



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DE TÍTULO DE
INGENIERÍA EN SISTEMAS COMPUTACIONALES**

TEMA

***“DESARROLLO DE UNA HERRAMIENTA DIDÁCTICA PARA LA GESTIÓN DE
PROYECTOS MEDIANTE LA MATRIZ DE MARCO LÓGICO”***

AUTOR

BASTIDAS GUERRA MILTON FABIAN

DIRECTOR

ING. PEDRO GRANDA

IBARRA – ECUADOR



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AUTORIZACIÓN DE USO Y PUBLICACIÓN

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

DATOS DE CONTACTO	
CEDULA DE IDENTIDAD	100371993-5
APELLIDOS Y NOMBRES	Bastidas Guerra Milton Fabian
DIRECCIÓN	Ibarra, Manuela Sáenz y María Esther León
EMAIL	mfbastidas@utn.edu.ec
TELÉFONO FIJO	022 632 565
TELÉFONO MÓVIL	0992627600

DATOS DE LA OBRA	
TÍTULO	“DESARROLLO DE UNA HERRAMIENTA DIDÁCTICA PARA LA GESTIÓN DE PROYECTOS MEDIANTE LA MATRIZ DE MARCO LÓGICO”
AUTOR	Bastidas Guerra Milton Fabian
FECHA	Enero del 2016
PROGRAMA	Pregrado
TITULO POR EL QUE OPTA	Ingeniería en Sistemas Computacionales
DIRECTOR	Ing. Pedro Granda

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, BASTIDAS GUERRA MILTON FABIAN, portador de la cédula de ciudadanía N° 100371993-5, en calidad de autor y titular de los derechos patrimoniales de la obra o Trabajo de Grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143



Bastidas Guerra Milton Fabián

CI: 100371993-5

Ibarra, Enero del 2016



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Bastidas Guerra Milton Fabián
C.I. 100571995-5

Ibarra, Enero del 2016



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Bastidas Guerra Milton Fabian, portador de la cédula de ciudadanía N° 100371993-5 manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6 en calidad de autor del Trabajo de Grado denominado: **“DESARROLLO DE UNA HERRAMIENTA DIDÁCTICA PARA LA GESTIÓN DE PROYECTOS MEDIANTE LA MATRIZ DE MARCO LÓGICO”**, con el aplicativo: **”MML Gestión de proyectos mediante marco lógico”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de a la Universidad Técnica del Norte de la ciudad de Ibarra.

Bastidas Guerra Milton Fabián

Bastidas Guerra Milton Fabian

CI: 100371993-5

Ibarra, Enero del 2016



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

Certifico que la tesis “**DESARROLLO DE UNA HERRAMIENTA DIDÁCTICA PARA LA GESTIÓN DE PROYECTOS MEDIANTE LA MATRIZ DE MARCO LÓGICO**”, ha o realizado en su totalidad por el señor: Bastidas Guerra Milton Fabian, portador de la cedula de identidad número: 100371993 – 5.

Ing. Pedro Granda

Director de tesis



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

En primer lugar a mis padres quienes dándome el regalo más preciado que es la vida, me han permitido ser una persona de bien; les agradezco por brindarme su apoyo a lo largo de mi carrera y crecimiento tanto personal como profesional.

Además a mis hermanos que como buenos amigos supieron darme sus consejos, y compartir junto a mí gratas experiencias.

Un agradecimiento especial a mis profesores que día tras día supieron exigirme para superar mis propias barreras, motivarme a dar lo mejor de mí.

Gracias a todas estas personas que invirtieron su valioso tiempo para guiarme de la mejor manera posible.



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

Este trabajo se lo dedico a mis padres que con tanto amor, esmero y siempre velando por mi bienestar me han sabido educar; para ellos que con gran satisfacción me verán cumplir uno de sus sueños al llegar a esta meta.

A todas las personas que me acompañaron y ayudaron con su experiencia a lo largo de este camino que se ha convertido en un viaje de crecimiento profesional y personal.

Finalmente dedico el fruto de mi esfuerzo a una personal especial que ha estado estos últimos años a mi lado, ofreciéndome todo su cariño y comprensión; además de ser quien ha inspirado mis ganas de seguir adelante cada día.

RESUMEN

El presente proyecto trata sobre el desarrollo de una herramienta informática para la generación de proyectos mediante la metodología de **Matriz de Marco Lógico**, haciendo uso de nuevas tecnologías como: **NodeJS**, **AngularJS** y **MongoDB** y desplegada en la nube de **Openshift**.

Se explica la metodología que usa la matriz de marco lógico así como los pasos que usa para generar proyectos de calidad.

Dentro del software se explica las nuevas herramientas y sus ventajas sobre las tradicionales dando por resultado un sistema web sobre una plataforma en la nube accesible para cualquier persona con acceso a internet.

Capítulo 1: En este capítulo se da una introducción sobre el proyecto definiendo el problema, los objetivos y el alcance del mismo.

Capítulo 2: En esta sección se definirán las herramientas o tecnologías: **NodeJS**, **AngularJS**, **MongoDB** y **Openshift** usadas para el desarrollo del sistema informático, así como una breve introducción de la metodología de desarrollo de **programación extrema (XP)**.

Capítulo 3: En este capítulo se hablara sobre la metodología de gestión de proyectos **Matriz de Marco Lógico**, su definición y forma de aplicación.

Capítulo 4: En esta sección del documento se mostrara los pasos de desarrollo del aplicativo en base a las etapas de la metodología de **programación extrema (XP)**.

Capítulo 5: En el capítulo final se definen las conclusiones y recomendaciones resultantes del desarrollo de este proyecto.

ABSTRACT

This project is about the development of a software tool for generating projects through the methodology of **Logical Framework**, making use of new technologies such as: **NodeJS, angularjs and MongoDB** and deployed in the cloud **OpenShift**.

The methodology of Logical Framework and its steps are explained used to make quality projects.

The software part it is about the new tools and their advantages over the traditional ones resulting in a web system on a platform in the cloud accessible to anyone with Internet access.

Dentro del software se explica las nuevas herramientas y sus ventajas sobre las tradicionales dando por resultado un sistema web sobre una plataforma en la nube accesible para cualquier persona con acceso a internet.

Chapter 1: This chapter gives an introduction to the project by defining the problem, objectives and scope.

Chapter 2: This part defines the tools: **NodeJS, angularjs, MongoDB and OpenShift** used to develop the computer system, and a brief introduction of the development methodology Extreme Programming (XP)

Chapter 3: This chapter will discuss the methodology for managing projects **Logical Framework Matrix**, its definition and application form.

Chapter 4: This section of the document will display the application development steps based on the method of extreme programming (XP).

Chapter 5: The final chapter presents the conclusions and recommendations resulting from the development of this project.

ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN	ii
CONSTANCIA	iv
SESIÓN DE DERECHOS	v
CERTIFICACIÓN	vi
AGRADECIMIENTO	vii
DEDICATORIA	viii
RESUMEN	ix
ABSTRACT	x
ÍNDICE DE CONTENIDOS	xi
ÍNDICE DE TABLAS	xvii
ÍNDICE DE FIGURAS	xxiv
1 INTRODUCCIÓN	2
1.1 ANTECEDENTES	2
1.2 SITUACIÓN ACTUAL	2
1.3 PROBLEMA	3
1.4 JUSTIFICACIÓN	4

1.5	OBJETIVOS	5
1.5.1	Objetivo General	5
1.5.2	Objetivos Específicos	5
1.6	ALCANCE	5
2	MARCO TEÓRICO	14
2.1	METODOLOGÍA XP (PROGRAMACIÓN EXTREMA)	14
2.1.1	Ciclo de vida	15
2.1.2	Valores	16
2.1.3	Artefactos	18
2.2	PLATAFORMA COMO SERVICIO (PAAS)	22
2.2.1	Características	26
2.2.2	Ret Hat Openshift	28
2.3	NODE.JS	35
2.3.1	Modelo orientado a eventos	37
2.3.2	Modelo non-blocking	38
2.3.3	Node.js packages	41
2.4	ANGULAR.JS	45

2.4.1	¿Qué es MVC? _____	48
2.4.2	¿Porque usar AngularJS? _____	50
2.4.3	Filosofía _____	51
2.5	MONGODB _____	56
2.5.1	Documentos _____	62
2.6	SERVICIOS REST (Representational State Transfer) _____	65
2.6.1	Servicios RESTful _____	69
2.6.2	REST vs SOAP _____	71
3	MARCO LÓGICO _____	75
3.1	DEFINICIÓN _____	75
3.2	METODOLOGÍA _____	76
3.2.1	Análisis de involucrados _____	78
3.2.2	Análisis del problema _____	78
3.2.3	Análisis de objetivos _____	80
3.2.4	Análisis de alternativas _____	82
3.2.5	Estructura analítica del proyecto _____	85
3.2.6	Matriz de marco lógico _____	86

3.2.7	Indicadores	90
3.2.8	Medios de verificación	91
3.2.9	Hipótesis o supuestos	92
3.2.10	Evaluación intermedia	92
3.3	HERRAMIENTAS	93
4	DESARROLLO	96
4.1	PLANIFICACIÓN	96
4.1.1	Historias de usuario	96
4.1.2	Plan de iteraciones (Iteration plan)	109
4.1.3	Plan de despliegue (Release plan)	137
4.2	ANÁLISIS DE REQUERIMIENTOS	144
4.2.1	Requerimientos funcionales	145
4.2.2	Requerimientos no funcionales	145
4.3	DISEÑO	146
4.3.1	Iteración 1	146
4.3.2	Iteración 2	148
4.3.3	Iteración 3	152

4.3.4	Iteración 4	154
4.3.5	Iteración 5	158
4.3.6	Iteración 6	164
4.3.7	Iteración 7	170
4.3.8	Iteración 8	176
4.4	DESARROLLO	180
4.4.1	Iteración 5	180
4.4.2	Iteración 6	183
4.4.3	Iteración 7	186
4.4.4	Iteración 8	190
4.5	PRUEBAS	191
4.5.1	Historia de usuario 1 – Control de usuarios	191
4.5.2	Historia de usuario 2 – Gestión de proyectos	195
4.5.3	Historia de usuario 3 – Análisis de involucrados	202
4.5.4	Historia de usuario 4 – Análisis del problema	206
4.5.5	Historia de usuario 5 – Análisis de objetivos	210
4.5.6	Historia de usuario 6 – Análisis de alternativas	211

4.5.7	Historia de usuario 7 – Generación de la matriz de marco lógico	214
4.5.8	Historia de usuario 8 – Gestión de seguimiento del proyecto	216
4.5.9	Historia de usuario 9 – Generación de reportes y ayuda	221
5	ANÁLISIS, CONCLUSIONES Y RECOMENDACIONES	226
5.1	ANÁLISIS DE IMPACTO	226
5.1.1	Social	226
5.1.2	Económico	228
5.1.3	Ambiental	230
5.2	CONCLUSIONES	233
5.3	RECOMENDACIONES	236
	BIBLIOGRAFÍA	238
	ANEXOS	240

ÍNDICE DE TABLAS

Tabla 2.1 Ventajas y desventajas de los servicios PAAS	26
Tabla 2.2 Resumen de características de los servicios de Openshift	34
Tabla 2.3 Tabla clasificatoria de las base de datos.	59
Tabla 2.4 Diferencias entre mongo y otros tipos de bases de datos.	61
Tabla 2.5 Métodos básicos del protocolo HTTP	67
Tabla 2.6 Diferencias entre SOAP y REST	72
Tabla 4.1 Historia de Usuario 1 – Control de usuarios	97
Tabla 4.2 Historia de Usuario 2 – Gestión de proyectos	99
Tabla 4.3 Historia de Usuario 3 – Análisis de involucrados	100
Tabla 4.4 Historia de Usuario 4 – Análisis del problema	101
Tabla 4.5 Historia de Usuario 5 – Análisis de objetivos	103
Tabla 4.6 Historia de Usuario 6 – Análisis de alternativas	104
Tabla 4.7 Historia de Usuario 7 – Generación de la matriz de marco lógico	106
Tabla 4.8 Historia de Usuario 8 – Gestión de indicadores, medios de verificación y supuestos en la matriz de marco lógico	108
Tabla 4.9 Historia de Usuario 9 – Generación de reportes de la matriz de marco lógico	109

Tabla 4.10 Historia de Usuario 1, Tarea 1 – Diseño de modelo de datos	_110
Tabla 4.11 Historia de Usuario 1, Tarea 2 – Definición de servicios REST para usuarios	_____111
Tabla 4.12 Historia de Usuario 1, Tarea 3 – Formulario de registro de usuarios	_____112
Tabla 4.13 Historia de Usuario 1, Tarea 4 – Formulario para acceso de usuarios registrados	_____113
Tabla 4.14 Historia de Usuario 2, Tarea 1 – Diseño de modelo de datos	_113
Tabla 4.15 Historia de Usuario 1, Tarea 2 – Definición de servicios REST para gestión de proyectos	_____114
Tabla 4.16 Historia de Usuario 2, Tarea 3 – Mostrar los proyectos asociados al usuario	_____115
Tabla 4.17 Historia de Usuario 2, Tarea 4 – Crear proyectos	_____116
Tabla 4.18 Historia de Usuario 2, Tarea 5 – Eliminar proyectos	_____117
Tabla 4.19 Historia de Usuario 2, Tarea 6 – Compartir proyectos	_____118
Tabla 4.20 Historia de Usuario 2, Tarea 7 – Historial de cambios	_____118
Tabla 4.21 Historia de Usuario 3, Tarea 1 – Diseño de modelo de datos	_119
Tabla 4.22 Historia de Usuario 3, Tarea 2 – Definición de servicios REST para la gestión involucrados	_____120

Tabla 4.23 Historia de Usuario 3, Tarea 3 – Gestión de involucrados _____	121
Tabla 4.24 Historia de Usuario 4, Tarea 1 – Diseño de modelo de datos _	122
Tabla 4.25 Historia de Usuario 4, Tarea 2 – Definición de servicios REST para la gestión de problemas _____	123
Tabla 4.26 Historia de Usuario 4, Tarea 3 – Gestión del árbol de problemas	123
Tabla 4.27 Historia de Usuario 5, Tarea 1 – Diseño de modelo de datos _	124
Tabla 4.28 Historia de Usuario 5, Tarea 2 – Definición de servicios REST para la gestión de objetivos _____	125
Tabla 4.29 Historia de Usuario 5, Tarea 3 – Gestión del árbol de objetivos	126
Tabla 4.30 Historia de Usuario 6, Tarea 1 – Diseño de modelo de datos _	127
Tabla 4.31 Historia de Usuario 6, Tarea 2 – Definición de servicios REST para la gestión de alternativas _____	127
Tabla 4.32 Historia de Usuario 6, Tarea 3 – Gestión de acciones _____	128
Tabla 4.33 Historia de Usuario 6, Tarea 4 – Gestión de alternativas _____	129
Tabla 4.34 Historia de Usuario 7, Tarea 1 – Diseño del modelo de datos _	130
Tabla 4.35 Historia de Usuario 7, Tarea 2 – Definición de servicios REST para la matriz de marco lógico _____	131

Tabla 4.36 Historia de Usuario 7, Tarea 3 – Gestión de la primera columna de matriz de marco lógico _____	131
Tabla 4.37 Historia de Usuario 8, Tarea 1 – Diseño del modelo de datos _____	132
Tabla 4.38 Historia de Usuario 8, Tarea 2 – Definición de servicios REST para seguimiento de la matriz de marco lógico _____	133
Tabla 4.39 Historia de Usuario 8, Tarea 3 – Gestión de indicadores _____	134
Tabla 4.40 Historia de Usuario 8, Tarea 4 – Gestión de medios de verificación _____	135
Tabla 4.41 Historia de Usuario 8, Tarea 5 – Gestión de supuestos _____	135
Tabla 4.42 Historia de Usuario 8, Tarea 4 – Gestión de supuestos _____	136
Tabla 4.43 Historia de Usuario 8, Tarea 4 – Gestión de supuestos _____	137
Tabla 4.44 Plan de ejecución de las tareas _____	140
Tabla 4.45 Servicios REST de usuario _____	149
Tabla 4.46 Servicios REST de MML _____	150
Tabla 4.47 Rutas de usuario _____	151
Tabla 4.48 Rutas de usuario _____	152
Tabla 4.49 Rutas de usuario _____	152
Tabla 4.50 Rutas de gestión de involucrados _____	153

Tabla 4.51 Rutas de gestión del problema _____	153
Tabla 4.52 Rutas de gestión de objetivos _____	154
Tabla 4.53 Rutas de usuario _____	155
Tabla 4.54 Rutas de gestión de involucrados _____	156
Tabla 4.55 Rutas de gestión de alternativas _____	156
Tabla 4.56 Rutas de generación de la matriz _____	157
Tabla 4.57 Rutas de gestión del problema _____	157
Tabla 4.58 Control de usuarios – validar acceso _____	192
Tabla 4.59 Control de usuarios – acceso _____	193
Tabla 4.60 Control de usuarios – validar registro _____	194
Tabla 4.61 Control de usuarios – registro _____	195
Tabla 4.62 Gestión de proyectos – validar nuevo proyecto _____	196
Tabla 4.63 Gestión de proyectos – validar nuevo proyecto _____	197
Tabla 4.64 Gestión de proyectos – validar nuevo proyecto _____	198
Tabla 4.65 Gestión de proyectos – validar nuevo proyecto _____	199
Tabla 4.66 Gestión de proyectos – validar nuevo proyecto _____	200
Tabla 4.67 Gestión de proyectos – validar nuevo proyecto _____	201

Tabla 4.68 Gestión de proyectos – validar nuevo proyecto _____	202
Tabla 4.69 Análisis de Involucrados – validar nuevo involucrado _____	202
Tabla 4.70 Análisis de Involucrados – validar nuevo involucrado _____	203
Tabla 4.71 Análisis de Involucrados – insertar nuevo involucrado_____	204
Tabla 4.72 Análisis de Involucrados – insertar nuevo involucrado_____	205
Tabla 4.73 Análisis de Involucrados – insertar nuevo involucrado_____	206
Tabla 4.74 Análisis de Problema – validar causa-efecto _____	207
Tabla 4.75 Análisis de Problema – insertar causa-efecto _____	208
Tabla 4.76 Análisis de Problema – editar causa-efecto _____	208
Tabla 4.77 Análisis de Problema – validar problema central _____	209
Tabla 4.78 Análisis de Objetivos – validar objetivo central_____	210
Tabla 4.79 Análisis de Objetivos – validar cambio de causa-efecto a medio-fin _____	211
Tabla 4.80 Análisis de Alternativas – validar generación de acciones _____	212
Tabla 4.81 Análisis de Alternativas – validar objetivo central_____	213
Tabla 4.82 Generación de la matriz de marco lógico – validar objetivo central	214

Tabla 4.83 Generación de la matriz de marco lógico – validar no edición de proyecto creado	215
Tabla 4.84 Generación de la matriz de marco lógico – validar al menos el indicador del objetivo principal	216
Tabla 4.85 Generación de seguimiento del proyecto – evaluaciones	217
Tabla 4.86 Generación de seguimiento del proyecto – insertar evaluaciones	218
Tabla 4.87 Generación de seguimiento del proyecto – validar insertar pruebas	219
Tabla 4.88 Generación de seguimiento del proyecto – no insertar más evaluaciones después del fin del indicador	220
Tabla 4.89 Generación de seguimiento del proyecto – insertar indicador	221
Tabla 4.90 Generación de reportes y ayuda – Solo acceder cuando el proyecto este generado	222
Tabla 4.91 Generación de reportes y ayuda – Descargar PDF	223
Tabla 4.92 Generación de reportes y ayuda – Descargar DOC	224
Tabla 5.1 Tabla de medición de indicadores sociales	226
Tabla 5.2 Tabla de medición de indicadores económicos	229
Tabla 5.3 Tabla de medición de indicadores ambientales	231

ÍNDICE DE FIGURAS

Figura 1.1 Plan básico a usarse en el desarrollo de este proyecto de tesis	_6
Figura 1.2 Arquitectura PasS	_____7
Figura 1.3 Arquitectura del aplicativo a desarrollarse	_____9
Figura 1.4 Pasos del desarrollo de la metodología XP	_____10
Figura 1.5 Diagrama de las herramientas	_____11
Figura 2.1 Grafico descriptivo de las etapas de la metodología XP	_____16
Figura 2.2 Logo del proveedor de PAAS Openshift	_____28
Figura 2.3 Servicios ofrecidos por Openshift	_____29
Figura 2.4 Paquete Libre de Openshift Online	_____31
Figura 2.5 Paquete Bronce de Openshift Online	_____32
Figura 2.6 Paquete Plata de Openshift Online	_____32
Figura 2.7 Logotipo de Node.js	_____35
Figura 2.8 Modelo de procesamiento de Node.js	_____38
Figura 2.9 Diferencias entre blocking y non-blockin	_____40
Figura 2.10 Logo npm	_____41
Figura 2.11 Logo de ExpressJS	_____42

Figura 2.12 Logo mongoose	44
Figura 2.13 Logotipo de AngularJS	45
Figura 2.14 Comparación de angulas con otras tecnologías	47
Figura 2.15 Partes del modelo MVC.	48
Figura 2.16 Dependency Injector AngularJS	54
Figura 2.17 Logo de mongoDB	56
Figura 2.18 Ejemplo de estructura de un documento de mongoDB	62
Figura 2.19 Diagrama de una petición y respuesta en el protocolo HTTP	66
Figura 2.20 Uso de HTTP mediante el método GET para obtener un recurso	68
Figura 3.1 Formato de la matriz de marco lógico	76
Figura 3.2 Estructura de la metodología de marco lógico	77
Figura 3.3 Ejemplo de un árbol de problemas. En base a un supuesto problema.	80
Figura 3.4 Árbol de objetivos en base al problema en la Figura 3.3	82
Figura 3.5 Relación entre componentes de los arboles	84
Figura 3.6 Lista de acciones en base al supuesto problema planteado en la Figura 3.3	84

