



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Estudios de Postgrado

Maestría de Tecnologías de la Información y Comunicación

DISEÑO DEL SISTEMA PARA LA CONSULTA DE CONTENIDO KIDSPEDIA

Ervin Encarnación López Lux

Asesorado por la Msc. Inga. María Elizabeth Aldana Díaz

Guatemala, agosto de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DEL SISTEMA PARA LA CONSULTA DE CONTENIDO
KIDSPEDIA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA Y LA ESCUELA DE ESTUDIOS DE
POSTGRADO
POR

ERVIN ENCARNACIÓN LÓPEZ LUX

ASESORADO POR LA MSC. INGA. MARÍA ELIZABETH ALDANA DÍAZ

AL CONFERÍRSELE EL TÍTULO DE

MAESTRO EN TECNOLOGÍA DE LA INFORMACIÓN Y LA COMUNICACIÓN

GUATEMALA, AGOSTO DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADOR	M. Sc. Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Luis Fernando Espino Barrios
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DEL SISTEMA PARA LA CONSULTA DE CONTENIDO KIDSPEDIA

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Postgrado, con fecha junio de 2017.

Ervin Encarnación Lopez Lux



FACULTAD DE
INGENIERÍA - USAC
ESCUELA DE
ESTUDIOS DE POSTGRADO

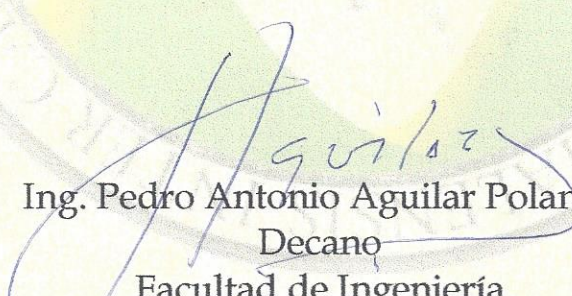
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Teléfono 2418-9142 / 24188000 Ext. 86226

APT-2017-015

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Postgrado, al Trabajo de Graduación de la Maestría en Artes en Tecnologías de la Información y la Comunicación titulado: **"DISEÑO DEL SISTEMA PARA LA CONSULTA DE CONTENIDO KISPEDIA"** presentado por el ingeniero en Ciencias y Sistemas Ervin Encarnación López Lux, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

"Id y Enseñad a Todos"


Ing. Pedro Antonio Aguilar Polanco
Decano

Facultad de Ingeniería
Universidad de San Carlos de Guatemala



Guatemala, agosto de 2017.

Cc: archivo/la

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.



FACULTAD DE
INGENIERÍA - USAC
EP
ESCUELA DE
ESTUDIOS DE POSTGRADO

Escuela de Estudios de Postgrado
Facultad de Ingeniería
Teléfono 2418-9142 / 24188000 Ext. 86226

APT-2017-015

El Director de la Escuela de Estudios de Postgrado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen y dar el visto bueno del revisor y la aprobación del área de Lingüística del Trabajo de Graduación titulado **"DISEÑO DEL SISTEMA PARA LA CONSULTA DE CONTENIDO KISPEDIA"** presentado por el Ingeniero en Ciencias y Sistemas Ervin Encarnación López Lux, correspondiente al programa de Maestría en Artes en Tecnología de la Información y la Comunicación; apruebo y autorizo el mismo.

Atentamente,

"Id y Enseñad a Todos"

MSc. Ing. Murphy Olympto Paiz Recinos
Director

Escuela de Estudios de Postgrado
Facultad de Ingeniería
Universidad de San Carlos de Guatemala



Guatemala, agosto de 2017.

Cc: archivo/la

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.



FACULTAD DE
INGENIERÍA - USAC
EP
ESCUELA DE
ESTUDIOS DE POSTGRADO

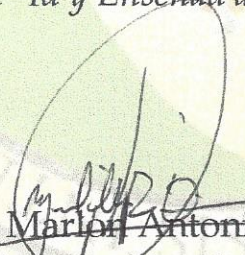
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Teléfono 2418-9142 / 24188000 Ext. 86226

APT-2017-015

Como Coordinador de la Maestría en Artes en Tecnologías de la Información y la Comunicación del Trabajo de Graduación titulado **"DISEÑO DEL SISTEMA PARA LA CONSULTA DE CONTENIDO KISPEDIA"** presentado por el Ingeniero en Ciencias y Sistemas Ervin Encarnación López Lux, apruebo y recomiendo la autorización del mismo.

Atentamente,

"Id y Enseñad a Todos"


MSc. Marlon Antonio Pérez Turrubiates
Coordinador de Maestría
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Universidad de San Carlos de Guatemala



Guatemala, agosto de 2017.

Cc: archivo/la

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.

ACTO QUE DEDICO A:

Dios

Por ser el creador del universo, de la vida, de mi vida y por haber permitido aun por más tiempo la vida de mi madre y mis seres queridos, estas personas son la razón de toda mi lucha y la razón de mí existir. Que se haga su voluntad y no la mía.

Mi madre

En especial a mi madre. Por ser la mujer que más quiero, por su amor incondicional a sus hijos, por su humildad, sencillez, por ser una mujer luchadora y permanecer aun a mi lado. A principio del 2015, casi la perdía, pero demostró ser una vez más que es una mujer luchadora y junto con Dios, salió con vida del intensivo.

Padre

Mi padre, nos dejaste varias veces pero al final estuviste con nosotros, tus hijos y mi madre. Por ser la persona que me inculcó desde pequeño los valores del trabajo.

Hermanos

Mis hermanos María Magdalena (q.e.p.d) y Mario Salvador (q.e.p.d), por ser importantes influencias en mi carrera ahora, un ángel que siempre me acompañará hasta el final de mis días.

Hermanos

A mis hermanos Sebastiana Estela y Carlos Enrique, por su apoyo y consejos que me han dado. Siempre es muy notorio su apoyo.

AGRADECIMIENTOS A:

**Universidad de San
Carlos de Guatemala**

Alma máter que me dio la oportunidad de ser útil a la sociedad.

Facultad de Ingeniería

Por brindarme los conocimientos, habilidades, valores y experiencias necesarias para desarrollar mis capacidades profesionales.

Escuela Marista

Por brindarme conocimiento y formación cristiana para la vida

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
GLOSARIO	XI
RESUMEN.....	XIII
PLANTEAMIENTO DEL PROBLEMA.....	XV
OBJETIVOS.....	XIX
INTRODUCCIÓN	XXI
1. ANTECEDENTES	1
1.1. Análisis y discusión de arquitecturas utilizadas por sistemas <i>e-learning</i>	1
1.1.1. Basada en servicios <i>web</i> y agentes inteligentes.....	2
1.1.2. Componentes y funcionalidades.....	3
1.1.3. Árbol de conocimiento: Una arquitectura distribuida para el aprendizaje en línea (<i>e- learning</i>) adaptativo	6
1.1.4. Componentes y funcionalidades.....	7
1.1.5. Una arquitectura de servicios <i>web</i> para contenido enriquecido desde clientes de aprendizaje móvil.....	9
1.1.6. Arquitectura basada en la <i>web</i> para el aprendizaje móvil.....	14
1.1.7. Entrega de material <i>e-learning</i> consciente del contexto de usuario: enfoque y arquitectura.....	19
1.1.8. Arquitectura del sistema	20
1.1.9. Arquitectura componente y su función.....	21

1.1.10.	Servicio de Ontología	22
1.1.11.	Servicio Matching	22
1.1.12.	Creación del programa de aprendizaje	23
1.1.13.	Servicio de comunicación.....	23
1.1.14.	Repositorio de ontología.....	23
1.1.15.	Administrador objetos <i>learning</i>	24
1.1.16.	Administrador de Contenido del usuario	24
2.	JUSTIFICACIÓN.....	25
3.	ALCANCES.....	27
3.1.	Alcances Investigativos.....	27
3.2.	Diseño de la arquitectura de <i>software</i>	27
3.3.	Diseño de la interfaz de usuario	28
3.4.	Alcances técnicos.....	29
3.5.	Resultados esperados.....	30
4.	MARCO TEÓRICO	33
4.1.	Plataformas aplicadas al soporte de contenido educativo.....	33
4.2.	Sistema de gestión de contenido	33
4.3.	Sistema de gestión de aprendizaje	34
4.4.	Entornos virtuales de aprendizaje	34
4.5.	Arquitectura de <i>software</i>	35
4.6.	La importancia de la arquitectura de <i>software</i>	36
4.7.	Diseño arquitectónico.....	36
4.8.	Estilos arquitectónicos.....	37
4.9.	Arquitecturas centradas de datos.	38
4.10.	Base de datos	39
4.11.	Definición de base de datos	39

4.12.	Sistema de Gestión de Base de Datos (SGBD)	40
4.13.	Modelo de datos	41
4.14.	Lenguaje de base de datos	43
4.15.	Caso de uso	43
4.16.	Diagrama de caso de uso.....	44
4.16.1.	Diseño de interfaz de usuario	45
4.16.2.	Diseño centrado en el usuario	46
4.16.3.	Proceso de diseño de interfaz de usuario.....	46
4.16.4.	Principios generales de diseño de interfaz de usuario.....	47
4.16.5.	Las mejores prácticas de usabilidad de Jakob Nielsen.....	49
4.16.6.	Los dispositivos móviles	51
4.16.7.	Definición de APP	51
4.16.8.	Aplicaciones móviles web	51
4.16.9.	Aplicaciones móviles nativas	52
4.16.10.	Aplicaciones móviles híbridas.....	52
4.16.11.	Sistema operativo android.....	52
5.	METODOLÓGICO.....	55
5.1.	Descripción del problema y elaboración del documento de requerimiento.....	55
5.2.	Elaboración del diseño de arquitectura de software	56
5.3.	Diseño de caso de uso	56
5.4.	Diseño de esquema de datos	57
5.5.	Diseño de una interfaz amigable	58
5.6.	Revisión y finalización del documento de arquitectura	62
6.	PRESENTACIÓN DE RESULTADOS.....	63

6.1.	Selección, descripción y diseño de la arquitectura del sistema para la consulta de contenido KIDSPEDIA	63
6.2.	Funcionalidades y componentes	64
6.2.1.	Servidor de base de datos.....	64
6.2.2.	Servidor de servicios	64
6.2.3.	Servidor de contenido dinámico	65
6.2.4.	Servidor de comprobación.....	66
6.2.5.	Funcionalidad.....	66
6.2.6.	Diagrama de actividad, consulta de contenido	66
6.2.7.	Arquitectura para dispositivos móviles	68
6.2.8.	Interfaz gráfica de usuario	69
6.2.9.	Lógica de negocios	70
6.2.10.	Base de datos	70
6.2.11.	Diseño de casos de uso del sistema para la consulta de contenido KIDSPEDIA	73
6.2.12.	Descripción de caso de uso	74
6.2.13.	Diseño de esquema de Base de Datos del Sistema para la consulta de contenido KIDSPEDIA	77
6.2.14.	Descripción y diseño de interfaz de usuario del sistema para la consulta de contenido KIDSPEDIA	82
6.2.15.	Propósito	82
6.2.16.	Ámbito de sistema.....	83
6.2.17.	Definición de la tecnología a utilizar	83
6.2.18.	Definición de los usuarios del sistema	84
6.3.	Diseño de prototipos de interfaz de usuario del sistema para la consulta de contenido KIDSPEDIA	85
6.3.1.	Pantalla de inicio	85

6.3.2.	Pantalla login	85
6.3.3.	Pantalla lista de contenido	86
6.3.4.	Pantalla para contenido de matemáticas	87
6.3.5.	Pantalla lista de resultados	88
6.3.6.	Diseño de interfaz de la consulta de contenido KIDSPEDIA.....	89
6.4.	Diseño de pantallas de interfaz del sistema	90
6.4.1.	Pantallas de diseño	90
6.4.2.	El icono	91
6.4.3.	Iconos Internos	95
6.4.4.	Diagrama de navegación	96
6.4.5.	Cuadro de dialogo y mensajes	97
6.4.6.	Diseño de menú.....	98
6.4.7.	Pantalla de bienvenida	98
6.4.8.	Pantalla Login	99
6.4.9.	Pantalla lista de contenido	100
6.4.10.	Diseño pantalla contenido de curso de matemáticas	101
6.4.11.	Diseño pantalla ejercicio de matemáticas.....	102
6.4.12.	Diseño de pantalla del curso de lectura.....	103
6.4.13.	Diseño pantalla progreso del curso de matemáticas.	104
6.4.14.	Diseño pantalla de error	105
6.4.15.	Diseño pantalla manual de la aplicación.....	106
6.4.16.	Diseño pantalla configuración de cuenta	107
6.4.17.	Diseño pantalla configuración de contraseña	108
6.5.	Validación de Interfaz	109
6.5.1.	Rubricas a utilizar	110
6.5.2.	Preguntas evaluación de diseño.....	110

6.5.3.	Preguntas evaluación de usabilidad.....	110
6.5.4.	Preguntas evaluación de accesibilidad	111
6.5.5.	Resultados de la validación.....	111
6.5.6.	Evaluación de diseño	112
7.	DISCUSIÓN DE RESULTADOS.....	119
7.1.	Discusión de resultados de validez de interfaz	119
7.2.	Necesidades del proceso de enseñanza-aprendizaje gestión y consulta de contenido	120
7.3.	Proceso de enseñanza-aprendizaje: gestión y consulta	121
7.4.	Proceso de enseñanza-aprendizaje sistema para la consulta de contenido KIDSPEDIA	123
7.4.1.	Interfaz amigable del software educativo	126
7.4.2.	Casos de usos relevantes del sistema	131
7.4.3.	Esquema de base de datos de la solución	133
7.4.4.	Impactos sociales.....	134
7.4.5.	Impacto tecnológicos.....	135
	CONCLUSIONES.....	137
	RECOMENDACIONES	141
	REFERENCIAS BIBLIOGRÁFICAS.....	143

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Arquitectura de sistema <i>e-learning</i> en servicios web y agentes inteligentes	3
2.	Principales componentes de la arquitectura Knowledge Tree distribuida.....	7
3.	El cliente proceso de interacción entre el servidor utilizando XML, a través de HTTP	10
4.	Implementación del <i>Patrón Front Controller</i> que procesa las solicitudes XML y las respuestas	12
5.	Arquitectura del sistema.....	15
6.	Arquitectura de alto nivel del sistema.....	22
7.	Arquitectura del sistema para la consulta de contenido Kidspedia	63
8.	Diagrama de actividad.....	67
9.	Arquitectura para dispositivos móviles	69
10.	Diagrama de bloque del sistema.....	71
11.	Diagrama de contexto del sistema KIDSPEDIA	72
12.	Diagrama de caso de uso	73
13.	Diagrama de caso de uso	74
14.	Diagrama entidad relación	78
15.	Tablas usuario y alumno	79
16.	Relaciones entre usuario, alumno y autenticación	79
17.	Representación gráfica de la tabla asignatura y contenido de ejemplo.....	80

18.	Representación gráfica de la tabla categoría y contenido de ejemplo	80
19.	Representación gráfica de las tablas categoria, categorialtem, subcategoria y sus relaciones con contenido de ejemplo	81
20.	Ejercicio, comprobación y sus relaciones	82
21.	Prototipo de la pantalla de inicio	85
22.	Prototipo de la pantalla login.....	86
23.	Prototipo de la pantalla lista de contenido	87
24.	Prototipo de la pantalla contenido de matemáticas.....	88
25.	Prototipo de la pantalla lista de resultados	89
26.	Primer diseño sobre marco de trabajo	90
27.	Resultado de búsqueda y aplicaciones instaladas, Google Play	91
28.	Las aplicaciones compiten con muchas otras.....	92
29.	Icono de la aplicación Kidspedia.....	93
30.	Elaboración de icono con fondo café degradado	93
31.	Icono kidspedia con fondo café degradado	94
32.	Elaboración de icono diseño 2.....	94
33.	Icono Kidspedia diseño 2.....	95
34.	Iconos internos.....	95
35.	Diagrama de navegación de la aplicación	96
36.	Diseño de cuadro diálogo y de mensajes	97
37.	Diseño de menú.....	98
38.	Diseño de la pantalla de bienvenida	99
39.	Diseño de la pantalla login.....	100
40.	Pantalla lista de contenido	101
41.	Pantalla contenido de curso de matemáticas	102
42.	Diseño pantalla ejercicio de matemáticas.....	103
43.	Diseño de pantalla del curso de lectura	104
44.	Diseño pantalla progreso del curso de matemáticas	105

45.	Diseño de pantalla de error	106
46.	Diseño pantalla manual de la aplicación	107
47.	Diseño pantalla configuración de cuenta.....	108
48.	Diseño pantalla configuración de contraseña.....	109
49.	Pregunta 1.....	111
50.	Gráfica PIE, pregunta 1.....	112
51.	Pregunta 2.....	112
52.	Gráfica PIE, pregunta 2.....	113
53.	Pregunta 3.....	113
54.	Gráfica PIE, pregunta 3.....	114
55.	Pregunta 4.....	114
56.	Gráfica PIE, pregunta 4.....	115
57.	Pregunta 5.....	115
58.	Gráfica PIE, pregunta 5.....	116
59.	Pregunta 6.....	116
60.	Gráfica PIE, pregunta 6.....	117
61.	Pregunta 7.....	117
62.	Gráfica PIE, pregunta 7.....	118

TABLAS

I.	Las 10 heurísticas de Jakob Nielsen	50
II.	Caso de uso autenticarse al sistema.....	74
III.	Caso de uso consultar programa	75
IV.	Caso de uso consultar contenido	75
V.	Caso de uso responder preguntas	76
VI.	Caso de uso consultar evaluación	76
VII.	Promedio y resultados obtenidos	119
VIII.	Promedio y resultados obtenidos	120

GLOSARIO

Hardware	Es la parte tangible de una computadora.
Logaritmo	Conjunto de sentencias, las cuales expresan la lógica de un programa.
MINEDUC	Ministerio de Educación de Guatemala.
Programa	Colección de instrucciones que le indican a la computadora qué hacer.
Learning Object (LO)	Son la unidad mínima de contenido de aprendizaje pedagógicamente.
Learning Object metadata(LOM)	Metadatos para objetos de aprendizaje, es una estructura de datos estándar utilizado para describir objetos de aprendizaje.

RESUMEN

El objetivo principal de este trabajo de graduación es presentar un diseño de una herramienta de software educativo, para el apoyo a la educación básica guatemalteca con las características intuitivo, fácil de utilizar, atractivo, adaptable y con información confiable y actualizada. Contenido de educación básica en Guatemala, se refiere al contenido de educación primaria para niños de siete a doce años que el Ministerio de Educación establece, con el fin de apoyar al cumplimiento curricular el contenido para alumnos y alumnas será sobre lectura, escritura, matemática, expresión y comprensión oral, el sentido artístico, creativo y cultural, conocimiento del medio natural y social.

El sistema para la consulta de contenido KIDSPEDIA es un diseño de una solución para dispositivos móviles orientados a niños, con el objetivo de ayudar a que los niños mejoren su aprendizaje en las diversas áreas, a través de los aspectos tecnológicos propuestos que son: un diseño e interfaz de usuario para dispositivos móviles amigable, eficiente y una arquitectura de software con los siguientes componentes:

Servidor de contenido dinámico contiene toda la lógica, los procesos requeridos para la creación de contenido dinámico y programas de contenido dinámicos. El servidor de comprobación que proporciona al usuario evaluaciones para medir la comprensión obtenida de cada material que se le ha dado por medio del servidor de contenido dinámico. Servidor de servicios hace uso de protocolos y estándares como XML. El servidor de base de datos el cual utiliza un potente Sistema de Gestión de Base de Datos SGBD, por medio del cual permite realizar consultas complejas sobre los datos.

- La solución se construyó en cinco etapas:

Fase de selección y diseño de la arquitectura, en esta fase se realizó un estudio sobre arquitecturas. El entregable en esta etapa o fase fue el diseño de la arquitectura de software del sistema.

Fase del análisis de casos de uso en el cual se analizaron los requerimientos de usuario, las acciones, se modelaron y se crearon escenarios. El entregable en esta fase fue el diseño de diagrama de casos de uso del sistema.

Fase de análisis de datos y diseño de base de datos. En esta fase se analizaron los datos, la estructura de los datos, el contenido a presentar a los usuarios, todas las fases son importantes y se relacionan una con otras, cuando se tiene bien claro el contenido, la forma de la información se realizó el diseño de la base de datos.

Fase de diseño de una interfaz amigable, se inició con la identificación de usuarios, tareas y requisitos del entorno que corresponde con el análisis de requerimiento. Identificada las tareas que pueden realizar los usuarios en la fase de diseño de casos de uso, se analizó cada uno de los escenarios y se definieron un conjunto de objetos y acciones de interfaz. A partir del análisis anterior se extrajo información que se utilizó para la elaboración del diseño preliminar de interfaz denominados prototipos de interfaz y finalmente se diseñaron los elementos de interfaz en donde se colocan iconos, objetos gráficos, texto, títulos de ventanas y la especificación de los elementos de los menús.

PLANTEAMIENTO DEL PROBLEMA

La educación es primordial para el desarrollo individual, familiar, social y de una nación. Por lo tanto, es indispensable contar con un eficaz y eficiente sistema de apoyo acorde con la programación dada en la formación básica del niño. Hoy en día se cuenta con el desarrollo de nuevas tecnologías que permiten elaborar el software educativo que pueden ser empleados como nueva estrategia que apoyen al aprendizaje y se espera que exista una mejora en el aprendizaje.

Actualmente, existe una variedad de software en finanzas, entretenimiento y software educativo. Este tipo de software, en su mayoría se enfoca a la población profesional. Otro problema es el enfoque inadecuado para los niños. Principalmente, por la necesidad de disponer de una herramienta en donde se pueda consultar el conocimiento necesario para la enseñanza-aprendizaje de los niños, ya que es importante para la educación. Esta herramienta hace uso de la tecnología, lo que facilita el trabajo de los maestros y mejora el aprendizaje de los niños.

El problema específico para la persona que presente esta propuesta es que no existe un software educativo apto para las condiciones de la población estudiantil de Guatemala y que sirva de apoyo a la educación de los niños.

- No poseen control o supervisión de calidad de los contenidos.
- Son creados para ciertas zonas geográficas, cultural o niveles educativos y no todos los usuarios son homogéneos. La mayor parte de este

problema se da porque muchos sistemas de software son de otros países y muy pocos desarrollados en Guatemala.

- Las interfaces de las aplicaciones educativas que existen, no poseen las siguientes características, intuitivo, fácil de utilizar aptas para los niños esto es uno de los grandes problemas con la mayoría de interfaces de usuario.

- El proyecto responde a la pregunta central:
 - ¿Cuál es el diseño de software educativo para el apoyo a la educación básica de Guatemala del nivel primaria para niños y niñas de siete a doce años que establece el Ministerio de Educación y que cumpla con las siguientes características, intuitivo, fácil de utilizar, atractivo, con la información confiable y actualizada?

- Además, se definen las siguientes preguntas auxiliares:
 - ¿Cuál es el diseño de la arquitectura de un sistema de software, que permita proporcionarle al software las capacidades necesarias para permitir su mantenimiento y evolución de acuerdo a las necesidades del negocio y las soluciones de los usuarios?

 - ¿Cuál es el diseño de interfaz amigable para el software educativo, para que su utilización sea fácil y aceptada por la mayoría de los usuarios y lograr el éxito de su aceptación por los usuarios?

- ¿Cuál es el diseño de caso de uso que muestre por completo las interacciones entre los usuarios y el sistema para iniciar y facilitar su desarrollo?

- ¿Cuál es el esquema de base de datos, para el manejo eficiente del contenido del sistema de software y proporcionar información correcta y resultados inmediatos al usuario?

OBJETIVOS

General

Diseñar una herramienta de software educativo para el apoyo a la educación básica guatemalteca del nivel primaria para niños y niñas de siete a doce años que el Ministerio de Educación establece, con las siguientes características: intuitivo, fácil de utilizar, atractivo, adaptable y con información confiable y actualizada.

Específicos

1. Diseñar una arquitectura de software que permita proporcionarle al software las capacidades necesarias para su mantenimiento y evolución, de acuerdo a las necesidades del negocio y las soluciones requeridas de los usuarios.
2. Describir y seleccionar la arquitectura de software a utilizar para el sistema, para la consulta KIDSPEDIA.
3. Describir y diseñar la interfaz amigable del software educativo para que su utilización sea fácil y aceptado por la mayoría de los usuarios.
4. Diseñar casos de uso, que muestren las principales interacciones entre los usuarios y el sistema para iniciar su desarrollo.

5. Diseñar el esquema de base de datos que se utilizará en el software educativo para que sea más eficiente en el manejo del contenido de software.

INTRODUCCIÓN

Mucho se habla de los cambios que trajo consigo la evolución de la informática, comunicación y dispositivos móviles. Pocos pensaron cómo iba a afectar a los niños estos cambios. Hoy en día existen muchas formas de conectarse a internet por medio de diversos dispositivos, cada vez hay más dispositivos que se conectan a internet a tal punto que a ese fenómeno se le denomina internet de las cosas.

La tecnología en la sociedad se extiende hacia la educación y la utilización de la misma en el ámbito educativo es indispensable para contribuir en el apoyo al mejoramiento de la calidad, para fomentar el desarrollo integral de los alumnos y maestros. Actualmente en el mercado mundial existen herramientas educativas similares a la propuesta, pero en su mayoría enfocadas a los adultos y hay pocas opciones para niños como la que propone el diseño de la investigación para la consulta KISPEDIA.

La solución propuesta en el presente trabajo se llama: Diseño de la Investigación de un sistema para la consulta de contenido KIDSPEDIA y tiene como objetivo principal el diseño de una herramienta de software educativo, para el apoyo a la educación básica guatemalteca con las características intuitivo, fácil de utilizar, atractivo, adaptable y con información confiable y actualizada.

El desarrollo del diseño de un sistema de software educativo lleva procesos similares a los de cualquier otro software y deben considerarse todos los aspectos, principalmente la interfaz gráfica que es la carta de presentación

del programa y en ocasiones resulta determinante para la aceptación o rechazo de todo un proyecto.

A continuación, se presenta la estructura de este trabajo de graduación:

- Capítulo 1. Antecedentes. En esta sección se realizó un estudio sobre cinco arquitecturas relacionadas con la gestión y entrega de contenido educativo.
- Capítulo 2. Justificación. En esta sección se da una explicación del por qué se realizó el proyecto que tiene como nombre diseño de la investigación para la consulta KISPEDIA.
- Capítulo 3. Alcances. En la sección de alcance se definieron claramente los puntos que se desea abarcar en el proyecto.
- Capítulo 4. Marco Teórico. En esta sección contiene toda la teoría, conceptos que se utilizaron para resolver el problema.
- Capítulo 5. Presentación de resultados. En esta sección se presentó los resultados de la solución que se propuso.
- Capítulo 6. Análisis y discusión de resultados. En esta sección se presentó, se comparó y se discutieron los resultados obtenidos tomando como referencia los resultados esperados.

1. ANTECEDENTES

1.1. Análisis y discusión de arquitecturas utilizadas por sistemas *e-learning*

El objetivo principal de este trabajo de graduación es diseñar una arquitectura de *e-learning* que cumpla con la consulta de contenido eficiente. Por lo que se analizaron cinco arquitecturas: *Basada en servicios web y agentes inteligentes* (Wei & Yan, 2009), *Árbol de Conocimiento* (Brusilovsky, 2014), *Una arquitectura de servicios web para contenido enriquecido desde clientes de aprendizaje móviles* (Parsons, Parsons, & University, 2006), *Arquitectura basada en la web para el aprendizaje móvil* (Alzaabi, Berri, & Jamal, 2010) y *entrega de material e-learning consciente del contexto de usuario: enfoque y arquitectura*(Schmidt & Winterhalter, 2003).

La arquitectura basada en servicios *web* y agentes inteligentes proporciona un modelo de integración flexible en el que todos los componentes de aprendizaje y aplicaciones están débilmente conectados y se pueden distribuir, a través de Internet. Además, mediante el uso de agentes, el contenido de aprendizaje se puede personalizar con inteligencia para adaptarse al contexto y las necesidades de aprendizaje de los usuarios. El árbol de conocimiento es una arquitectura basada en la distribución de actividades de aprendizaje inteligente y reusable. El objetivo de esta arquitectura es cerrar la brecha entre los enfoques modernos de educación basada en web y las tecnologías hipermedia adaptativas. Una arquitectura de servicios *web* para contenido enriquecido desde clientes de aprendizaje móviles una arquitectura móvil basada en servicios web. Arquitectura basada en la web para el

aprendizaje móvil es una arquitectura que hace uso de recursos web multimedia disponibles para la gestión de la creación y la entrega de material de aprendizaje para usuarios con teléfonos inteligentes. Y por último, *Entrega de material e-learning consciente del contexto de usuario: enfoque y arquitectura*, arquitectura que tiene como base el modelado semántico del entorno del alumno. Con este fin la arquitectura hace uso de ontologías y tecnologías de la Web Semántica para obtener contenido.

A continuación, se describe cada arquitectura detalladamente para determinar las fortalezas y debilidades existentes en el funcionamiento.

1.1.1. Basada en servicios web y agentes inteligentes

La arquitectura de *e-learning* se basa en servicios web y agentes inteligentes, como se muestra en la Figura 1. La tecnología de servicios web permite la selección y la integración de la funcionalidad requerida de un conjunto de servicios web de *e-learning* relacionados existentes. Al mismo tiempo, los agentes inteligentes se utilizan para permitir la entrega personalizada e inteligente de contenido de aprendizaje.

- Simple Object Access Protocol (SOAP)
- Universal Description, Discovery and Integration (UDDI) y
- Web Services Description Language (WSDL)

Se puede definir al agente inteligente como una entidad de software que, basándose en su propio conocimiento, realiza un conjunto de operaciones destinadas a satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos se lo requiere. Los atributos de un agente son:

- **Persistencia:** los agentes pueden funcionar constantemente sin comando externo.
- **Autonomía:** los agentes pueden seleccionar la tarea y tener la capacidad de decidir la prioridad de las tareas sin la intervención humana.
- **Habilidades sociales:** los agentes pueden comunicarse con otros componentes a través de la lengua agente y cooperar para completar una tarea.
- **Reactividad:** los agentes pueden hacer que las reacciones correspondientes de acuerdo con el entorno cambiante de percibir el contexto.

Los agentes de software inteligentes se integran al sistema *e-learning* para proporcionar personalización. Ya que pueden aprender los intereses del usuario, preferencias, hábitos, comportamientos y estilo de aprendizaje.

En general, hay varios grupos de personas que participan en un sistema de *e-learning*: autores, estudiantes, administradores y formadores.

