



UNIVERSIDAD INTERNACIONAL DEL ECUADOR - LOJA

FACULTAD DE INFORMÁTICA Y MULTIMEDIA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INFORMÁTICA Y MULTIMEDIA**

TEMA:

**“DESARROLLO DE UNA PLATAFORMA WEB Y MÓVIL DE
COMUNICACIÓN PARA EL GOBIERNO AUTÓNOMO DESCENTRALIZADO
DE LA PARROQUIA DE MALACATOS”**

AUTORA:

NADIA KATHERINE NARVÁEZ FIGUEROA

DIRECTOR:

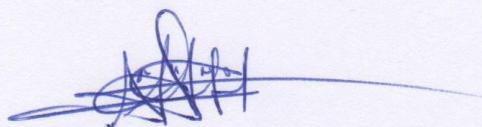
ING. LUIS CUENCA

FEBRERO 2015

LOJA ECUADOR

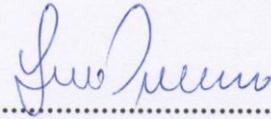
Yo, Nadia Katherine Narváez Figueroa, declaro bajo juramento, que el trabajo aquí descrito es de mi autoría, que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, reglamento y leyes



.....
Nadia Katherine Narváez Figueroa
CI: 1104597776

Yo, Luis Cuenca, certifico que conozco a la autora del presente trabajo, siendo la responsable exclusiva tanto de su originalidad y autenticidad, como de su contenido.

A handwritten signature in blue ink, appearing to read 'Luis Cuenca', is positioned above a horizontal dotted line.

Ing. Luis Cuenca

CI: 0704682012

Resumen

El Gobierno Autónomo Descentralizado de la Parroquia de Malacatos es una Institución pública ubicada en la ciudad de Loja, al sur del Ecuador. Institución caracterizada por promover una orden activo y creativo apoyando al desarrollo de su parroquia, con principios morales, ético y cívico.

La parroquia Malacatos se ubica al Sur de la República del Ecuador, pertenece a la Región 7, Provincia y cantón Loja. Limita al Norte por la parroquia urbana San Sebastián del Cantón Loja y parroquia rural El Tambo del cantón Catamayo; al Sur por la parroquia rural San Pedro de Vilcabamba; al Este por el cantón Palanda provincia de Zamora Chinchipe; y al Oeste por la parroquia rural Purunuma del cantón Gonzanamá y una pequeña parte del cantón Quilanga.

El motivo primordial del desarrollo de éste proyecto se debe a que el GAD Parroquial de Malacatos como Institución Pública principal de la Parroquia, necesita la implementación de una plataforma web y una móvil para mejorar su comunicación entre empleados y manejar información de manera eficiente y sencilla con toda la población Malacatense.

En el presente trabajo se describen las tareas realizadas según la metodología de desarrollo llamada Kanban: empezando en la fase de análisis y diseño de la plataforma, utilizando diferentes métodos de recopilación de información; posteriormente la implementación de la solución, pruebas generales y finalmente conclusiones y recomendaciones

Así mismo se muestra el capítulo del Estado del Arte, donde se detallan fundamentos teóricos generales de las tecnologías empleadas para la implementación de la solución del presente proyecto.

Abstract

The Autonomous Decentralized Government Parish Malacatos is a public Institution located in the city of Loja, in the south of the Ecuador. Institution characterized for promoting an active and creative order supporting to the development of its parish, with moral, ethical and civic beginning.

The parish Malacatos is located on the South of the Republic of the Ecuador, there belongs to the Region 7, Province and canton Loja. It borders to the North for the urban parish San Sebastian of the township Loja and rural parish The Tambo of the township Catamayo; on the South for the rural parish Saint Peter from Vilcabamba; to the east for the township Palanda province of Zamora Chinchipe; and on the West for the rural parish Purunuma of the township Gonzanamá and a small part of the township Quilanga.

The essential motive of the development of this project is due to the fact that the Parochial GAD of Malacatos like main Public Institution of the Parish, he needs the implementation of a web platform and the mobile to improve its communication between workers and to managing information with efficient and simple way with the whole population Malacatense.

In the present work describes the tasks realized according to the methodology of development called Kanban: starting in the phase of analysis and design of the platform, using different methods of compilation of information; later the implementation of the solution, general tests and finally conclusions and recommendations

Likewise shows the chapter of the State of the Art, where detailed general theoretical essentials of the technologies used for the implementation of the solution of the present project.

El presente trabajo de Tesis agradezco a Dios por haberme dado salud, inteligencia y sabiduría para alcanzar este logro profesional de mi vida universitaria.

A mis padres Angela Figueroa Zambrano y Juan José Narvárez Sarango por ser el pilar indispensable en mi vida y por todos sus consejos y sacrificios que me han llevado a alcanzar esta meta profesional.

A mis hermanos Angélica y Juan por ser mi compañía más preciada, por compartir los momentos más preciosos de mi vida y por su ayuda y apoyo incondicional que me brindaron en los momentos que más lo necesité, mis sinceros agradecimientos.

A Paul por ser mi fortaleza en mi diario vivir y por sus palabras de aliento que me motivan a culminar mis metas propuestas.

A mi director de Tesis Ing. Luis Cuenca por su don de gente y experiencia profesional por impartirme sus conocimientos y motivación para la culminación de mi Tesis Universitaria, y sin olvidar a todos mis docentes quien han sido principales actores de mi formación.

Nadia Katherine Narvárez Figueroa

**“DESARROLLO DE UNA PLATAFORMA WEB Y MÓVIL DE
COMUNICACIÓN PARA EL GOBIERNO AUTÓNOMO DESCENTRALIZADO
DE LA PARROQUIA DE MALACATOS”**

Introducción	11
Problematización	12
Objetivos.....	13
Generalidades.....	14
CAPÍTULO 1	15
1. Marco Institucional	15
1.1 Fundamentos Teóricos.....	16
1.2 Tecnologías	17
1.2.1 Tecnologías Móviles	17
1.2.1.1 HTML5	17
1.2.1.2 CSS y HTML	20
1.2.1.3 JavaScript	21
1.2.2 Tecnologías Web.....	23
1.2.2.1 WordPress	23
1.2.2.2 Lenguaje De Programación PHP	26
1.2.2.3 Framework SLIM.....	27
CAPÍTULO 2.	29
2. Estado del Arte	29
2.1 jQuery Mobile	29
2.1.1 Selección de Elementos.....	29
2.1.2 Conceptos Clave: GET vs. POST	30
2.2 Phonegap.....	32
2.3 Sencha Touch	33
2.3.1 Características de Sencha Touch.....	33
2.4 Ventajas de Los Tres Frameworks.....	34
2.5 Desventajas De Los Tres Frameworks.....	35
2.6 Metodología de Desarrollo Kanban	35
2.6.1 Los principios de la metodología Kanban.....	36
2.6.2 Pasos para configurar tu estrategia de Kanban.....	37
2.6.3 Las tres reglas de Kanban	38
2.6.4 Tablero Kanban.....	39
CAPÍTULO 3	40
3. Análisis y Diseño	40

3.1 Actores:.....	40
3.2 Ámbito del Sistema	40
3.3 Diagrama de Casos de Uso	41
3.4 Partes del Sistema.....	42
3.5 Diseño de la Base de Datos	43
3.6 Administración De Requerimientos	44
3.6.1 Entrevistas	44
3.6.2 Casos De Uso Expandidos.....	47
3.6.2.1 Plataforma Movil	47
3.6.2.2 Plataforma Web	51
CAPITULO 4.	54
4. Implementación de la Solución.....	54
4.1 Plataforma Web	54
4.2 Plataforma Movil.....	54
4.3 Web Service.....	58
4.3.1 Métodos	61
CAPÍTULO 5.	64
5. Pruebas	64
5.1 Pruebas en la Plataforma Movil.....	64
5.2 Pruebas en la Plataforma Web.....	66
6. Conclusiones y Recomendaciones	68
7. Bibliografía	70
8. ANEXOS	72
ANEXO 1.....	72
ANEXO 2.....	75
ANEXO 3.....	81
ANEXO 4.....	107

Índice de Ilustraciones

Ilustración 1. Familia HTML5.....	17
Ilustración 2. Diseño Común web.....	19
Ilustración 3 Funcionamiento Básico.....	20
Ilustración 4. Inicio de WordPress.....	24
Ilustración 5. Escritorio de WordPress	24
Ilustración 6. Publicaciones WordPress.....	25
Ilustración 7. Funciones del Escritorio.....	25
Ilustración 8. Plataformas Móviles	32
Ilustración 10. Tablero Kanban.....	39
Ilustración 11. Diagrama de Caso de Uso }.....	41
Ilustración 12. Partes del Sistema	42
Ilustración 13. Esquema de base de Datos.....	43
Ilustración 14. Diseño plataforma Movil	64
Ilustración 15. Diseño Plataforma Web.....	66
Ilustración 16. Visualización de los Proyectos en la Plataforma Web.....	66
Ilustración 17. Oficina Malacatos	72
Ilustración 18. Oficina Malacatos 1	72
Ilustración 19. Tesorería Malacatos	73
Ilustración 20. Tesorera.....	73
Ilustración 21. Presidenta del GAD.....	74
Ilustración 22. Secretaria del GAD.....	74
Ilustración 23. Plataforma Web Inicio	75
Ilustración 24. Plataforma Web Misión.....	75
Ilustración 25. Plataforma Web Malacatos	76
Ilustración 26. Plataforma Web Miembros GAD	76
Ilustración 27. Plataforma Web Proyectos.....	77
Ilustración 28. Plataforma Web Turismo.....	77
Ilustración 29. Plataforma Web Chat	78
Ilustración 30. Plataforma Movil Ingreso	78
Ilustración 31. Plataforma Movil Login.....	79
Ilustración 32. Plataforma Movil Registrarse	79
Ilustración 33. Plataforma Movil Administración Proyectos	80
Ilustración 34. Plataforma Movil Listado de Proyectos	80

Índice de Tablas

Tabla 1. Ficha Técnica.....	14
Tabla 2. Ficha Técnica (Fundamentos Generales)	16
Tabla 3. Ventajas de los Frameworks (jQuery Mobile, Phonegap y Sencha Touch)	34
Tabla 4. desventajas de los Frameworks (jQuery Mobile, Phonegap y Sencha Touch)	35
Tabla 5. Principios Kanban	37
Tabla 6. Resultados de Entrevista 1.....	45
Tabla 7. Explicación Entrevista 1	45
Tabla 8. Resultados Entrevista 2	46
Tabla 9. Explicación Entrevista 2	46
Tabla 10. Plataforma Movil, Caso 1, Curso Normal de Eventos.....	47
Tabla 11. Plataforma Movil, Caso 1, Curso alterno de Eventos	48
Tabla 12. Plataforma Movil, Caso 2, Curso Normal de Eventos.....	48
Tabla 13. Plataforma Movil, Caso 3, Curso Normal de Eventos.....	49
Tabla 14. Plataforma Movil, Caso 4, Curso Normal de Eventos.....	50
Tabla 15. Plataforma Movil, Caso 5, Curso Normal de Eventos.....	51
Tabla 16. Plataforma web, Caso 1, Curso Normal de Eventos.....	52
Tabla 17. Plataforma Web, Caso 1, Curso Alterno de Eventos	52
Tabla 18. Plataforma Web, Caso 2, Curso Normal de Eventos	53
Tabla 19. Método POST.....	62
Tabla 20. Método GET.....	62
Tabla 21. Método PUT	63
Tabla 22. Método DELETE	63
Tabla 23. Pruebas Plataforma Movil	65
Tabla 24. Pruebas Plataforma Web.....	67

Introducción

En nuestra sociedad y principalmente en nuestro país existen diferentes medios de comunicación e información que son parte del diario vivir, los mismos que sirven para comunicarse por medio del lenguaje oral y escrito utilizando diferentes tecnologías y aplicaciones.

El uso del internet ha traído un sin número de aplicaciones, como lo es el Correo Electrónico, las redes sociales como Facebook, Twitter, y sitios web que facilitan la interrelación de la humanidad.

El Gobierno Autónomo Descentralizado de la Parroquia de Malacatos, tomando como referencia la práctica del internet y tecnologías en la actualidad, decidió implementar una plataforma que mejoraría el proceso de comunicación, logrando la interacción entre empleados y población en general.

Por tanto se realizó un estudio de las posibilidades y formas de implementarlo, se trabajó en un diseño de fácil acceso, así mismo se detalló los requerimientos manifestados por el personal de la Institución, realizando diferentes pruebas tanto en el diseño como en su funcionalidad.

De esta manera se entregó una herramienta tecnológica actual, llamada plataforma web y móvil, que involucra a los dispositivos móviles que en nuestro tiempo es de uso esencial en la persona y a las computadoras y portátiles en general.

Esta plataforma permite una mejor comunicación entre sus miembros de forma sencilla y segura, contribuyendo al adelanto del Gobierno Autónomo Descentralizado de la Parroquia de Malacatos.

Problematización

El problema más agravante que se ha podido observar y detectar en el GAD de Malacatos, es que existe poca comunicación entre el personal de la Institución, así mismo pocos medios informativos para la población de la Parroquia en general, por ello, es necesario crear las siguientes herramientas:

- Una plataforma móvil que permita mejor comunicación y viabilice los proyectos entre empleados, y administrativos en general.
- Una plataforma web que facilite informar de las obras realizadas por el GAD a todos los habitantes de la Parroquia de Malacatos.

Uno de los principales problemas del GAD de Malacatos es que no cuentan con un medio de comunicación con el público en general, y constantemente se debe realizar afiches y papelería en general, lo que representa un gasto para que la institución pueda mantener informada a toda la comunidad. La comunicación entre empleados y administrativos es insuficiente lo que impide que los proyectos e iniciativas progresen con rapidez.

Por lo expuesto anteriormente se puede concluir que es necesario desarrollar una plataforma tecnológica específica que contribuya a mejorar las falencias antes mencionadas del GAD de Malacatos, para lo cual se determinó implementar una plataforma móvil para facilitar la comunicación entre el personal de la Institución y una plataforma web informativa para que los habitantes se mantengan informados de los proyectos institucionales del GAD de Malacatos.

Por tal motivo esta plataforma se implementará con la mejor metodología que se base en tareas definidas y complementarias para realizarlas, esta modularidad permitirá un mejor entendimiento y cubrirá con todos los requerimientos de los funcionarios públicos.

Objetivos

Objetivo General.

- Desarrollar una plataforma web y móvil para la interrelación de la comunidad Malacatense y el Gobierno Autónomo Descentralizado Parroquial de Malacatos.

Objetivo Específico.

- Realizar el estado del arte sobre las tecnologías para dispositivos móviles y así determinar la mejor tecnología aplicada al presente proyecto.
- Investigar la metodología KANBAN para realizar la plataforma siguiendo etapas o fases.
- Diseñar una Plataforma Web, mediante un Sistema de Gestión de Contenidos CMS en WordPress.
- Desarrollar la plataforma para dispositivos móviles usando las tecnologías HTML5, CSS3 y JavaScript.
- Determinar el mejor Dominio y Hosting a utilizar en la Plataforma Web, por ser una Institución Pública.
- Desarrollar la plataforma para dispositivos móviles con el Sistema Android
- Comunicar las dos plataformas Web y Móvil para transmitir datos mediante un Servidor Web.
- Realizar Pruebas en el Diseño y Funcionalidad de las plataformas Web y Móvil.
- Capacitar al usuario final sobre el uso de la plataforma Web y Móvil.

Generalidades

En los antecedentes del proyecto, se consideró el perfil del anteproyecto aprobado anteriormente, la información del GAD Parroquial de Malacatos (Institución en donde se implementaron los sistemas), su historia e información general, y principalmente los requerimientos planteados de acuerdo a la metodología desarrollada.

FICHA TECNICA	
CONTENIDO	Marco teórico: Marco institucional Descripción: Misión Visión
PROBLEMAS	SOLUCIONES
Problemas: Existen problemas de comunicación e información entre empleados y la colectividad	Soluciones: Realizar plataformas tecnológicas con la información del Gad de Malacatos.
INFORMACIÓN ADICIONAL	
OBSERVACIONES:	La mayor parte de la información se consiguió mediante la entrevista.

Tabla 1. Ficha Técnica
 Autor: Nadia Narvaez Figueroa

CAPÍTULO 1

1. Marco Institucional

Con el propósito de mejorar la organización de la parroquia de Malacatos se resuelve crear un Gobierno Autónomo con su respectivo presidente y demás miembros del comité, los cuales son elegidos a nivel democrático por el pueblo Malacatense, con un principio de ayuda y colaboración hacia sus habitantes promoviéndoles de proyectos que involucran las principales falencias sobre todo en recursos vitales para su sobrevivencia, estructurando procesos cognitivos, estimulando la capacitación profesional y el desarrollo de actitudes positivas frente a la naturaleza y a los seres humanos, ofreciendo servicios a la colectividad y fortaleciendo su cultura y tradiciones en rescate de la identidad Malacatense. El GAD de la Parroquia de Malacatos está ubicado en las calles Miguel y Bolívar frente al parque central de Malacatos y diagonal a la Iglesia Parroquial

La Misión: Es nuestro deber gestionar, planificar y administrar los recursos del Gobierno Parroquial, direccionados a contribuir al mejoramiento del nivel de vida de nuestros ciudadanos y al cumplimiento de nuestras competencias otorgadas en la Constitución, de manera participativa con la comunidad y con otros niveles de gobierno, buscando la optimización de los recursos con transparencia y equidad, gestionar y dotar de la reglamentación necesaria para que nuestra parroquia pueda desarrollarse con igualdad de oportunidades para todos en un marco de justicia, orden y respeto.

El Gobierno Parroquial siempre está direccionado a conseguir que nuestra población tenga un nivel de organización adecuado, considerando que es la forma más efectiva de garantizar la equidad, para que toda nuestra población pueda tener acceso a los servicios básicos que deben ser de calidad, dotar de una buena infraestructura vial para asegurar que la enorme producción agropecuaria, ganadera y artesanal pueda ser transportadas de forma segura y ordenada a los centros de distribución.

La Visión es: Malacatos será un destino turístico posicionado a nivel nacional e internacional, por ser una parroquia que demuestra respeto al medio ambiente, mediante el manejo sostenible de los recursos naturales, contando con saneamiento ambiental adecuado, con un Plan de Desarrollo y Ordenamiento Territorial en plena ejecución y con una población con excelente educación que se refleja en el respeto y la práctica de sus valores, costumbres y tradiciones.

Actualmente el GAD de Malacatos cuenta con un presidente, un vicepresidente y 3 vocales. La infraestructura y el equipamiento se componen de oficinas administrativas, un centro de laboratorio de cómputo, una sala de reuniones y conferencias, y un salón social de eventos.

1.1 Fundamentos Teóricos

Se realizó un estudio de todos los conceptos básicos para la implementación de la Plataforma web y Plataforma móvil.

FICHA TÉCNICA	
CONTENIDO	Descripción.: Fundamentos de HTML5, Java Script, CSS3 Conceptos de Dispositivos Móviles Web Services con el Lenguaje de Programación PHP
PROBLEMAS	SOLUCIONES
Problemas: falta de información sobre el empaquetamiento de aplicaciones para que se adapten a diferentes Sistemas.	Soluciones: Pedir horas de asesoría a expertos en la materia
INFORMACIÓN ADICIONAL	
OBSERVACIONES:	Hacer Referencia al tipo de Smartphones que manejan los empleados del GAD de Malacatos para tener una correcta instalación y funcionamiento.

Tabla 2. Ficha Técnica (Fundamentos Generales)
 Autor: Nadia Narvaez Figueroa

1.2 Tecnologías

1.2.1 Tecnologías Móviles

1.2.1.1 HTML5

La versión HTML5 la menciona (Gauchat, 2012) como un nuevo concepto para la construcción de plataforma web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Anteriormente fue una simple versión de HTML propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información, como no tenía mucho alcance HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas simples inventos se transformaron en sofisticadas aplicaciones,

El lenguaje que se utiliza para escribir páginas web Según (MacDonal, 2011) es HTML. La idea básica de HTML es utilizar elementos de la estructura de su contenido en etiquetas, incluso las páginas web más antiguas todavía funcionan perfectamente bien en el la mayoría de los navegadores web modernos, también se lo conoce como lenguaje vivo, la actual W3C ha llevado a una disposición bastante inusual. Técnicamente, el W3C se encarga de determinar qué es y no es HTML5 oficial. Pero al mismo tiempo HTML siguen manteniéndose en su estructura intacta, conociéndolo como un lenguaje vivo. A continuación se muestra una ilustración citando algunos elementos:

Category	Elements
Semantic elements for structuring a page	<article>, <aside>, <figcaption>, <figure>, <footer>, <header>, <hgroup>, <nav>, <section> <details>, <summary>
Semantic elements for text	<mark>, <time> <wbr> (previously supported, but now an official part of the language)
Web forms and interactivity	<input> (not new, but has many new subtypes) <datalist>, <keygen>, <meter>, <progress>, <command>, <menu>, <output>
Audio, video, and plug-ins	<audio>, <video>, <source> <embed> (previously supported, but now an official part of the language)
Canvas	<canvas>
Non-English language support	<bdo>, <rp>, <rt>, <ruby>

Ilustración 1. Familia HTML5

Autor: (MacDonal, 2011)

Fuente: HTML5: The missing Manual

Las características de HTML5 que indica (Gauchat, 2012) son: estructura, estilo y funcionalidad, nunca fueron declaradas oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto que es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento.