Figura 3.7 Estructura analítica del proyecto sobre el problema planteado en la	
Figura 3.3 _____	86
Figura 3.8 Relación de la estructura analítica del proyecto con la matriz de marco	
lógico. _____	88
Figura 3.9 Ejemplo de la relación de la estructura analítica con la matriz de marco	
lógico _____	89
Figura 3.10 Tabla para generación de índices. _____	91
Figura 4.1 Diagrama de entidades _____	147
Figura 4.2 Sistema de archivos del modelo de datos _____	148
Figura 4.3 Formulario acceso de usuarios _____	158
Figura 4.4 Formulario registro de usuarios _____	159
Figura 4.5 Formulario perfil de usuario _____	160
Figura 4.6 Formulario gestión de proyectos _____	161
Figura 4.7 Formulario eliminar proyecto _____	162
Figura 4.8 Formulario compartir proyecto _____	163
Figura 4.9 Formulario ver proyecto _____	163
Figura 4.10 Formulario de gestión de involucrados _____	164
Figura 4.11 Formulario editar involucrado _____	165

Figura 4.12	Formulario eliminar involucrado	165
Figura 4.13	Formulario gestión del problema	166
Figura 4.14	Formulario ver edición e inserción de una causa-efecto	167
Figura 4.15	Formulario gestión de objetivos	168
Figura 4.16	Formulario ver edición e inserción de medio-fin	168
Figura 4.17	Formulario de gestión de alternativas	169
Figura 4.18	Formulario de generación de acciones por cada medio	170
Figura 4.19	Primera columna de la matriz.	171
Figura 4.20	Gestión de indicadores, medios de verificación y supuestos.	172
Figura 4.21	Gestión de los resúmenes narrativos del proyecto.	173
Figura 4.22	Muestra los indicadores, medios de verificación y supuestos de cada componente de la matriz, así como las últimas evaluaciones.	173
Figura 4.23	Formulario para realizar seguimiento de cada indicador mediante evaluaciones.	174
Figura 4.24	Formulario para agregar evaluaciones.	175
Figura 4.25	Formulario para agregar enlaces de los documentos o imágenes de pruebas de una evaluación.	175
Figura 4.26	Formulario para verificar el cumplimiento del indicador o su falla.	176

Figura 4.27 Pop up de ayuda al usuario con información de cada paso de la matriz.	177
Figura 4.28 Configuración de ayudas.	178
Figura 4.29 Reporte de matriz de marco lógico.	179
Figura 4.30 Reporte del proyecto	179
Figura 4.31 Reporte de seguimiento	180
Figura 4.32 Estructura del proyecto	181
Figura 4.33 Acceso al sistema	181
Figura 4.34 Gestión de proyectos	182
Figura 4.35 Gestión de un proyecto	183
Figura 4.36 Gestión de involucrados	183
Figura 4.37 Gestión del problema	184
Figura 4.38 Gestión del objetivo	185
Figura 4.39 Gestión de estrategia	186
Figura 4.40 Generar proyecto, gestión de seguimiento	187
Figura 4.41 Generar proyecto, resúmenes narrativos	188
Figura 4.42 Ver indicadores	189

Figura 4.43 Gestión de evaluaciones_____189

Figura 4.44 Pop up de ayuda_____190

Figura 4.45 Configuración de ayudas _____191



CAPITULO I



- 4. INTRODUCCIÓN
 - 4.1. ANTECEDENTES
 - 4.2. SITUACIÓN ACTUAL
 - 4.3. PROBLEMA
 - 4.4. JUSTIFICACIÓN
 - 4.5. OBJETIVOS
 - 4.6. ALCANCE

1 INTRODUCCIÓN

1.1 ANTECEDENTES

Los proyectos de calidad son aquellos que constan de partes bien definidas y coherentes entre sí, es decir que tengan sentido unas con otras. Muchos de los proyectos realizados se los elabora de manera empírica sin tomar en cuenta la relación de sus componentes, se trata cada paso como un contenido independiente sin tener en cuenta que son un todo que está relacionado entre sí.

La gestión de proyectos es un campo que con el tiempo ha venido tomando fuerza dentro del medio globalizado, últimamente dentro del medio, han surgido varias metodologías para la realización y gestión de proyectos, pero las mismas necesitan un conocimiento profundo y avanzado sobre las mismas para su uso. Siendo así herramientas que necesitan cierta experticia para su máxima explotación.

1.2 SITUACIÓN ACTUAL

En el medio existen herramientas que ayudan a la gestión de proyectos, pero estas requieren así mismo conocimientos de la metodología, lo que dificulta su uso y comprensión sin ser didácticas con el usuario, y dejándolo sin un soporte para la gestión de sus proyectos, otras herramientas en cambio no presentan todas las funcionalidades de las metodologías como las herramientas de generación de cronograma de actividades, llenado así solo ciertas partes de un proyecto completo y de calidad. Dejando ciertos componentes de un proyecto fuera del contexto lo que no permite la generación completa, eficaz y efectiva de

un proyecto de calidad que cumpla completamente con las normas de una metodología.

Con la evolución de las tecnologías, la gestión de proyectos puede tomar grandes pasos y brindara maneras más eficaces y efectivas para su uso. Ahora se puede realizar proyectos mejores. Pero sin el conocimiento y la guía adecuada los proyectos tendrán una mala inicialización lo que podría llevar a posibles problemas en el futuro.

1.3 PROBLEMA

El problema principal es la relación de contenidos de los proyectos e imposibles de realizar un seguimiento apropiado, esto debido a la falta de herramientas didácticas que usen metodologías para la gestión de proyectos de calidad dejando así a estudiantes, docentes, clubes con la difícil tarea de aprender de manera independiente metodologías diferentes y creando un ambiente no unificado para el seguimiento de los mismos los mismos en la elaboración de proyectos. Conllevando a un ambiente no incluyente; donde cada entidad genera su propia metodología, creando formas de gestión no estandarizadas.

Esto genera un desarrollo de proyectos por parte de la comunidad universitaria precario, sin formas de gestionar los mismos poniendo una traba desde el inicio de los mismos lo que generara problemas a futuro si el proyecto continua.

1.4 JUSTIFICACIÓN

La matriz de marco lógico es una metodología simple y efectiva que permitirá mediante un procedimiento definir las piezas fundamentales de un proyecto gracias a organizadores gráficos y técnicas. Esta metodología también genera proyectos con contenidos coherentes y relacionados entre sí de manera simple y rápida usando una matriz en la cual podemos apreciar todo el proyecto de manera rápida y concreta.

Una herramienta dinámica y didáctica que facilite la gestión de proyectos usando la metodología de marco lógico ha sido un elemento faltante dentro de nuestro sistema académico, tomando en cuenta las herramientas existentes no cumplen con todos los pasos de una metodología, automatizando ciertos pasos y omitiendo otros. Esto genera vacíos dentro del proyecto confundiendo al usuario y malinterpretando el uso de estas herramientas.

Una herramienta de gestión de proyectos usando marco lógico permitirá a los estudiantes aprender una forma de generar proyectos con las partes fundamentales del mismo, y a los docentes de llevar a cabo un seguimiento de proyectos de manera fácil.

El principal propósito de la herramienta es ayudar tanto a docentes como estudiantes dentro de la producción de proyectos de manera fácil, rápida y cumpliendo ciertas normativas de calidad para la gestión y seguimiento de proyectos mediante la metodología de marco lógico.

1.5 OBJETIVOS

1.5.1 Objetivo General

- Generar una herramienta para la gestión y seguimiento de proyectos mediante la metodología de marco lógico.

1.5.2 Objetivos Específicos

- Definir la situación actual.
- Establecer marco teórico para el desarrollo del proyecto.
- Estudiar la metodología de marco lógico.
- Desarrollo del aplicativo usando XP como metodología de desarrollo
- Establecer conclusiones y recomendaciones

1.6 ALCANCE

El sistema es una herramienta didáctica que servirá para los docentes, estudiantes, clubes de la universidad y personas con acceso a internet que deseen gestionar proyectos mediante esta metodología. La herramienta funcionará sobre un servicio de **PaaS**¹ mediante **Openshift**², estará en la nube, es

¹ **Plataform As A Service**: Servicio que nos ofrece equipos para la implementación y despliegue de aplicaciones.

² **Openshift**: Parte de la empresa RedHat que ofrece servicios como plataforma para la implementación y despliegue de aplicaciones.

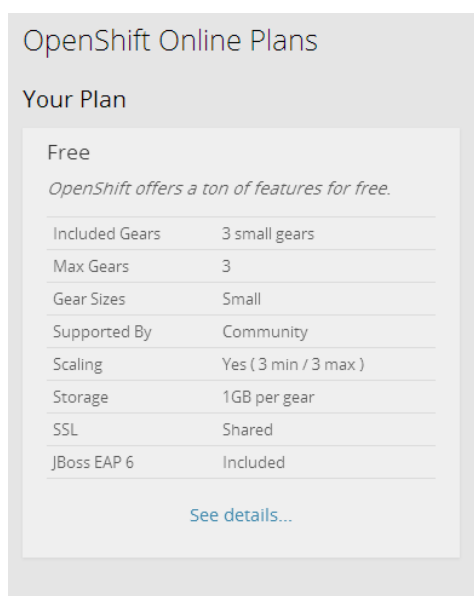
decir sobre servidores ubicados en diferentes partes del mundo interconectados mediante redes. Gracias a esto será accesible para todos los usuarios con conexión a internet.

Dando así soporte a la gestión de proyectos de calidad y facilitando el desarrollo de los mismos brindando una metodología que es usada por la **SENESCYT**³ para el control y gestión de proyectos.

OpenShift ofrece de manera gratuita el servicio de **PaaS** pero limita los recursos disponibles para la aplicación se tomar el paquete básico y gratuito.

Este ofrece las siguientes características:

Paquete de **OpenShift**:



OpenShift Online Plans

Your Plan

Free	
<i>OpenShift offers a ton of features for free.</i>	
Included Gears	3 small gears
Max Gears	3
Gear Sizes	Small
Supported By	Community
Scaling	Yes (3 min / 3 max)
Storage	1GB per gear
SSL	Shared
JBoss EAP 6	Included

[See details...](#)

Figura 1.1 Plan básico a usarse en el desarrollo de este proyecto de tesis

³ **SENESCYT**: Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación.

Fuente: <https://www.openshift.com/products/pricing>

Este paquete nos ofrece la gestión de 3 posibles aplicaciones por tiempo indefinido con un espacio en base de datos de 1 GB⁴.

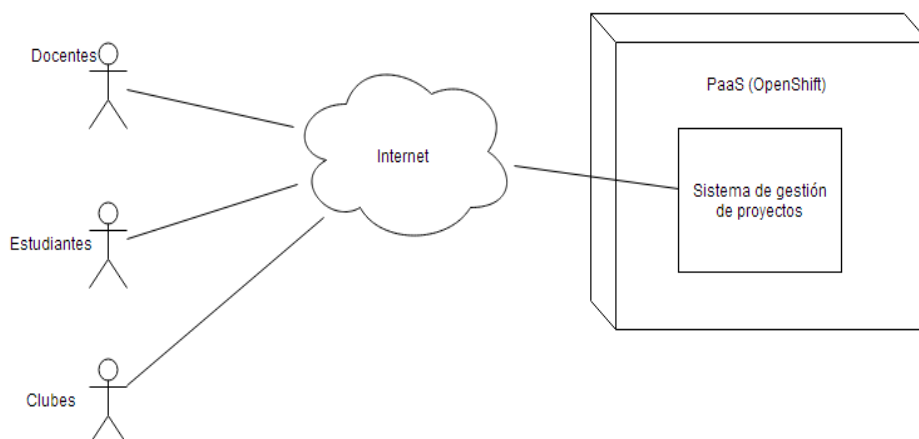


Figura 1.2 Arquitectura PaaS

Fuente: Propia

El sistema contará con cuatro módulos esenciales como lo indica el gráfico 4.3:

- **Módulo de ayuda al usuario:** Contendrá de la parte didáctica del sistema brindando comentarios de ayuda al usuario y guiándolo en el uso del aplicativo.

⁴ **GigaByte:** Unidad de medida de cantidad de datos en sistemas informáticos.

- **Módulo de marco lógico:** Será el encargado de la generación de la cabecera del proyecto es decir los datos informativos necesarios de un proyecto.
- **Módulo de seguimiento:** Estará encargado del control usuarios y el seguimiento a los proyectos mediante indicadores.
- **Módulo de generación:** de proyectos es el que se encargara de presentar un reporte con un formato dado del proyecto realizado en el sistema.

4.3 Arquitectura del aplicativo

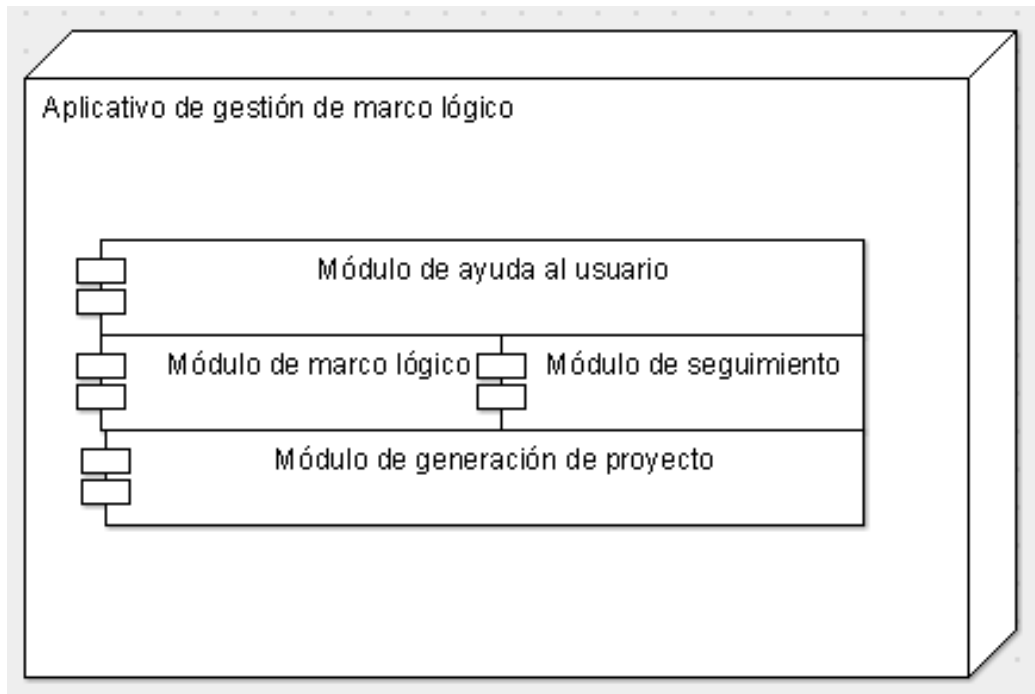


Figura 1.3 Arquitectura del aplicativo a desarrollarse

Fuente: Propia

Para el desarrollo del aplicativo se usara una metodología de desarrollo rápido como la es XP⁵, para la generación de documentación del sistema y un correcto desarrollo del mismo.

Fases da la metodología XP según *Ian Sommerville*

⁵ **Extreme Programming:** Metodología ágil de desarrollo de sistemas informáticos.

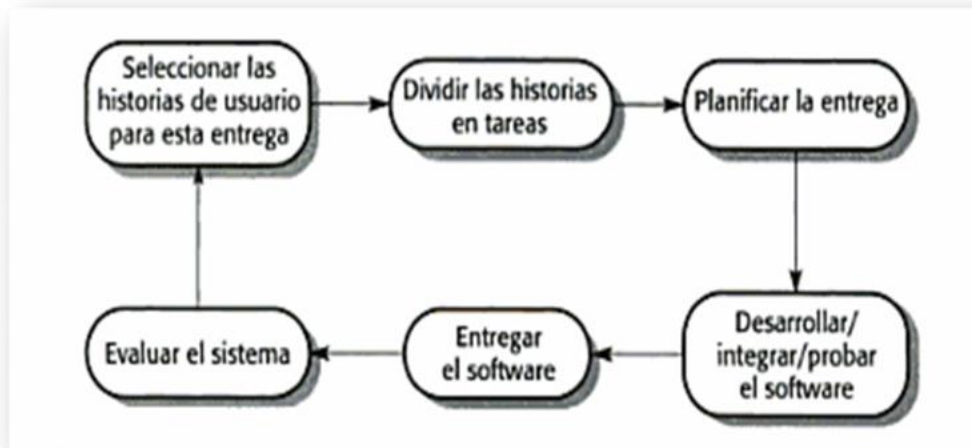


Figura 1.4 Pasos del desarrollo de la metodología XP

Fuente:

http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_SCRUM_y_XP#METODOLOG.C3.8DA_XP_.28EXTREME_PROGRAMMING.29

Herramientas

Para la elaboración del aplicativo se usará un stack ⁶nuevo de tecnologías, el aplicativo será una aplicación web.

⁶ **Stack:** Conjunto de tecnologías que tienen buena relación entre ellas y que permiten generar sistemas informáticos de manera rápida.

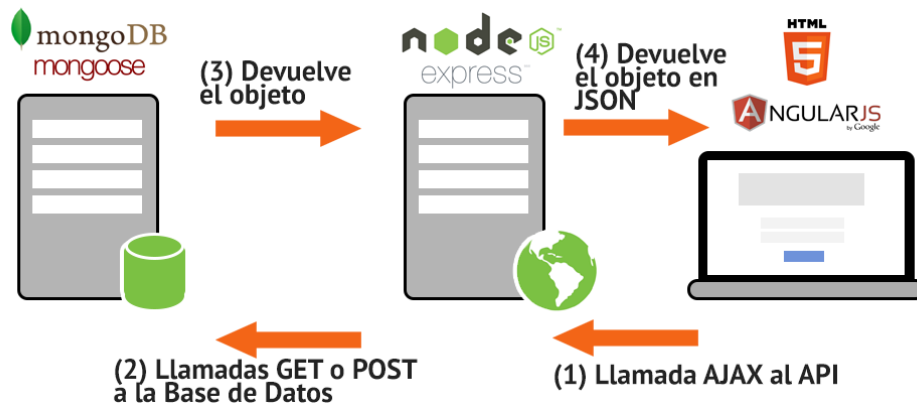


Figura 1.5 Diagrama de las herramientas

Fuente: <http://carlosazaustre.es/blog/tutorial-ejemplo-de-aplicacion-web-con-angular-js-y-api-rest-con-node/>



CAPITULO II



- 5. MARCO TEÓRICO
 - 5.1. METODOLOGÍA XP
 - 5.2. PLATAFORMA
COMO SERVICIO
 - 5.3. NODE.JS
 - 5.4. ANGULAR.JS
 - 5.5. MONGODB
 - 5.6. SERVICIOS REST

2 MARCO TEÓRICO

2.1 METODOLOGÍA XP (PROGRAMACIÓN EXTREMA)

La metodología de desarrollo XP⁷, es una metodología de desarrollo rápido, las metodologías de desarrollo rápido a diferencia que sus iguales, metodologías tradicionales, se centran en el desarrollo del sistema y las relaciones con el cliente mas no en la documentación de cada etapa. Usan una filosofía flexible que puede cambiar en el proceso de desarrollo que permite una más rápida y menos estricta para generar sistemas informáticos en grupos pequeños.

La programación extrema viene como una nueva corriente y alternativa para las metodologías tradicionales, brindando un entorno enfocado al desarrollo, eliminando pasos que cree innecesarios.

El principal creador y autor de esta metodología Kent Beck propone esta metodología como una forma rápida y sin tabas para los desarrolladores creando ciclos más cortos y rápidos de desarrollo conocidos como iteraciones y aceptando que en cada ciclo puede haber cambios en los requerimientos.

Los autores mencionan lo siguiente sobre la metodología:

XP es una metodología eficiente, ligera, de bajo riesgo, flexible, predecible y una divertida manera de desarrollar software. (Beck & Andres, 2005)

⁷ **Extreme Programming (Programación Extrema):** Metodología de desarrollo ágil

2.1.1 Ciclo de vida

La metodología XP presenta un ciclo de vida diferente al de las conocidas metodologías, pero usa un esquema de etapas básico de las metodologías de desarrollo conocidas:

- Planificación
- Diseño
- Codificación
- Pruebas

Las etapas anteriores pueden variar desacuerdo al diseño e implementación de la metodología, pueden ser más de acuerdo a la planificación e implementación de la misma, pero las etapas anteriormente mencionadas son las principales. La característica más importante de XP es el desarrollo por iteraciones, que son pequeños ciclos repetitivos en los que las etapas anteriores se repiten constantemente generando soluciones en cada iteración y permitiendo cambios en cualquier momento del desarrollo.

