplicación con eviamnet.com

Una vez que ingresa correctamente sus credenciales en **eviamnet.com**, se le presentará una pantalla de **solicitud de autorización** donde usted deberá autorizar el acceso de la aplicación a sus datos finalizando así con el proceso de inicio de sesión como se muestra en la **Figura B-3**.



UNIVERSIDAD DE CUENCA



Figura B-3 Finalización del proceso de inicio de sesión

PANTALLA DE INICIO

Una vez que usted haya iniciado sesión en la aplicación (Ver numeral anterior) se le presentará la pantalla de **Inicio** como se puede ver en la **Figura B-4**. En esta pantalla usted podrá acceder a las principales funcionalidades de la aplicación.



Figura B-4 Pantalla de inicio de la aplicación

PANTALLA DE PREFERENCIAS

Para modificar las preferencias de la aplicación, se debe presionar el botón **Menú** del teléfono, el cual abrirá un pequeño menú en la parte inferior de la



UNIVERSIDAD DE CUENCA

pantalla. En este menú se debe escoger la opción **Preferencias** para acceder a la **Pantalla de Preferencias** como se observa en la **Figura B-5**.



Figura B-5 Acceso a las preferencias de la aplicación

Dentro de las preferencias que pueden ser modificadas en la aplicación están:

- **Activar caché:** la aplicación guarda la información descargada desde el servidor en una caché interna, para que la próxima vez que se desee acceder a ellos, se carguen localmente aumentando la velocidad de la aplicación. Esta opción se encuentra activada por defecto.
- **Refrescar datos cada:** Los datos que se encuentran almacenados en la caché tienen un período dentro del cual son considerados como datos recientes. Cuando este periodo ha vencido la aplicación vuelve a obtener los datos desde el servidor y actualiza la caché. Esta opción permite modificar este período de validez. El valor por defecto es de 10 minutos.



UNIVERSIDAD DE CUENCA

ELEMENTOS GENERALES DE LA INTERFAZ



Figura B-6 Elementos generales de la interfaz de usuario

LISTAR LOS CURSOS Y SUS CONTENIDOS

En la pantalla de Inicio se debe escoger la opción Mis Cursos. Esto hará que la aplicación abra la pantalla **Mis Cursos** donde se muestra la lista de cursos en los que el usuario se encuentra matriculado. Esto se muestra en la **Figura B-7**.



UNIVERSIDAD DE CUENCA



Figura B-7 Listar los cursos del usuario

Cuando se escoge uno de los cursos de la lista, se presenta la pantalla **Contenidos** que muestra una lista con todos los contenidos que se encuentran actualmente dentro del curso. Esto se puede ver en la **Figura B-8**.



Figura B-8 Lista de contenidos del curso

LISTAR COMPAÑEROS DE CURSO Y SUS PERFILES



UNIVERSIDAD DE CUENCA

En la pantalla de Inicio se debe escoger la opción Mis Compañeros. Esto hará que la aplicación abra la pantalla **Compañeros** donde se muestra la lista de compañeros del usuario separados por curso. Esto se muestra en la **Figura B-9**.



Figura B-9 Lista de compañeros del usuario

Si se presiona sobre el nombre de un compañero podemos ver los detalles de su perfil como se observa en la **Figura B-10**.



Figura B-10 Perfil de un compañero de curso