En un sistema de *e-learning* tradicional, todos los módulos funcionales que necesitan los usuarios residen en un servidor. En esta arquitectura, Solamente un conjunto de módulos funcionales claves necesarios por los usuarios son los únicos que residen en el servidor. Todos los módulos funcionales se implementan como servicios *web* que se pueden integrar y reutilizar fácilmente. Por lo tanto, los sistemas de *e-learning* con esta arquitectura son altamente interoperables, flexibles y ligeros, y pueden extenderse por la elección de la funcionalidad requerida de *e-learning* de servicios *web* y se pueden utilizar de manera remota, a través de Internet de acuerdo con las necesidades del usuario.

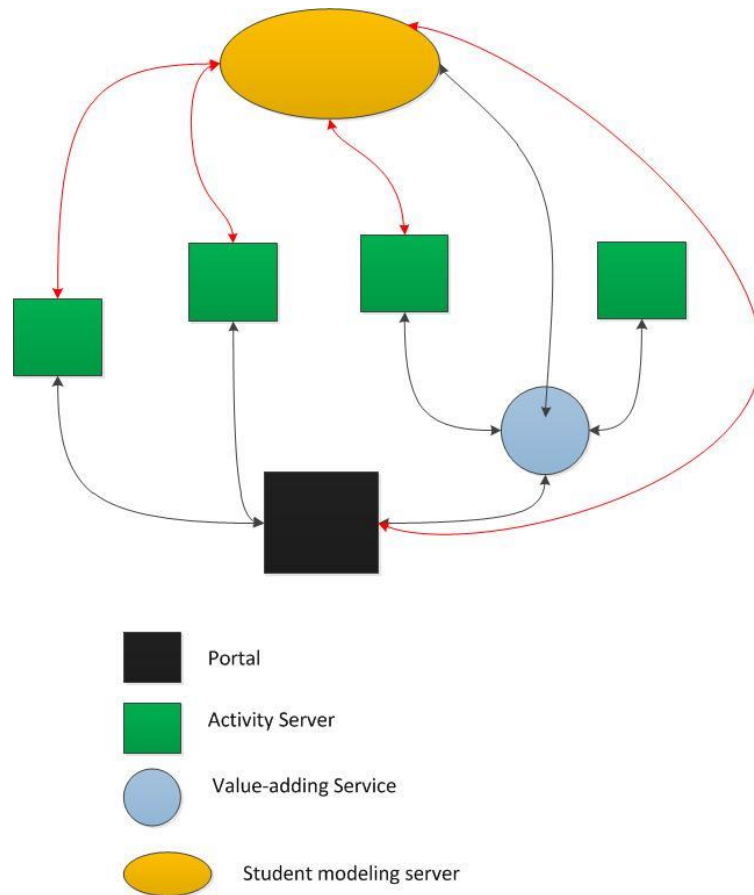
- Los módulos funcionales clave se describen a continuación:
 - Servicio de ingreso: Este servicio es responsable de la conexión del usuario. Una vez que el usuario de inicio de sesión en el sistema. El servicio de inicio de sesión buscará la base de datos de perfiles de usuario y se comunicará con el *user agent*.
 - Agente de usuario: el agente de usuario realiza las cuatro funciones siguientes: Reaccionar a *login service* y activar los servicios de búsqueda, abordar el perfil del usuario para obtener o almacenar la información del usuario, intercambiar información con el *authoring service* y con el agente de contenido.
 - Servicio de búsqueda: este servicio se encarga de buscar contenido propio de Internet.

- Servicio de gestión de contenidos: este servicio se comunica con el servicio de creación del contenido y luego se encarga de la transferencia del contenido al usuario.
- Servicio de creación: este servicio es el responsable de informar a los autores, para desarrollar los contenidos de aprendizaje y transferir contenido al servicio de gestión de contenidos.
- Servicio de entrega: después de preparar el contenido, el servicio de entrega se hará cargo de entregar contenido adecuado para el usuario.
- Otros servicios: Estos servicios se distribuyen a través de Internet y se pueden utilizar cuando sea necesario.

1.1.3. Árbol de conocimiento: Una arquitectura distribuida para el aprendizaje en línea (e-learning) adaptativo

La arquitectura del árbol es el conocimiento de una arquitectura para sistemas *e-learning* basados en la distribución de actividades de aprendizaje inteligentes y reusables. Esta arquitectura surge como consecuencia del éxito de los sistemas de gestión de cursos (*course management systems, CMS*) y pretende dar a poyo a los profesores y estudiantes en *e-learning*. Está compuesto básicamente por cuatro componentes: Servidor de actividades, portal de aprendizaje, servidor del modelado del estudiante y servidor de valor agregado como lo muestra la Figura 2.

Figura 2. Principales componentes de la arquitectura Knowledge Tree distribuida



Fuente: elaboración propia,

1.1.4. Componentes y funcionalidades

El portal es un componente de la arquitectura que se centra para dar apoyo de un curso completo. Interactuando con la funcionalidad familiar de un Sistema de Administración del Aprendizaje (*LMS*), proporciona una interfaz para el profesor y mantiene una interfaz en tiempo de ejecución para el estudiante.

La arquitectura *Árbol de Conocimiento* permite múltiples portales que pueden apoyar diferentes paradigmas y enfoques educativos al mismo tiempo que proporciona acceso al mismo universo de contenidos y servicios distribuidos.

El material de aprendizaje estático e interactivo no se almacena en el portal. El material estático es procesado por un servidor llamado AnnotatED mientras que el material interactivo por los servidores de actividad.

Servidor de Actividad forma parte de esta arquitectura, Se centra en contenido y servicios reutilizables. Desempeña un papel similar al de un repositorio educativo con los modernos métodos de reutilización de cursos, en el sentido de que alberga contenidos de aprendizaje reutilizables. La diferencia entre esto y un repositorio de aprendizaje tradicional son las siguiente.

Primero a diferencia de los repositorios estos almacena información estática, objetos simples, etc. Un servidor de actividad puede albergar contenidos de aprendizaje altamente interactivo y adaptativo. También puede albergar servicios de aprendizaje interactivos, como foros de discusión o anotaciones compartidas.

Segundo, un servidor de actividad utiliza de forma diferente volver a usar el contenido. Mientras que los objetos de aprendizaje simples se reutilizan por la copia y se inserta en los nuevos cursos, una actividad se reutiliza haciendo referencia y luego es entregado por un servidor.

La necesidad de servidores de actividad se deriva de la naturaleza de las actividades de aprendizaje avanzado de adaptación. Estas actividades no

pueden ser copiadas como archivos, tienen que ser utilizado desde servidores Web dedicados esto son mantenidos por los proveedores de contenidos.

El Servidor de Valor Agregado adopta las características de un portal y un servidor de actividad. Como portal es capaz de consultar los servidores de actividad y las actividades de acceso. Como un servidor de actividad puede ser accedida desde un portal.

Los servicios de valor añadido son mantenidos por los proveedores de servicios. Los servicios son neutros y pueden ser reutilizados en múltiples cursos para proporcionar bloques más grandes y son utilizados por los maestros que construyen el sistema de e-learning con la ayuda de un portal.

El contenido y los servicios se procesan en bruto permitiendo añadir funciones valiosas como secuencia adaptativa, anotación, visualización, o la integración de contenidos.

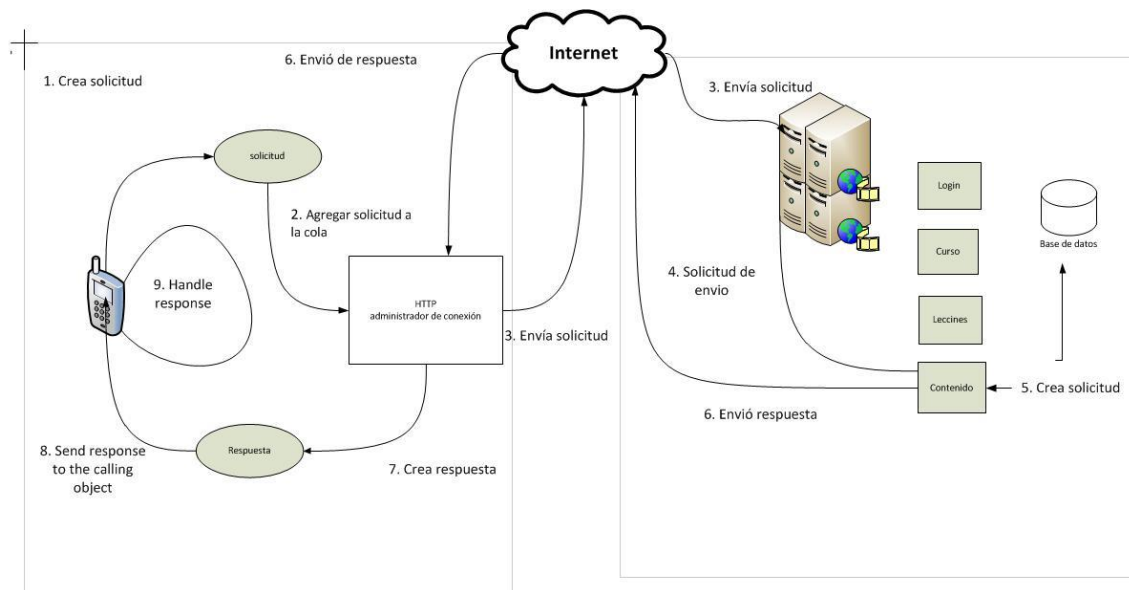
Las necesidades y las perspectivas de los estudiantes en el proceso de e-learning son representadas por el Servidor de Modelado del Estudiante. Este tipo de servidor permite distribuir *e-learning* altamente personalizado. Por medio de la recopilación del estudiante, datos sobre su desempeño en cada portal y cada servidor de actividad.

1.1.5. Una arquitectura de servicios web para contenido enriquecido desde clientes de aprendizaje móvil

La arquitectura es cliente-servidor y utiliza la tecnología Java para el mecanismo de comunicación, para capturar el modelo de datos del contenido

de aprendizaje y para manejar los aspectos de la interfaz de usuario. Los componentes de esta arquitectura se describen en la figura 3.

Figura 3. **El cliente proceso de interacción entre el servidor utilizando XML, a través de HTTP**



Fuente: elaboración propia.

- Los principales componentes son:
 - XML: es un lenguaje de marcas, utilizado para almacenar datos en forma legible para estructurar documentos grandes. XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integran información
 - HTTP: Es un protocolo de red, usado en cada transacción de la WWW. HTTP define la sintaxis y la semántica que utilizan los

elementos de software de la arquitectura *web* (clientes, servidores, proxies) para comunicarse.

- Servlet JAVA: Tecnología java y conjunto de estándares de red utilizada para dar más capacidad a los servidores, dicha tecnología utiliza el modelo de programación HTTP Request/Response. Los servlets son utilizados comúnmente para extender la funcionalidad de los servidores web. Los servlets son la contraparte Java de otra tecnología de contenido dinámico *web*, como PHP y ASP.NET.

El proceso comienza con el cliente Java ME al crear un objeto de solicitud que contiene información como el servlet de destino, la operación de servlets y los parámetros asociados.

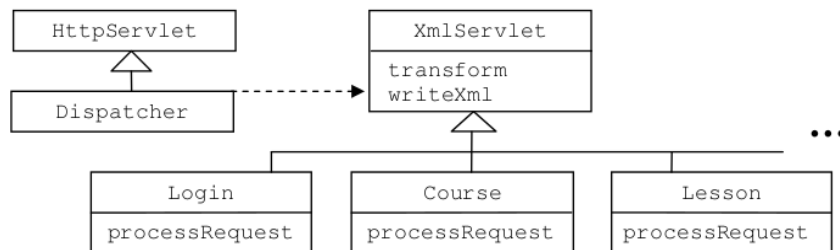
Una vez que el objeto de la petición se ha configurado, el cliente envía la solicitud al administrador de conexión HTTP, un componente que actúa como un proxy del lado del cliente para el servidor de montaje y desmontaje de los mensajes XML.

Desde aquí, el administrador de conexión HTTP envía la solicitud al servidor como un mensaje XML y se detecta una respuesta. El mensaje XML es recibido por el servlet *Dispatcher* que toma el mensaje y lo transforma en un objeto de solicitud para ser procesado por el servlet apropiado para gestionar el contenido del usuario finalmente este objeto de respuesta es enviado de vuelta al cliente, la respuesta es recibida por el administrador de conexión HTTP del cliente, la respuesta es XML que es transformado en un objeto respuesta que se envía al cliente.

Los componentes del lado del servidor son principalmente servlets Java, estos se encargan de procesar las solicitudes de cliente.

Cada tipo servlet de solicitud se ocupa de una funcionalidad específica; esencialmente la lógica de negocio se empaqueta en estas clases. El objeto de solicitud contendrá el nombre del método solicitado y parámetros asociados. El servlet consulta este objeto de petición y ejecuta el método apropiado. La mayoría de estos métodos se basan en la solicitud de información por lo que el servlet se comunicará con la capa de acceso a datos para obtener la información requerida. Cada objeto de datos implementa la interfaz XMLGenerator que contiene el método getXML (). Este método es el que utiliza el servlet para generar el XML de los objetos de datos. Una vez que el servlet tiene el documento XML apropiado lo enviará de nuevo al cliente.

Figura 4. **Implementación del *Patrón Front Controller* que procesa las solicitudes XML y las respuestas**



Fuente: elaboración propia.

El almacenamiento de contenidos en la base de datos es un patrón de diseño estándar para aplicaciones *web* dinámicas que tienen contenido que puede cambiar con frecuencia.

- Características:
 - Utiliza la base de datos relacional MySQL (MySQL AB 2006).
 - La asignación entre el modelo de objetos del lado del servidor y la base de datos se realiza utilizando la herramienta de mapeo objeto relacional HibernateCore (Hibernate 2006).
 - La parte del modelo que encapsula el contenido del curso proporciona la ayuda para ampliar el conjunto de tipos de contenido que se pueden utilizar en la aplicación.
 - El contenido a manejar y que se admiten es:
 - Texto (texto informativo básico para una lección en particular).
 - Multimedia como el vídeo o contenido de audio
 - XML como medio de mecanismo de transporte. RSS (Really Simple Syndication) es un formato XML para syndicar o compartir contenido en la web. Por lo tanto, el uso de XML como mecanismo de transporte abre la posibilidad de la integración de la aplicación en un estilo Web 2.0.
 - Un MIDlet es una aplicación J2ME para dispositivos móviles cuyas limitaciones entran dentro de la especificación MIDP. Debido a la filosofía Java ("*write one, run anywhere*") se puede ejecutar sobre un amplio rango de dispositivos sin necesidad de modificarlo.

La razón principal por la que seleccionaron XML como el formato del mensaje es debido a su flexibilidad, lo que permite diferentes tipos de cliente de volver a utilizar el mismo contenido desde el servidor.

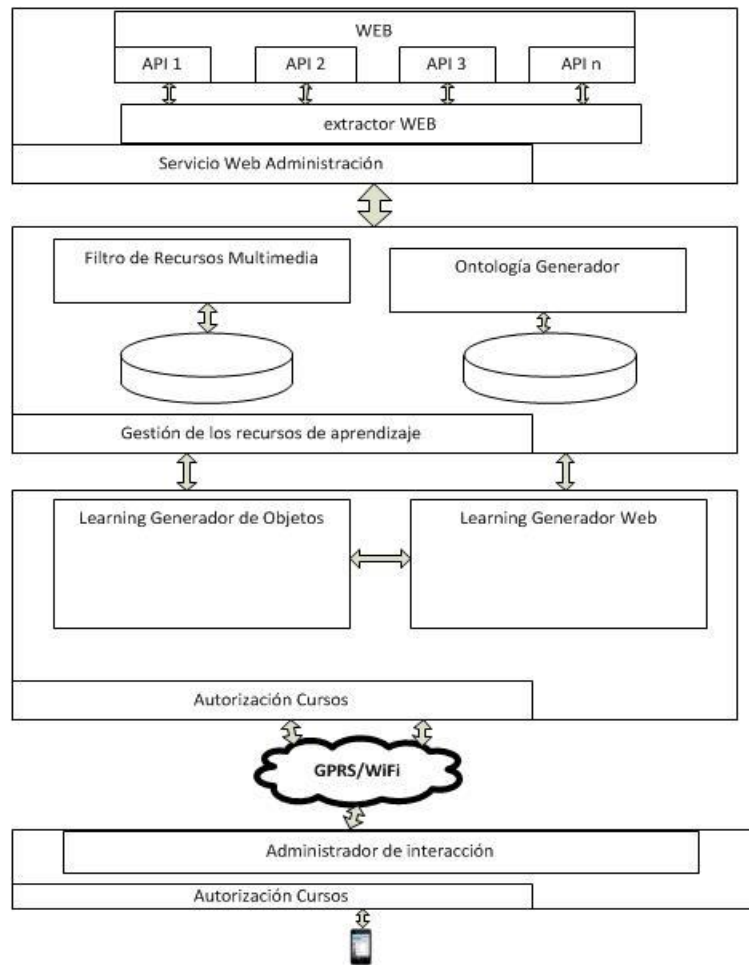
El uso de HTTP como protocolo de transferencia garantiza que un gran número de dispositivos puedan actuar como clientes, incluso los teléfonos móviles utilizando MIDP 1.0. Aunque en algunos dispositivos de Java ME pueden ser capaces de comunicarse con otros protocolos tales como el protocolo de datagramas de usuario (UDP).

- El tamaño de la aplicación del lado del cliente, los MIDlets tienen que ser instalado en pequeños dispositivos con memoria limitada que es compartido por otras aplicaciones, imágenes, juego.
- El consumo de ancho de banda de los tipos de contenido que se transportan y la velocidad resultante de la carga y descarga.
- El poder de procesamiento requerido y cómo eso podría tener un impacto en diferentes plataformas de cliente.
- Decidir la compensación entre el tráfico de red, el uso de memoria y espacio de almacenamiento.
- La interfaz de usuario de diferentes dispositivos móviles. J2ME Polish incluye una base de datos XML de más de 300 dispositivos móviles y pre-procesa el código fuente de Java para la presentación.

1.1.6. Arquitectura basada en la *web* para el aprendizaje móvil

La figura 5 se observa la arquitectura basada en la *web* para el aprendizaje móvil. La arquitectura presentada en este artículo es acerca del uso de recursos *web* multimedia disponibles para la gestión de la creación y la entrega de material de aprendizaje para usuarios con teléfonos inteligentes.

Figura 5. **Arquitectura del sistema**



Fuente: elaboración propia.

- La arquitectura está compuesta por cuatro módulos principales:
 - Administrador de servicios *web*
 - Gestión de recursos de aprendizaje
 - Creación de cursos
 - Interfaz gráfica de usuarios

La arquitectura se basa en el modelo cliente-servidor. En el lado del cliente tenemos la interfaz gráfica de usuario instalado en el dispositivo móvil que soporta ya sea una conexión GPRS o WiFi. El procesamiento principal se encuentra en el lado del servidor. El material se consulta de un conjunto de servicios *web*, es empaquetado y se estructura y luego se entrega al alumno. El usuario mediante un dispositivo móvil interactúa con el sistema mediante el envío de una consulta que incluye palabras clave que representan un tema específico en el que él o ella están interesados en aprender. El sistema guía al alumno indicando el flujo típico de aprendizaje a través del *Learning Web* (LW), pero permite al alumno elegir el camino de aprendizaje que él o ella desea lograr. En las siguientes secciones se describen los cuatro módulos del sistema.

El administrador de servicios *web* es responsable de la recopilación de material de aprendizaje desde la web en respuesta a una consulta del usuario. *Web Services Management module* (WSM) invoca diferentes servicios *web* disponibles para recuperar los recursos multimedia de aprendizaje para eso hace uso del Web extractor. Los recursos de aprendizaje obtenidos de la web se componen de texto, imágenes y videos.

Por lo tanto, con el fin de acceder a múltiples repositorios se ejecuta un programa específico que invoca recursos. Cada API es desarrollado por un proveedor de servicios web, disponible para acceder a su repositorio multimedia.

- Se utilizó tres repositorios *web* en el sistema:
 - Wikipedia para la construcción de ontologías y el contenido del texto,

- Yahoo para recuperar imágenes relacionadas con un tema,
- Youtube, el repositorio de vídeo de aprendizaje,

El extractor web se implementó utilizando Python, que es un lenguaje de programación que interpreta y es interactivo. Se utiliza para desarrollar aplicaciones de gran tamaño, especialmente para la web, ya que incorpora características de gran alcance para analizar las páginas web.

Gestión de Recursos de Aprendizaje (LRM) este módulo es el responsable de la filtración, el almacenamiento y la estructuración del material de aprendizaje recuperado por el Extractor Web. LRM consta de dos componentes principales:

- El filtro de Recursos Multimedia (MRF)
- y el generador de Ontología (GO).

El filtro de Recursos Multimedia, clasifica los recursos multimedia (imágenes y videos) consultados de acuerdo a lo más relevantes del usuario y su perfil. MRF utiliza una función de categoría muy básica; para seleccionar la primera información consultado por el Extractor Web, ya que es proporcionada por el proveedor de servicios. Sin embargo, esta función puede ser refinada, considerando factores relevantes del proveedor de servicios, tales como: el más reciente, los más vistos, los más votados y los más comentados.

Este módulo crea una ontología para cada sesión iniciada por una consulta del usuario. Se genera exclusivamente de la sección de contenido de Wikipedia. La *web* extractor consulta en Wikipedia el contenido almacenado en la base de datos y se recupera a petición del alumno. Web Extractor utiliza

principalmente numeración, título y sangría para generar la estructura física. Una vez generada la ontología, se almacena en una base de datos.

Couserware Authoring se encarga de empaquetar los *Objet authoring* para ser entregados al usuario y crear el *Learning web* definir el curso de aprendizaje y guiar la navegación del alumno.

Módulo de Autoría Course ware se compone de dos componentes:

- El generador de objetos de aprendizaje (LOG)
- Generador Web de Aprendizaje (GDP)

Este componente reúne todos los recursos de texto y multimedia a partir de la base de datos y genera un objeto de aprendizaje en función del perfil del usuario y las especificaciones del dispositivo móvil utilizado. Se definieron cuatro esquemas de "*Learning Object*" que son utilizados por el *Learning Object* generador de paquetes *Learning Objects*. Los esquemas que aseguran que cada categoría de aprendizaje reciba *Learning Object* que se adapte a sus preferencias específicas de aprendizaje. Los usuarios se han clasificado por su edad en cuatro categorías, cada una de ellas con un diseño específico. Por ejemplo, el diseño de la categoría "adolescentes" incluye un pequeño texto de introducción, una imagen, un hipervínculo a un streaming de vídeo y los enlaces a Los relacionada con el concepto actual como se aprecia en la figura 5.

Learning Web Generator asigna el *Learning Web* de la ontología. Mientras que la ontología es un árbol de conceptos, *el Learning Web* es un gráfico de *Learnig Objects* donde cada *Learning Object* es una materialización de un concepto en la ontología. *El Learning Web* guía al alumno. *Learing Object* aparece en el teléfono móvil, los hipervínculos a la posible siguiente y anterior

Learning Objects que él / ella puede tener acceso. De esta manera, la progresión de usuario en el material del curso se guía durante la navegación Learning Web. El usuario es libre de elegir cualquier ruta en *el Learning Web*.

La interfaz gráfica de usuario instalado en el dispositivo móvil del usuario incluye el gestor de Interacción (IM) que escucha las peticiones formuladas por el usuario antes o durante una sesión de aprendizaje y en consecuencia invoca el servicio adecuado. Además, IM gestiona la identificación del usuario, permitiendo el acceso a los usuarios registrados que tienen un perfil almacenado. Esta funcionalidad permite al sistema mejorar el material de aprendizaje entregado se adapte a cada usuario.

1.1.7. Entrega de material *e-learning* consciente del contexto de usuario: enfoque y arquitectura

E-learning se ha convertido en una importante idea de que el aprendizaje es básicamente la construcción y el perfeccionamiento de los conocimientos estructuras en las mentes de los alumnos.

Las soluciones actuales de *e-Learning* no son conscientes del contexto del alumno, es decir las características del individuo y el contexto de la organización, tales como los procesos de trabajo y tareas (por ejemplo, que proceso pedagógico o enfoque se ajusta mejor). Esto se puede lograr mediante objetos de aprendizaje modulares y metadatos semánticos para su contextualización. La arquitectura propone alcanzarlo mediante siete pasos principales que se describen a continuación.

- Descomponer cursos en unidades modulares (objetos de aprendizaje)
- Hacer objetos de aprendizaje adaptables

- Hacer que las relaciones semánticas y didácticas entre los objetos de aprendizaje explícito.
- Contexto modelo del alumno a lo largo de varias dimensiones (personal, organizacional, tónica, etc.) y sus requisitos de conocimientos.
- Contextualizar los objetos de aprendizaje de acuerdo con este modelo
- Adquirir conocimientos acerca de la situación del usuario y sus requisitos de conocimientos.
- Encontrar los correspondientes objetos de aprendizaje y adaptar su estructura interna para el alumno.

1.1.8. Arquitectura del sistema

El enfoque se basa en gran medida en el modelado semántico del entorno del alumno. Para lograr este fin la arquitectura propone el uso de ontologías y tecnologías de la Web Semántica.

Con el fin de mantener la tarea de modelado manejable, se divide la ontología en varias sub-ontologías.

- Ontología Organizacional (papeles, departamentos)
- Proceso de la ontología (representaciones de flujo de trabajo)
- Ontología de tareas
- Ontología área de conocimiento

1.1.9. Arquitectura componente y su función

Objetos de aprendizaje (LO) es la unidad mínima de contenido de aprendizaje pedagógicamente razonable. Este objeto de aprendizaje consiste en contenido (como texto, imágenes, animaciones, secuencias de vídeo, etc.).

Los metadatos (LOM) describen los objetos de aprendizaje.

La arquitectura hace uso de cuatro principales roles de usuario: Coordinador (para la gestión de la ontologías y propiedades globales), Autor (para la creación de objetos de aprendizaje), estudiantes y administradores (para la administración del sistema).

- Editor Ontología. Herramienta para editar otologías.
- Herramienta de edición.

Esta herramienta no está diseñada para crear contenido existente, sino para la creación de objetos de aprendizaje, a partir de contenido existente y contextualizarlos con referencias a la ontología. Los objetivos y requisitos previos de un objeto de aprendizaje tienen que ser especificados.

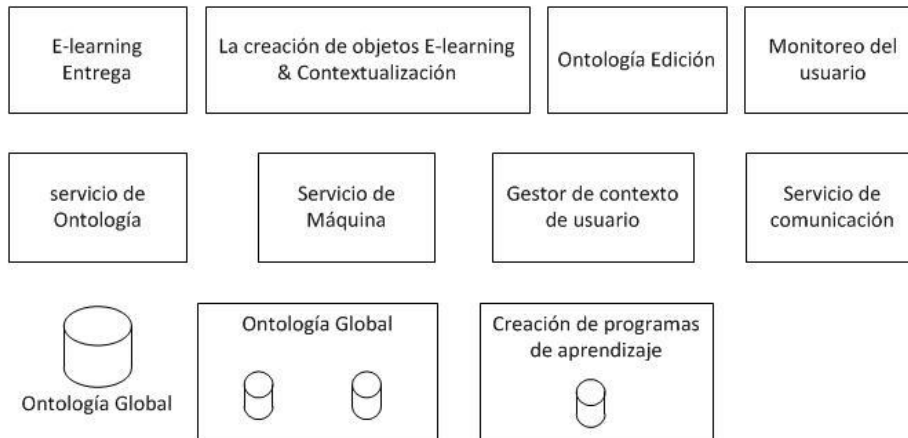
- Plataforma de entrega

Este la interfaz del alumno en donde se le hace entrega de objetos de aprendizaje y gestión de progreso en el aprendizaje.

- Monitoreo del usuario

Permite la captura de información sobre las acciones del usuario.

Figura 6. **Arquitectura de alto nivel del sistema**



Fuente: elaboración propia.

A continuación tenemos una capa de servicios que contienen la lógica de la aplicación.

1.1.10. Servicio de ontología

Este servicio proporciona capacidad de persistencia y de consulta para los datos ontológicos. También ofrece posibilidades para definir puntos de vista y registrar factores desencadenantes que deban ejecutarse cuando ocurren ciertos cambios.

1.1.11. Servicio matching

La tarea de este servicio es reunir la situación del usuario y los objetos de aprendizaje disponibles. Para ello, calcula los requisitos de conocimiento del contexto actual y los controles de objetos de aprendizaje que pueden ofrecer las competencias que faltan.

1.1.12. Creación del programa de aprendizaje

Este componente se encarga de compilar programas completos, que contiene todo lo necesario para comprender satisfactoriamente un conjunto de objetos de aprendizaje.

1.1.13. Servicio de comunicación

Dado que el aprendizaje se ha reconocido que es altamente dependiente de las interacciones sociales, la comunicación es una característica muy importante en el sistema. Se cuenta con infraestructuras de mensajería como groupware y el sistema de mensajería instantánea, sino porque sabemos lo que hacen los usuarios, lo que aprenden.

Dos de estos servicios, que constituyen las principales innovaciones de LIP, son el servicio de búsqueda y el servicio de la creación de programas de aprendizaje, que se describirá a continuación con más detalle.

La capa inferior consta de los siguientes servicios:

1.1.14. Repositorio de ontología

Este es el almacenamiento persistente real de los datos ontológicos (basados en las tecnologías de bases de datos).

1.1.15. Administrador objetos *Learning*

Este componente almacena objetos incluidos sus metadatos y su contenido y los hace accesibles para los servicios de la capa superior de aprendizaje.

1.1.16. Administrador de contenido del usuario

El gestor de contenido del usuario es una parte muy importante del sistema. Sólo el conocimiento sobre el usuario permite que el sistema pueda sugerir material de aprendizaje apropiado para el usuario. El gestor de contenido del usuario recoge toda la información suministrada por diferentes fuentes de contenido y proporciona una vista agregada, teniendo en cuenta la imperfección de la información.

2. JUSTIFICACIÓN

El presente trabajo se desarrolló en la línea de investigación Tecnologías de la Información y la Comunicación para el apoyo a la educación. Es un tema extenso, pero es evidente que el avance de estas tecnologías informáticas y de comunicación, han influenciado en lo académico y en otras aéreas a nivel mundial, por tales razones se ha convertido en un medio valioso para impartir enseñanza de calidad.

Por tal motivo, estos recursos tecnológicos a la educación son de gran utilidad porque produce enriquecimiento en el desarrollo de nuevas formas de aprendizaje. Cada vez es necesario buscar nuevas formas de aprendizaje, cambiar la enseñanza y aprendizaje tradicional por nuevas herramientas educativas que permitan hacer el uso de tecnologías que apoye a la enseñanza, con el fin de optimizar el proceso de aprendizaje.

