



Universidad Autónoma de Baja California
Facultad de Ingeniería, Arquitectura y Diseño



**IMPLEMENTACIÓN DE UNA HERRAMIENTA PARA
DESARROLLAR APLICACIONES EDUCATIVAS PARA
DISPOSITIVOS MÓVILES (M-LEARNING) PARA
MÚLTIPLES PLATAFORMAS.**

Alumno

Luis Marcel Celaya Lomelí

Director

Dr. Christian Xavier Navarro Cota

Febrero de 2020

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

**IMPLEMENTACIÓN DE UNA HERRAMIENTA PARA DESARROLLAR APLICACIONES
EDUCATIVAS PARA DISPOSITIVOS MÓVILES (M-LEARNING) PARA MÚLTIPLES
PLATAFORMAS**

TESIS

PARA CUBRIR LOS REQUISITOS NECESARIOS PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

LUIS MARCEL CELAYA LOMELÍ

Aprobada por:



Dr. Christian Xavier Navarro Cota
Director
(Presidente)



MC. Sergio Omar Infante Prieto
Sinodal
(Secretario)



MI. Victor Rafael Nazario Velázquez
Sinodal
(Vocal)

AGRADECIMIENTOS

En primer instancia agradezco a mis profesores quienes me han permitido aprender un poco de su experiencia y me han llevado a conocer y a aprender todo lo que conozco en este momento, profesores que quisiera destacar por su impacto que han tenido a lo largo de mi estancia en la universidad, empezando por el profesor Pedro Yepiz quien fue el primer contacto que me ayudó a aprender las bases de programación, al maestro Odin Meling quien siempre estuvo apoyando en el devenir de este camino, a los maestros Victor Velazquez y Sergio Infante los cuales me mostraron que estaba por el camino correcto y a encariñarme con mi profesión, a la maestra Luz Evelia la cual siempre estuvo apoyándonos en cualquier situación que se pudiera haber presentado y al Dr. Christian Navarro quien me estuvo apoyando a lo largo de toda la carrera y me presentó la oportunidad de presentar este trabajo así como su experiencia para poder concretarlo y en general a todos los demás profesores que no he mencionado por su conocimiento transmitido.

También, quisiera agradecer a mis compañeros de clase, los cuales siempre me apoyaron y motivaron a seguir adelante, mencionando a Bryan Quiroz el cual fue un compañero y amigo que siempre estuvo apoyándome, a mi compañero en las materias de electrónica Ivan Ramirez que muy similar a mi siempre nos apoyamos para terminar los trabajos a tiempo, a Darío Diaz, Victor Guardado, Alberto Lozano, Ernesto Murillo los cuales estuvieron desde que entré a la carrera conmigo y a todos mis demás compañeros que me formaron parte de toda esta gran experiencia.

Para terminar, quiero agradecer a mis padres que siempre me estuvieron empujando y motivando para llegar a este momento de mi vida y que con su experiencia, apoyo y motivación he podido llegar y completar todos los retos que se me han presentado.

ÍNDICE GENERAL

1.	Introducción	8
1.1.	Planteamiento del problema	9
1.2.	Justificación	10
1.3.	Objetivos	13
1.4.	Metodología	14
2.	Fundamentos teóricos y antecedentes.....	21
2.1.	Tendencias de la educación.....	21
2.2.	m-learning en la educación	23
2.3.	Definición de m-learning	25
2.4.	Usabilidad en dispositivos móviles.....	26
2.5.	Framework para diseñar aplicaciones m-learning	32
2.5.1.	Usabilidad Pedagógica	34
2.5.2.	Usabilidad de la interfaz de Usuario	41
3.	Diseño e implementación de una herramienta para desarrollar aplicaciones móviles educativas multiplataformas	49
3.1.	Programación en capas.....	49
3.2.	Servicios Web	51
3.3.	Requerimientos del Software.....	52
3.3.1.	Requerimientos funcionales del sistema propuesto.....	53
3.3.2.	Requerimientos funcionales del administrador.....	55

3.3.3.	Requerimientos funcionales para el usuario final	60
3.3.4.	Requerimientos no funcionales	62
3.4.	Diagrama de Arquitectura	64
3.5.	Diagrama de Base de datos.....	69
3.6.	Patrones de diseño	71
3.7.	Implementación de la herramienta	73
3.8.	Interfaces de usuario	79
4.	Conclusiones	90
	Referencias bibliográficas.....	92
	Referencia de páginas Web.....	95

ÍNDICE DE FIGURAS

Figura 1.1. Descripción gráfica del modelo en “V”	18
Figura 2.1. Modelo de Calidad del Producto (ISO/IEC 25010:2011).	27
Figura 2.2. Modelo de Calidad en Uso (ISO/IEC 25010:2011).	27
Figura 2.3. Marco conceptual de evaluación de sistemas <i>m-learning</i>	33
Figura 2.4. Aspectos de la <i>usabilidad pedagógica</i> y sus criterios de evaluación.	35
Figura 2.5. Aspectos y criterios de evaluación de la <i>usabilidad de la interfaz de usuario</i> o <i>tecnológica</i>	42
Figura 3.1. Diagrama de arquitectura del sistema propuesto.	65
Figura 3.2. Diagrama de Base de datos del sistema propuesto.....	70
Figura 3.3. Clases que pertenecen a la capa de Dominio del sistema.....	73
Figura 3.4. Capa de Datos del sistema	74
Figura 3.5. Capa de negocio del sistema	75
Figura 3.6. Archivos de Enumerables	76
Figura 3.7. Capa de servicio del sistema.	76
Figura 3.8. Clases del proyecto MVC	77
Figura 3.9. Estructura del proyecto de Ionic	79
Figura 3.10. Interfaz en la cual se enlistan los módulos creados por el administrador.	80
Figura 3.11. Formulario para crear un nuevo módulo.	81
Figura 3.12. Interfaz que enlista un módulo creado.	82

Figura 3.13. Interfaz para crear preguntas.....	82
Figura 3.14. Interfaz donde se pueden incluir diferentes respuestas.	83
Figura 3.15. Interfaz donde se pueden eliminar respuestas o elegir cual es la correcta.	84
Figura 3.16. Interfaz de usuario de la aplicación móvil en la cual se presentan los módulos o cursos	85
Figura 3.17. Interfaz móvil con un ejemplo del contenido educativo de un módulo creado.....	86
Figura 3.18. Interfaz de la aplicación móvil reproduciendo un video contenido en un módulo.....	87
Figura 3.19. Interfaz móvil con un ejemplo de un cuestionario de evaluación	88
Figura 3.20. Interfaz donde se muestra el resultado de un cuestionario de evaluación.	89

ÍNDICE DE TABLAS

Tabla 3.1: Requerimientos Funcionales del Sistema propuesto.....	53
Tabla 3.2: Requerimientos del Sistema para el Administrador del sistema.....	56
Tabla 3.3: Requerimientos del sistema correspondientes al Usuario Final (estudiante).....	60
Tabla 3.4: Requerimientos no funcionales	63

Capítulo 1. Introducción

Actualmente, los dispositivos de cómputo móviles forman parte de la vida diaria y la cultura de muchas personas. Estos dispositivos están proporcionando un acceso histórico a la comunicación y la información. En el informe "*Ericsson Mobility Report of 2016*", se realizaba una predicción, en la que se afirmaba que para el 2021 más del 90% de la población mundial estará cubierta por las redes móviles de banda ancha. (Ericsson, 2016).

La facilidad, flexibilidad de uso y la mejora constante de las capacidades de estos dispositivos plantean infinidad de posibilidades que benefician a los usuarios, siendo una de dichas áreas la educación. Dichos dispositivos resultan herramientas muy útiles, que nos pueden acompañar a donde quiera que vayamos. Nos encontramos, por tanto, ante una herramienta ideal para cualquier aprendiz durante su proceso de aprendizaje, ya que la información está disponible en cualquier momento y lugar. Incluso, la UNESCO ha considerado los dispositivos móviles como plataforma para ampliar el acceso, la calidad y la igualdad de la enseñanza en diferentes países (West & Ei, 2014). Además, esta organización ha sido anfitrión de su *Mobile Learning Week* durante varios años

consecutivos, con el objetivo de promover la adopción del *m-learning* e intentar potenciar el ámbito educativo.

Sin embargo, los dispositivos móviles poseen ciertas limitaciones, principalmente relacionadas con el reducido tamaño de sus pantallas y la cantidad de información que se puede presentar a la vez. Dichos aspectos, más relacionados con la *usabilidad*, podrían dificultar el uso de los móviles como soporte para tareas de enseñanza-aprendizaje. Por otro lado, los *aspectos pedagógicos* constituyen otra de las fases importantes en el diseño de aplicaciones educativas móviles, ya que su calidad didáctica depende en gran medida del hecho de que se encuentre la necesaria coherencia entre el objetivo que se quiere alcanzar, los contenidos que se tratan, y las actividades interactivas soportadas por dichas aplicaciones (Marqués, 1995). Los *aspectos pedagógicos* también se relacionan con la usabilidad tecnológica. Así, Gebera(2012) emplea el término *usabilidad pedagógica* para referirse al estudio global de los *aspectos pedagógicos* de las aplicaciones interactivas.

1.1 Planteamiento del Problema

Comúnmente, existe un alto índice de reprobación en la educación superior. Algunos de los cursos en los cuales los alumnos tienen problemas son los siguientes: materias de cálculo, ecuaciones diferenciales, circuitos, estática, programación, entre otras. Esto se debe a distintos factores, como la falta de educación personalizada, la falta de ejercicios para ser resueltos, la falta de retroalimentación de las tareas asignadas, incluso la falta de seguridad para preguntar dudas al profesor. Pero es importante resaltar que las

personas tienen diferentes tipos de aprendizaje: algunos de ellos son más visuales, y otros más auditivos o kinestésicos. Es por ello que han surgido distintas herramientas para que las personas aprenden, por ejemplo, existen los libros, artículos en internet, videos en *Youtube* y cursos virtuales, en el cual es posible mezclar diferentes tipos de contenidos (lecturas, videos y ejercicios) para los distintos tipos de aprendizaje. En este tipo de cursos el alumno tiene la flexibilidad de elegir el contenido que más se adapte a su estilo de aprendizaje, puede leer el contenido, ver videos y/o realizar ejercicios, ya sea de forma individual o por equipo.

Por otra parte, existe el aprendizaje móvil (*m-Learning*), que es una forma de aprender que se ha vuelto muy popular en los últimos años, ya que facilita el aprendizaje en todo tiempo y en todo lugar, gracias a que existen diversos tipos de dispositivos móviles como *Tablets, teléfonos móviles smartphone, Readers, entre otros*.

Es por esto, desarrollar aplicaciones móviles educativas puede ser muy útil para que los alumnos tengan un conocimiento reforzado, sin la necesidad de tener que hacer tantas intervenciones con los maestros, y con esto tratar de reducir el número de alumnos reprobados.

1.2 Justificación:

El rol que tienen los dispositivos móviles en la vida diaria permite el *aprendizaje ubicuo* de las personas, lo cual, en el ritmo de vida tan acelerado que se vive actualmente es

fundamental que existan herramientas que permitan esto. El *m-learning* no es algo nuevo, ha existido, y seguirá existiendo, sin embargo, debido a la evolución constante de la tecnológica cada día existen nuevas herramientas que pueden permitir este tipo de aprendizaje de manera que es importante construir una plataforma que permita no solo consumir temas de interés, si no que permita crearlos para que puede ser utilizado por cualquier persona/creador, como lo puede ser un profesor, un aficionado, un experto o simplemente una persona que quiere aprender algo.

En los últimos años se han desarrollado una gran cantidad de aplicaciones educativas destinadas a ser utilizadas con dispositivos móviles (o *apps*¹). Se calcula que a día de hoy existen más de 80.000 aplicaciones educativas en las diferentes plataformas de venta de *apps* para *smartphones* y *tablets*: *AppStore*, *Google Play*, *Samsung Apps*, entre otras. Sin embargo, la deficiente usabilidad de muchas de ellas es una de las principales causas que hace que los estudiantes o usuarios se desmotiven y dejen de utilizarlas. Así, por ejemplo, la navegación puede derivar en desinterés por parte del usuario, cuando ésta se considera complicada y tediosa a la hora de completar una tarea, o la presentación de contenidos puede resultar muy complicada o elaborada para algunos usuarios o muy trivial para otros, dependiendo de factores como la edad, su conocimiento previo sobre el tema tratado, entre otros.

¹Término en inglés y abreviado para referirse a aplicaciones para teléfonos móviles inteligentes.

El *m-learning* permite la retención debido a que son ejercicios que están completamente diseñados para los temas que necesita aprender el usuario y fuerzan a comprender el tema para poder continuar con las lecciones.

Las opciones que existen actualmente de *m-learning* usualmente son de centros específicos de enseñanza los cuales no solo no son restrictivos a ciertas áreas específicas, tampoco permiten que sus plataformas sean utilizadas por terceras personas para la enseñanza de conocimiento.

Por lo tanto, se requiere que el desarrollador de aplicaciones lleve a cabo una cuidadosa planificación y un diseño antes de implementarlas. Asimismo, es necesario realizar un proceso de evaluación de la usabilidad de dichas aplicaciones. Esta evaluación se puede realizar a lo largo de las diferentes etapas de desarrollo del *software* o en la etapa final. El objetivo es asegurar que la aplicación sea útil, eficaz y fácil de manejar, desde la perspectiva del estudiante, para así poder dar soporte al proceso de enseñanza-aprendizaje.

Son muchos los autores que han investigado y propuesto recomendaciones (guías, principios, etc.) para el desarrollo de aplicaciones *m-learning* (Duarte Filho & Barbosa, 2013; Economides, 2008; Elias, 2011; Parsons, Ryu, & Cranshaw, 2007), pero pocos son los que consideran aspectos específicos de usabilidad (Fetaji, Ebibi, & Fetaji, 2011). Por otro lado, los autores que tratan el desarrollo de aplicaciones *m-learning* usables no

consideran los aspectos puramente pedagógicos. Consideramos que desarrollar este tipo de aplicaciones se convierte en un aspecto crítico, ya que en ellas no solo es importante lograr los objetivos de aprendizaje, sino lograr una buena usabilidad (tecnológica y pedagógica), además de un ambiente que sea atractivo y motivador para los estudiantes.

Ante esta situación creemos necesaria una herramienta para desarrollar aplicaciones educativas, considerando aspectos pedagógicos y de usabilidad, y orientado específicamente a las aplicaciones educativas móviles (*m-learning*), que contemple los objetivos propios de este dominio. Para ello, es necesario analizar las mejores fórmulas para enriquecer sus contenidos educativos, sus interfaces, la experiencia del usuario, y así lograr prácticas educativas más motivadoras. Esto podría permitir establecer hasta qué punto los componentes de la aplicación cumplen los requisitos de usabilidad para dar soporte al proceso de enseñanza-aprendizaje.

1.3 Objetivos

A continuación, se plantea una serie de objetivos y sub-objetivos a cumplir en el contexto de la presente tesis:

Objetivo general

El objetivo general de esta tesis es diseñar e implementar una herramienta para desarrollar aplicaciones educativas, considerando aspectos pedagógicos y de usabilidad, y orientado específicamente a las aplicaciones educativas móviles (*m-learning*), que

contemple los objetivos propios de este dominio y la diversidad de sistemas operativos para estos dispositivos.