Para trabajar y crear plataforma webs y aplicaciones con HTML5, se necesita saber primero cómo esa estructura es construida. Crear fundamentos fuertes nos ayudará más adelante a aplicar el resto de los componentes para aprovechar completamente estas nuevas tecnologías.

Un documento HTML es un archivo de texto. Si usted no posee ningún programa para desarrollo web, puede simplemente utilizar el Bloc de Notas de Windows o cualquier otro editor de textos. El archivo debe ser grabado con la extensión .html.

Para más información de HTML, revisar el Anexo 4, página 107.

La ilustración 2. Representa un diseño común dice (Gauchat, 2012) encontrado en la mayoría de los plataforma webs estos días. A pesar del hecho de que cada diseñador crea sus propios diseños, en general se podrá identificar las siguientes secciones en cada plataforma web:



Ilustración 2. Diseño Común web
Autor: (Gauchat, 2012)
Fuente: El Gran Libro HTML5, CSS3 y Javascript

En la parte superior, descrito como Cabecera, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del plataforma web o la página.

Inmediatamente debajo, se puede ver la Barra de Navegación en la cual casi todos los desarrolladores ofrecen un menú o lista de enlaces con el propósito de facilitar la navegación a través de la plataforma. Los usuarios son guiados desde esta barra hacia las diferentes páginas o documentos, normalmente pertenecientes a la misma plataforma web.

El contenido más relevante de una página web se encuentra, en casi todo diseño, ubicado en su centro. Esta sección presenta información y enlaces valiosos. La mayoría de las veces es dividida en varias filas y columnas. En el ejemplo de la Información Principal y Barra Lateral, pero esta sección es extremadamente flexible y normalmente diseñadores la adaptan acorde a sus necesidades insertando más columnas, dividiendo cada columna entre bloques más pequeños o generando diferentes distribuciones y combinaciones. El contenido presentado en esta parte del diseño es usualmente de alta prioridad.

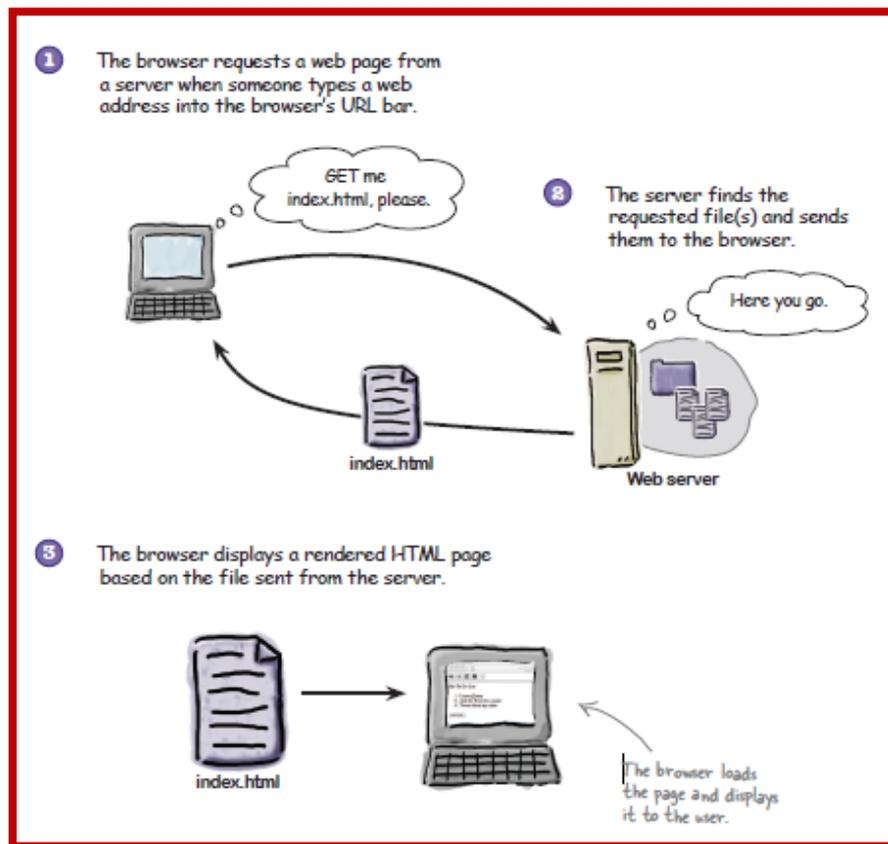


Ilustración 3 Funcionamiento Básico
 Autor (Cranley, 2011)
 Fuente: Head First JQuery

1.2.1.2 CSS y HTML

La nueva especificación de HTML (HTML5) menciona (Gauchat, 2012) no describe solo los nuevos elementos HTML o el lenguaje mismo. En este nuevo paradigma, HTML se presenta junto con CSS y JavaScript como un único instrumento integrado.

La función de cada tecnología ya ha sido explicada en capítulos previos, así como los nuevos elementos HTML responsables de la estructura del documento. Ahora es momento de analizar CSS, su relevancia dentro de esta unión estratégica y su influencia sobre la presentación de documentos HTML.

CSS nada tiene que ver con HTML5 describe (Gauchat, 2012) CSS no es parte de la especificación y nunca lo fue. Este lenguaje es, de hecho, un complemento

desarrollado para superar las limitaciones y reducir la complejidad de HTML. CSS pronto fue adoptado como la forma de separar la estructura de la presentación, pero siempre desarrollado en paralelo, enfocado en las necesidades de los diseñadores y apartado del proceso de evolución de HTML. CSS es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc.

A pesar de que cada navegador garantiza estilos por defecto para cada uno de los elementos HTML, estos estilos no necesariamente satisfacen los requerimientos de cada diseñador. Normalmente se encuentran muy distanciados de lo que queremos para nuestras plataformas webs. Diseñadores y desarrolladores a menudo deben aplicar sus propios estilos para obtener la organización y el efecto visual que realmente desean.

Para aprender cómo podemos crear nuestra propia organización de los elementos en pantalla, debemos primero entender cómo los navegadores procesan el código HTML. Los navegadores consideran cada elemento HTML como una caja. Una página web es en realidad un grupo de cajas ordenadas siguiendo ciertas reglas. Estas reglas son establecidas por estilos provistos por los navegadores o por los diseñadores usando CSS.

Para más información de CSS ver el Anexo 4, página 113.

1.2.1.3 JavaScript

JavaScript es un lenguaje de programación que le permite impulsar el potencial de HTML en animación, interactividad y efectos visuales dinámicos dice (McFarland, 2011). JavaScript puede hacer que las páginas web sean más útiles mediante el suministro de información inmediata. Por ejemplo, una página de compras JavaScript puede mostrar inmediatamente un costo total, con impuestos y envío, en el momento que un visitante selecciona un producto para comprar, JavaScript además muestra un mensaje de error inmediatamente cuando alguien intenta enviar un formulario web incompleto

JavaScript también permite crear divertidas, dinámicas interfaces interactivas. Por ejemplo, con el lenguaje Java, puede transformar una página estática, en una animada. O usted puede hacer algo más sutil, donde el usuario final tenga a un solo clic a lo que necesite sin causar confusión en el cliente. También se puede añadir algo útil y atractivo, como información sobre herramientas emergentes que proporcionan información adicional para los artículos en su página web.

JavaScript es un lenguaje interpretado menciona (Gauchat, 2012) usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código.

La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web.

Existen tres técnicas para incorporar código JavaScript dentro de HTML según (Gauchat, 2012). Sin embargo, al igual que en CSS, solo la inclusión de archivos externos es la recomendada a usar en HTML5.

Los manejadores de eventos más usados son, en general, los relacionados con el ratón, como por ejemplo onclick, onMouseOver, u onMouseOut. Sin embargo, se encontrará plataformas web que implementan eventos de teclado y de la ventana, ejecutando acciones luego de que una tecla es presionada o alguna condición en la ventana del navegador cambia (por ejemplo, onload u onfocus).

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este texto es el título del documento</title>
</head>
<body>
```

```
<div id="principal">  
<p onclick="alert('hizo clic!')">Hacer Clic</p>  
<p>No puede hacer clic</p>  
</div>  
</body>  
</html>
```

El uso de JavaScript dentro de etiquetas HTML está permitido en HTML5, pero por las mismas razones que en CSS, esta clase de práctica no es recomendable. El código HTML se extiende innecesariamente y se hace difícil de mantener y actualizar. Así mismo, el código distribuido sobre todo el documento complica la construcción de aplicaciones útiles.

Para más información de Java Script ver el Anexo 4 página 119.

1.2.2 Tecnologías Web

1.2.2.1 WordPress

WordPress es una actual plataforma que se utiliza para la creación de páginas o plataformas web, presenta una variedad de diseños y modelos para la realización de las mismas, además facilita en tiempo la creación y publicación de las páginas web, ya que posee dos formas de edición la visual y mediante código, acoplándose de esta manera a la persona que haga uso de esta herramienta.

Para explorar esta plataforma visite **wordpress.com**

Para comenzar con WordPress primero se debe registrar con el dominio a utilizar en la página web propuesta.

Ilustración 4. Inicio de WordPress
Fuente Wordpress.com
Autora: Nadia Narvaez Figueroa

Para publicar una entrada sencilla son los siguientes:

1. Selecciona un título para la entrada.
2. En la caja de texto comienza a escribir el mensaje. Puedes añadir enlaces, negritas, sangrías, entre otros.
3. Añade las etiquetas y categorías que definen a esta entrada.

Aquí se muestra el escritorio general de WordPress para editar tu página Web

Ilustración 5. Escritorio de WordPress
Autora: Nadia Narvaez Figueroa
Fuente Wordpress.com

Así mismo si se añade una entrada se puede seleccionar la fecha en que aparecerá publicada.

WordPress nos permite insertar imágenes y videos además de textos, comentarios, logearse, etc. Simplemente se procede a editar y luego se pulsa Publicar para su visualización en la Web.



Ilustración 6. Publicaciones WordPress
Autora: Nadia Narvaez Figueroa
Fuente WordPress.com

Tienes muchas opciones para y funciones más avanzadas para editar una página web se muestran las siguientes:

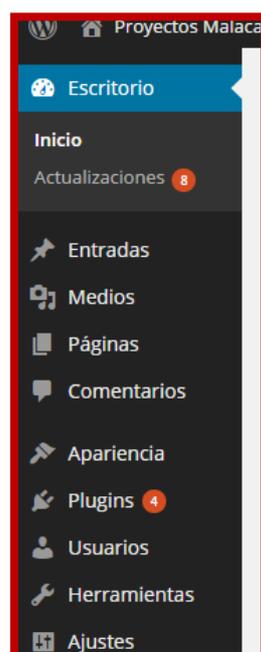


Ilustración 7. Funciones del Escritorio
Autora: Nadia Narvaez Figueroa
Fuente WordPress.com

WordPress actualmente es una de los mejores CMS (administradores de contenido), ya que no que permite crear blogs gratuitos y páginas web en general mediante Servidores, posee una plataforma con una interfaz anímica para ser utilizado por cualquier persona, así mismo propone temas gratuitos y pagados según el gusto y beneficio de sus clientes.

1.2.2.2 Lenguaje De Programación PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto menciona el (Olson, 2015) muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Ejemplo</title>
</head>
<body>
<?php
echo ";Hola de PHP!";
?>
</body>
</html>
```

PHP facilita la opción de usar demasiados comandos como se lo hace en html, utilizando código incrustado lo señala (Olson, 2015). Del lado del cliente PHP (JavaScript) ejecuta el código desde el servidor, produciendo HTML y enviando así al cliente, el mismo recibirá el resultado. El servidor web puede ser configurado para que procese todos los ficheros HTML con PHP, por tal motivo los usuarios no sabrán que se tiene. La ventaja de PHP se refleja en su fácil utilización para principiantes y para programadores avanzados las grandes características que presenta.

1.2.2.3 Framework SLIM

Slim es un micro framework PHP para crear aplicaciones REST con una sintaxis muy simple lo indica en (BARABASZ, 2013). Desde el router asociamos cierto pedido HTTP con funciones anónimas, de esta manera:

```
$app = new \Slim\Slim ();
// Asociamos una URL a una función deduciendo el parámetro name
$app->get('/hello/:name', function ($name) {
    echo "Hello, $name";
});
// Ponemos en marcha el router
$app->run();
```

Al ingresar en /hello/world se ejecutará la función y entonces se verá en pantalla el mensaje.

El framework Slim, tiene las siguientes características según (Lockhart, 2011):

- Enrutador Potente
 - Métodos HTTP estándar y personalizados
 - Parámetros de la ruta con comodines y condiciones
 - Redirección de ruta, detenerse, y pase
 - Middleware Ruta
- Prestación plantilla con vistas personalizadas
- Los mensajes flash
- Galletas seguras con cifrado AES-256
- HTTP caché
- Inicio de sesión con los escritores de registro personalizados
- El manejo de errores y depuración.
- Middleware y el gancho de la arquitectura
- Configuración sencilla

Existen pasos generales en el Framework Slim, indica (Lockhart, 2011): Primero requiere el Marco Delgado para PHP 5 en el archivo index.php. Se Cambia la ruta del archivo si es necesario.

Se crea una instancia de una aplicación de slim. Proporcionar un conjunto opcional de ajustes para configurar la aplicación.

```
// Con los ajustes predeterminados
$ Aplicación = nuevo Slim ();
// Con los ajustes personalizados
$ Aplicación = nuevo Slim (array ( 'Log.enable' => true, 'log.path' => './logs', 'Log.level' => 4,
'Vista' => 'MyCustomViewClassName' ));
```

Se Define las rutas de la aplicación de los métodos que se muestran en el siguiente ejemplo.

```
// Ruta GET
$ App-> get ('/ hola /: nombre', function ($ nombre) {
echo "Hola $ name";
});
// Ruta POSTE
$ App-> post ('/ persona', function () {
// Crear nueva persona
});

// Ruta PUT
$ App-> put ('/ persona /: id', function ($ id) {
// Actualización persona identificado por $ id
});

// Borrar ruta
$ App-> delete ('/ persona /: id', function ($ id) {
// Eliminar persona identificado por $ id
});
```

Se ejecuta el marco delgado para php 5 aplicación web. Esta suele ser la declaración final ejecutado en el archivo index.php.

```
$ App-> run ();
```

CAPÍTULO 2.

2. Estado del Arte

En este capítulo se analizara tres framework para desarrollo de las plataformas como son:

2.1 jQuery Mobile

Según (Master Univ.)jQuery es un framework desarrollado originalmente por John Resig que replantea totalmente el valor y la forma de escribir JavaScript. Debido a su potencia y a su sintaxis intuitiva ha ganado una enorme popularidad en los últimos años, y en la actualidad es utilizado en multitud de páginas Web (utilicen o no AJAX). Puede ser considerado como la referencia en materia de frameworks JavaScript. Es fruto de un intenso desarrollo, dando lugar a sucesivas versiones, y existe una amplia comunidad de usuarios y una gran cantidad de documentación sobre las funciones que incorpora.

2.1.1 Selección de Elementos

El concepto más básico de jQuery es el de “seleccionar algunos elementos y realizar acciones con ellos “señala (Murphey, 2011). La biblioteca soporta gran parte de los selectores CSS3 y varios más no estandarizados.

Este es el fragmento de código que utiliza jQuery para determinar cuándo un elemento es visible o no. Se incorporaron los comentarios para que quede más claro su entendimiento:

```
jQuery.expr.filters.hidden = function( elem ) {  
var width = elem.offsetWidth, height = elem.offsetHeight,  
skip = elem.nodeName.toLowerCase() === "tr";  
  
// ¿el elemento posee alto o, ancho o y no es un <tr>?
```

```

return width === 0 && height === 0 && !skip ?

// entonces debe estar oculto (hidden)
true :

// pero si posee ancho y alto
// y no es un <tr>
width > 0 && height > 0 && !skip ?

// entonces debe estar visible
false :

// si nos encontramos aquí, es porque el elemento posee ancho
// y alto, pero además es un <tr>,
// entonces se verifica el valor del estilo display
// aplicado a través de CSS
// para decidir si está oculto o no
jQuery.curCSS(elem, "display") === "none";
};

jQuery.expr.filters.visible = function( elem ) {
return !jQuery.expr.filters.hidden( elem );
};

```

2.1.2 Conceptos Clave: GET vs. POST

Los dos métodos HTTP más comunes para enviar una petición a un servidor son GET y POST menciona (Murphey, 2011). Es importante entender la utilización de cada uno.

El método GET debe ser utilizado para operaciones no-destructivas es decir, operaciones en donde se está “obteniendo” datos del servidor, pero no modificando. Por ejemplo, una consulta a un servicio de búsqueda podría ser una petición GET. Por otro lado, las solicitudes GET pueden ser almacenadas en la cache del navegador, pudiendo conducir a un comportamiento impredecible si no se lo espera. Generalmente, la información enviada al servidor, es enviada en una cadena de datos (en inglés *query string*).

El método POST debe ser utilizado para operaciones destructivas es decir, operaciones en donde se está incorporando información al servidor. Por ejemplo,

cuando un usuario guarda un artículo en un blog, esta acción debería utilizar POST. Por otro lado, este tipo de método no se guarda en la cache del navegador. Además, una cadena de datos puede ser parte de la URL, pero la información tiende a ser enviada de forma separada.

(Alvarez, 2012) indica el slogan de jQuery: "Write Less, Do More". Pues jQuery Mobile Framework es esa misma idea, pero elevado a la siguiente potencia. Se destaca lo siguiente:

1. No tener que lidiar con las particularidades de cada navegador. Desarrollar una vez con código jQuery y que se vea correctamente en todos los navegadores del mercado. Incluso, cuando saquen otro navegador, o versiones nuevas de los existentes, que no tengas que retocar tu código para adaptarlo también a ellos.
2. También sirve para escribir menos código fuente y hacer cosas más espectaculares.

Tomando en cuenta:

1. Que con los dispositivos móviles se han multiplicado el número de navegadores y de plataformas. Tenemos muchos fabricantes, de tablets y smartphones y diversos dispositivos con características distintas, como tamaños de pantallas, sistemas operativos diferentes y diversos navegadores basados en cada uno de ellos. Vamos, que si antes con los navegadores para PCs había problemas de compatibilidad, cuando teníamos apenas 3 sistemas operativos y 3 navegadores populares, ahora con los móviles la cosa todavía se hace más complicada.
2. Porque el desarrollo de plataformas web con jQuery Mobile es todavía más automático de lo que era trabajar con jQuery. Con mucho menos código haces muchas más cosas.

Los creadores del framework comentan entre sus características que se han esforzado para cubrir el mayor número de plataformas de dispositivos móviles posible.

Dicen que el target que han buscado es mayor que el del resto de frameworks disponibles en el mercado.



Ilustración 8. Plataformas Móviles

Autor: Anónimo

Fuente (GOMEZ, 2013)

2.2 Phonegap

PhoneGap es un framework para el desarrollo de aplicaciones móviles señala (Rodríguez, 2011) que posibilita a los desarrolladores a que implementen sus proyectos utilizando las tecnologías estándar de web: HTML5, CSS3 y JavaScript, y esta herramienta las convierte a aplicaciones híbridas, es decir, aplicaciones nativas de las diferentes plataformas móviles existentes que tienen acceso a gran parte del API nativo.

PhoneGap es una propuesta de licencia OpenSource (código abierto), basada en JavaScript, que permite crear aplicaciones para dispositivos móviles utilizando HTML5 yCSS3.

Actualmente este framework soporta seis de las principales plataformas móviles del mercado: iOS, Android, Blackberry, PalmOS, Windows Mobile y Symbian.

Existen al día de hoy muchos tutoriales y documentación para poder a utilizar PhoneGap en los desarrollos de aplicaciones móviles que requieran correr en varias plataformas.

2.3 Sencha Touch

Sencha touch es un framework JavaScript para construir aplicaciones web indica (Clark, 2012) para IOS y Android muy orientado a pantallas táctiles y con apariencia de aplicación nativa.

Sencha Touch tiene una curva de aprendizaje bastante grande, se construyen las aplicaciones web casi completamente con JavaScript, y esto es un cambio bastante grande para los que estamos acostumbrados a programar con html, aquí se muestra completamente vacío y será el framework el que se ocupará de llenarlo con html5 en función de lo que hagas con el JavaScript.

2.3.1 Características de Sencha Touch

- Al ser hermano de ExtJS usando el core del mismo, ya contempla una robustez tremendamente amplia.
- Su forma de uso es muy similar a ExtJS. En caso de conocer y trabajar con ExtJS considero que te será sencillo trabajar con Sencha Touch.
- Así como en ExtJS, se escribe muchísimo. Esto podría ser tomado como un pro y como un contra. Es bueno porque escribimos mucho ya que contiene demasiadas opciones útiles que podemos usar, mientras que lo malo es que comparándolo con jQuery Mobile, “realmente se escribe mucho”.
- Contiene temas para el diseño de la interfaz muy vistosos y de simple configuración.
- Tiene un API muy completa como documentación.
- Basado en HTML 5
- Efectos de transición entre páginas.

Se ha resumido conceptos generales de cada Framework tomando en cuenta algunas de sus ventajas y desventajas.