Diseño de la investigación del sistema para la consulta de contenido Kisdpedia es un proyecto innovador, porque hace uso de recursos tecnológicos con la finalidad de apoyar la enseñanza y aprendizaje de los niños de Guatemala, utiliza conceptos modernos de aprendizaje se da por la combinación de la informática y las telecomunicaciones. Es así como la computadora, los dispositivos móviles están apoyando actualmente al estudiante, proporcionándole grandes herramientas como el acceso a importantes fuentes de información. Sin embargo, las nuevas tecnologías no siempre son incluidas de la manera eficiente, y muchas veces su aplicación genera descontentos por los nuevos problemas en lugar de facilitar los procesos de aprendizaje.

Diseño de la investigación del sistema para la consulta de contenido Kidspedia es un proyecto tecnológico intuitivo y fácil de utilizar con información confiable y actualizada, pero es necesario que los profesionales relacionados con la educación conozcan las TICs para poder tomar decisiones adecuadas y orientar a sus alumnos. Deben de existir más proyectos innovadores utilizando las tecnologías y de esa manera se puede lograr una mejora en el aprendizaje.

3. ALCANCES

3.1. Alcances investigativos

Con la finalidad de alcanzar los objetivos que se propone previo a iniciar el diseño del sistema propuesto se realiza la investigación sobre arquitecturas de software en general, se analizan arquitecturas de éxito, interfaz de usuario, estándares de diseño de interfaz, prototipo de diseño de software, modelado de casos de uso, modelado de base de datos.

3.2. Diseño de la arquitectura de *software*

El alcance consiste en la selección, descripción y diseño de los componentes de la arquitectura para el sistema. La arquitectura del software es la parte medular de todo software y cuando se desea construir un sistema debe hacerse de la mejor manera para construir un software de calidad, por lo tanto es de utilidad tener referencias de arquitecturas de éxito. El prototipo que se creará no utilizará la arquitectura de *software* que en este proyecto se diseña.

Para la selección y descripción de la arquitectura se partirá del análisis de cinco arquitecturas y se tomarán características importantes para el proyecto. Elaboración de diseño de alto nivel de la arquitectura que describe los componentes principales del sistema y cómo interactúan entre sí se incluirá en la sección de presentación de resultados.

El diseño de la arquitectura para este proyecto su base son los componentes, se realizará de tal forma que permita su mantenimiento y su

evolución fácilmente. Cada componente de la arquitectura es un objeto de software destinada a interactuar con otros componentes, los principales componentes que se diseñaran son los siguientes: servidor de base de datos, servidor de contenido dinámico, servidor de servicios y servidor de comprobación.

En el alcance también incluyen en el diseño de software los siguientes artefactos que forman parte y apoyan en proveer una visión más amplia de la arquitectura de software a los interesados y son lo siguiente:

- Diagrama del sistema para la consulta de contenido Kidspedia.
- Diagrama bloque de sistema.
- Diagrama de contexto del sistema
- Diseño de casos de uso
- Diseño de base de datos
- Diseño e Interfaz de usuario

3.3. Diseño de la interfaz de usuario

La interfaz de usuario es un elemento importante en el software, mediante la interfaz de usuario, ellos se comunican con el sistema. La importancia radica en que si los usuarios no están contentos con la interfaz sin importar el poder computacional del sistema. El sistema fracasará.

Debido a la importancia de la interfaz de usuario en este proyecto se desarrolló un prototipo haciendo uso de herramientas que permitan la realización de prototipos de una forma fácil, sencilla y ordenada. El prototipo todas sus pantallas tendrán consistencia, cada pantalla estará diseñada y formada por: estilos de letras, texto, menús, la agrupación de los menús,

iconos, cuadros con información, etc. El prototipo otorga al usuario únicamente lo necesario para navegar e interactuar en toda la aplicación. La información que se utilizara en la interfaz se almacenara en variables porque el prototipo no se conectara a un sistema de base de datos. Una vez finalizado el prototipo se procederá a probar y revisar los distintos diseños de interfaz con los usuarios, se involucrará a los usuarios con la finalidad de obtener retroalimentación del diseño.

Para la elaboración del diseño se utilizan herramientas de diseño, el alcance de investigación incluyen la investigación de las herramientas de diseño que se utilizan para la elaboración. Para dar inicio con el diseño debe completarse toda la investigación requerida.

En el alcance del diseño de interfaz incluye la identificación de usuarios, tareas y requisitos del entorno que corresponde un análisis de requerimientos. En la fase de diseño de casos de uso, se analizará cada uno de los escenarios y se definirán un conjunto de objetos y acciones de interfaz, con la información obtenida se utilizará para la elaboración del diseño preliminar de interfaz denominados prototipos de interfaz.

3.4. Alcances técnicos

Utilizar herramientas que permita diseñar un *software* de acuerdo a las necesidades del negocio y las soluciones requeridas de los usuarios.

Utilizar la herramienta Microsoft Visio 2007, para la elaboración del diseño de la arquitectura de software a utilizar para el sistema.

Utilizar las herramientas Justinmind y Phoshop para la elaboración del prototipo, con la finalidad de evaluar el diseño de interfaz con ello se pretende obtener una retroalimentación y es en esos momento en donde pueden surgir problemas, que quizás lleven a rediseñar alguna parte del modelo, o incluso a desecharlo completamente por no ajustarse a los requerimientos señalados. El prototipo que se creó cuenta con diferentes objetos de interfaz los cuales son: menús, ventanas, iconos, colores, imágenes, contenidos gráficos. La información en las diferentes pantallas es almacenada en variables, ya que no es posible la utilización de una base de datos en la herramienta que se utilizó.

Utilizar la herramienta Microsoft Visio 2007, para el diseño de casos de uso, que muestren por completo las interacciones entre los usuarios y el sistema para iniciar su desarrollo.

Utilizar las herramientas, Navicat, Erwin, Microsfot Visio 2007, para diseñar el esquema de base de datos que se utilizará en el software educativo que sea más eficiente en el manejo del contenido del software.

3.5. Resultados esperados

La selección y descripción de la arquitectura de software a utilizar para el sistema. De la selección final se obtendrá un diseño de arquitectura de software que facilite la comunicación, interpretación entre interesados del sistema, en el diseño destacará las decisiones que tendrán profundo impacto en el éxito y permita proporcionarle al software las capacidades necesarias para su mantenimiento y evolución, de acuerdo a las necesidades del negocio y las soluciones requeridas de los usuarios. Adicionalmente, los siguientes diseños;

- Diseño de la interfaz amigable del *software* educativo para que su utilización sea fácil y aceptado por la mayoría de los usuarios.
- Diseño de casos de uso, que muestren por completo las interacciones entre los usuarios y el sistema para iniciar su desarrollo.
- Diseño de esquema de base de datos que se utilizará en el *software* educativo para que sea más eficiente en el manejo del contenido de *software*.

El resultado del trabajo es el diseño sobre un *software* educativo para el apoyo a la educación básica guatemalteca con las siguientes características: intuitivo, fácil de utilizar, atractivo, adaptable y con información confiable y actualizada. Contenido de educación básica en Guatemala se refiere al contenido de educación primaria para niños de siete a doce años que el ministerio de educación establece, con el fin de apoyar al cumplimiento curricular el contenido para alumnos y alumnas será sobre lectura, escritura, matemática, expresión y comprensión oral, el sentido artístico, creativo y cultural, conocimiento del medio natural y social.

En resumen, los resultados esperados son definir los procesos, técnicas y principios con el propósito de definir un sistema, con los detalles necesarios y suficientes para su interpretación y realización física por personas con el perfil adecuado.

4. MARCO TEÓRICO

4.1. Plataformas aplicadas al soporte de contenido educativo

Actualmente existen aplicaciones educativas utilizadas para dar apoyo a cualquier nivel educativo, comúnmente se utiliza para universitarios también existe algunas que posee características especiales para personas especiales porque no todas las personas aprenden de la misma forma. Hoy en día obtener conocimiento a distancia es fácil, rápido y sencillo. Se debe a las nuevas tecnologías que brindan variedad de herramientas útiles como lo es el correo electrónico, blogs, foros y las aulas virtuales. El obtener conocimiento se convierte en ubicuo.

Se aprende cualquier cosa, en cualquier momento y en cualquier lugar utilizando tecnologías e infraestructuras de informática ubicua. Uno de los objetivos últimos de la enseñanza es incrementar la calidad de la vida diaria. Así, el sujeto esencial de aprendizaje existe en nuestro ambiente diario, no en aulas o libros de texto. Tradicionalmente, es muy difícil aprender desde el entorno habitual, porque no se tiene método para ello. Recientemente, el desarrollo de la tecnología de informática ubicua permite compartir información y comunicarse sin esfuerzo, constante y continuamente a lo largo del día. (De Castro Lozano, 2012).

4.2. Sistema de gestión de contenido

Es una herramienta de software comúnmente apoyado por una base de datos que permite la creación y administración de contenido web dinámico por los usuarios y en donde el cliente o usuarios finales, tienen a su disposición

contenido y material de aprendizaje. Su principal tarea es difundir recursos de aprendizajes. Ejemplos; Portales, Blogs, Tiendas online, Galerías de imágenes.

4.3. Sistema de gestión de aprendizaje

“Learning Management System (LMS) o Sistema de Gestión del Aprendizaje, es un software instalado generalmente en un servidor web (puede instalarse en una intranet), que se emplea para crear, aprobar, administrar, almacenar, distribuir y gestionar las actividades de formación virtual (puede utilizarse como complemento de clases presenciales o para el aprendizaje a distancia).”(Clarenc, Castro, López, Moreno, & Tosco, 2013)

LMS son también conocidos como e-learning o campus virtual por ser un espacio virtual de aprendizaje orientado a facilitar la experiencia de la toma de conocimiento a distancia, comúnmente utilizados para instituciones educativas también para empresas que necesitan capacitar a su personal.

La herramienta de software que se encarga de crear contenido de aprendizaje es el LCMS (Learning Content Management System). Un LMS tiene como función principal gestionar contenidos creados por diversas fuentes. Son utilizados por los maestros para complementar su material de clase y los alumnos que acceden a la herramienta para desarrollar sus tareas o completar sus conocimientos.

4.4. Entornos virtuales de aprendizaje

Son aplicaciones que se basan en la WEB. Son herramientas que brindan a los maestros y alumnos, recursos que mejoran la entrega y gestión de la educación.

4.5. Arquitectura de software

Partiendo de los siguientes conceptos de autores reconocidos, se entiende que la arquitectura de *software* es importante a la hora de desarrollar aplicaciones y sobre todo si son complejas. Adoptar la arquitectura correcta para un sistema de software puede ser crítico cuando se quiere alcanzar los objetivos que se establecen como lo son: la escalabilidad, flexibilidad, facilidad de mantenimiento, etc. actualmente es indispensable porque los programas de computadoras son cada vez más complejos, ya no son simples programas o software de varias líneas.

- Definiciones de varios autores que valen la pena destacar:

“La arquitectura de un sistema de software intensivo es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externamente visibles de los elementos, y las relaciones entre ellos.”(Bachmann, *et al.*, 2010)

La arquitectura de software de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos. (Pressman, 2010). La arquitectura de un software se conforma por componentes de un sistema, interfaces de comunicación para que se interactúen los componentes.

En otras palabras, los sistemas complejos actuales se forman de subsistemas que componen el sistema, las interfaces de comunicación de los sistemas y las reglas de interacción entre ellos.

4.6. La importancia de la arquitectura de software

Es importante porque cada vez los requerimientos del software actualmente son complejos por diversas razones y la representación de la arquitectura del software hace que la comunicación entre los interesados sea fácil y ayuda a la toma de decisiones desde su inicio de diseño. Un importante actor destaca y menciona tres importantes razones y que al utilizarla proporciona ventajas a la hora de diseñar la arquitectura de software.

Las representaciones de la arquitectura de software facilitan la comunicación entre todas las partes interesadas en el desarrollo del sistema de software (Pressman, 2010).

La arquitectura destaca decisiones tempranas de diseño que tendrán un profundo impacto en todo el trabajo de ingeniería del software, y es importante en el éxito final del sistema a la hora que sea utilizado (Pressman, 2010).

La arquitectura establece un modelo pequeño e intelectualmente comprensible de cómo está estructurado el sistema y de cómo trabajan juntos sus componentes (Pressman, 2010).

4.7. Diseño arquitectónico

Es la representación de la arquitectura de software detallando sus diversos aspectos utilizando modelos o vistas. Cada arquitectura de software debe describir sus aspectos y para que sea más comprensible por los distintos involucrados deben utilizarse distintos modelos o vistas. Cada modelo o vista

describe de diferente forma la misma arquitectura, cada vista debe ser coherente entre sí porque se trata del mismo sistema.

Las descripciones se hace utilizando notaciones y lenguaje existentes para representar el diseño de artefactos de software. Estas notaciones son generalmente usadas durante la etapa de diseño del software. En la etapa del diseño arquitectónico en muchas ocasiones se conciben ideas y decisiones importantes y que tendrá un impacto profundo. El diseño arquitectónico es como un plan de diseño, es fácil de comprender lo que se requiere del nuevo sistema y es una manera de documentar los acuerdos entre los clientes, los que desarrollan el sistema y las demás partes involucradas (Sommerville, 2005).

4.8. Estilos arquitectónicos

Cuando se inicia un proyecto de software no siempre es necesario crear una nueva arquitectura, con fines prácticos se puede adoptar una arquitectura siempre que cumpla con las necesidades del sistema a desarrollar.

Cada uno de estos estilos arquitectónicos describe clase de arquitectura o piezas de las arquitecturas. Cuando surgió la arquitectura del software se generalizó la arquitectura cliente-servidor, después aparecieron otros estilos arquitectónicos como las capas, la que se basa en componentes y otros. Cada estilo contiene un conjunto de componentes, base de datos y módulos que realizan una función que se requiere en el sistema.

Un conjunto de conectores que permiten la comunicación, coordinación y cooperación entre los componentes. Restricciones que definen como se integran los componentes para formar el sistema. Modelos semánticos que

permiten que un diseñador entienda las propiedades generales del sistema al analizar las propiedades conocidas de su parte (Pressman, 2010).

Aunque se han creado muchos sistemas, los patrones arquitectónicos comúnmente utilizados para el software en su mayoría pueden clasificarse.

4.9. Arquitecturas centradas de datos

El principal componente de esta arquitectura es una un fuente de datos. Pudiendo ser un documento o una base de datos al que otros componentes acceden con frecuencia para actualizar, añadir, borrar o bien modificar los datos.

- Arquitecturas de flujo de datos. Esta arquitectura se aplica cuando los datos de entrada son transformados a través de una serie de componentes computacionales o manipulativos en los datos de salida (Pressman, 2010).
- Arquitecturas estratificadas. Se crean diferentes capas y cada una realiza operaciones que progresivamente se aproximan más al cuadro de instrucciones de la máquina. En la capa externa, los componentes sirven a las operaciones de interfaz de usuario. En la capa interna, los componentes realizan operaciones de interfaz del sistema. Las capas intermedias proporcionan servicios de utilidad y funciones del software de aplicaciones (Pressman, 2010).

Arquitecturas de llamada y retorno. Este estilo arquitectónico permite al diseñador del *software* (arquitecto del sistema) construir una estructura de programa relativamente fácil de modificar y ajustar a escala (Pressman, 2010).

4.10. Base de datos

Utilizados en los diversos sistemas informáticos y son utilizados en distintos campos, desde pequeñas empresas hasta grandes organizaciones incluso en los hogares. Su uso se da por la necesidad de manipular datos y por ser una solución factible para el manejo de grandes volúmenes de datos y la complejidad de su manejo como es la extracción de datos y la concurrencia de datos, en otras palabras el acceso simultáneo a los datos y otras características.

Sin embargo, no siempre el uso de la base de datos es eficiente y se debe a la deficiencia en el diseño en ese caso proporcionará información no actualizada, errónea o contradictoria. Con la finalidad de maximizar sus beneficios potenciales, es importante comprender los fundamentos teóricos, estructura interna, diseño y gestión de las bases de datos (M. Ricardo, 2009).

4.11. Definición de base de datos

La base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular (SILBERSCHATZ, ABRAHAM; F. KORTH, HENRY, 2007).

4.12. Sistema de gestión de base de datos (SGBD)

Los Sistemas de Gestión de Base de Datos (DataBase Management System) Son un tipo de software que permite definir, construir y manipular los datos para diversos propósitos, sirve de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. La mayor parte de estos datos es de importancia para el individuo u organización; en otras palabras (J. Date, 2001).

Un sistema de base de datos sirve para integrar los datos. Lo componen los siguientes elementos (SILBERSCHATZ, ABRAHAM; F. KORTH, HENRY, 2007):

- Datos

Son los que se requieren almacenar y los metadatos también son datos que describir lo que se almacena en la base de datos. La característica fundamental de la información es de estar integrada y compartida.

La característica integrada se refiere a que la base de datos también se le puede considerar como una unificación de varios ficheros de datos y que son tratados como uno solo, y en el que es posible la eliminación totalmente, en partes la redundancia de datos. Y por compartida, se refiere a que los datos pueden ser compartidas entre varios usuarios con la posibilidad de que estos usuarios puedan acceder en el mismo tiempo, la misma información. En informática, a esto se le denomina concurrencia.

- Hardware

El *hardware* son los dispositivos físicos y sobre estos dispositivos físicos reside la base de datos. El hardware se compone de procesador, memoria principal, espacio de almacenamiento en disco de grandes dimensiones. Las unidades de disco constituyen el principal mecanismo de almacenamiento para las bases de datos y son indispensables, ya que permiten el acceso directo, sin él no será posible el procesamiento de las bases de datos.

- Software

Un sistema de base de datos incluye diversos tipos de software principalmente el administrador de base de datos (DBMS) que es el encargado de administrar la base de datos, por medio de este software se controlan las solicitudes de acceso y peticiones a la base de datos tales como agregar, eliminar, recuperar y almacenar los datos. Software de aplicación, que usa las facilidades del SGBD para manipular la base de datos y las herramientas utilizadas para desarrollar aplicaciones, aplicaciones como diseñadores de pantallas, menús, entre otros.

4.13. Modelo de datos

Un modelo de datos es una colección de conceptos utilizados para describir datos, sus relaciones, su semántica y las restricciones de consistencia. Principalmente establece el cómo se almacena, organiza y manipulan los datos. Los más comunes se encuentran (SILBERSCHATZ, ABRAHAM; F. KORTH, HENRY, 2007):

- Modelo relacional. Es el modelo de datos más utilizado actualmente en aplicaciones comerciales de procesamiento de datos, y una gran mayoría de sistemas de bases de datos actuales se basan en el modelo relacional. Se basa en la teoría de conjunto. Para representar los datos el modelo relacional utiliza tablas bidimensionales denominadas relación. Las tablas se representan gráficamente como una estructura rectangular formada por filas y columnas. Cada columna almacena información sobre una propiedad determinada de la tabla denominada atributo. Cada fila representa una ocurrencia de la relación el cual se le llama tupla.
- Modelo entidad-relación. Es un modelo de datos el cual es posible la abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos llamados entidades y relaciones. El modelo de datos es posible su representación, a través de diagramas.

Una entidad es un objeto o cosa que representa el mundo real del cual se abstrae información para luego almacenarla. Las entidades se componen de atributos que son los datos que definen al objeto. Del conjunto que no se repite se le llama clave de la entidad.

Relación son las relaciones que se dan entre las entidades el cual permite crear una dependencia entre varias entidades. La relación permite exigir que varias entidades compartan algunos atributos de forma indispensable.

- Modelo de datos orientado a objeto. Se basa en el paradigma del concepto de programación orientada a objetos y utiliza el concepto del paradigma orientado a objetos como la herencia, la identidad de los objetos, la encapsulación u ocultación de la información. Hace uso de métodos que ofrecen interfaz para el manejo de los objetos.

El modelo de datos orientado a objetos también soporta un sistema elaborado de tipos, incluidos los tipos estructurados y las colecciones. El modelo orientado a objetos es considerado una extensión del modelo E-R utilizando los conceptos de objetos.(SILBERSCHATZ, F. KORTH, & SUDARSHAN, 2007).

4.14. Lenguaje de base de datos

Los sistemas de bases de datos proporcionan un lenguaje de definición de datos para especificar el esquema de la base de datos y un lenguaje de manipulación de datos para expresar las consultas y las modificaciones de la base de datos. En la práctica, los lenguajes de definición y manipulación de datos no son dos lenguajes diferentes; en cambio, simplemente forman parte de un único lenguaje de bases de datos, como puede ser el muy usado SQL (SILBERSCHATZ, ABRAHAM; F. KORTH, HENRY, 2007).

4.15. Caso de uso

Los casos de uso es una herramienta utilizada para el modelado de los requerimientos del sistemas de software y forma parte de del *Unified Modeling Language* (UML), que es una notación basada en diagramas adoptada como estándar para describir sistemas de software. UML es una técnica para la especificación de sistemas en cada una de sus fases, se utiliza con éxito en diversos diseños de sistemas de las diversas industrias.

- Descripción de caso de uso se hace textualmente. Consistente en la descripción de cómo interactúan los actores y los casos de uso. Se enfoca en el comportamiento externo del sistema, ignorado lo que

sucede realmente dentro del sistema. El objetivo es describir como inicia el caso de uso, que actor inicia la ejecución del caso de uso, el flujo de mensajes entre actores y el caso de uso, flujo alternativo en el caso de uso y como finaliza el caso de uso con un valor para el actor.(Pressman, 2010).

Un caso de uso representa una funcionalidad completa tal como la percibe un actor. Un caso de uso en UML se define como una secuencia de acciones que realiza un sistema y que conduce a un resultado perceptible.

4.16. Diagrama de caso de uso

Forma parte de UML. Contiene elementos gráficos que modelan el sistema, permiten identificar rápidamente los actores externos del sistema y las formas básicas en que lo utilizan. Un diagrama de caso uso contiene: actores, casos de uso y las diferentes relaciones entre estos elementos. A continuación se definen los elementos del diagrama de Caso de uso.(Fowler, Kobryn, Booch, Jacobson, & Rumbaugh, 203)

- Actores: Es toda entidad externa al sistema que tiene una relación e interactúa solicitado una funcionalidad. Esto incluye personas, sistemas o maquinas etc. Es importante destacar que el actor representa un rol frente al sistema, no a un usuario individual en el sistema.
- Caso de uso: Representa una funcionalidad perceptible por el usuario.
- Relaciones entre actores y caso de uso: Relación de extensión se da cuando se crea un caso de uso a partir de un caso existente, añadiendo nuevas acciones, con la finalidad de extender la funcionalidad del caso

de uso. Relación de inclusión cuando se repiten acciones en varios casos de uso, es posible la agrupación de las características en común, y agruparlos en un nuevo caso de uso que será aprovechado en otros casos de uso. Relación de generalización similar a la relación de extensión utilizada únicamente para casos de uso y no para actores, es utilizada para ampliar la funcionalidad o refinar la funcionalidad mediante a un agregado para representar una funcionalidad diferente a la original.

4.16.1. Diseño de interfaz de usuario

“Si el software es difícil de usar, fuerza al usuario a cometer errores, o si frustra sus esfuerzos para alcanzar las metas, entonces no le gustará, sin que importe el poder computacional que tenga, el contenido que entregue o las funciones que ofrezca. La interfaz tiene que estar bien hecha porque moldea la percepción que el usuario tiene del software.” (Pressman, 2010) .

Obviamente se debe cuidar cualquier detalle de la aplicación y no olvidar la parte funcional. Porque otra funcionalidad de la interfaz importante es la de servir como capa donde nacen todas las interacciones entre el usuario y lo funcional de sistema.

Es muy importante que el sistema cuente con una buena interfaz, porque es el medio por el cual el ser humano se comunica con el sistema. Si el software tiene una interfaz difícil de usar, el usuario comete muchos errores al usarlo, el usuario se esfuerza utilizando la aplicación y no consigue sus metas, es seguro que el sistema fracasará no importando el poder computacional que tenga, el contenido que entregue o las funciones que ofrezca. La interfaz tiene que estar bien diseñada.

4.16.2. Diseño centrado en el usuario

Esta filosofía de diseño se relaciona con un conjunto de metodologías y técnicas que tienen como objetivo común el usuario. Comprender sus necesidades, limitaciones, comportamiento. Todas las características del usuario. Para lograr los objetivos establecidos de diseño de interfaz de usuario se sugiere seguir con un conjunto de lineamientos, principios de interfaz. Básicamente tres reglas doradas (Pressman, 2010,266) las cuales son: Dejar el control al usuario, reducir la carga de memoria del usuario y hacer que la interfaz sea consistente.

4.16.3. Proceso de diseño de interfaz de usuario

Para el proceso de diseño de interfaces gráficas se identifican los usuarios, las actividades que realizan cada usuario por medio de caso de uso, un análisis de requisitos. Luego de Identificar las actividades de los usuarios en los casos de uso, se analiza cada uno de los escenarios y se define un conjunto de objetos y acciones de interfaz.

Del análisis se extrae información que servirá para la creación del diseño inicial de la interfaz. Indicando iconos, colores, fuentes, texto, ventanas, objetos gráficos en general y especificaciones de elementos de menús. Para ello, se utilizan herramientas para crear prototipos de interfaces gráficas.

4.16.4. Principios generales de diseño de interfaz de usuario

Existen principios para el diseño de aplicaciones. Los principios heurísticos de Jakob Nielsen son conocidos pero no son los únicos, también existen otros expertos que han hecho buenas propuestas sobre diseño de interfaces de usuario, tal es el caso de Bruce Tognazzini que se mencionará a continuación. (Pressman, 2010,285)

- **Anticipación** Este principio sugiere anticiparse a las necesidades del usuario, mostrando al usuario toda la información y herramientas que le serán útiles en cada momento.
- **Comunicación.** La interfaz debe comunicar el estado de cualquier actividad iniciada por el usuario. Por ejemplo, por medio de etiquetas, un diálogo, entre otros.
- **Consistencia.** Este principio sugiere que todas las características de la interfaz debe ser consistentes en la APP. Un ejemplo muy claro podría ser el uso de mensajes en la interfaz. Existen varios tipos de mensajes, que podrían ser informativo, advertencia o error y cada uno con un tipo de color que distinga el tipo de mensaje. La consistencia implica el uso de control de navegación, iconos, menús etc. Todo lo relacionado con la interfaz.
- **Autonomía y control del usuario sobre la aplicación móvil.**
- **Eficacia.** El diseño de la APP y su interfaz debe optimizar la eficiencia del trabajo del usuario. Los involucrados del proyecto deben entender la importancia que la meta número uno es la de construir un sistema

eficiente y dar poder a un usuario eficiente, hacer que la productividad del usuario sea la meta.

- Flexibilidad. Flexibilidad al usuario y que permitan reversibilidad sobre acciones realizadas.
- Ley de Fitts: “El tiempo para llegar a un objetivo está en función de la distancia que hay hasta él y del tamaño que tenga” (Pressman, 2010). Significa que a menor distancia y mayor tamaño el usuario tendrá mayor facilidad para interactuar.
- Objeto de interfaz de usuario. Uso de estándares y objetos familiares en la interfaz. Cualquier objeto de interfaz puede ser mejor aprovechado por otro usuario y puede obtenerse de librerías de interfaz de usuario no sería malo que sean utilizados.
- Reducción de latencia. Optimización del tiempo de espera del usuario y la espera deben explicarse de tal forma que, el usuario entienda lo que esta pasado.
- Aprendizaje. La interfaz debe diseñarse y reducir el tiempo de proceso de aprendizaje necesario por parte del usuario. Para que se cumpla este principio en el diseño, se debe hacer énfasis a un diseño sencillo, intuitivo que organice el contenido y funcionalidad en cualidades que resulten obvias para el usuario.
- Metáforas. Uso adecuado de metáforas facilita la comprensión del concepto global pero se debe asegurar de que la metáfora elegida sea conocida por los usuarios y para garantizarlos debe utilizarse imágenes y conceptos de la experiencia del usuario y que no debe ser una reproducción exacta del mundo real.

- Mantener la integridad de un producto de trabajo. Un producto de trabajo se refiere, por ejemplo, cuando un usuario esté llenando un formulario y a causa de un error se pierda toda la información ingresada, no es nada agradable si se pierde la información. La interfaz debe apoyar a la protección del trabajo del usuario es importante y se debe asegurar que su trabajo no se pierda a causa de un error y proveer de mecanismos la recuperar la información ingresada por el usuario.
- Legibilidad. La información que se le presente al usuario debe ser legible usando los recursos disponibles. Utilizados estilos, letras, tamaño y el color del fondo, todo debe contrastar.
- Seguimiento del estado y de las acciones del usuario. La interfaz debe proporcionar mecanismos para darle seguimiento al estado de las interacciones de usuario y guárdalo. Resulta atractivo que los usuarios pueden desconectarse en cualquier momento y recuperar su estado al momento de conectarse nuevamente.
- Navegación visible. Este principio sugiere que un buen diseño de la interfaz genera “la ilusión de que los usuarios están en el mismo sitio, pero con el trabajo traído hacia ellos” (Pressman, 2010).

4.16.5. Las mejores prácticas de usabilidad de Jakob Nielsen

A la hora de diseñar una interfaz de usuario no basta el uso de herramientas de diseño, también la creatividad en muchas ocasiones, un problema enorme por parte de los diseñadores a la hora de usar la creatividad es la de pensar en ellos mismos cuando se elabora el diseño de interfaz. Antes

de empezar a diseñar es recomendable revisar y analizar algunos conceptos y experiencias de autores conocidos. Es difícil cubrir con la mayoría de conceptos que existen y no garantiza el éxito, pero no cabe duda de que facilita el trabajo.

La usabilidad es un concepto bastante utilizado no solo para el diseño de interfaces. En este caso, la usabilidad es utilizada para medir la calidad de experiencia que tiene un usuario cuando interactúa con el sistema. Jakob Nielsen define la usabilidad como el atributo de calidad que mide la facilidad de usar una interfaz web.