Objetivos específicos:

1. Identificar los aspectos pedagógicos que debe tener una aplicación de *m-learning* para mejorar el aprendizaje del estudiante.
2. Identificar los elementos de usabilidad que permiten a los usuarios tener una buena experiencia al utilizar sus aplicaciones para dispositivos móviles.
3. Diseñar e implementar una herramienta que permita desarrollar aplicaciones educativas para dispositivos móviles, considerando los aspectos obtenidos en los dos objetivos específicos anteriores, así como los diferentes sistemas operativos para estos dispositivos.
4. Desarrollar de una aplicación educativa multiplataforma (Android/iOS/Web), utilizando a la herramienta desarrollada, con el objetivo de probar su funcionalidad.

1.4 Metodología.

Para este proyecto es necesario definir un alcance, identificando las necesidades por las cuales es requerido este proyecto. De esta manera se podrá llegar a un Sistema, el cual nos permitirá en un principio solucionar los problemas obtenidos en investigación de esta tesis. Ya que se pretende desarrollar una aplicación tipo cliente/servidor se propone una metodología en V, la cual nos permitirá iterativamente llevar un control de cambios y

errores, sin descuidar que la aplicación este codificado con buenos estándares de manera que este sea susceptible a cambios, escalable y pueda tener un desarrollo posterior.

Se propone el modelo en V debido a que solo una persona va a desarrollar esta aplicación, ya que otras metodologías que pueden ser Agiles, tales como Scrum, Lean o XP, no aplican para este tipo de desarrollos. Además, una metodología de cascada tradicional puede contener muchos errores al final de esta.

Como se mencionó anteriormente, se pretende definir una arquitectura cliente/servidor, en el cual, nuestro servidor va a ser el encargado de tener toda la lógica relacionada a nuestro modelo de la aplicación. Pensando en los requerimientos que se van a obtener la aplicación móvil será utilizada por los alumnos como plataforma de *m-learning*, y la aplicación Web fungirá como sistema de gestión de cursos, la cual se encargará de agregar, modificar o eliminar contenido en la aplicación, de manera que si un contenido cambia, se agrega o se elimina dentro de la aplicación Web, la aplicación de *m-learning* necesitará sincronizarse de nuevo para garantizar que el contenido educativo de la aplicación sea la más reciente. Una vez planteado esto se pretende realizar la fase de pruebas para validar que los sistemas estén correctamente integrados.

La metodología en V consiste en 8 fases iterativas (ver Figura 1.1), las cuales se explican a continuación:

- **Modelado de requerimientos:** Esta fase permite al proyecto recopilar información y modularizarla en requerimientos, los cuales contienen pedazos pequeños de funcionalidad y están descritos para poder ser implementados de una manera clara y objetiva.
- **Diseño de la arquitectura:** Esta fase se encarga de realizar un diseño de como los requerimientos se pretenden implementar en la aplicación, de manera que estén bien acoplados.
- **Diseño de los componentes:** Debido a que los componentes deben tener un diseño esta fase permite que saber cómo se deben de ver para no tener claridad sobre lo que se debe implementar.
- **Generación de código:** Con esta fase se pretende hacer funcional las fases que fueron descritas teóricamente en las fases previas a esta, esto con base en código de computadora.
- **Pruebas unitarias:** Cada función que se realiza en la generación de código debe tener una prueba unitaria garantizando así que el código sea de calidad y cumpla únicamente con la función para la cual se fue hecho.
- **Pruebas de integración:** Esta fase nos permite validar que el diseño de los componentes fue realizado correctamente de manera que todos los componentes estén correctamente acoplados.

- **Pruebas de sistema:** Esta fase nos permite validar el diseño de la arquitectura de manera que nuestra aplicación funcione de manera óptima.
- **Pruebas de aceptación:** Esta fase nos permite validar la fase de requerimientos ya que nos permite validar cada requerimiento con funcionalidad que ya fue probada anteriormente.

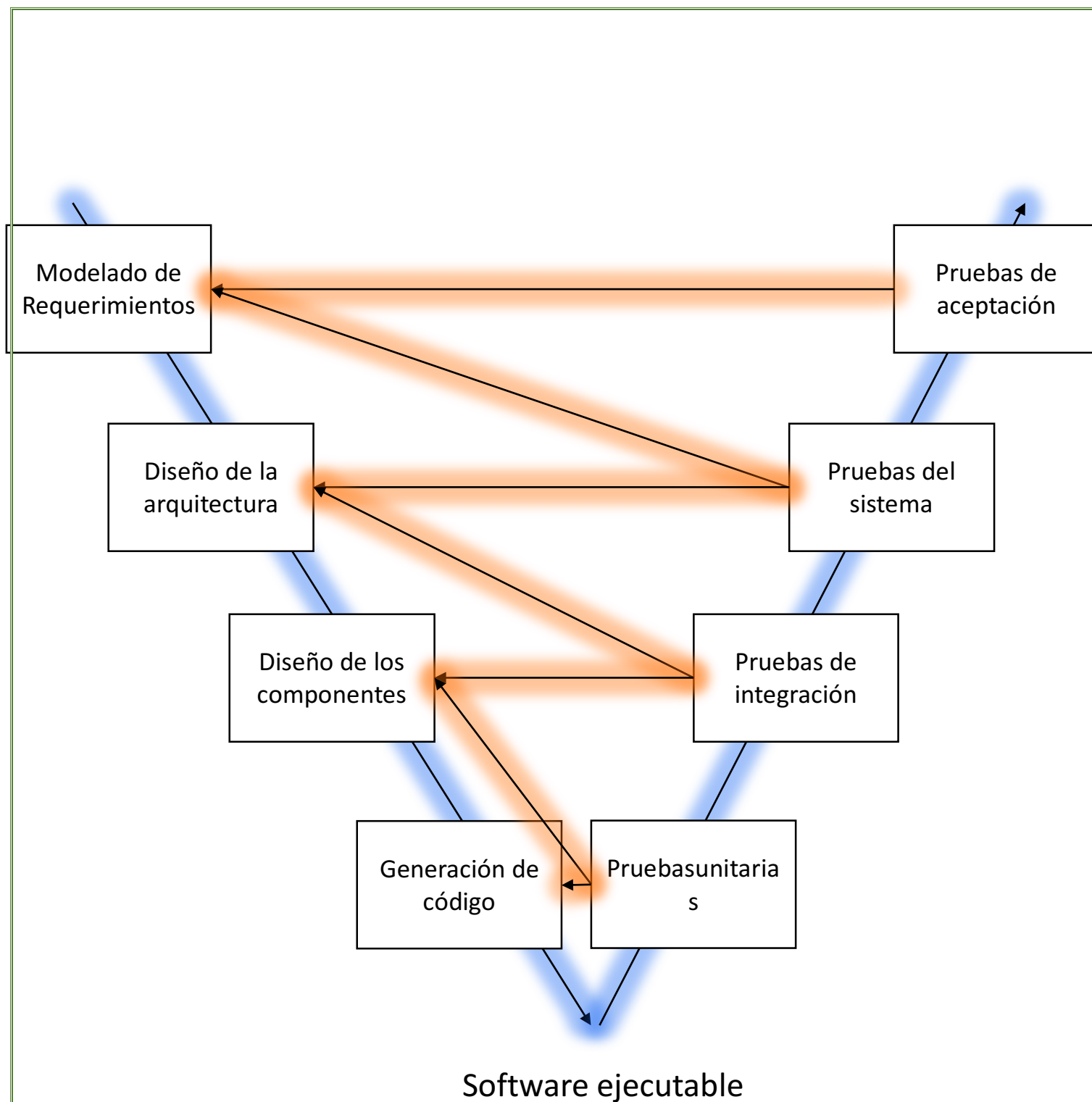


Figura 1.1. Descripción gráfica del modelo en "V"

Esta metodología como se puede observar está representada por niveles del 1 al 4 y se explican brevemente a continuación

- El **nivel 1** (Modelado de requerimientos – Pruebas de Aceptación) se trata de la documentación de la funcionalidad deseada para este proyecto
- El **nivel 2** (Diseño de la arquitectura – Pruebas de sistema) revisa las características del sistema y genera documentación de análisis funcional.
- El **nivel 3** (Diseño de los componentes – Pruebas de integración) se le denomina arquitectura del sistema y pretende mostrar y validar cada componente que se va a realizar
- El **nivel 4** (Generación de código – Pruebas unitarias) permite plasmar y probar todo lo que fue requerido anteriormente.

Los objetivos de esta metodología pretenden considerar los siguientes aspectos:

- **Minimizar los riesgos del proyecto:** Cada fase que pudiera tener el desarrollo de software tiene una contraparte dedicada a probar la funcionalidad permite validación de todo lo que se hace.

- **Mejora continua:** Al ser una metodología iterativa permite arreglar cambios que fueron hechos o simplemente cambiar funcionalidad

Entre las muchas ventajas que tiene el modelo en "V" nos permite llevar un control de resultados por cada fase existente, esto, nos permite documentarlo de manera que no solo tenemos resultados finales, si no parciales y corregibles. Esto nos permite ajustar cada fase para esperar nuestro resultado deseado permitiendo que el software se adecue a las necesidades tanto de usuarios maestros y usuarios alumnos. Si bien existen diferentes aplicaciones móviles educativas, no se encontró ninguna que proporcione un sistema para administrar o crear diferentes cursos móviles, por lo cual es una ventaja clara ante todas las aplicaciones ya existentes permitiendo agregar un sinfín de temas para aprender.

Finalmente, se pretende medir el impacto de una la aplicación educativa, creada con la herramienta planteada, utilizándose en un ambiente real considerando una cantidad de usuarios, para obtener una retroalimentación sobre los resultados esperados.

Capítulo 2. Fundamentos Teóricos y Antecedentes

El objetivo del presente capítulo es describir las bases teóricas y conceptuales que soportan el desarrollo de esta tesis.

2.1. Tendencias en la educación

El informe *NMC Horizon Report* identifica cuales de las nuevas tecnologías presentan un mayor potencial en el ámbito de la educación (Johnson, et al., 2013). Estas tendencias son ya un hecho en algunas de las instituciones más innovadoras del mundo. Así, a corto plazo, los MOOC (por sus siglas en inglés, *Massively Open Online Courses*) destacan como una de estas nuevas tendencias. Dicho informe indicaba que los MOOC tendrán una aceptación muy amplia, principalmente en el ámbito de la educación superior. Los MOOC proveen una opción gratuita y de gran calidad en educación. Algunos de los que han tenido una mayor promoción son *Coursera*, *edX*, *Miriada X* y *Udacity*, los cuales cuentan con cientos de miles de usuarios registrados. La principal ventaja de los MOOC es que facilitan el aprendizaje continuo y avanzado de forma gratuita, y permite a cualquier estudiante o profesional adquirir habilidades, conocimientos o aptitudes para encontrar empleo o desarrollarse efectivamente en su propia empresa (Johnson, et al., 2013).

Otra tecnología que es apuntada como una de las que a corto plazo tendrán mayor aceptación son las *tablets* (o simplemente dispositivos móviles con pantallas de gran tamaño). Sus principales ventajas, además del tamaño de la pantalla, son la conectividad, multifuncionalidad, posibilidades de interacción y portabilidad, por lo que están demostrando ser un medio útil para impulsar la educación sin importar el tiempo y el espacio; permitiendo el acceso a materiales educativos y sirviendo como herramienta de apoyo a la gestión docente (Durall Gazulla, Gros Salvat, Maina, Johnson, & Adams, 2012). Además, cada vez existe un mayor número de compañías dedicadas a la fabricación de *tablets*, lo cual fomenta la competitividad de precios y productos, a la vez que mejora la oferta y la disponibilidad. Este mercado creciente favorece a los estudiantes e instituciones las posibilidades de aprendizaje (Johnson, et al., 2013).

Otra tendencia en la educación es el *aprendizaje basado en juegos*. Durante la última década se ha estudiado la posibilidad de que los juegos sean herramientas para el desarrollo del conocimiento y el aprendizaje. Y cuanto más atención genera el tema en el ámbito educativo, más implementaciones se generan para apoyar el aprendizaje (Durall Gazulla, et al., 2012).

Otra tendencia es la *analítica del aprendizaje (learning analytics)*, que tiene que ver con la interpretación de los datos que generan los estudiantes al interactuar con los contextos de aprendizaje. Los estudios recientes muestran que su uso llevará a crear entornos de

aprendizaje personalizados, adaptando los contenidos a presentar a los aprendices a lo largo de su proceso de enseñanza según su propio progreso (Johnson, et al., 2013).

Otras tendencias de la educación que menciona Moreno-Ger, Martinez-Ortiz, Freire, Manero, and Fernandez-Manjon (2014) son: la *evaluación encubierta* (*stealth assessment*) y los *juegos serios* con accesibilidad. La *evaluación encubierta* contempla técnicas de la *analítica del aprendizaje* para recolectar datos, y utiliza la información derivada del uso de una aplicación para evaluar el progreso del aprendiz de manera transparente para éste. Asimismo, los *juegos serios* deben promover el uso universal, para evitar que el juego pueda suponer un obstáculo para personas con discapacidad. Por lo tanto, la tendencia es que el desarrollo de *juegos serios* contemple características que favorezcan la accesibilidad.

2.2. *m-learning* en la educación

Como se mencionó en el capítulo anterior, la UNESCO ha publicado una serie de informes sobre el aprendizaje móvil, en los que analiza cómo la tecnología móvil, que ahora es más accesible, puede facilitar la igualdad y la eficacia de la educación en diferentes países. Por otro lado, los dispositivos móviles están en continua mejora en cuanto a sus capacidades y las funciones que ofrece al usuario, además de que su disponibilidad está creciendo. Todo ello plantea nuevas formas de apoyo al proceso de enseñanza/aprendizaje. Diferentes estudios y publicaciones sobre el aprendizaje móvil

surgidos en todo el mundo han hecho evidente este potencial y, de un modo u otro, muchos de esos proyectos o la mayoría están ayudando a que las personas puedan acceder al aprendizaje en este nuevo escenario (UNESCO, 2013).

El *m-learning* se ha estudiado en diversos contextos educativos, entre los cuales destacamos la educación formal e informal y el aprendizaje continuo. Se asume de manera generalizada que hay una división significativa entre el aprendizaje formal, que se da en las aulas, y el aprendizaje informal, que acontece en casa o en contextos diversos fuera del aula. Consideramos, como muchos otros autores, que el *m-learning* podría ayudar a disminuir la separación entre estos dos tipos de aprendizaje.

Dentro del aprendizaje formal hay dos modelos populares de aprendizaje móvil en las escuelas, uno es de los *programas 1:1* en los que se provee a cada estudiante de un dispositivo, y otro las *iniciativas BYOT (Bring Your Own Technology)*, en la cual la mayoría de los estudiantes tienen sus propios dispositivos y los llevan a la clase (Cisco, 2012).

Dentro del aprendizaje informal podemos considerar el llamado *aprendizaje permanente o continuo*, que se traduce en la educación a lo largo de la vida. Éste no cambia con el currículum oficial, sino que avanza con el transcurrir de la vida de la persona dependiendo de su situación profesional y personal. El aprendizaje permanente puede darse en cualquier espacio y sus principales características son la integración e innovación (Valero, Redondo, & Palacín, 2012).

2.3. Definición del *m-learning*

Actualmente, los dispositivos móviles son ampliamente utilizados en la educación, con diferentes enfoques dependiendo de los autores, las regiones o la audiencia. Todas estas formas diferentes de uso han dado lugar a diferentes términos que definen el uso de estos dispositivos móviles con propósitos educativos, como el *m-learning* (O'Malley, et al., 2005), *ubiquitous learning* (Sakamura & Koshizuka, 2005), *seamless learning* (Wong & Looi, 2011), *blended learning* (Yen & Lee, 2011) o *smart education* (Zhou, Huang, Chen, & Dai, 2015).