2.4 Ventajas de Los Tres Frameworks

jQuery Mobile	PhoneGap	Sencha Touch
<p>Los autores (VILLA, 2012) (NAVAJAS, 2015) indican:</p> <ul style="list-style-type: none"> • Conocido por proporcionar una serie de Widgets prediseñados que nos ayudan a montar interfaces y elementos de UI. • Consumen por norma general muchos más recursos que el desarrollo de un equivalente en código JS+HTML+CSS. • Soporta iOS, Android, Windows, Phone, BlackBerry, Symbian, Palm y más dispositivos. • Cuenta con un sistema de bloques ideal para construir plantillas. • Es muy fácil de aprender y utilizar. Si se tienen adecuados conocimientos de HTML y CSS. • Tiene soporte para Ajax, una tecnología para crear aplicaciones dinámicas que se ejecuten del lado cliente. 	<p>Los autores (VILLA, 2012) (NAVAJAS, 2015) indican:</p> <ul style="list-style-type: none"> • Se pueden ejecutar las aplicaciones en nuestro navegador web, sin depender de un simulador dedicado a esta tarea, y por otra parte, tenemos la posibilidad de soportar funciones sobre frameworks como Sencha Touch o JQuery Mobile. • Soporta los estándares HTML5, CSS3 y JavaScript. • El framework soporta iPhone, Android, Blackberry, Symbian y Palm. • Codifica una vez, distribuye en muchas plataformas. 	<p>Los autores (VILLA, 2012) (NAVAJAS, 2015) indican:</p> <ul style="list-style-type: none"> • Conocido por proporcionar una serie de Widgets prediseñados que nos ayudan a montar interfaces y elementos de UI. • Consumen por norma general muchos más recursos que el desarrollo de un equivalente en código JS+HTML+CSS. • Soporta Android, Apple iOS • Al ser basado en la web, permite hacer cambios sin requerir una actualización por parte del usuario. • Puede ser "transformado" en una aplicación nativa usando un wrapper como PhoneGap fácilmente • Documentación extensiva. • Es un framework maduro con muchas funcionalidades.

Tabla 3. Ventajas de los Frameworks (jQuery Mobile, Phonegap y Sencha Touch)

Autor: Nadia Narvaez Figueroa

Fuente: Referencia Web Antonio Navajas

2.5 Desventajas De Los Tres Frameworks

jQuery Mobile	PhoneGap	Sencha Touch
<ul style="list-style-type: none"> • Según (DIMAGGIO, 2013) Como es necesario invocar a un archivo para utilizar sus funciones, ralentiza levemente la carga de la página. • Su manejo de CSS suele resultar innecesariamente complejo. A veces cuesta saber qué clases utilizar. • No existen muchas plantillas prediseñadas sobre las cuales empezar a construir nuestra plataforma. 	<ul style="list-style-type: none"> • Según (VILLA, 2012) Dado que PhoneGap está diseñado para soportar diferentes plataformas, no contará con las características nuevas de cada plataforma tan pronto como estas se publiquen. • Muchas de las características core de las plataformas podrán no estar disponibles. • Como utiliza JavaScript, html y css, existen potenciales problemas de compatibilidad con browsers 	<ul style="list-style-type: none"> • Licencia dual: para proyectos no OpenSource tiene una licencia comercial. Actualmente esa licencia no tiene costo, pero eso no quiere decir que vaya a ser siempre así.

Tabla 4. desventajas de los Frameworks (jQuery Mobile, Phonegap y Sencha Touch)

Autor: Nadia Narvaez Figueroa

Fuente: Referencia Web Antonio Navajas

Según este estudio de los frameworks para el empaquetar el aplicativo móvil, he decidido usar JQuery Mobile y PhoneGap.

2.6 Metodología de Desarrollo Kanban

Durante las pasadas tres décadas, Kanban, que se define como “un sistema de producción altamente efectivo y eficiente” dice (Gilibets, Comunidad IEBS, 2013), ha contribuido a generar un panorama manufacturero óptimo y competitivo. El origen de la metodología Kanban debemos buscarlo en los procesos de producción “just-in-time”

(JIT) ideados por Toyota, en los que se utilizaban tarjetas para identificar necesidades de material en la cadena de producción.

Kanban ha pasado a formar parte de las llamadas metodologías ágiles, cuyo objetivo es gestionar de manera general cómo se van completando las tareas. Kanban es una palabra japonesa que significa “tarjetas visuales”, donde Kan es “visual”, y Ban corresponde a “tarjeta”.

El método Kanban permite sincronizar las etapas menciona (Gilibets, Comunidad IEBS, 2013) sistematizando el proceso de producción logrando:

- Disminuir o eliminar los stocks intermedios (entre procesos).
- Cumplir los tiempos de entrega demandados por el cliente.
- Mejorar la calidad del producto por una mejor detección de los defectos del mismo.
- Evitar el manejo excesivo de materiales.
- Facilitar el control de la producción.
- Obtener un sistema de producción flexible según la demanda.

Esta forma de administrar la producción es muy distinta a los sistemas tradicionales que centraban la atención en el proceso de fabricación mismo, sin tener en cuenta las necesidades y tiempos del mercado, en la creencia de que lo eficiente era que las máquinas estuvieran ocupadas produciendo el mayor tiempo posible, ignorando el costo de acumulación de stocks inmovilizados.

2.6.1 Los principios de la metodología Kanban

La metodología Kanban se basa en una serie de principios indica (Gilibets, Comunidad IEBS, 2013) que la diferencian del resto de metodologías conocidas como ágiles:

Calidad garantizada	Todo lo que se hace debe salir bien a la primera, no hay margen de error. De aquí a que en Kanban no se premie la rapidez, sino la calidad final de las tareas realizadas.
Reducción del desperdicio	Kanban se basa en hacer solamente lo justo y necesario, pero hacerlo bien. Esto supone la reducción de todo aquello que es superficial o secundario.
Mejora continua	Kanban no es simplemente un método de gestión, sino también un sistema de mejora en el desarrollo de proyectos, según los objetivos a alcanzar.
Flexibilidad	Lo siguiente a realizar se decide del <i>backlog</i> (o tareas pendientes acumuladas), pudiéndose priorizar aquellas tareas entrantes según las necesidades del momento (capacidad de dar respuesta a tareas imprevistas).

Tabla 5. Principios Kanban
Fuente (Gilibets, Comunidad IEBS, 2013)
Autora: Nadia Narvaez Figueroa

2.6.2 Pasos para configurar tu estrategia de Kanban

La aplicación del método Kanban implica según (Gilibets, Comunidad IEBS, 2013) la generación de un tablero de tareas que permitirá mejorar el flujo de trabajo y alcanzar un ritmo sostenible. Para implantar esta metodología, deberemos tener claro los siguientes aspectos:

- 1. Definir el flujo de trabajo de los proyectos:** para ello, simplemente deberemos crear nuestro propio tablero, que deberá ser visible y accesible por parte de todos los miembros del equipo. Cada una de las columnas corresponderá a un estado concreto del flujo de tareas, que nos servirá para saber en qué situación se encuentra cada proyecto.
- 2. Visualizar las fases del ciclo de producción.** Kanban se basa en el principio de desarrollo incremental, dividiendo el trabajo en distintas partes. Esto significa que no hablamos de la tarea en sí, sino que lo dividimos en distintos pasos para

agilizar el proceso de producción. Al final, el objetivo de la visualización es clarificar al máximo el trabajo a realizar.

3. **Stop Starting, start finishing.** Este es el lema principal de la metodología Kanban. De esta manera, se prioriza el trabajo que está en curso en vez de empezar nuevas tareas. Precisamente, una de las principales aportaciones del Kanban es que existe un número máximo de tareas a realizar en cada fase.

4. **Control del Flujo.** la metodología Kanban mezcla tareas y proyectos. Se trata de mantener a los trabajadores con un flujo de trabajo constante, las tareas más importantes en cola para ser desarrolladas para no tener que interrumpir al trabajador en cada momento. Asimismo, dicha metodología de trabajo nos permite hacer un seguimiento del trabajo realizado, almacenando la información que nos proporcionan las tarjetas.

2.6.3 Las tres reglas de Kanban

Con tan solo tres simples reglas dice (Bahit, 2011), Kanban demuestra ser una de las metodologías adaptativas que menos resistencia al cambio presenta. Dichas reglas son:

1. Mostrar el proceso
2. Limitar el trabajo en curso
3. Optimizar el flujo de trabajo

2.6.4 Tablero Kanban

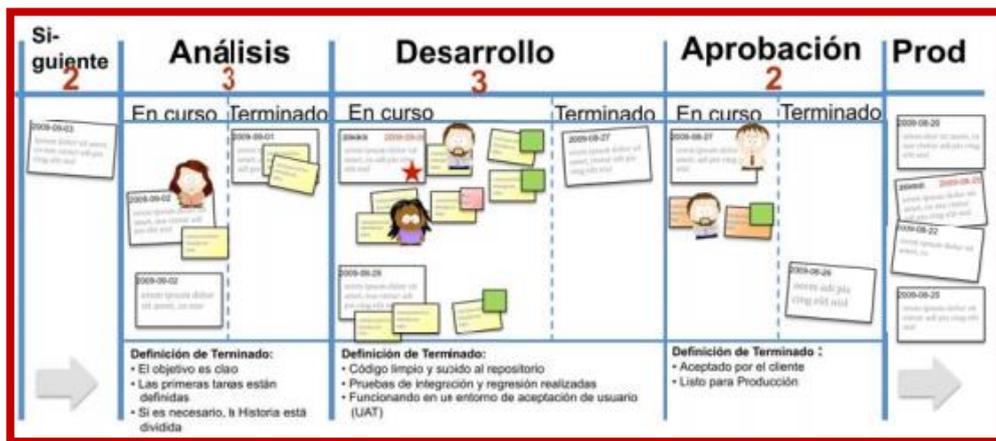


Ilustración 9. Tablero Kanban
Autor: Skarin, Henrik Kniberg y Mattias
Fuente Kanban y Scrum

En los tableros Kanban se puede trabajar con algunos productos o temas para diferenciarlos por los colores en las tareas a realizarse indica (Skarin, 2010). Kanban es un sistema de planificación tipo "Pull", en donde se elige cuando y cuanto trabajo a realizar, se asemeja a una impresora que solo imprime la siguiente página cuando acaba la anterior, utiliza procesos de optimización continuos y empíricos. Kanban es que el flujo de trabajo debe ser visual, y que el trabajo en curso debe estar limitado. El objetivo es crear un flujo suave a través del sistema y minimizar el tiempo de entrega.

CAPÍTULO 3

3. Análisis y Diseño

El GAD Parroquial de Malacatos es una entidad Pública que se encarga de administrar y gobernar la comunidad Malacatense, para lo cual es necesario reforzar la comunicación entre los mismos.

El GAD de Malacatos cuenta con una directiva en que son; Presidente, Vicepresidente y tres vocales, pero a estos se suman otros actores que vendrían a ser los que manejarían las plataformas.

3.1 Actores:

1. Presidente
2. Secretaria
3. Contadora
4. Vicepresidente
5. Vocal 1
6. Vocal 2
7. Vocal 3

3.2 Ámbito del Sistema

En el presente trabajo de Tesis se realizará dos plataformas una Web y una Móvil, las mismas que se comunicara con un WebServices para crear, actualizar, presentar y eliminar proyectos, así mismo por separado tiene diferentes funciones e información.

Estos sistemas facilitan la comunicación entre empleados y para la comunidad en general

3.3 Diagrama de Casos de Uso

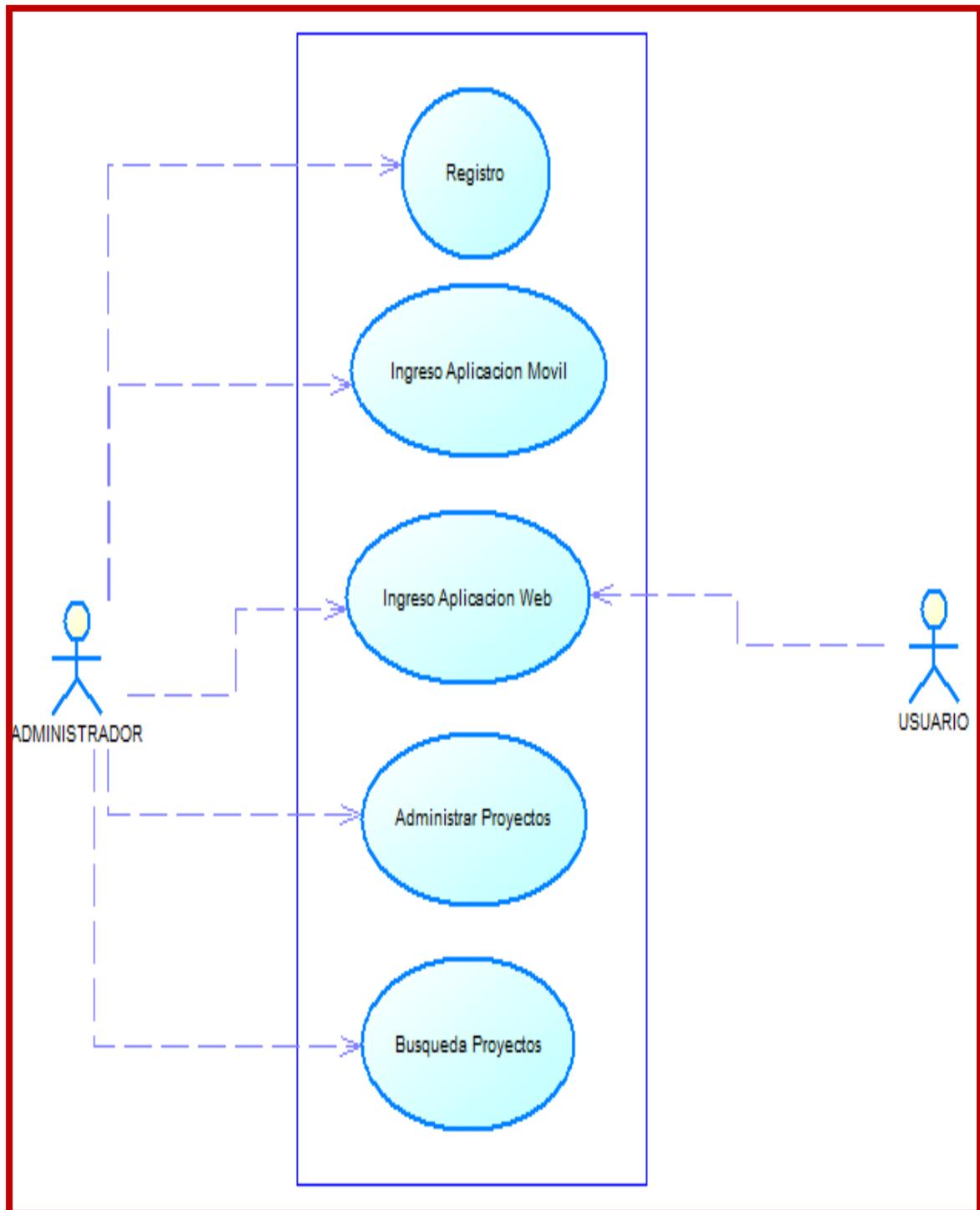


Ilustración 10. Diagrama de Caso de Uso }
Autora: Nadia Narvaez Figueroa

3.4 Partes del Sistema

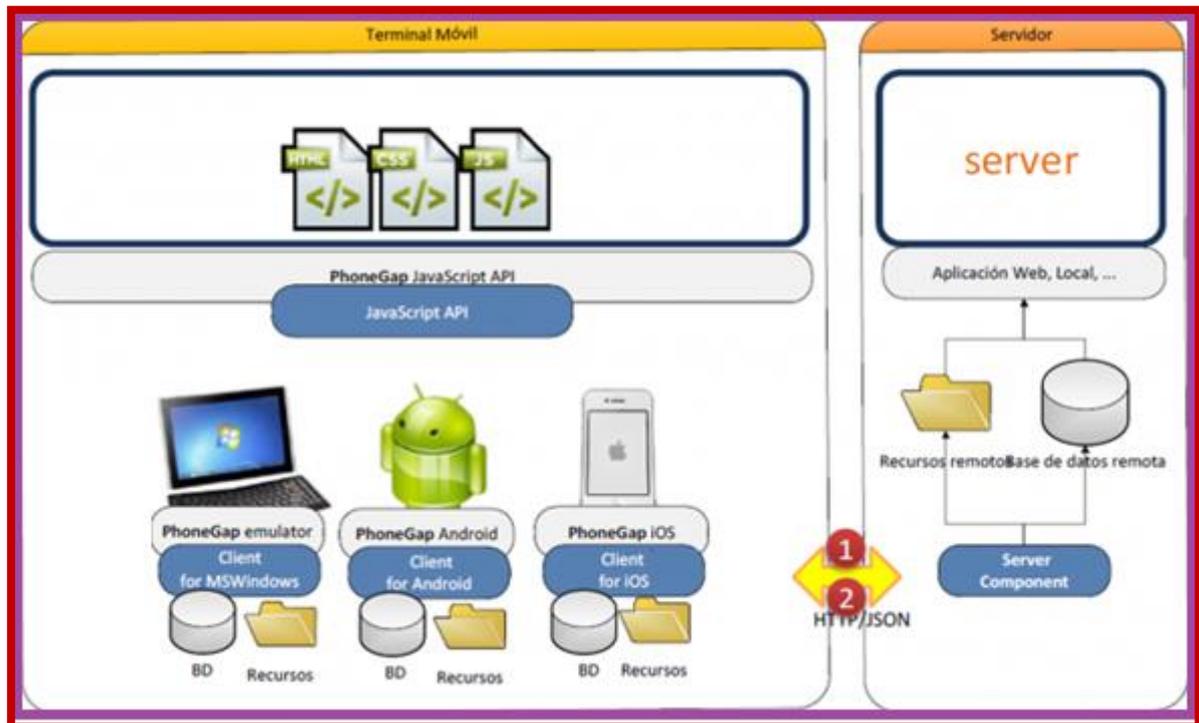


Ilustración 11. Partes del Sistema
 Autora: Nadia Narvaez Figueroa

En esta imagen se aprecian 2 partes generales la del Terminal Móvil y el Servidor, en cuanto a la primera se trabaja con HTML5, CSS3 y JavaScript para múltiples plataformas con una sola base de código usando las API'S estándar de la web, así mismo La aplicación está desarrollada con PhoneGap y JQuery que a la vez.

Por tanto la aplicación cliente con PhoneGap se comunica con una aplicación en el servidor para recibir/enviar datos y la aplicación en el servidor gestiona la lógica de negocio y se comunica con la Base de Datos, pero se debe tener presente que la aplicación no se comunica directamente con una base de datos. Esta comunicación es gestionada a través de una aplicación en el servidor. La comunicación con el servidor se basa en peticiones HTTP standard para contenido HTML, como RESTFUL, JSON.

3.5 Diseño de la Base de Datos

Esta desarrollada en MySQL

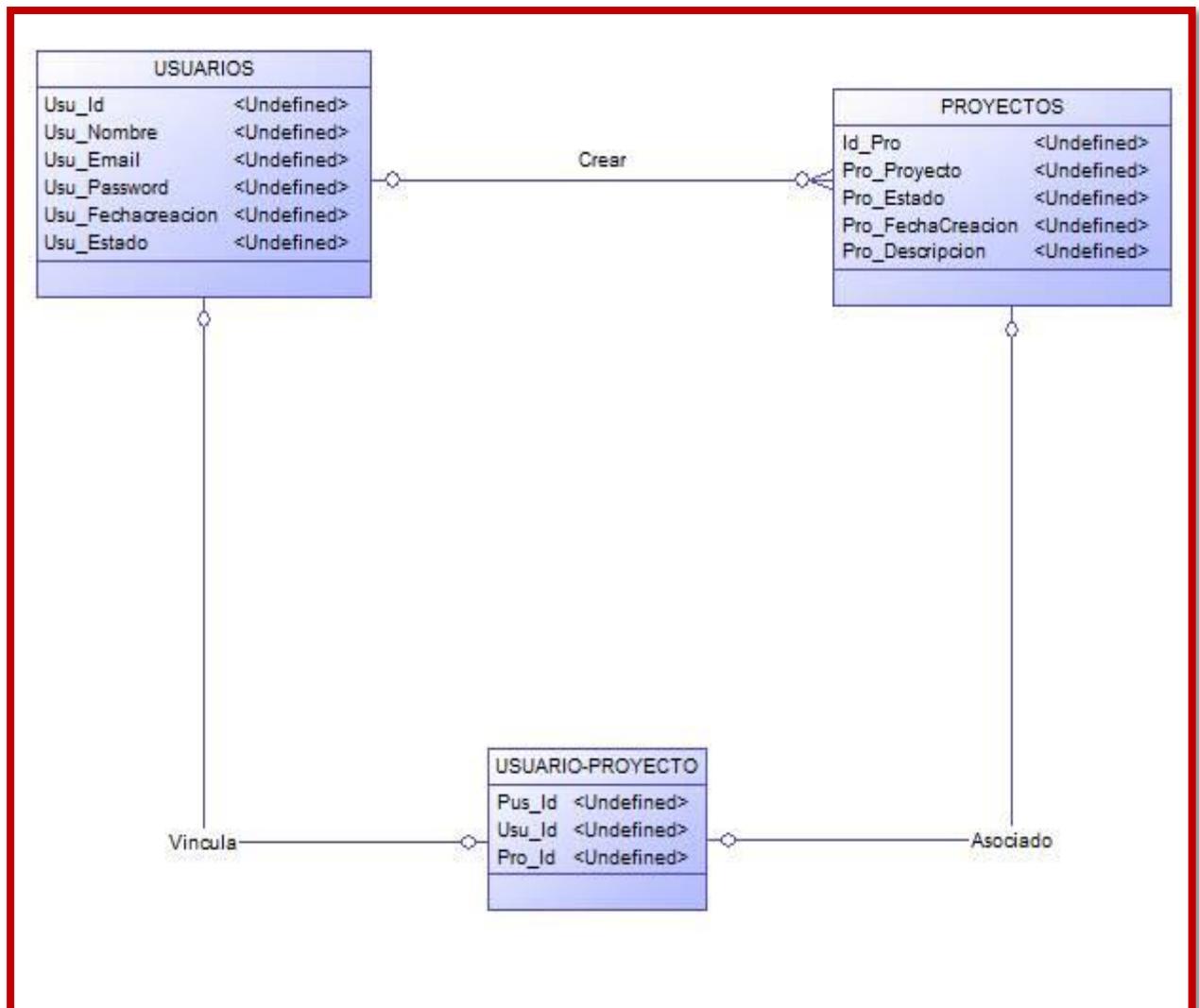


Ilustración 12. Esquema de base de Datos
 Autora: Nadia Narvaez Figueroa

Aquí se observa las siguientes tablas:

3.5.1 Tabla Usuarios

Esta tabla se utiliza para registrar usuarios, se requiere algunos campos como: Id de la tabla, Nombre, email, password, fecha creación y estado.