Tabla I. Las 10 heurísticas de Jakob Nielsen

.	Visibilidad del estado del sistema	El sistema debe informar de lo que sucede o ocurre y proporcionar respuesta en un tiempo razonable al usuario.
.	Consistencia entre el sistema y el mundo real	Facilitar a los usuarios una comunicación haciendo uso de palabras, frases y conceptos que el usuario entienda o que familiarice. La información conviene aparecer en un orden lógico y natural, sin hacer uso de términos propios del sistema.
.	El usuario es libre y tiene el control	El sistema debe proveer alguna opción clara y fácil cuando los usuarios eligen opciones por error. El uso de mecanismos de desasear la acción, habilitar acceso a la ventana principal o "Home" da libertad y control a los usuarios en la aplicación.
.	Consistencia y estándares	El uso de convenios asegura y ayudan a los usuarios no estar adivinando las diferentes palabras, situaciones o acciones que significan lo mismo.
.	Prevención de errores	Es mejor que generar buenos mensajes de error, resulta más efectivo e importante prevenir la aparición de estos errores.
.	Mejor reconocer que memorizar	Todos los objetos, acciones y opciones útiles al usuario deben estar visibles o fácilmente localizables para que no tenga que recordar y verse obligado a usar la memoria.
.	Flexibilidad y eficiencia de uso	Los usuarios se convierten de novatos a expertos o también se da el caso de que los usuarios realizan tareas frecuentes, el uso de aceleradores o atajos puede ofrecer una interacción rápida para estos usuarios, el sistema puede proveerles todo estos tipos de beneficios. El sistema debe permitir que los usuarios configuren estas acciones frecuentes.
.	Diseño estético y minimalista	Diálogos no deben contener información que sea irrelevante o de rara utilidad. Cada unidad adicional de información en un diálogo compete con las unidades pertinentes de información y disminuye su visibilidad relativa
.	Ayuda al usuario a reconocer, diagnosticar y recuperarse de los errores	Los mensajes de error deben ser entendibles y sugerir una solución o una salida al problema, también tienen que escribirse en un lenguaje que el usuario entienda y sin tecnicismo.
0.	Ayuda y documentación	Es importante que el sistema sea intuitiva a la hora de utilizarse pero en muchas ocasiones el usuario necesita consultar y ayuda de la documentación. La ayuda debe ser fácil de encontrar y debe ser claro en los pasos necesarios y no extensos.

Fuente: elaboración propia.

4.16.6. Los dispositivos móviles

“La tecnología móvil ha evolucionado desde la voz a la información inalámbrica” (Yoshihiko, 2015). Desde enormes terminales móviles a teléfonos inteligentes. En la actualidad se tiene la posibilidad de comunicarse con cualquier persona, en cualquier momento y desde casi cualquier lugar. Gracias a estos avances la computación tiende a ser ubicua (Mitleton-Kelly, 2013).

El trabajar con tecnología móvil implica varios conceptos y tecnologías relacionados. Para iniciar con el diseño es necesario conocer sobre qué tecnología está destinado este diseño porque no se abarcará para todas las tecnologías. El sistema operativo que se ha seleccionado es el sistema operativo Android y aplicaciones móviles nativas.

4.16.7. Definición de APP

También conocido como aplicación móvil. “Una aplicación móvil consiste en un software que funciona en un dispositivo móvil (teléfonos y tabletas) y ejecuta ciertas tareas para el usuario” (Association, 2011). En los últimos años se ha experimentado un mayor crecimiento al uso de estas aplicaciones, esto se debe a las enormes ventajas que ofrece los dispositivos móviles y cada vez adquiere más importancia para los nuevos teléfonos inteligentes.

4.16.8. Aplicaciones móviles *web*

Son aplicaciones móviles desarrolladas con tecnología web, pero optimizadas para usarse en dispositivo móvil o tabletas. Estas aplicaciones son accedidas utilizando el navegador que viene por defecto dentro de los dispositivos para en este caso para Android.

4.16.9. Aplicaciones móviles nativas

Son las aplicaciones informáticas que se desarrollan para un específico sistema operativo, para poderlo utilizar el usuario instala la aplicación en el dispositivo. Esto significa que para cada sistema operativo deben desarrollarse y instalarse la aplicación. Como ventaja considerable aprovecha muy bien todos los recursos del dispositivos en donde es instalado, ventajas como mejor aprovechamiento de la memoria y hardware del dispositivo. El alto costo de desarrollo aplicaciones para cada dispositivo móvil lo hace una opción no muy atractiva.

4.16.10. Aplicaciones móviles híbridas.

Es la combinación de lo web y lo nativo. En el desarrollo de estas aplicaciones toma lo mejor lo mejor de las dos tecnologías anteriores. Las APPs híbridas se desarrollan con las siguientes tecnologías HTML, Javascript y CSS esto permite su uso en diferentes plataformas. Pero dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo.

4.16.11. Sistema operativo android.

“Android es un sistema operativo móvil que se basa en una versión modificada de Linux. Originalmente fue desarrollado por Android, Inc. En 2005. Como parte de su estrategia para entrar en el espacio móvil, Google compró Android y se hizo cargo de su trabajo de desarrollo así como su equipo de desarrollo”(Lee, 2012).

Android es un sistema operativo y los sistemas operativos administran todos los recursos del dispositivo que utiliza que podrían ser teléfonos inteligentes, tabletas etc. Recursos como software y hardware de cada dispositivo. Si los dispositivos no contaran con un sistema operativo el uso de los dispositivos sería muy difícil.

5. METODOLÓGICO

El proyecto se realizó en seis fases, en las cuales se generó la siguiente documentación para evidenciar el cumplimiento de los objetivos establecidos:

- Documento de requerimientos del sistema
- Documento del diseño de arquitectura de software
- Diseño de caso de uso
- Diseño de esquema de datos
- Diseño de interfaz de usuario

5.1. Descripción del problema y elaboración del documento de requerimiento

El punto de partida de cualquier proyecto es la definición del problema y la investigación que se realice con respecto a esta problemática. Este paso es fundamental porque es una declaración clara y concisa que describe los síntomas del problema que se desea analizar y resolver. Definir correctamente el problema proporciona grandes beneficios por qué significa, en ocasiones, más que de la mitad de la solución.

En esta fase, se incluyó la investigación de las distintas herramientas de software que apoyaron al diseño del sistema y la elaboración del documento de requerimientos del sistema. Los requerimientos específicos incluyen los requerimientos funcionales y los no funcionales del sistema del software educativo.

5.2. Elaboración del diseño de arquitectura de software

Luego de terminada la fase anterior que viene siendo el proceso análisis y la toma de requerimiento se inició con el proceso de diseño de arquitectura de software, el cual implicó concebir la arquitectura de sistema que soportara los requisitos del proceso educativo en línea.

Se revisó la literatura disponible sobre arquitectura de sistemas educativos en línea para determinar los componentes tecnológicos necesarios para el sistema de consulta de contenido KIDSPEDIA.

Finalmente, se diseñó la arquitectura basada en módulos interoperables, a través de la definición de protocolos de interacción y comunicación. Para la elaboración de los prototipos de la interfaz gráfica se utilizó la herramienta Microsoft Visio.

5.3. Diseño de caso de uso

Dentro de la fase de análisis y requerimiento que corresponde la fase uno, se identificaron los casos de uso y sus actores. Los requerimientos del sistema se expresan a través de casos de uso. Un caso de uso describe el comportamiento del sistema desde la perspectiva externa. Esta fase es fundamental y en todo proceso de desarrollo de software porque en él se generaron modelos del sistema que se incluyeron en la documentación y guiaran en el proceso de desarrollo de software.

En esta fase se elaboraron las descripciones de caso de uso y los diagramas de caso de uso. Los casos de uso se realizaron con base a los requerimientos del sistema de software obtenidos en la fase uno. Los casos de uso que se define en esta fase determinan lo que hace el sistema y definen las restricciones de operación o implantación. Para la elaboración de los diagramas de casos de uso se utilizó la herramienta Microsoft Viso. A continuación lo siguientes son casos de usos que fueron identificados.

- Caso de uso autenticarse al sistema.
- Caso de uso consultar programa.
- Caso de uso consultar contenido.
- Caso de uso responder preguntas.
- Caso de uso consultar evaluación.

5.4. Diseño de esquema de datos

Se elaboró con éxito un buen diseño de base de datos, pero fue necesario hacer un detallado análisis de los requerimientos del sistema y todos los requerimientos eso se obtuvieron de la fase uno para comprender lo que necesitaba el sistema para la consulta de contenido KIDSPEDIA. Para realizar el análisis fue necesario recabar toda la información posible sobre el sistema para la consulta de contenido KIDSPEDIA, para luego iniciar con el análisis detenidamente, desde distintos puntos de vista con la finalidad de lograr diseñar un modelo que la represente de manera abstracta lo más exactamente posible.

Se efectuaron preguntas y premisas con el propósito de realizar el análisis del sistema.

- ¿Cuáles son las entidades a procesar por el sistema?

- ¿Cómo se componen cada uno de estas entidades y qué atributos los describen?
- ¿Cuáles son las relaciones entre las entidades?
- ¿Qué procesos se ejecutan en el sistema para la consulta de contenido KIDSPEDIA?
- ¿Cuándo se ejecutan?
- ¿Quién los ejecutan?
- ¿Qué intercambio de información existe dentro de los componentes del sistema y con el exterior?
- Conocer los límites del sistema.
- ¿Existen excepciones a tener en cuenta para la realización de los procesos?
- ¿Cuáles son las excepciones que se debe tomar en cuenta para la elaboración de los procesos?
- ¿Cómo se almacena la información?
- ¿Qué datos se almacenarán?
- ¿Quién almacenaran los datos?

La herramienta a utilizar fue Microsoft Visio, el cual es una herramienta que brinda diversas plantillas para diseño de base de datos.

5.5. Diseño de una interfaz amigable

Diseño de una interfaz amigable comenzó con la identificación de usuarios, tareas y requisitos del entorno que corresponde con el análisis de requerimiento. Identificada las tareas que pueden realizar los usuarios en la fase de diseño de casos de uso, se analizó cada uno de los escenarios y se definieron un conjunto de objetos y acciones de interfaz. A partir del análisis

anterior, se extrajo información que se utilizó para la elaboración del diseño preliminar de interfaz denominados prototipos de interfaz.

En la elaboración de los prototipos se colocaron iconos, objetos gráficos, texto, títulos de ventanas y la especificación de los elementos de los menús. Para ello, se utilizaron herramientas de prototipo de interfaces gráficas, las herramientas que se utilizaron y apoyaron en la elaboración de interfaz son las siguientes: BALSAMIQ MOCKUPS, MOCKINGBIRD, MOCKFLOW y Microsoft VISIO.

A continuación se detallan las tareas que se siguieron en la etapa del diseño de interfaces de usuario:

- Investigación sobre interfaz de usuario y herramientas

El punto de partida para esta fase fue la investigación sobre conceptos críticos sobre interfaz de usuario, estándares, buenas prácticas y puntos de vistas de personas reconocidas con el tema. También se investigaron herramientas para elaborar las distintas figuras utilizadas en la fase de diseño de interfaz.

- Análisis del usuario

La filosofía de diseño que se siguió se relaciona con un conjunto de metodologías y técnicas que tienen como objetivo común el usuario. Comprender sus necesidades, limitaciones, comportamiento. Todas las características del usuario fueron analizadas en esta etapa.

Lo crítico de esta etapa de diseño fue el analizar las actividades del usuario que son soportados por el sistema. Para entender lo que el usuario quiere hacer con el sistema se utiliza las siguientes técnicas análisis de tarea y de observación.

- Elaboración de pantallas de interfaz del sistema

Se elaboró para cada acción obtenida en el análisis de caso de uso una serie de imágenes de la interfaz que indican como ésta responde a las interacciones del usuario.

- Elaboración de prototipos del sistema

La necesidad de realizar prototipos fue grande debido a que se necesitó probar algunas partes del sistema, verificar algunas funcionalidades, conocer más sobre la interfaz de sistema, crear y validar mapas de navegación, probar nuevas técnicas entre otros.

Conforme se contaban con algunos prototipos se realizaron comprobaciones y se permitió la participar del usuario en el desarrollo del mismo. Cabe destacar que los prototipos son más que demostraciones del producto, se utilizaron para conocer las impresiones del usuario que servirán como retroalimentación en el diseño de la interfaz.

- Evaluación de la interfaz

En esta etapa se evaluó la utilización de una interfaz y se verificó si cumple con los requerimientos del sistema.

Los prototipos de interfaz de usuario que fueron diseñadas en esta fase fueron las siguientes:

- Pantalla de inicio
- Pantalla login
- Pantalla lista de contenido
- Pantalla para contenido de matemáticas
- Pantalla lista de resultados

Los objetos de interfaces que fueron diseñadas en esta fase fueron las siguientes:

- Cuadro de diálogo y mensajes.
- Diseño del icono del sistema.
- Diseño de iconos del sistema.
- Diseño de diagrama de navegación.
- Diseño de cuadro de diálogos y mensajes.
- Pantalla de bienvenida.
- Pantalla login.
- Pantalla lista de contenido.
- Pantalla contenido de curso de matemática.
- Pantalla ejercicio de matemática.
- Pantalla del curso de Lectura.
- Pantalla progreso del curso de matemáticas.
- Pantalla de error.
- Pantalla configuración de cuenta.
- Pantalla configuración de contraseña.

5.6. Revisión y finalización del documento de arquitectura

Entrega final, revisión, actualización del documento de arquitectura del sistema y de todos los siguientes artefactos:

- Especificación de requerimientos
- Diagrama de casos de uso
- Especificación de casos de uso
- Diagrama de arquitectura de datos
- Diseño de interfaces de usuario

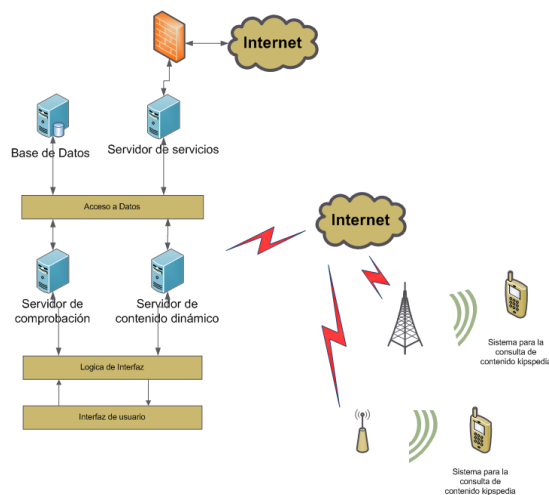
6. PRESENTACIÓN DE RESULTADOS

6.1. Selección, descripción y diseño de la arquitectura del sistema para la consulta de contenido KIDSPEDIA

En esta sección se describe el diseño de la arquitectura de software del sistema para la consulta de contenido en KISPEDIA.

El diseño de la arquitectura parte después del análisis de 5 arquitecturas. En la Figura 7, se puede visualizar un diseño de alto nivel que describe los componentes principales del sistema y cómo interactúan entre sí, para lograr los objetivos del diseño mencionados.

Figura 7. **Arquitectura del sistema para la consulta de contenido Kidspedia**



Fuente: elaboración propia.

6.2. Funcionalidades y componentes

6.2.1. Servidor de base de datos

Este servidor está formado por un sistema de gestión de base de datos SGBD. Un SGBD es una herramienta de software para el manejo eficiente de los datos. Permite consultas complejas sobre los datos utilizando lenguaje SQL y el acceso concurrente de muchos usuarios autorizados a los datos.

Se utilizará un servidor de base de datos para almacenar, eliminar y gestionar: la información del usuario, información de cursos, contenido, texto, imágenes, audio etc. La importancia del servidor de datos es enorme porque en ella se almacena toda la información del sistema y muchos procesos, servidores acceden a él para solicitarle datos.

El servidor de contenido dinámico tiene la tarea de crear contenido para el usuario, para ello, busca la información necesaria en la base de datos del sistema. Cuando se encuentra, se crea el contenido y es entregado al usuario. Cuando no se encuentra, se utiliza un servicio que se localiza en el servidor de servicios que es el encargado de buscar la información en otros sistemas por medio del internet, cuando se localiza es devuelto la información al servidor de contenido dinámico este último prepara el contenido y por último se lo entrega al usuario.

6.2.2. Servidor de servicios

Este servidor está definido por un conjunto de aplicaciones llamados servicios, utilizados para proveerle dinamismo al sistema.

Los servicios son tecnología que hace uso de protocolos y estándares como XML. La base principal de esta tecnología es el intercambio de mensajes. Los servicios son programados y utilizados para el intercambio de datos entre distintas aplicaciones sin importar como es su implementación. Funcionan como interfaces donde los usuarios realizan solicitudes, se localizan y finalmente utilizados desde una aplicación a través de la *web*.

Su función principal, se utiliza para obtener información no encontrada en la base datos del sistema de consulta de contenido. La información es solicitada por el usuario, se busca primero en la base de datos local, al no ser encontrada la información solicitada por el usuario, se hará uso de un servicio web para buscar la información en otros sistemas. El servicio se comunica, interactúa con otros sistemas, cuando la información es encontrada, se obtiene la información del otro sistema, se presenta la información al usuario y se almacena en la base de datos local.

6.2.3. Servidor de contenido dinámico

Este servidor contiene el proceso, la lógica para la creación de contenido y programas de aprendizaje dinámico.

Este percibe el entorno para proporcionar contenido personalizado al usuario. Lo hará por medio de la información preliminar del perfil de cada usuario. Luego de que el usuario consulte el contenido se le hará una evaluación con respecto al contenido, esto servirá para mostrarle también contenido dinámico. Las actividades de este servidor principalmente son las siguientes:

- Crear programas de aprendizaje.

- Proporcionar a cada usuario contenido de acuerdo a su perfil y avances obtenidos.

6.2.4. Servidor de comprobación

Este servidor contiene el proceso, la lógica para la creación de las evaluaciones, las estadísticas y resultados del aprendizaje de cada usuario después de que el usuario haya interactuado con el contenido.

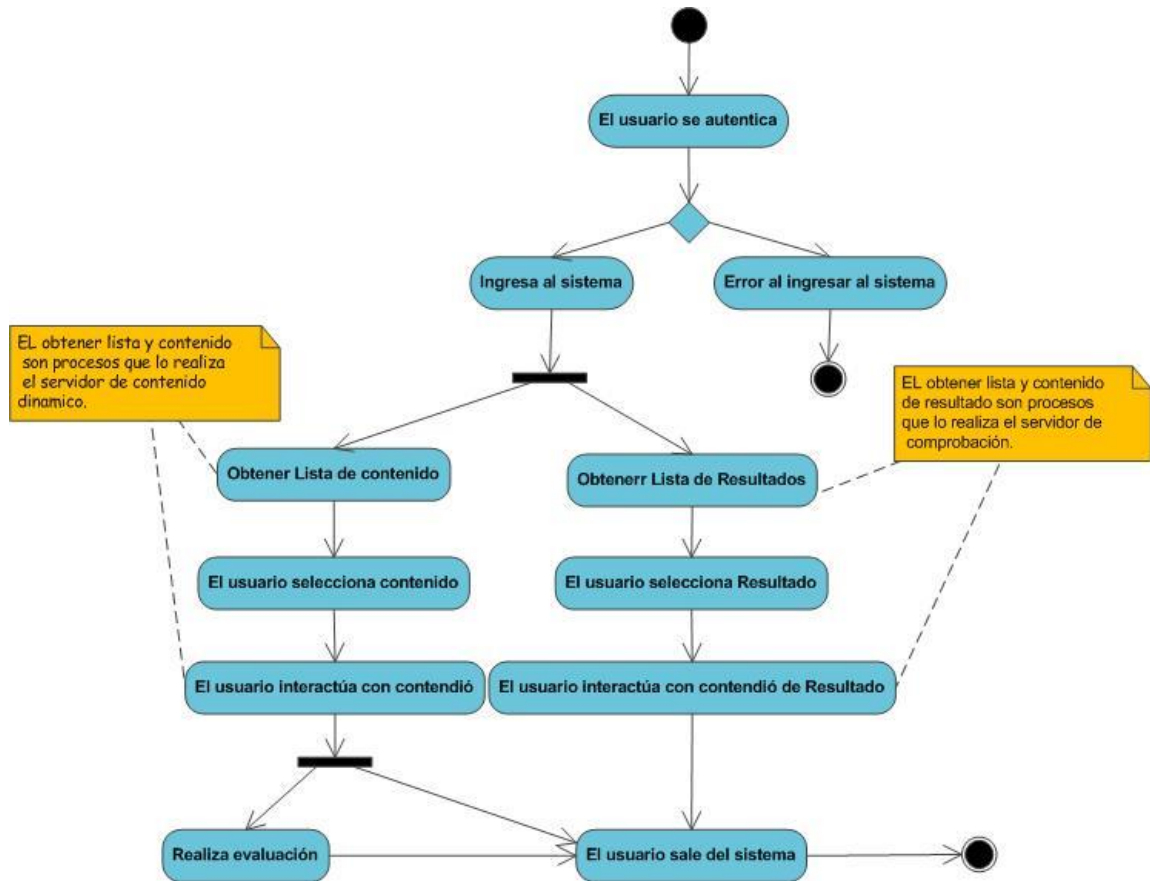
6.2.5. Funcionalidad

La función principal de este servidor proporcionar al usuario evaluaciones para medir la comprensión obtenida de cada material que se le ha dado por medio del servidor de contenido dinámico. Servirá para medir y conocer el logro, el avance de cada usuario y proporcionarle material dinámico según el desempeño de sus evaluaciones.

6.2.6. Diagrama de actividad, consulta de contenido

La figura 8 muestra el diagrama de actividades, en él se detalla el comportamiento dinámico del sistema por medio de una serie de acciones que realiza el usuario.

Figura 8. Diagrama de actividad



Fuente: elaboración propia.

El sistema para la consulta de contenido, es la herramienta para visualizar el contenido interactivamente, a través de esta herramienta el usuario final, podrá: interactuar, revisar, leer, consultar y aprender haciendo uso del contenido, podrán aprender matemáticas, por medio del módulo de matemáticas.

Para consultar el contenido el usuario hace uso de una interfaz. La interfaz es la herramienta utilizada para la comunicación usuario y sistema de consulta

de contenido. Primero el usuario debe autenticarse si el usuario no logra autenticarse no ingresa al sistema y como resultado una pantalla de error.

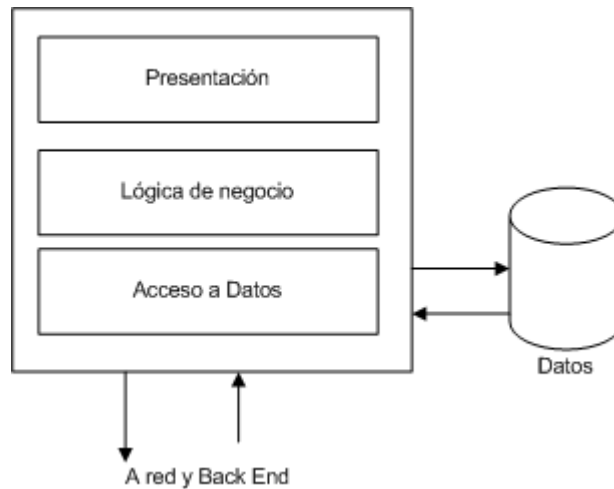
Cuando el usuario ingresa satisfactoriamente. El servidor de contenido dinámico crea los programas de aprendizaje dinámico para cada usuario y es entregado a cada usuario. A la vez, el servidor de comprobación entrega el listado de desempeño a cada usuario.

El contenido es dinámico según el perfil que presenta cada usuario y los avances que cada usuario ha tenido en el sistema y es detectado en las evaluaciones. El contenido dinámico se procesa en el servidor de contenido dinámico, este servidor realiza la búsqueda de la información en la base de datos del sistema, cuando se encuentra la información se procesa, al final es entregado al usuario que realizó la petición. Cuando no se localiza la información en base de datos, el sistema utiliza un servicio para buscar la información solicitada a otros sistemas, cuando se localiza la información se almacena en base de datos y es entregada la información.

6.2.7. Arquitectura para dispositivos móviles

Esta parte de la arquitectura es muy importante, es utilizado por el usuario para interactuar con el sistema de consulta de contenido. Por medio de la interfaz de usuario, el usuario opera, controla, envía una retroalimentación y realizar tareas en el sistema.

Figura 9. **Arquitectura para dispositivos móviles**



Fuente: Mobile Application Architecture Guide Figura 1.

La figura 9 representa la arquitectura que permite a las aplicaciones móviles almacenar en una base de datos local la información que se encuentra o se obtiene de un servidor y se accede a esta, a través de servicios API, los cuales comunican a la base con la interfaz de la aplicación.

6.2.8. Interfaz gráfica de usuario

Es el medio por el cual el usuario y sistema se comunican, por lo tanto, el diseño de interfaz es importante, de ello depende su aceptación del sistema.

El usuario accederá al sistema a través de dispositivos móviles. Los dispositivos móviles son actualmente muy conocidos y gran parte de la población cuenta con uno de ellos, debido a su bajo costo de producción y la gran demanda que tienen. Los dispositivos móviles cuentan con una variedad de características como lo son: marca, tamaño, color, diseño. Pero sus

características principales actualmente se asemejan a una computadora de diferentes tamaños con capacidad de procesamiento, memoria, almacenamiento de datos y conexión a internet.

Los usuarios accederán al sistema a través de dispositivos móviles usando una aplicación para dispositivos móviles. Esta aplicación para dispositivo móviles permitirá a que cualquier usuario que tenga teléfono inteligente, tablets y otros dispositivos móviles el acceso el sistema.

Aplicación para dispositivos móviles, este contiene toda la lógica para poder acceder al sistema. Los usuarios lo utilizan para capturar y enviar información al sistema. Es un software que se compone de pantallas, pantallas de espera, pantalla de error, menú, botones, iconos, imágenes y objetos gráficos. Todo lo anterior le proporciona un entorno visual para permitir el intercambio de datos usuario y sistema.

6.2.9. Lógica de negocios

En esta parte de esta arquitectura se recibe las peticiones del usuario y se procesa. En esta etapa se establecen todas las reglas que deben cumplirse. Reglas como procesar datos, eliminar, almacenar datos a través de la capa de datos, presentar información mediante la capa de presentación o interfaz gráfica de usuario.

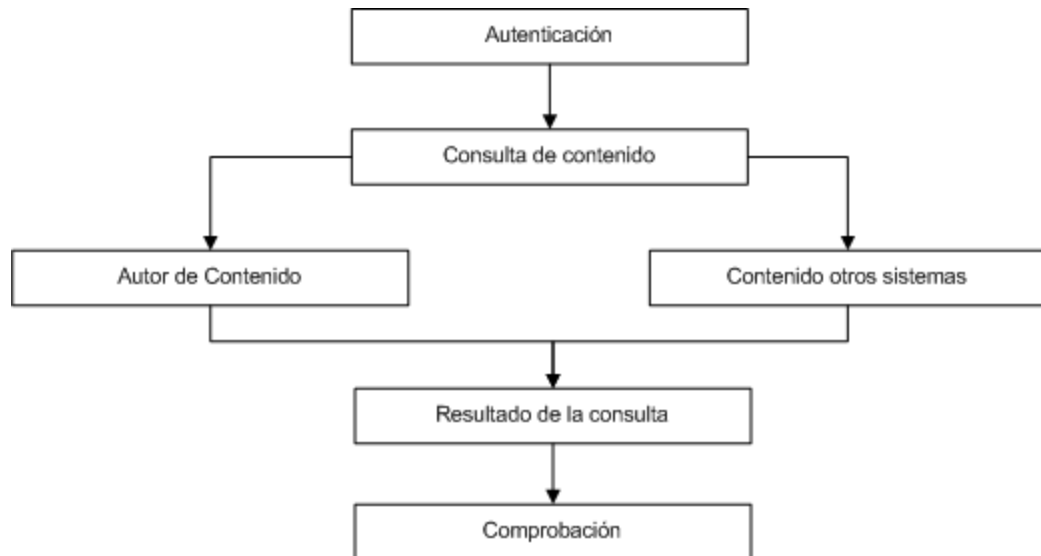
6.2.10. Base de datos

Se debe tomar en cuenta que en el dispositivo móvil casi siempre se encuentra solo una parte de los datos o un subconjunto de los datos esto se

debe a las limitaciones de memoria. La aplicación utilizará un gestor de base de datos porque esta herramienta ofrece enormes ventajas sobre los datos.

Un gestor de bases de datos para móviles como SQLite, por sus características se utiliza en muchas aplicaciones conocidas como, como Skype, Mozilla Firefox, entre otros.

Figura 10. **Diagrama de bloque del sistema**



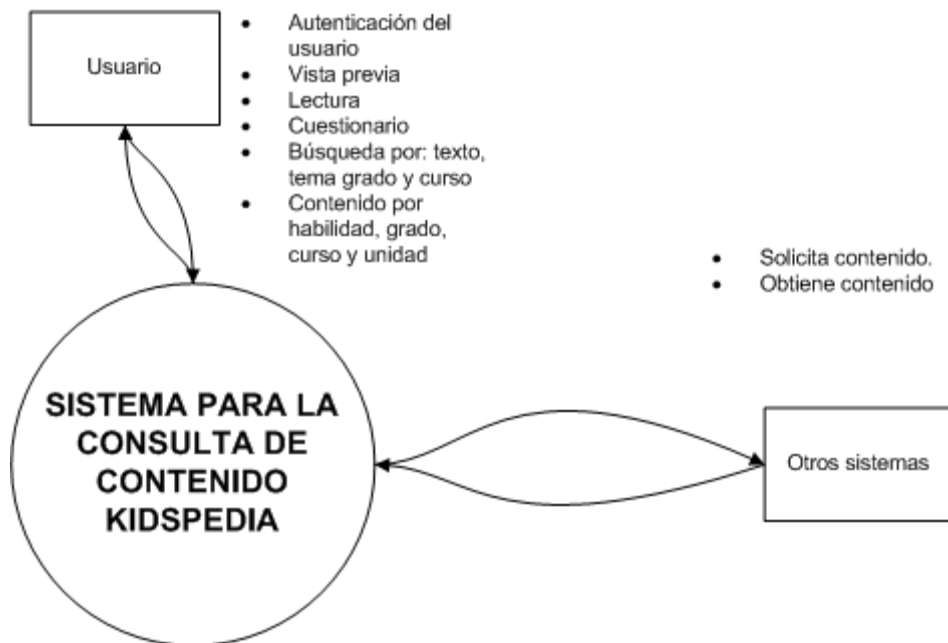
Fuente: elaboración propia.

La figura 10 es un diagrama de bloque que presenta la arquitectura de KIDSPEDIA, El sistema para la consulta de contenido KIDSPEDIA.

Este diagrama representa el funcionamiento interno del sistema, mediante bloques y sus relaciones. Además definen la organización de todo el proceso interno, sus entradas y sus salidas.

El usuario ingresa al sistema y lo primero que hace es autenticarse, luego desea hacer la revisión del su contenido, el contenido es buscado en el sistema local, cuando no se encuentra se busca en otros sistemas ajeno al sistema de consulta de contenido. El usuario interactúa con el contenido, al final de interactuar con el contenido, el sistema provee de una comprobación este resultado queda registrado en el servidor de base de datos.