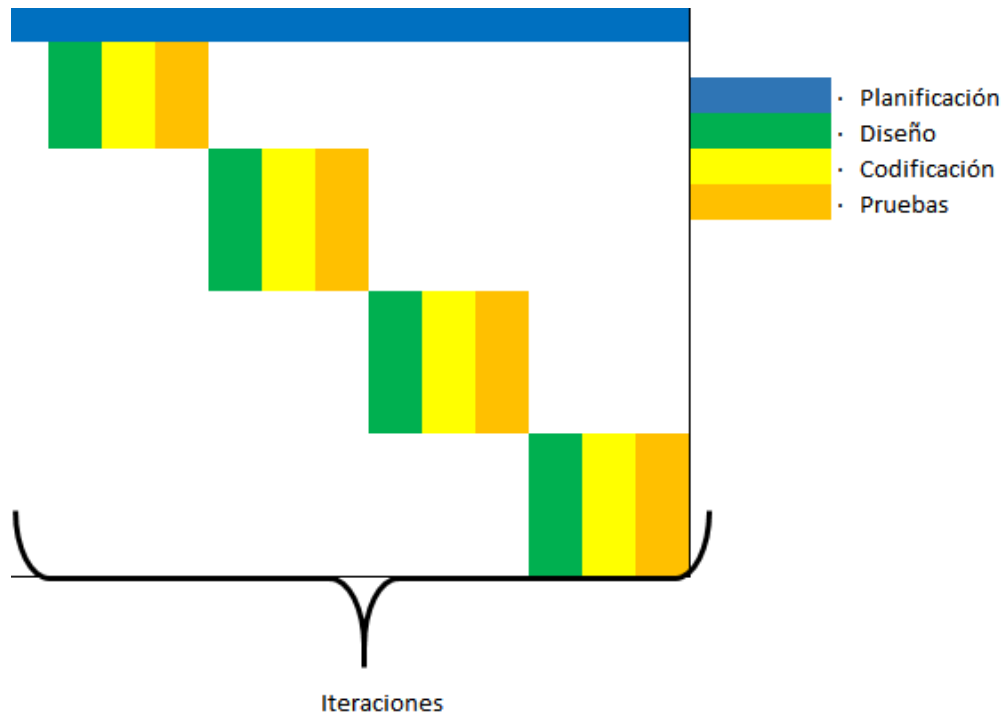


Figura 2.1 Grafico descriptivo de las etapas de la metodología XP

Fuente: Propia

2.1.2 Valores

La metodología XP no propone ciertos valores a la hora de su aplicación como bases fundamentales de su uso y como reglas implícitas del juego, que en el caso de esta metodología son valores que tienen que ser respetados y aplicados para obtener los mejores resultados posibles.

- **Comunicación**

Según el argumento de José Joskowicz que dice:

Muchos de los problemas que existen en proyectos de software (así como en muchos otros ámbitos) se deben a problemas de comunicación entre las

personas. La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto. (Joskowicz, 2008)

Como menciona el autor la comunicación es un aspecto fundamental dentro de cualquier metodología no solo con el cliente si no con las personas que nos rodean, los problemas de comunicación son muy frecuentes pero tenemos que aprender a evitarlos mejorando nuestras habilidades sociales que es una caracteriza esencial para el desarrollo de sistemas informáticos.

- **Simplicidad**

Según José Joskowicz:

XP, como metodología ágil, apuesta a la sencillez, en su máxima expresión. Sencillez en el diseño, en el código, en los procesos, etc. La sencillez es esencial para que todos puedan entender el código, y se trata de mejorar mediante recodificaciones continuas. (Joskowicz, 2008)

El autor nos da la pauta de en qué campos debemos aplicar la sencillez pero tomando en cuenta que es una metodología de desarrollo rápido debemos siempre tener en cuenta que tenemos que ser rápidos y tratar de complicarnos lo menos posible en el desarrollo, sin dejar a un lado la calidad del trabajo.

- **Retroalimentación**

“Para aprender las lecciones apropiadas en cada oportunidad posible.”

(Shore & Warden, 2007)

Se tiene que dejar de lado el orgullo y aprender continuamente de los errores para así evitar retrasos dentro de la elaboración del proyecto.

- **Coraje**

Los autores de “*The art of agile development*” dicen:

Para tomar las decisiones correctas, incluso cuando son difíciles, y para decir la verdad cuando las partes interesadas que necesitan oírlo. (Shore & Warden, 2007)

Los autores claramente nos explican la importancia de este valor dentro de la metodología, dándonos a conocer, que se debe ser directos en toda cuestión y etapa del desarrollo para evitar así posibles retrasos.

Al final los valores antes mencionados tienen una correlación entre sí, dando a entender que si no se aplicara alguno de estos los demás fallarían y no se cumplirá con la meta de la metodología.

2.1.3 Artefactos

Al ser una metodología de desarrollo rápido se centra más en requisitos y la comunicación con el cliente que en la documentación generada por cada etapa. Pero también al ser una metodología software hace uso de ciertos artefactos que sirven de documentación.

Dentro de cada etapa de la metodología de desarrollo se generan los siguientes entregables que son los documentos de respaldo de cada etapa, en cada etapa tenemos los siguientes artefactos:

- **Planificación**

Dentro de esta etapa se generan los siguientes documentos:

Historias de usuario

“Las historias de usuario representan elementos individuales y autónomos del proyecto. Estas tienden a corresponder a una característica del sistema y típicamente representan uno o dos días de trabajo.” (Shore & Warden, 2007)

Las historias de usuario son el pilar fundamental de XP ya que proveen el paso inicial para poder realizar las iteraciones de manera rápida y ordenada. En estas se detalla la información de cada requerimiento para después ser clasificadas y divididas en tareas para su elaboración.

Plan de iteraciones (Iteration Plan)

Compone un desglose de las historias de usuario en tareas y las iteraciones a realizarse por cada una de las historias de usuario.

(Joskowicz, 2008):

“Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.”

Plan de publicaciones (Release Plan)

“El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.).” (Joskowicz, 2008)

En este documento se ordena las historias de usuario para ser publicadas o elaboradas, sirve para establecer las fechas de cada historia y su tiempo de desarrollo.

- **Diseño**

Diseño de modelos de datos

El modelo de datos a usar dentro del aplicativo en donde se define el tipo de cada dato y su relación con otros datos.

Debido a que la base de datos no es una base de datos relacional el modelo de datos no es un diagrama de base de datos debido a las diferencias entre bases de datos relacionales y no relacionales. En este caso el modelo de datos es un diagrama de objetos JSON.

Diseño de interfaces de usuario

Diseños previos de las interfaces de usuario, posiciones de elementos y paleta de colores a usarse para el desarrollo del aplicativo.

Conocidos también como **mockups**⁸ son diseños bosquejados de lo que se pretende tener como producto final para tener una guía de diseño.

- **Codificación**

Manual Técnico

Documento en el que constan especificaciones técnicas sobre el código y definiciones de métodos para que técnicos puedan entender el funcionamiento del sistema.

Manual de usuario

Documento en el que constan especificaciones para que los usuarios en general pueden hacer uso de todas la características del sistema.

Contiene descripciones de las interfaces gráficas y detalla los procesos.

- **Pruebas**

Pruebas

Documentación en donde se detalla las pruebas unitarias realizadas al sistema para comprobar su correcto funcionamiento.

Glosario de términos

⁸ **Un mockup, mock-up, o maqueta:** es un modelo previo de diseño como guía de desarrollo.

Documento que contiene los nombre de métodos y clases para su reutilización a futuro.

2.2 PLATAFORMA COMO SERVICIO (PAAS)

Con la evolución del internet, el crecimiento y la velocidad del mismo han surgido nuevas tecnologías que nos permiten mejorar nuestra calidad de vida y generar oportunidades de crecimiento. La nube⁹ es una de estas que brinda tecnologías por medio del internet, pueden ser desde simples servicios hasta infraestructuras de hardware¹⁰ y software¹¹ para nuestro uso.

“La Computación en Nube o Informática en Nube (Cloud Computing) se ha convertido en un nuevo paradigma tecnológico de gran impacto social. La Nube (The Cloud) es el conjunto «infinito» de servidores de información (computadores) desplegados en centros de datos, a lo largo de todo el mundo donde se almacena millones de aplicaciones Web y enormes cantidades de datos (big data), a disposición de miles de organizaciones y empresas, y cientos de miles de usuarios que se descargan y ejecutan directamente los programas y aplicaciones de software almacenados en dichos servidores tales como Google Maps, Gmail, Facebook, Tuenti o Flickr. La Nube está propiciando una nueva revolución industrial soportada en las fábricas de «datos» (Centros de Datos, Data Centers) y fábricas de

⁹ **Nube o Cloud Computing:** tecnología que permite manejo de datos en servidores remotos.

¹⁰ **Hardware:** componentes físicos de un sistema informático

¹¹ **Software:** Componentes lógicos de un sistema informático

«aplicaciones Web» (Web Apps). Esta nueva revolución producirá un gran cambio social, económico y tecnológico, pero al contrario que otras revoluciones será «silenciosa» al igual que lo ha sido la implantación Internet y la Web en la Sociedad.” (Joyanes, 2010)

La nube se la puede clasificar de acuerdo al tipo de servicio que ofrece desde solo aplicaciones en internet hasta servidores y equipos. De acuerdo a esto se clasifica en tres grandes grupos que son los siguientes:

- **SAAS**

Software as a service, que en español significa software como servicio, este tipo de servicio consta de ubicar sistemas informáticos sobre la nube para distribuirlos como un servicio. Los servicios más conocidos de este son:

- Correo electrónico(Outlook, Gmail)
- Editores de texto(Google Drive, Word online)
- Editores de imágenes(Phixr)
- Editores de video(Wevideo, Magistro)

- **PAAS**

Platform as a service, que en español significa plataforma como servicio, este se encuentra en medio brindando una mezcla de características de los otros dos servicios dentro de esta clasificación. PAAS nos brinda una facilidad de manipular los equipos pero nos sin tener que preocuparnos de aspectos de tan bajo nivel

como son la configuración de redes o de equipos físicos, provee un entorno de trabajo para desarrolladores de software, ayuda a gestionar nuestros datos o aplicaciones de manera fácil y segura. Dentro de esta categoría existen algunos proveedores pero lo más conocidos son:

- Openshift
- Google App Engine
- Windows Azure
- Heroku
- Caspio

- **IASS**

Infrastructure as a service, que en español es infraestructura como servicio, este nos propone un nivel más profundo que en el PAAS ya que nos brinda equipos para su uso completo es decir podemos hacer lo que queramos con los mismos. Es como tener un **data center**¹² que ubicado en una posición diferente a la de nuestra organización. Dentro de estos no existen muchos proveedores, en comparación con los servicios anteriormente mencionados, ya que este tipo de servicio requiere una gran cantidad de infraestructura física y mantenimiento de misma. Pero los principales proveedores de la mis son:

- Amazon AWS

¹² **Data center o centro de datos:** es un espacio donde se ubican los equipos y se procesa la información de una organización.

- Windows Azure
- Google Compute Engine
- RackSpace
- IBM SmartCloud

Teniendo en cuenta las definiciones anteriores PAAS es un servicio que nos brinda un entorno virtual dentro de un equipo en la nube para poder desplegar aplicaciones de manera gratuita o pagada dependiendo del proveedor.

Sobre el costo de los mismos varía dependiendo de cada proveedor pero al final mantienen un estándar de cobrar por lo que se usa es decir se cobra por la capacidad de equipos usada por el cliente.

Los servicios PAAS a diferencia de los otros anteriormente mencionados, al estar en medio, tiene las ventajas de los dos servicios anteriormente mencionados y minimizan las desventajas de los mismos ofreciendo un punto intermedio dentro de la nube.

Las principales ventajas y desventajas del uso de estos servicios que van incursionando más y más sobre la tecnología que nos rodea esta la siguiente tabla donde se habla de las principales y más importantes ventajas y desventajas de su uso.

VENTAJAS	DESVENTAJAS
----------	-------------

Mas enfoque en el desarrollo, que en los equipos de hardware	Recursos de hardware limitados y compartidos
Mejoras de seguridad y actualizaciones son automáticas y por parte del proveedor.	No se obtienen las actualizaciones a tiempo hasta que sean completamente probadas por el proveedor.
Costos reducidos, se paga por lo que se usa.	Limitaciones en los recursos ofrecidos.
Rápido despliegue e implementación de aplicaciones.	Requiere de conocimiento sobre el proveedor de PAAS, cambiar la forma de despliegue dependiendo del mismo

Tabla 2.1 Ventajas y desventajas de los servicios PAAS

Fuente: Propia

2.2.1 Características

Cada servicio ofrece distintas características dentro del mercado, para así obtener una ventaja competitiva sobre los demás proveedores, pero las principales características y más comunes por las que se puede diferenciar estos son las siguientes:

- **Tecnologías**

Son los paquetes soportados e instalados dentro del servicio, es decir, los framework's¹³, API's¹⁴, lenguajes de programación, servidores web, servidores de archivos, servidores de correo.

- **Escalamiento:**

Es una característica esencial dentro de los PAAS nos indica que tipo de crecimiento pueden tener nuestras aplicaciones. Existen 2 formas horizontales que es el aumento de equipos de procesamiento paralelos, vertical que aumenta la capacidad de un solo equipo aumentando sus características físicas y funcionales.

Algunos proveedores ofrecen escalamiento automático es decir las capacidades aumentan dependiendo de la demanda de la aplicación.

- **Acceso**

Pueden ser de acceso público o privado, público cuando cualquier persona puede acceder a nuestra aplicación y privados se restringe el acceso.

¹³ **Framework:** Conjunto de componentes para facilitar el desarrollo de sistemas.

¹⁴ **API(Application Programming Interface):**Conjunto de métodos y funciones para facilitar el desarrollo

- **Infraestructura**

Esta característica nos indica que tipo de infraestructura usa nuestro proveedor y en donde está ubicada. Tomando en cuenta que entre más lejos estén ubicados aumentara la latencia¹⁵ y las aplicaciones se ejecutarán más lentamente por la distancia que los datos tienen que viajar.

Por esto los proveedores de PAAS disponen de varias ubicaciones para así evitar este problema.

- **Costo**

El costo es una característica primordial dentro de una decisión ya que es un punto calve dentro del desarrollo. Muchos servicios PAAS ofrecen sus servicios de manera gratuita pero con restricciones sobre los recursos brindados.

2.2.2 Ret Hat Openshift



Figura 2.2 Logo del proveedor de PAAS Openshift

¹⁵ **Latencia:** es la demora de propagación de paquetes dentro de una red.

Fuente: <http://www.techamericafoundation.org/content/wp-content/uploads/2013/05/OpenShiftLogo.png>

Openshift es un proveedor de PAAS que nos ofrece gran variedad de tecnologías y una arquitectura robusta para el desarrollo rápido y escalable de aplicaciones sobre la nube. Es una plataforma desarrollada por la compañía Red Hat y escrita en Ruby¹⁶, su primera aparición fue en mayo del 2011. Liberada bajo la licencia de Apache Licence 2.0.

Openshift provee 3 servicios de PAAS que son los siguientes:

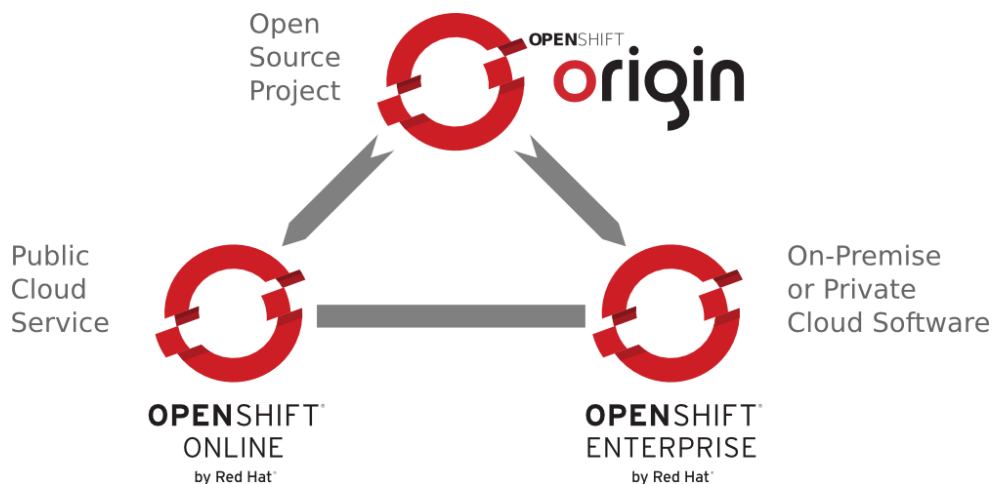


Figura 2.3 Servicios ofrecidos por Openshift

¹⁶ **Ruby:** es un lenguaje de programación.

Fuente: Getting Started with OpenShift (Pousty & Miller, 2014)

- **Openshift Origin:**

OpenShift Origin, es la versión gratuita y de código abierto de OpenShift, este proyecto está sobre las otras dos versiones existentes. Está disponible en GitHub y e libre uso bajo una licencia Apache 2. Todos los cambios en la base de código de pasar por el repositorio público, tanto para Red Hat y desarrolladores externos. Si desea utilizar esta versión tendrás que instalarlo en su propia infraestructura. (Pousty & Miller, 2014)

Este proyecto de Openshift se beneficia de la comunidad para crecer y mejorar, cada cierto tiempo esta versión se despliega como una versión de Openshift Online.

- **Openshift Online:**

Openshift Online, Red Hat se encarga de organizar y montar el servicio de PAAS sobre AWS¹⁷, es decir la infraestructura esta sobre este proveedor de IAAS. Tareas tales como la actualización del sistema operativo y la gestión de redes, está cubierto por las operaciones del equipo de OpenShift. Usted es libre para centrarse en su aplicación y su código. (Pousty & Miller, 2014)

¹⁷ **Amazon Web Services (AWS):** es un proveedor de servicios de infraestructura.

Este servicio es gratuito pero solo con una cantidad de recursos limitados, aumentar sus capacidades tiene un coste adicional dependiendo de las características que se desee.

Para poder tener accesos a este servicio se debe registrar en la página oficial:

<https://www.openshift.com>

Y generar una cuenta, después nos brindaran el paquete básico y gratuito, en el caso de necesitar mayores beneficios se puede comprar los paquetes que ofrece este proveedor.

Los principales paquetes que ofrece Openshift online son los siguientes:



Figura 2.4 Paquete Libre de Openshift Online

Fuente: <https://www.openshift.com/products/pricing>



Figura 2.5 Paquete Bronce de Openshift Online

Fuente: <https://www.openshift.com/products/pricing>



Figura 2.6 Paquete Plata de Openshift Online

Fuente: <https://www.openshift.com/products/pricing>

- **OpenShift Enterprise:**

OpenShift Enterprise, que se libera alrededor de una vez trimestre. Esta versión de OpenShift le permite tomar el servicio de PAAS y ejecutarlo en cualquier lugar

que desee, desde el los servidores en su centro de datos hasta Rackspace¹⁸ o AWS. Es un paquete completo con Red Hat Enterprise Linux y todos los bits OpenShift adjuntos al mismo. También cuenta completamente con el soporte técnico de Red Hat y está pensado para los clientes que quieren estabilidad. (Pousty & Miller, 2014)

Esta versión en si nos ofrece la posibilidad de tener nuestra propia nube sobre nuestro centro de datos o sobre un proveedor de IAAS. La nube seria completamente privada o pública en el caso de necesitarlo.

Openshift nos ofrece gran variedad de posibilidades para mejorar nuestras aplicaciones brindándonos la posibilidad de concentrarnos netamente en el desarrollo y dejando a un lado ciertos aspectos de los equipos que son complejos y requiere costes adicionales.

En resumen Openshift nos ofrece los siguientes servicios:

	OpenShift Online	OpenShift Empresa	OpenShift Origen
¿Qué es?	Servicio	PaaS PaaS	Privado Open Source
	Alojado	Producto	Proyecto PaaS

¹⁸ **RackSpace:** es un proveedor de servicios de infraestructura.

¿Cómo me puede ayudar?	Desarrolle rápidamente, aplicaciones a escala en la nube pública.	Acelerar la entrega de servicios y agilizar el desarrollo de aplicaciones.	Utilice un PaaS de código abierto o ayudar a extender OpenShift.
¿Cómo es un precio?	Gratis o Planes Premium	Suscripción anual software	Libre y de código abierto
¿Quién proporciona apoyo?	Apoyo comunitario para gratis y bronce Planes o Soporte de Red Hat para el Plan de Plata	Red Hat	Comunidad
¿Quién es bueno?	Startups, desarrolladores, empresas pequeñas, e incluso empresas	Las empresas que deseen ejecutar su propia nube	Cualquier persona que quiera trastear sobre lo último en software de código abierto

Tabla 2.2 Resumen de características de los servicios de Openshift

Fuente: <https://www.openshift.com/products>

2.3 NODE.JS

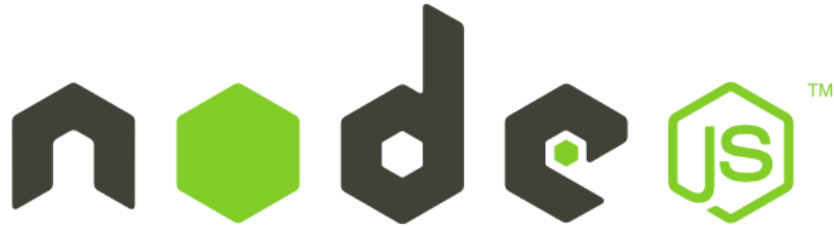


Figura 2.7 Logotipo de Node.js

Fuente: <http://www.nodejs.org/>

Node.js® es una plataforma construida sobre el motor de javascript ¹⁹ V8²⁰ de google Chrome, para construir fácilmente aplicaciones web rápidas y escalables. Node.js utiliza modelos de **controladores de eventos** y **non-blocking code**²¹, lo que hace que sea ligero y eficiente, ideal para aplicaciones en tiempo real de datos intensivos que se ejecutan a través de dispositivos distribuidos. (Joyent, 2014)

¹⁹ **JavaScript:** es un lenguaje de programación.