3.5.2 Tabla Proyectos

En esta tabla nos referimos a los proyectos que gestiona y administra el GAD de Malacatos y tiene los siguientes campos como: Id de la tabla, proyecto, estado, fecha de creación, descripción.

3.5.3 Tabla Usuario- Proyecto

Esta tabla es para asociar y vincular a los proyectos con los usuarios es decir las dos tablas antes mencionadas, debemos tomar en cuenta que un usuario debe registrarse y luego que este logeado puede administrar y crear los proyectos.

3.6 Administración De Requerimientos

3.6.1 Entrevistas

ENTREVISTA 1

REALIZADA A: María Augusta Ojeda

RESPONSABLE: Nadia Narváez

1. Importancia de la plataforma web denominado Proyectos Malacatos .com
Actualmente es importante mantener un plataforma web en toda institución ya que es uno de los medios más inmediatos para informarse según nuestros intereses.
2. Importancia de mantener una plataforma móvil para modificar proyectos
El celular es lo más rápido y parte personal de las personas por eso es importante que desde la comodidad de su teléfono e independientemente del lugar que se encuentre posea la facilidad de avisar y modificar proyectos.

<p>3. En el plataforma Web que función tiene el chat</p> <p>El chat es una forma de comunicarnos y la manera más rápida es mediante la página web ya que nos evita entrar a otros correos y manejar todo mediante la página o plataforma web.</p> <p>4. En el plataforma Web también modifica los proyectos ¿cree que es importante?</p> <p>Si porque muchas veces tenemos inconvenientes con los planes de internet en los teléfonos y desde la página web podremos ver y modificar los mismos y sobre todo comunicar al presidente desde las oficinas de GAD de Malacatos.</p>
--

Tabla 6. Resultados de Entrevista 1

Autora: Nadia Narvaez Figueroa

<p>Aplicada a : María Augusta Ojeda Tesorera del GAD de Malacatos</p> <p>Propósito: Recoger información que permita identificar requerimientos de las plataformas a desarrollar,</p> <p>Objetivos: Determinar las pantallas que se deben realizar, determinar la seguridad y rapidez del sistema</p> <p>Instrucciones: Ser breve en la entrevista para evitar cansancio, creando un clima de confianza</p>
<p>OBSERVACIONES:</p>
<p>El entrevistado sugiere que se establezca una plataforma móvil y una web para manejar proyectos.</p>

Tabla 7. Explicacion Entrevista 1

Autora: Nadia Narvaez Figueroa

ENTREVISTA 2

REALIZADA A: Sandra Rodríguez

RESPONSABLE: Nadia Narváez

1. Importancia de las plataformas Web y Móvil

Es muy importante ya que debemos ir acorde a la tecnología de la actualidad y de esa manera acostumbramos a nuestros empleados a trabajar con tecnología

2. ¿Quién maneja las plataformas móviles y la web?

Todos los empleados del GAD manejarán la plataforma móvil con un usuario y contraseña y web mientras que la comunidad únicamente la plataforma web

3. ¿si posee la tecnología para la implementación?

Si lo tenemos pero se debería realizar una capacitación del uso de las plataformas.

Tabla 8. Resultados Entrevista 2
Autora: Nadia Narvaez Figueroa

Aplicada a : Sandra Rodríguez- Presidenta del GAD de Malacatos

Propósito: Recoger información que permita identificar requerimientos de las plataformas a desarrollar y la necesidad de la misma

Objetivos: Determinar los módulos que llevara el sistema

Instrucciones: Ser breve en la entrevista para evitar cansancio, creando un clima de confianza

OBSERVACIONES:

La entrevistada sugiere que se establezca una plataforma que controle autenticación para la modificación de proyectos en el móvil.

Tabla 9. Explicación Entrevista 2
Autora: Nadia Narvaez Figueroa

3.6.2 Casos De Uso Expandidos

3.6.2.1 Plataforma Movil

CASO DE USO 1: Registrar Usuarios

ACTORES: Directivos, Secretaria y Contadora. (Empleados)

PROPÓSITO: Ingresar a la plataforma

DESCRIPCIÓN: Los empleados ingresan en la plataforma por medio de usuario y contraseña.

TIPO: Primario, Esencial.

Curso Normal de Eventos

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma móvil	
2		Se muestra ingreso por usuario y contraseña
3	El empleado ingresa con sus datos	Verifica los datos y si no le pide que se registre
4	El empleado se registra	
6		Verifica que los datos y accede a la plataforma
7		Muestra el menú de la plataforma móvil
8	Visualiza lo requerido.	

Tabla 10. Plataforma Movil, Caso 1, Curso Normal de Eventos.

Autora: Nadia Narvaez Figueroa

Curso alterno de eventos

Pasos	Acciones
3-4-7-8	Datos incompletos No se han ingresado todos los datos de carácter obligatorio. Se informa al usuario lo anterior a través de un mensaje de error.

Tabla 11. Plataforma Movil, Caso 1, Curso alterno de Eventos

Autora: Nadia Narvaez Figueroa

CASO DE USO 2: Crear Proyectos

ACTORES: Directivos, Secretaria y Contadora.(Empleados)

PROPÓSITO: Crear proyectos

DESCRIPCIÓN: Los empleados se autentifican en la plataforma e ingresan para crear proyectos

TIPO: Primario, Esencial.

Curso Normal de Eventos

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma de proyectos con su usuario y contraseña	
2		La plataforma le muestra la opción proyectos
3	El empleado ingresa a proyectos	Se visualizan los proyectos se muestra los campos nombre y descripción
4	El empleado escribe su nuevo proyecto	
6		Verifica la función y guarda
7		Luego se actualizan los datos en la lista de proyectos
8	salir del sistema	

Tabla 12. Plataforma Movil, Caso 2, Curso Normal de Eventos

Autora: Nadia Narvaez Figueroa

CASO DE USO 3: Eliminar Proyectos**ACTORES:** Directivos, Secretaria y Contadora. (Empleados)**PROPÓSITO:** Eliminar proyectos**DESCRIPCIÓN:** Los empleados se autentifican en la plataforma e ingresan para eliminar proyectos**TIPO:** Primario, Esencial.**Curso Normal de Eventos**

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma proyectos con su usuario y contraseña	
2		La plataforma le muestra la opción proyectos
3	El empleado ingresa a proyectos	Se visualizan los proyectos se muestra los campos nombre y descripción
4	El empleado selecciona un proyecto	
6		Verifica en los campos y se presiona botón eliminar
7		Luego se actualizan los datos en la lista de proyectos
8	salir del sistema	

Tabla 13. Plataforma Movil, Caso 3, Curso Normal de Eventos

Autora: Nadia Narvaez Figueroa

CASO DE USO 4: Modificar Proyectos**ACTORES:** Directivos, Secretaria y Contadora. (Empleados)**PROPÓSITO:** Modificar proyectos

DESCRIPCIÓN: Los empleados se autentifican en la plataforma e ingresan para modificar proyectos

TIPO: Primario, Esencial.

Curso Normal de Eventos

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma proyectos con su usuario y contraseña	
2		La plataforma le muestra la opción proyectos
3	El empleado ingresa a proyectos	Se visualizan los proyectos se muestra los campos nombre y descripción
4	El empleado selecciona el proyecto a editar	
6		Verifica en los campos nombre y descripción y edita la información y presione Guardar
7		Luego se actualizan los datos en la lista de proyectos
8	salir del sistema	

Tabla 14. Plataforma Movil, Caso 4, Curso Normal de Eventos
Autora: Nadia Narvaez Figueroa

CASO DE USO 5: Obtener Proyectos**ACTORES:** Directivos, Secretaria y Contadora. (Empleados)**PROPÓSITO:** Obtener proyectos**DESCRIPCIÓN:** Los empleados se autentifican en la plataforma e ingresan para obtener proyectos**TIPO:** Primario, Esencial.**Curso Normal de Eventos**

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma proyectos con su usuario y contraseña	
2		La plataforma le muestra la opción proyectos
3	El empleado ingresa a proyectos	Se visualizan los proyectos en una lista y se muestra los campos nombre y descripción
4	El empleado visualiza o selecciona el proyecto para ver su descripción	
6	Salir del sistema	

Tabla 15. Plataforma Movil, Caso 5, Curso Normal de Eventos
 Autora: Nadia Narvaez Figueroa

3.6.2.2 Plataforma Web**CASO DE USO 1:** Administración de Chat**ACTORES:** Directivos, Secretaria, Contadora y comunidad.**PROPÓSITO:** Administrar Chat**DESCRIPCIÓN:** Los empleados se autentifican en la plataforma e ingresan al chat**TIPO:** Primario, Esencial.

Curso Normal de Eventos

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma del plataforma web con su respectivo user y password	
2		La plataforma le muestra la opción de registrarse en el caso no tenga su user y password
3	El empleado ingresa al chat	Se visualizan los contactos activos y registrados
4	El empleado digita su mensaje	
6		Verifica y envía se mensaje
7		Luego se actualizan los datos por su respuesta
8	salir del chat y de la página web	

Tabla 16. Plataforma web, Caso 1, Curso Normal de Eventos
 Autora: Nadia Narvaez Figueroa

Curso alterno de eventos

Pasos	Acciones
1, 2	Datos incompletos No se han ingresado todos los datos de carácter obligatorio. Se informa al usuario lo anterior a través de un mensaje de error.

Tabla 17. Plataforma Web, Caso 1, Curso Alterno de Eventos
 Autora: Nadia Narvaez Figueroa

CASO DE USO 2: Obtener Proyectos**ACTORES:** Directivos, Secretaria, Contadora, comunidad.**PROPÓSITO:** Obtener proyectos**DESCRIPCIÓN:** Los empleados ingresan a la página para visualizar los proyectos**TIPO:** Primario, Esencial.**Curso Normal de Eventos**

N ro	Acciones del Actor	Respuestas del Sistema
1	Este caso de uso comienza cuando el empleado ingresa a la plataforma web	
2		La plataforma le muestra la opción proyectos
3	El empleado ingresa a proyectos	Se visualizan los proyectos en una lista
4	El empleado escoge un proyecto y le da clic	
6		Se muestra la descripción del proyecto.
7	Cuando finaliza solo sale de la página web.	

Tabla 18.Plataforma Web, Caso 2, Curso Normal de Eventos
Autora: Nadia Narvaez Figueroa

CAPITULO 4.

4. Implementación de la Solución.

4.1 Plataforma Web

La plataforma web se realizó en un gestor de contenidos llamada WordPress versión 3.9.3 tomando en cuenta los requerimientos de la Parroquia de Malacatos, además posee fácil interfaz de usuario su dominio es el siguiente:

www.proyectosmalacatos.com.

Todo el pueblo de Malacatos puede informarse por medio de esta página así mismo los empleados se registrar para chatear, y visualizar los cambios en los proyectos.

El gestor de contenido que se utilizo es WordPress, así mismo para la utilización del chat se instaló un plug-ins de WordPress que permita dicha funcionalidad, además se habilito la función de este gestor de contenidos, para el registro de usuarios.

En cuanto a los proyectos se utilizó el servidor web mediante Response y Request para la visualización de los proyecto en la página web, mientras los mismos son actualizados y administrados en el móvil.

4.2 Plataforma Movil

Se desarrolló la plataforma en PHP con servicios REST y con el framework Slim. La API REST es la encargada de comunicarse entre la plataforma y el servidor

Mediante los servicios REST se obtiene los métodos HTTP más conocidos comúnmente como GET, POST, PUT, DELETE.

Pasos a seguir para crear la plataforma:

Se creó un archivo de configuración llamado Config.php, este contiene la configuración del proyecto con los parámetros de la conexión a la base de datos.

```
<?php
/**
 * Database configuration
 */
define('DB_USERNAME', 'proyecto_usuario');
define('DB_PASSWORD', 'Soxx@6&Dm$3&');
define('DB_HOST', 'localhost');
define('DB_NAME', 'proyecto_bdmalacatos');

define('USER_CREATED_SUCCESSFULLY', 0);
define('USER_CREATE_FAILED', 1);
define('USER_ALREADY_EXISTED', 2);
?>
```

Se creó Db.Connect.php que es el encargado de hacer la conexión con la base de datos

```
class DbConnect {
private $conn;
function __construct() {
}
function connect() {
include_once dirname(__FILE__) . './Config.php';
$this->conn = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD,
DB_NAME);
if (mysqli_connect_errno()) {
```

```

echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
return $this->conn;
}
}

```

Luego se creó otro archivo DbHandler.php, esta se encarga de crear las funciones necesarias para realizar el CRUD en la base de datos.

```

public function createUser($usu_nombre, $usu_email, $password) {
require_once 'PassHash.php';
$response = array();
if (!$this->isUserExists($usu_email)) {
$password_hash = PassHash::hash($password);
$usu_api_key = $this->generateApiKey();
$stmt = $this->conn->prepare("INSERT INTO usuarios(usu_nombre, usu_email,
password_hash, usu_api_key, usu_estado) values(?, ?, ?, ?, 1)");
$stmt->bind_param("ssss", $usu_nombre, $usu_email, $password_hash,
$usu_api_key);
$result = $stmt->execute();
$stmt->close();
if ($result) {
return USER_CREATED_SUCCESSFULLY;
} else {
return USER_CREATE_FAILED;}
} else {
return USER_ALREADY_EXISTED;}
return $response;}

```

Explicación De Código

```
//crea un arreglo
$response = array();

// Comprueba si el usuario ya existía en db
if (!$this->isUserExists($usu_email)) {

// Genera password hash
$password_hash = PassHash::hash($password);

// Generat AI key
$usu_api_key = $this->generateApiKey();

// se crea la sentencia para insertar el usuario
$stmt = $this->conn->prepare("INSERT INTO usuarios(usu_nombre, usu_email,
password_hash, usu_api_key, usu_estado) values(?, ?, ?, ?, 1)");
$stmt->bind_param("ssss", $usu_nombre, $usu_email, $password_hash,
$usu_api_key);
$result = $stmt->execute();
$stmt->close();

// Verifica si la inserción se realizó con éxito
if ($result) {

// Usuario insertado con éxito
return USER_CREATED_SUCCESSFULLY;
} else {

// Falló al crear el usuario
return USER_CREATE_FAILED;}

} else {

// Usuario que ya existe en la bd
return USER_ALREADY_EXISTED;}

//parámetro enviado a la función JSON
return $response;}
```

4.2.1 Proceso para empaquetar en Android.

Con el id eclipse y el adt y sdk de Android se crea la plataforma Android y en esta mismo se la empaqueta, con una extensión .apk

4.3 Web Service

Una vez creada todas las funciones del CRUD necesarias se procede a crear otro archivo index.php para manejar el llamado individual de estas funciones con el servicio REST.

```
<?php
$app->post('/registrar_usuario', function() use ($app) {
verifyRequiredParams(array('usu_nombre', 'usu_email', 'password'));
$response = array();
$usu_nombre = $app->request->post('usu_nombre');
$usu_email = $app->request->post('usu_email');
$password = $app->request->post('password');
validateusu_email($usu_email);
$db = new DbHandler();
$res = $db->createUser($usu_nombre, $usu_email, $password);
if ($res == USER_CREATED_SUCCESSFULLY) {
header('Status: 301 Moved Permanently', false, 301);
echo "<script language='JavaScript'>alert('Usuario se ha registrado con
éxito;');</script>";
include 'index.html';
exit();
} else if ($res == USER_CREATE_FAILED) {
header('Status: 301 Moved Permanently', false, 301);
echo "<script language='JavaScript'>alert('Se produjo un error al
registrarse;');</script>";
include 'registrar.html';
```

```

exit();
} else if ($res == USER_ALREADY_EXISTED) {
$response["error"] = true;
$response["message"] = "este email ya existe";
header('Status: 301 Moved Permanently', false, 301);
echo "<script language='JavaScript'>alert('este email ya existe;');</script>";
include 'registrar.html';
exit();
}
echoResponse(201, $response);
});
?>

```

Explicación Del Código

```

<?php
$app->post('/registrar_usuario', function() use ($app) {
// verificar parámetros requeridos
verifyRequiredParams(array('usu_nombre', 'usu_email', 'password'));
$response = array();
// leer parametros POST
$usu_nombre = $app->request->post('usu_nombre');
$usu_email = $app->request->post('usu_email');
$password = $app->request->post('password');
// validar si el email es válido
validateusu_email($usu_email);
$db = new DbHandler();
// Llama al método de crear el usuario
$res = $db->createUser($usu_nombre, $usu_email, $password);
// comprobar si el usuario fue creado satisfactoriamente
if ($res == USER_CREATED_SUCCESSFULLY) {
header('Status: 301 Moved Permanently', false, 301);

```

```

//presentar mensaje JavaScript
echo "<script language='JavaScript'>alert('Usuario se ha registrado con
éxito;');</script>";

//re direccionar a la página de inicio para logearse de nuevo
include 'index.html';
exit();

// registro de usuario no exitoso
} else if ($res == USER_CREATE_FAILED) {
header('Status: 301 Moved Permanently', false, 301);
echo "<script language='JavaScript'>alert('Se produjo un error al
registrarse;');</script>";
include 'registrar.html';
exit();
} else if ($res == USER_ALREADY_EXISTED) {
$response["error"] = true;
$response["message"] = "este email ya existe";
header('Status: 301 Moved Permanently', false, 301);
echo "<script language='JavaScript'>alert('este email ya existe;');</script>";
include 'registrar.html';
exit();
}
echoResponse(201, $response);
});
?>

```

El servicio web que une las dos plataformas tanto la móvil como la web en el servidor es el encargado de realizar un conjunto de funciones concretas y bien definidas, como lo son: crear, modificar, eliminar y obtener proyectos.

Se creó una función hecho response que se encarga de hacer un eco de las respuestas. Al llamar a todas las solicitudes, una respuesta JSON se emitirá como un

código de estado HTTP en el lado del cliente, que tiene que verificar el código de estado HTTP de respuesta.

Si el estado es 200 la solicitud se ha procesado correctamente, si el valor de error es true significa que se produjo un error durante el procesamiento de los datos del usuario.

- Cuando se registró exitosamente se emite la siguiente respuesta:

```
{
"error": false,
"message": "Usted se registró correctamente"
}
```

- Si en la plataforma no se encuentran parámetros obligatorios se emite el siguiente JSON:

```
{
"error": true,
"message": "Se requiere parámetros obligatorios"
}
```

En conclusión si un empleado edita en el móvil un proyecto en particular el mismo se actualiza en la Plataforma web, esta interacción se realizó gracias al WebServices.

4.3.1 Métodos

MÉTODOS POST	
NOMBRE:	Registrar Usuario
URL:	http://www.proyectosmalacatos/v1/registrar_usuario
PARAMETROS:	usu_nombre, usu_email, password
RESULTADO:	Error: Si hay sucede algún error devuelve true, caso contrario false. Usuario creado

NOMBRE:	Iniciar Sesión
URL:	http://www.proyectosmalacatos/v1/iniciar_sesion
PARAMETROS:	usu_email, password
RESULTADO:	Error: Si hay sucede algún error devuelve true, caso contrario false. usu_nombre, usu_email, usu_fecha_creacion
NOMBRE:	Crear Proyecto
URL:	http://www.proyectosmalacatos/v1/crear_proyecto
PARAMETROS:	pro_proyecto, pro_estado, pro_fecha_creacion, pro_descripcion, pro_localizacion, pro_actividad, pro_inversion
RESULTADO:	Error: Si hay sucede algún error devuelve true, caso contrario false. Proyecto Creado

Tabla 19. Método POST

Autora: Nadia Narvaez Figueroa

MÉTODOS GET	
NOMBRE:	Obtener Proyectos
URL:	http://www.proyectosmalacatos/v1/obtener_proyectos
PARAMETROS:	usu_id
RESULTADO:	Error: Si hay sucede algún error devuelve true, caso contrario false. Pro_id, pro_proyecto, pro_estado, pro_fecha_creacion, pro_descripcion, pro_localizacion, pro_actividad, pro_inversion
NOMBRE:	Obtener Proyecto
URL:	http://www.proyectosmalacatos/v1/proyectos/:pro_id pro_id
PARAMETROS:	Error: Si hay sucede algún error devuelve true, caso contrario false. Pro_id, pro_proyecto, pro_estado, pro_fecha_creacion, pro_descripcion, pro_localizacion, pro_actividad, pro_inversion

Tabla 20. Método GET

Autora: Nadia Narvaez Figueroa

MÉTODO PUT	
NOMBRE:	Actualizar Proyecto
URL:	http://www.proyectosmalacatos/v1/actualizar_proyecto/:pro_id
PARAMETROS:	pro_id
RESULTADO:	Error: Si hay sucede algún error devuelve true, caso contrario false. Pro_id, pro_proyecto, pro_estado, pro_fechacreacion, pro_descripcion, pro_localizacion, pro_actividad, pro_inversion

Tabla 21.Método PUT

Autora: Nadia Narvaez Figueroa

MÉTODO DELETE	
NOMBRE:	Eleminar Proyecto
URL:	http://www.proyectosmalacatos/v1/eliminar_proyecto
PARAMETROS:	pro_id
RESULTADO:	Error: Si hay sucede algún error devuelve true, caso contrario false. Proyecto Eliminado

Tabla 22.Método DELETE

Autora: Nadia Narvaez Figueroa

CAPÍTULO 5.