Figura 11. **Diagrama de contexto del sistema KIDSPEDIA**



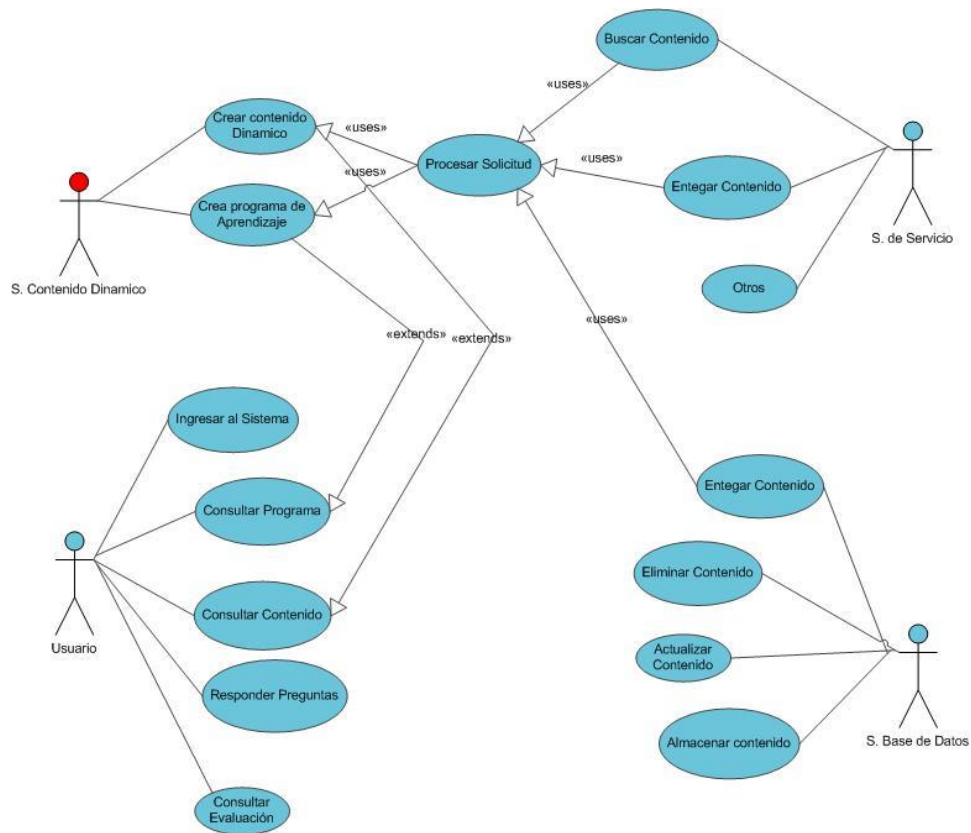
Fuente: elaboración propia.

En la figura 11 de diagrama de contexto del sistema, representa los límites y también los sistemas que interactúan con el Sistema para la Consulta de Contenido KIDSPEDIA. En la figura 11, el círculo representa el sistema de consulta KIDSPEDIA y los dos cuadros las entidades que interactúan con el sistema.

6.2.11. Diseño de casos de uso del sistema para la consulta de contenido KIDSPEDIA

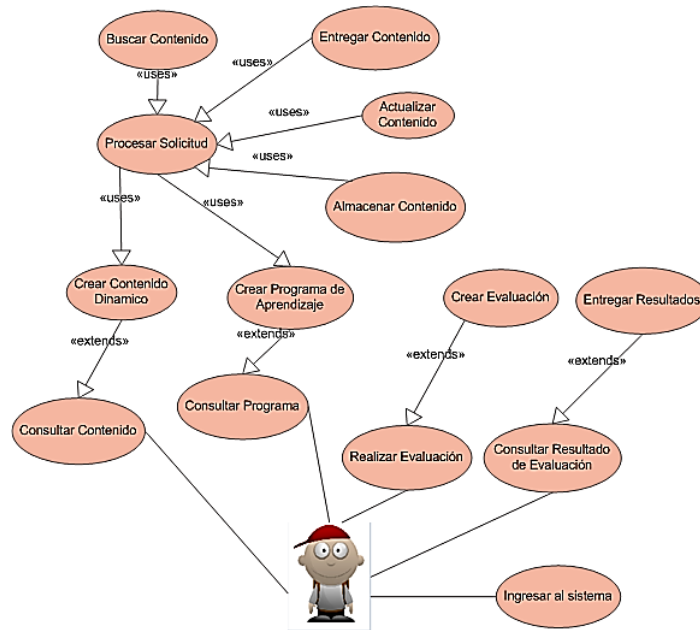
En esta sección, diseño de casos de uso y el diagrama de caso de uso, son dos cosas diferentes. Por medio del diagrama se hará un bosquejo de lo que el actor realiza en el sistema, y el sistema hace sin resaltar los elementos internos del sistema. Pero profundizando más en esta sección también se definirán los casos de uso, que es la descripción a detalle, de la interacción sistema y sus actores.

Figura 12. Diagrama de caso de uso



Fuente: elaboración propia.

Figura 13. Diagrama de caso de uso



Fuente: elaboración propia.

6.2.12. Descripción de caso de uso

Tabla II. Caso de uso autenticarse al sistema

Caso de uso: Autenticarse al sistema
Autor: Usuario
Descripción: Permite al usuario valido ingresar al sistema colocando sus credenciales.
Precondición: El usuario debe colocar nombre de usuario y contraseña válidos.
Flujo Normal: El actor ingresa al sistema y ve dos cuadros blancos uno para el nombre de usuario, el otro para colocar la contraseña. El sistema solicita que ingrese el nombre de usuario. El usuario ingresa el nombre de usuario. El sistema solicita que ingrese la contraseña de usuario El usuario ingresa contraseña. El usuario presiona aceptar.
Flujo alternativo: El sistema hace una validación de la información ingresada por el actor. Si son válidos los datos ingresados, el usuario ingresa al sistema.
Post condiciones: El usuario ingresa al sistema.

Fuente: elaboración propia.

Tabla III. **Caso de uso consultar programa**

Caso de uso: Consultar programa
Autor: Usuario
Descripción: Permite al usuario consultar programas sobre contenido de curso.
Precondición: 1. El usuario tiene que estar autenticado
Flujo Normal: 1. El usuario ingresa al área de programas. 2. Presiona clic sobre un programa 3. El servidor de contenido dinámico, recibe la solicitud. 4. El servidor de contenido busca la información 5. El servidor de contenido dinámico, crea el programa al usuario. 6. El servidor de contenido dinámico, entrega el programa. 7. El usuario ve el programa
Flujo alternativo: 1. El servidor de contenido no encuentra la información. 2. El servidor de contenido no devuelve información.
Post condiciones: 1. La acción se registra en la base de datos

Fuente: elaboración propia.

Tabla IV. **Caso de uso consultar contenido**

Caso de uso: Consultar contenido
Autor: Usuario
Descripción: Permite consultar el contenido
Precondición: 1. El usuario ingresa al sistema. 2. El usuario tiene que estar autenticado en el sistema
Flujo Normal: 1. El usuario ingresa al área de contenido dinámico. 2. El usuario presiona clic para hacer la consulta de contenido 3. El servidor de contenido dinámico, recibe la solicitud. 4. Se busca la información. 5. El servidor de contenido dinámico, crea el contenido para el usuario. 6. El servidor de contenido dinámico entrega el contenido dinámico
Flujo alternativo: 1. El servidor de contenido no encuentra la información. 2. El servidor de contenido no devuelve información.
Post condiciones: 1. La acción se registra en la base de datos.

Fuente: elaboración propia.

Tabla V. **Caso de uso responder preguntas**

Caso de uso: Responder preguntas
Autor: Usuario
Descripción: Permite responder preguntas creadas por el servidor de comprobación
Precondición: 1. El usuario visualiza el contenido dinámico.
Flujo Normal 1. El usuario termina de visualizar el contenido dinámico. 2. El servidor de comprobación, recibe la solicitud. 3. El servidor de comprobación, crea las preguntas 4. El servidor de comprobación, entrega las preguntas 5. El usuario presiona clic para empezar a responder preguntas. 6. El usuario responde preguntas. 7. El usuario termina de responder preguntas
Flujo alternativo: 1. El servidor de comprobación, no encuentra información 2. El servidor de comprobación, no crea las preguntas 3. El servidor de comprobación, no entrega las preguntas.
Pos Condiciones: 1. El resultado de la comprobación se almacena en base de datos.

Fuente: elaboración propia.

Tabla VI. **Caso de uso consultar evaluación**

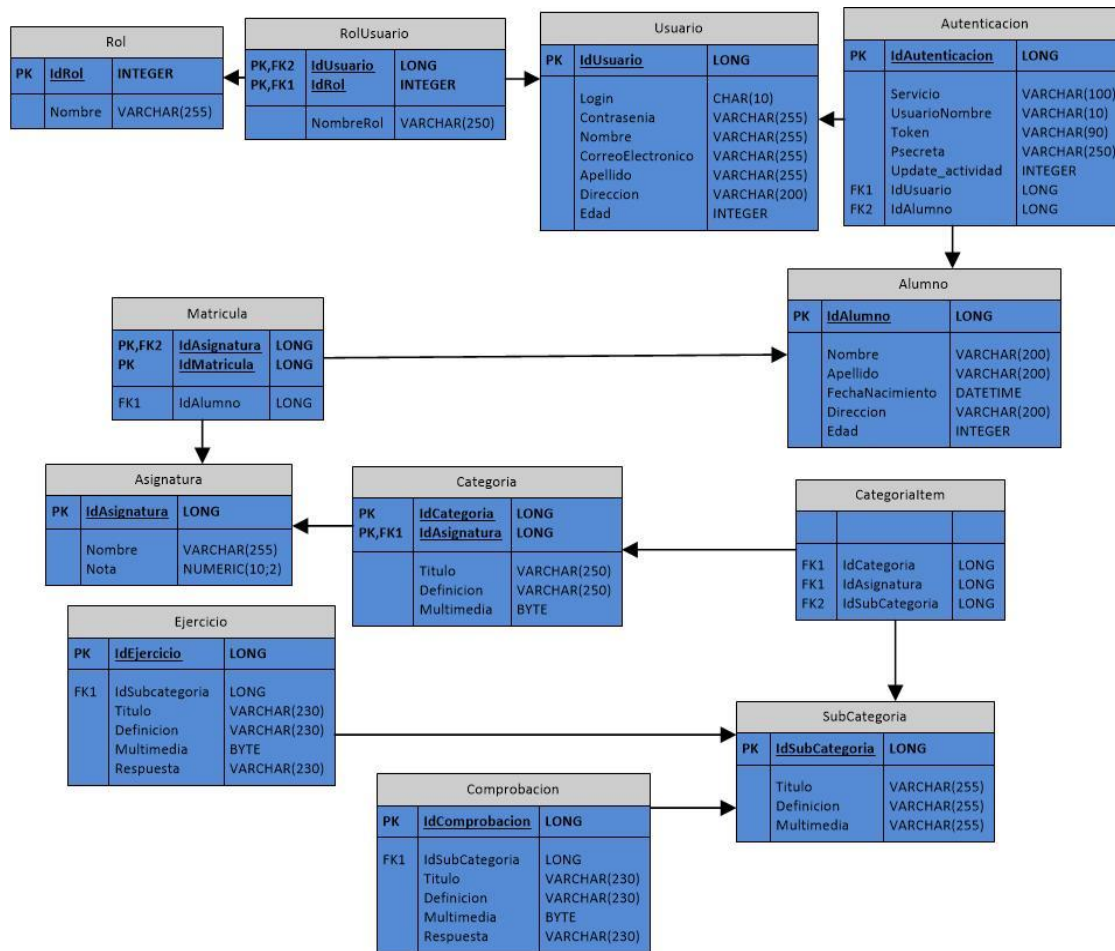
Caso de uso: Consultar evaluación
Autor: Usuario
Descripción: El usuario visualiza sus evaluaciones.
Precondición: El usuario ingresa al área de contenido dinámico.
Flujo Normal: El usuario se encuentra en el área de contenido dinámico. El usuario presiona clic en el botón de resultados de evaluaciones. El servidor de comprobación, recibe la solicitud. El servidor de comprobación, prepara resultados El servidor de comprobación, entrega los resultados. El usuario ve resultados y estadísticas.
Flujo alternativo: El servidor de comprobación, prepara resultados. El servidor de comprobación, entrega los resultados.
Post Condiciones: La acción queda registrada en base de datos.

Fuente: elaboración propia.

6.2.13. Diseño de esquema de base de datos del sistema para la consulta de contenido KIDSPEDIA

A continuación, el diseño de base de datos que describe como se representan los datos en el sistema. Todos los datos se manejarán en el siguiente esquema que está representado en la figura 14. Diagrama entidad relación de la base de datos, en ella se observa la abstracción de los datos en tablas con entidades y atributos y su relaciones que existen entre ello. Este esquema le provee al sistema, el manejo eficiente del contenido esto ayuda a proporcionarle al usuario información correcta y en el menor tiempo posible.

Figura 14. Diagrama entidad relación



Fuente: elaboración propia.

A continuación, una breve descripción de las tablas importantes para el sistema, el esquema define todas las tablas, sus relaciones entre ellas y las características de sus columnas. En la figura 15 se observa al usuario y alumno. La tabla utilizada para almacenar información de todo usuario que no sea alumno se llama usuario en ella se almacenará información importante del usuario como lo es nombre, dirección, edad etc. La tabla alumno es utilizada

para almacenar toda la información del alumno, el usuario alumno es un tipo de usuario, pero se pretende separar este tipo de usuario con el resto de usuarios.

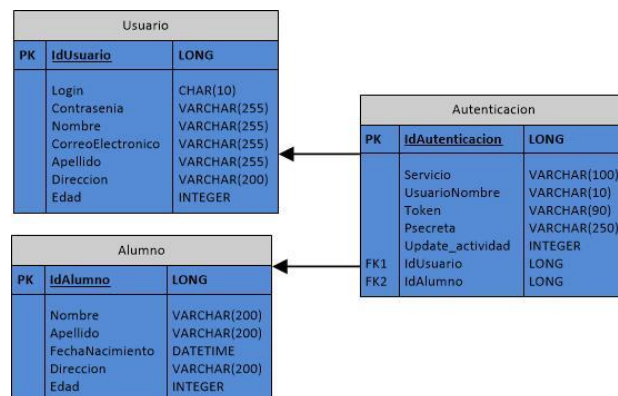
Figura 15. **Tablas usuario y alumno**



Fuente: elaboración propia.

La figura 16 se observa la tabla usuario y alumno se relaciona con la tabla autenticación. La tabla autenticación es utilizada para almacenar el estado que posee cada usuario en el sistema cuando se hace uso del sistema.

Figura 16. **Relaciones entre usuario, alumno y autenticación**



Fuente: elaboración propia.

Las tablas que se describen a continuación son especialmente para el manejo del contenido del sistema. En la figura 17 se observa la tabla asignatura, es utilizada para almacenar información de los cursos de cada alumno.

Figura 17. **Representación gráfica de la tabla asignatura y contenido de ejemplo**

Asignatura		
PK	<u>IdAsignatura</u>	LONG
	Nombre	VARCHAR(255)
	Nota	NUMERIC(10;2)

IdAsignatura	Nombre	Nota
0	Matematica	0
1	Inglés	0
3	Idioma Español	0

Fuente: elaboración propia.

En la figura 18 se observa la tabla categoría y se utiliza para el manejo de los temas que contendrá cada asignatura. Los campos título es utilizado para almacenar el título del tema, el campo definición es utilizado para almacenar la definición base y el campo multimedia es utilizado para almacenar material multimedia en el caso lo hubiera.

Figura 18. **Representación gráfica de la tabla categoría y contenido de ejemplo**

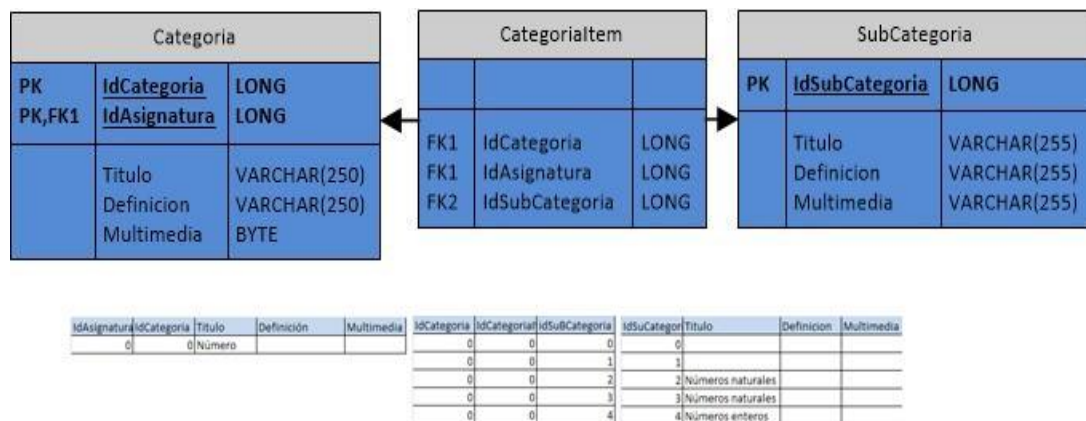
Categoría		
PK	<u>IdCategoria</u>	LONG
PK,FK1	<u>IdAsignatura</u>	LONG
	Título	VARCHAR(250)
	Definición	VARCHAR(250)
	Multimedia	BYTE

IdAsignatura	IdCategoria	Título	Definición
0	0	Número	
0	1	Conjunto	
0	2	Suma	
0	3	Resta	

Fuente: elaboración propia.

En la figura 19 se observan tres tablas una ya se mencionó con anterioridad, pero se relacionan, las siguientes con nombre Categoricaltem y Subcategoria son utilizadas para agregarle más definiciones en caso existiera al tema principal que se almacena en la tabla categoría. También es utilizado para almacenar sub temas al título principal si existiera. El campo denominado multimedia es utilizado para almacenar información multimedia relacionada.

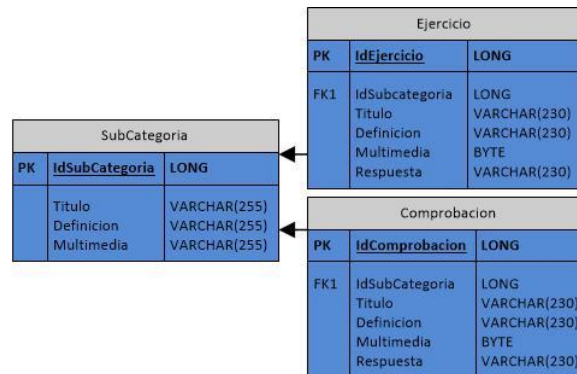
Figura 19. **Representación gráfica de las tablas categoría, categoricaltem, subcategoria y sus relaciones con contenido de ejemplo**



Fuente: elaboración propia.

En la figura 20, se observa dos nuevas tablas que tienen como nombre ejercicio y comprobación. La tabla ejercicio es utilizada para el manejo de los ejercicios de cada tema y subtema. Por último, la tabla comprobación y es en donde se manejará la información relacionada con el test o comprobación de cada tema.

Figura 20. **Ejercicio, comprobación y sus relaciones**



Fuente: elaboración propia.

6.2.14. Descripción y diseño de interfaz de usuario del sistema para la consulta de contenido KIDSPEDIA

En esta sección se inicia el diseño de interfaz de usuario, la filosofía a seguir es la de “El diseño centrado en el usuario”. Esta filosofía de diseño se relaciona con un conjunto de metodologías y técnicas que tienen como objetivo común el usuario. Comprender sus necesidades, limitaciones, comportamiento. Todas las características del usuario. Para lograr los objetivos establecidos se siguieron un conjunto de lineamientos, principios de interfaz. Básicamente tres reglas doradas (Pressman, 2010,p.266), las cuales son: dejar el control al usuario, reducir la carga de memoria del usuario y hacer que la interfaz sea consistente.

6.2.15. Propósito

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El

documento va dirigido al equipo de desarrollo y a la comunidad de posibles usuarios finales. Este documento será la vía de comunicación entre las partes implicadas, tomando parte en su confección miembros de cada parte. Esta especificación está sujeta a revisiones por las partes implicadas, especialmente por los potenciales usuarios, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

6.2.16. Ámbito de sistema

El sistema para la consulta de contenido KIDSPEDIA, es la herramienta para visualizar el contenido interactivamente, en esta herramienta el usuario final, los niños, podrán: interactuar, revisar, leer, consultar y aprender haciendo uso del contenido.

La idea de la solución, el proyecto se llama sistema para la consulta de contenido KIDSPEDIA, la idea es diseñar una enciclopedia con contenido especial para los niños, la tecnología es indispensable, porque a través de ella, su acceso, por medio de las diferentes tecnologías.

El ámbito del sistema desarrollado llega hasta la realización de las interfaces de usuario del sistema para la consulta de contenido KIDSPEDIA, dicho diseño de sistema debe ajustarse a los objetivos y alcances definidos

6.2.17. Definición de la tecnología a utilizar

El diseño para la consulta de contenido, su diseño se centra en los dispositivos móviles. Se especifica sobre que tecnología funciona el diseño, debido a que actualmente existen diversas tecnologías que el usuario puede

utilizar. Se ha seleccionado la tecnología móvil, debido a que el uso de dispositivos móviles se ha convertido en algo cotidiano y la evolución que han tenido en la comunicación en tan poco tiempo. Las tecnologías, todo relacionado con móviles han ido mejorando, actualmente ofrece al usuario una amplia gama de beneficios y prácticamente al alcance de los usuarios, debido a su bajo costo.

El trabajar con tecnología móvil implica varios conceptos y tecnologías relacionados. Para iniciar con el diseño es necesario conocer sobre que tecnología está destinado este diseño porque no se abarcara para todas las tecnologías. El sistema operativo que se ha seleccionado es el sistema operativo Android y aplicaciones móviles nativas.

6.2.18. Definición de los usuarios del sistema

En esta sección se identifica los principales usuarios implicados son los niños y niñas de siete a doce años, estos niños se encuentran en la primaria.

También debe considerarse lo siguiente sobre los usuarios y será la base para los principios de diseño, principios que se aplican a todos los diseños de interfaz de usuario.

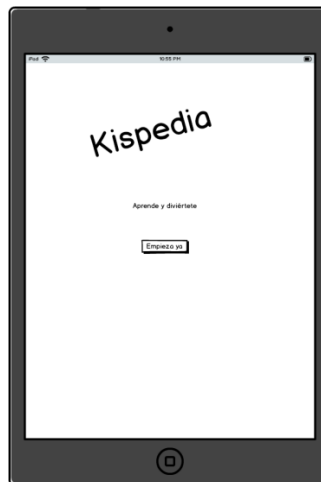
- Tienen memoria limitada a corto plazo
- Cometen errores, los usuarios poseen un rango de capacidades físicas
- Poseen diferentes preferencias de interacción

6.3. Diseño de prototipos de interfaz de usuario del sistema para la consulta de contenido KIDSPEDIA

6.3.1. Pantalla de inicio

La figura 21 es el prototipo de la pantalla de inicio de la aplicación y es la primera pantalla que se visualiza cuando se presiona clic sobre el icono para ingresar al sistema KIDSPEDIA. Esta pantalla da la bienvenida a todo usuario registrado o no registrado. Para ingresar y usar la aplicación, el usuario tiene que estar registrado.

Figura 21. Prototipo de la pantalla de inicio



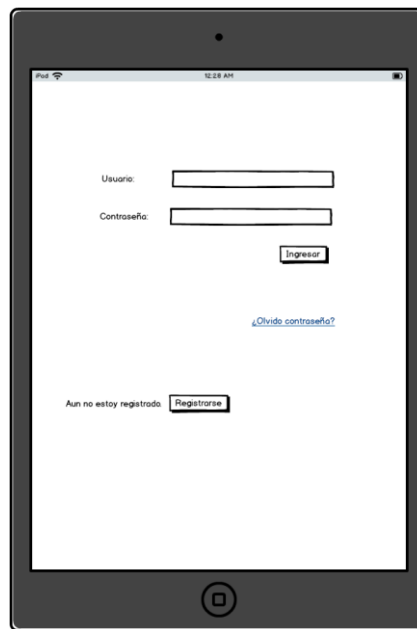
Fuente: elaboración propia, con la herramienta Balsamiqmockups.

6.3.2. Pantalla login

Prototipo de la pantalla login es utilizada para ingresar al sistema KIDSPEDIA, antes el usuario debe colocar su nombre de usuario y contraseña,

el sistema verifica los datos y si el usuario está registrado podrá ingresar y utilizar la aplicación. Los usuarios que no están registrados no podrán ingresar pero podrán registrarse y cuando lo hagan podrán aprender y disfrutar.

Figura 22. **Prototipo de la pantalla login**

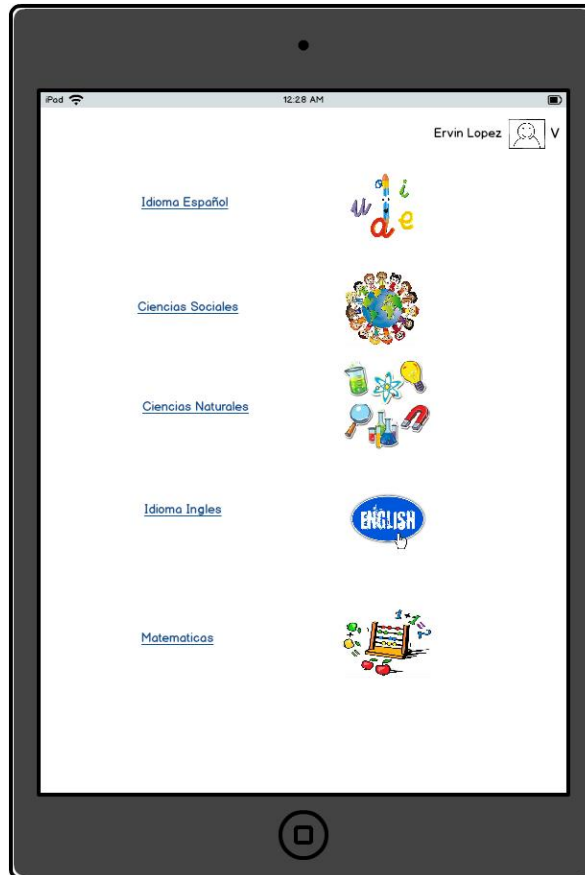


Fuente: elaboración propia, con la herramienta Balsamiqmockups.

6.3.3. **Pantalla lista de contenido**

Figura 23: Prototipo de la pantalla lista de contenido el cual muestra un listado dinámico de cursos disponible para cada usuario. El servidor de cursos dinámicos es el encargado de evaluar y proporcionar cursos. Para interactuar con el curso el usuario tiene que presionar sobre el link del curso.

Figura 23. **Prototipo de la pantalla lista de contenido**

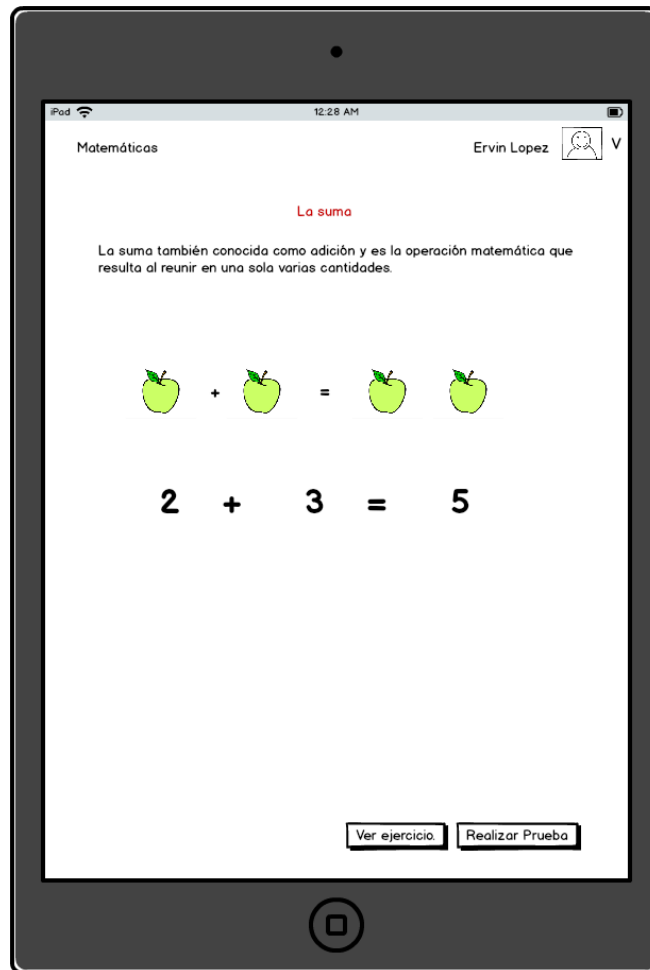


Fuente: elaboración propia, con la herramienta Balsamiqmockups,

6.3.4. **Pantalla para contenido de matemáticas**

Figura 24: Prototipo de la pantalla contenido de matemáticas el cual muestra de cómo se muestra el contenido de matemáticas. El mundo de las matemáticas es inmenso y es importante que los niños aprendan matemáticas y las lecciones no deben ser tan aburridas. Con la ayuda de KIDSPEDIA aprender matemáticas es divertido.

Figura 24. **Prototipo de la pantalla contenido de matemáticas**

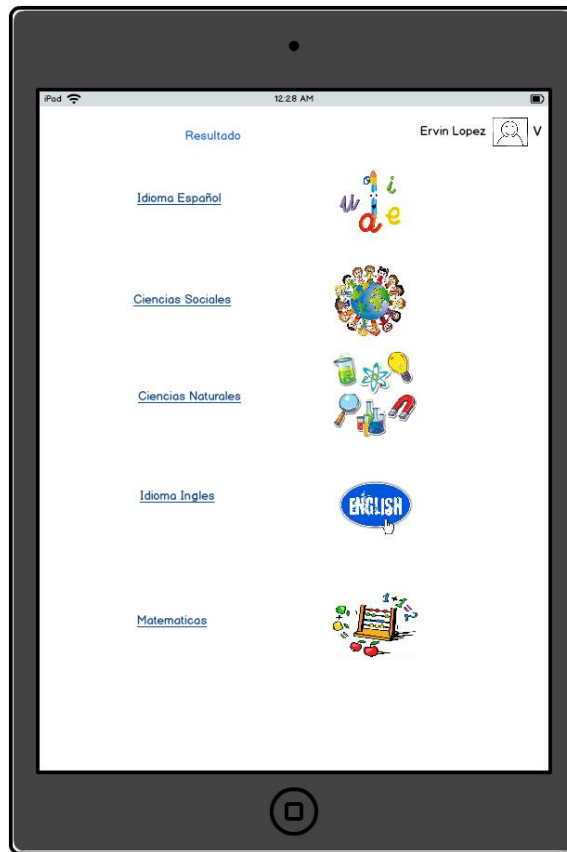


Fuente: elaboración propia, con la herramienta Balsamiqmockups.