²⁰ **V8:** es el nombre del motor de compilación de JavaScript.

²¹ **Blocking code:** es una forma de ejecución de código en los motores de compilación.

Una de las grandes cualidades de Node.js es su simplicidad. A diferencia de PHP o ASP, no hay separación entre el servidor web y el código, ni tenemos que personalizar los archivos de configuración de gran tamaño para obtener el comportamiento deseado. Con Node.js, podemos crear el servidor web, personalizarlo, y entregar el contenido. Todo esto se puede hacer a nivel de código. (Clements, 2014)

Esta plataforma fue diseñada por Ryan Lienhart Dahl, lanzada el 27 de mayo del 2009. Su versión estable hasta la fecha es **Node.js 0.10.33** lanzada el 20 de octubre del 2014. Desarrollado sobre el motor V8 de javascript en los lenguajes de C++ lo que garantiza su eficiencia, es multiplataforma y licenciado bajo MIT²².

Node.js es el motor de ejecución de javascript, V8 de Google Chrome, modificado para un entorno fuera del navegador ofreciéndonos la posibilidad de usar este lenguaje fuera del entorno de programación de scripting²³ dentro de las páginas web, basado en una arquitectura orientada a eventos es un nuevo paradigma de programación que trata de generar aplicaciones en tiempo real poniendo énfasis en los eventos y controlándolos mediante **callbacks**²⁴ que es el envío de funciones como parámetros de otra función. Esto gracias a una de las

²² **Licencia MIT:** es una de tantas licencias de software que ha empleado el Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology)

²³ **Scripting:** son pequeñas porciones de código que realizar funciones básicas de dentro de un sitio

²⁴ **Callback:** en la programación, es una función que se ejecuta en respuesta a un evento determinado.

características más importantes de javascript que es su modelo de **non-blocking code**.

Node.js es una plataforma que brinda a los desarrolladores una nueva forma de desarrollo de sistemas y cuenta con miles de paquetes²⁵ para aumentar la rapidez del desarrollo sobre esta plataforma. Desarrollados por la comunidad para la misma y de manera gratuita.

2.3.1 Modelo orientado a eventos

El modelo orientado a eventos es una arquitectura que constantemente está a la espera de sucesos dentro del entorno, es decir que mientras realiza su ejecución normal paralelamente está a la espera de cualquier evento para su ejecución sin detener sus procesos actuales. Como lo muestra la siguiente figura:

²⁵ **Paquetes:** son componentes de middleware para facilitar el desarrollo de aplicaciones.

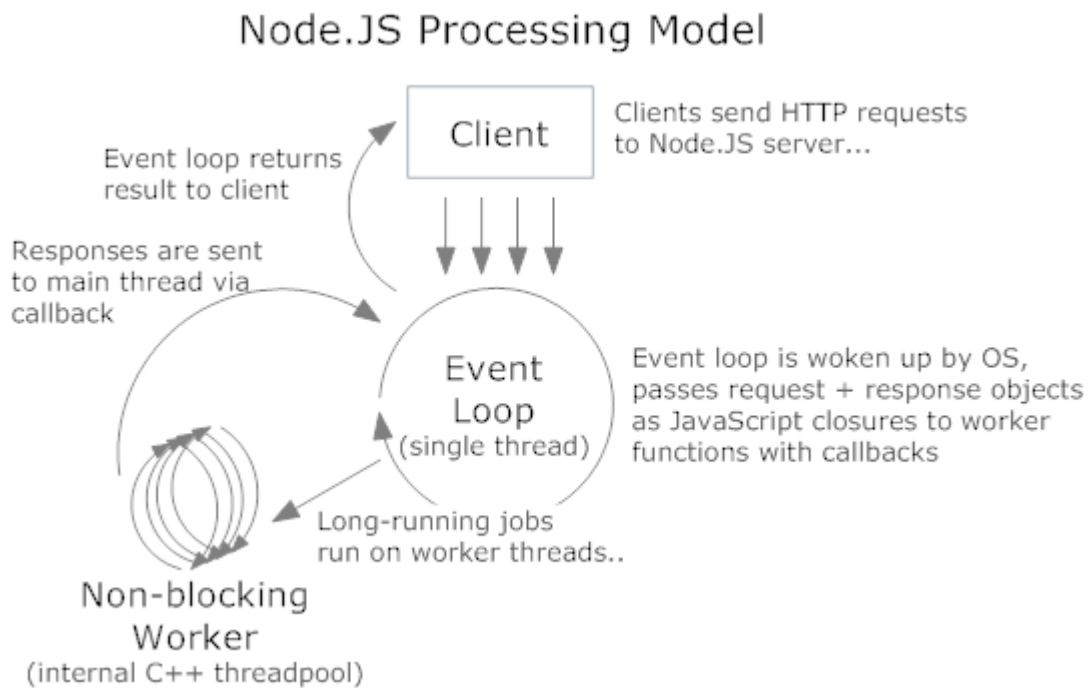


Figura 2.8 Modelo de procesamiento de Node.js

Fuente: <http://stackoverflow.com/questions/14795145/how-the-single-threaded-non-blocking-io-model-works-in-node-js>

2.3.2 Modelo non-blocking

Por otro lado el modelo de **non-blocking code** es una de las características más importantes de javascript y de Node.js, es la parte más importante ya que es lo que nos permite la realización de aplicaciones en tiempo real y explota al máximo el multiprocesamiento de eventos.

“JavaScript tiene ciertas características que lo hacen muy diferente de otros lenguajes dinámicos, es decir, que no tiene ningún concepto concurrencia. Su modelo de concurrencia es completamente en torno a su modelo de eventos”. Dahl Ryan (2009)

El principal creador de node.js uso javascript por su uso natural de concurrencia²⁶ que es basado en su modelo a eventos y es casi invisible para el desarrollador, javascript usa la concurrencia de manera nativa mediante los **callbacks** dándonos un control más limpio de los procesos, aunque un poco complejo de entender debido a su diferencia a los lenguajes de programación actuales que en su mayoría son orientados a objetos y manejan la concurrencia mediante objetos conocidos como Threads²⁷.

La diferencia entre lenguajes de código **blocking** y **non-blockin** está en su proceso de ejecución, como lo muestra la siguiente figura:

²⁶ **Concurrecia:** es el procesamiento paralelo de procesos

²⁷ **Threads:** conocidos como hilos son procesos paralelos de ejecución.

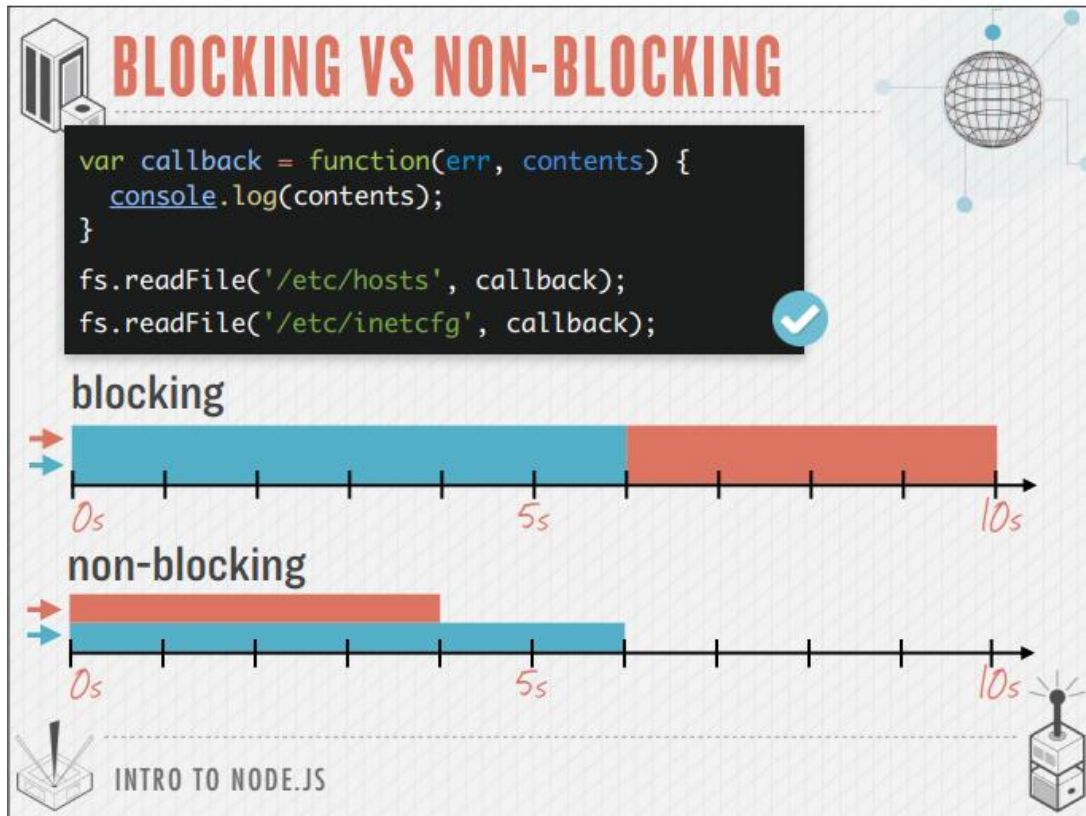


Figura 2.9 Diferencias entre blocking y non-blockin

Fuente: http://courseware.codeschool.com/node_slides.pdf

En el grafico anterior se puede apreciar la diferencia entre estos dos modelos, como vemos nos propone el proceso de la lectura de un archivo. En este caso el modelo **blocking** primero lee el primer archivo en 5 segundos al terminar su lectura continua con la siguiente orden de leer el segundo archivo en otros 5 segundos. LO que nos da a entender es que este modelo es lineal y mientras no termine el paso anterior no proseguirá con la siguiente instrucción.

En el caso de modelo **non-blocking** ejecuta la primera lectura y deja que el **callback** se encargue del procesamiento de la lectura y continúa con la siguiente instrucción, es decir al mismo tiempo comienza la lectura de los dos archivos

paralelamente aprovechando las capacidades de multiprocesamiento y reduciendo el tiempo de ejecución.

2.3.3 Node.js packages

Como se mencionó anteriormente los paquetes son componente o librerías de código desarrolladas en Node.js que hacen de middleware²⁸ para facilitar el desarrollo y evitar reescribir código ya existente, desarrollados por la comunidad estos son gratuitos.

Para su instalación y uso node.js usa un archivo de configuración y programa que gestiona los paquetes, conocido como **npm**²⁹ este programa se encarga del control de los paquetes sobre una aplicación en node.js mediante el archivo de configuración **packgae.json** que contiene información de los paquetes instalados en cierta aplicación, su versión mínima requerida y datos del autor.



Figura 2.10 Logo npm

²⁸ **Middleware:** conjunto de métodos y librerías para facilitar el desarrollo.

²⁹ **Npm:** es el gestor de paquetes de node.js.

Fuente: <https://www.npmjs.com/>

Todos los componentes instalados mediante el gestor están ubicados en el directorio **Node_modules** ubicado dentro de los proyectos de Node.JS.

- **EXPRESSJS**



Figura 2.11 Logo de ExpressJS

Fuente: <http://expressjs.com/>

Tan bueno como el módulo HTTP de Node.js es expressjs, contiene toda la funcionalidad para proporcionarnos una interfaz fácil y completa para el desarrollo web rápido casi sin fricción y demora. (Brown, 2014)

Actualmente se encuentra en su versión 4.0 y se encuentra en desarrollo la versión 5.0, en la versión 4.0 se separó algunos paquetes del servidor para mejorar su funcionalidad y dejar que los paquetes crezcan como dependencias aparte del servidor y sean integradas externamente.

Es un middleware que nos brinda un conjunto de métodos completos para el desarrollo de aplicaciones web, este paquete mejora las funcionalidades nativas del servicio HTTP³⁰ de node.js facilitándonos el manejo y mejorando su presentación mediante motores de vistas que facilita la generación de contenido HTML³¹ fácil rápido.

Este servidor de aplicaciones para Node.JS permite la implementación de motores de compilación de vistas que son los que brindan una funcionalidad para poder trabajar con templates o plantillas para las páginas HTML.

Los principales motores de plantillas de Express son:

- Dust
- JADE
- EJS
- HAML-Coffee
- Handlebars
- Hogan
- Jazz

Este módulo de node.js presenta las siguientes características según (Pollack & Souza, 2015):

³⁰ **HTTP** es un protocolo de transferencia de hipertexto por la red.

³¹ **HTML** Lenguaje de marcado que usan las páginas web para mostrar contenido.

- Fácil manejo de rutas
- Middleware para conexiones HTTP
- Configuración de entorno
- Fácil re direccionamiento
- Subida de archivos

En pocas palabras es un servidor de aplicaciones web integrado a la plataforma de node.js que nos facilita el desarrollo de aplicaciones web. Brindándonos un entorno para la programación de aplicaciones web en tiempo real.

- **MONGOOSEJS**



Figura 2.12 Logo mongoose

Fuente: <http://mongoosejs.com/>

Mongoose proporciona una solución basada en esquemas de modelado de datos de la aplicación e incluye una función de conversión de tipos, validación,

construcción de consultas, ganchos de lógica de negocios y más. (Mongoose, 2011)

Este paquete nos permite realizar conexiones a la base de datos no relacional MongoDB, también nos brinda métodos para la gestión de datos de manera rápida y sencilla.

Este paquete además de realizar la conexión con la base de datos aumenta la funcionalidad de la misma agregando ciertas funciones que la base de datos no presenta mejorando la funcionalidad de las aplicaciones y facilitando el uso de este paquete.

Mongoose es una de las librerías o paquetes de Node.JS más usados para las conexiones a mongoDB esto debido a su robustez y sencillez a la hora de su uso, al tener una gran similitud con los métodos y funcionalidades nativas de mongoDB.

2.4 ANGULAR.JS



Figura 2.13 Logotipo de AngularJS

Fuente: <https://angularjs.org>

AngularJS es un “**superheroic**” framework de JavaScript que usa el patrón de diseño MVC³² para la Web. Lo llamamos “**superheroic**” porque AngularJS hace tanto por nosotros que sólo tenemos que centrarnos en el núcleo aplicación y dejar que AngularJS se encargue de todo lo demás. Este nos permite aplicar estándares y prácticas de ingeniería de software tradicionalmente utilizados en el lado del servidor sobre del lado del cliente para acelerar el desarrollo **frontend**³³. Proporciona una estructura consistente y escalable que hace que sea muy fácil de desarrollar aplicaciones grandes y complejas como parte de un equipo. (Shyam & Brad, 2014)

AngularJS técnicamente por su propósito es un framework que nos ayuda en gran medida al desarrollo de aplicaciones web. Este “**superheroic**” framework nos brinda la oportunidad de mover ciertas tareas del servidor al usuario reduciendo la carga sobre el servidor y repartiéndola al usuario, esto a la vez que es una ventaja puede ser su peor cualidad ya que depende de la capacidad de procesamiento del equipo del usuario.

Esta tecnología desarrollada por uno de los gigantes de la informática Google y la comunidad al ser un proyecto de código abierto nos permite tener acceso a su código. Bajo la licencia de **MIT**. Esta librería programada en javascript implementa una programación sobre el lado del cliente creando un nuevo paradigma de programación web.

³² **MVC (Modelo Vista Controlador)**: es un patrón arquitectónico de diseño de software.

³³ **Front-end**: es la parte del softwares que está en contacto con los usuarios.

AngularJS es bastante nuevo. La versión 1.0 fue lanzado en 2012. Miško Hevery, un empleado de Google, comenzó a trabajar en AngularJS en 2009. La idea resultó muy bueno, y el proyecto ahora está respaldado oficialmente por un equipo de desarrollo de Google. (W3C, 2015)

Como se menciona anteriormente esta tecnología es nueva y ha tenido un vertiginoso crecimiento ya aceptación por parte de la comunidad por su gran facilidad y su filosofía de desarrollo.

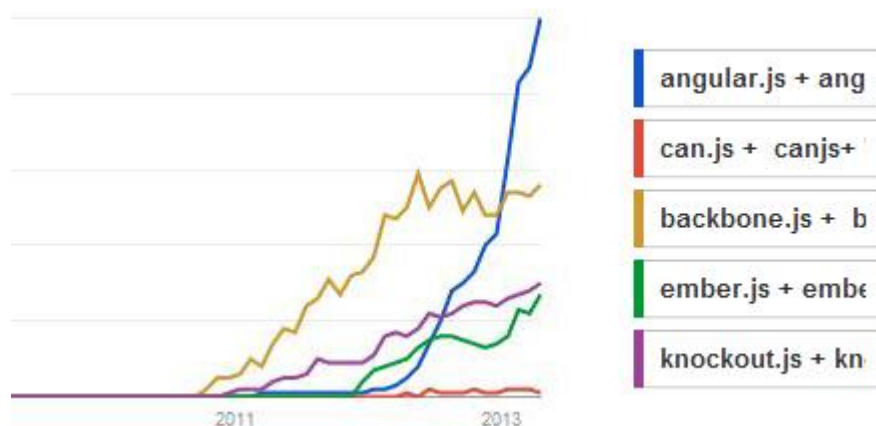


Figura 2.14 Comparación de angulas con otras tecnologías

Fuente: [http://www.google.com/trends/explore#cat=0-](http://www.google.com/trends/explore#cat=0-5&q=angular.js%20%2B%20angularjs%20%2B%20%22angular%20js%22,%20can.js%20%2B%20%20canjs%2B%20%22can%20js%22,%20backbone.js%20%2B%20%20backbonejs%20%2B%20%22backbone%20js%22,%20ember.js%20%2B%20%20emberjs%20%2B%20%22ember%20js%22,%20knockout.js%20%2B%20knockoutjs%20%2B%20%22knockout%20js%22&cmpt=q)

[5&q=angular.js%20%2B%20angularjs%20%2B%20%22angular%20js%22,%20can.js%20%2B%20%20canjs%2B%20%22can%20js%22,%20backbone.js%20%2B%20%20backbonejs%20%2B%20%22backbone%20js%22,%20ember.js%20%2B%20%20emberjs%20%2B%20%22ember%20js%22,%20knockout.js%20%2B%20knockoutjs%20%2B%20%22knockout%20js%22&cmpt=q](http://www.google.com/trends/explore#cat=0-5&q=angular.js%20%2B%20angularjs%20%2B%20%22angular%20js%22,%20can.js%20%2B%20%20canjs%2B%20%22can%20js%22,%20backbone.js%20%2B%20%20backbonejs%20%2B%20%22backbone%20js%22,%20ember.js%20%2B%20%20emberjs%20%2B%20%22ember%20js%22,%20knockout.js%20%2B%20knockoutjs%20%2B%20%22knockout%20js%22&cmpt=q)

2.4.1 ¿Qué es MVC?

El MVC (Modelo Vista Controlador) es un patrón de diseño que en la actualidad ha tenido gran acogida en el medio de los desarrolladores debido a que es fácil de comprender por otros lo que facilita el desarrollo en grupos y organiza de manera clara y concisa la codificación de una aplicación.

Este patrón de diseño está comprendido por 3 partes que cumplen una función específica y por la cual se agrupa las diferentes partes de un software, esta son:

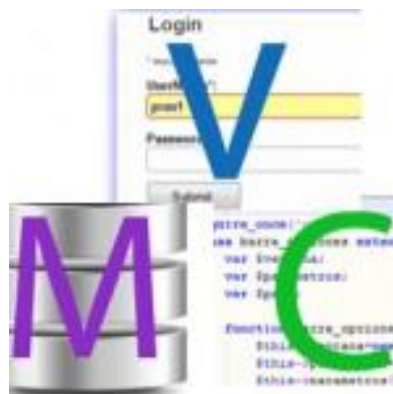


Figura 2.15 Partes del modelo MVC.

Fuente: <http://codigobase.com/wp-content/uploads/2012/06/mvc-150x150.jpg>

- **Modelo**

El modelo es el motor de la aplicación. Esto es por lo general los datos detrás la aplicación, generalmente obtenidos de una base de datos. Cualquier interfaz con datos, que el usuario ve, se deriva a partir del modelo, o un subconjunto del modelo. (Shyam & Brad, 2014)

El modelo es el núcleo de la aplicación, es una base de la misma ya que es el nivel más bajo, el que tiene acceso a los datos alojados en bases de datos,

ficheros planos u otros servicios. Este tiene la tarea de gestionar las estructuras de datos de la aplicación.

- **Controlador**

El controlador es la capa de lógica de negocio y presentación, que realiza acciones como el manejo de datos y su procesamiento, y toma decisiones como la forma de presentar el modelo, que partes de él para mostrar, etc. (Shyam & Brad, 2014)

Después de que un modelo de datos esta definido tenemos que controlarlo, es decir manejar distribuir, estos datos según la lógica del negocio, esta capa intermedia entre el modelo y la vista y tiene la tarea de comunicar a las dos capas.

- **Vista**

La vista es la interfaz que el usuario ve e con la que interactúa. Es dinámica, debe estar basada en el modelo de la aplicación. (Shyam & Brad, 2014)

La vista es todo lo que el usuario puede ver de una aplicación, y con lo que puede interactuar y esta debe ser un medio para que el contenido sea mostrado de manera fácil y accesible para el usuario brindándole una vista amigable y de fácil uso.

Este patrón de diseño sobre las aplicaciones web, se trabaja sobre el servidor y la parte de la vista sobre el cliente, pero la generación de las vistas y templates³⁴ los realiza el servidor tomando así una tarea que cuesta en rendimiento AngularJS, viene como una innovadora idea, tomando en cuenta la evolución de los equipos y que cada vez el rendimiento de los mismos mejora.