5. Pruebas

Se realizó pruebas generales en la plataforma Web y Movil

5.1 Pruebas en la Plataforma Movil

La plataforma movil gráficamente se muestra en la siguiente ilustración:

The screenshot displays a mobile application interface with a green header and a white background. The header contains the text 'Hola bailoterapia' and 'Administración Proyectos Malacatos'. Below the header, there are several input fields: 'proyecto', 'responsable', and a date field containing '22/12/2014'. A dropdown menu is set to 'INICIADO'. Below these fields is a text area for 'observaciones'. At the bottom, there are three green buttons: 'Registrar/Actualizar', 'Nuevo Proyecto', and 'Eliminar'.

Ilustración 13. Diseño plataforma Movil
Autora Nadia Narvaez

Se realizó diferentes pruebas en la plataforma movil a continuación se detallan las siguientes:

Denominación de Prueba	Observación	Conclusión
Ingresar a la plataforma movil sin correo es decir (@)	No permite ingresar sin correo, ni al momento de registrarse.	Los empleados del GAD deben usar únicamente su correo electrónico.
En la plataforma movil se realizó el control probando la aplicación en 20 dispositivos móviles android al mismo tiempo.	La plataforma movil funciona de manera correcta en todos los dispositivos android	No es necesario aumentar el número de dispositivos para su prueba ya que en el GAD de Malacatos únicamente la ejecutaran 4 empleados como máximo
Se realizó pruebas en dispositivos android desde su versión 2.2 hasta la 4.1	La plataforma funciona correctamente en las versiones señaladas de dispositivos android	La plataforma tiene adaptabilidad para la mayoría de versiones de android.
En la plataforma movil se procedió a guardar proyectos con los campos incompletos	No permite guardar con datos incompletos	Los campos que posee la plataforma movil son datos generales e importantes se consideró llenar por completo los mismos.
Se probó la aplicación en diferentes anchos de banda como: <ul style="list-style-type: none"> • GSM(2G) • GPRS(2.5G) • EDGE(2.75G) • UMTS(3G) • HSDPA(3.5G) 	La aplicación respondió correctamente desde el UMTS (3G) en adelante y presente lentitud desde el EDGE (2.75G) hacia atrás.	La plataforma movil debe utilizarse desde el una conexión de UMTS (3G) en adelante.

Tabla 23. Pruebas Plataforma Movil
Autora Nadia Narvaez

5.2 Pruebas en la Plataforma Web

La plataforma web gráficamente se muestra en la siguiente ilustración:

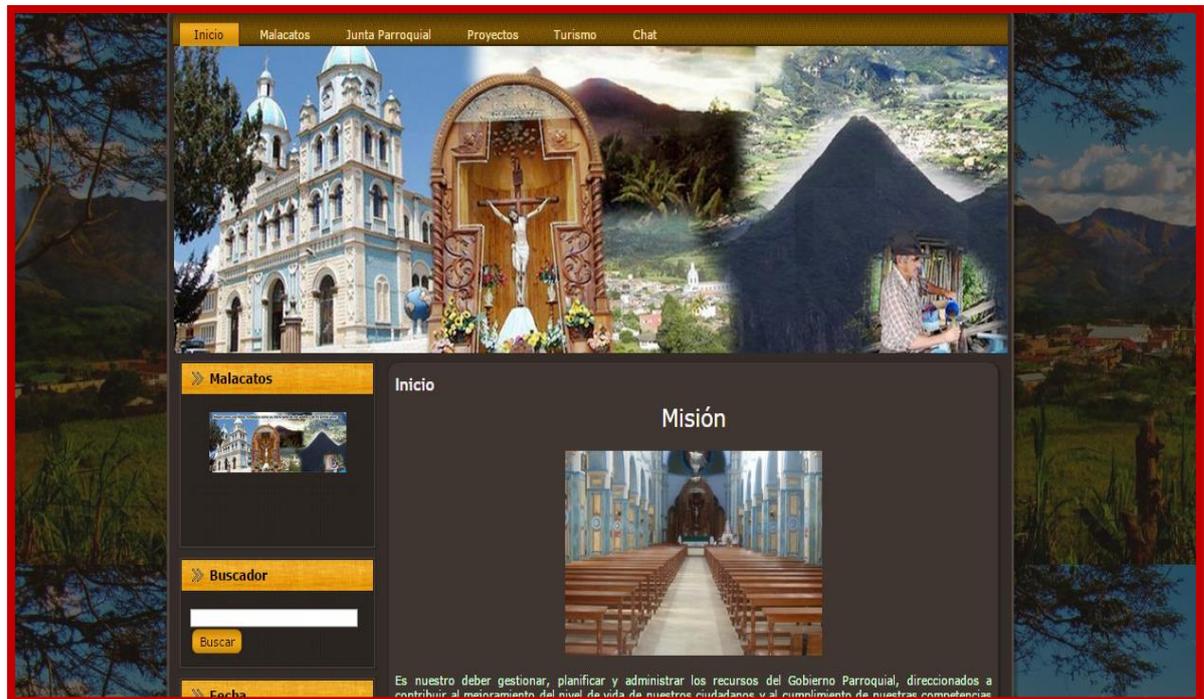


Ilustración 14. Diseño Plataforma Web
Autora Nadia Narvaez

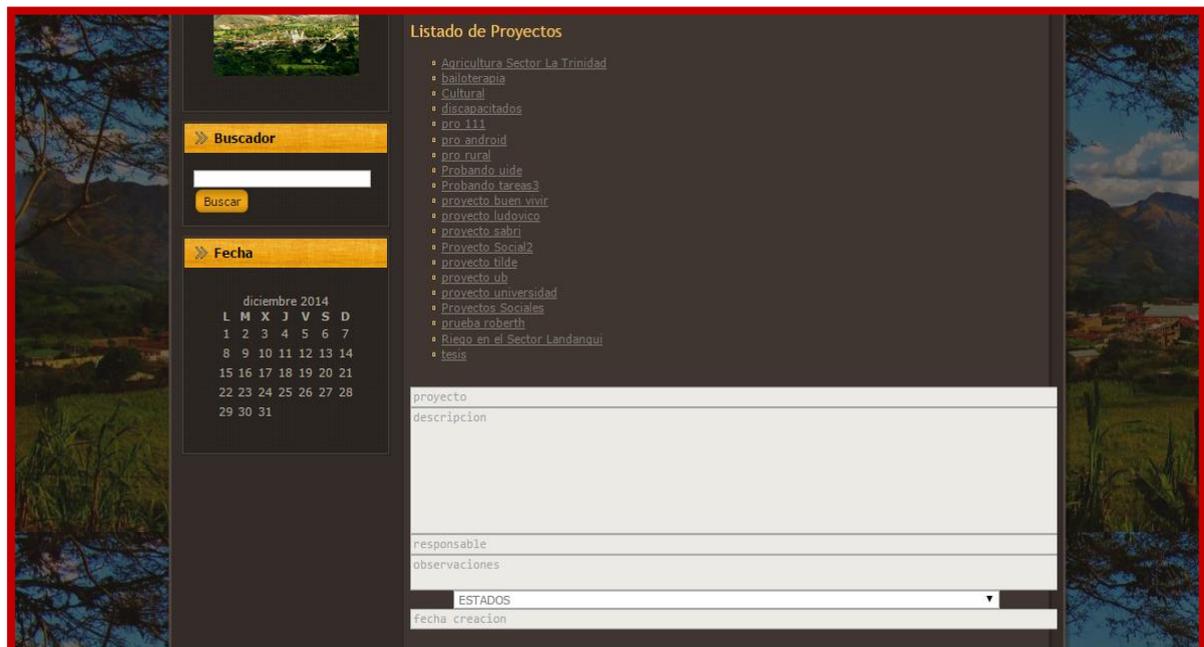


Ilustración 15. Visualización de los Proyectos en la Plataforma Web
Autora Nadia Narvaez

Se realizó diferentes pruebas en la plataforma web, cabe recalcar que la misma es más de carácter informativa y visualiza los proyectos ingresados en la plataforma móvil, por cuanto no presenta mucha complejidad al usuario final a continuación se detallan las siguientes pruebas generales:

Denominación de Prueba	Observación	Conclusión
Logearse en la plataforma web para chatear, datos incompletos	No permite ingresar sin llenar todos los campos del registro de usuario.	La comunidad de Malacatos debe logearse a la plataforma web si desea chatear desde la misma.
20 usuarios chateando al mismo tiempo	Se observa normalidad en las conversaciones (CHAT) es un plug-ins de WordPress ya comprobado para la visita de varios usuarios a la vez	La colectividad de Malacatos puede chatear sin ningún inconveniente dentro de la Plataforma Web.

Tabla 24. Pruebas Plataforma Web
Autora Nadia Narvaez

6. Conclusiones y Recomendaciones

Conclusiones

Las plataformas móviles y web, se deben implementar en la mayoría de instituciones públicas y privadas debido a la facilidad de comunicación que estas presentan ya sea para estar informado o para la toma de decisiones de las mismas, como se lo implementó en el Gad Parroquial de Malacatos donde se obtuvo resultados satisfactorios de acuerdo a las necesidades y requerimientos planteados.

- ✓ Un buen análisis, diseño e implementación se alcanzó gracias al uso de la metodología de desarrollo Kanban tanto en la plataforma móvil, como en la Web, se realizó un tablero de tareas específicas tomando en cuenta un correcto levantamiento de requerimientos de los miembros de la institución y la visualización de las necesidades en general.
- ✓ Los servicios web son significativos en la vinculación de dos aplicaciones, mediante estos servicios se pudo realizar la interacción entre la plataforma Web y Movil del Gad Parroquial de Malacatos, se usó servicios Rest ya que no consumen muchos recursos y generalmente son fáciles de construir y adoptar.
- ✓ En el presente trabajo se empleó el Framework Slim que permite un desarrollo rápido de las aplicaciones, utilizando APIs con métodos HTTP estándar y personalizados, lo que facilitó la implementación para la administración de los Proyectos de la Plataforma Movil y Web.

Recomendaciones

- ✓ Se recomienda realizar una mejora continua al presente proyecto, ya que la optimización de las plataformas según la demanda y el tiempo de espera, mejoran y aumentan el interés de los usuarios finales.
- ✓ Se recomienda incluir en la plataforma Web una funcionalidad que presente el Historial de los Proyectos para lograr un mejor tratamiento y control de los mismos.

- ✓ Capacitación permanente a los encargados del centro de cómputo para que faciliten el uso de la Plataforma Web a toda la colectividad de Malacatos, así mismo a los empleados del Gad para que optimicen su labor en proyectos de la Parroquia.
- ✓ La plataforma movil está diseñada para dispositivos Android, se recomienda implementarlos en otros sistemas operativos móviles, según la demanda tecnológica de teléfonos o smartphones, como por ejemplo: iOS, Blackberry.
- ✓ Se recomienda el uso de la plataforma Movil para dispositivos Android desde el rango UMTS (3G) de ancho de banda en adelante, para lograr su correcta y óptima funcionalidad.
- ✓ Ejecutar un mantenimiento preventivo, periódico y continuo, tanto del hardware como del software, a fin de evitar los daños físicos y lógicos en el centro de cómputo que impida la utilización normal de la Plataforma Web por la colectividad Malacatense.

7. Bibliografía

- Alvarez, M. A. (19 de 10 de 2012). *Desarrollo Web*. Obtenido de Copyright: desarrolloweb.com/manuales/manual-jquery-mobile.html
- Bahit, E. (08 de 11 de 2011). *Desarrolloweb.com*. Obtenido de <http://www.desarrolloweb.com/articulos/desarrollo-agil-kanban.html>
- BARABASZ, A. (04 de 07 de 2013). *ELSERVER*. Obtenido de <http://www.elsevier.com/slim-mini-framework-rest-para-php/>
- Bstia. (2010). *La escalabilidad en la informática*. Recuperado el 4 de Junio de 2013, de <http://bstia.blogspot.es/1267138271/>
- Clark, B. P. (2012). *Sencha Touch Mobile JavaScript Framework*. Packt Publishing.
- CORONEL, G. V. (2013). Cuartos de Telecomunicaciones. *Publicación de Telemática*.
- Cranley, R. B. (2011). *Head First JQuery*. USA: O Reilly.
- DIMAGGIO, M. (23 de 08 de 2013). *4R BLOG*. Obtenido de <http://www.4rsoluciones.com/ventajas-y-desventajas-de-jquery-mobile/>
- Gauchat, J. D. (2012). *El Gran Libro HTML5, CSS3 y Javascript*. España: MARCOMBO.
- Gerrero, L. (Abril de 2013). *blogspot*. Recuperado el 23 de Julio de 2013, de <http://lizethguerrerod.blogspot.com/2013/04/cableadoestructurado.html>
- Gil, P., & Candelas, F. (2010). *Redes y Transmision de Datos*. España: Publicaciones Universidad de Alicante.
- Gilibets, L. (31 de 07 de 2013). *Comunidad IEBS*. Recuperado el 27 de 06 de 2014, de <http://comunidad.iebschool.com/iebs/general/metodologia-kanban/>
- Gilibets, L. (31 de Julio de 2013). *Comunidad IEBS*. Obtenido de <http://comunidad.iebschool.com/iebs/general/metodologia-kanban/>
- GOMEZ, S. (13 de 03 de 2013). *Agencia Digital*. Obtenido de <http://tuagenciadigital.com/servicios/desarrollo-aplicaciones/>
- Griera, J., & Jose, B. (2009). *Estructura de redes de computadoras*. Editorial UOC.
- Guilling, T. (2009). *El compañero de un principiante en COBIT*. Editorial Matador-Trovador.
- Isaca, O. (2013). *Isaca*. Recuperado el 4 de Junio de 2013, de <http://www.isaca.org/Knowledge-Center/cobit/Pages/Downloads.aspx>

- JACOBS, J. (2013). *Telecommunications Industry Association*. Recuperado el 3 de Julio de 2013, de <http://www.tiaonline.org/>
- Lockhart, J. (15 de Septiembre de 2011). *SLIM*. Obtenido de <http://www.slimframework.com/>
- Lopez, M. (2012). *Ideas y Opiniones*. Recuperado el 10 de Junio de 2013, de <http://mglopezc.blogspot.com/2011/04/framework-para-el-gobierno-de-ti-goita.html>
- MacDonal, M. (2011). *HTML5: The missing Manual*. EEUU: O'Reilly.
- Markgraf, B. (2012). Recuperado el 28 de julio de 2013, de Especificaciones de los cables Cat 5 y 6: http://www.ehowenespanol.com/especificaciones-cables-cat-5-cat-6-info_197062/
- Master Univ., e. I. (s.f.). *Modulo 4 - JQuery*. Copyright.
- McFarland, D. S. (2011). *JavaScript & jQuery: The Missing Manual*. EEUU: O'Reilly.
- Mediaprogramas. (2013). *My hardware gratis*. Recuperado el 20 de Junio de 2013, de <http://test-my-hardware.programas-gratis.net/>
- Micro, T. (2013). *Titanium Internet Security*. Recuperado el 15 de Junio de 2013, de <http://www.trendmicro.es/productos/titanium-internet-security/>
- MULLENWEG, M. (2003). *WORDPRESS*. Recuperado el 11 de 2014, de <http://es.wordpress.org/>
- Murphey, R. (2011). *Fundamentos jQuery*. USA: Copyright.
- NAVAJAS, A. (2015). *antonionavajas.com*. Obtenido de www.antonionavajas.com/blog/?p=1305
- Olson, P. (16 de 01 de 2015). *MANUAL DE PHP*. Obtenido de <http://php.net/manual/es/intro-what-is.php>
- Oracle. (2013). *Oracle*. Recuperado el 4 de Junio de 2013, de Administracion Oracle: http://docs.oracle.com/cd/E26921_01/html/E25871/ipplan-4.html
- Rodríguez, A. (16 de 10 de 2011). *androideity*. Obtenido de <http://androideity.com/2011/10/16/documentacion-de-phonegap-en-espanol/>
- Siemon. (2013). *Siemon*. Recuperado el 3 de julio de 2013, de http://www.siemon.com/la/white_papers/11-19-07-cabling-lifecycles.asp
- Skarin, H. K. (2010). *Kanban y Scrum* – . Estados Unidos: © C4Media Inc. .
- VILLA, R. D. (13 de 04 de 2012). *MOVILIDAD*. Obtenido de PhoneGap vs. Código Nativo: <http://rauldevilla-movilidad.blogspot.com/2012/04/phonegap-vs-codigo-nativo.html>
- Winaudit, P. (2013). *Winaudit*. Recuperado el 20 de Junio de 2013, de <http://winaudit.programas-gratis.net/>