6.3.5. **Pantalla lista de resultados**

Figura 25 Prototipo de la pantalla lista de resultados, esta pantalla muestra el listado de cursos de cada usuario, pero a la hora de presionar sobre cada cursos lo que vera, será el desempeño que ha tenido el usuario en sus cursos, útil porque se conoce el desempeño del usuario.

Figura 25. **Prototipo de la pantalla lista de resultados**



Fuente: elaboración propia, con la herramienta Balsamiqmockups.

6.3.6. Diseño de interfaz de la consulta de contenido KIDSPEDIA

El diseño de esta interfaz se basa en aplicaciones nativas y sistema operativo Android. Las interfaces nativas utilizan interfaces nativas. Los elementos que conforman una interfaz nativa son comúnmente botones, gráficos, íconos, fondos, menú, listas y encabezados. Estos tienen una apariencia visual diferente y vienen preestablecidos para cada sistema operativo.

Obviamente se debe cuidar cualquier detalle de la aplicación y no olvidar la parte funcional. Porque otra funcionalidad de la interfaz importante es la de servir como capa donde nacen todas las interacciones entre el usuario y lo funcional de la APP.

6.4. Diseño de pantallas de Interfaz del sistema

6.4.1. Pantallas de diseño

En esta sección se inicia la elaboración de las pantallas de diseño usando acabados más estéticos y siguiendo lineamientos de personas que tienen mucha experiencia en la materia. En la figura 26, se observa el primer diseño de fondo sobre el marco de trabajo. Este fondo fue tomado de un conjunto de objetos de interfaz gráfica.

Figura 26. Primer diseño sobre marco de trabajo



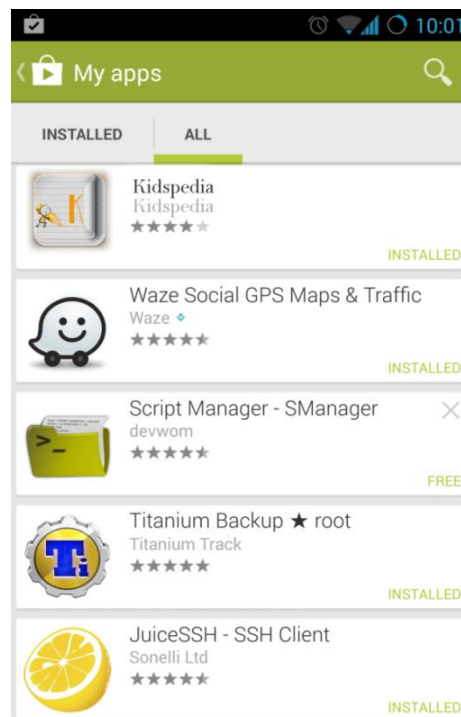
Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2 y Adobe Photoshop CS2.

6.4.2. El ícono

Un ícono mal diseñado, desagradable visualmente o que no se entiende lo que ofrece puede ser decisivo.

Figura 28. Muestra el resultado de búsqueda de aplicaciones en Google Play. El icono aporta información inicial al usuario. Google Play Store es un servicio para la distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android. Es importante destacar y presentar un diseño atractivo y se debe empezar por el ícono.

Figura 27. **Resultado de búsqueda y aplicaciones instaladas, Google Play**



Fuente: elaboración propia, con la herramienta Adobe Photoshop CS2.

La Figura 29. Muestra el diseño del icono junto con otras aplicaciones de interés para el usuario. Los usuarios quieren tener cosas agradables en sus dispositivos móviles. El icono que representara la aplicación su diseño debe ser atractivo y amigable para llamar la atención del usuario, es importante mencionar que el icono debe verse bien en diferentes tamaños para los diferentes dispositivos móviles.

Figura 28. Las aplicaciones compiten con muchas otras



Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2 y Adobe Photoshop CS2.

Figura 29. **Icono de la aplicación Kispedia**



Fuente: elaboración propia, con la herramienta Adobe Photoshop CS2.

La figura 31 y 32 muestran la elaboración y el acabado del ícono con otro fondo. El icono tiene que verse bien en diferente tamaño y sobre diferentes fondos. Un icono con texto será ilegible y quedará con aspecto de mancha cuando lo ven en los dispositivos móviles, el uso de la letra inicial o unas siglas es una buena práctica o su defecto usar una imagen fácilmente reconocible y memorizable.

Figura 30. **Elaboración de icono con fondo café degradado**



Fuente: elaboración propia, con la herramienta Adobe Photoshop CS2.

Figura 31. **Icono kidspedia con fondo café degradado**



Fuente: elaboración propia, con la herramienta Adobe Photoshop CS2.

La figura 33 y 34 muestran la elaboración y el acabado del ícono con otro fondo. El uso de formas reconocibles y de tamaño adecuado permitirá que el usuario reconozca el contenido. Para el diseño del icono se tomó en cuenta el tamaño de los dispositivos también el uso del color debido a la importancia en el diseño, se tomó en cuenta a los usuarios a que va dirigido. Si son adultos es recomendable usar colores serios u oscuros. Y si son niños se debe optar por colores vivos y coloristas.

Figura 32. **Elaboración de icono diseño 2**



Fuente: elaboración propia, con la herramienta Adobe Photoshop CS2.

Figura 33. **Icono Kispedia diseño 2**

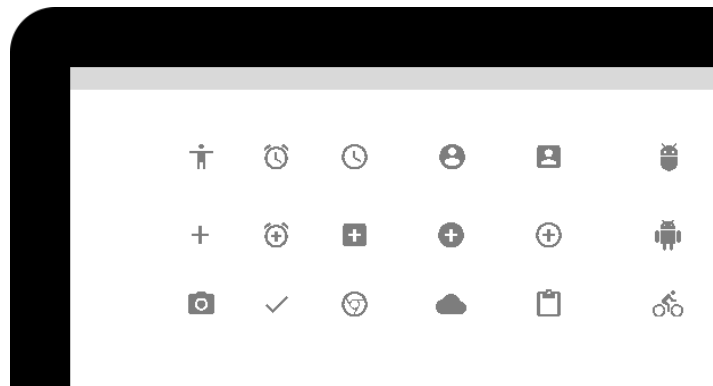


Fuente: elaboración propia, con la herramienta Adobe Photoshop CS2.

6.4.3. **Iconos Internos**

En la figura 35 se observan un conjunto de íconos. Son íconos que se encuentran dentro de la aplicación, la función es más funcional e importante.

Figura 34. **Íconos internos**

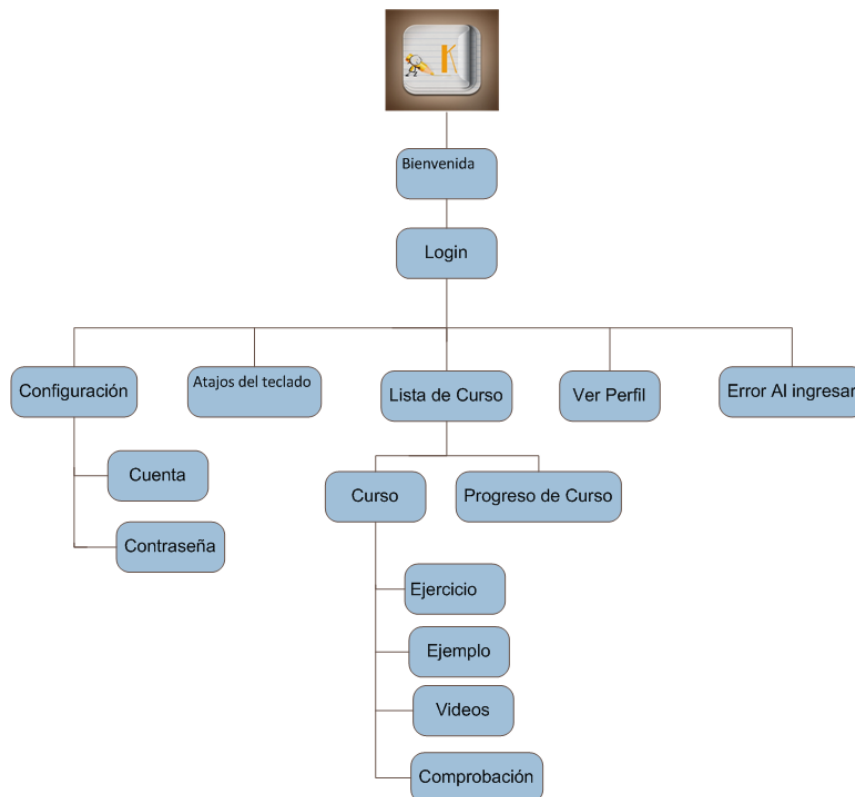


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.4. Diagrama de navegación

La figura 36 Diagrama de navegación de la aplicación. Se observa la estructura jerárquica de la APP, desde la pantalla de inicio o home a cualquier otra pantalla o contenido enlazado, muestra como es la navegación de la aplicación. Los nodos son las pantallas que visualiza el usuario que se encuentre utilizando la aplicación. Donde cada botón, link o tecla logra inducir una acción de vinculación así otro nodo, la navegación en algunos nodos es hacia una dirección y bidimensional.

Figura 35. Diagrama de navegación de la aplicación

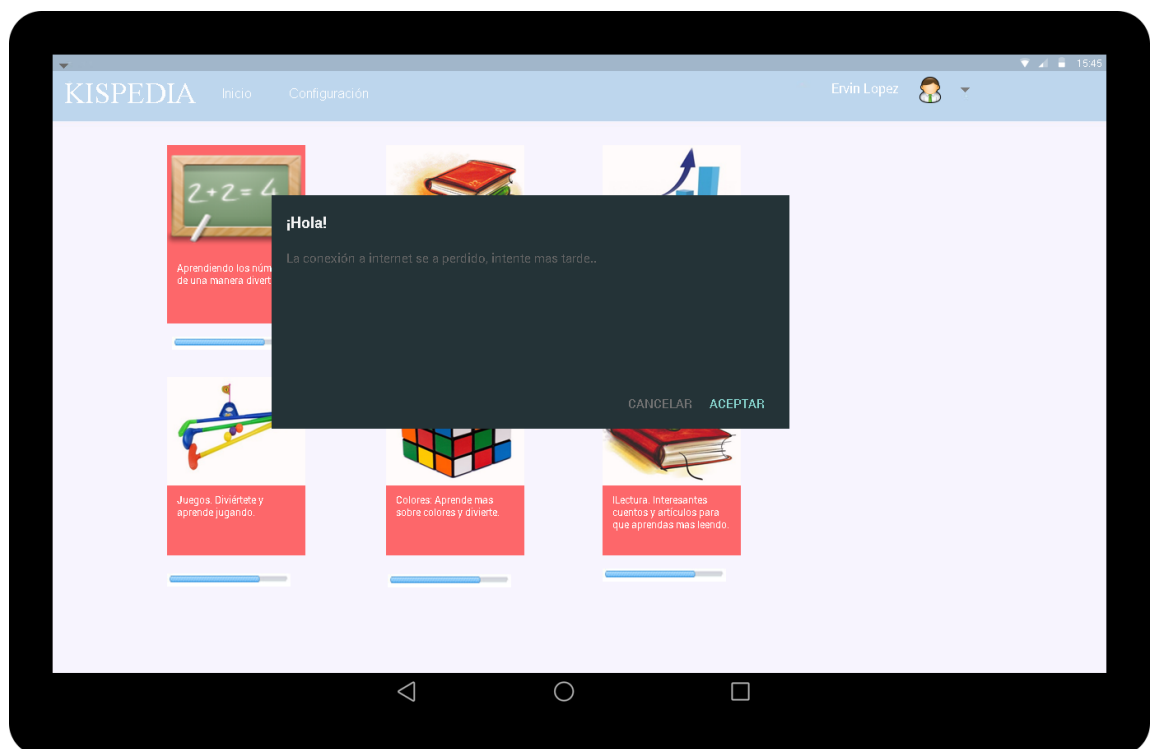


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.5. Cuadro de diálogo y mensajes

Figura 37 se observa el diseño de cuadro diálogo y de mensajes. Son elementos de interfaz de usuario. Los cuadros de diálogo y de mensaje se muestran siempre en la misma ubicación de pantalla, aparecen en respuesta a una acción del usuario. El cuadro de diálogo es utilizado para mostrar información específica al usuario, información urgente, mensaje de error, recopilar información del usuario. En muchos casos plantean pregunta o preguntas al usuario, que debe responder para eliminar el bloqueo del flujo principal de la aplicación.

Figura 36. Diseño de cuadro diálogo y de mensajes

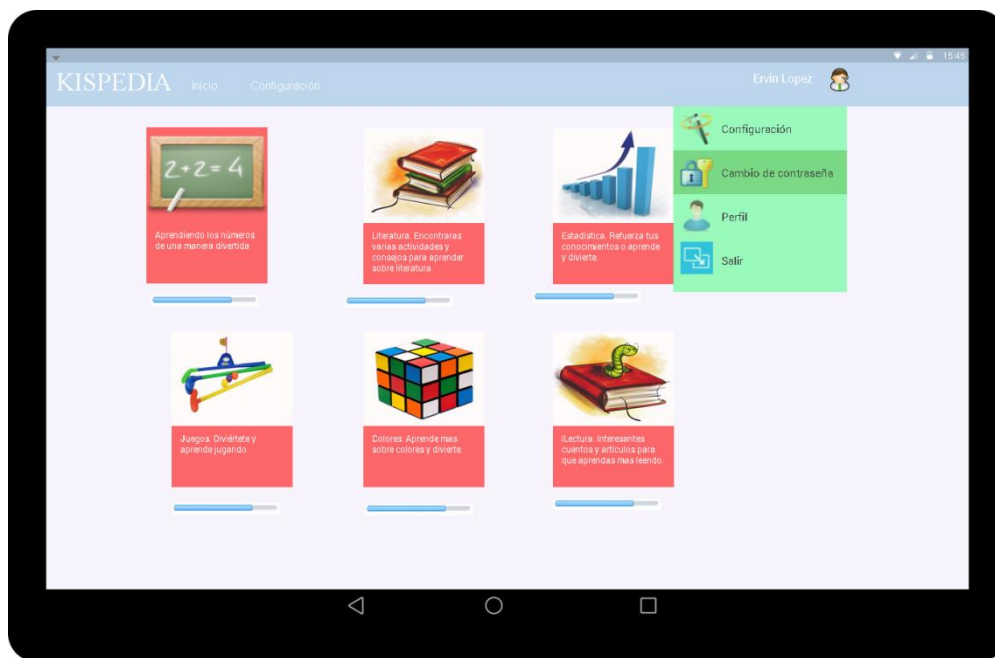


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.6. Diseño de menú

En la figura 38 se observa el diseño de menú desplegable. Los menús son elementos de interfaz utilizados en la mayoría de interfaz gráfica. El uso es fundamental porque le permite al usuario que tenga más control en la aplicación. El usuario no tendrá que memorizar comandos y cometer errores, por medio de los menús el usuario puede ejecutar comandos.

Figura 37. Diseño de menú

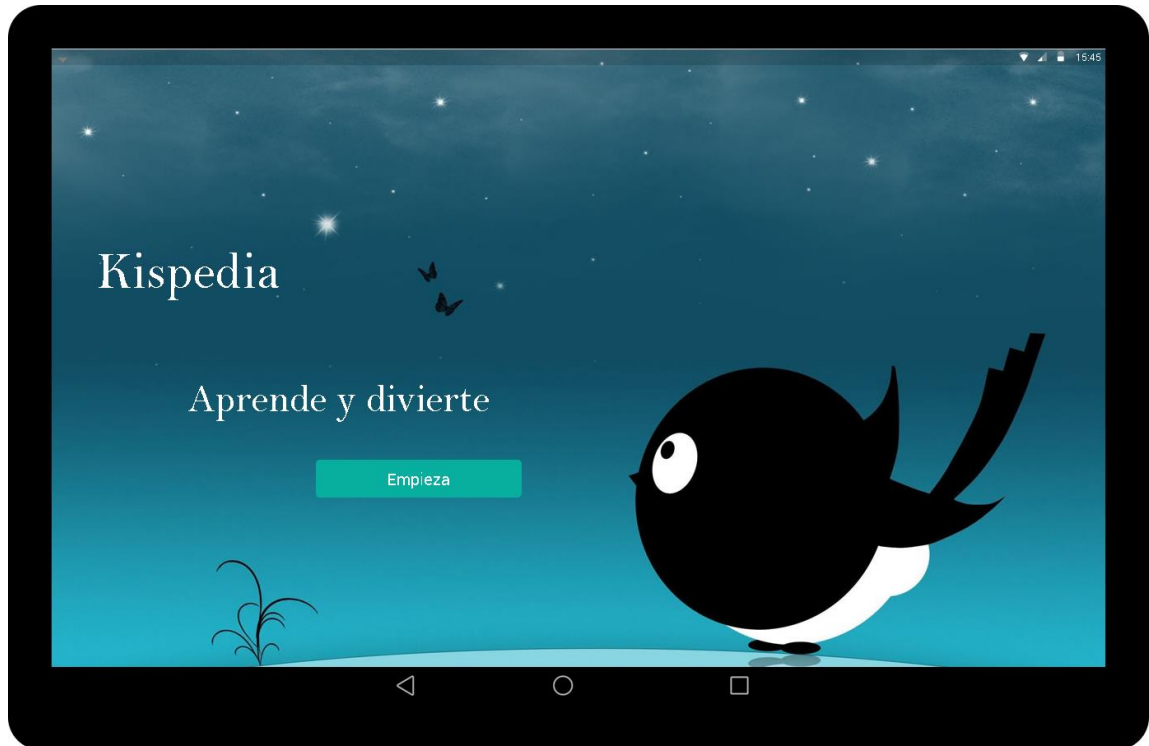


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.7. Pantalla de bienvenida

La Figura 39 es el diseño de la pantalla de bienvenida. Esta pantalla es la primera que visualiza el usuario luego de presionar el ícono de la aplicación y es visible para cualquier usuario que haya descargado la aplicación.

Figura 38. **Diseño de la pantalla de bienvenida**

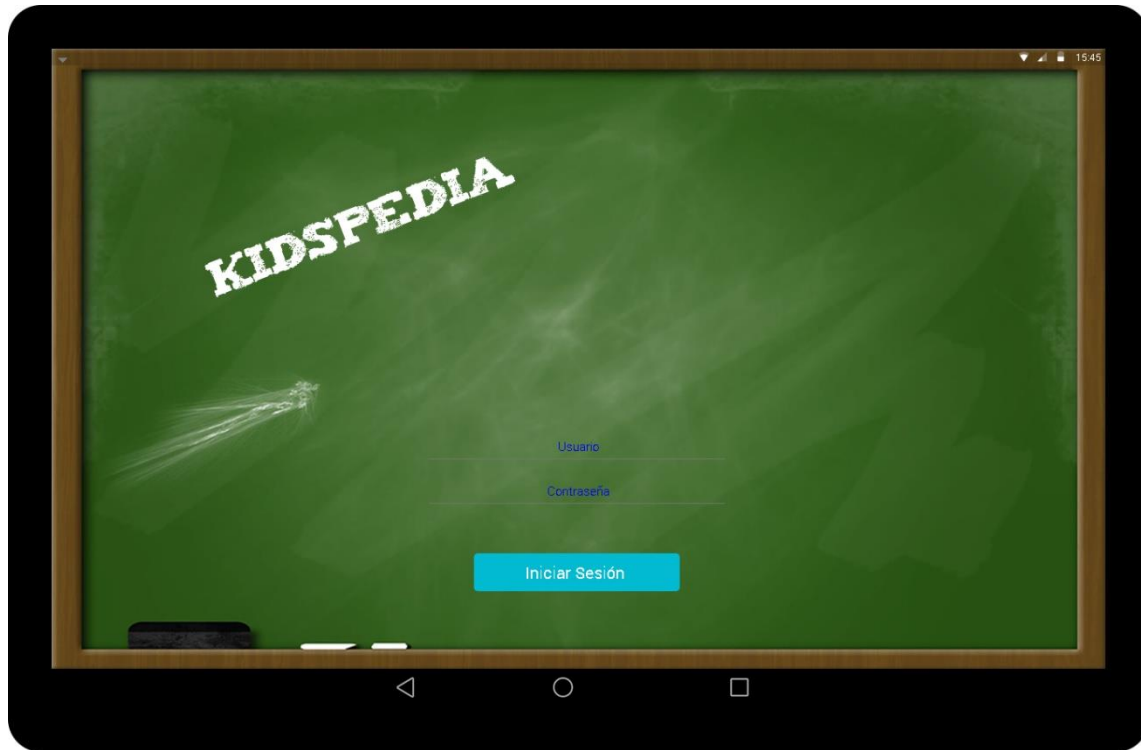


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.8. Pantalla login

La figura 40 muestra el diseño de la pantalla login. La pantalla de login o de inicio de KIDSPEDIA es utilizada para que el usuario se registre en la aplicación y tenga privilegios asociados a su cuenta. Esta pantalla muestra un formulario donde el usuario debe ingresar el nombre de usuario y la contraseña de la cuenta, la pantalla contará con un botón para ejecutar el proceso de autenticación. Su diseño incluye una imagen de fondo, esta imagen puede ser modificada y poner la que el usuario considere la apropiada.

Figura 39. **Diseño de la pantalla login**

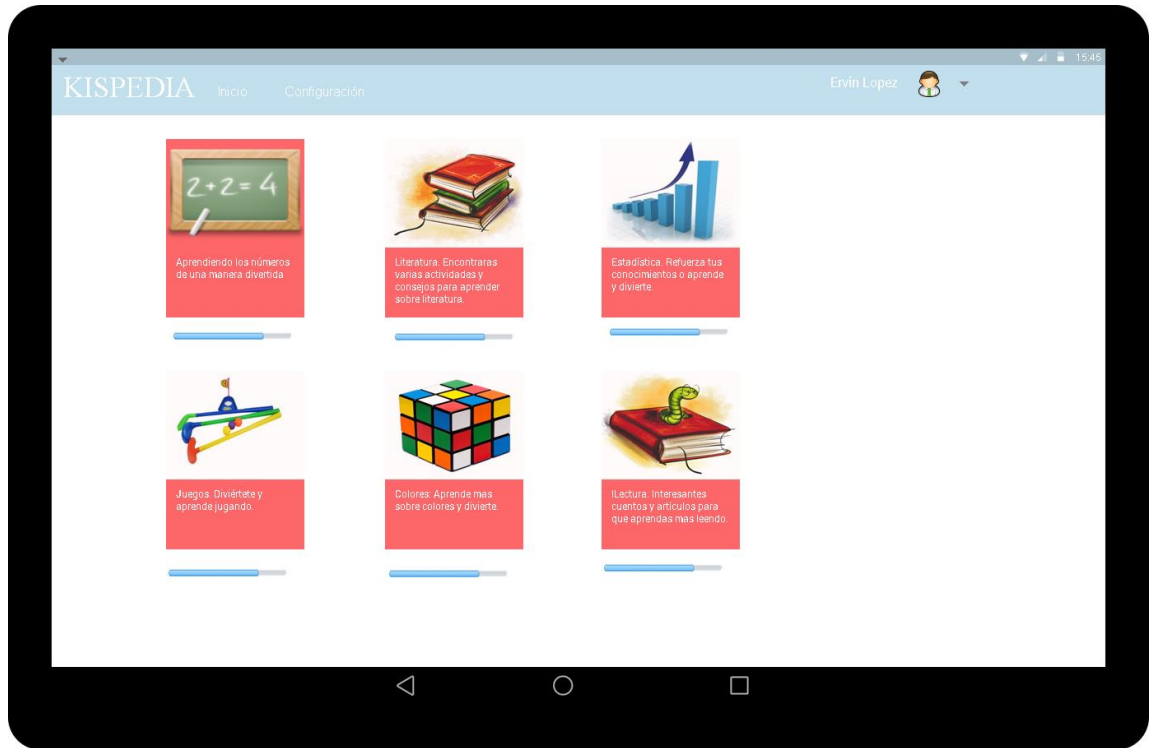


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.9. Pantalla lista de contenido

La figura 41 pantallas lista de contenido. Esta pantalla muestra todos los cursos que el estudiante tiene a su disposición. Luego del que el usuario se autentique ingresara a esta pantalla, el usuario podrá consultar el contenido del curso presionando doble sobre el ícono de del curso que el usuario dese. Para cada curso el diseño muestra e avance que el usuario tiene sobre cada curso.

Figura 40. **Pantalla lista de contenido**

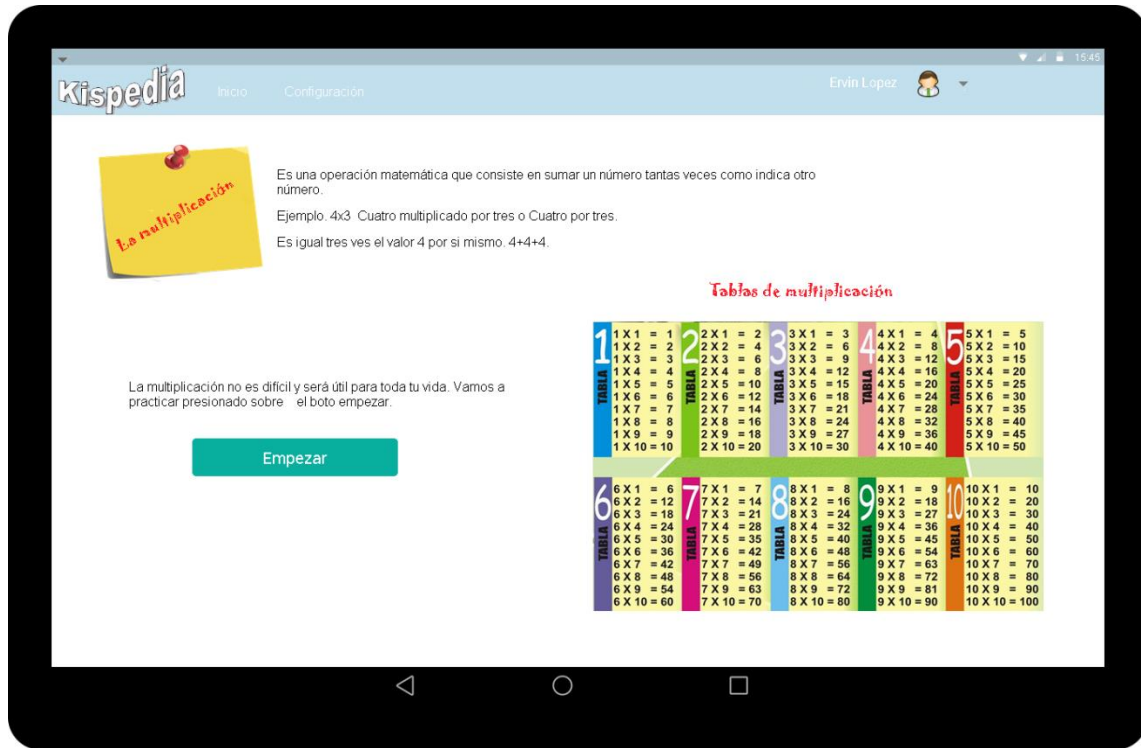


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.10. **Diseño pantalla contenido de curso de matemáticas**

La figura 42. Pantalla contenido de curso de matemáticas. Esta pantalla es el diseño para el contenido de matemáticas.

Figura 41. Pantalla contenido de curso de matemáticas

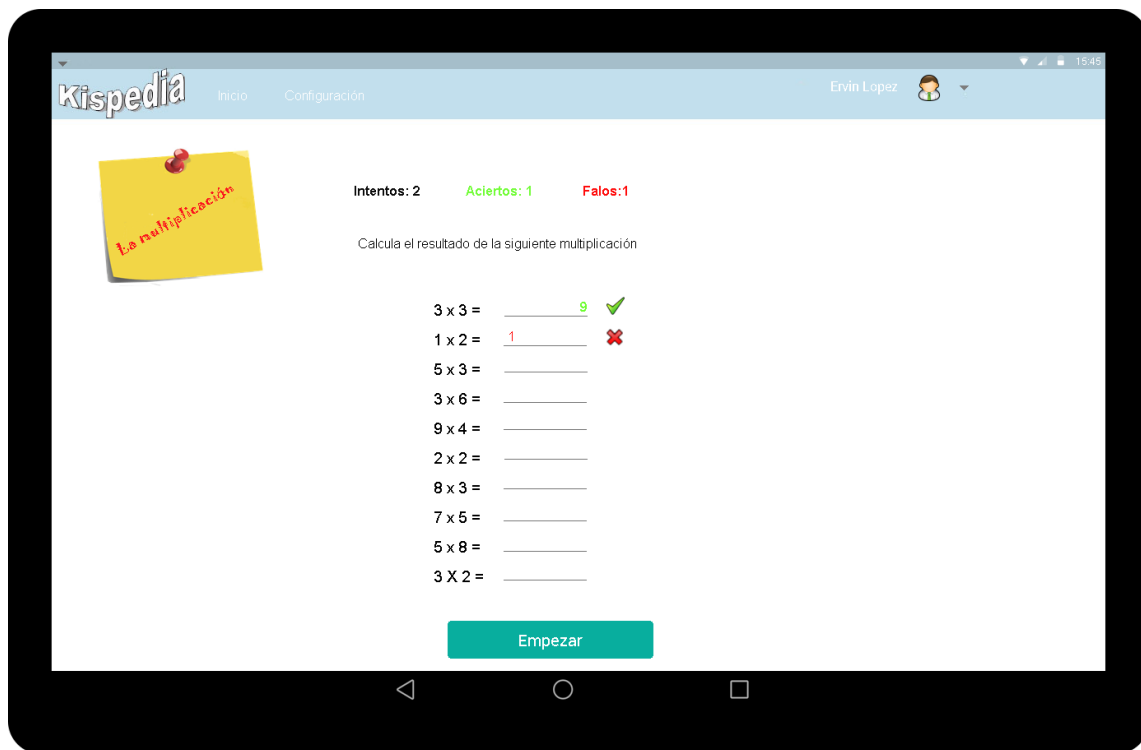


Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.11. Diseño pantalla ejercicio de matemáticas

La Figura 43 Pantalla Ejercicio de matemáticas. La siguiente pantalla el usuario ejercita el contenido consultado. La mejor forma de aprender matemáticas es practicando, resolver problemas una y otra vez. Es una buena forma de memorizar lo fundamental. Para aprender matemáticas el concepto tiene que ser aprendido y la base es la repetición por medio de ejercicios.