AngularJS retira algunas tareas de la vista del lado del servidor, y también del controlador dejando al servidor la única tarea de manejo de datos y conexión con la fuente de datos. Pasando así a ser un simple servicio similar a un **servicio web**³⁵ dejando al cliente el manejo, esto no quiere decir que el usuario tenga acceso a toda la lógica del negocio, parte de la lógica la más sensible queda en el servidor para evitar problemas de seguridad. La parte de enrutamiento y manejo de información pasa al lado del cliente.

2.4.2 ¿Porque usar AngularJS?

Según su página oficial:

HTML es ideal para declarar documentos estáticos, pero se tambalea cuando tratamos de utilizarlo para declarar vistas dinámicas en aplicaciones web. AngularJS permite ampliar vocabulario HTML para su aplicación. El ambiente resultante es extraordinariamente expresivo, legible y rápida de desarrollar. (GoogleInc, 2014)

³⁴ **Template:** Plantillas de diseño para su uso en diversas vistas de un aplicativo.

³⁵ **Web:** Es un conjunto de tecnologías y protocolos usados para el intercambio entre aplicaciones diferentes.

El sitio de aprendizaje (CodeSchool, 2015) nos da las siguientes razones por las que esta tecnología debería ser usada en el desarrollo de nuestras aplicaciones:

- Organiza el código JavaScript
- Ayuda a la creación de sitios responsivos³⁶ de manera rápida.
- Tiene una buena relación con JQuery³⁷ (AngularJS usa una versión pequeña de JQuery en caso de que no esté presente dentro de nuestra aplicación)
- Es fácil de probar.

2.4.3 Filosofía

Los autores (Shyam & Brad, 2014) definen 5 aspectos principales que son la filosofía de AngularJS y que definen las principales características de este framework a su perfección y son las siguientes:

- **Data-Binding (enlazado de datos)**

En las aplicaciones los enlaces de datos son medios por los cuales un componente se enlaza a un dato específico y cambia automáticamente si este dato lo hace brindando una aplicación interactiva.

³⁶ **Responsivo:** se refiere a que es compatible con diversos dispositivos.

³⁷ **JQuery:** es una librería de javascript para mejorar su rendimiento.

Dentro de las aplicaciones web esto es costoso debido a que cuando se requiere cambiar una parte de una página se requiere de cierto código ya sea **javascript** o usando **JQuery** se requiere de al menos 4 líneas de código para cambiar una parte de la página y esto de solo escritura o lectura e intercalando constantemente entre estos caminos para así tener una aplicación interactiva.

AngularJS nos provee una forma de enlazar variables con datos para que en cualquier lectura o escritura el enlace se encargue del dato y su manejo.

- **Declarativo**

El framework permite declarar nuestras propias etiquetas o atributos **HTML** mediante directivas que contienen el comportamiento y funcionalidad de estas nuevas etiquetas o atributos. Esto es debido a que AngularJS está orientado a un paradigma declarativo al contrario de una aplicación web tradicional que usa paradigmas imperativos en donde se sabe la funcionalidad de un elemento por su ubicación o clase más no por su nombre.

Gracias a esta modalidad de trabajo se puede mejorar la interpretación del código facilitando el trabajo en grupos y el mantenimiento del mismo. Mejorando la funcionalidad de **single-page aplicación** con la que se trabaja al usar este framework, que consta de evitar la recarga de la página y solo se recargan

mediante AJAX³⁸ los componentes necesarios mediante la llamada a fragmentos de código en el servidor.

- **Separación de contenido**

AngularJS al trabajar con del modelo MVC tiende a organizar y separar su código y funcionalidades, lo que facilita la refacción de código y su mantenimiento separando las funcionalidades de cada componente.

Este framework separa sus funcionalidades en algunas características como son:

- Enrutamiento
- Consumo de servicios
- Controladores
- Directivas
- Animaciones

Cada una de las funcionalidades anteriormente mencionadas están en diferentes partes de lo que el framework nos ofrece, separadas inclusive en diferentes librerías y archivos para evitar que se tenga código que no se va a usar.

³⁸ **AJAX**: es una tecnología de javascript que nos permite hacer llamadas asincrónicas al servidor.

- **Inyección de dependencias**

AngularJS usa métodos de inyección de código para poder hacer llamado a los diferentes servicios que posee y que se generan, mediante la inyección se puede importar servicios dentro de otros módulos o inclusive dentro de servicios.

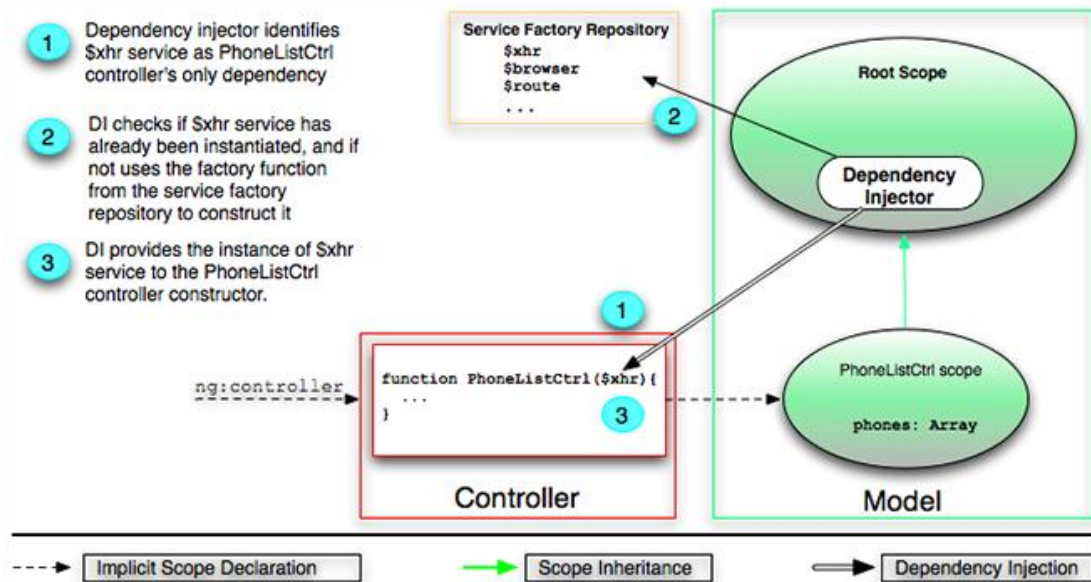


Figura 2.16 Dependency Injector AngularJS

Fuente: <https://gist.github.com/jhiguxin/10019060>

Como en la imagen anterior se muestra el motor de AngularJS se encarga de pasar como parámetros los servicios hacia las funciones de los módulos o servicios que se crea.

Dentro de los servicios de AngularJS existe un estándar en el cual todos los servicios que ofrece tienen el prefijo \$, por eso recomiendo no usar este prefijo en la creación de módulos o servicios para evitar posibles problemas.

- **Extensible**

AngularJS además de proveer las directivas que anteriormente se mencionó, nos permite mejorarlas, con esto se puede tomar directivas que vienen ya creadas y mejorarlas o tomar directivas de otros y así mismo mejorarlas o cambiar su comportamiento.

En el núcleo de este framework vienen integradas ya algunas directivas que son las que dan gran funcionalidad al código HTML como son las directivas de tipo atributo **ng** que son las principales de este framework. Entre algunas de estas están:

- **ng-model** define un modelo de datos para un elemento.
- **ng-hide** oculta el elemento según el parámetro que se le pasa.
- **ng-repeat** tiene la funcionalidad de un ciclo **for**
- **ng-controller** define un controlador para un elemento en el HTML

- **Pruebas**

AngularJS está diseñado para ser probado gracias a una librería conocida como **Protractor**³⁹ que nos brinda la posibilidad de realizar diferentes pruebas con diferentes escenarios mediante código esto facilita la realización de pruebas unitarias y funcionales y su documentación.

³⁹ **Protractor**: es el nombre del proyecto de AngularJS en los repositorios de GIT.

2.5 MONGODB



Figura 2.17 Logo de mongoDB

Fuente: <http://www.mongodb.com/>

Hablar de mongoDB es hablar de una nueva corriente que toma fuerza dentro del mundo de las bases de datos, en donde las bases de datos relacionales han tenido una larga y próspera monarquía, donde han sido los reyes de los datos por mucho tiempo pero desde abajo las nuevas corrientes vienen trayendo consigo tecnologías nuevas y esperan derrocar a este grande.

Estas nuevas tecnologías se podría decir que son relativamente nuevas por el crecimiento que han venido teniendo en los últimos tiempos pero viene ya desde hace mucho tiempo atrás, pero con los nuevos paradigmas que encontramos ahora es que han tenido un gran empujón.

Los grandes las usan: Google, Facebook, Ebay, Adobe, entre otros, pero no es por una moda pasajera es si no por su eficiencia al trabajar con inmensas cantidades de datos. Estas bases de datos no relacionales como se les conoce en el medio, realmente pertenecen a una clasificación de bases de datos, los autores (Eric & Jim R., 2012) nos proponen la siguiente clasificación:

Nombre	Descripción
Relacionales	<p>Unas de las más conocidas, hacen uso de un lenguaje SQL ⁴⁰ y organizan los datos en tablas bidimensionales (filas y columnas). Para obtener estructuras de datos hacen uso de relaciones entre tablas usando una o varias columnas. Entre las más conocidas PostgreSQL, Oracle, MySQL, SQLServer, etc.</p>

Llave-Valor (Key-Value)

Usan una de las estructuras de datos más simples de llave-valor, conocidas en los lenguajes de programación como mapas o tablas hash en donde los valores se almacenan tiene un código por el cual son buscados y gestionados. Entre estas tenemos a **Apache**

⁴⁰ **SQL**: lenguaje de consultas estructurado para manejo de bases de datos.

Cassandra, Riak, Redis.

Orientadas a columnas

Usadas por los grandes como lo es Google con su base de datos **BigData**, estas tienen una filosofía de crecimiento horizontal, es decir se insertan columnas con facilidad y cada fila puede tener cero o varias columnas evitando así los campos nulos. Una de las grandes dentro de este campo es **BigData** de Google pero existe una implementación gratuita de esta que no es igual pero tiene gran similitud. **HBase** es una base de datos orientada a columnas con grandes características y modelos similares a la grande de Google.

Documentales

Este modelo de almacenar datos se basa en que cada objeto dentro del lenguaje de programación es un documento y es almacenado con un id único dentro de una colección de documentos, estos documentos son

estructuras de datos tan simples como un solo dato hasta estructuras complejas con datos anidados. Estas bases de datos han tomado en los últimos tiempos una gran acogida debido a su versatilidad y simplicidad de modelo de datos. Las más grandes dentro del mercado son **mongoDB y couchDB.**

Graficas

Unas de las menos conocidas y no tan usadas son las bases de datos graficas que orientan su modelo a estructuras de datos como los son lo grafos, donde presentan nodos enlazados entre sí, esto nos ayuda cuando tenemos datos con numerosos enlaces entre sí. En esta clasificación están bases de datos como **Neo4J, Polyglot**

Tabla 2.3 Tabla clasificatoria de las base de datos.

Fuente: Propia

Teniendo en cuenta un panorama más amplio sobre esta nueva corriente, se puede tomar decisiones sobre cómo se almacenaran los datos que son el elemento más importante dentro de un sistema informático.

MongoDB es una base de datos documental de código abierto que proporciona un alto rendimiento, alta disponibilidad y escalamiento automático. (mongodb, 2014)

MongoDB es una poderosa, flexible, y escalable base de datos para propósitos generales. Combina la habilidad de escalar con características como índices secundarios, rangos de clasificación, agregaciones e índices geoespaciales. (Chodorow, 2013)

Desarrollada sobre C++ esta base de datos promete eficacia y eficiencia, fue lanzada el 2009 y actualmente se encuentra en su versión 2.6.0 y licenciada bajo **GNU AGPL v3.0**⁴¹, Esta base de datos hace uso del popular y conocido modelo **JSON**⁴² para el almacenamiento pero en su forma binaria para ahorro de espacio.

Las principales diferencias entre esta base de datos con algunas de las más populares actualmente son mostradas en la siguiente tabla:

⁴¹ **Licencia publica general (Generic Public Licence):** es un formato de licencia de acceso público.

⁴² **JSON (JavaScript Object Notation):** notación para definir objetos o estructuras de datos complejas en javascript.

		MongoDB	Relacional	Valor-clave
Rich Data Model		Sí	No	No
Esquema dinámico		Sí	No	Sí
Con tipo de datos		Sí	Sí	No
Datos Localidad		Sí	No	Sí
Actualizaciones Campo		Sí	Sí	No
Fácil para los programadores		Sí	No	No Cuando Modelización de estructuras complejas de datos

Tabla 2.4 Diferencias entre mongo y otros tipos de bases de datos.

Fuente: <http://www.mongodb.com/mongodb-architecture>

Esta base de datos, actualmente dentro del mercado, tiene gran acogida debido a su modelo de datos simple y flexible que nos permite cambios que en los modelos

relacionales son demasiados costos o en casos más extremos requieren un rediseño del modelo de datos. MongoDB usa el modelo documental pero a su vez también puede usar modelos relacionales debido a su flexibilidad permitiendo a los desarrolladores cambiar de base de datos de una manera más fácil.

2.5.1 Documentos

Los documentos son estructuras de datos, que en el caso de mongoDB son estructuras de datos **BJSON**⁴³.

Dentro del manual oficial de mongoDB nos brinda la siguiente definición de los documentos:

Un registro de mongoDB es un documento, que es una estructura de datos compuestos por pares de campo y de valor. Los documentos tienen la estructura de los objetos JSON. Los valores de los campos pueden incluir otros documentos, vectores y vectores de documentos. (mongodb, 2014)

```
{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}
```



The diagram shows a JSON document structure with four lines of code. To the right of each line, there is a blue arrow pointing left towards the colon in the key-value pair, followed by the text "field: value".

Figura 2.18 Ejemplo de estructura de un documento de mongoDB

⁴³ **BJSON**: es la versión binaria de JSON.

Fuente: <http://www.mongodb.com/mongodb-architecture>

Para tener un mejor concepto de lo que son los documentos en comparación con una estructura de una base de datos relacional como son las tablas, un documento es una fila dentro de una tabla, y una colección es la tabla en sí. Los documentos como las tablas pueden tener referencias a otros documentos y la diferencia con los modelos tradicionales de bases de datos relacionales es que estos pueden tener documentos dentro no como referencias si no como datos embebidos dentro del mismo.

La ventaja de tener documentos embebidos es la reducción de juntas comúnmente conocidas como **JOINS** dentro de los modelos relacionales. Estas juntas son costosas dentro de una base de datos porque comparar dos tablas registró por registro buscado las coincidencias, esto dentro de cantidades de datos gigantes es un gran inconveniente, en el caso de mongoDB estos datos ya vienen dentro de mi documento por ende no es necesario hacer juntas.

Otra característica importante del modelo de datos de mongoDB es su flexibilidad. Con esto nos referimos a que sus modelos de datos no son estáticos, no se definen una vez y cambiarlos deja de ser un completo dolor de cabeza para los desarrolladores. Dentro de una colección puede haber varios documentos con estructuras de datos distintas. La base de datos no trabaja con esquemas simplemente es un almacén de datos según como la aplicación envíe los objetos, estos se convertirán en documentos y se podrán dentro de una colección.

La flexibilidad es una ventaja increíble nos deja tener modelos de datos cambiantes, pero también es una desventaja si se usa de manera errónea ya que al no ser fijos los modelos de datos el mantenimiento de las aplicaciones se dificulta, este debido a tener diferentes modelos de datos.

Los autores de “**Seven Databases in seven weeks**” después de un análisis de las principales bases de datos nos brinda una perspectiva de la principal fortaleza y debilidad de esta base de datos:

- **Fortaleza**

La fuerza principal de Mongo reside en su capacidad para manejar grandes cantidades de datos (una enorme cantidad de solicitudes) por la replicación⁴⁴ y escalamiento horizontal que ofrece. Pero también tiene un beneficio adicional de un modelo de datos muy flexible, ya que no necesita siempre ajustarse a un esquema y simplemente puede anidar a datos complejos que en el caso de los modelos relacionales se lograría con JOINS (Eric & Jim R., 2012)

- **Debilidad**

MongoDB promueve la desnormalización⁴⁵ de modelos de datos (por no tener ninguno fijo). Que es un poco difícil de pasar para algunos desarrolladores. Algunos desarrolladores encuentran tranquilizador las restricciones de las bases

⁴⁴ **Replicación:** es tener varias instancias iguales de una base de datos en caso de que una de estas falle entre la otra como respaldo, son un espejos o copias de una base de datos.

⁴⁵ **Desnormalización:** es el proceso de inverso a normalizar a una base de datos.

de datos relacionales. El simple hecho de insertar valores con modelos antiguos puede traer horas y horas de problemas dentro el funcionamiento de la aplicación. La flexibilidad de Mongo generalmente no es importante si su modelo de datos ya es bastante maduro y bloqueado. (Eric & Jim R., 2012)

2.6 SERVICIOS REST (Representational State Transfer)

Previo a adentrarnos sobre el concepto de esta tecnología hay que tener claro conceptos sobre el protocolo HTTP⁴⁶ que son necesarios para la correcta aplicación y funcionamiento de REST.

El **rfc**⁴⁷ de HTTP nos define este protocolo como:

HTTP es un protocolo sin estado petición / respuesta que opera por el intercambio de mensajes a través de un transporte- fiable o capa de sesión "conexión" (Sección 6). Un "cliente" HTTP es un programa que establece una conexión con un servidor para el propósito del envío de una o más solicitudes de HTTP. Un "servidor" HTTP es un programa que acepte conexiones con el fin de atender peticiones HTTP y devolver respuestas HTTP. (Fielding & Reschke, s.f.)

⁴⁶ **HTTP** Protocolo de transferencia de hipertexto

⁴⁷ **RFC** son documentos que definen tecnologías creados por expertos.

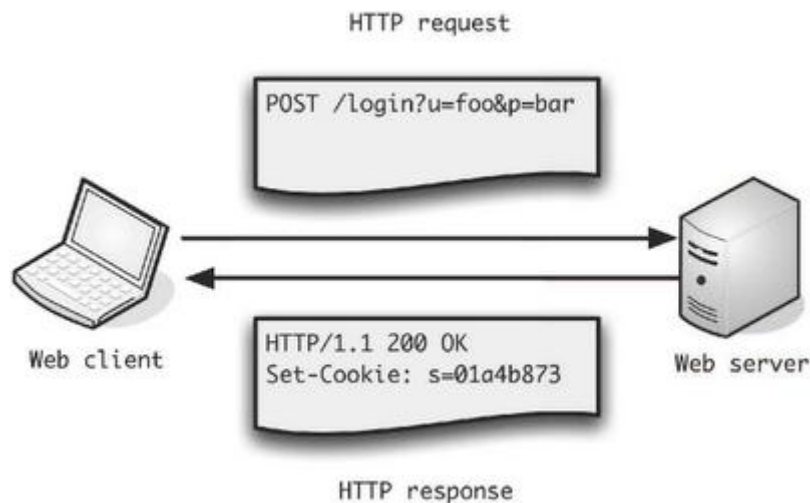


Figura 2.19 Diagrama de una petición y respuesta en el protocolo HTTP

Fuente: <http://pratapreddypilaka.blogspot.com/2011/11/cookie-capturing-using-httpwebrequest.html>

- **Métodos del protocolo**

Las especificaciones HTTP han definido un conjunto de métodos de petición comunes. Por ejemplo, el método **GET** consigue un documento desde un servidor, el método **POST** envía los datos a un servidor para su procesamiento, y el método **OPTIONS** determina las capacidades generales de un servidor web o las capacidades de un servidor web para un recurso específico. (Gourley & Totty, 2002)

Método	Descripción	Tiene contenido (body)
GET	Obtiene un recurso del servidor.	NO

HEAD	Obtiene únicamente las cabeceras de un recurso.	NO
POST	Envía datos al servidor para su procesamiento.	SI
PUT	Guarda el contenido del body en el servidor.	SI
TRACE	Traza la ruta del mensaje a través de los servidores proxy hasta el servidor.	NO
OPTIONS	Determina que métodos operan en el servidor.	NO
DELETE	Remueve un recurso del servidor.	NO

Tabla 2.5 Métodos básicos del protocolo HTTP

Fuente: HTTP The Definitive Guide (Gourley & Totty, 2002)

Muchos servidores pueden implementar sus propios métodos ya que HTTP es un protocolo moldeable que permite esto. Cada servidor tendrá uno o algunos de los métodos antes mencionados y también podrá tener sus propias implementaciones de métodos.

Los autores de “**REST in practice**” (Webber, Parastatidis, & Robinson, 2010) comentan:

REST describe la web como una aplicación distribuida de hipermedia que permite una comunicación de recursos enlazados mediante el intercambio de la representación del estado del recurso.