La definición de *m-learning* ha evolucionado en los últimos años, de manera que distintos autores lo han definido de distinta forma. Por ejemplo, Clark Quinn (2000) lo define como “un tipo de *e-learning* a través de dispositivos móviles”. O'Malley, et al. (2005) lo describe como “el aprendizaje que tiene lugar cuando el estudiante se beneficia de las oportunidades ofrecidas por las tecnologías móviles”. El trabajo de Sharples, Taylor, and Vavoula (2005) cambió la forma de pensar acerca de *m-learning*, enfocándolo en el aprendiz. Según este autor es el aprendiz quien tiene la movilidad y no la tecnología. Los estudiantes eligen la tecnología que esté a su mano mientras se mueven entre contextos, incluyendo teléfonos móviles, sus propias computadoras y las de otros, así como libros y *notepads*. Y en el 2013 Crompton (2013) lo define en los siguientes términos: “el *m-learning* es un aprendizaje en múltiples contextos, a través de interacciones sociales y de contenido, usando dispositivos electrónicos personales”. Esta última definición es la que

tomamos como principal referencia, ya que está más centrada en el aprendiz y su proceso de aprendizaje.

2.4. Usabilidad en dispositivos móviles

En esta sección se presentan una serie de definiciones de la usabilidad, las cuales han sido propuestas por prestigiosas organizaciones y autores destacados en el tema. Además, se describen los atributos principales que deben ser considerados al evaluar la *usabilidad en dispositivos móviles*.

El estándar internacional *ISO/IEC 25010:2011* define dos modelos de calidad que proveen una terminología consistente para especificar, medir y evaluar sistemas y/o productos de calidad del software (ISO, 2011):

- Un *modelo de calidad del producto*, compuesto por ocho características relacionadas con las propiedades estáticas del *software* y propiedades dinámicas del sistema computacional. La Figura 2.1 y Figura 2.2. muestran las características contempladas en este modelo.

- Un *modelo de calidad en uso*, compuesto de cinco características relacionadas con el resultado de la interacción, cuando un producto es usado en un contexto particular.



Figura 2.1. Características del modelo de calidad de un producto Software.



Figura 2.2. Modelo de Calidad en Uso (ISO/IEC 25010:2011).

A su vez la ISO/IEC 25010:2011 define el término *usabilidad* como el grado en el cual un sistema puede ser usado con efectividad, eficiencia y satisfacción en un contexto específico de uso.

La *usabilidad* dentro del *modelo de calidad del producto* tiene a su vez seis subdivisiones:

- **Reconocimiento apropiado.** Grado en el cual el usuario puede reconocer si un producto o sistema es apropiado para sus necesidades.

- **Aprendizaje.** Grado en el cual un producto o sistema puede ser usado por un usuario para lograr metas específicas de aprendizaje, para usar el producto o sistema con efectividad, eficiencia, libre de riesgo y satisfacción en un contexto específico de uso.
- **Operatividad.** Grado en el cual un sistema tiene atributos que lo hacen fácil de operar y controlar.
- **Protección de error de usuario.** Grado en el cual un sistema protege a los usuarios de cometer errores.
- **Estética de la interfaz de usuario.** Grado en el cual una interfaz permite una interacción agradable y satisfactoria para el usuario.
- **Accesibilidad.** Grado en el cual un sistema puede ser usado por usuarios con un amplio rango de características y capacidades para lograr una meta en un contexto de uso determinado.

El *modelo de calidad en el uso* está compuesto de cinco características relacionadas con el resultado de la interacción cuando un producto es usado en un contexto particular. A continuación, se definen dichas características:

- **Efectividad.** Es la precisión y plenitud con que los usuarios alcanzan sus objetivos.

- **Eficiencia.** Se define como los recursos empleados en relación con la precisión y plenitud con la que los usuarios alcanzan sus objetivos.
- **Satisfacción.** Es el grado en el cual un usuario está satisfecho durante el uso del sistema en un contexto específico.
- **Libre de riesgo.** Es el grado hasta el cual un sistema mitiga el riesgo relacionado con lo económico, la vida humana, salud y el ambiente.
- **Cobertura del contexto.** Es el grado en el cual un producto puede ser usado con efectividad, eficiencia y satisfacción en todos los contextos de uso especificados, y en aquellos contextos que van más allá de los inicialmente establecidos.

La *usabilidad móvil* puede considerarse como una especialización o evolución del concepto de *usabilidad* aplicada a aplicaciones móviles. Los investigadores del campo de la Interacción Persona-Computador (IPO) han encontrado que, para producir sistemas computacionales con una adecuada *usabilidad*, es importante entender los factores psicológicos, ergonómicos, organizacionales y sociales que determinan como los usuarios trabajan (Kukulska-Hulme, 2007).

Jakob Nielsen ha estudiado la *usabilidad* desde 1993. Es el fundador del movimiento “*discount usability engineering*” (“Ingeniería de la usabilidad rebajada”) que destaca el uso de métodos eficaces para mejorar la calidad de las interfaces diseñadas para usuarios y ha aplicado sus métodos también en el área de los dispositivos móviles a través de investigaciones empíricas publicadas en su libro *Usabilidad en dispositivos móviles* (Nielsen & Budiu, 2013). Este autor explica la *usabilidad* en términos de la aceptación general de un sistema, que incluye la aceptación social y aspectos prácticos como confiabilidad, costo, compatibilidad y utilidad (Nielsen, 1993). En el 2012 define la *usabilidad* como “un atributo de calidad que evalúa lo fácil que resulta usar una *interfaz de usuario*”. La palabra *usabilidad* también se refiere a los métodos para mejorar la facilidad de uso durante el proceso de diseño (Nielsen, 2012).

Los atributos de *usabilidad* propuestos por (Nielsen, 1993) y su aplicación en el campo de los dispositivos móviles son:

- **Facilidad de aprendizaje.** Los sistemas deberán ser sencillos de aprender, incluso para usuarios que no tengan experiencia con un sistema o con los sistemas en general. El aprendizaje tiene relación con el tiempo que tarda un usuario en hacer un uso efectivo y completo de las funcionalidades que ofrece el sistema interactivo, aun cuando no haya tenido acceso con anterioridad al mismo. Esta característica se evalúa también en el caso de los dispositivos móviles, haciendo referencia tanto al aprendizaje de las aplicaciones como del propio equipo.

- **Eficiencia.** Tiene sentido hablar de este atributo cuando un usuario ya sabe usar un sistema, y hace referencia a la alta productividad en tareas diarias. En el caso de los dispositivos móviles es recomendable que se pueda personalizar la forma en que se presenta la información, si queremos que dicho diseño impacte en la eficiencia.
- **Memorización.** Hace referencia a que la *interfaz de usuario* debe favorecer el reconocimiento de las funcionalidades y el acceso a ellas, frente al recuerdo y memorización del uso del mismo. En el caso de los dispositivos móviles el reto está en el reducido tamaño de la pantalla, en la cual es necesario desplegar la información relevante, sin saturarla, para lo cual es recomendable, siempre que se pueda, reemplazar texto por iconos.
- **Tasa de errores.** Este atributo se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. En el caso de los dispositivos móviles, los errores suelen estar relacionados con los métodos de entrada, dado el reducido tamaño de la pantalla, el empleo de iconos de pequeño tamaño, y los inconvenientes que plantea la interacción táctil (propensión a mayor número de errores).
- **Satisfacción subjetiva.** Un usuario deberá estar satisfecho cuando usa un sistema, debe tener una experiencia agradable de uso, lo cual redundará en una opinión positiva hacia el mismo. La satisfacción del usuario depende principalmente de la

correcta funcionalidad del equipo y de que la aplicación cumpla con las necesidades del usuario.

Entre las técnicas más utilizadas para medir la *usabilidad* destaca el uso de *evaluaciones heurísticas*. Sin embargo, las existentes no están adaptadas a las características de los dispositivos móviles, ya que no consideran el contexto de uso y la movilidad. Por lo tanto, es necesario adaptar los métodos de evaluación a los aspectos relacionados con la variabilidad propia de un contexto móvil, así como a las particularidades de este tipo de dispositivos (Cuadrat Seix, Veloso, & Soler, 2012).

2.5. Framework para diseñar aplicaciones m-Learning

En esta sección se describe la propuesta desarrollada por Navarro, 2016. Dicho *framework* considera *aspectos pedagógicos*, la *usabilidad en dispositivos móviles* y aquellos aspectos que favorecen la adopción de este tipo de aplicaciones. El *framework* está dividido en dos categorías que corresponden con las fases de evaluación de la *usabilidad pedagógica* y de la *usabilidad de interfaz de usuario* o *tecnológica*, respectivamente (Figura 2.3).

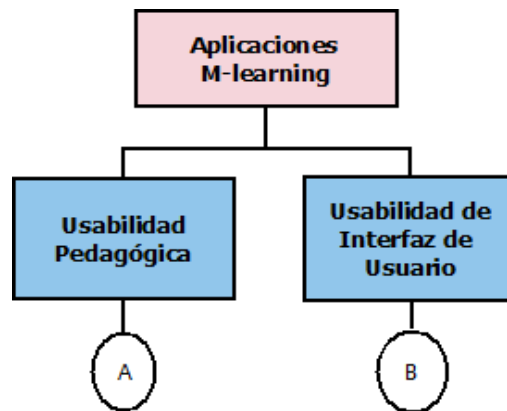


Figura 2.3. Marco conceptual de evaluación de sistemas *m-learning*.

Cada una de estas categorías, de mayor nivel, incluye varios aspectos y sus respectivos criterios de evaluación, que serán explicados en detalle en las siguientes secciones.

2.5.1 Usabilidad Pedagógica

Los entornos de *aprendizaje móvil* deben considerar diferentes aspectos educativos, pedagógicos y de usabilidad para poder facilitar y apoyar las actividades de aprendizaje. Estos factores proveerán el contexto apropiado para la práctica educativa. La dimensión *usabilidad pedagógica*, según el *framework* propuesto, se considerará dividida en cinco subdimensiones: *contenido*, *multimedia*, *tareas o actividades*, *interacción social* y *personalización*. Cada una de estas subdimensiones se dividirá, a su vez, en diferentes criterios que permiten definirlas (ver Figura 2.4), los cuales se explican detalladamente a continuación.

Contenido

Esta dimensión hace referencia al *contenido educativo* que se presenta en los dispositivos móviles. Dicho *contenido* debe considerar una lista de requerimientos pedagógicos que se utilizan para fomentar un aprendizaje eficaz, los cuales corresponden con los criterios que se indican a continuación:

- **Organización.** El contenido debe estar organizado en módulos o unidades pequeñas, su secuencia es importante y los temas claves deben tener prioridad.
- **Objetivos.** Estos deben ser definidos al principio de la secuencia didáctica. Deben estar formulados en un lenguaje sencillo y preciso. Los estudiantes deberán entender lo que se espera que aprendan después de completar un módulo, para así aumentar la probabilidad de que el aprendiz se comprometa con el aprendizaje.

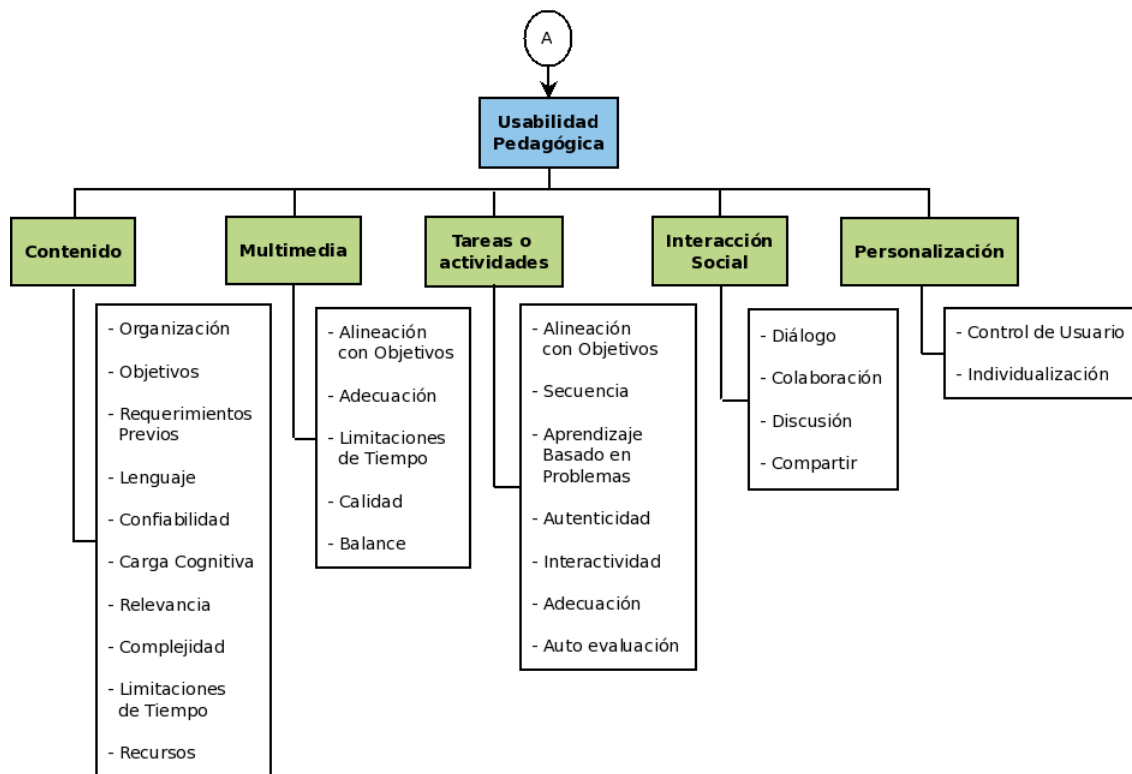


Figura 2.4. Aspectos de la *usabilidad pedagógica* y sus criterios de evaluación.

- **Requerimientos Previos.** El estudiante debe ser informado con anticipación acerca de las habilidades o conocimientos necesarios. Es necesario proveer enlaces a materiales previos o cursos relacionados a estos requerimientos.
- **Lenguaje.** El lenguaje debe ser sencillo, claro y apropiado al nivel de los estudiantes.
- **Confiabilidad.** La información debe estar actualizada y libre de errores.
- **Carga cognitiva.** Los contenidos deben estar divididos en porciones apropiadas, y de complejidad manejable, para que los estudiantes puedan procesarlos sin tener una

carga excesiva. La cantidad de nuevos conceptos claves debe limitarse a uno o dos, dependiendo de su complejidad.

- **Relevancia.** Los contenidos deben estar centrados en las características, intereses, motivaciones de los estudiantes, y deben ser útiles para su futuro.
- **Complejidad.** Los materiales de aprendizaje deben retar al estudiante, con un nivel de complejidad adecuado a su nivel conocimientos y sus destrezas.
- **Limitaciones de tiempo.** El *aprendizaje móvil* debe presentar contenidos educativos en unidades de corta duración, que requieran entre 30 segundos y 10 minutos (Hu, 2011).
- **Recursos.** Se deberá proporcionar acceso a recursos externos, apropiados para el contexto de aprendizaje, incluyendo enlaces a la *World Wide Web*. Además, el formato de presentación y la información debe estar adaptada para su visualización en dispositivos móviles.