8. ANEXOS

ANEXO 1

FOTOS E IMÁGENES DE LAS INSTALACIONES DEL GAD PARROQUIAL DE MALACATOS



Ilustración 16. Oficina Malacatos
Autora: Nadia Narvaez

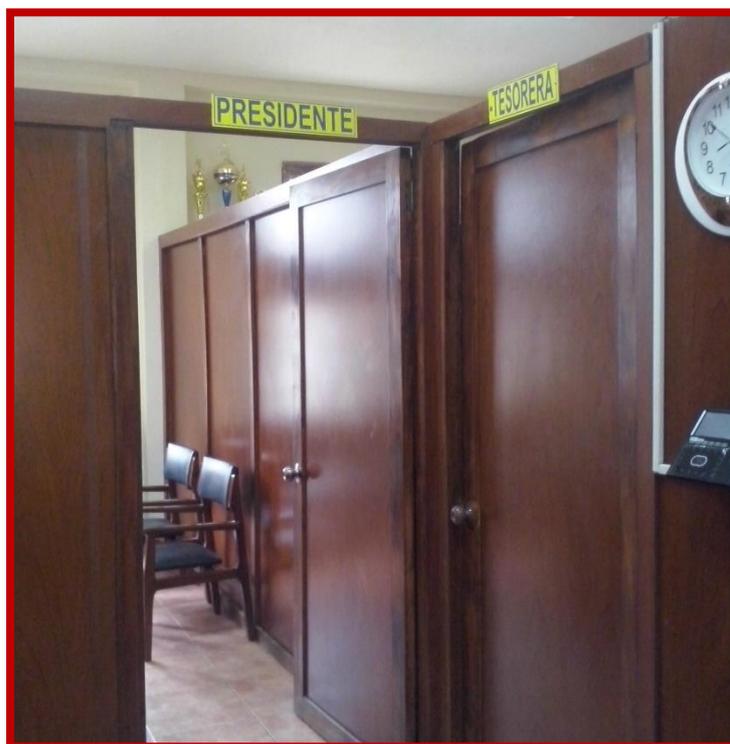


Ilustración 17. Oficina Malacatos 1
Autora: Nadia Narvaez

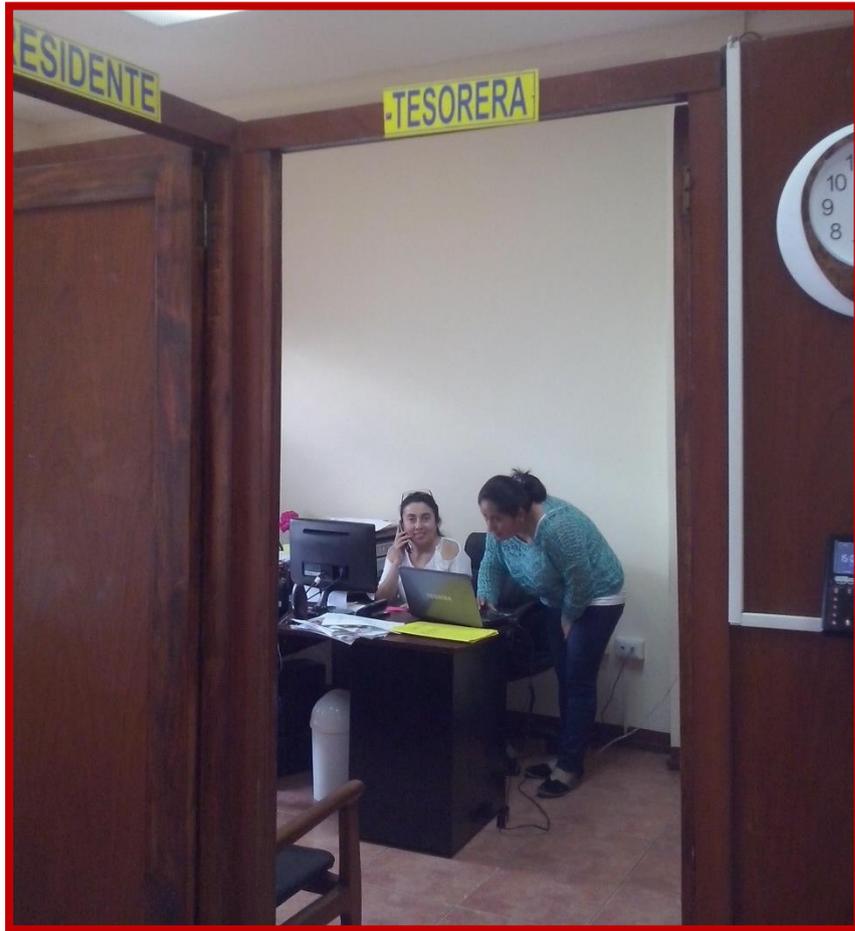


Ilustración 18. Tesorería Malacatos
Autora: Nadia Narvaez

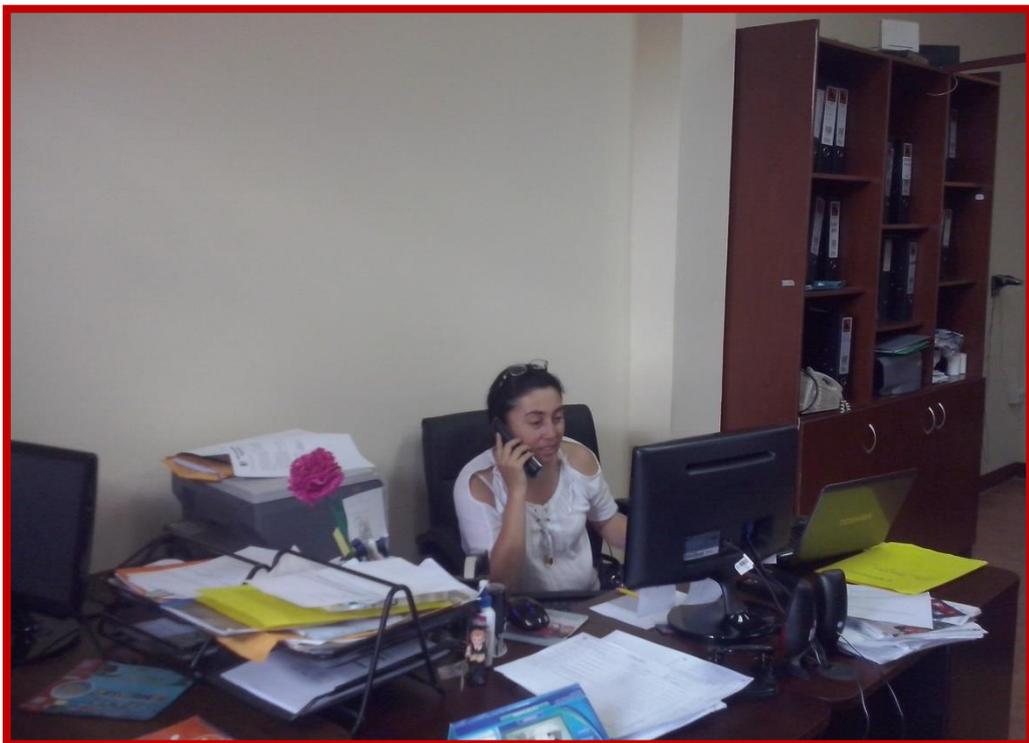


Ilustración 19. Tesorera
Autora: Nadia Narvaez



Ilustración 20. Presidenta del GAD
Autora: Nadia Narvaez



Ilustración 21. Secretaria del GAD
Autora: Nadia Narvaez

ANEXO 2 MANUAL DEL USUARIO

PLATAFORMA WEB

Página Principal (www.proyectosmalacatos.com)

1. En la página inicial y principal se muestra un menú de botones como lo son: Inicio, Malacatos, Junta Parroquial, Proyectos, Chat, Turismo.

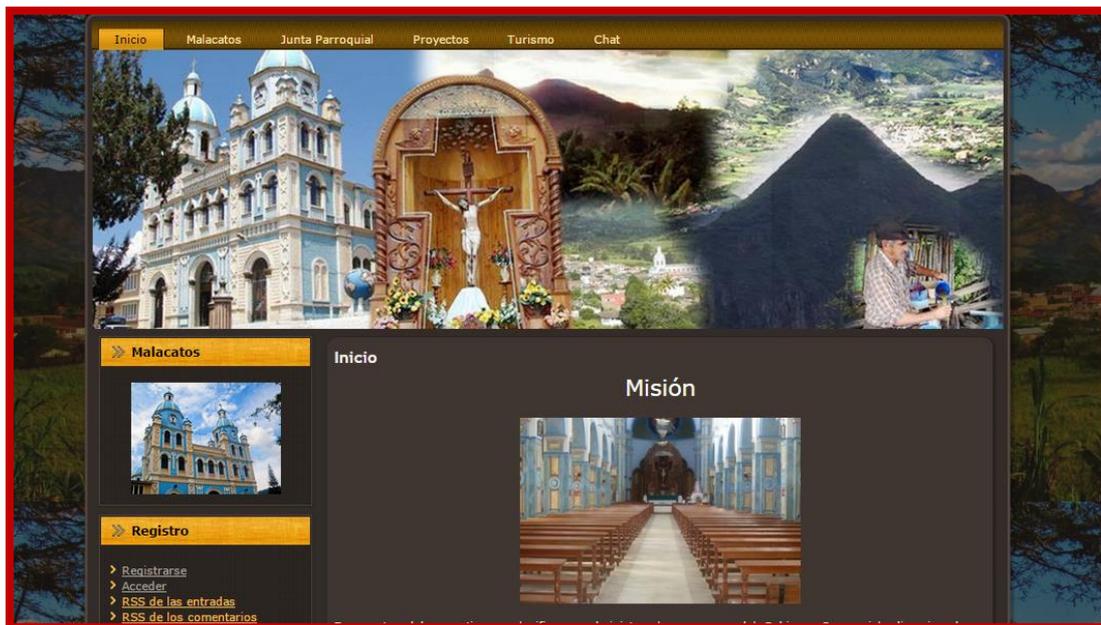


Ilustración 22. Plataforma Web Inicio
Autora: Nadia Narvaez

2. Luego de los botones principales y la portada inicial se muestra:



Ilustración 23. Plataforma Web Misión
Autora: Nadia Narvaez

3. Presionando en botón Malacatos se muestra una breve historia de la parroquia:



Ilustración 24. Plataforma Web Malacatos

Autora: Nadia Narvaez

4. Presionando en botón Junta Parroquial se muestran los miembros del GAD parroquial:

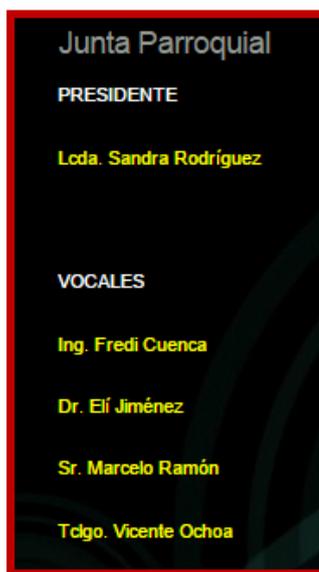


Ilustración 25. Plataforma Web Miembros GAD

Autora: Nadia Narvaez

5. Presionando en botón Proyectos se muestran sus proyectos:



Ilustración 26. Plataforma Web Proyectos
 Autora: Nadia Narvaez

6. Presionando en botón Turismo se destacan los lugares más reconocidos de Malacatos



Ilustración 27. Plataforma Web Turismo
 Autora: Nadia Narvaez

7. Presionando Chat se muestra, un listado de los participantes y si ingresa con usuario y contraseña podrás chatear.

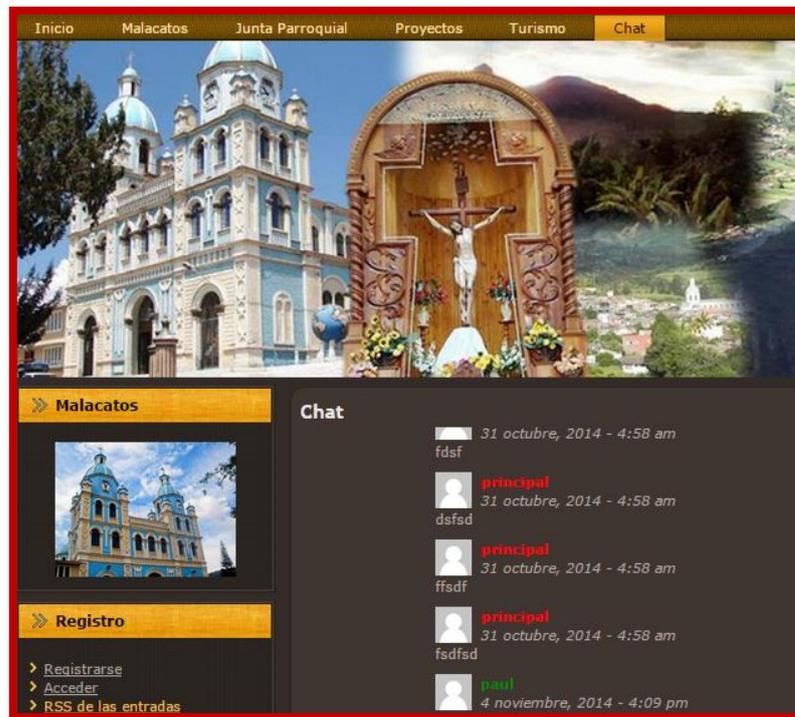


Ilustración 28. Plataforma Web Chat
 Autora: Nadia Narvaez

PLATAFORMA MOVIL

Debe de ser empleado del Gad de Malacatos, primero la secretaria le envía el archivo de instalación para ser instalada en su movil, luego de instalado se mostrara:

1. Se presenta en el menú del dispositivo móvil la App Malacatos

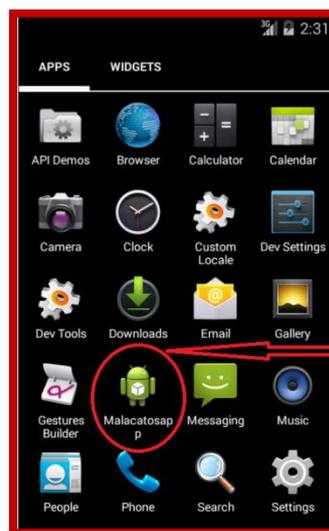
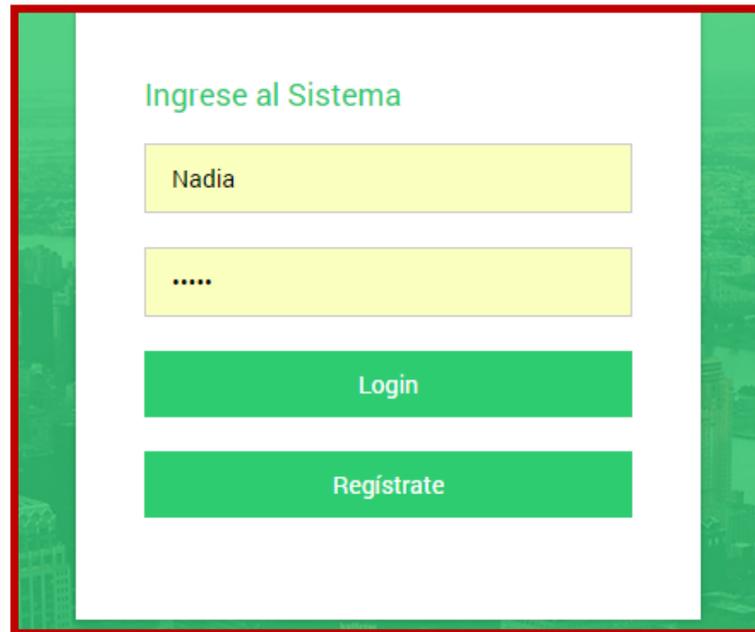


Ilustración 29. Plataforma Movil Ingreso
 Autora: Nadia Narvaez

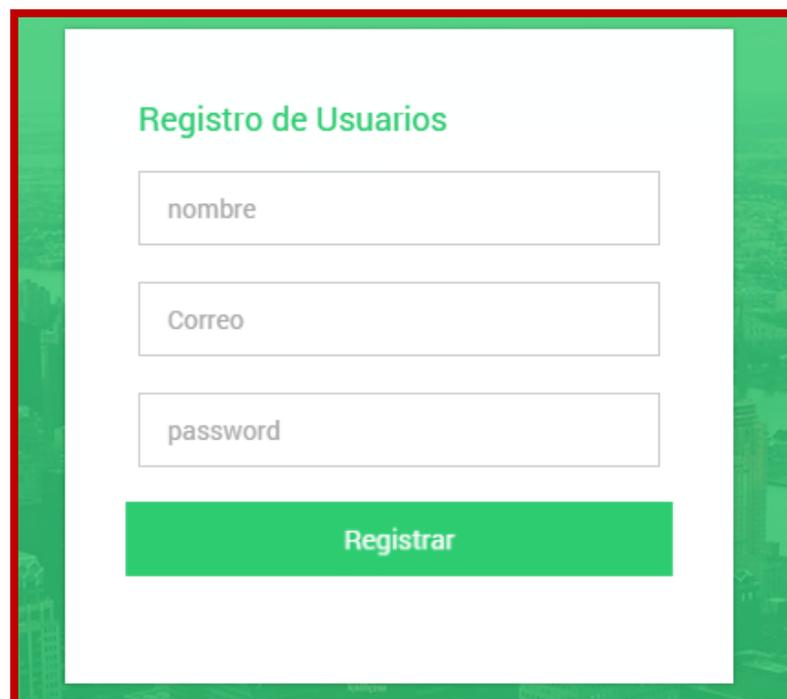
- Al abrir la plataforma se muestra la siguiente pantalla donde debes registrarte.



The screenshot shows a mobile login interface with a white background and green accents. At the top, the title "Ingrese al Sistema" is displayed in green. Below the title are two yellow input fields: the first contains the name "Nadia" and the second contains six dots representing a password. Underneath these fields are two green buttons: "Login" and "Regístrate".

Ilustración 30. Plataforma Movil Login
Autora: Nadia Narvaez

- Si presiona registrarte se muestra los siguientes campos



The screenshot shows a mobile registration interface with a white background and green accents. At the top, the title "Registro de Usuarios" is displayed in green. Below the title are three white input fields with gray placeholder text: "nombre", "Correo", and "password". At the bottom of the form is a green button labeled "Registrar".

Ilustración 31. Plataforma Movil Registrarse
Autora: Nadia Narvaez

4. Al presionar Login se muestra la plataforma de proyectos, aquí tiene las opciones de Registrar/ actualizar, Nuevo proyecto y Eliminar. Los proyectos se enlistan en la parte inferior de la ventana.



Administración Proyectos Malacatos

prueba roberth

responsable

16/12/2014

INICIADO

Prueba de validacion de inicio

observaciones

Registrar/Actualizar

Nuevo Proyecto

Eliminar

Ilustración 32. Plataforma Movil Administración Proyectos
Autora: Nadia Narvaez

5. En la parte inferior se muestra el listado de proyectos



Lista de Proyectos

Agricultura Sector	Cultural
La Trinidad	discapacitado
Proyecto Social2	proyecto tilde
proyecto ub	prueba robert
Riego en el Sector	
Landangui	

Ilustración 33. Plataforma Movil Listado de Proyectos
Autora: Nadia Narvaez

ANEXO 3

MANUAL DEL PROGRAMADOR

Códigos y explicación de los métodos utilizados:

```
<?php
session_start();
require_once '../include/DbHandler.php';
require_once '../include/PassHash.php';
require '../libs/Slim/Slim.php';
//require '../alertas.js';

\Slim\Slim::registerAutoloader();

$app = new \Slim\Slim();

// User usu_id from db - Global Variable
$usu_id = NULL;
//$usu_id = 24;

function authenticate(\Slim\Route $route) {
    // Getting request headers
    print('yaaaaa');
    $headers = apache_request_headers();
    $response = array();
    $app = \Slim\Slim::getInstance();

    // Verifying Authorization Header
    if (isset($headers['Authorization'])) {
        $db = new DbHandler();

        // get the api key
        $usu_api_key = $headers['Authorization'];
        // validating api key
        if (!$db->isValidApiKey($usu_api_key)) {
            // api key is not present in usuarios table
```

```

$response["error"] = true;
$response["message"] = "Access Denied. Invalid Api key";
echoResponse(401, $response);
$app->stop();
} else {
    global $usu_id;
    // get user primary key usu_id
    $usu_id = $db->getUserId($usu_api_key);
}
} else {
    // api key is missing in header
    $response["error"] = true;
    $response["message"] = "Api key is misssing";
    echoResponse(400, $response);
    $app->stop();
}
}
/**
 * ----- METHODS WITHOUT AUTHENTICATION -----

```

⇒ Registro de Usuarios

Esta función maneja el registro de usuarios. Cuando un usuario no se encuentra registrado en el sistema este tendrá que hacerlo ingresando sus datos. En la siguiente tabla se puede encontrar la información de la URL, el método HTTP y los parámetros necesarios para ser publicado.

URL	/ registra_usuario
Método	POST
Parámetros	usu_nombre, usu_email, password

```

*/
/**
 * User Registration
 * url - /registrar_usuario
 * method - POST
 * params - usu_nombre, usu_email, password

```

```

*/
$app->post('/registrar_usuario', function() use ($app) {
    // check for required params
    verifyRequiredParams(array('usu_nombre', 'usu_email', 'password'));
    print_r('verifyRequiredParams');
    $response = array();

    // reading post params
    $usu_nombre = $app->request->post('usu_nombre');
    $usu_email = $app->request->post('usu_email');
    $password = $app->request->post('password');

    // validating usu_email address
    validateusu_email($usu_email);

    $db = new DbHandler();
    $res = $db->createUser($usu_nombre, $usu_email, $password);

    if ($res == USER_CREATED_SUCCESSFULLY) {
        $response["error"] = false;
        $response["message"] = "Se ha registrado con Ã©xito";
    } else if ($res == USER_CREATE_FAILED) {
        $response["error"] = true;
        $response["message"] = "Se produjo un error al registrarse";
    } else if ($res == USER_ALREADY_EXISTED) {
        $response["error"] = true;
        $response["message"] = "este email ya existe";
    }
    // echo json response
    echoResponse(201, $response);
});

```

⇒ Ingresar Registro

El siguiente código sirve para manejar la conexión del usuario. Después de verificar las credenciales de usuario.

En la siguiente tabla se puede encontrar la información de la URL, el método HTTP y los parámetros necesarios para ingresar.