Dentro de otras definiciones encontramos:

REST define un conjunto de principios de la arquitectura mediante los cuales se puede diseñar servicios web que se centran en los recursos del sistema, incluyendo cómo se abordan y se transfieren a través de HTTP por una amplia gama de clientes escritos en diferentes lenguajes. (Rodríguez, 2008)

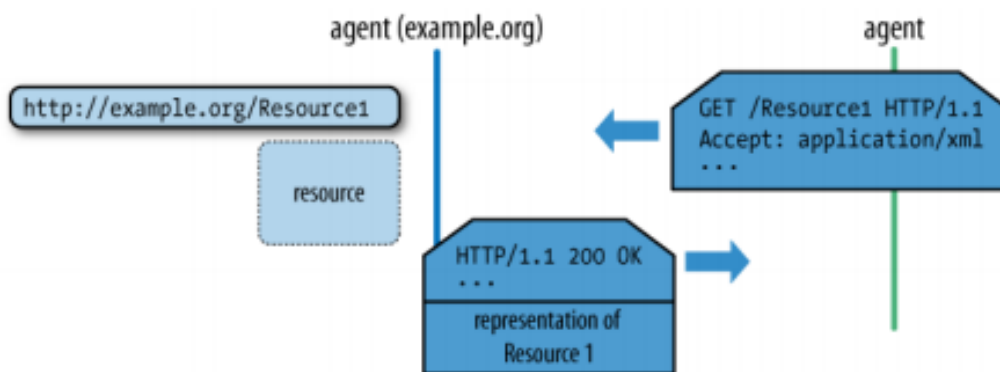


Figura 2.20 Uso de HTTP mediante el método GET para obtener un recurso

Fuente: (Webber, Parastatidis, & Robinson, 2010)

REST es una arquitectura web usada dentro de sistemas distribuidos que permite el intercambio de recursos mediante representaciones de los mismos e

intercambiando sus estados mediante los métodos HTTP como GET, POST, OPTIONS, etc.

Para poder identificar los recursos REST hace uso de URI que son los identificadores que permiten ubicar un recurso mediante la dirección del mismo por ejemplo www.organizacion.com/recurso

A diferencia de otras arquitecturas distribuidas esta pone énfasis en la definición de los recursos más no de los métodos de cada recurso, esto debido a que al hacer uso de HTTP usa los métodos de este para definir las operaciones de los recursos que brinda.

Por ejemplo se tiene un recurso: www.organizacion.com/recurso.

La arquitectura usa:

- GET para obtener el recurso.
- POST para crear un recurso
- PUT para modificar un recurso
- DELETE para eliminar un recurso

2.6.1 Servicios RESTful

Hablar de estos servicios es hablar de REST, estos servicios son una implementación de servicios web usando la arquitectura REST, los servicios

RESTful son una alternativa a los conocidos servicios web que hacen uso de SOAP⁴⁸.

En conclusión los servicios RESTful son la implementación de REST dentro de nuestras aplicaciones para poder armar una arquitectura orientada a servicios.

Con los nuevos paradigmas de programación y la variedad de dispositivos la programación ha tomado un camino orientado a los servicios para así desligarse de la parte visual o en que dispositivo funcionara la aplicación, esto debido a como se mencionó anteriormente la gran variedad de dispositivos que existen en el mercado.

Gracias a las nuevas arquitecturas como MVC que nos permiten separa la lógica de lo visual tenemos una alternativa para diseño de aplicaciones pero aplicar estos diseños arquitectónicos es complejo dentro de aplicaciones web. La programación orientada a servicios viene como una alternativa en la que se desarrolla todo como servicios y los clientes son los que se encargan de decorar los resultados de los servicios.

Los servicios web (Web Services) basados en SOAP son una alternativa para programar orientado a servicios y permite tener diferentes clientes independientemente del lenguaje o dispositivo que usen gracias a esto el mantenimiento y la escalabilidad de las aplicaciones mejora. Pero estos tienen una gran desventaja y es la gran cantidad de middleware que necesitan y la

⁴⁸ **SOAP:** Protocolo simple de acceso a objetos para creación de servicios web.

complejidad a la hora de su desarrollo, pero por otro lado son servicios con una gran seguridad.

Por otro lado y como una alternativa a los servicios web basados en SOAP viene una nueva corriente de servicios web basados en REST que son los **RESTful web services**. Estos nos brindan una manera más simple y fácil de implementar servicios sin tanto middleware y haciendo uso de los métodos del protocolo de HTTP.

Si se mide por el número de servicios web actualmente en la red que lo utilizan REST o SOAP, REST ha surgido en los últimos años como un modelo de diseño de servicios Web predominante. De hecho, REST ha tenido un impacto tan grande en la web que ha desplazado a SOAP y WSDL en el diseño de servicios web, porque es un estilo mucho más sencillo de usar. (Rodríguez, 2008)

2.6.2 REST vs SOAP

Las principales diferencias entre estas tecnologías son expuestas en la siguiente tabla:

REST	SOAP
Permite el uso de JSON y XML para la transferencia de datos.	Solo permite el uso de XML.

Soporta SSL⁴⁹

Soporta SSL y WS-Security

Caching de lecturas

No soporta el caching de lecturas

Usa HTTP para transferencia y manejo de datos

Usa HTTP solo para la transferencia de datos

No define tipos de datos

Define tipos de datos

Tabla 2.6 Diferencias entre SOAP y REST

Fuente: Propia

Después de tener muy en cuenta las diferencias de estas tecnologías la decisión de cual se debe aplicar recae dentro de lo que se desee cada una de estas presenta grandes ventajas pero dependiendo de lo que se requiera hacer.

⁴⁹ **SSL:** capa de seguridad de sockets para seguridad en transferencia de datos.



CAPITULO III



- 6. MARCO LÓGICO
 - 6.1. DEFINICIÓN
 - 6.2. METODOLOGÍA
 - 6.3. HERRAMIENTAS

3 MARCO LÓGICO

3.1 DEFINICIÓN

La Metodología de Marco Lógico es una herramienta para facilitar el proceso de conceptualización, diseño, ejecución y evaluación de proyectos. Su énfasis está centrado en la orientación por objetivos, la orientación hacia grupos beneficiarios y el facilitar la participación y la comunicación entre las partes interesadas. (Ortegón, Pacheco, & Prieto, 2005)

Esta metodología para gestión de proyectos nace debido a la falta de correlación entre objetivos y actividades dentro de los proyectos, esta metodología ofrece una secuencia de pasos para la correcta identificación del problema principal determinado mediante una secuencia de pasos y dándonos ciertas pautas para lograrlo de la manera correcta, así mismo nos brinda métodos fáciles de usar para poder determinar los objetivos y sus correspondientes actividades.

El principal producto derivado de esta metodología es la **matriz de marco lógico** que es un organizador grafico en el que constan los elementos esenciales de un proyecto correlacionándolos de manera simple y resumida. Este organizador grafico es el producto final después de un proceso que la metodología brinda, y principalmente consta de los siguientes aspectos mostrados en la siguiente figura:

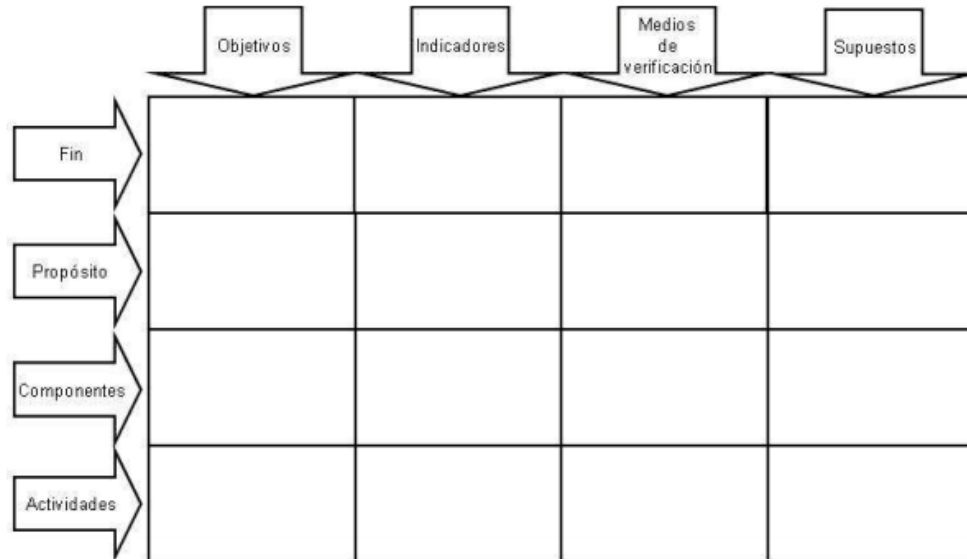


Figura 3.1 Formato de la matriz de marco lógico

Fuente: (Aldunate & Córdoba, 2011)

Esta matriz permite de manera concreta y rápida entender como está estructurado un proyecto, cual se su alcance y de qué forma se garantiza su ejecución todo en una matriz que relaciona en la que cada celda está relacionada con las demás.

La metodología es aplicable para todo tipo de proyectos y sirve como un indicador de calidad ya que se asegura de generar un proyecto con una correcta estructura y correlación.

3.2 METODOLOGÍA

Esta es conocida como metodología de marco lógico debido a su forma de ser, es decir esta consta de una secuencia de pasos en los que cada uno necesita del anterior para poder proseguir, y en cada paso se nos brinda un método para cumplir con los objetivos del mismo.

Los pasos de esta metodología constan de la siguiente estructura mostrada en la siguiente imagen:

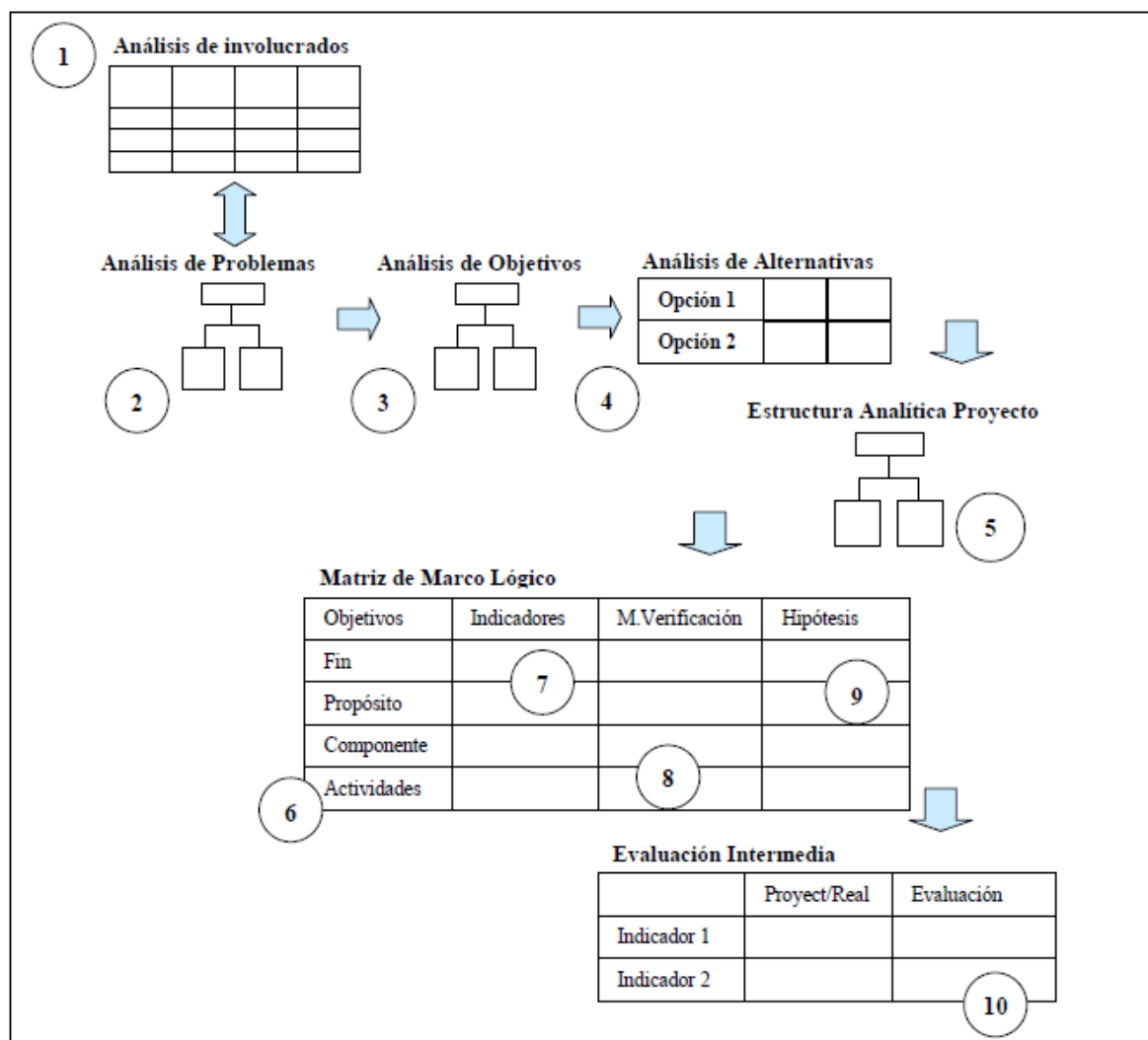


Figura 3.2 Estructura de la metodología de marco lógico

Fuente: (Ortegón, Pacheco, & Prieto, 2005)

3.2.1 Análisis de involucrados

La metodología marco lógico contempla como factor importante la participación de los principales involucrados desde el inicio del proceso, por lo tanto identificar los grupos y organizaciones que pudieran estar directa o indirectamente relacionados con el problema y analizar su dinámicas y reacciones frente al avance del proyecto, permitirá darle mayor objetividad al proceso de planificación y concitar acuerdos entre involucrados, al considerar diversos puntos de vista y fomentar un sentido de pertenencia por parte de los beneficiarios. (Ortegón, Pacheco, & Prieto, 2005)

Este es el paso introductorio a la generación de un proyecto porque como requisito primordial primero determinaremos quienes son los afectados y participantes del proyecto para tener ya una visión panorámica de lo que nos espera.

Identificando a los involucrados se tendrá una primera impresión de viabilidad del proyecto tomando en cuenta ya la disponibilidad de los involucrados y su grado de relación con el proyecto.

3.2.2 Análisis del problema

Con este análisis se trata de identificar los problemas reales que afectan al colectivo beneficiario con el que se quiere trabajar y no los problemas posibles, potenciales o futuros. Conviene señalar que un problema no es la ausencia de una solución, sino un estado negativo existente. Por ejemplo, la falta de un centro de salud no es un problema, el problema es la fuerte incidencia de enfermedades en una determinada población. (Sanchez, 2007)

Para poder realizar esta difícil tarea la metodología hace uso de una herramienta conocida como árbol de problemas, que es un conjunto de recuadros conectados por líneas. Se divide en tres segmentos verticales el segmento inferior es en se encontraran las causas de problema, en la parte central deberá ir el problema central a resolver mediante este proyecto que como se explicó antes los problemas no son la ausencia de algo si no un estado negativo. Y en la parte superior del mismo tendremos los efectos de cada causa.

En este paso lograremos identificar el problema central que bien puede ser con la idea que se partió al iniciar el proyecto o bien esa idea es una simple causa del verdadero problema.

Al finalizar se obtendrá una estructura como lo muestra la siguiente figura:

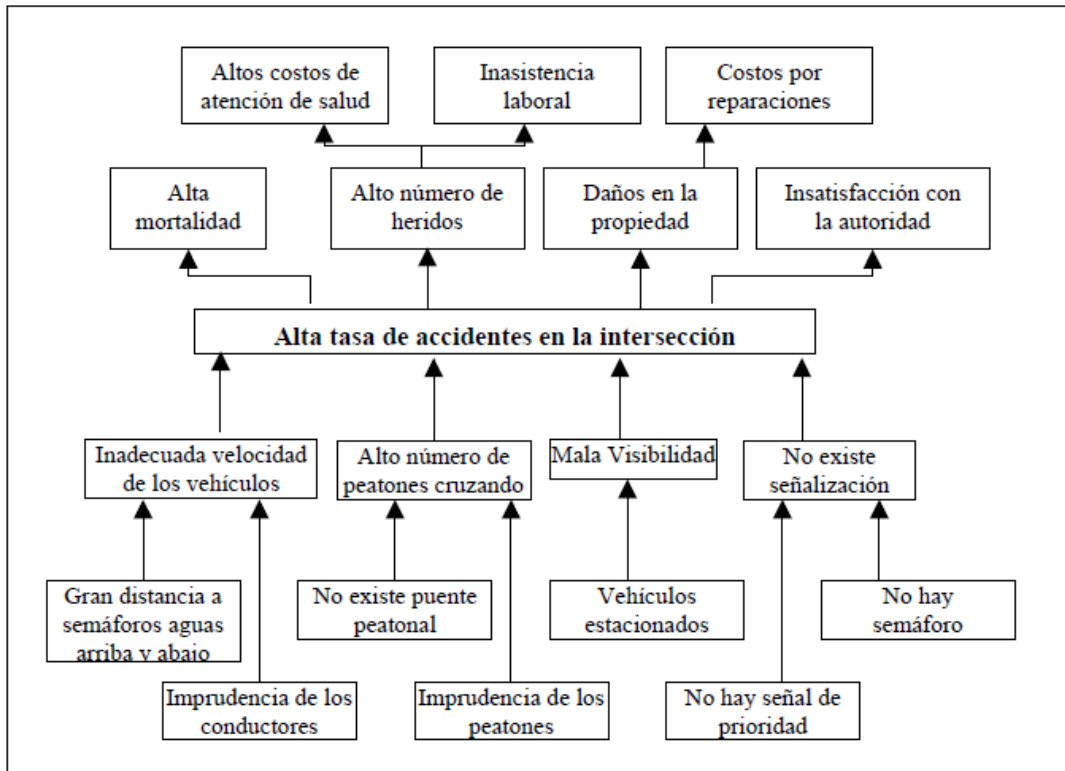


Figura 3.3 Ejemplo de un árbol de problemas. En base a un supuesto problema.

Fuente: (Ortegon, Pacheco, & Prieto, 2005)

3.2.3 Análisis de objetivos

Según (Camacho, Cámara, Cascante, & Sainz):

El análisis de objetivos es un paso de la identificación de un proyecto de desarrollo que se construye sobre los resultados obtenidos en el anterior análisis de los problemas. Los problemas que habían sido descritos como “situaciones negativas percibidas como tal por algunos de los implicados” pasan ahora a ser definidos como “estados alcanzados positivos que se establecen sobre la resolución de los problemas anteriormente identificados”. Es decir, para el

enfoque del marco lógico, los objetivos de desarrollo se construyen sobre la solución de problemas concretos que afectan a personas concretas y cuya definición y relaciones se han establecido en el paso anterior. De esa manera, se trata de construir un árbol de objetivos que, en principio, es una copia en positivo del árbol de problemas, pero donde la relación causal pasa a convertirse en una relación de carácter instrumental, donde las tarjetas inferiores son los medios para alcanzar las superiores que, con respecto a aquellas, son los fines que se esperan alcanzar.

Como los autores mencionan este paso está relacionado con el anterior y si no se completó el anterior no se puede realizar este, por esto la metodología de marco lógico es simple y secuencial lo que cada paso será el inicio del siguiente brindando una relación total entre los componentes del proyecto.

En este paso se usa el diagrama generado anteriormente para poder generar otro diagrama conocido como árbol de objetivos que consta de la misma estructura. Cambia en el aspecto de nomenclatura de los segmentos. Es un conjunto de recuadros unidos por líneas hacia el centro que se divide en tres segmentos verticales, solo que en este caso el segmento inferior es conocido como medios, el segmento central es el objetivo principal del proyecto y la parte superior son los fines.

Para poder llegar a este diagrama se toma el diagrama de árbol de problemas y en cada recuadro se toma el texto y se lo traduce a un estado positivo.

Al final tomando en cuenta el grafico anterior (**Figura 3.3**) el árbol de objetivos quedaría de la siguiente manera como se expone en la siguiente figura:

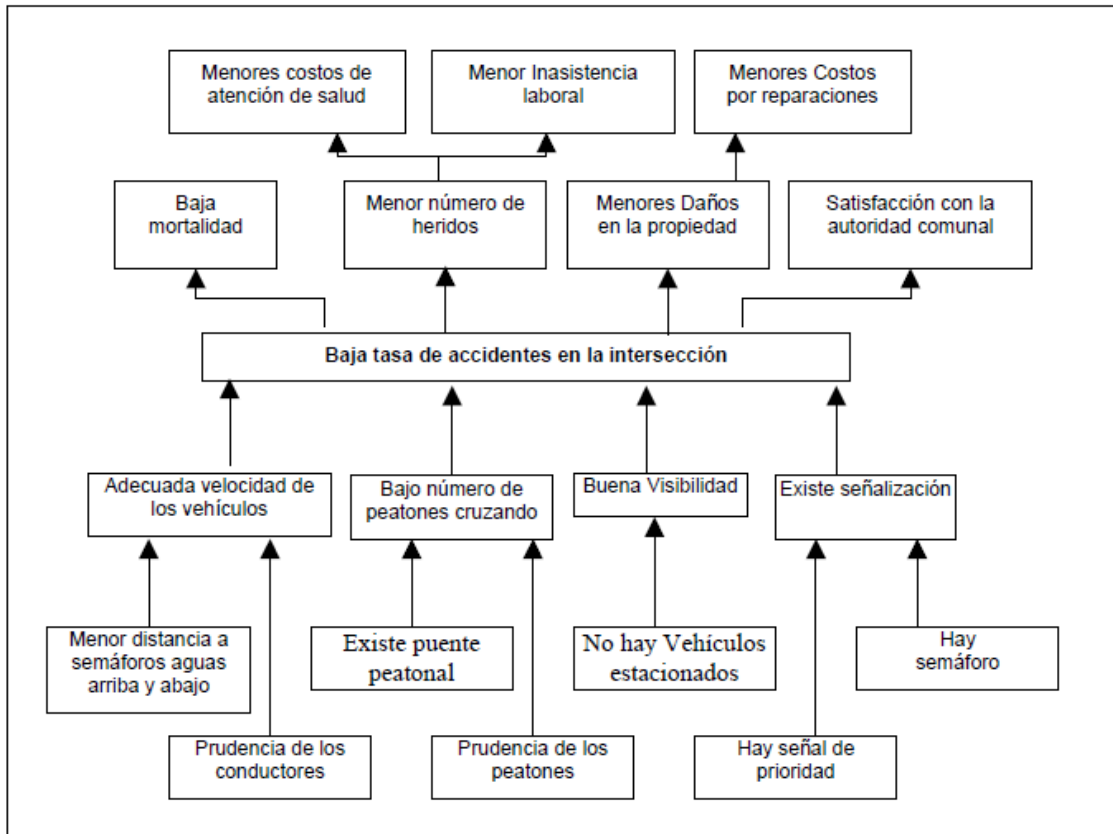


Figura 3.4 Árbol de objetivos en base al problema en la Figura 3.3

Fuente: (Ortegón, Pacheco, & Prieto, 2005)

3.2.4 Análisis de alternativas

Una vez completados los pasos anteriores se tendrá toda la información necesaria para poder generar las alternativas para dar solución a este problema que es sobre el cual el proyecto va a tratar.