Multimedia

Las aplicaciones *m-learning* deben soportar diferentes tipos de recursos multimedia, como video, audio, textos o animaciones, todas de alta calidad, desde la

óptica de los dispositivos móviles. Para ello se consideran los criterios que se indican a continuación:

- **Alineación con objetivos.** Los contenidos multimedia deben tener una estrecha conexión con los objetivos. Los videos, audios, e imágenes deberán ayudar a los estudiantes a alcanzar sus objetivos de aprendizaje.
- **Adecuación.** Los materiales multimedia deben ser presentados en el formato que más facilite el aprendizaje de conceptos.
- **Limitaciones de tiempo.** La duración de las animaciones, multimedia, videos y/o audios debe estar entre los 30 segundos y los 10 minutos (Hu, 2011).
- **Calidad.** El contenido multimedia debe tener buena calidad de video y fidelidad, incluir imágenes atractivas, y el tamaño de los archivos debe ser apropiado para el dispositivo.
- **Balance.** Debe existir una proporción equilibrada de recursos multimedia en el contenido. Los materiales deben presentar múltiples perspectivas del conocimiento y/o las tareas.

Actividades o tareas

Las tareas o actividades en un entorno de aprendizaje son una parte importante del proceso de aprendizaje. En este sentido, se consideran los siguientes criterios:

- **Alineación con objetivos.** Las tareas o actividades deben tener una conexión estrecha con los objetivos pedagógicos.
- **Secuencia.** Las tareas deben permitir a los alumnos integrar la nueva información con aprendizaje previo, para generar así nuevo conocimiento.
- **Aprendizaje basado en problemas.** Las tareas deben promover que los estudiantes comparen y clasifiquen información, para hacer deducciones y promover la creatividad.
- **Autenticidad.** Las tareas deben reflejar situaciones de la vida real, relevantes a la práctica profesional, generando interés e involucrando a los estudiantes. Deberán promover la transferencia de habilidades fuera del entorno de aprendizaje y el pensamiento crítico.
- **Interactividad.** Las tareas deben involucrar a los estudiantes en problemas a resolver en los que se aprovechen las ventajas de uso de la tecnología móvil (investigaciones de campo, toma de fotografías, videos, realidad aumentada, o uso de códigos QR, entre otros).

- **Adecuación.** Las tareas deben ser coherentes con el contenido y el nivel educativo.
- **Autoevaluación.** Un entorno móvil debe permitir la autoevaluación, que permita comprobar el nivel de aprendizaje alcanzado, así como registrar los avances del alumno.

Interacción social

La socialización es fundamental en el proceso de aprendizaje y un entorno *m-learning* debe promoverla y facilitarla. Para valorar esta dimensión se consideran los criterios que se indican a continuación:

- **Diálogo.** Un ambiente *m-learning* debe permitir a los estudiantes comunicarse con sus compañeros y profesores (chat, tablón de anuncios o redes sociales).
- **Colaboración.** Un entorno *m-learning* debe permitir el trabajo en grupo entre los estudiantes.
- **Discusión.** Un entorno de *m-learning* debe incorporar posibilidades de interacción, discusión y otras actividades colaborativas. La discusión requiere que los estudiantes

participen publicando acerca de su aprendizaje, para recibir retroalimentación, evaluar y aprender de otros grupos o estudiantes a través de trabajos publicados u opiniones.

- **Compartir.** Un entorno *m-learning* debe permitir a los estudiantes compartir fotos, videos o cualquier otro tipo de documentos relacionados con su trabajo, a través de redes sociales (como *facebook* o *twitter*).

Personalización

La personalización proporciona a los estudiantes la libertad de controlar diferentes opciones relacionadas con su estilo de aprendizaje o con respecto a la secuencia de estudio. Para valorar el grado de *personalización* se consideran los criterios siguientes:

- **Control de usuario.** Los estudiantes deben tener la libertad de dirigir su aprendizaje individual o en grupo, y así tener un sentido de pertenencia. Deberán elegir la secuencia a seguir, a través de rutas opcionales para su progreso.
- **Individualización.** Los estudiantes deberán elegir el material que más se adapte a sus estilos de aprendizaje (la información podrá ser revisada en videos, texto o audios).

Una vez descrita la categoría de *usabilidad pedagógica*, así como sus subdimensiones y criterios, se pasa a describir la categoría de *usabilidad de la interfaz de usuario* o *tecnológica* en la próxima sección.

2.5.2 Usabilidad de la interfaz de usuario

La *usabilidad de la interfaz de usuario* (en adelante interfaz) es fundamental para lograr la aceptación y satisfacción por parte de los estudiantes. En el contexto de los entornos de *aprendizaje móvil*, la interfaz debe ser fácil de usar para que los usuarios la aprendan, reconozcan y recuerden. En este *framework* se han identificado cinco subdimensiones: *diseño*, *navegación*, *customización*, *retroalimentación* y *motivación*, que a su vez, se dividen en diferentes criterios (ver Figura 2.5), que se explicarán en detalle a continuación.

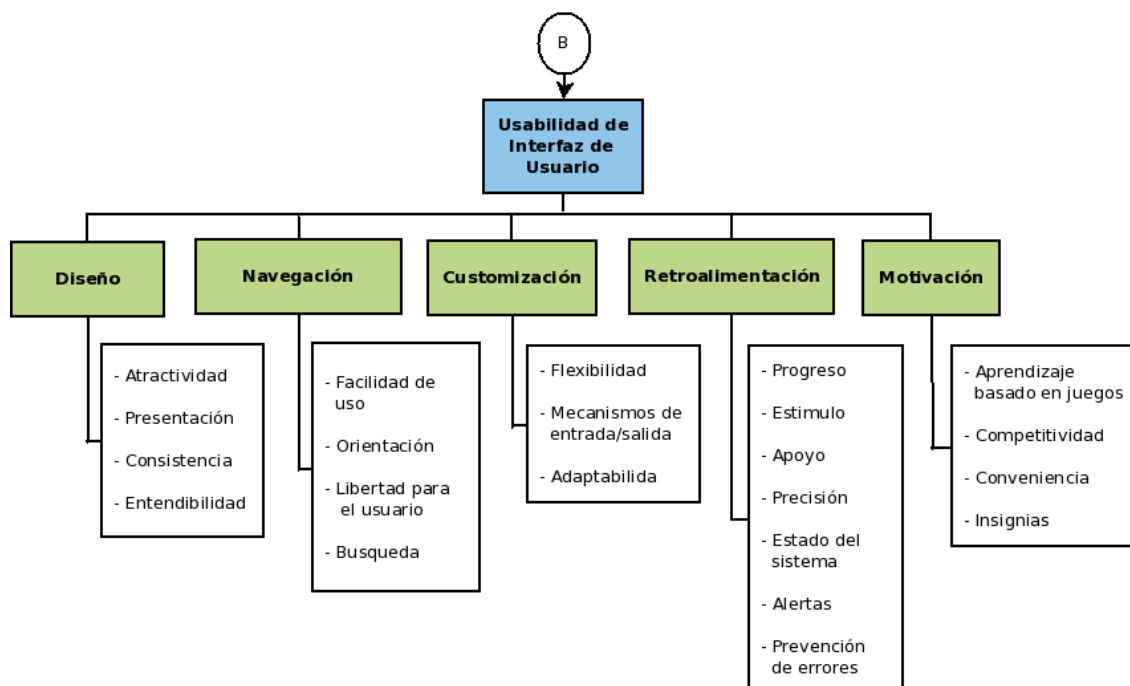


Figura 2.5. Aspectos y criterios de evaluación de la *usabilidad de la interfaz de usuario* o *tecnológica*.

Diseño de la interfaz

Un buen diseño es fundamental para mejorar la *usabilidad* de una interfaz de usuario. Para valorar este aspecto se utilizan los criterios que se indican a continuación:

- **Atractivo.** El diseño deberá ser estético y atractivo para los estudiantes. La apariencia deberá ser placentera y no debe contener información irrelevante. La interfaz no debe estar sobrecargada de información ni incluir gamas de colores que afecten negativamente a la percepción visual de los contenidos.
- **Presentación.** La interfaz debe seguir buenas pautas de presentación, con respecto a la organización (*layout*) y el diseño. Las opciones de selección deben ser visibles y fácilmente localizables. La información debe estar organizada en un orden lógico y las páginas y/o pantallas deben corresponder o adaptarse al tamaño de la pantalla.
- **Consistencia.** Los elementos de una interfaz deben ser estables y coherentes a través de todo el diseño. Deberá seguir estándares convencionales, con una selección consistente de estilo en cuanto al texto, botones y ventanas.

- **Entendibilidad.** Los iconos, textos y otros elementos deben ser simples e intuitivos, y escritos en un lenguaje común, para que los estudiantes puedan entender fácilmente los propósitos de las funciones del sistema.

Navegación

La navegación debe ser sencilla para el usuario. Además, el usuario debe estar constantemente informado de dónde está, dónde ha estado y hacia dónde puede ir. Esto se puede valorar por medio de los siguientes criterios:

- **Facilidad de uso.** La navegación debe incluir funciones que sean fáciles de entender, recordar y utilizar. Las opciones deben tener una correcta visibilidad. No se debe requerir más de tres clics para llegar desde la página o ventana de inicio hasta los contenidos.
- **Orientación.** El estudiante debe identificar claramente dónde está, cómo volver al menú principal y navegar fácilmente. Cada ventana o cuadro de diálogo debe desplegar su título.
- **Libertad para el usuario.** Cuando un estudiante elige una función por error, la aplicación debe permitir volver a ventanas anteriores. Opciones como *deshacer* y

rehacer son importantes para fomentar y mantener la libertad del usuario al interactuar con el sistema.

- **Búsqueda.** La aplicación debe ofrecer mecanismos para ayudar al usuario a encontrar contenido.

Customización

La customización es una adaptación del término inglés *customize*, que se refiere a modificar algo de acuerdo con las preferencias personales. Este aspecto es importante porque permite a los estudiantes cambiar el diseño o la navegación en sus dispositivos. Dicho subdimensión se valora haciendo uso de los siguientes criterios:

- **Flexibilidad.** La aplicación debe permitir a los estudiantes cambiar elementos de la interfaz. Debe proporcionar atajos y aceleradores, mejorando con esto la interacción e identificando acciones frecuentes. La flexibilidad hace posible la facilidad de uso para diferentes tipos de usuarios, desde novatos hasta expertos.
- **Mecanismos de entrada/salida.** Este criterio es esencial en el contexto móvil, ya que una de sus limitaciones es la interacción. Por lo tanto, las aplicaciones deben permitir diferentes formas de entrada y salida para mejorar la interacción del estudiante.

- **Adaptabilidad.** En un entorno de *aprendizaje móvil* el contenido debe adaptarse a las pantallas de diferentes dispositivos (*smartphones, tablets*, entre otros) sin la necesidad de modificar la configuración, evitando la superposición de objetos o la pérdida de información. Las pruebas y evaluaciones deberán estar adaptadas a las habilidades de los estudiantes.

Retroalimentación

La aplicación debe ayudar al estudiante a consolidar los conocimientos adquiridos y reforzar sus nuevas destrezas, durante la interacción en el proceso de aprendizaje. En esta dimensión, se utilizan los criterios que se indican a continuación:

- **Progreso.** La aplicación debe presentar el avance general del alumno en el curso y en cada unidad.
- **Estímulos.** La aplicación debe proporcionar retroalimentación constructiva, cuando los alumnos hayan tenido un avance significativo. Esto anima y ayuda a generar confianza.
- **Apoyo.** La aplicación debe proporcionar ayuda útil al estudiante para lograr sus objetivos de aprendizaje. Por ejemplo, si el estudiante comete un error al realizar una

tarea, la aplicación debe ofrecer oportunidades para dar con la respuesta correcta o, al menos, proporcionar alguna una explicación. También debe incluir mecanismos de comunicación para extender y ofrecer la retroalimentación de instructores, expertos, compañeros u otros.

- **Precisión.** La retroalimentación debe ser apropiada al contenido, problema o tarea, y debe ser proporcionada de forma inmediata.
- **Estado del sistema.** La aplicación debe presentar información general como la hora, el estado de la batería y la señal de conectividad a la red.
- **Alertas.** La aplicación debe proporcionar retroalimentación a través de alertas, recordando fechas límites, notificando eventos o actividades.
- **Prevención de errores.** Es más importante prevenir la aparición de errores que generar buenos mensajes de error. Por tanto, la aplicación debe presentar información contextual preventiva en el punto problemático o pedir confirmación de las acciones de los usuarios.

Motivación

La motivación busca aumentar el interés del estudiante en el proceso de aprendizaje a través de diferentes estrategias lúdicas y pedagógicas, aplicadas a la dinámica de las tareas. Se consideran los siguientes criterios:

- **Aprendizaje basado en juegos.** Cuando un estudiante aprende a través de juegos, desarrolla un incremento en la motivación e interés en el aprendizaje a través de la diversión.
- **Competitividad.** Las aplicaciones que soportan *m-learning* deben permitir la competitividad entre los estudiantes, a través de oportunidades como trabajar en la misma tarea al mismo tiempo, o mostrar tablas de liderazgo con las puntuaciones de cada estudiante.
- **Conveniencia.** Los estudiantes deberán considerar y valorar como útil aprender ciertos temas mediante *aprendizaje móvil*, frente al empleo de métodos más convencionales (como, por ejemplo, las lecciones magistrales y presenciales en el aula). Un entorno *m-learning* debe proveer una forma rápida y fácil de aprender un nuevo tema o de revisar temas anteriores.
- **Insignias.** La aplicación deberá generar insignias o recompensas simbólicas cuando el estudiante obtiene un logro significativo.

En este capítulo se ha realizado un análisis de una propuesta desarrollada para diseñar aplicaciones *m-learning*, así como aquellos trabajos que proponen heurísticas de *usabilidad en dispositivos móviles*. Como consecuencia de este análisis, se identificaron una serie de requisitos pedagógicos y de usabilidad, que deberían ser tenidos en cuenta a la hora de desarrollar aplicaciones móviles educativas utilizando la herramienta propuesta en esta tesis.

Capítulo 3. Diseño e implementación de una herramienta para desarrollar aplicaciones móviles educativas multiplataformas

En este capítulo se describen las principales características del análisis de requisitos y del diseño de la herramienta que ha sido desarrollada para crear aplicaciones educativas para dispositivos móviles. Entre estas características se encuentran los requerimientos funcionales, requerimientos no funcionales, diagrama de arquitectura, diagrama de clases y el diagrama de base de datos. Además, se presentan los resultados de la implementación de la herramienta propuesta.

3.1. Programación en capas

La programación por capas es un modelo de desarrollo de Software, en el cual su objetivo primordial es la separación (*desacoplamiento*) de las partes que componen un sistema de Software (Wikipedia, 2013). Por ejemplo, en un sistema con arquitectura cliente-servidor se considerará la lógica de negocios, la capa de presentación y la capa de datos. De esta forma, será más sencillo crear y mantener las interfaces de usuario (capa de presentación) del mismo sistema, sin necesidad de hacer cambios en la capa de datos o lógica

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo afectará al nivel requerido sin tener que revisar en el código fuente de otros módulos, dado que se habrá reducido el acoplamiento informático hasta una interfaz de paso de mensajes. Además, permite distribuir el desarrollo de una aplicación por niveles; de este modo, cada grupo de trabajo

está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

Actualmente, en el diseño de sistemas informáticos se suelen usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). A continuación, se describen brevemente las distintas etapas de este tipo de programación:

- **Capa de Dominio:** En esta capa se encuentran todas las definiciones de nuestra aplicación en forma de clases.
- **Capa de Datos:** Esta capa recibe y envía las peticiones de la base de datos, cada vez que algún componente necesita un dato. Esta capa será la que haga la operación necesaria.
- **Capa de Negocio:** En esta capa se encuentran todas las reglas de negociación. Estas reglas son estatutos que definen el comportamiento de la aplicación, por ejemplo, la fecha de nacimiento de una persona no puede ser mayor al día de hoy.
- **Capa de Aplicación:** Esta capa se encarga de mostrar al cliente los datos, y es la encargada de recibir los datos que el cliente ingresa. Se define cliente como la entidad que interactúa con nuestra aplicación (persona/servicio).