URL	/ iniciar_sesion
Método	POST
Parámetros	usu_email, password

```
/**
 * User Login
 * url - /iniciar_sesion
 * method - POST
 * params - usu_email, password
 */
$app->post('/iniciar_sesion', function() use ($app) {
    // check for required params
    verifyRequiredParamsLogin(array('usu_email', 'password'));

    // reading post params
    $usu_email = $app->request()->post('usu_email');
    $password = $app->request()->post('password');
    $response = array();
    //$response = new stdClass();

    $db = new DbHandler();
    // check for correct usu_email and password
    if ($db->checkLogin($usu_email, $password)) {
        // get the user by usu_email
        $user = $db->getUserByusu_email($usu_email);
        if ($user != NULL) {
            $response["error"] = false;
            $response['nombre'] = $user['usu_nombre'];
            $response['email'] = $user['usu_email'];
            $response['apiKey'] = $user['usu_api_key'];
            $response['fechacreacion'] = $user['usu_fechacreacion'];
            // $usu_api_key = $user['usu_api_key'];
            // global $usu_id;
```

```

// $usu_id = $db->getUserId($usu_api_key);
global $usu_id;
$usu_nombre = $user['usu_nombre'];
$usu_email = $user['usu_email'];
$usu_id = $db->getUserByusu_email2($usu_email);

//funcion listar proeyectos
$response = array();
$result = $db->getAllUserTasks($usu_id);

//$response["error"] = false;
$response["proyectos"] = array();
// looping through result and preparing proyectos array
while ($pro_proyecto = $result->fetch_assoc()) {

    $tmp = array();
    //$tmp["pro_id"] = $pro_proyecto["pro_id"];
    $tmp["pro_proyecto"] = $pro_proyecto["pro_proyecto"];
    //$tmp["pro_fecha_creacion"] = $pro_proyecto["pro_fecha_creacion"];
    array_push($response["proyectos"], $tmp);
}
//$proprueba json_encode($response);
//fin listar proyectos
//echo $id_usu;
//$proy_usuario = $db->getAllUserTasks($usu_id);
//echo $proy_usuario;
//print_r($proy_usuario);
//probando utilizar para pasar parametros de js a js
// ?>
<!-- // <script type='text/javascript' id='identificador_js' src='main.js?nombre=<? echo
$usu_id; ?>' ></script> -->
<?php
//fin probando

$serial=urlencode(serialize($response));
header('Status: 301 Moved Permanently', false, 301);

```

```

        //header("Location:
cellar3/process_proyecto_obtener.php?id_usu=$usu_id&nombre=$usu_nombre&nuevoArray=$serial
");
        //header("Location: prueba.html?id_usu=$usu_id&nombre=$usu_nombre");
        //header("Location: cellar3/index.html?nuevoArray=$serial");

        echo 'Bienvenid@: '. $usu_nombre .';
        $_GET['id_usu']=$usu_id;

        exit();

    } else {
        $response['error'] = true;
        $response['message'] = "Ha ocurrido un error. Por favor, inténtalo de nuevo";

        header('Status: 301 Moved Permanently', false, 301);
        include 'prueba.html';
        exit();
    }
} else {
    $response['error'] = true;
    $response['message'] = 'Error de inicio de sesion. credenciales incorrectas';

    header('Status: 301 Moved Permanently', false, 301);
    include 'prueba.html';
    exit();

    // $mens = "Error de inicio de sesion. credenciales incorrectas";
    //header('Status: 301 Moved Permanently', false, 301);
    //header("Location: cellar3/process_proyecto_obtener_copia.php?mensaje=$mens");

}
echoResponse(200, $response);
});

/*
 * ----- METHODS WITH AUTHENTICATION -----
 */

```

```

/**
 * Listing all proyectos of partical user
 * method GET
 * url /obtener_proyectos
 */
//$app->get('/obtener_proyectos', 'authenticate', function() {
$app->get('/obtener_proyectos', function() {
    echo "yaaaaaaaa";
    global $usu_id;
    $response = array();
    $db = new DbHandler();

    // fetching all user proyectos
    $result = $db->getAllUserTasks($usu_id);

    $response["error"] = false;
    $response["proyectos"] = array();

    // looping through result and preparing proyectos array
    while ($pro_proyecto = $result->fetch_assoc()) {
        $tmp = array();
        $tmp["pro_id"] = $pro_proyecto["pro_id"];
        $tmp["pro_proyecto"] = $pro_proyecto["pro_proyecto"];

        $tmp["pro_fecha_creacion"] = $pro_proyecto["pro_fecha_creacion"];
        array_push($response["proyectos"], $tmp);
    }

    echoResponse(200, $response);
});

/**
 * Listing single pro_proyecto of partical user
 * method GET
 * url /proyectos/:pro_id
 * Will return 404 if the pro_proyecto doesn't belongs to user
 */

```

```

//$app->get('/obtener_proyecto/:pro_id', 'authenticate', function($pro_id) {
$app->get('/obtener_proyecto/:pro_id', function($pro_id) {
    global $usu_id;
    $response = array();
    $db = new DbHandler();

    // fetch pro_proyecto
    $result = $db->getTask($pro_id, $usu_id);

    if ($result != NULL) {
        $response["error"] = false;
        $response["pro_id"] = $result["pro_id"];
        $response["pro_proyecto"] = $result["pro_proyecto"];
        $response["pro_estado"] = $result["pro_estado"];
        $response["pro_fecha_creacion"] = $result["pro_fecha_creacion"];
        echoResponse(200, $response);
    } else {
        $response["error"] = true;
        $response["message"] = "The requested resource doesn't exists";
        echoResponse(404, $response);
    }
});

```

⇒ Crear nuevo Proyecto

Este método nos permite crear un nuevo proyecto en la base de datos a partir de que el usuario ingrese al sistema

En la siguiente tabla se puede encontrar la información de la URL, el método HTTP y los parámetros necesarios para ingresar y crear un nuevo proyecto.

URL	/ crear_proyecto/
Método	POST
Parámetros	Usu_nombre

```

/**
 * Creating new pro_proyecto in db
 * method POST
 * params - usu_nombre
 * url - /crear_proyecto/
 */
//$app->post('/crear_proyecto', 'authenticate', function() use ($app) {
$app->post('/crear_proyecto', function() use ($app) {
    // check for required params
    verifyRequiredParamsMenu(array('pro_proyecto'));
    $response = array();
    $pro_proyecto = $app->request->post('pro_proyecto');
    $pro_estado = $app->request->post('pro_estado');
    $pro_descripcion = $app->request->post('pro_descripcion');
    global $usu_id;
    $usu_id = $app->request->post('usu_id');
    $db = new DbHandler();
    //echo "usuEnIndex: " . $usu_id;
    // creating new pro_proyecto
    $pro_id = $app->request->post('pro_id');

    if ($pro_id != NULL) {
        $result = $db->updateTask($pro_id, $pro_proyecto, $pro_estado, $pro_descripcion);

        if ($result) {
            // pro_proyecto updated successfully
            $response["message"] = "Proyecto actualizado satisfactoriamente";

            header('Status: 301 Moved Permanently', false, 301);
            echo "<script language='javascript'>alert('estoy en el process;');</script>";
            include 'prueba.html';
            exit();
        } else {
            // pro_proyecto failed to update
            $response["message"] = "Actualizacion de proyecto no satisfactoria. Intente de nuevo!";
        }
    }
    echoResponse(200, $response);
}

```

```

    }
    else {
        $pro_id = $db->createTask($usu_id, $pro_proyecto, $pro_descripcion);
        if ($pro_id != NULL) {
            $response["message"] = "Proyecto creado satisfactoriamente";
            echoResponse(201, $response);
        } else {
            $response["message"] = "Failed to create pro_proyecto. Please try again";
            echoResponse(200, $response);
        }
    }
}
});

```

⇒ Modificar un Proyecto

Este método nos permite modificar proyectos de la base de datos, esto lo puede realizar cualquier usuario registrado. En la siguiente tabla se puede encontrar la información de la URL, el método HTTP y los parámetros necesarios para modificar un proyecto.

URL	/ actualizar_proyecto/:pro_id
Método	PUT
Parámetros	pro_proyecto, pro_estado

```

/**
 * Updating existing pro_proyecto
 * method PUT
 * params pro_proyecto, pro_estado
 * url - /actualizar_proyecto/:pro_id
 */
// $app->put('/actualizar_proyecto/:pro_id', 'authenticate', function($pro_id) use($app) {
$app->put('/actualizar_proyecto', function() use($app) {
    // check for required params
    verifyRequiredParams(array('pro_proyecto', 'pro_estado'));

    global $usu_id;
    $pro_id = $pro_id;
    $pro_proyecto = $app->request->put('pro_proyecto');

```

```

$pro_estado = $app->request->put('pro_estado');
echo " proid:" . $pro_id;
echo " proyeycto:" . $pro_proyecto;
echo " proestado:" . $pro_estado;

$db = new DbHandler();
$response = array();

// updating pro_proyecto
//No borrar$result = $db->updateTask($usu_id, $pro_id, $pro_proyecto, $pro_estado);
$result = $db->updateTask($pro_id, $pro_proyecto, $pro_estado);
if ($result) {
    // pro_proyecto updated successfully
    $response["error"] = false;
    $response["message"] = "Task updated successfully";
} else {
    // pro_proyecto failed to update
    $response["error"] = true;
    $response["message"] = "Task failed to update. Please try again!";
}
echoResponse(200, $response);
});

```

⇒ Eliminar un Proyecto

Este método nos permite eliminar proyectos de la base de datos siempre y cuando el usuario que los elimine sea el usuario que lo creo. En la siguiente tabla se puede encontrar la información de la URL, el método HTTP y los parámetros necesarios para eliminar un proyecto.

URL	/ eliminar_proyecto
Método	DELETE
Parámetros	-

```

/**
 * Deleting pro_proyecto. Users can delete only their proyectos
 * method DELETE
 * url /eliminar_proyecto
 */
$app->delete('/eliminar_proyecto/:pro_id', 'authenticate', function($pro_id) use($app) {
$app->post('/eliminar_proyecto', function() use($app) {
    global $usu_id;
    $usu_id = $app->request->put('usu_id');
    $pro_id = $app->request->put('id_proyectos');

    $db = new DbHandler();
    $response = array();
    //$result = $db->deleteTask($usu_id, $pro_id);
    $result = $db->deleteTask($pro_id);
    if ($result) {
        // pro_proyecto deleted successfully
        $response["error"] = false;
        $response["message"] = "Task deleted succesfully";
    } else {
        // pro_proyecto failed to delete
        $response["error"] = true;
        $response["message"] = "Task failed to delete. Please try again!";
    }
    echoRspnse(200, $response);
});

```

```

/**
 * Verifying required params posted or not
 */
function verifyRequiredParams($required_fields) {
    $error = false;
    $error_fields = "";
    $request_params = array();
    $request_params = $_REQUEST;
    // Handling PUT request params
    if ($_SERVER['REQUEST_METHOD'] == 'PUT') {

```

```

    $app = \Slim\Slim::getInstance();
    parse_str($app->request()->getBody(), $request_params);
}
foreach ($required_fields as $field) {
    if (!isset($request_params[$field]) || strlen(trim($request_params[$field])) <= 0) {
        $error = true;
        $error_fields .= $field . ', ';
    }
}

if ($error) {
    // Required field(s) are missing or empty
    // echo error json and stop the app
    $response = array();
    $app = \Slim\Slim::getInstance();
    $response["error"] = true;
    $response["message"] = 'Los campo(s) obligatorio(s) ' . substr($error_fields, 0, -2) . ' faltan o
estÃ¡n vacios';
    echoResponse(400, $response);
    $app->stop();
}
}

function verifyRequiredParamsLogin($required_fields) {
    $error = false;
    $error_fields = "";
    $request_params = array();
    $request_params = $_REQUEST;
    if ($_SERVER['REQUEST_METHOD'] == 'PUT') {
        $app = \Slim\Slim::getInstance();
        parse_str($app->request()->getBody(), $request_params);
    }
    foreach ($required_fields as $field) {
        if (!isset($request_params[$field]) || strlen(trim($request_params[$field])) <= 0) {
            $error = true;
            $error_fields .= $field . ', ';
        }
    }
}

```

```

if ($error) {
    $response = array();
    $app = \Slim\Slim::getInstance();
    header('Status: 301 Moved Permanently', false, 301);
    echo "<script language='javascript'>alert('Faltan campos obligatorios por llenar;');</script>";
    include 'login.html';
    exit();
    $app->stop();
}
}

function verifyRequiredParamsMenu($required_fields) {
    $error = false;
    $error_fields = "";
    $request_params = array();
    $request_params = $_REQUEST;
    if ($_SERVER['REQUEST_METHOD'] == 'PUT') {
        $app = \Slim\Slim::getInstance();
        parse_str($app->request()->getBody(), $request_params);
    }
    foreach ($required_fields as $field) {
        if (!isset($request_params[$field]) || strlen(trim($request_params[$field])) <= 0) {
            $error = true;
            $error_fields .= $field . ', ';
        }
    }
}
if ($error) {
    $response = array();
    $app = \Slim\Slim::getInstance();
    header('Status: 301 Moved Permanently', false, 301);
    echo "<script language='javascript'>alert('Faltan campos obligatorios por llenar;');</script>";
    include 'Prueba.html';
    exit();
    $app->stop();
}
}

```

⇒ Validar Usuario

Este método nos permite validar los datos del usuario para ingresar al sistema.

```
/**
 * Validating usu_email address
 */
function validateusu_email($usu_email) {
    $app = \Slim\Slim::getInstance();
    if (!filter_var($usu_email, FILTER_VALIDATE_EMAIL)) {
        $response["error"] = true;
        $response["message"] = 'email no es válida';
        echoResponse(400, $response);
        $app->stop();
    }
}

//validar idusuario
function validateusu_id($usu_nombre) {

    if ($usu_nombre != NULL) {
        $response["error"] = true;
        $response["message"] = 'usu_email address is not valid';
        echoResponse(400, $response);
        $app->stop();
    }
}

/**
 * Echoing json response to client
 * @param String $usu_estado_code Http response code
 * @param Int $response Json response
 */
function echoResponse($usu_estado_code, $response) {
    $app = \Slim\Slim::getInstance();
    // Http response code
    $app->usu_estado($usu_estado_code);

    // setting response content type to json
```

```

$app->contentType('application/json');

//echo json_encode($response);

header('Status: 301 Moved Permanently', false, 301);
echo '<script> alert(' . json_encode($response["message"]) . ');</script>';
exit();

//probando
//$varjson->message = json_encode($response, JSON_FORCE_OBJECT);
//echo "<script language='javascript'>alert('$varjson->message;');</script>";

//$_SESSION["tuarray"][2] = json_encode($response);

//echo "<script language='javascript'>alert('echo json_encode($response);')</script>";
//include 'login.html';
//otra forma
// echo '<script language="javascript">';
// echo 'alert("message successfully sent");'
// echo '</script>';
//echo "<script> alert(' . json_encode($response) . ');</script>";

//header('Status: 301 Moved Permanently', false, 301);
//header('Location: prueba.html');
//exit();
//echo '<script language="JavaScript"> alert(" Debe introducir una fecha "); </script>';
//mensajes($response);
//echo "<script language='javascript'>alert('echo json_encode($response);')</script>";

// $pregunta = new stdClass();

// $json = json_encode($pregunta);

// echo $json;
}
$app->run();
?>

```

DB.HANDLER

Esta clase nos va a servir para el manejo de todas las operaciones de la base de datos dbMalacatos y para la interacción con el servicio.

CRUD: Create, reade, update, delete.

```
<?php

/**
 * Clase para manejar todas las operaciones dbMalacatos
 * This class will have CRUD methods for database tables
 *
 * @author Nadia
 * @link URL http://localhost:9095/proyecto_bdmalacatos/v1/...
 */
class DbHandler {

    private $conn;

    function __construct(){
        require_once dirname(__FILE__) . '/DbConnect.php';
        // opening db connection
        $db = new DbConnect();
        $this->conn = $db->connect();
    }

    /* ----- `usuarios` table method ----- */
```

⇒ Crear un Nuevo Usuario

En este método vamos a poder crear un nuevo usuario teniendo como parámetros el nombre del usuario, dirección de correo electrónico y contraseña. Si el usuario no existe en la base de datos obtendrá un mensaje de confirmación que el usuario se ha creado satisfactoriamente y si el usuario a registrar ya consta en la base de datos obtendrá un mensaje que el usuario ya existe.

```
/**
 * Creating new user
 * @param String $usu_nombre User full usu_nombre
```

```

* @param String $usu_email User iniciar_sesion usu_email usu_id
* @param String $password User iniciar_sesion password
*/
public function createUser($usu_nombre, $usu_email, $password) {
    require_once 'PassHash.php';
    $response = array();

    // First check if user already existed in db
    if (!$this->isUserExists($usu_email)) {
        // Generating password hash
        $password_hash = PassHash::hash($password);

        // Generating API key
        $usu_api_key = $this->generateApiKey();

        // insert query
        $stmt = $this->conn->prepare("INSERT INTO usuarios(usu_nombre, usu_email,
password_hash, usu_api_key, usu_estado) values(?, ?, ?, ?, 1)");
        $stmt->bind_param("ssss", $usu_nombre, $usu_email, $password_hash, $usu_api_key);

        $result = $stmt->execute();

        $stmt->close();

        // Check for successful insertion
        if ($result) {
            // User successfully inserted
            return USER_CREATED_SUCCESSFULLY;
        } else {
            // Failed to create user
            return USER_CREATE_FAILED;
        }
    } else {
        // User with same usu_email already existed in the db
        return USER_ALREADY_EXISTED;
    }

    return $response;
}

```

⇒ Inicio de Sesión del Usuario

En este método se verificará si el usuario ingresa correctamente al sistema al ingresar su email y contraseña.

```
/**
 * Checking user iniciar_sesion
 * @param String $usu_email User iniciar_sesion usu_email usu_id
 * @param String $password User iniciar_sesion password
 * @return boolean User iniciar_sesion usu_estado success/fail
 */
public function checkLogin($usu_email, $password) {
    // fetching user by usu_email
    $stmt = $this->conn->prepare("SELECT password_hash FROM usuarios WHERE usu_email = ?");

    $stmt->bind_param("s", $usu_email);

    $stmt->execute();

    $stmt->bind_result($password_hash);

    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        // Found user with the usu_email
        // Now verify the password

        $stmt->fetch();

        $stmt->close();

        if (PassHash::check_password($password_hash, $password)) {
            // User password is correct
            return TRUE;
        } else {
            // user password is incorrect
            return FALSE;
        }
    }
}
```

```

    } else {
        $stmt->close();

        // user not existed with the usu_email
        return FALSE;
    }
}

/**
 * Checking for duplicate user by usu_email address
 * @param String $usu_email usu_email to check in db
 * @return boolean
 */
private function isUserExists($usu_email) {
    $stmt = $this->conn->prepare("SELECT usu_id from usuarios WHERE usu_email = ?");
    $stmt->bind_param("s", $usu_email);
    $stmt->execute();
    $stmt->store_result();
    $num_rows = $stmt->num_rows;
    $stmt->close();
    return $num_rows > 0;
}

/**
 * Fetching user by usu_email
 * @param String $usu_email User usu_email usu_id
 */
public function getUserByusu_email($usu_email) {
    $stmt = $this->conn->prepare("SELECT usu_nombre, usu_email, usu_api_key, usu_estado,
usu_fechacreacion FROM usuarios WHERE usu_email = ?");
    $stmt->bind_param("s", $usu_email);
    if ($stmt->execute()) {
        // $user = $stmt->get_result()->fetch_assoc();
        $stmt->bind_result($usu_nombre, $usu_email, $usu_api_key, $usu_estado,
$usu_fechacreacion);
        $stmt->fetch();
        $user = array();
        $user["usu_nombre"] = $usu_nombre;
        $user["usu_email"] = $usu_email;
    }
}

```

```

        $user["usu_api_key"] = $usu_api_key;
        $user["usu_estado"] = $usu_estado;
        $user["usu_fechacreacion"] = $usu_fechacreacion;
        $stmt->close();
        return $user;
    } else {
        return NULL;
    }
}

/**
 * Fetching user api key
 * @param String $usu_id user usu_id primary key in user table
 */
public function getApiKeyById($usu_id) {
    $stmt = $this->conn->prepare("SELECT usu_api_key FROM usuarios WHERE usu_id = ?");
    $stmt->bind_param("i", $usu_id);
    if ($stmt->execute()) {
        // $usu_api_key = $stmt->get_result()->fetch_assoc();
        // TODO
        $stmt->bind_result($usu_api_key);
        $stmt->close();
        return $usu_api_key;
    } else {
        return NULL;
    }
}

/**
 * Fetching user usu_id by api key
 * @param String $usu_api_key user api key
 */
public function getUserId($usu_api_key) {
    $stmt = $this->conn->prepare("SELECT usu_id FROM usuarios WHERE usu_api_key = ?");
    $stmt->bind_param("s", $usu_api_key);
    if ($stmt->execute()) {
        $stmt->bind_result($usu_id);
        $stmt->fetch();
    }
}

```

```

// TODO
// $usu_id = $stmt->get_result()->fetch_assoc();
$stmt->close();
return $usu_id;
} else {
    return NULL;
}
}
}
/**
 * Validating user api key
 * If the api key is there in db, it is a valid key
 * @param String $usu_api_key user api key
 * @return boolean
 */
public function isValidApiKey($usu_api_key) {
    $stmt = $this->conn->prepare("SELECT usu_id from usuarios WHERE usu_api_key = ?");
    $stmt->bind_param("s", $usu_api_key);
    $stmt->execute();
    $stmt->store_result();
    $num_rows = $stmt->num_rows;
    $stmt->close();
    return $num_rows > 0;
}

/**
 * Generating random Unique MD5 String for user Api key
 */
private function generateApiKey() {
    return md5(uniqid(rand(), true));
}

/* ----- `proyectos` table method ----- */

```

⇒ Crear un Nuevo Proyecto

```

/**
 * Creating new pro_proyecto
 * @param String $usu_id user pro_id to whom pro_proyecto belongs to
 * @param String $pro_proyecto pro_proyecto text

```

```

*/
public function createTask($usu_id, $pro_proyecto, $pro_descripcion) {

    $stmt = $this->conn->prepare("INSERT INTO proyectos(pro_proyecto, pro_descripcion)
VALUES(?,?)");
    $stmt->bind_param("ss", $pro_proyecto, $pro_descripcion);
    $result = $stmt->execute();
    $stmt->close();

    if ($result) {
        // pro_proyecto row created
        // now assign the pro_proyecto to user
        $new_task_id = $this->conn->insert_id;
        $res = $this->createUserTask($usu_id, $new_task_id);

        if ($res) {
            // pro_proyecto created successfully
            return $new_task_id;
        } else {
            // pro_proyecto failed to create
            return NULL;
        }
    } else {
        // pro_proyecto failed to create
        return NULL;
    }
}

/**
 * Fetching single pro_proyecto
 * @param String $pro_id pro_id of the pro_proyecto
 */
public function getTask($pro_id, $usu_id) {
    $stmt = $this->conn->prepare("SELECT t.pro_id, t.pro_proyecto, t.pro_estado,
t.pro_fecha_creacion from proyectos t, proyecto_usuario ut WHERE t.pro_id = ? AND ut.pro_id =
t.pro_id AND ut.usu_id = ?");
    $stmt->bind_param("ii", $pro_id, $usu_id);
    if ($stmt->execute()) {