El análisis de alternativas es un paso fundamental dentro de la gestión de una intervención, aunque inevitablemente presenta un nivel de indefinición que resulta enojoso a la hora de plantear una explicación de carácter más bien esquemático y superficial. Dicho de una manera rápida, puede decirse que la función básica del análisis de alternativas, también llamado discusión de estrategias según la Comisión Europea, es la de comparar las diferentes opciones que pueden identificarse en el árbol de objetivos, rechazando las que no nos parezcan deseables o presentan un grado excesivo de incertidumbre y seleccionando, dentro de la alternativa que parece óptima, la tarjeta-objetivo que va a convertirse en el objetivo del futuro proyecto de desarrollo. (Camacho, Cámara, Cascante, & Sainz)

Según los autores este paso es fundamental para la ejecución de un proyecto debido a que identificara las diferentes estrategias y conjuntos de acciones para dar solución al problema que bien pueden ser las que se tenía como idea al iniciar el proyecto o nos brinde mejores alternativas con mejores resultados.

Lo que se pretende en este paso es hacer uso de los organizadores generados en los pasos anteriores y generar una lista de acciones y agruparlas según su viabilidad. Según cada medio dentro del árbol de objetivos se sacara una acción para llegar a ese medio. Tomando en cuenta la siguiente relación entre competes:

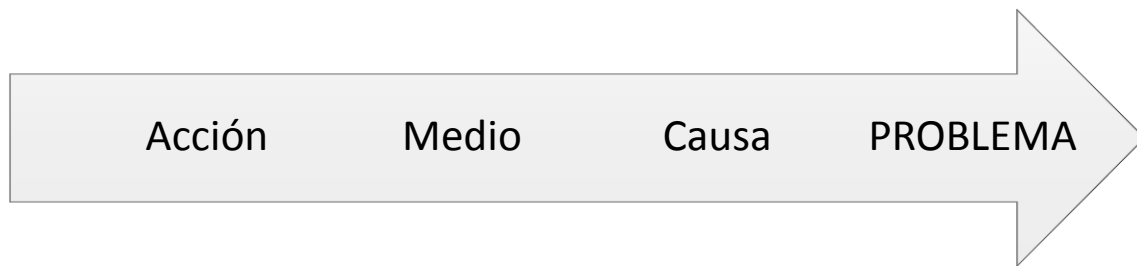


Figura 3.5 Relación entre componentes de los arboles

En base al ejemplo usado para la explicación de la metodología en base a las Figura 3.3 y Figura 3.4 se obtendrán las siguientes acciones:

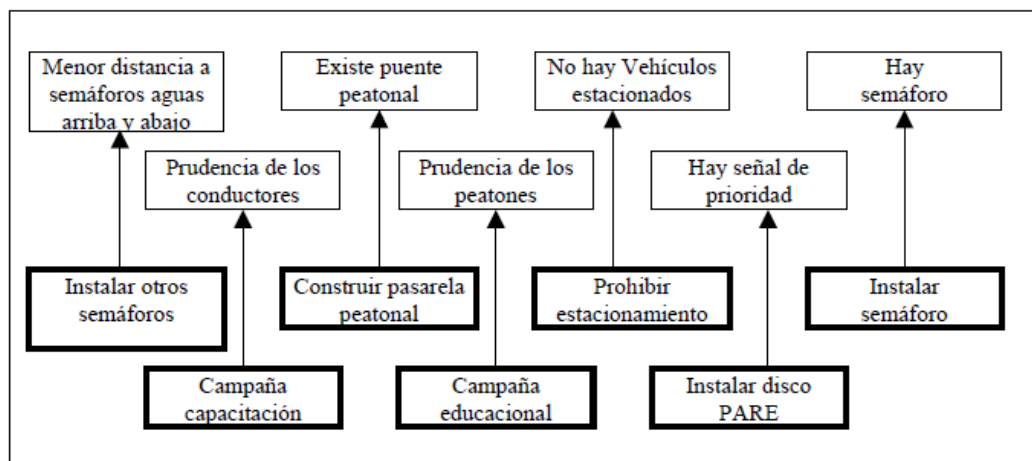


Figura 3.6 Lista de acciones en base al supuesto problema planteado en la

Figura 3.3

Fuente: (Ortegón, Pacheco, & Prieto, 2005)

Una vez identificadas las acciones se las agrupa para llegar a obtener una alternativa, pero antes de ello se las clasificara dependiendo de su acción a tomar en:

Complementarias: Son aquellas que pueden ir en conjunto con otras, es decir no interfieren con el desarrollo de otra acción. Un ejemplo de acciones completaría en base a la Figura 3.6 Es la campaña de educación porque esta no interfiere con ninguna otra acción y puede realizarse en conjunto con las demás sin afectar a ninguna otra.

Excluyentes: Estas al contrario de las complementarias solo pueden ejecutarse una no varias solo se escogerá una, esto debido a que si se realizara una de estas ya no se podría realizar otra.

Un ejemplo para una acción excluyente es la instalar un semáforo porque si se instala un semáforo esta acción choca con la de instalar una señal de pare por lo que son excluyentes entre si y no pueden estar dentro de la misma alternativa.

Una vez clasificadas las acciones se agruparan en base a las acciones excluyentes que son las que definen cada alternativa y se generaran las diferentes alternativas.

Para tomar la mejor alternativa se verificara la viabilidad de cada una y se tomara la más eficaz y eficiente según el criterio del autor del proyecto. Tomando en cuenta parámetros como tiempo y costo.

3.2.5 Estructura analítica del proyecto

El siguiente paso dentro de la metodología de marco lógico es generar un diagrama que simplificara el proyecto tomando como base el árbol de objetivos generado anteriormente y añadiendo otro segmento en la parte inferior en donde

se alinearan las acciones correspondientes de la alternativa escogida en el paso anterior, alienaremos las acciones correspondientes a cada medio y obtendremos la siguiente estructura:

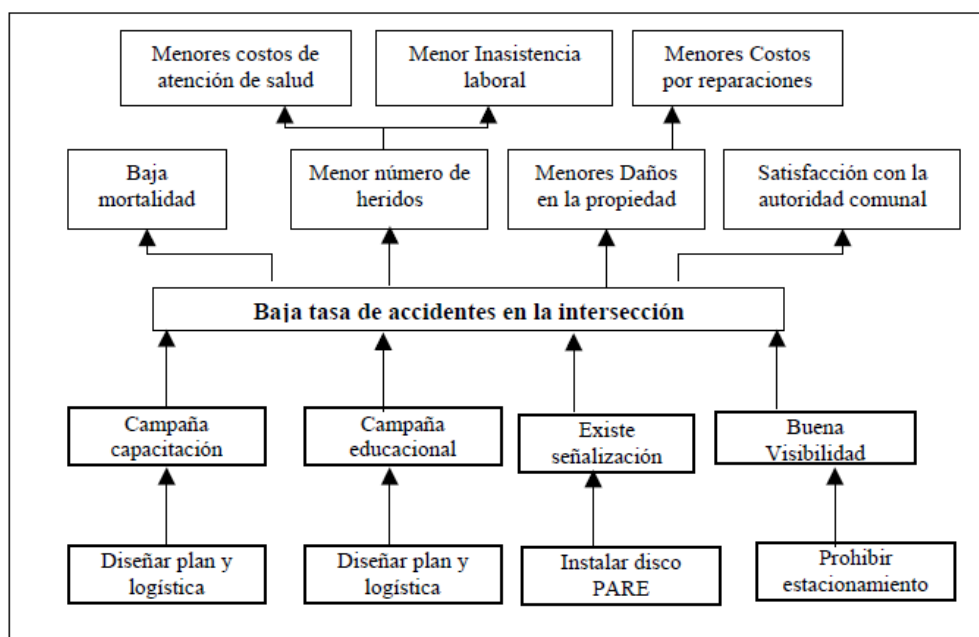


Figura 3.7 Estructura analítica del proyecto sobre el problema planteado en la

Figura 3.3

Fuente: (Ortegón, Pacheco, & Prieto, 2005)

Con esta estructura se puede tener claro cada acción que se va a realizar y el camino que nos llevara a un fin en particular que son los resultados esperados al realizar dicha acción.

3.2.6 Matriz de marco lógico

La matriz de planificación del proyecto es el documento–herramienta más característico del enfoque del marco lógico. De hecho, el marco lógico comenzó siendo una matriz y hasta que la GTZ estableció la secuencia de pasos previos

que conducen hasta ella, lo que constituye la principal originalidad del método ZOPP, la identificación entre un término y otro fue absoluta. En la actualidad, aunque la mayoría de las versiones del método asumen esa secuencia como un todo, la matriz continúa siendo el elemento más típico del EML. (Camacho, Cámara, Cascante, & Sainz)

Como los autores indican y como se menciona anteriormente la matriz de marco lógico es el principal derivado de la metodología que contiene de manera resumida y practica los datos de un proyecto.

Para la construcción de esta matriz empezamos de derecha a izquierda generando la primera columna que es un resumen narrativo de los siguientes aspectos del proyecto **FIN, PROPÓSITO, COMPONENTES, ACTIVIDADES.**

Para tener una base de que se debe poner en cada celda tomamos como base la estructura analítica anteriormente generada y en cada segmento vertical nos brinda información para cada fila de la matriz. La siguiente figura nos muestra la relación de la estructura con la matriz:

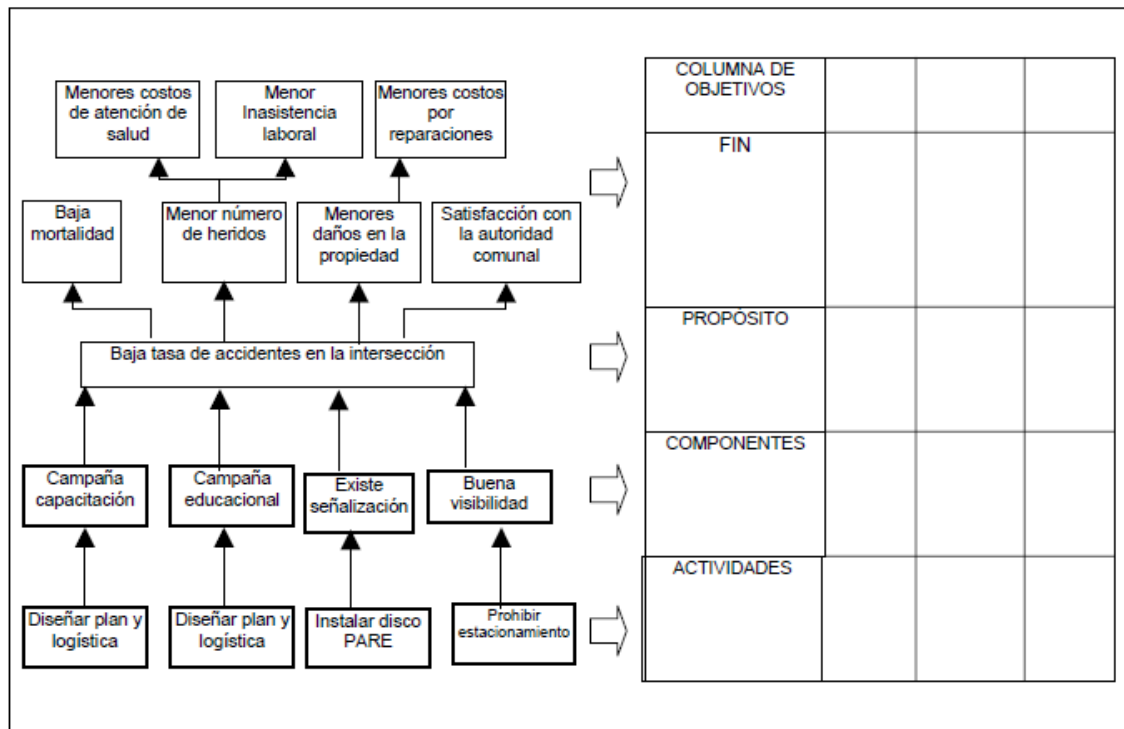


Figura 3.8 Relación de la estructura analítica del proyecto con la matriz de marco lógico.

Fuente: (Ortegón, Pacheco, & Prieto, 2005)

Para tener más claro se muestra otra figura con otra estructura analítica de un proyecto y la relación con la matriz:

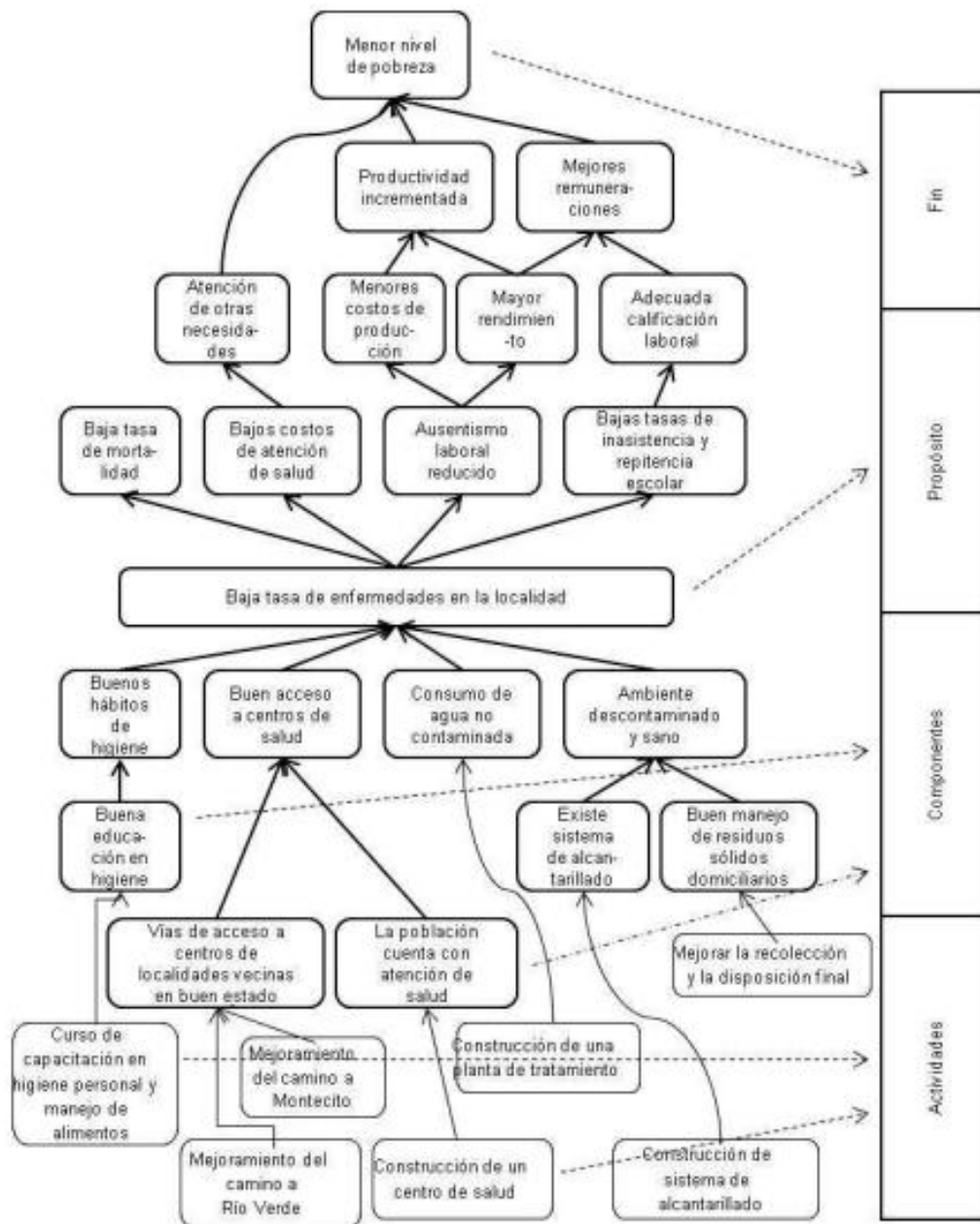


Figura 3.9 Ejemplo de la relación de la estructura analítica con la matriz de marco lógico

Fuente: (Aldunate & Córdoba, 2011)

Como muestran las figuras anteriores nos damos cuenta que en caso de fin pueden ser varios según los fines del árbol de objetivos, el propósito es solo uno y los componentes son los medios. Las actividades vienen del análisis de alternativas.

3.2.7 Indicadores

Se definen para el fin, propósito, componentes y actividades. Los indicadores aparecen en cada nivel de la matriz, se convierten en el punto de referencia y “carta de navegación” para guiar las actividades de gestión/monitoreo y evaluación del proyecto. Los indicadores bien formulados aseguran una buena gestión del proyecto y permiten que los gerentes de proyecto decidan si serán necesarios componentes adicionales o correcciones de rumbo para lograr el propósito del proyecto. (Sanchez, 2007)

Los indicadores como la autora nos indica son puntos de medición, cualitativos y cuantitativos que sirven para medir el desempeño y realización del proyecto mediante metas trazadas al inicio para tener una visión clara de lo que se desea conseguir en el proyecto.

Dentro de la matriz los indicadores ocupan la 2 columna y deberán ir según cada fila de la matriz identificando los indicadores que se desea cumplir en cada etapa.

Los indicadores se generaran en base a los siguientes aspectos mostrados en la siguiente figura:

Nivel	Resumen Narrativo	Indicador	Meta				Grupo Social
			Cantidad	Calidad	Tiempo	Lugar	
Fin							
Propósito							
Componentes							
Actividades							

Figura 3.10 Tabla para generación de índices.

Fuente: (Ortegón, Pacheco, & Prieto, 2005)

3.2.8 Medios de verificación

Se debe precisar los métodos y fuentes de recolección de información que permitirá evaluar y monitorear los indicadores. (Sanchez, 2007)

Los medios de verificación va ligados de los indicadores ya que estos nos permiten de manera tangible verificar que se cumplen los indicadores, son los elementos que permiten validar el cumplimiento de un indicador, por ejemplo el siguiente indicador:

- Reducir la tasa de accidentes e un 5 %.

Para poder comprobar que se dio este hecho necesitamos un medio por el cual se pueda comprobar esta reducción, que en este caso sería un censo de los accidentes después de la aplicación del proyecto.

3.2.9 Hipótesis o supuestos

Los supuestos en este caso son situaciones en las que no se podría dar la factibilidad del proyecto, dentro del proyecto sería los riesgos que se podría tener dentro de la ejecución. Estos van ubicados en la cuarta columna de la matriz de marco lógico y permiten saber las contras de la ejecución para determinar la viabilidad del proyecto.

Además su función es identificar las posibles causas de fracaso para tener muy en cuenta que es lo que el proyecto puede llegar a ser y que no.

Al igual que los indicadores los supuestos va por cada fila de la matriz y por cada ítem de las filas determinando supuestos por cada elemento.

Los supuestos se basan en aspectos:

- Financiero
- Político
- Social
- Ambiental
- Legal

3.2.10 Evaluación intermedia

El proceso de evaluación intermedia se centra en la verificación de los indicadores y el porcentaje de cumplimiento de los mismos, en este paso nosotros verificamos que indicadores se cumplieron y el porcentaje de completado de los demás mediante constantes evaluaciones al proyecto verificando mediante los medios anteriormente generados.

Este proceso se lo realiza iterativamente para verificar el estado del proyecto, verificando que no se suscite ningún supuesto y que el ciclo de vida del proyecto transcurra según lo esperado.

Para realizar este proceso simplemente se genera una tabla con el tiempo de verificación en la cabecera y el porcentaje de cumplimiento de cada indicador.

3.3 HERRAMIENTAS

En el mercado existen ya algunas herramientas para la gestión de proyectos, una de las más conocidas es Microsoft Project, que es un gestor de proyectos que nos permite generar cronogramas y relacionarlos con involucrados además de asignar tareas y verificar el cumplimiento de las mismas.

Estas herramientas solo sirven para cuando el usuario tiene clara ya la idea, es decir conoce completamente el problema central del proyecto y sabe que alternativa dará solución al mismo.

Dentro del medio existe una herramienta de escritorio que usa la metodología de marco lógico y es **TeamUP**. Esta herramienta de escritorio nos permite generar proyectos pero antes debemos ya tener cierto conocimiento sobre la metodología lo que la hace difícil de usar, además no hace uso de las últimas tecnologías.

Las herramientas para gestión de proyectos son muchas en el medio pero las que en verdad hacen uso de una metodología de desarrollo son escasas y poco conocidas.