3.2 Los servicios Web

El World Wide Web Consortium (W3C) define un servicio Web como un sistema de Software designado para dar soporte a la interacción de máquina a máquina interoperativa a través de una red (IBM, 2014). Este tipo de servicios realizan una tarea específica o un conjunto de tareas, y se describe mediante una descripción de servicio en una notación XML estándar llamada WSDL (Web Services Description Language). La descripción de servicio proporciona todos los detalles necesarios para interactuar con el servicio, incluidos los formatos de mensaje (que detallan las operaciones), los protocolos de transporte y la ubicación.

Con otras palabras, un servicio se puede definir como la entidad que permite enviar y recibir datos a través de nuestras aplicaciones (estos están basados en una API REST) mediante JSON. En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese Software. Twitter, YouTube, los sistemas de identificación con Facebook, y otras empresas generan negocios gracias a REST y las APIs REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de APIs para servicios de Internet. Básicamente, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON (BBVA, 2016).

Por otra parte, los archivos JSON (JavaScript Object Notation) provienen de un estándar de la Web que permite interactuar entre distintos lenguajes de manera que puedan ser transformados a objetos nativos del lenguaje.

3.3 Requerimientos del Software

Un documento de requerimientos de Software es el lugar donde se describen las características y requisitos de un Software, producto, programa o conjunto de programas. Los requisitos se expresan en lenguaje natural, sin consideraciones ni términos técnicos (pmoinformatica, 2018).

Hasta hace poco los requisitos de un sistema interactivo solo hacían referencia al componente funcional, dejando otras connotaciones fuera del alcance del sistema. Afortunadamente, hoy en día los aspectos relacionados con los usuarios y con el uso de los sistemas acaparan mayor grado de interés.

A continuación, se presentan los requerimientos obtenidos para la herramienta de Software que presenta en esta tesis de licenciatura:

3.3.1 Requerimientos Funcionales del sistema propuesto

En la Tabla 3.1 se presentan los requerimientos considerados en nuestro sistema, estos tienen como característica que no interactúan con un usuario en particular, sino que a través de mensajes con otras capas superiores se encarga de obtener, registrar, eliminar

o actualizar datos guardados, así como también poder enviar mensajes a una capa superior para poder mostrarlos a un usuario.

Tabla 3.1: Requerimientos Funcionales del Sistema propuesto

Id	Requerimientos	Patrón Arquitectónico	Lenguaje Sugerido	Descripción
RF1	El sistema debe mostrar una lista de módulos que representan unidades de aprendizaje.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de obtener todos los módulos de un curso que esta almacenado en la base de datos.
RF2	El Sistema debe poder crear módulos.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de crear nuevos módulos a un curso.
RF3	El sistema debe poder eliminar módulos.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de eliminar cualquier módulo de un curso.
RF4	El sistema debe poder modificar o	Layers / Arquitectura	C#	Este requerimiento se refiere a la capacidad de cambiar el nombre,

	actualizar módulos.	orientada a servicios		la descripción o cualquier información del módulo.
RF5	El Sistema debe poder agregar información a los módulos.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de agregar información respecto un tema en algún módulo creado.
RF6	El sistema debe poder agregar videos a los módulos.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de incluir videos en los módulos.
RF7	El sistema debe poder agregar preguntas a los módulos.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de añadir preguntas sobre la información de algún módulo, con el fin de evaluar el aprendizaje.
RF8	El sistema debe permitir contestar las preguntas al estudiante.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a la capacidad de generar preguntas aleatoriamente para que los estudiantes las respondan y poder evaluar el aprendizaje.

RF9	El sistema debe poder agregar posibles respuestas a cada una de las preguntas del módulo, especificando la respuesta correcta.	Layers / Arquitectura orientada a servicios	C#	Este requerimiento se refiere a que a cada una de las preguntas se le deberá permitir un conjunto de respuestas posibles e indicar cual es la correcta, con el fin de que el sistema pueda evaluar de manera automática.
-----	--	---	----	--

3.3.2 Requerimientos funcionales del administrador

En esta sección se presentan los requerimientos de tipo administrador, definido como el usuario capaz de tener el control total sobre la aplicación. En una perspectiva de negocio este usuario controla el flujo completo de uso, es decir, es la persona con los privilegios de subir el contenido temático, agregar información de los módulos, incluir las preguntas, videos y otros requerimientos que serán descritos en la Tabla 3.2.

Tabla 3.2: Requerimientos del sistema para el Administrador del sistema

Id	Requerimiento	Patrón Arquitectónico	Lenguaje Sugerido	Descripción
----	---------------	-----------------------	-------------------	-------------

RF10	Este usuario debe poder crear módulos.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad que tiene el usuario de crear módulos a través de una interfaz.
RF11	Este usuario debe poder consultar todos los módulos.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad que tiene el usuario de ver todos los módulos a través de una interfaz.
RF12	Este usuario debe poder ver cada módulo	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad que tiene el usuario de ver el contenido de cada módulo a través de una interfaz.
RF13	Este usuario debe poder eliminar módulos	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad que tiene el usuario de poder eliminar cualquier módulo a través de una interfaz.
RF14	Este usuario debe poder actualizar módulos.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de cambiar cualquier campo modificable

				del módulo a través de una interfaz.
RF15	Este usuario debe poder agregar videos al módulo.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de poder incluir videos a los módulos.
RF16	Este usuario debe poder agregar preguntas a los módulos.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de incluir preguntas en los módulos a través de una interfaz.
RF17	Este usuario debe poder ver preguntas.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de visualizar todas las preguntas que existen en la base de datos a través de una interfaz.
RF18	Este usuario debe poder eliminar preguntas.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de eliminar preguntas mediante una interfaz.
RF19	Este usuario debe poder agregar respuestas.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de poder agregar respuestas a

				cualquier pregunta mediante una interfaz.
RF20	Este usuario debe poder eliminar respuestas	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de eliminar respuestas de cualquier pregunta mediante una interfaz.
RF21	Este usuario debe poder indicar respuestas correctas.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de indicar cuales son las respuestas correctas, con el fin de que el sistema genere la evaluación automáticamente.
RF22	Este usuario debe poder asignar preguntas a módulo.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de asignar posibles preguntas a cualquier módulo
RF23	Este usuario debe poder visualizar las preguntas dentro de un módulo	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de visualizar las preguntas que existen en cada módulo.

RF24	Este usuario debe poder visualizar las respuestas dentro de una pregunta.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere a la capacidad de visualizar las respuestas en cada una de las preguntas.
RF25	Este usuario debe poder eliminar preguntas dentro de un módulo.	Modelo - Vista - Controlador	C#	Este requerimiento se refiere al hecho de eliminar posibles preguntas dentro de cualquier módulo
RF26	Este usuario debe poder eliminar respuestas dentro de un módulo	Modelo - Vista - Controlador	C#	Este requerimiento se refiere al hecho de eliminar posibles respuestas dentro de un módulo

3.3.3 Requerimientos funcionales del sistema para el Usuario Final (estudiante).

En la Tabla 3.3 se listan los requerimientos del sistema correspondiente al Usuario Final (estudiante). Estos requerimientos corresponden a la interacción de los estudiantes con el contenido educativo almacenado en el sistema.

Tabla 3.3: Requerimientos del sistema correspondientes al Usuario Final (estudiante)

Id	Requerimiento	Patrón Arquitectónico	Lenguaje Sugerido	Descripción
RF25	El estudiante debe poder consultar todos los módulos desde un móvil.	Modelo - Vista - Controlador / Connected App	Javascript / Ionic Framework	Este requerimiento especifica que el usuario final podrá ver todos los módulos disponibles de un curso a través de un dispositivo móvil tipo Smartphone o Tablet.
RF26	El estudiante debe poder ingresar al contenido de un módulo desde un móvil.	Modelo - Vista - Controlador / Connected App	Javascript / Ionic Framework	Este requerimiento se refiere a que el estudiante debe poder elegir y ver el contenido de un módulo a través de un dispositivo móvil tipo Smartphone o Tablet.
RF27	Desde un móvil se puede ver la información de los módulos	Modelo - Vista - Controlador / Connected App	Javascript / Ionic Framework	El siguiente requerimiento es para el usuario final el cual podrá ver la información que tiene cada módulo

RF28	El estudiante debe poder reproducir los videos de los módulos desde el móvil.	Modelo - Vista - Controlador / Connected App	Javascript / Ionic Framework	Este requerimiento se refiere a que el estudiante podrá reproducir los videos desde un dispositivo móvil tipo Smartphone o Tablet.
RF29	El estudiante debe poder responder las preguntas de los módulos desde un móvil.	Modelo - Vista - Controlador / Connected App	Javascript / Ionic Framework	Este requerimiento se refiere a que el estudiante podrá responder a las preguntas generadas por el sistema a través de un dispositivo móvil tipo Smartphone o Tablet, con el fin de evaluar el aprendizaje de alguno de los módulos.
RF30	El estudiante podrá visualizar el resultado de sus evaluaciones desde el móvil.	Modelo - Vista - Controlador / Connected App	Javascript / Ionic Framework	Este requerimiento especifica que el estudiante podrá visualizar el resultado de

				su evaluación después de responder a las preguntas generadas en alguno de los módulos, con el fin de evaluar su aprendizaje.
--	--	--	--	--

3.3.4 Requerimientos no funcionales

En la Tabla 3.4 se enlistan los requerimientos no funcionales del sistema propuesto, estos conforman las necesidades en cuanto a características de funcionamiento en lugar de funciones a realizar, en inglés, se les conoce como *-ilitys*, cuya traducción en español sería *-ilidad*, debido a que se enfocan en atributos como lo puede ser portabilidad, usabilidad, entra otras.

Tabla 3.4: Requerimientos no funcionales

Id	Requerimiento	Descripción	Prioridad
RNF1	Usabilidad	El sistema debe considerar las características relacionadas con la usabilidad de los sistemas, para que sea útil, fácil y agradable para el usuario	5
RNF2	Portabilidad	El sistema debe ser utilizado por diferentes tipos de dispositivos móviles, tales como Smartphones o Tablets.	5

RNF3	Multiplataforma	Debe ser capaz de utilizarse desde dispositivos móviles Android y iOS.	4
RNF4	Rendimiento	Debe poder manejar una gran cantidad de información debido a todos los módulos posibles	4
RNF5	Desempeño	El Sistema debe soportar gran cantidad de información, ya que es posible que existan varios cursos almacenadas.	4
RNF6	Mantenibilidad	El Sistema debe de ser fácil de mantener, es decir, hacerle revisiones y actualizaciones periódicamente de una manera sencilla.	3
RNF7	Disponibilidad	La información de los cursos debe estar disponible en un porcentaje alto del tiempo.	4
RNF8	Modularidad	El Sistema debe manejarse de manera modular para que sea fácil de conectarse o desconectarse con otros elementos.	3

3.4 Diagrama de arquitectura

Un diagrama de arquitectura es una representación gráfica del funcionamiento de nuestros sistemas a través de cajas y flechas, cada caja representa un subsistema o una funcionalidad y cada flecha representa en qué dirección existirá la comunicación. Esto

nos permite poder identificar de una manera sencilla que elementos interactúan entre sí y donde deben existir flujos de trabajo, en este caso tenemos tres sistemas los cuales van a estar interactuando entre sí y una base de datos para el almacenamiento de la información (ver Figura 3.1).

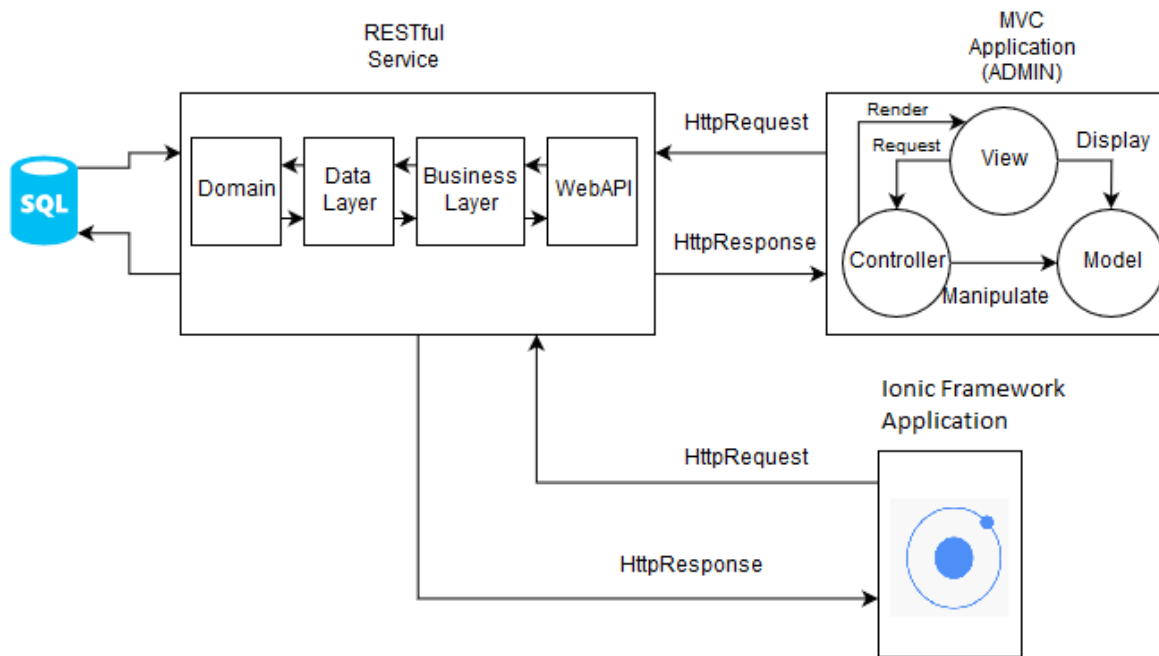


Figura 3.1. Diagrama de arquitectura del sistema propuesto.

A continuación, se describe brevemente cada uno de los módulos de nuestra arquitectura:

1. **Base de datos SQL:** Es nuestra instancia de la base de datos y es la que permitirá almacenar y proporcionar toda la información reunida en nuestro sistema. Esta base de datos será creada en SQL Server, aprovechando la oportunidad de integración nativa con el lenguaje de programación C#. Como su nombre lo dice está basado en

SQL (Structured Query Langague) el cual nos permite hacer consultas de una manera sencilla.

2. **RESTful Service:** Nuestra instancia de SQL estará conectada a una aplicación hecha en C#, la cual consiste en una serie de servicios Web y controla la lógica y reglas de negocio de nuestra aplicación. Estos servicios, basados en la arquitectura en capas mencionada anteriormente, nos dan la posibilidad de interactuar a través de distintos sistemas, proporcionando capas de protección y de interacción a nuestra base de datos.
3. **MVC Application (ADMIN):** Aplicación realizada con ASP.NET, la cual se encargará de enviar y consumir todos los aspectos del lado administrativo de la aplicación:
 - a. Agregar preguntas
 - b. Actualizar preguntas
 - c. Eliminar preguntas
 - d. Agregar módulos
 - e. Actualizar módulos
 - f. Eliminar módulos

MVC significa "Modelo – Vista – Controlador". Este tipo de aplicaciones consta de un modelo de datos, información de presentación y de información de control. El patrón

requiere que cada uno de estos elementos esté separado en distintos objetos (IBM, Knowledge Center, 2018).