```

```

$res = array();
$stmt->bind_result($pro_id, $pro_proyecto, $pro_estado, $pro_fecha_creacion);
// TODO
// $pro_proyecto = $stmt->get_result()->fetch_assoc();
$stmt->fetch();
$res["pro_id"] = $pro_id;
$res["pro_proyecto"] = $pro_proyecto;
$res["pro_estado"] = $pro_estado;
$res["pro_fecha_creacion"] = $pro_fecha_creacion;
$stmt->close();
return $res;
} else {
    return NULL;
}
}

/**
 * Fetching all user proyectos
 * @param String $usu_id pro_id of the user
 */
public function getAllUserTasks($usu_id) {
    $stmt = $this->conn->prepare("SELECT t.pro_proyecto FROM proyectos t, proyecto_usuario ut
WHERE t.pro_id = ut.pro_id AND ut.usu_id = ?");
    $stmt->bind_param("i", $usu_id);
    $stmt->execute();
    $proyectos = $stmt->get_result();
    $stmt->close();
    return $proyectos;
}

```

⇒ Modificar un Proyecto

```

/**
 * Updating pro_proyecto
 * @param String $pro_id pro_id of the pro_proyecto
 * @param String $pro_proyecto pro_proyecto text
 * @param String $pro_estado pro_proyecto pro_estado
 */
//No borrar public function updateTask($usu_id, $pro_id, $pro_proyecto, $pro_estado) {

```

```

public function updateTask($pro_id, $pro_proyecto, $pro_estado, $pro_descripcion) {
    //no borrar $stmt = $this->conn->prepare("UPDATE proyectos t, proyecto_usuario ut set
t.pro_proyecto = ?, t.pro_estado = ? WHERE t.pro_id = ? AND t.pro_id = ut.pro_id AND ut.usu_id = ?");
    $stmt = $this->conn->prepare("UPDATE proyectos p set p.pro_proyecto = ?, p.pro_estado = ?,
p.pro_descripcion = ? WHERE p.pro_id = ?");
    $stmt->bind_param("sisi", $pro_proyecto, $pro_estado, $pro_descripcion, $pro_id);
    $stmt->execute();
    $num_affected_rows = $stmt->affected_rows;
    $stmt->close();
    return $num_affected_rows > 0;
}

/**
 * Deleting a pro_proyecto
 * @param String $pro_id pro_id of the pro_proyecto to delete
 */
// public function deleteTask($usu_id, $pro_id) {
//     $stmt = $this->conn->prepare("DELETE t FROM proyectos t, proyecto_usuario ut WHERE
t.pro_id = ? AND ut.pro_id = t.pro_id AND ut.usu_id = ?");
//     $stmt->bind_param("ii", $pro_id, $usu_id);
//     $stmt->execute();
//     $num_affected_rows = $stmt->affected_rows;
//     $stmt->close();
//     return $num_affected_rows > 0;
// }

```

⇒ Eliminar un Proyecto

```

/**
 * Deleting a pro_proyecto
 * @param String $pro_id pro_id of the pro_proyecto to delete
 */
public function deleteTask($pro_id) {
    // $stmt = $this->conn->prepare("DELETE t FROM proyectos t, proyecto_usuario ut WHERE
t.pro_id = ? AND ut.pro_id = t.pro_id AND ut.usu_id = ?");
    $stmt = $this->conn->prepare("DELETE p FROM proyectos p WHERE p.pro_id = ?");
    $stmt->bind_param("i", $pro_id);
    $stmt->execute();
    $num_affected_rows = $stmt->affected_rows;
}

```

```

    $stmt->close();
    return $num_affected_rows > 0;
}
/* ----- `proyecto_usuario` table method ----- */

/**
 * Function to assign a pro_proyecto to user
 * @param String $usu_id usu_id of the user
 * @param String $pro_id pro_id of the pro_proyecto
 */
public function createUserTask($usu_id, $pro_id) {
    $stmt = $this->conn->prepare("INSERT INTO proyecto_usuario(usu_id, pro_id) values(?, ?)");
    $stmt->bind_param("ii", $usu_id, $pro_id);
    $result = $stmt->execute();
    if (false === $result) {
        die('execute() failed: ' . htmlspecialchars($stmt->error));
    }
    $stmt->close();
    return $result;
}

//sacar usu_id con el usu_email
public function getUserByusu_email2($usu_email) {
    $stmt = $this->conn->prepare("SELECT usu_id FROM usuarios WHERE usu_email = ?");
    $stmt->bind_param("s", $usu_email);
    if ($stmt->execute()) {
        // $user = $stmt->get_result()->fetch_assoc();
        $stmt->bind_result($usu_id);
        $stmt->fetch();
        // $user = array();
        // $user["usu_id"] = $usu_id;
        $user = $usu_id;
        $stmt->close();
        return $user;
    } else {
        return NULL;
    }
}
}
}

```

ANEXO 4.

HTML

Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas.

<!DOCTYPE>

En primer lugar se necesita indicar el tipo de documento que se está creando. Esto en HTML5 es extremadamente sencillo:

<!DOCTYPE html>

Esta línea debe ser la primera línea del archivo, sin espacios o líneas que la precedan. De esta forma, el modo estándar del navegador es activado y las incorporaciones de HTML5 son interpretadas siempre que sea posible, o ignoradas en caso contrario.

Luego de declarar el tipo de documento, se debe comenzar a construir la estructura HTML. Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento <html>. Este elemento envolverá al resto del código:

<!DOCTYPE html>

<html lang="es">

</html>

HTML usa un lenguaje de etiquetas para construir páginas web. Estas etiquetas HTML son palabras clave y atributos rodeados de los signos mayor y menor (por ejemplo, <html lang="es">). Existen las etiquetas de apertura y cierre y verá que la de cierre se distingue por una barra invertida antes de la palabra clave (por ejemplo, </html>). El resto de nuestro código será insertado entre estas dos etiquetas: <html>...</html>.

El código HTML insertado entre las etiquetas <html> tiene que ser dividido entre dos secciones principales. Al igual que en versiones previas de HTML, la primera

sección es la cabecera y la segunda el cuerpo. El siguiente paso, por lo tanto, será crear estas dos secciones en el código usando los elementos `<head>` y `<body>` ya conocidos.

El elemento `<head>` va primero, por supuesto, y al igual que el resto de los elementos estructurales tiene una etiqueta de apertura y una de cierre:

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
</head>  
</html>
```

La etiqueta no cambió desde versiones anteriores y su propósito sigue siendo exactamente el mismo. Dentro de las etiquetas `<head>` se definirá el título de nuestra página web, además se declarará el set de caracteres correspondiente, se proveerá información general acerca del documento y se incorporarán los archivos externos con estilos, códigos.

JavaScript o incluso imágenes necesarias para generar la página en la pantalla.

Excepto por el título y algunos íconos, el resto de la información incorporada en el documento entre estas etiquetas es invisible para el usuario.

La siguiente gran sección que es parte principal de la organización de un documento HTML es el cuerpo. El cuerpo representa la parte visible de todo documento y es especificado entre etiquetas `<body>`. Estas etiquetas tampoco han cambiado en relación con versiones previas de HTML:

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
</head>  
<body>  
</body>  
</html>
```

Es momento de construir la cabecera del documento. Algunos cambios e innovaciones fueron incorporados dentro de la cabecera, y uno de ellos es la etiqueta que define el juego de caracteres a utilizar para mostrar el documento. Ésta es una etiqueta **<meta>** que especifica cómo el texto será presentado en pantalla:

```
<!DOCTYPE html>
<html lang="es">
<head>
<metacharset="iso-8859-1">
</head>
<body>
</body>
</html>
```

Posteriormente la etiqueta **<title>**, como siempre, simplemente especifica el título del documento, y no hay nada nuevo para comentar:

```
<html lang="es">
<head>
<meta charset="iso-8859-1">
<meta name="description" content="Ejemplo de HTML5">
<meta name="keywords" content="HTML5, CSS3, JavaScript">
<title>Este texto es el título del documento</title>
</head>
<body>
</body>
</html>
```

El texto entre las etiquetas **<title>** es el título del documento que se está creando. Normalmente este texto es mostrado en la barra superior de la ventana del navegador.

Estructura Del Cuerpo

La estructura del cuerpo (el código entre las etiquetas <body>) generará la parte visible del documento. Este es el código que producirá nuestra página web.

HTML siempre ofreció diferentes formas de construir como <table> que permitían a los diseñadores acomodar datos, texto, imágenes y herramientas dentro de filas y columnas de celdas, incluso sin que hayan sido concebidas para este propósito.

Con el elemento <div> surgieron las webs más interactivas y la integración de HTML, CSS y JavaScript, el uso de <div>se volvió una práctica común.

En HTML5 las secciones más importantes son diferenciadas y la estructura principal ya no depende más de los elementos <div>o <table>.

Referencia Rápida

Según (Gauchat, 2012)En la especificación HTML5, HTML está a cargo de la estructura del documento y provee un grupo completo de nuevos elementos para este propósito. La especificación también incluye algunos elementos con la única tarea de proveer estilos.

Esta es una lista de los que se considera más relevante:

Para una completa referencia de los elementos HTML incluidos en la especificación, visite nuestro sitio web y siga los enlaces correspondientes a este capítulo.

- **<header>**Este elemento presenta información introductoria y puede ser aplicado en diferentes secciones del documento. Tiene el propósito de contener la cabecera de una sección pero también puede ser utilizado para agrupar índices, formularios de búsqueda, logos, etc...
- **<nav>**Este elemento indica una sección de enlaces con propósitos de navegación, como menús o índices. No todos los enlaces dentro de una página

web tienen que estar dentro de un elemento `<nav>`, solo aquellos que forman partes de bloques de navegación.

- **`<section>`** Este elemento representa una sección general del documento. Es usualmente utilizado para construir varios bloques de contenido (por ejemplo, columnas) con el propósito de ordenar el contenido que comparte una característica específica, como capítulos o páginas de un libro, grupo de noticias, artículos, etc
- **`<aside>`** Este elemento representa contenido que está relacionado con el contenido principal pero no es parte del mismo. Ejemplos pueden ser citas, información en barras laterales, publicidad, etc
- **`<footer>`** Este elemento representa información adicional sobre su elemento padre. Por ejemplo, un elemento
- **`<footer>`** insertado al final del cuerpo proveerá información adicional sobre el cuerpo del documento, como el pie normal de una página web. Puede ser usado no solo para el cuerpo sino también para diferentes secciones dentro del cuerpo, otorgando información adicional sobre estas secciones específicas.
- **`<article>`** Este elemento representa una porción independiente de información relevante (por ejemplo, cada artículo de un periódico o cada entrada de un blog). El elemento `<article>` puede ser anidado y usado para mostrar una lista dentro de otra lista de ítems relacionados, como comentarios de usuarios en entradas de blogs.
- **`<hgroup>`** Este elemento es usado para agrupar elementos H cuando la cabecera tiene múltiples niveles (por ejemplo, una cabecera con título y subtítulo).
- **`<figure>`** Este elemento representa una porción independiente de contenido (por ejemplo, imágenes, diagramas o videos) que son referenciadas desde el contenido principal. Esta es información que puede ser removida sin afectar el fluido del resto del contenido.

- **<figcaption>**Este elemento es utilizado para mostrar una leyenda o pequeño texto relacionado con el contenido de un elemento **<figure>**, como la descripción de una imagen.
- **<mark>**Este elemento resalta un texto que tiene relevancia en una situación en particular o que ha sido mostrado en respuesta de la actividad del usuario.
- **<small>**Este elemento representa contenido al margen, como letra pequeña (por ejemplo, descargos, restricciones legales, declaración de derechos, etc...).
- **<cite>**Este elemento es usado para mostrar el título de un trabajo (libro, película, poema, etc...).
- **<address>**Este elemento encierra información de contacto para un elemento **<article>**o el documento completo. Es recomendable que sea insertado dentro de un elemento **<footer>**.
- **<time>**Este elemento se utiliza para mostrar fecha y hora en formatos comprensibles por los usuarios y el navegador.

El valor para los usuarios es ubicado entre las etiquetas mientras que el específico para programas y navegadores es incluido como el valor del atributo `datetime`. Un segundo atributo optativo llamado `pubdate` es usado para indicar que el valor de `datetime` es la fecha de publicación.

CSS Y HTML

CSS tiene un set predeterminado de propiedades destinados a sobrescribir los estilos provistos por navegadores y obtener la organización deseada. Estas propiedades no son específicas, tienen que ser combinadas para formar reglas que luego serán usadas para agrupar cajas y obtener la correcta disposición en pantalla. La combinación de estas reglas es normalmente llamada modelo o sistema de disposición. Todas estas reglas aplicadas juntas constituyen lo que se llama un modelo de caja.

Existe solo un modelo de caja que es considerado estándar estos días, y muchos otros que aún se encuentran en estado experimental. El modelo válido y ampliamente adoptado es el llamado Modelo de Caja Tradicional, el cual ha sido usado desde la primera versión de CSS.

Aunque este modelo ha probado ser efectivo, algunos modelos experimentales intentan superar sus deficiencias, pero la falta de consenso sobre el reemplazo más adecuado aún mantiene a este viejo modelo en vigencia y la mayoría de los sitios webs programados en HTML5 lo continúan utilizando.

Conceptos básicos sobre estilos

Antes de comenzar a insertar reglas CSS en nuestro archivo de estilos y aplicar un modelo de caja, debemos revisar los conceptos básicos sobre estilos CSS que van a ser utilizados en el resto del libro.

Aplicar estilos a los elementos HTML cambia la forma en que estos son presentados en pantalla. Como explicamos anteriormente, los navegadores proveen estilos por defecto que en la mayoría de los casos no son suficientes para satisfacer las necesidades de los diseñadores. Para cambiar esto, podemos sobrescribir estos estilos con los nuestros usando diferentes técnicas.

Estilos en línea

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo style.

A continuación se muestra un documento HTML simple que contiene el elemento <p>modificado por el atributo style con el valor font-size: 20px. Este estilo cambia el tamaño por defecto del texto dentro del elemento <p>a un nuevo tamaño de 20 píxeles.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este es el título del documento</title>
</head>
<body>
<p style="font-size: 20px">Mi texto</p>
</body>
</html>
```

Usar la técnica demostrada anteriormente es una buena manera de probar estilos y obtener una vista rápida de sus efectos, pero no es recomendado para aplicar estilos a todo el documento. La razón es simple: cuando usamos esta técnica, debemos escribir y repetir cada estilo en cada uno de los elementos que queremos modificar, incrementando el tamaño del documento a proporciones inaceptables y haciéndolo imposible de mantener y actualizar. Solo imagine lo que ocurriría si decide que en lugar de 20 píxeles el tamaño de cada uno de los elementos <p>debería ser de 24 píxeles.

Tendría que modificar cada estilo en cada etiqueta <p>en el documento completo.

Métodos

(Gauchat, 2012) Existen varios métodos para seleccionar cuáles elementos HTML serán afectados por las reglas CSS:

- referencia por la palabra clave del elemento
- referencia por el atributo id
- referencia por el atributo class

Referenciando con palabra clave

Al declarar las reglas CSS utilizando la palabra clave del elemento afectamos cada elemento de la misma clase en el documento. Por ejemplo, la siguiente regla cambiará los estilos de todos los elementos `<p>`:

```
p { font-size: 20px }
```

Esta es la técnica presentada previamente en el Listado 2-2. Utilizando la palabra clave `p` al frente de la regla le estamos diciendo al navegador que esta regla debe ser aplicada a cada elemento `<p>` encontrado en el documento HTML. Todos los textos envueltos en etiquetas `<p>` tendrán el tamaño de 20 pixeles.

Por supuesto, lo mismo funcionará para cualquier otro elemento HTML. Si especificamos la palabra clave `span` en lugar de `p`, por ejemplo, cada texto entre etiquetas `` tendrá un tamaño de 20 pixeles:

```
span { font-size: 20px }
```

Referenciando con el atributo id

El atributo `id` es como un nombre que identifica al elemento. Esto significa que el valor de este atributo no puede ser duplicado. Este nombre debe ser único en todo el documento.

Para referenciar un elemento en particular usando el atributo `id` desde nuestro archivo CSS la regla debe ser declarada con el símbolo `#` al frente del valor que usamos para identificar el elemento:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<p id="texto1">Mi texto</p>
</body>
</html>
```

El resultado de este procedimiento es que cada vez que hacemos una referencia usando el identificador `texto1` en nuestro archivo CSS, el elemento con ese valor de `id` será modificado, pero el resto de los elementos `<p>`, o cualquier otro elemento en el mismo documento, no serán afectados.

Esta es una forma extremadamente específica de referenciar un elemento y es normalmente utilizada para elementos más generales, como etiquetas estructurales. El atributo `id` y su especificidad es de hecho más apropiado para referencias en JavaScript, como veremos en próximos capítulos.

Referenciando con el atributo `class`

La mayoría del tiempo, en lugar de utilizar el atributo `id` para propósitos de estilos es mejor utilizar `class`. Este atributo es más flexible y puede ser asignado a cada elemento HTML en el documento que comparte un diseño similar:

```
.texto1 { font-size: 20px }
```

Para trabajar con el atributo `class`, debemos declarar la regla CSS con un punto antes del nombre. La ventaja de este método es que insertar el atributo `class` con el valor `texto1` será suficiente para asignar estos estilos a cualquier elemento que queramos:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<p class="texto1">Mi texto</p>
<p class="texto1">Mi texto</p>
<p>Mi texto</p>
</body>
</html>
```

Los elementos `<p>` en las primeras dos líneas dentro del cuerpo del código en el Listado 2-9 tienen el atributo `class` con el valor `texto1`. Como dijimos previamente, la misma regla puede ser aplicada a diferentes elementos en el mismo documento.

```
p.texto1 { font-size: 20px }
```

Aquí creamos una regla que referencia la clase llamada `texto1` pero solo para los elementos de tipo `<p>`.

Si cualquier otro elemento tiene el mismo valor en su atributo `class` no será modificado por esta regla en particular.

Referenciando con cualquier atributo

Aunque los métodos de referencia estudiados anteriormente cubren un variado espectro de situaciones, a veces no son suficientes para encontrar el elemento exacto. La última

versión de CSS ha incorporado nuevas formas de referenciar elementos HTML. Uno de ellas es el Selector de Atributo. Ahora podemos referenciar un elemento no solo por los atributos id y class sino también a través de cualquier otro atributo:

```
p[name] { font-size: 20px }
```

Se cambia solo elementos <p>que tienen un atributo llamado name. Para imitar lo que hicimos previamente con los atributos id y class, podemos también especificar el valor del atributo:

```
p[name="mitexto"] { font-size: 20px }
```

segúnJavaScript (McFarland, 2011)

Aplicando CSS a La Plantilla

Todo elemento estructural es considerado una caja y la estructura completa es presentada como un grupo de cajas. Las cajas agrupadas constituyen lo que es llamado un Modelo de Caja.

Siguiendo con los conceptos básicos de CSS, vamos a estudiar lo que es llamado el Modelo de Caja Tradicional. Este modelo has sido implementado desde la primera versión de CSS y es actualmente soportado por cada navegador en el mercado, lo que lo ha convertido en un estándar para el diseño web.

Todo modelo, incluso aquellos aún en fase experimental, pueden ser aplicados a la misma estructura HTML, pero esta estructura debe ser preparada para ser afectada por estos estilos de forma adecuada. Nuestros documentos HTML deberán ser adaptados al modelo de caja seleccionado.

JAVA SCRIPT

Manejadores De Eventos

El código JavaScript es normalmente ejecutado luego de que el usuario realiza alguna acción. Estas acciones y otros eventos son procesados por manejadores de eventos y funciones JavaScript asociadas con ellos.

Existen tres diferentes formas de registrar un evento para un elemento HTML: podemos agregar un nuevo atributo al elemento, registrar un manejador de evento como una propiedad del elemento o usar el nuevo método estándar `addEventListener ()`.

Conceptos básicos: En JavaScript las acciones de los usuarios son llamadas eventos. Cuando el usuario realiza una acción, como un clic del ratón o la presión de una tecla, un evento específico para cada acción y cada elemento es disparado. Además de los eventos producidos por los usuarios existen también otros eventos disparados por el sistema (por ejemplo, el evento `load` que se dispara cuando el documento es completamente cargado).

Estos eventos son manejados por códigos o funciones. El código que responde al evento es llamado manejador. Cuando registramos un manejador lo que hacemos es definir cómo nuestra aplicación responderá a un evento en particular. Luego de la estandarización del método `addEventListener ()`, este procedimiento es usualmente llamado “escuchar al evento”, y lo que hacemos para preparar el código que responderá a ese evento es “agregar una escucha” a un elemento en particular.

Manejadores De Eventos En Línea

Se trata simplemente de utilizar los atributos provistos por HTML para registrar eventos para un elemento en particular. Esta es una técnica en desuso pero aun

extremadamente útil y práctica en algunas circunstancias, especialmente cuando necesitamos hacer modificaciones rápidas para testeo.

Manejadores De Eventos Como Propiedades

Para evitar las complicaciones de la técnica en línea (inline), debemos registrar los eventos desde el código JavaScript.

Usando selectores JavaScript podemos referenciar el elemento HTML y asignarle el manejador de eventos que queremos como si fuese una propiedad. Los nombres de los manejadores de eventos son contruidos agregando el prefijo on al nombre del evento. Por ejemplo, el nombre del manejador de eventos para el evento click es onclick. Cuando estamos hablando sobre onclick usualmente hacemos referencia al código que será ejecutado cuando el evento click se produzca.

Antes de HTML5, esta era la única técnica disponible para usar manejadores de eventos desde JavaScript que funcionaba en todos los navegadores. Algunos fabricantes de navegadores estaban desarrollando sus propios sistemas, pero nada fue adoptado por todos hasta que el nuevo estándar fue declarado. Por este motivo recomendamos utilizar esta técnica por razones de compatibilidad, pero no la sugerimos para sus aplicaciones HTML5.