Figura 42. **Diseño pantalla ejercicio de matemáticas**



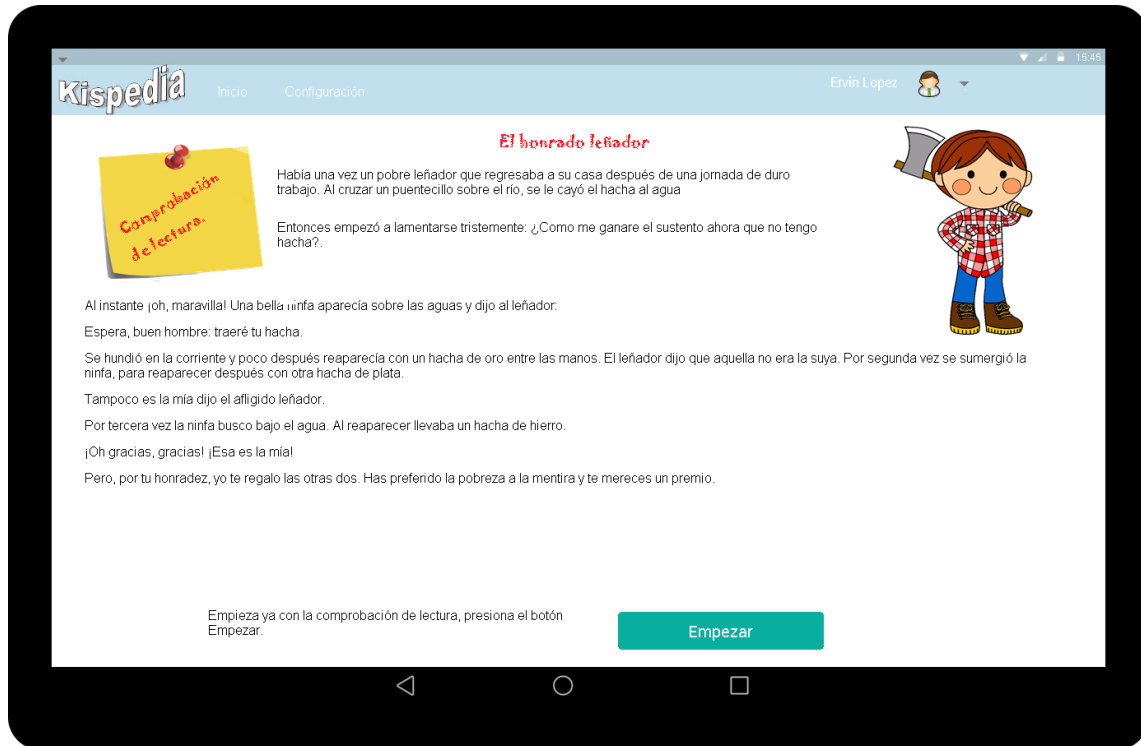
Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.12. **Diseño de pantalla del curso de lectura**

El diseño muestra el curso de lectura, con el avance de las tecnologías móviles la educación se beneficia grandemente ahora los estudiantes cuentan con muchos recursos, libros, medios electrónicos esto ahora espacio y ellos ya no tienen que cargar mucho peso.

La lectura es una actividad importante y útil para todo ser humano, es una forma que el ser humano cuenta para aprender y obtener conocimiento. La tecnología móvil ofrece grandes ventajas para la lectura.

Figura 43. **Diseño de pantalla del curso de lectura**



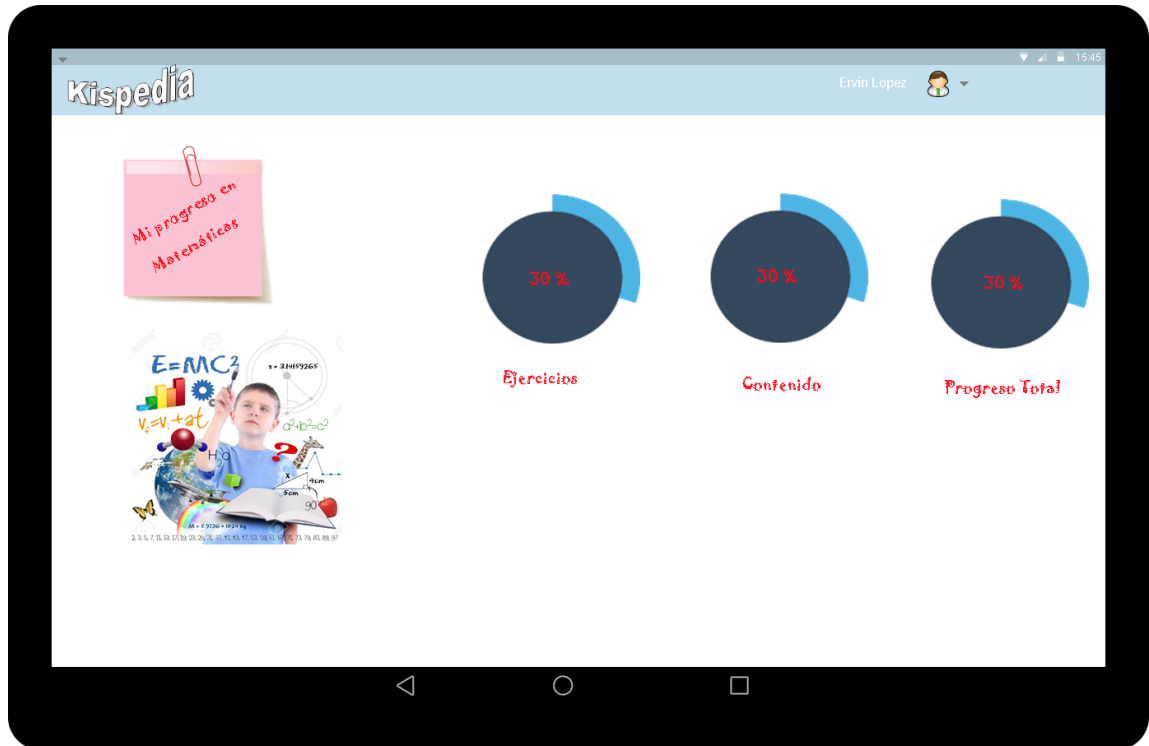
Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.13. Diseño pantalla progreso del curso de matemáticas.

El objetivo de esta pantalla es mostrar detalladamente el progreso de cada curso, como ejemplo se tomó el curso de matemáticas. En muchas ocasiones, el usuario necesita saber cuál es el desempeño que se tiene en el curso.

El diseño sigue un estándar con las otras pantallas, el fondo es de color blanco con una cabecera de color celeste. Los objetos gráficos fueron seleccionados de un conjunto de objetos gráficos. El diseño es sencillo y no complicado de entender, contiene poca información, lo que el usuario necesita.

Figura 44. **Diseño pantalla progreso del curso de matemáticas**



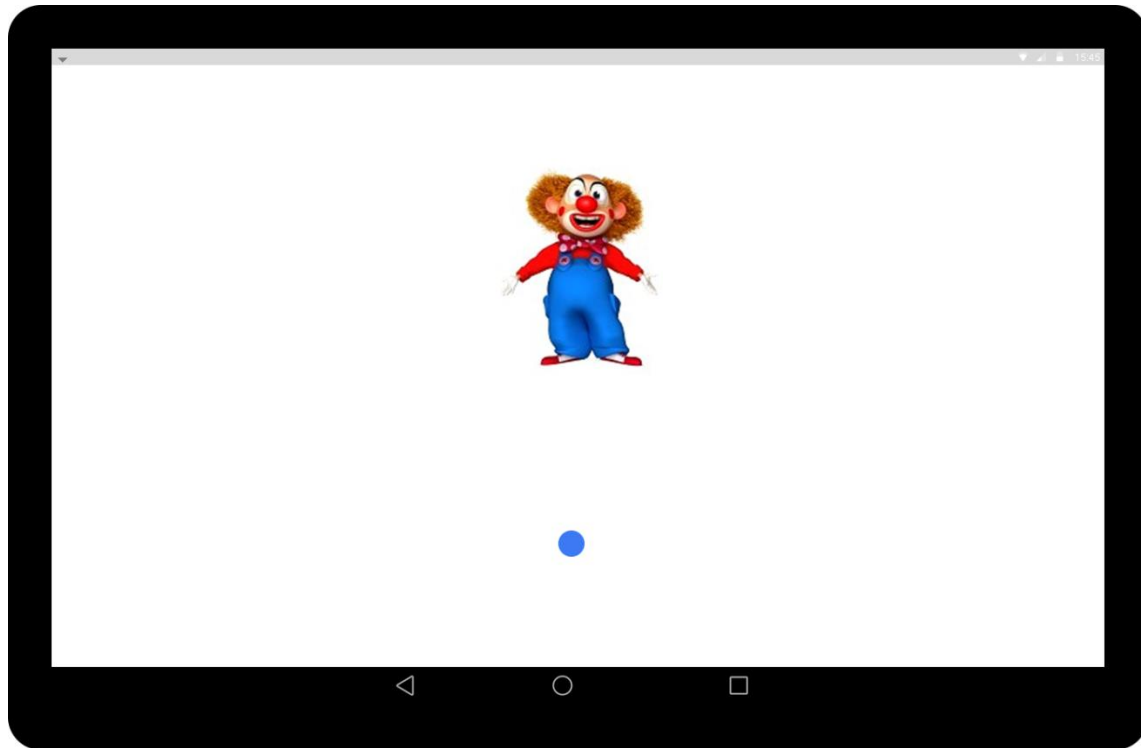
Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.14. **Diseño pantalla de error**

Los errores se dan en toda aplicación o programa de computadora y es el resultado de una acción no valida en la aplicación. Existen varios tipos de errores. Este diseño es sobre las pantallas de error y los mensajes de error que debe informar al usuario la situación de problema.

El diseño es muy sencillo y cumple con los objetivos de informar al usuario. La pantalla cuenta con un fondo blanco y una imagen amigable formada también por un mensaje que le indicara al usuario que sucede y que debe hacer.

Figura 45. **Diseño de pantalla de error**



Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

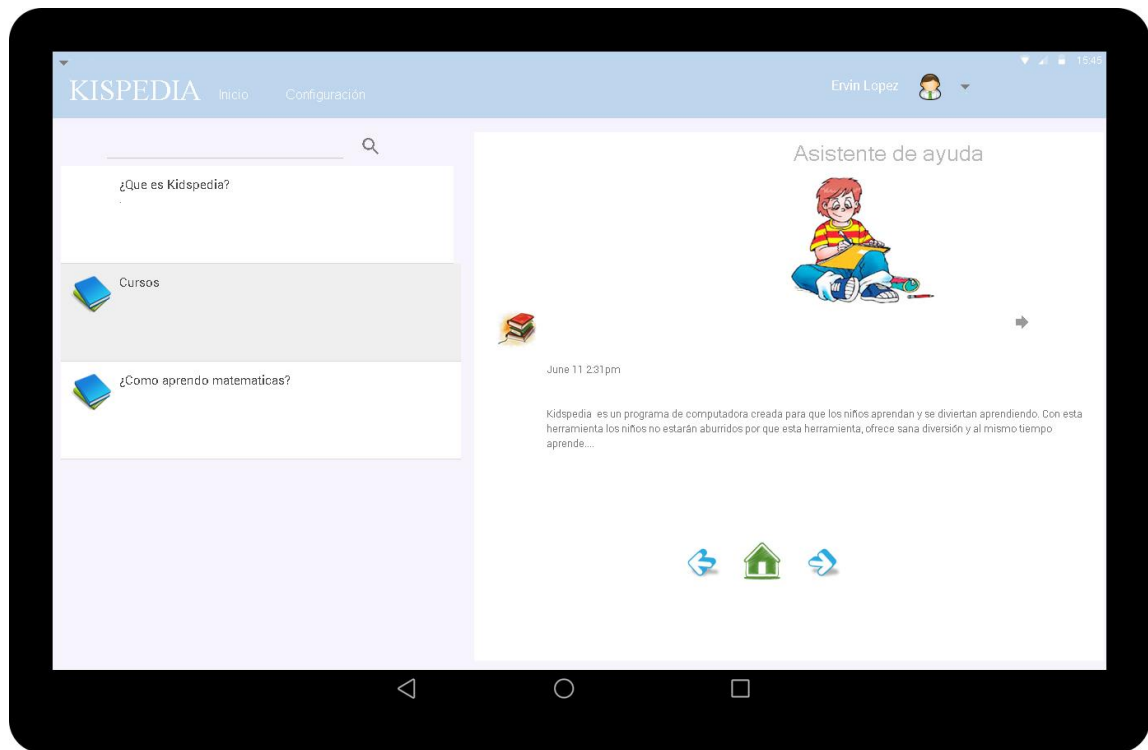
6.4.15. Diseño pantalla manual de la aplicación

Un objetivo de diseño es que la aplicación sea fácil e intuitiva de usar, pero una buena práctica de usabilidad también sugiere el diseño de ayuda, manuales con la finalidad de ayudar al usuario. Los manuales son guías básicas que indican cómo realizar cada una de las actividades que se lleva a cabo en el sistema.

Esta guía para el usuario está diseñada de tal forma que el usuario no le costara su uso, especialmente para aquellos usuarios novatos. El diseño cuenta con dos columnas. En la columna izquierda aparecen los términos básicos y la

columna derecha información sobre el término también cuenta con un buscador de términos.

Figura 46. **Diseño pantalla manual de la aplicación**



Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

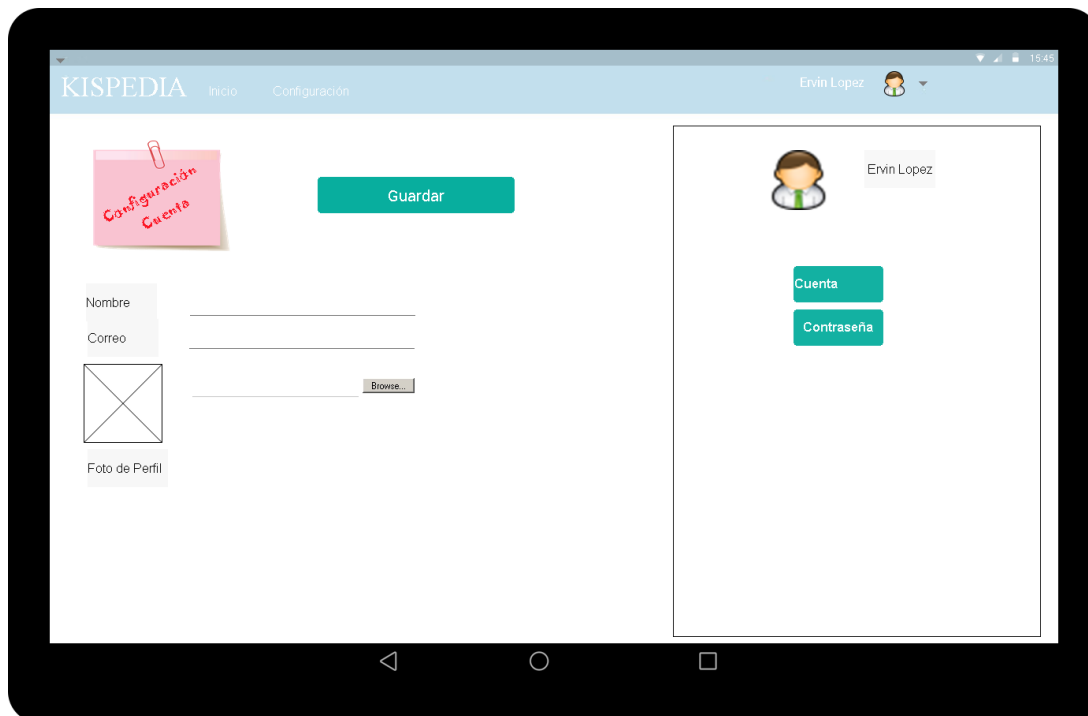
6.4.16. Diseño pantalla configuración de cuenta

El usuario podrá configurar su cuenta, cambiando su nombre de usuario, su correo electrónico y foto para el perfil.

El control del usuario es una regla de oro y esto quiere decir que el usuario se siente cómodo con una interfaz que ayude al usuario a controlar la situación en la aplicación. El diseño de esta pantalla cuenta con el fondo blanco y dos

columnas, en la columna izquierda están los elementos que el usuario desea cambiar y para que se guarde los cambios el botón guardar.

Figura 47. **Diseño pantalla configuración de cuenta**



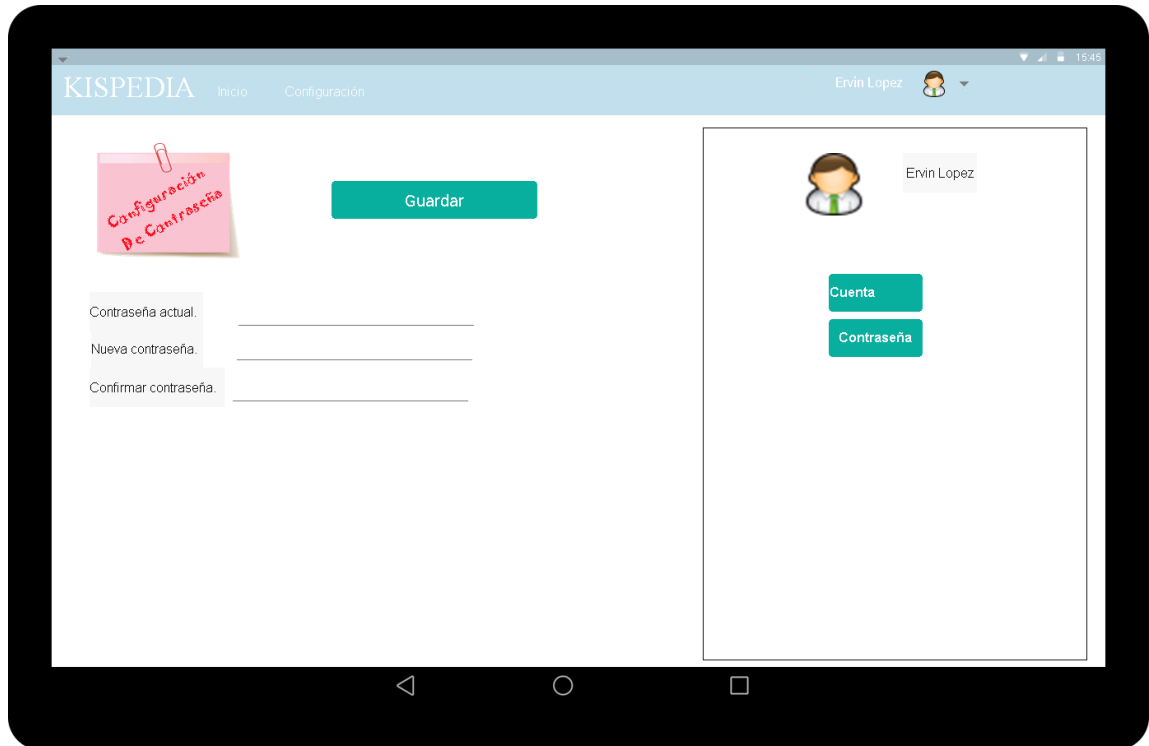
Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.4.17. Diseño pantalla configuración de contraseña

El usuario podrá cambiar la contraseña de su cuenta.

Este diseño cuenta con un fondo blanco. El usuario debe ingresar la contraseña actual, la nueva contraseña y confirma la contraseña. Por último, el botón de guardar es utilizado por el usuario para guardar el cambio de contraseña.

Figura 48. **Diseño pantalla configuración de contraseña**



Fuente: elaboración propia, con la herramienta JustinmindPrototyper 6.5.2.

6.5. Validación de interfaz

Se evalúan los efectos que haya producido en el usuario, en cuanto a logros, actitud e interés; los puntos a medir son: Intuitivo, fácil de utilizar, atractivo, adaptable, la interfaz Fue aceptado por la mayoría de los usuarios., etc. Una vez que todos los elementos en forma individual han sido probados, se procede a la prueba en conjunto de los mismos, esto para comprobar que funcionan perfectamente; en este momento pueden surgir problemas, que quizás lleven a rediseñar alguna parte del modelo, o incluso a desecharlo completamente por no ajustarse a los requerimientos señalados.

6.5.1. Rubricas a utilizar

Herramientas que utilice para estimar la calificación del desarrollo de interfaz, haciendo claros los objetivos y expresos los criterios que empleé para evaluarlos.

En este documento se encuentra las rubricas que se emplearon. Se estuvo experimentando con diseños de interfaz, por lo que es posible que requiera ajustar los criterios. Se espera que los usuarios indiquen si creen que cualquiera de los aspectos parece tener muchos excesos o podría mejorarse.

- Intuitivo
- Fácil de utilizar
- Atractivo
- Adaptable
- Fue aceptado por la mayoría de usuarios.

6.5.2. Preguntas evaluación de diseño

- ¿Te gustaron los colores del programa?
- ¿Te gustaron las figuras del programa?
- ¿Te pareció muy bonito y divertido aprender con este programa?

6.5.3. Preguntas evaluación de usabilidad

- ¿Encontraste el curso de matemáticas?
- ¿Encontraste el curso de lectura?
- ¿Te costó utilizar el programa?




6.5.4. Preguntas evaluación de accesibilidad

- ¿Desde qué dispositivos se puede acceder el programa?

6.5.5. Resultados de la validación

El experimento se llevó a cabo el veinticuatro de octubre de 2016. Lugar en donde se realizó la encuesta zona 5 de Mixco. Escuela Estado de Israel. El experimento consistió en la evaluación de la interfaz por un grupo de niñas estudiantes de nivel básico. Las alumnas interactuaron con la interfaz luego respondieron a una encuesta para conocer lo que ellas pensaban sobre la interfaz. A continuación el resultado de la encuesta. La primera pregunta se utilizó para dar indicaciones de cómo llenar la encuesta.

Figura 49. **Pregunta 1**

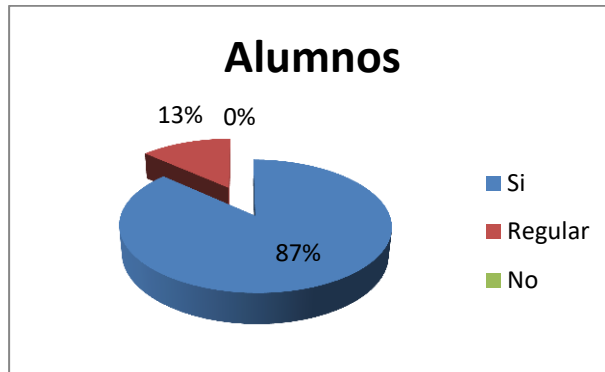
¿Cómo te encuentras en estos momentos de ánimo?		
 Muy feliz	 Regular	 Triste

Núm.	Respuesta
1	Regular
2	Triste
3	Regular
4	Muy feliz
5	Muy feliz
6	Muy feliz
7	Muy feliz
8	Muy feliz
9	Regular
10	Regular
11	Muy feliz
12	Muy feliz
13	Muy feliz
14	Muy feliz
15	Muy feliz

Respuesta	Alumnos
Muy feliz	10
Regular	4
Triste	1

Fuente: elaboración propia.

Figura 50. Gráfica PIE pregunta 1



Fuente: elaboración propia.

6.5.6. Evaluación de diseño

Figura 51. Pregunta 2

¿Te gustaron los colores del programa?

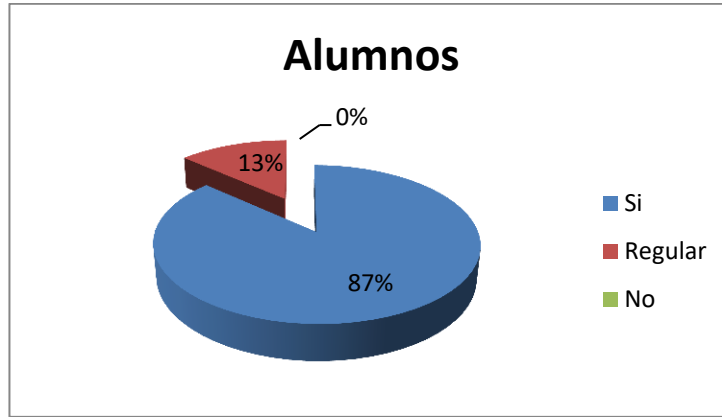
Los alumnos contestaron. Sí, regular o no.

No.	Respuesta
1	Si
2	Si
3	Si
4	Si
5	Si
6	Si
7	Si
8	Si
9	Si
10	Si
11	Si
12	Si
13	Si
14	Si
15	Regular

Respuesta	Alumnos
Si	14
Regular	1
No	0

Fuente: elaboración propia.

Figura 52. Gráfica PIE pregunta 2



Fuente: elaboración propia.

Figura 53. Pregunta 3

¿Te gustaron las figuras del programa?

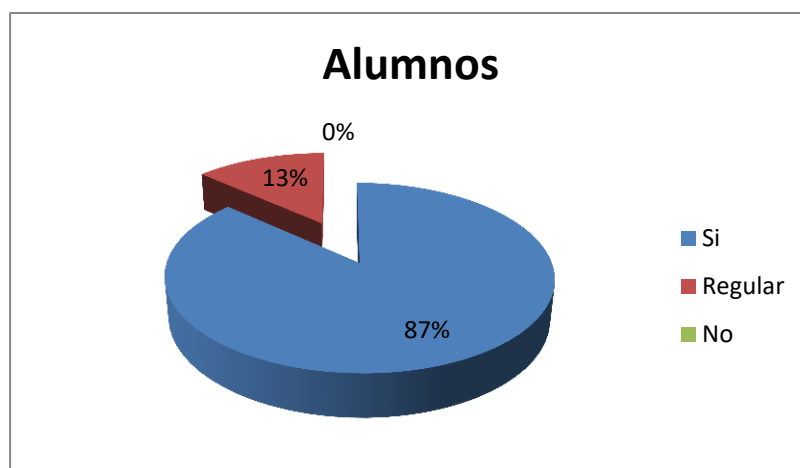
Los alumnos contestaron. Si, regular o no.

No.	Respuesta
1	Si
2	Si
3	Si
4	Si
5	Regular
6	Regular
7	Si
8	Si
9	Regular
10	Si
11	Regular
12	No
13	Regular
14	Si
15	Si

Respuesta	Alumnos
Si	9
Regular	5
No	1

Fuente: elaboración propia.

Figura 54. Gráfica PIE pregunta 3



Fuente: elaboración propia.

Figura 55. Pregunta 4

¿Te pareció muy bonito y divertido aprender con este programa?

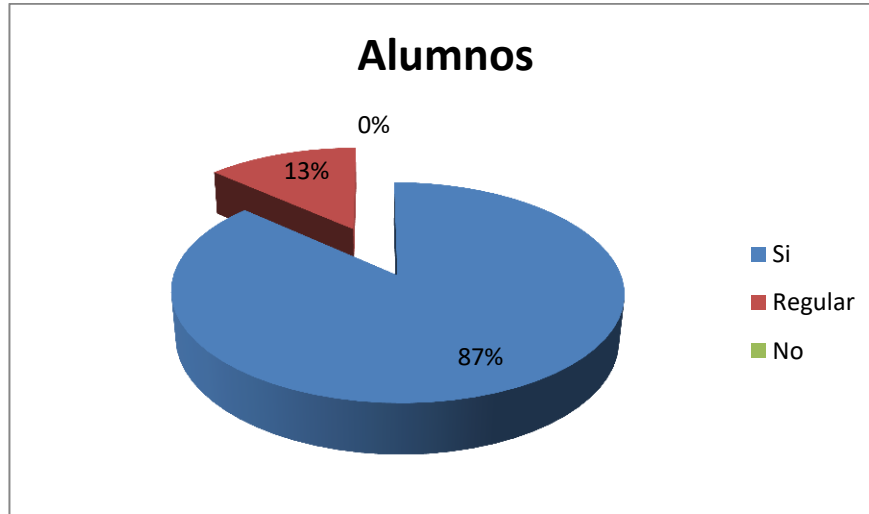
Los alumnos contestaron. Sí, regular o no

No.	Respuesta
1	Si
2	Si
3	Si
4	Regular
5	Si
6	Si
7	Si
8	Si
9	Regular
10	Si
11	Si
12	Si
13	Si
14	Si
15	Si

Respuesta	Alumnos
Si	13
Regular	2
No	0

Fuente: elaboración propia.

Figura 56. Gráfica PIE pregunta 4



Fuente: elaboración propia.

Figura 57. Pregunta 5

¿Encostraste el curso de matemáticas?

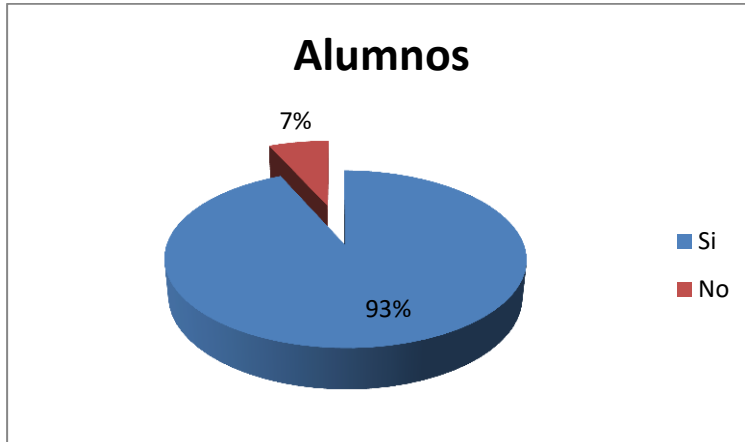
Sí. No.

No.	Respuesta
1	Si
2	No
3	Si
4	Si
5	Si
6	Si
7	Si
8	Si
9	Si
10	Si
11	Si
12	Si
13	Si
14	Si
15	Si

Respuesta	Alumnos
Si	14
No	1

Fuente: elaboración propia.

Figura 58. Gráfica PIE pregunta 5



Fuente: elaboración propia.

Figura 59. Pregunta 6

¿Encontraste el curso de lectura?

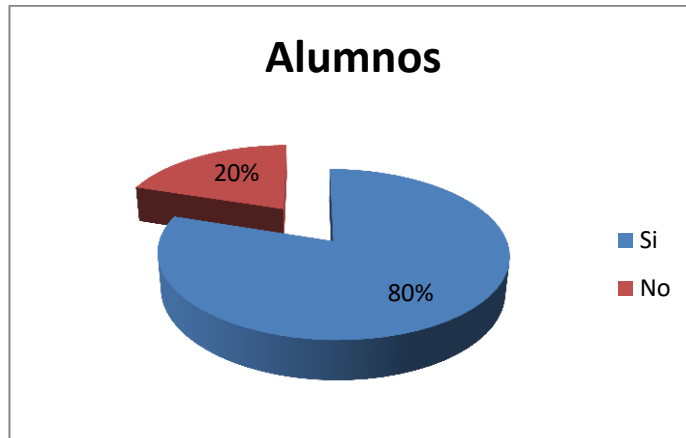
Sí. No.