A continuación se describe cada uno de estos elementos:

- El **modelo** contiene únicamente los datos puros de la aplicación y no contiene la lógica que describe cómo pueden presentarse los datos a un usuario.
- La **vista** presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.
- El **controlador** está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la acción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

La mayoría de las aplicaciones hoy en día siguen este patrón, muchas con ligeras variaciones. Por ejemplo, algunas aplicaciones combinan la vista y el controlador en una clase porque ya están estrechamente unidos. Todas las variaciones recomiendan enérgicamente la separación de los datos de su presentación. Esto no sólo simplifica la estructura de una aplicación, sino que también permite reutilizar el código.

4. **Ionic Framework Application:** Esta aplicación está basada con el *Framework* de JavaScript Ionic, el cual nos permite realizar un componente Web, y exportarlo a distintas plataformas para poder hacer una aplicación que corra en dispositivos móviles. La aplicación permite instalarse en dispositivos iOS y Android, y nos permitirá ser la aplicación con la cual se consumirá principalmente nuestros servicios. Este componente manipula los módulos, preguntas y respuestas por parte del estudiante, de manera que solo sean de tipo “solo lectura”.

En general, la arquitectura propuesta permite que la base de datos “SQL” le dé servicio a nuestro Sistema Web maestro “RESTful Service”. La conexión se hará a través del lenguaje de programación C# y la conexión a la base de datos por medio de Microsoft SQL Server. La primera capa con la que interactúa la base de datos es con la de *dominio*, identificando la entidad que se le está solicitando. Después esto pasa a través de la *capa de datos* hacia la *capa de negocio*, validando con las reglas de negocio la información correspondiente y verificando que sea válida y correcta, y así poder exponerla o introducirla mediante nuestro Servicio Web (WebAPI) mediante el equivalente de datos

en JSON. Tanto el flujo de la aplicación MVC y la *aplicación Ionic* se exponen mediante HTTP, de esta manera se generaliza el flujo y se puede exponer hacia cualquier aplicación. En el caso de MVC se consume el controlador para poder ver el contenido deseado modificando el modelo. Ionic framework funciona generalmente de la misma manera en MVC, esto mediante las bibliotecas correspondientes de Angular.

3.5 Diagrama de Base de datos.

En la Figura 3.2 se muestra el diagrama de Base de datos de la herramienta propuesta, la cual está compuesta de ocho tablas que nos permite establecer las definiciones que tendrá nuestra aplicación, al igual que otras clases que definiremos posteriormente. A continuación, se describe cada una de estas 8 tablas:

- **Questions** (Tabla de preguntas): En esta tabla se guardarán las preguntas definidas por el administrador de la aplicación.
- **Answers** (Tabla de respuestas): Esta tabla almacena las respuestas del usuario.
- **Modules** (Tabla de módulos): Esta tabla almacena las preguntas e información correspondientes de cada módulo.
- **Status** (Tabla de estatus). Esta tabla define los posibles estatus de los módulos.
- **Options** (Tabla de opciones). Esta tabla contiene las opciones o posibles respuestas de cada pregunta.
- **QuestionTypes** (Tabla de tipos de pregunta). Esta tabla contiene los tipos de pregunta, estas pueden ser tipo Abierta, Opción Múltiple, entre otras).

- **ModuleAnswers:** Esta tabla representa un catálogo que contiene la relación de los módulos y sus respuestas.
- **QuestionModules:** Esta tabla representa otro catálogo que contiene la relación de preguntas con los módulos.

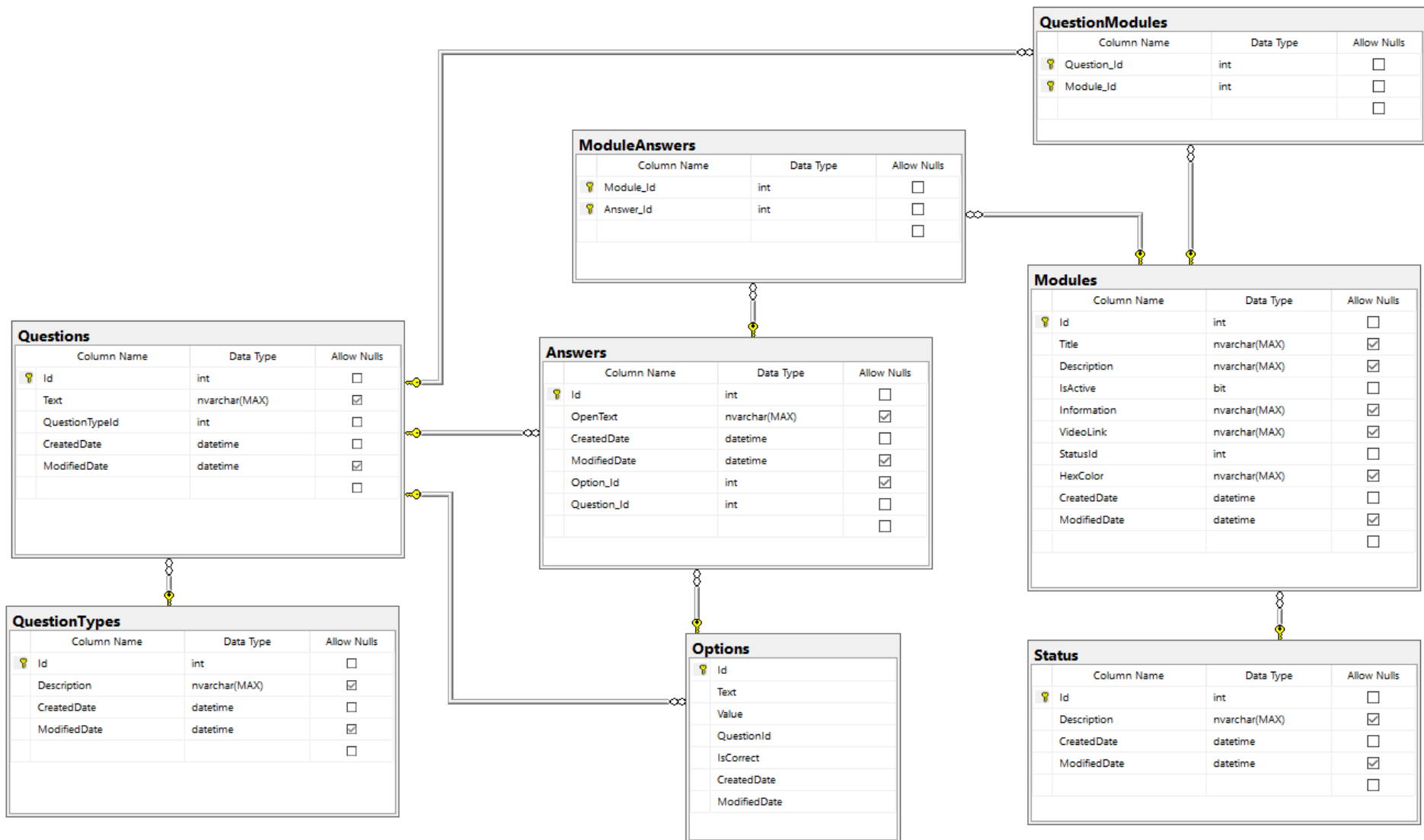


Figura 3.2. Diagrama de base de datos del sistema propuesto.

3.6 Patrones de Diseño

Un Patrón de Diseño (en inglés Design Pattern) consiste en un Diagrama de Objetos que ofrece una solución a un problema frecuente. El Diagrama de Objetos está constituido por un conjunto de objetos descritos por clases y las relaciones que enlazan los objetos. Los patrones responden a problemas de diseño de aplicaciones en el marco de la Programación Orientada a Objetos. Se trata de soluciones conocidas y probadas cuyo diseño proviene de la experiencia de los programadores. No existe un aspecto teórico en los patrones, en particular no existe una formalización (a diferencia de los algoritmos). Los Patrones de Diseño están basados en las buenas prácticas de la Programación Orientada a Objetos.

Como se mencionó en las secciones anteriores es necesario separar la lógica del negocio y el acceso a base de datos, y esto lo vamos a lograr a través mediante el patrón repositorio. Este patrón se encarga de separar la lógica mediante la cual se accede a los datos almacenados para ser mapeados al modelo de negocio o dominio de aplicación, realizar consultas a la base de datos o impactar nuestra entidad del modelo al origen de datos utilizados. Esto nos permite que la capa de negocio sea agnóstica en cierta forma de la tecnología de almacenamiento que se utilice (Granata N., 2017).

A continuación, se describen los patrones de diseño utilizados en el sistema propuesto:

- **Diseño por contrato:** El Diseño por Contrato es una metodología para el diseño e implementación de aplicaciones y componentes popularizada por el lenguaje de programación Eiffel. Consiste en considerar los elementos de diseño como participantes de una relación similar al contrato de negocios. Así, se pueden diseñar los componentes asumiendo que se cumplirán ciertas condiciones de entrada (pre-condiciones), mientras que se deberán verificar ciertas condiciones de salida (post-condiciones), así como la invariante de clase (propiedades que se mantienen invariantes a pesar del procesamiento realizado por el componente) (Wikipedia, 2019).
- **Singleton:** Permite asegurar que cada clase concreta existe en una única instancia y proporciona un método único que la devuelve (Debrauwer L., 2013).
- **Decorators:** Permite agregar dinámicamente funcionalidades suplementarias a un objeto (Debrauwer L., 2013).
- **MVC** (Modelo – Vista – Controlador) nos permite separar nuestra aplicación en tres secciones, separando la lógica de la aplicación de la vista, de manera que tenga que usar siempre un controlador para manipular cualquier aspecto del modelo.

3.7 Implementación de la herramienta

En esta sección se describe el proceso relacionado con la etapa de Implementación del sistema propuesto, la cual consta de tres proyectos interconectados:

1. **WebService:** Este proyecto contiene el Servicio Web que se ha desarrollado y servirá como plataforma para recibir todas las peticiones que se necesiten atender utilizando la base de datos. Este proyecto no interactúa con el usuario, si no que los demás proyectos interactúan con él.

Esta solución cuenta con cinco proyectos y representan las capas mencionadas en la arquitectura del sistema:

- **Capa de dominio (Domain):** Esta capa tiene el modelo del sistema propuesto, la cual hace referencia a la definición de nuestras entidades (ver Figura 3.3).

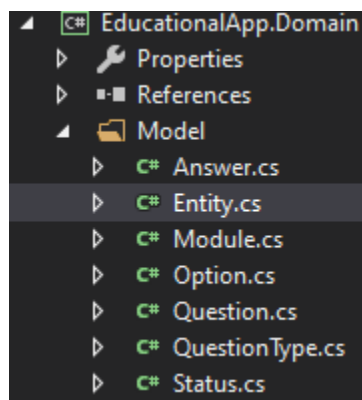


Figura 3.3. Clases que pertenecen a la capa de Dominio del sistema

Como se puede observar en esta figura esta capa solo tiene la referencia al modelo descrito anteriormente.

- **Capa de datos (Data):** Esta capa tiene como objetivo hacer que nuestra conexión con la base de datos funcione. Se eligió el Patrón de Repositorio para este cometido, de manera que nuestras tablas sean repositorios de datos y puedan ser manipuladas a través de las interfaces.

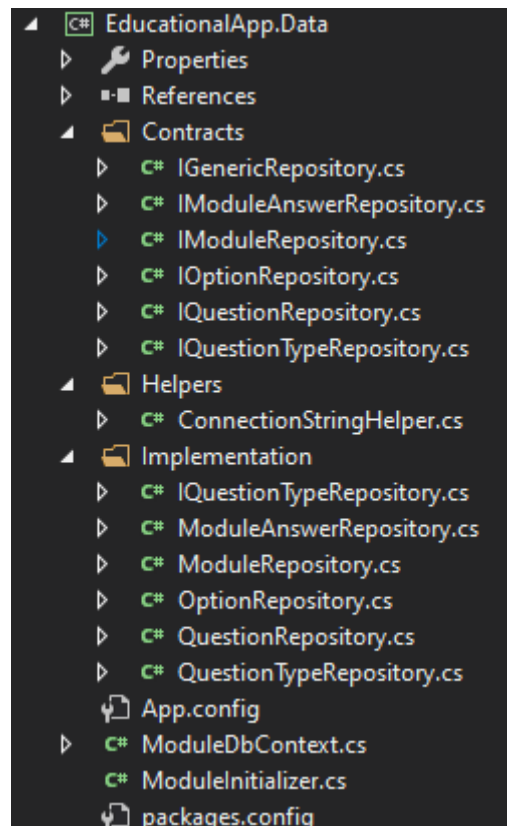


Figura 3.4 Capa de Datos del Sistema

Como se puede observar en la Figura 3.4 existen tres folders principales, las cuales se describen brevemente a continuación:

- **Contracts:** Esta carpeta contiene los contratos (Interfaces) que nos servirán para consumir nuestros datos.
 - **Helpers:** Esta carpeta contiene el "Helper" o ayudante que nos sirve para conectarnos con nuestra base de datos y define una cadena de conexión.
 - **Implementation:** Esta carpeta hace referencia a la implementación de la capa de datos, y contiene la lógica correspondiente para poder hacer las operaciones necesarias.
-
- **Capa de negocio (Business):** Esta capa contiene la lógica de negocio de nuestra aplicación. Como se puede observar en la Figura 3.5 esta capa tiene los contratos necesarios para la aplicación y la implementación de sus reglas de negocio.

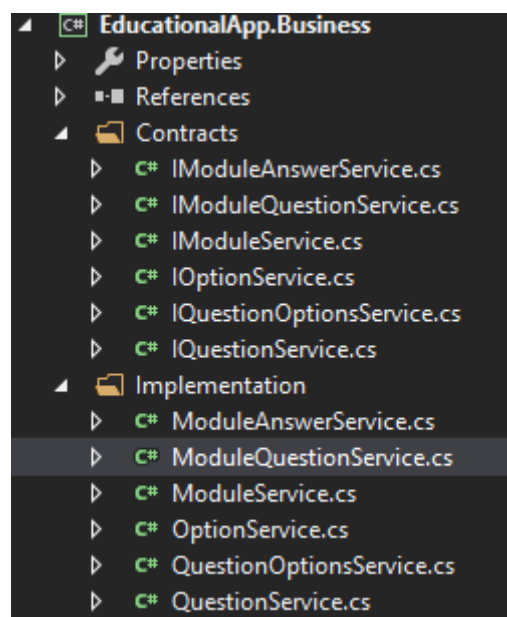


Figura 3.5: Capa de negocio del Sistema

- **Capa de Enumerables:** Esta capa contiene la definición de los enumerables, los cuales representan una constante numérica a través de una palabra usados a lo largo de la aplicación (ver Figura 3.6).

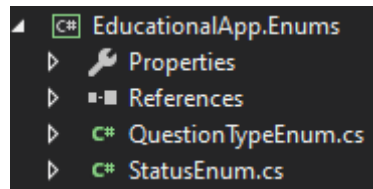


Figura 3.6: Archivos de Enumerables

- **Capa de Servicio. (WebAPI):** Esta capa tiene la funcionalidad de exponer nuestra aplicación para que sea utilizada a través de HTTP y REST (ver. Figura 3.7)

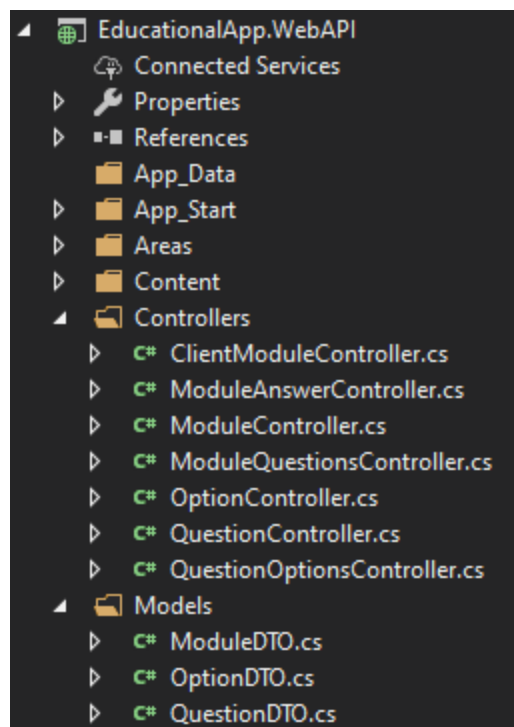


Figura 3.7: Capa de servicio del Sistema.

Las áreas principales de esta capa son los controladores “Controllers”, los cuales contienen la lógica de conexión con las capas inferiores y el servicio a la capa de aplicación/presentación.

Se generó un nuevo proyecto en C# usando su *Framework* para el manejo de Modelo – Vista – Controlador, de esta manera desacoplamos la presentación/aplicación de la capa que da servicio a los datos. La Figura 3.8 muestra las clases del proyecto en sus distintos niveles:

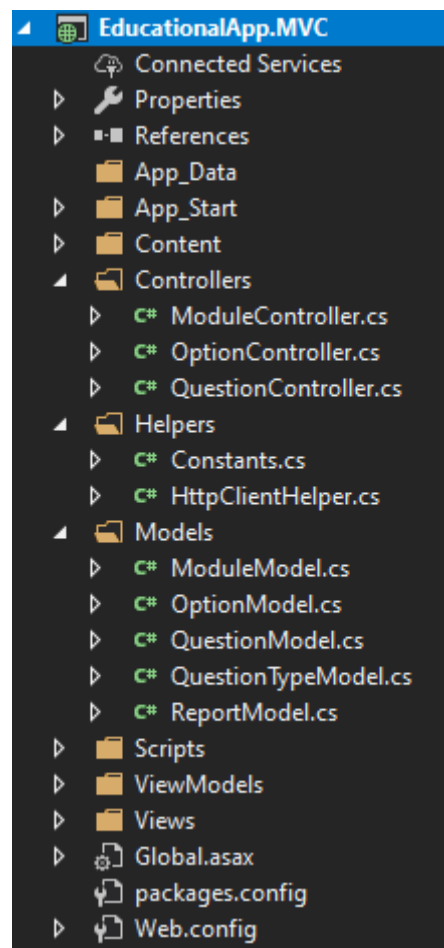


Figura 3.8. Clases del proyecto MVC

Existen principalmente cuatro carpetas las cuales contienen la lógica de nuestra aplicación:

- **Models:** Esta carpeta contiene la información de los modelos utilizados.
- **Controllers:** Esta carpeta contiene la lógica que utiliza la vista para interactuar con cualquier elemento.
- **Helpers:** Esta carpeta tiene los "Ayudantes" de nuestra aplicación, en este caso existe una clase para las constantes y otra clase para hacer conexiones con el HTTP.
- **Vistas:** Esta carpeta tiene las interfaces de usuario de nuestra aplicación.

2. Proyecto Móvil realizado con Ionic Framework (app cliente)

Este proyecto contiene la aplicación en Ionic Framework que será el cliente de nuestra aplicación Web, es decir, la vista de la aplicación móvil.

Este proyecto contiene archivos de configuración, archivos de lógica, así como bibliotecas para poder hacer uso del *Framework*, pero en este caso se hará uso principalmente de dos carpetas.

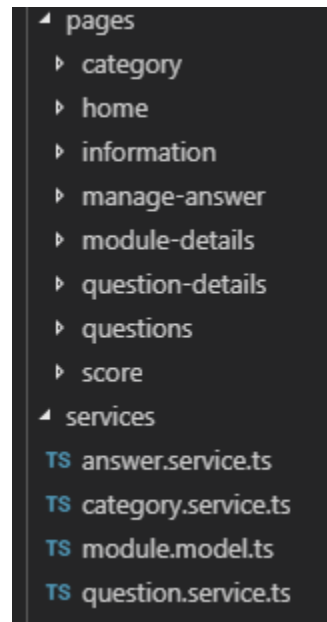


Figura 3.9 Estructura del proyecto de Ionic

La sección *Pages* contiene la vista de nuestra aplicación (HTML / CSS / JS) (ver Figura 3.9) y en la carpeta de *Services* los archivos que se encargan de la lógica de servicio a nuestra aplicación. Cabe mencionar que los archivos son *.ts*, debido al *framework* que utiliza como base Angular y el lenguaje de programación es *Typescript*.

3.8 Interfaces de Usuario

Las siguientes pantallas muestran el sistema Web, el cual utiliza el encargado de hacer la parte administrativa de los módulos, contenidos, preguntas, entre otros.

En la Figura 3.10 se muestra la interfaz donde se presentará el listado de los módulos creados y su respectiva información.

Todos los módulos

[Crear nuevo módulo](#)

Título:	Descripción:	Fecha de Creación:	Fecha de Modificación:	Estatus:	¿Esta habilitado?
---------	--------------	--------------------	------------------------	----------	-------------------

© 2019 - Universidad Autónoma de Baja California

Figura 3.10: Interfaz en la cual se enlistan los módulos creados por el administrador.

En la Figura 3.11 se muestra la interfaz para crear un nuevo módulo, la cual se accede con la opción “crear módulos”. Como se puede observar contiene cinco cajas de textos, las cuales el administrador es el encargado de proporcionar la información o contenido que se mostrará en la aplicación Web:

- **Título:** Es necesario especificar el nombre del módulo a incluir en la aplicación.
- **Descripción:** Es necesario escribir un breve resumen del módulo.
- **Información:** Hace referencia al contenido educativo que pertenece al módulo a crear.
- **Enlace al video:** Se especifica el enlace del video a añadir dentro de un módulo.
- **Color del módulo:** Color que se le utilizará en la interfaz de la aplicación móvil.

Crear un nuevo modulo

Título:



Descripción:

Información:



Link de video:

Color del modulo



© 2019 - Universidad Autonoma de Baja California

Figura 3.11. Formulario para crear un nuevo módulo.

La Figura 3.12 muestra la pantalla de la aplicación Web, pero esta vez con el módulo que fue creado previamente. Esta vista nos proporcionara una tabla con todos los módulos creados e incluidos en la base de datos.

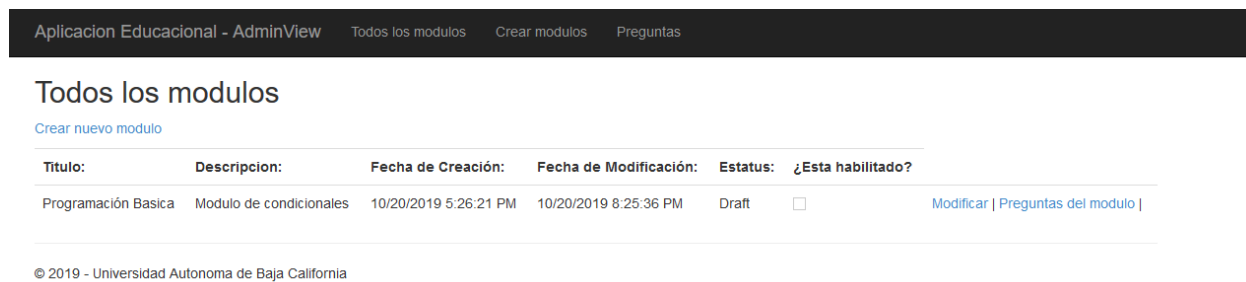


Figura 3.12. Interfaz que enlista un módulo creado.

La Figura 3.13 muestra la interfaz en la cual se crean las preguntas. Esta interfaz nos muestra las preguntas que existen en nuestra base de datos, las cuales se pueden utilizar en distintos módulos o unidades del curso. También contiene la opción para crear una pregunta sin necesidad de cambiarse de pantalla, de manera que el administrador pueda buscar su pregunta, y si no existe podrá agregarla.

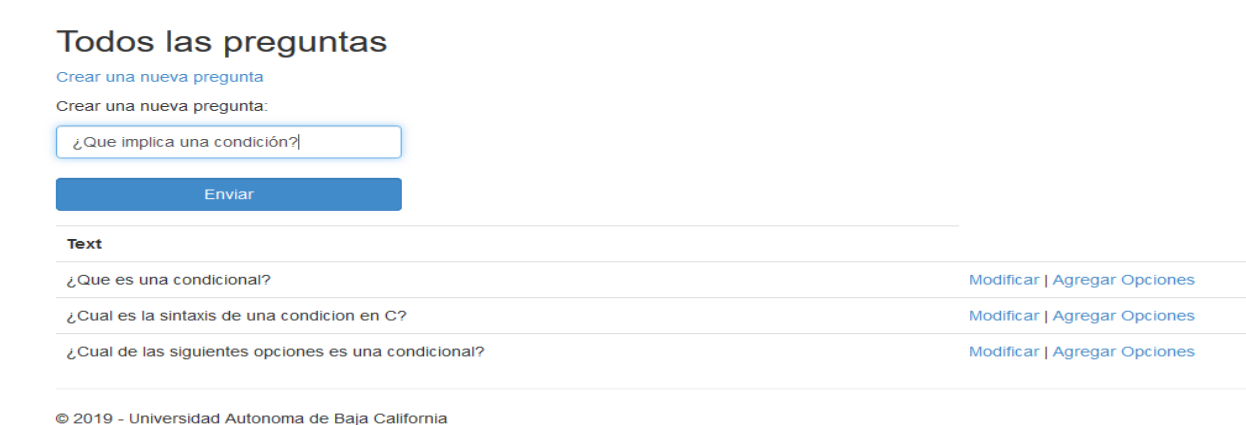


Figura 3.13. Interfaz para crear preguntas.

En la Figura 3.14 se muestra la interfaz que permite agregar opciones a la pregunta. Esta pantalla muestra la pregunta a la que será posible incluir distintas respuestas, utilizando una caja de texto y una marca de selección. También se especifica cual será la respuesta correcta.

Agregar opciones

Usa la caja de texto para añadir una opcion a la pregunta
selecciona el checkbox si la opción es la correcta

Texto de la pregunta: ¿Que es una condicional?

☐	Ninguna de las anteriores
--------------------------	---------------------------

Agregar Opciones

Figura 3.14. Interfaz donde se pueden incluir diferentes respuestas.

Una vez que la pregunta tenga sus posibles respuestas, esta pantalla mostrará un submenú, el cual nos permitirá eliminar opciones (Ver Figura 3.15), de modo que el administrador pueda borrar las opciones mediante un checkbox.

Agregar opciones

Usa la caja de texto para añadir una opcion a la pregunta
selecciona el checkbox si la opción es la correcta

Texto de la pregunta: ¿Que es una condicional?

☐ Ninguna de las anteriores

Agregar Opciones

Deselecciona si deseas eliminar alguna opción

Opcion 1: ☒ Una decision tomada con base en un valor

Opcion 2: ☒ Una respuesta

Opcion 3: ☒ No se

Eliminar Opciones

© 2019 - Universidad Autonoma de Baja California

Figura 3.15. Interfaz donde se pueden eliminar respuestas o elegir cual es la correcta.

En la Figura 3.16 se muestra la interfaz de aplicación móvil que utiliza el usuario final (estudiante). Como se mencionó previamente, se realizó mediante *Ionic Framework*. Esta aplicación le permitirá al estudiante consultar la información de cada módulo desde cualquier dispositivo móvil. Debido a las ventajas de este *Framework*, se permite usarse tanto en Android como en iOS de manera nativa, así como desde un navegador Web. Esta aplicación educativa muestra el contenido generado por la aplicación “maestra” y permite al alumno acceder a toda la información que el administrador ingresó a la aplicación.

En esta figura se muestra el módulo creado anteriormente, y es aquí donde se enlistarán todos los módulos creados por el administrador y podemos acceder a ellos por medio de un toque.

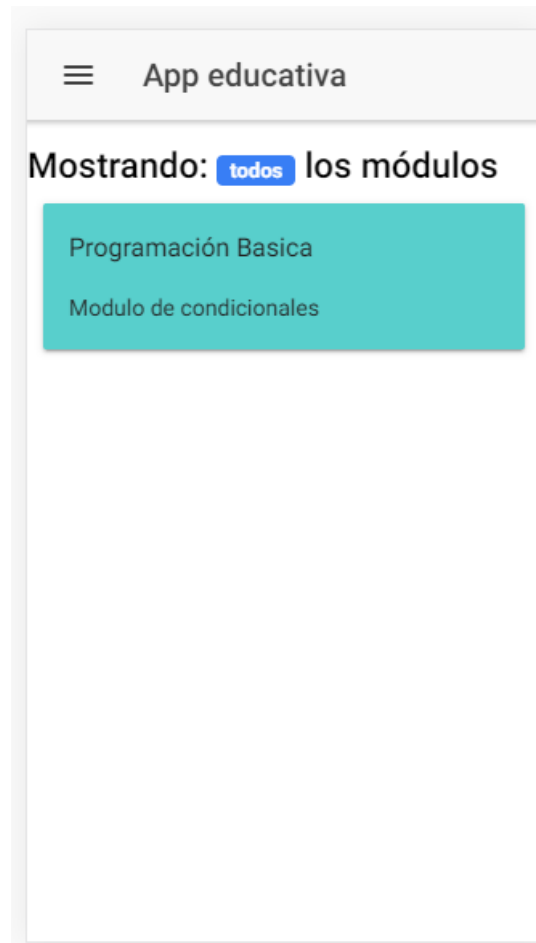


Figura 3.16. Interfaz de usuario de la aplicación móvil en la cual se presentan los módulos o cursos.

Cuando se accede a alguno de los módulos de la pantalla anterior se muestra el contenido educativo de un módulo creado por el administrador introdujo al módulo (Figura 3.17).