No.	Respuesta
1	Si
2	Si
3	Si
4	Si
5	Si
6	Si
7	Si
8	No
9	No
10	No
11	Si
12	Si
13	Si
14	Si
15	Si

Respuesta	Alumnos
Si	12
No	3

Fuente: elaboración propia.

Figura 60. Gráfica PIE pregunta 6



Fuente: elaboración propia.

Figura 61. Pregunta 7

¿Te costó utilizar el programa?

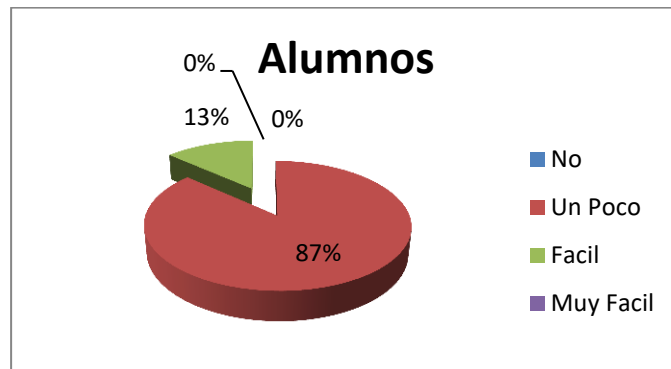
No Un Poco Fácil Muy Fácil

No.	Respuesta
1	Un poco
2	Un poco
3	Un poco
4	Un poco
5	un poco
6	Un poco
7	Fácil
8	Un poco
9	Un poco
10	Un poco
11	Un poco
12	Fácil
13	Un poco
14	Un poco
15	Un poco

Respuesta	Alumnos
No	0
Un Poco	13
Fácil	2
Muy Fácil	0

Fuente: elaboración propia.

Figura 62. Gráfica PIE pregunta 7



Fuente: elaboración propia.

7. DISCUSIÓN DE RESULTADOS

7.1. Discusión de resultados de validez de interfaz

- Preguntas evaluación de diseño
 - ¿Te gustaron los colores del programa?
 - ¿Te gustaron las figuras del programa?
 - ¿Te pareció muy bonito y divertido aprender con este programa?

Tabla VII. Promedio y resultados obtenidos

Pregunta	Si	Regular	No	Total
1	93	7	0	100
2	60	33	7	100
3	87	13	0	100
Promedio	80	17.67	2.33	100

Fuente: elaboración propia.

- Preguntas evaluación de usabilidad
 - ¿Encontraste el curso de matemáticas?
 - ¿Encontraste el curso de lectura?
 - ¿Te costó utilizar el programa?

Tabla VIII. **Promedio y resultados obtenidos**

Pregunta	Si	Regular	No	
1	93	0	7	100
2	80	0	20	100
3	13	87	0	100
Promedio	62	29	9	100

Fuente: elaboración propia.

- Preguntas evaluación de accesibilidad
 - ¿Desde qué dispositivos se puede acceder el programa?

No fue posible su evaluación, debido a que no se contaba con otros dispositivos.

Se determinó que la expectativa fue satisfactoria y hubo gran aceptación en cuanto colores, gráficas, dibujos e imágenes y el sistema fue intuitivo de utilizar. En cuanto evaluación de diseño en promedio se obtuvo un 80 % de aceptación, usabilidad en promedio el 62 % fue de aceptación, en cuanto a la evaluación de accesibilidad no fue posible su comprobación, debido a que no se contaba con otros dispositivos.

7.2. Necesidades del proceso de enseñanza-aprendizaje gestión y consulta de contenido

- Gestión de contenido
- Consulta de contenido

El software educativo principalmente debe poseer diferentes tipos de módulos que permita a los diferentes usuarios realizar actividades y funciones diferentes. Un módulo que gestione el contenido para satisfacer la necesidad de que el usuario pueda subir y transformar documentos y contenidos, como también publicar pruebas de evaluación con diferentes niveles de dificultad, que ayudarán a comprobar el grado de avance que puede tener el estudiante en el curso. Otra necesidad básica es la consulta de contenido para que el estudiante utilice y aprenda del contenido y realice las diferentes evaluaciones cuando se le indique.

7.3. Proceso de enseñanza-aprendizaje: gestión y consulta

La funcionalidad de gestión de contenido es el proceso de llevar a cabo acciones que hacen posible la realización de operaciones sobre el contenido como crear, editar, publicar contenido digital multimedia etc. con la finalidad de manejar de manera independiente el contenido y el diseño. Las cinco arquitecturas analizadas con anterioridad utilizan diferentes tecnologías para la gestión y consulta de contenido a continuación se mencionarán:

La arquitectura basada en servicios web y agentes inteligentes. Hace uso de servicios web para la selección y la integración de la funcionalidad que se requieren de un conjunto de servicios web de *e-learning* relacionado y existente. Para la consulta de contenido los agentes inteligentes a su vez van construyendo contenido dinámico para los usuarios según el perfil de cada estudiante.

Árbol de conocimiento: Una arquitectura distribuida para el aprendizaje en línea (*e-learning*). Esta arquitectura hace uso de servidores distribuidos y servicios web para la gestión y consulta de contenido, se centra en tres capas

que son: servidor de actividades, portales de aprendizaje y servidor de modelado de estudiantes.

Una Arquitectura de servicios web para contenido enriquecido desde clientes de aprendizaje móvil. Toda la gestión y distribución de contenido lo hace utilizando la arquitectura cliente servidor y con el uso de servicios web se conecta a otros sistemas para obtener contenido que luego es proporcionado al usuario a diferente dispositivo móvil.

Arquitectura basada en la web para el aprendizaje móvil. La arquitectura se basa en el modelo cliente-servidor hace uso de servicios web para enriquecerse de contenido, el contenido es gestionado desde un servidor web, el procesamiento principal se encuentra en el lado del servidor. El administrador de Servicios Web es responsable de la recopilación de material de aprendizaje desde la web en respuesta de una consulta del usuario. El administrador de servicios web invoca varios servicios web disponibles para obtener la información necesaria al finalizar el contenido es entregado al dispositivo móvil del usuario que realizó la petición.

Entrega de material e-learning consciente del contexto de usuario: enfoque y arquitectura. La base fundamental de esta arquitectura es el uso de ontologías y web semántica. La web semántica permite la estructuración semántica de los contenidos especialmente en la web. Para el modelado semántico se hará uso de la tecnología ontológica, ya que permite la anotación eficiente de los contenidos.

7.4. Proceso de enseñanza-aprendizaje sistema para la consulta de contenido KIDSPEDIA

Se han identificado tres capacidades básicas que debe poseer todo software educativo y que son contenidas en módulos por los diversos software educativos. La primera es el módulo de gestión de contenido, el segundo la consulta de contenido, y por último, un módulo de administración de cursos y estudiantes.

Respondiendo a las capacidades descritas, se incluye en la arquitectura los siguientes componentes:

Servidor de contenido dinámico, servidor de comprobación, servidor de servicios y servidor de base de datos.

Servidor de contenido dinámico contiene toda la lógica, los procesos requeridos para la creación de contenido dinámico y programas de contenido dinámicos. El contenido dinámico se crea por medio de la información preliminar del perfil de cada usuario. Luego que el usuario consulte el contenido se le hará una evaluación con respecto al contenido, esto servirá para mostrarle también contenido dinámico. Las actividades fundamentales del servidor son: Crear programas de aprendizaje y proporcionar a cada usuario contenido de acuerdo a su perfil y avances obtenidos.

El servidor de comprobación proporciona al usuario evaluaciones para medir la comprensión obtenida de cada material que se le ha dado por medio del servidor de contenido dinámico. Servirá para medir y conocer el logro, el avance de cada usuario y proporcionarle material dinámico según el desempeño de sus evaluaciones.

Servidor de servicios hace uso de protocolos y estándares como XML. La base principal de esta tecnología es el intercambio de mensajes. Utilizada en el sistema para obtener información no encontrada en la base de datos del sistema de consulta de contenido. La información es solicitada por el usuario, se busca primero en la base de datos local, al no ser encontrada la información solicitada por el usuario, se hará uso de un servicio web para buscar información en otros sistemas. El servidor de servicios se comunica, interactúa con otros sistemas, cuando la información es encontrada, se obtiene la información del otro sistema, se presenta la información al usuario y se almacena en la base de datos local.

El servidor de base de datos utiliza un potente Sistema de Gestión de Base de Datos SGBD por medio del cual permite realizar consultas complejas sobre los datos utilizando lenguaje SQL y el acceso concurrente de muchos usuarios autorizados a los datos. Entre sus funciones principales están: almacenar, eliminar y gestionar la información del usuario, información de cursos, contenido, texto, imágenes, audio etc. La importancia del servidor de datos es enorme porque en ella se almacena toda la información del sistema y muchos procesos, servidores acceden a él para solicitarle datos.

Los componentes descritos anteriormente proveen las siguientes capacidades: Interoperabilidad es una característica que poseen algunas arquitecturas la cual implica el intercambio de información entre sistemas heterogéneos. La tecnología que a menudo es utilizada para lograr tal propósito es la de Servicios Web entre otras tecnologías.

Aprendizaje personalizado, el contenido es dinámico, no es simplemente almacenar en base de datos y presentar siempre lo mismo a los usuarios, el

aprendizaje personalizado implica en la arquitectura funcionalidad para proveerle al software el crear contenido dinámico para cada usuario.

Uso de recursos externos, actualmente es necesario para la integración con otros sistemas y aprovechar el universo que existe en internet además colaborar con otras aplicaciones similares o complementarias para proporcionarle al software una mayor capacidad. Existen tecnologías para lograr esta características un ejemplo claro de tecnología a utilizar son los servicios web.

La seguridad es una característica de calidad que todo software debe considerar y poseer, más cuando se trata de proteger toda clase de información en muchos casos el uso de seguridad extrema implica la disminución de otras características de calidad e importantísimas debe existir un balance en el tema de seguridad y utilizar la necesaria.

E-learning acceso móvil es el enfoque que le dan las arquitecturas para la consulta del contenido, con el avance de la tecnología actualmente los dispositivos móviles es posible ver cualquier contenido de cualquier software pero cuando el contenido se especializa para dispositivos móviles es sin duda mucho mejor y transparente para el usuario su uso con respecto al contenido, no significa que para el resto de las arquitecturas que no sea posible visualizar el contenido desde un dispositivo móvil sea malo simplemente el enfoque es diferente.

Portable es una característica importante de calidad y pocas arquitecturas de software la tienen. Portabilidad no implica relacionarse directamente con el código fuente de la aplicación, más bien el concepto de portabilidad se refiere a la propiedad que posee un software ser ejecutado en diferentes plataformas o

sistemas operativos y hardware. La ventaja que poseen las arquitecturas de software con estas características es la de poder migrar tanto en software como hardware cuando sea necesario. Es definitivo que en muchas ocasiones es necesario migrar hacia un distinto sistema operativo y hardware con mejores características sin afectar el servicio y en general con el buen funcionamiento del software.

Escalable es una característica deseable para un sistema el cual implica la habilidad de la arquitectura para reaccionar, adaptarse y manejar el crecimiento continuo de trabajo de manera fluida sin perder calidad del servicio que ofrece el sistema. También para estar preparado para hacerse más grande el software en cuanto agregar más componentes de software sin perder calidad en los servicios que se ofrecen.

7.4.1. Interfaz amigable del software educativo

La premisa inicial del diseño de la interfaz fue crear interfaces no solo atractivas sino sencillas y funcionales con la finalidad de ofrecer la mejor forma a la comunicación con dispositivos móviles en donde se implementará el sistema.

Previamente a la elaboración del diseño de interfaz, se realizaron prototipos de la interfaz de usuario. Con la ayuda de los prototipos que mostraban un panorama general de la interfaz de pantallas.

- Diseño del icono del sistema para la consulta KISPEDIA

Es el icono que representa al sistema de la consulta de contenido KISPEDIA se visualizará en el dispositivo móvil cuando se instale la aplicación y servirá como acceso directo. El resultado final es un ícono con imagen con

forma reconocible por los usuarios, tamaño adecuado que permitirá que el usuario reconozca el contenido. Diseño dirigido de acuerdo a los usuarios potenciales. En el diseño principal se visualiza un conjunto de hojas sueltas, sobre las hojas un niño con un crayón dibujando la primera letra de nombre principal del sistema.

- Diseño de diagrama de navegación

Se diseñó con el objetivo de observar cuál es la estructura jerárquica de la aplicación, en el diagrama se visualiza desde la pantalla de inicio hasta cualquier otra pantalla o contenido enlazado. Los nodos son las pantallas que visualiza el usuario que se encuentre utilizando la aplicación. Donde cada botón, tecla o link puede provocar una acción de vinculación así otro nodo, la navegación en algunos nodos es unidireccional y bidimensional.

- Diseño de cuadro de diálogo y mensajes

Los cuadros de diálogo y de mensaje se muestran o aparece siempre en la misma ubicación de pantalla, producto de una acción del usuario. El cuadro de diálogo fue creado para mostrar información específica al usuario, información urgente, mensaje de error, recopilar información del usuario. En muchos casos plantean preguntas al usuario, que debe responder para eliminar el bloqueo del flujo principal de la aplicación.

- Diseño de menú

Esta pantalla es la primera que visualiza el usuario luego de presionar el icono de la aplicación y es visible para cualquier usuario que haya descargado la aplicación. El menú se diseñó utilizando componentes básicos que

proporcionan la herramientas de diseño, imágenes que describen las funciones a ejecutar y colores configurables que en conjunto la hace agradables de utilizar.

- Pantalla de bienvenida

Esta pantalla es la primera que visualiza el usuario luego de presionar el icono de la aplicación y es visible para cualquier usuario que haya descargado la aplicación. El diseño cuenta con pocos objetos de interfaz de usuario pero se logra el objetivo principal que es presentar el sistema. Los objetos de interfaz son los siguientes: imagen de fondo agradable, textos de bienvenida y botón utilizado para poder ingresar a la aplicación.

- Pantalla login

Creada para la autenticación del usuario. El usuario debe ingresar su nombre de usuario y contraseña, el sistema verifica los datos y si el usuario está registrado podrá ingresar y utilizar la aplicación. Los usuarios que no están registrados no podrán ingresar, pero podrán registrarse y cuando lo hagan podrán aprender y disfrutar todo lo que ofrece la aplicación.

- Pantalla de lista de contenido

Creada para presentar el listado dinámico de cursos disponible para cada usuario. El servidor de cursos dinámicos es el encargado de evaluar y proporcionar cursos. Para interactuar con el curso el usuario tiene que presionar sobre el link del curso.

Esta pantalla muestra todos los cursos que el estudiante tiene a su disposición. Luego del que el usuario se autentique ingresará a esta pantalla, el usuario podrá consultar el contenido del curso presionando doble clic sobre el icono del curso que el usuario desee. Para cada curso el diseño muestra el avance que el usuario tiene sobre cada curso.

- Pantalla contenido de curso de matemáticas

Creada para interactuar con el contenido de matemáticas. El diseño es sencillo, no se sobrecarga con objetos de interfaz porque no se desea confundir al usuario con sobrecarga de objetos de interfaz en muchos casos no necesarios. La pantalla presenta el contenido dinámico del curso de matemática, el fondo de la pantalla es de color blanco.

- Pantalla ejercicio de matemáticas

Se creó esta pantalla para presentar al usuario ejercicios de matemáticas. La mejor forma de aprender matemáticas es practicando, resolver problemas una y otra vez. El diseño de la interfaz presenta contenido dinámico, el enfoque es presentar ejercicios utilizados para realizar ejercicios de matemáticas para diferentes niveles educativos, el fondo es de color blanco igual que la mayoría de pantallas de interfaz descritas anteriormente y no se sobrecarga de objetos de interfaz.

- Pantalla del curso de lectura

El siguiente diseño muestra el curso y contenido de lectura. La lectura es una actividad importante y útil para todo ser humano, es una forma que el ser humano cuenta para aprender y obtener conocimiento. La pantalla de curso de

lectura ofrece grandes ventajas para la lectura, principalmente por el diseño que atrae a los potenciales usuarios, el fondo a utilizar es el color blanco, el contenido es dinámico e interactivo e ilustrado con imágenes relacionadas con la lectura.

- Pantalla progreso del curso de matemáticas

Utilizada para mostrar detalladamente el progreso de cada curso, en muchas ocasiones el usuario necesita saber cuál es el desempeño que se tiene en el curso o cursos. Los objetos gráficos que se utilizan para la pantalla progreso del curso de matemática son: fondo blanco, imágenes atractivas e informativas y objetos de progreso de contenido, utilizadas para informar al usuario y no generar cansancio por la saturación de objetos de interfaz. El diseño es sencillo y fácil de entender, contiene poca información, solamente lo que el usuario necesita.

- Pantalla de error

Este diseño es sobre las pantallas de error y los mensajes de error que debe informar al usuario la situación de problema. Los errores se dan en toda aplicación o programa de computadora y dispositivos móviles, es el resultado de una acción no válida en la aplicación. Existen varios tipos de errores, los tipos de errores más comunes son los que se producen por fallos inesperados en el sistema como por ejemplo la pérdida de conexión de internet, memoria insuficiente en otros errores y los errores producidos por el usuario al realizar acciones no permitida en sistema.

El diseño es muy sencillo y cumple con los objetivos de informar al usuario. La pantalla cuenta con un fondo blanco y una imagen amigable

formada también por un mensaje que le indicara al usuario que sucede y que debe hacer.

- Pantalla configuración de cuenta
- Pantalla configuración de contraseña

7.4.2. Casos de usos relevantes del sistema

Los casos de usos que a continuación mencionaremos son los más relevantes y en conjunto con los requerimientos no funcionales, permitieron descubrir y diseñar la arquitectura del sistema.

- Autenticación del sistema

Importante para la mayoría de software e implica la funcionalidad del sistema permitir al usuario ingresar a la aplicación, por medio de un código de usuario y contraseña. El usuario se registrará al sistema en el caso de que no lo estuviera y los usuarios registrados tendrán a disposición todos los servicios que le corresponde como usuario.

El sistema para la consulta de contenido KIDSPEDIA es utilizado solamente por aquellos usuarios registrados y autorizados.

- Consultar programas

Utilizado para especificar el comportamiento del sistema cuando un usuario realiza la consulta sobre el contenido de un curso, para que permita realizar tal acción el usuario debe estar correctamente autenticado. El usuario correctamente autenticado ingresa al área de programas y realiza la petición el

servidor por su parte recibe, busca la información, crea el programa al usuario y finalmente entrega información del programa al usuario que realiza la petición.

- Consultar contenido

Similar al caso de uso anterior a diferencia de que en vez de programa el servidor estará entregando contenido de curso después de que el usuario haya realizado la petición. Es un caso de uso importante porque sin esta funcionalidad del sistema para la consulta de contenido KISPEDIA no sería posible la consulta de contenido.

Utilizado para especificar el comportamiento del sistema cuando un usuario realiza la consulta sobre el contenido de un curso, para que el sistema permita realizar tal acción el usuario debe estar correctamente autenticado. El usuario correctamente autenticado ingresa al área de cursos y realiza la petición. El servidor por su parte recibe, busca la información, crea el contenido al usuario y finalmente entrega información del contenido al usuario que realiza la petición.

- Responder preguntas

Utilizado para especificar el comportamiento del sistema cuando un usuario realiza la acción de responder preguntas. Las preguntas son creadas por el servidor dinámicamente y entregadas al usuario, el usuario por medio de un dispositivo móvil visualiza el contenido, responde preguntas.

Esta funcionalidad es importante, el sistema para la consulta de contenido KISPEDIA, por medio de esta funcionalidad del sistema se verifica el avance que tiene el usuario con respecto a los cursos que le fueron asignados.

- Consultar evaluación

Utilizado para especificar el comportamiento del sistema cuando el usuario realiza la acción de consultar evaluación o nota de evaluaciones. Para que el usuario realice esta acción el usuario debe ser un usuario autorizado, debidamente registrado en el sistema. El usuario ingresa al área de contenido dinámico, presiona en el link o botón de resultados de evaluación, el servidor de comprobación recibe la solicitud, prepara los resultados de cursos y finalmente los resultados son entregados al usuario que realiza la petición.

7.4.3. Esquema de base de datos de la solución

Esquema de base de datos, la mayoría de software necesita de alguna manera almacenar información, el sistema para la consulta de contenido KISPEDIA no es la excepción, por esa razón, se optó por la utilización de base de datos.

Se utilizó una base de datos relacional, que satisface las necesidades en cuanto a todo lo que se refiera a almacenar y manejo de información.

El uso de base de datos relacional implica la elaboración de diseño de base de datos, en el diagrama diseñado se observa la abstracción de los datos en tablas con entidades, atributos y sus relaciones que existen entre ellos. Este esquema le provee al sistema el manejo eficiente del contenido, esto ayuda a proporcionar al usuario información correcta y en el menor tiempo posible.

El diseño de esquema de base de datos que se propuso representa una posible solución sencilla de la gestión de contenido dinámico para el sistema para la consulta de contenido KISPEDIA. Existen conceptos que hay que manejar como: usuarios, autenticación, alumnos, cursos, asignatura, categoría, contenido y ejercicio.

7.4.4. Impactos sociales

Sistema KISPEDIA ofrece una nueva propuesta eficaz y eficiente sistema educativo de apoyo acorde con la programación dada en la formación básica del niño, su enfoque es sobre contenido educativo dinámico. El proyecto propuesto cambia la visión que tienen los alumnos hacia las habituales formas de aprendizaje rompiendo los límites del salón de clase tradicional.

Se trata con el software educativo disminuir la reprobación y la falta de interés por parte de los alumnos que tienen sobre el aprender. En un futuro, se espera mejoren la calidad de vida de su familia, ya que se conoce que una buena educación es la base de cualquier individuo, elementos vitales de su formación para concluir una profesión.

Las cualidades que ofrece el sistema para la consulta de contenido KISPEDIA sin duda harán que los alumnos tomen el gusto a las matemáticas por medio de las diversas actividades que se realizan en la aplicación. Aprender matemáticas será divertido y dejarán atrás sus ideas negativas de las mismas.

Otro impacto social es la permitir el derecho de niñas y niños a la comunicación y el acercamiento a uso de las tecnologías.

7.4.5. Impacto tecnológicos

La educación vive una realidad con el surgimiento y desarrollo de la Tecnología diferente a lo se venía haciendo, se presentan cada vez más como una necesidad de la sociedad donde los rápidos cambios, el aumento de los conocimientos y las constantes demandas de educación de alto nivel constantemente actualizada, cada vez es necesario nuevos proyecto tecnológicos educativos y de calidad.

Sistema para la consulta de contenido KIDSPEDIA es un proyecto tecnológico innovador porque hace uso de recursos tecnológicos con la finalidad de apoyar la enseñanza y aprendizaje de los niños de Guatemala, utiliza conceptos modernos de aprendizaje se da por la combinación de la informática y las telecomunicaciones. Las tecnologías a utilizar por el sistema que se propones son las siguientes:

Arquitectura se software novedoso que facilita la comunicación, interpretación entre interesados del sistema, en el diseño destaca las decisiones que tendrán profundo impacto en el éxito y permita proporcionarle al software las capacidades necesarias para su mantenimiento y evolución, de acuerdo a las necesidades del negocio y las soluciones requeridas de los usuarios. Un sistema gestor de base de datos y diseño de base de datos que hace eficiente el manejo de la información. Interfaz amigable del software educativo para que su utilización sea fácil y aceptada por la mayoría de los usuarios. Las tecnologías que se propone es para que la gestión y entrega de contenido sea eficiente y dinámico su enfoque es hacia los dispositivos móviles.

Otro aspecto que hace del SISTEMA PARA LA CONSULTA DE CONTENIDO KIDSPEDIA relevante es porque es un proyecto tecnológico

intuitivo y fácil de utilizar con información confiable y actualizada, pero también es necesario que los profesionales relacionados con la educación conozcan las TICs para tomar decisiones adecuadas y orientar a sus alumnos. Deben de existir más proyectos innovadores utilizando las tecnologías y de esa manera se puede lograr una mejora en el aprendizaje.

CONCLUSIONES

1. A través de la evaluación de interfaz de usuario, realizada en la Escuela Estado de Israel zona 5 de Mixco, el 24 de octubre del 2016 se determinó que la expectativa fue satisfactoria y hubo gran aceptación en cuanto a colores, gráficas, dibujos, imágenes y diseño de las interfaces evaluadas. En cuanto a evaluación de diseño, en promedio se obtuvo un 80 % de aceptación, usabilidad en promedio el 62 % fue de aceptación. En cuanto a la evaluación de accesibilidad no fue posible su comprobación, debido a que no se contaba con otros dispositivos. Ver capítulo 6 validación de interfaz y capítulo 7 discusiones de resultados.
2. El diseño de la arquitectura se basa en componentes, se realizó de forma que permite su mantenimiento y su evolución fácilmente. Los principales componentes son los siguientes: servidor de base de datos, servidor de contenido dinámico, servidor de servicios y servidor de comprobación.
3. Cada componente de la arquitectura es un objeto de *software* destinada a interactuar con otros componentes. Los principales componentes fue el producto de un análisis de cinco arquitecturas que se mencionan en este documento en la sección de antecedente y se tomó lo mejor para el proyecto en cuanto a costo y tecnología. Los principales componentes de la arquitectura son los siguientes: servidor de base de datos su función principal es la de gestionar toda la información del sistema. Servidor de contenido dinámico es el encargado de crear contenido para el usuario, se crea el contenido y es entregado al usuario. Servidor de servicios su función principal es la de obtener información no encontrada en el

servidor de base datos del sistema de consulta de contenido, la información es solicitada por el usuario, se busca primero en el servidor de base de datos, al no ser encontrada la información solicitada por el usuario, se hará uso de un servicio web para buscar la información en otros sistemas. Servidor de comprobación su función es medir y conocer el logro y avance de cada usuario, proporcionarle material dinámico según el desempeño de sus evaluaciones.

4. Se diseñaron las principales interfaces de usuario y objetos de interfaz. Para su diseño se utilizaron las mejores prácticas y estándares de diseño. Las herramientas de prototipado de interfaces gráficas son las siguientes: BALSAMIQ MOCKUPS, MOCKINGBIRD, MOCKFLOW y Microsoft Visio. Los objetos de interfaz de usuario diseñados fueron los siguientes: Cuadro de diálogos, mensajes informativos y de error, iconos, diagrama de navegación. También las diferentes pantallas: de bienvenida, login, lista de contenido, ejercicios, contenido y configuraciones.
5. Se diseñaron cinco casos de uso que son las principales para el sistema: Autenticación del usuario, consulta de programa de cursos, consulta de contenido, responder preguntas y consulta de evaluación.
6. El esquema de base de datos se diseñó utilizando modelo entidad-relación, porque permite la abstracción, representación de datos y cómo se relaciona entre ellos. El diseño le provee al sistema, el manejo eficiente del contenido, esto ayuda a proporcionarle al usuario información correcta y en el menor tiempo posible. Se diseñaron 12 entidades porque cumplen con las diversas funcionalidades requeridas por el sistema, las principales son: autenticación de usuarios, para el

control y autorización del usuario. Almacenamiento de contenido para la gestión de contenido como lo es la creación de contenido, creación de programas, creación de ejercicios, creación de listado de cursos.

RECOMENDACIONES

1. La arquitectura de software e interfaz de usuarios de este proyecto está enfocado en dispositivos móviles, un programa de software enfocado a dispositivos móviles en muchos casos funciona diferente cuando es utilizada en una computadora personal. Una de las mejoras que podría realizarse es agregar un mecanismo de visualización que se ajuste a la pantalla de despliegue del dispositivo a través del cual se acceda.
2. El servidor de base de datos y el esquema de base de datos su estructura se enfoca en contenido para alumnos de nivel básico de Guatemala y todo el contenido es en español. Guatemala es un país multicultural rica en tradiciones, razas y lenguas. Una de las mejoras que podría realizarse es agregar un nuevo servidor de base datos y diseñar un nuevo esquema de base de datos para poder almacenar y gestionar los nuevos requerimientos de información.
3. No existe software perfecto, pero sí es posible acercarse a tal definición. La usabilidad es un término intangible y bastante utilizado cuando se desarrolla un software, pero implica gran esfuerzo para dotar a todo software, características importantes para su aceptación, por lo tanto es importante que el término usabilidad sea medible, para lograr tales fines es necesario tener recursos, con el fin de realizar experimentos necesarios que permitan medir la usabilidad del producto que se desea diseñar y desarrollar.

4. Para el diseño de interfaces de usuario y otros objetos de interfaz deben utilizarse las mejores prácticas y estándares de diseño, también se recomienda involucrar durante el diseño a niños, ya que es la única forma de asegurar un diseño final usable y accesible porque no basta tener presente que los niños en cuanto a sus habilidades, preferencias, necesidades son diferentes, cambian conforme van creciendo.

REFERENCIAS BIBLIOGRÁFICAS

1. Alzaabi, M., Berri, J., y Jamal, M. Z. (Marzo de 2010). *Web-based Architecture for mobile learning*.
2. Association, M. M. (2011). *Libro Blanco de Apps*. España.
3. Bachmann, F., Bass, L., C. Clements, P., Garlan, D., Ivers, J., Little, R., y otros. (2010). *Documenting Software Architectures: Views and Beyond, Second Edition*. Addison-Wesley Professional.
4. Booch, G., Jacobson, I., & Rumbaugh, J. (2006). *Using UML Software engineering with objects and components 2nd edition*. Pearson.
5. Brusilovsky, P. (2014). KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. 10.
6. Clarenc, C. A., Castro, S. M., López, C., Moreno, M. E., & Tosco, N. B. (2013). *SE ANALIZÓ 19 PLATAFORMAS DE E-LEARNING*.
7. De Castro Lozano, C. (30 de septiembre de 2012). El futuro de las tecnologías digitales aplicadas al aprendizaje de personas con necesidades educativas especiales. Recuperado el 25 de abril de 2016, de RED, *Revista de Educación a Distancia*: <http://www.um.es/ead/red/32/carlos.pdf>

8. Fowler, M., Kobryn, C., Booch, G., Jacobson, I., & Rumbaugh, J. (2003). *UML Distilled Third Edition A Brief Guide to the Standard Object Modeling Language*. Pearson Education, Inc.
9. Date J., C. (2001). *Introducción a los sistemas de base de datos*. México: Pearson Educación.
10. Lee, W.-M. (2012). *Beginning Android 4 Application Development*. Estados Unidos de América.
11. Ricardo M., C. (2009). *Base de Datos*. McGraw-Hill.
12. Meier, J., Homer, A., Hill, D., Taylor, J., Bansode, P., Wall, L., y otros. (2009). *Mobile Application Architecture Guide*.
13. Pressman, Roger S. (2010). *Ingeniería del software un enfoque práctico*.