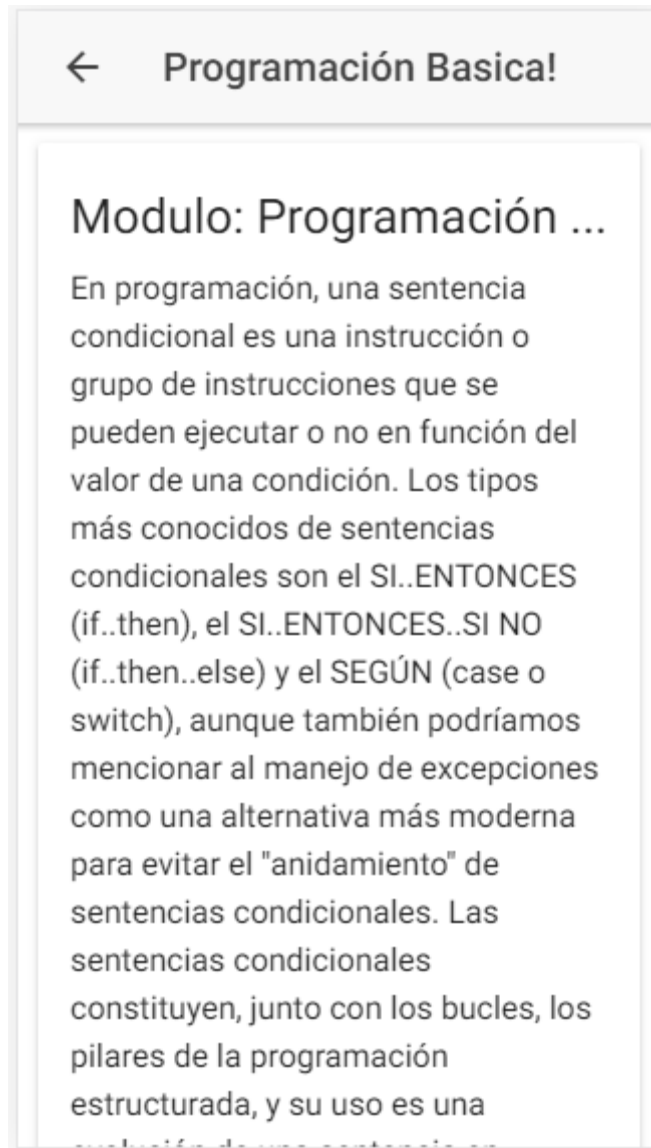


Figura 3.16. Interfaz móvil con un ejemplo del contenido educativo de un módulo creado.

En la Figura 3.18 se presenta la interfaz donde se presenta un video reproducido desde la aplicación móvil.

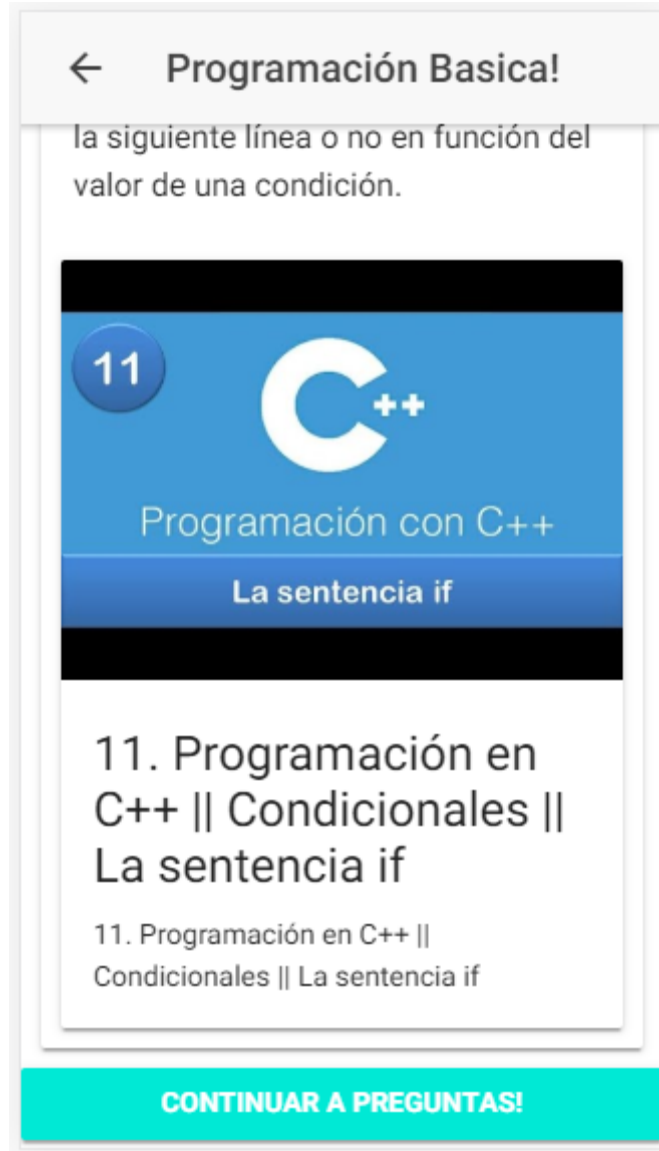


Figura 3.18. Interfaz de la aplicación móvil reproduciendo un video contenido en un módulo.

En la Figura 3.19 se muestra un ejemplo de un cuestionario de evaluación del aprendizaje, la cual deben ser contestadas por los estudiantes. Como se puede observar se presenta una pregunta y sus posibles respuestas, de manera que el estudiante deberá contestarlas para obtener una calificación.

← Questions!

¿Que es una condicional?

Respuestas

Una decision tomada en base ... ☐

Una respuesta ☐

No se ☐

Desliza a la derecha para continuar

Figura 3.19. Interfaz móvil con un ejemplo de un cuestionario de evaluación.

En la Figura 3.20 se muestra la interfaz donde se informa la calificación final del estudiante, la cual consta de una escala de 1 a 100, calculando el promedio de las preguntas que se presentan.

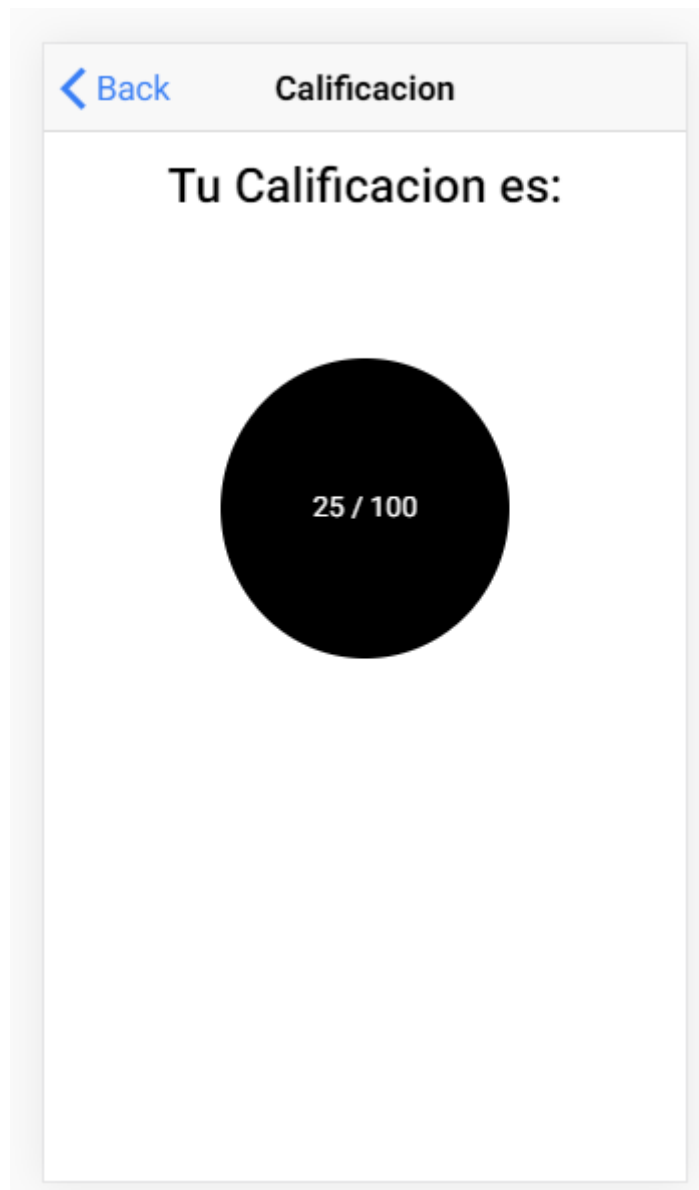


Figura 3.20. Interfaz donde se muestra el resultado de un cuestionario de evaluación.

Capítulo 4. Conclusiones

A lo largo de este proyecto se llevó a cabo un análisis para entender de qué forma se puede motivar a una persona al aprender un tema. Se entendió que las personas, a lo largo de su día a día, pasan mucho tiempo en sus dispositivos móviles, y por lo tanto se decidió desarrollar este proyecto. Sin embargo, existen diferentes aplicaciones en el mercado para este tema (*m-Learning*), pero no consideran los aspectos pedagógicos ni tecnológicos con el fin de aumentar el éxito de estas aplicaciones. Por lo tanto, entender al usuario y sus necesidades fue uno de los principales objetivos de investigación, para que las aplicaciones fuesen accesibles y aceptadas por muchas personas. Con base a este estudio se logró una herramienta que permite desarrollar aplicaciones educativas móviles para distintas plataformas, las cuales consideran lineamientos de usabilidad y pedagógicos. Por ejemplo, se recomiendan incluir módulos cortos para que las personas las puedan acceder en periodos cortos de tiempo, pero que constantemente puedan acceder al contenido educativo de su interés, y de esta manera puedan aprender el contenido considerando sus diferentes estilos de aprendizaje (Kinestésico, visual o auditivo).

Debido a que existe la necesidad de crear nuevo contenido o actualizarlo, se optó por desarrollar un sistema Web con una base de datos centralizada y una aplicación maestra que puede ser añadida mediante uno o varios administradores para que posteriormente

se vea reflejada inmediatamente en la aplicación móvil, independientemente de su sistema operativo. Esto nos permite corregir los módulos creados de los módulos de una manera sencilla.

Código de los Proyectos: <https://github.com/marcelcelaya/ProyectoTesis>

Referencias Bibliográficas

- Ericsson. (2016). Ericsson Mobility Report: On the pulse of the networked society. In. Stockholm, Sweden.
- Marqués, P. (1995). Metodología para la elaboración de software educativo. *Barcelona (España). Editor. Estel.*
- Gebera, O. W. T. (2012). Criterios de valoración sobre la usabilidad pedagógica en la formación continua docente. *Razón y palabra*(80), 48-22.
- Duarte Filho, F., & Barbosa, E. F. (2013). A contribution to the quality evaluation of mobile learning environments. In *Frontiers in Education Conference, 2013 IEEE* (pp. 379-382): IEEE.
- Fetaji, B., Ebibi, M., & Fetaji, M. (2011). Assessing Effectiveness in Mobile Learning by Devising MLUAT (Mobile Learning Usability Attribute Testing) Methodology. *International Journal of Computers and Communications*, 5(3), 178-187.
- Johnson, L., Adams Becker, S., Cummins, M., Estrada, V., Freeman, A., & Ludgate, H. (2013). Horizon report: 2013 higher education edition.
- Johnson, L., Becker, S., Estrada, V., & Freeman, A. (2014). Horizon report: 2014 Higher Education Edition Austin, Texas, Estados Unidos: The New Media Consortium.
- Moreno-Ger, P., Martinez-Ortiz, I., Freire, M., Manero, B., & Fernandez-Manjon, B. (2014). Serious games: A journey from research to application. In *Frontiers in Education Conference (FIE), 2014 IEEE* (pp. 1-4): IEEE.
- UNESCO. (2013). The Future of mobile Learning: Implications for Policy Markers and Planners. In: United Nations Educational, Scientific and Cultural Organization.
- Cisco. (2012). Costumer case study. University Embraces Bring-Your-Own-Device with Wireless Network.
- O'Malley, C., Vavoula, G., Glew, J., Taylor, J., Sharples, M., Lefrere, P., Lonsdale, P., Naismith, L., & Waycott, J. (2005). Guidelines for learning/teaching/tutoring in a mobile environment.
- Wong, L.-H., & Looi, C.-K. (2011). What seams do we remove in mobile-assisted seamless learning? A critical review of the literature. *Computers & Education*, 57(4), 2364-2381.

- Zhou, X., Huang, S., Chen, Y.-S., & Dai, W. (2015). Emotional intelligence system for smart education in ubiquitous learning. In *Information Technology and Applications: Proceedings of the 2014 International Conference on Information technology and Applications (ITA 2014), Xian, China, 8-9 August 2014* (pp. 167): CRC Press.
- Crompton, H. (2013). A historical overview of mobile learning: Toward learner-centered education. *Handbook of mobile learning*, 3-14.
- Kukulska-Hulme, A. (2007). Mobile usability in educational contexts: what have we learnt? *The International Review of Research in Open and Distributed Learning*, 8(2).
- Nielsen, J. (1993). *Usability Engineering*. San Francisco, CA. USA.: Morgan Kaufmann Publishers Inc.
- Nielsen, J. (2012). Usability 101: Introduction to Usability (2012). URL: <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>[accessed 2015-06-03][WebCite Cache].
- Nielsen, J., & Budiu, R. (2013). *Usabilidad en dispositivos móviles*: Anaya Multimedia.
- Cuadrat Seix, C., Veloso, M. S., & Soler, J. J. R. (2012). Towards the validation of a method for quantitative mobile usability testing based on desktop eyetracking. In *Proceedings of the 13th International Conference on Interacción Persona-Ordenador* (pp. 49): ACM.
- Debrauwer, L. (2013). *Patrones de diseño en Java: Los 23 modelos de diseño: descripción y solución ilustradas en UML 2 y Java*. Ediciones ENI.
- West, M., & Ei, C. H. (2014). *Reading in the mobile era: a study of mobile reading in developing countries*: UNESCO.
- Economides, A. A. (2008). Requirements of mobile learning applications. *International Journal of Innovation and Learning*, 5(5), 457-479.
- Elias, T. (2011). Universal instructional design principles for mobile learning. *The International Review of Research in Open and Distributed Learning*, 12(2), 143-156.
- Parsons, D., Ryu, H., & Cranshaw, M. (2007). A design requirements framework for mobile learning environments. *Journal of Computers*, 2(4), 1-8.
- Durall Gazulla, E., Gros Salvat, B., Maina, M. F., Johnson, L., & Adams, S. (2012). Perspectivas tecnológicas: educación superior en Iberoamérica 2012-2017.

- Valero, C. C., Redondo, M. R., & Palacín, A. S. (2012). Tendencias actuales en el uso de dispositivos móviles en educación. *La Educación Digital Magazine*, 147, 1-21.
- Sakamura, K., & Koshizuka, N. (2005). Ubiquitous computing technologies for ubiquitous learning. In *Wireless and Mobile Technologies in Education, 2005. WMTE 2005. IEEE International Workshop on* (pp. 11-20): IEEE.
- Yen, J.-C., & Lee, C.-Y. (2011). Exploring problem solving patterns and their impact on learning achievement in a blended learning environment. *Computers & Education*, 56(1), 138-145.
- Quinn, C. (2000). mLearning: Mobile, wireless, in-your-pocket learning. *LiNE Zine*, 2006.
- Sharples, M., Taylor, J., & Vavoula, G. (2005). Towards a theory of mobile learning. In *Proceedings of mLearn* (Vol. 1, pp. 1-9).
- ISO, I. (2011). IEC25010: 2011 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—System and software quality models. *International Organization for Standardization*, 34.
- Hu, Z. (2011). Emerging vocabulary learning: From a perspective of activities facilitated by mobile devices. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (Vol. 2011, pp. 1334-1340).

Páginas Web

(Wikipedia, 2013). https://es.wikipedia.org/wiki/Programación_por_capas

(IBM, 2014).

https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.ertools.mft.doc/ac55710_.htm

(BBVA, 2016). <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>

(pmoinformatica, 2018). http://www.pmoinformatica.com/2018/04/documento-de-requerimientos-de-software_37.html

(IBM, Knowledge Center, 2018).

https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm

(Granata N., 2017). <https://nicolasgranata.com/2017/05/09/implementacion-patron-repositorio-c-entity-framework-net-core-parte-1/>

(Wikipedia, 2019). https://es.wikipedia.org/wiki/Diseño_por_contrato