CAPITULO IV



- 7. DESARROLLO
 - 7.1 PLANIFICACIÓN
 - 7.2 REQUERIMIENTOS
 - 7.3 DISEÑO
 - 7.4 DESARROLLO
 - 7.5 PRUEBAS

4 DESARROLLO

4.1 PLANIFICACIÓN

En la etapa de planificación se define los requerimientos iniciales del proyecto a elaborarse mediante las historias de usuario, mediante las cuales la metodología plantea los diferentes requerimientos del sistema.

Una vez las historias están listas se procede con la elaboración del plan de ejecución para lo cual se eligen las historias y su prioridad para ser realizadas.

4.1.1 Historias de usuario

El sistema consta con las siguientes historias de usuario definidas por las siguientes tablas:

- Control de usuarios

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 1	Usuario: Ing. Irving Reascos
Nombre historia: Control de usuarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Fecha: 30/03/2015	Iteración asignada:

Descripción:

En el sistema se registrara los usuarios que a su vez podrán tener varios proyectos de matriz de marco lógico, los usuarios se registraran mediante un formulario y se requerirá los siguientes datos:

- Correo del usuario
- Nombres y apellidos de usuario
- Contraseña
- Institución educativa

Crear una interfaz de acceso para los usuarios ya registrados.

Observaciones:

Validar los siguientes campos:

- Correo del usuario (correo válido)
- Contraseña (encriptar en la base de datos)

Tabla 4.1 Historia de Usuario 1 – Control de usuarios

Fuente: Propia

- Gestión de proyectos

Historia de Usuario

Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 2	Usuario: Ing. Irving Reascos
Nombre historia: Gestión proyectos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Fecha: 30/03/2015	Iteración asignada:
<p>Descripción:</p> <p>Generar una interfaz que permita a un usuario crear, editar, compartir, eliminar un proyecto usando la metodología de matriz de marco lógico.</p> <p>Un usuario puede compartir un proyecto a otros usuarios para trabajar en conjunto, así mismo puede quitar a personas anteriormente asignadas y cambiar sus permisos.</p> <p>También mostrar un historial de los cambios realizados en el proyecto</p>	
<p>Observaciones:</p> <p>Los usuarios podrán ver otros proyectos de marco lógico siempre y cuando sean asignados por el creador del proyecto.</p> <p>Para compartir un proyecto se lo hará mediante la dirección de correo</p> <p>Solo el usuario que crea el proyecto puede compartir el mismo.</p>	

La aplicación tendrá dos tipos de permisos al ver un proyecto de otro usuario:

- Lectura: solo permite ver los datos.
- Escritura: permite modificar los datos.

Tabla 4.2 Historia de Usuario 2 – Gestión de proyectos

Fuente: Propia

- Análisis de involucrados

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 3	Usuario: Ing. Irving Reascos
Nombre historia: Análisis de involucrados	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Fecha: 30/03/2015	Iteración asignada:
<p>Descripción:</p> <p>Generar una interfaz que permita ingresar, cambiar y eliminar los nombres de las personas o instituciones involucradas en el proyecto y asignarles un grado de</p>	

relación con el mismo, estas a su vez relacionadas con un proyecto

Observaciones:

No se puede proseguir al siguiente paso si este no es completado es decir si no se ingresa ningún involucrado no se puede continuar con el proceso de generación de proyecto.

Tabla 4.3 Historia de Usuario 3 – Análisis de involucrados

Fuente: Propia

- Análisis del problema

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 4	Usuario: Ing. Irving Reascos
Nombre historia: Análisis del problema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Fecha: 30/03/2015	Iteración asignada:

Descripción:

Generar una interfaz similar al organizador gráfico árbol de problemas que me permita gestionar un árbol de problemas para identificar el problema principal. Este estará relacionado a un proyecto.

La interfaz costa de tres partes:

- Causa: Se puede ingresar varias causas que a su vez pueden tener sub causas. Y cada una se relaciona con un efecto
- Problema: Solo existe un problema central.
- Efecto: Se puede ingresar varios efectos y estos a su vez pueden tener sub efectos. Cada uno relacionado con una causa

Observaciones:

Solo un árbol de problemas estará relacionado al proyecto.

Se debe haber culminado el paso anterior para poder estar en esta interfaz.

Todos los datos a ingresarse son texto, validar que cada causa tenga un efecto y viceversa.

Si no se realiza el proceso del árbol de problemas impedir que se pueda continuar al siguiente paso.

Tabla 4.4 Historia de Usuario 4 – Análisis del problema

Fuente: Propia

- Análisis de objetivos

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 5	Usuario: Ing. Irving Reascos
Nombre historia: Análisis de objetivos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Fecha: 30/03/2015	Iteración asignada:
<p>Descripción:</p> <p>Generar una interfaz similar al organizador gráfico de árbol de objetivos que permita gestionar un árbol de problemas relacionado aun proyecto.</p> <p>Este costa de 3 partes:</p> <ul style="list-style-type: none"> • Fin: Tiene relación con los efectos del árbol de problemas un fin tiene un efecto. • Objetivo principal: Es el objetivo principal del proyecto y tiene relación con el problema central del árbol de problemas • Medio: Tiene relación con las causas del árbol de problemas un medio tiene una causa. 	

Observaciones:

Solo un árbol de objetivos estará relacionado al proyecto.

Se debe haber culminado el paso anterior para poder estar en esta interfaz.

Validar que haya un árbol de problemas con datos.

Todos los datos a ingresarse son texto, validar las relaciones entre árbol de problemas y objetivos.

Tabla 4.5 Historia de Usuario 5 – Análisis de objetivos

Fuente: Propia

- Análisis de alternativas

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 6	Usuario: Ing. Irving Reascos
Nombre historia: Análisis de alternativas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio

Fecha: 30/03/2015	Iteración asignada:
<p>Descripción:</p> <p>Crear una interfaz que permita generar acciones por cada medio en el árbol de objetivos, clasificar por:</p> <ul style="list-style-type: none"> • Complementarias: Aquellas que no tienen ningún problema y pueden ser realizadas en conjunto con cualquier otra acción. • Excluyentes: Aquellas que tienen una o varias acciones que interfieren con estas y si esta se realiza las excluyentes a esta no se pueden realizar. <p>Generar las alternativas de manera dinámica, realizar todas las agrupaciones según su tipo, y generar alternativas las cuales una será la principal y la que se ejecutara en el proyecto. Esta será elegida por el usuario.</p>	
<p>Observaciones:</p> <p>No dejar acceder a la interfaz si el anterior paso no ha sido completado.</p> <p>Validar que existan medios en el árbol de objetivos.</p> <p>Una alternativa es el conjunto de acciones, estas a su vez tiene relación con un medio del árbol de problemas.</p> <p>Seleccionar solo una alternativa como la adecuada a tomarse en el proyecto</p>	

Tabla 4.6 Historia de Usuario 6 – Análisis de alternativas

Fuente: Propia

- Generación de la matriz de marco lógico

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 7	Usuario: Ing. Irving Reascos
Nombre historia: Generación de la matriz de marco lógico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Fecha: 30/03/2015	Iteración asignada:
<p>Descripción:</p> <p>Generar la matriz de marco lógico del proyecto, y llenar los datos de la primera columna en base a los pasos anteriores, análisis de involucrados, árbol de problemas, árbol de objetivos, análisis de alternativas.</p> <p>La relación de las filas de la primera columna con los datos anteriores son las siguientes :</p> <ul style="list-style-type: none"> • Fin: Son los fines del árbol de problemas • Propósito: Es el objetivo principal del árbol de objetivos. • Componentes: Son todos los medios del árbol de objetivos. 	

<ul style="list-style-type: none"> • Actividades: Son las acciones de la alternativa elegida para el proyecto.
<p>Observaciones:</p> <p>No se puede acceder si no se completó el paso anterior.</p> <p>Validar que existan datos en el árbol de objetivos, y se haga escogido una alternativa.</p> <p>Generar automáticamente los campos de propósito, componentes, actividades.</p> <p>Permitir la edición de los capos generados.</p>

Tabla 4.7 Historia de Usuario 7 – Generación de la matriz de marco lógico

Fuente: Propia

- Gestión de seguimiento del proyecto

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 8	Usuario: Ing. Irving Reascos
Nombre historia: Gestión de seguimiento del proyecto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media

Fecha: 30/03/2015

Iteración asignada:

Descripción:

Permitir agregar indicadores asociados a cada fila y sub-fila de la primera columna de la matriz de marco lógico.

Los indicadores son parámetros con valores cuantitativos y cualitativos de cada elemento al que se asocien. Como:

- Cantidad
- Calidad
- Tiempo
- Lugar
- Grupo social

Permitir asociar medios de verificación a cada indicador que se agregue, por cada indicador debe haber un medio por el cual se pueda verificar su cumplimiento.

Permitir agregar supuestos que son los riesgos de ejecución del proyecto.

Así como la evaluación de los mismos, agregando pruebas como enlaces de google drive droop box para verificar el cumplimiento en base a los medios de verificación ya descritos

Observaciones:

No permitir acceder si no se generó la matriz de marco lógico.

Cada indicador tiene un medio de verificación. Y ninguno o varios supuestos.

Tabla 4.8 Historia de Usuario 8 – Gestión de indicadores, medios de verificación y supuestos en la matriz de marco lógico

Fuente: Propia

- Generación de reportes y ayuda al usuario

Historia de Usuario	
Sistema de gestión de proyectos mediante matriz de marco lógico.	
Número: 9	Usuario: Ing. Irving Reascos
Nombre historia: Generación de reportes y ayuda al usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Fecha: 30/03/2015	Iteración asignada:
Descripción: Se requiere poder tener un documento en formato PDF, DOC. Dentro de esos se debe mostrar: <ul style="list-style-type: none"> • Matriz de marco lógico 	

<ul style="list-style-type: none"> • Proyecto con resúmenes narrativos • Estado de seguimiento <p>Además de los reporte se pide una interfaz que guie y brinde ayudas para cada paso del proceso de generación del proyecto.</p>
<p>Observaciones:</p> <p>El formato del documento puede ser PDF o DOC.</p> <p>Permitir al usuario realizar reportes cuantas veces se requiera.</p>

Tabla 4.9 Historia de Usuario 9 – Generación de reportes de la matriz de marco lógico

Fuente: Propia

4.1.2 Plan de iteraciones (Iteration plan)

Las tareas o iteraciones son las que se desprenden como las actividades a realizar en cada historia de usuario.

- Historia de Usuario 1 – Control de usuarios

Tarea
Historia: Control de usuarios

Número historia: 1	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Modelado de base de datos	Puntos estimados: 1
Descripción: <ul style="list-style-type: none"> • Diseñar el modelo de datos para la gestión de usuarios en el sistema, el modelo de datos será en JSON debido a la naturaleza de la base de datos MongoDB. 	

Tabla 4.10 Historia de Usuario 1, Tarea 1 – Diseño de modelo de datos

Fuente: Propia

Tarea	
Historia: Control de usuarios	
Número historia: 1	Numero tarea: 2
Nombre tarea: Definición de servicios REST para usuarios	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Fecha inicio: 30/03/2015	Fecha fin:

Descripción:

Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de gestión de usuarios.

Tabla 4.11 Historia de Usuario 1, Tarea 2 – Definición de servicios REST para usuarios

Fuente: Propia

Tarea	
Historia: Control de usuarios	
Número historia: 1	Numero tarea: 3
Nombre tarea: Formulario de registro de usuarios	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Fecha inicio: 30/03/2015	Fecha fin:
Descripción: Crear un formulario para ingresar usuarios en la base de datos con los siguientes parámetros: <ul style="list-style-type: none"> • Correo del usuario 	

- Nombre de usuario
- Contraseña
- Institución educativa

Tabla 4.12 Historia de Usuario 1, Tarea 3 – Formulario de registro de usuarios

Fuente: Propia

Tarea	
Historia: Control de usuarios	
Número historia: 1	Numero tarea: 4
Nombre tarea: Formulario para acceso de usuarios registrados	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
<p>Descripción:</p> <p>Crear un formulario para que los usuarios ya registrados accedan al sistema mediante su usuario o correo y su contraseña.</p> <p>En caso de no existir re direccionar al formulario de registro.</p>	

Tabla 4.13 Historia de Usuario 1, Tarea 4 – Formulario para acceso de usuarios registrados

Fuente: Propia

- Historia de Usuario 2 – Gestión proyectos

Tarea	
Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Diseño	Puntos estimados: 1
<p>Descripción:</p> <p>Diseñar el modelo de datos de un proyecto, un usuario puede tener varios proyectos y puede compartir los mismos con otros usuarios en modo lectura o escritura. También que almacene una revisión de cambios en el mismo.</p> <p>La relación usuario proyecto se la realizara mediante referencias, y el modelos será en JSON debido a la naturales de mongoDB</p>	

Tabla 4.14 Historia de Usuario 2, Tarea 1 – Diseño de modelo de datos

Fuente: Propia

Tarea	
Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 2
Nombre tarea: Definición de servicios REST para la gestión de proyectos	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Fecha inicio: 30/03/2015	Fecha fin:
Descripción: Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de gestión de proyectos.	

Tabla 4.15 Historia de Usuario 1, Tarea 2 – Definición de servicios REST para gestión de proyectos

Fuente: Propia

Tarea

Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 3
Nombre tarea: Mostrar los proyectos asociados al usuario	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
<p>Descripción:</p> <p>Crear una página donde se liste los proyectos creados por el usuario y los que hayan sido compartidos por otros usuarios al mismo. Su fecha de creación, nombre, estado y descripción de cada proyecto.</p>	

Tabla 4.16 Historia de Usuario 2, Tarea 3 – Mostrar los proyectos asociados al usuario

Fuente: Propia

Tarea	
Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 4
Nombre tarea: Crear proyectos	

Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Diseñar un formulario para crear nuevos proyectos, dentro de los proyectos serán necesarios los siguientes parámetros: <ul style="list-style-type: none"> • Nombre del proyecto • Descripción • Lista de usuarios a los que se desea compartir 	

Tabla 4.17 Historia de Usuario 2, Tarea 4 – Crear proyectos

Fuente: Propia

Tarea	
Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 5
Nombre tarea: Eliminar proyectos	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción:	

Permitir al usuario cambiar el estado de un proyecto a estado de eliminado.

También diseñar un formulario para ver los proyectos con estado eliminado y permitir eliminarlos de la BD en caso de que el usuario así lo requiera.

- Nombre del proyecto
- Descripción
- Lista de usuarios a los que se desea compartir

Tabla 4.18 Historia de Usuario 2, Tarea 5 – Eliminar proyectos

Fuente: Propia

Tarea	
Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 6
Nombre tarea: Compartir proyectos	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Diseñar un formulario para gestionar los usuarios a los cuales se comparte un	

proyecto.

Tabla 4.19 Historia de Usuario 2, Tarea 6 – Compartir proyectos

Fuente: Propia

Tarea	
Historia: Gestión proyectos	
Número historia: 2	Numero tarea: 7
Nombre tarea: Historial de cambios	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Mostrar los últimos cambios realizados sobre el proyecto así como quien los hizo y la fecha	

Tabla 4.20 Historia de Usuario 2, Tarea 7 – Historial de cambios

Fuente: Propia

- Historia de Usuario 3 – Análisis de involucrados

Tarea	
Historia: Análisis de involucrados	
Número historia: 3	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Diseño	Puntos estimados: 1
<p>Descripción:</p> <p>Diseñar el modelo de datos para ingresar varios involucrados en cada proyecto, este modelo ira embebido dentro del modelo de proyecto.</p> <p>El modelo será en formato JSON debido a la naturaleza de mongoDB</p>	

Tabla 4.21 Historia de Usuario 3, Tarea 1 – Diseño de modelo de datos

Fuente: Propia

Tarea	
Historia: Análisis de involucrados	
Número historia: 3	Numero tarea: 2
Nombre tarea: Definición de servicios REST para la gestión involucrados	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de gestión de involucrados de una matriz de marco lógico.	

Tabla 4.22 Historia de Usuario 3, Tarea 2 – Definición de servicios REST para la gestión involucrados

Fuente: Propia

Tarea	
Historia: Análisis de involucrados	
Número historia: 3	Numero tarea: 3

Nombre tarea: Gestión de involucrados	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Diseñar un formulario para poder agregar o eliminar involucrados a un proyecto. Un involucrado consta de nombre y grado de afinidad el proyecto. Si este formulario no es completado o no hay datos no permitir continuar al siguiente formulario de árbol de problemas	

Tabla 4.23 Historia de Usuario 3, Tarea 3 – Gestión de involucrados

Fuente: Propia

- Historia de Usuario 4 – Análisis del problema

Tarea	
Historia: Análisis del problema	
Número historia: 4	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Diseño	Puntos estimados: 1

Descripción:

Diseñar el modelo de datos de un árbol de problemas que consta de varias causas, varios efectos y un problema central. Un proyecto tiene un solo árbol de problemas, y estará de forma embebida dentro del mismo.

El modelo será en formato JSON debido a la naturaleza de mongoDB

Tabla 4.24 Historia de Usuario 4, Tarea 1 – Diseño de modelo de datos

Fuente: Propia

Tarea	
Historia: Análisis del problema	
Número historia: 4	Numero tarea: 2
Nombre tarea: Definición de servicios REST para la gestión de problemas	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de gestión de problemas de una matriz de marco lógico.	

Tabla 4.25 Historia de Usuario 4, Tarea 2 – Definición de servicios REST para la gestión de problemas

Fuente: Propia

Tarea	
Historia: Análisis del problema	
Número historia: 4	Numero tarea: 3
Nombre tarea: Gestión del árbol de problemas	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
<p>Descripción:</p> <p>Diseñar un formulario para generar un árbol de problemas, agregar o cambiar las diferentes partes del árbol de problemas.</p>	

Tabla 4.26 Historia de Usuario 4, Tarea 3 – Gestión del árbol de problemas

Fuente: Propia

- Historia de Usuario 5 – Análisis de objetivos

Tarea	
Historia: Análisis de objetivos	
Número historia: 5	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Diseño	Puntos estimados: 1
<p>Descripción:</p> <p>Diseñar el modelo de datos de un árbol de objetivos que consta de varios medios, varios fines y un objetivo central. Un proyecto tiene un solo árbol de objetivos, y estará de forma embebida dentro del mismo.</p> <p>El modelo será en formato JSON debido a la naturaleza de mongoDB</p>	

Tabla 4.27 Historia de Usuario 5, Tarea 1 – Diseño de modelo de datos

Fuente: Propia

Tarea
Historia: Análisis de objetivos

Número historia: 5	Numero tarea: 2
Nombre tarea: Definición de servicios REST para la gestión de objetivos	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de gestión de objetivos de una matriz de marco lógico.	

Tabla 4.28 Historia de Usuario 5, Tarea 2 – Definición de servicios REST para la gestión de objetivos

Fuente: Propia

Tarea	
Historia: Análisis de objetivos	
Número historia: 5	Numero tarea: 3
Nombre tarea: Gestión del árbol de objetivos	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1

Descripción:

Diseñar un formulario para generar un árbol de objetivos, agregar o cambiar las diferentes partes del árbol de objetivos.

Tabla 4.29 Historia de Usuario 5, Tarea 3 – Gestión del árbol de objetivos

Fuente: Propia

- Historia de Usuario 6 – Análisis de alternativas

Tarea	
Historia: Análisis de alternativas	
Número historia: 6	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Diseño	Puntos estimados: 1
Descripción: Diseñar el modelo de datos para as diferentes acciones en base a cada medio generado, y el modelo para almacenar un conjunto de acciones que será la alternativa del proyecto. El modelo será en formato JSON debido a la naturaleza de mongoDB	

Tabla 4.30 Historia de Usuario 6, Tarea 1 – Diseño de modelo de datos

Fuente: Propia

Tarea	
Historia: Análisis de alternativas	
Número historia: 6	Numero tarea: 2
Nombre tarea: Definición de servicios REST para la gestión de alternativas	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de gestión de alternativas de una matriz de marco lógico.	

Tabla 4.31 Historia de Usuario 6, Tarea 2 – Definición de servicios REST para la gestión de alternativas

Fuente: Propia

Tarea	
Historia: Análisis de alternativas	
Número historia: 6	Numero tarea: 3
Nombre tarea: Gestión de acciones	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Generar un formulario para agregar acciones por cada medio que existe dentro el árbol de objetivos.	

Tabla 4.32 Historia de Usuario 6, Tarea 3 – Gestión de acciones

Fuente: Propia

Tarea
Historia: Análisis de alternativas

Número historia: 6	Numero tarea: 4
Nombre tarea: Gestión de alternativas	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Generar las alternativas mediante las combinaciones de las acciones, y permitir al usuario editar las mismas y elegir cual es la alternativa que se va a usar en este proyecto.	

Tabla 4.33 Historia de Usuario 6, Tarea 4 – Gestión de alternativas

Fuente: Propia

- Historia de Usuario 7 – Generación de la matriz de marco lógico

Tarea	
Historia: Generación de la matriz de marco lógico	
Número historia: 7	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	

Tipo tarea: Diseño	Puntos estimados: 1
Descripción: Diseñar el modelo para un fin, un propósito, varios componentes y varias actividades asociadas al proyecto, estarán de manera embebida dentro del modelo del proyecto. El modelo será en formato JSON debido a la naturaleza de mongoDB	

Tabla 4.34 Historia de Usuario 7, Tarea 1 – Diseño del modelo de datos

Fuente: Propia

Tarea	
Historia: Generación de la matriz de marco lógico	
Número historia: 7	Numero tarea: 2
Nombre tarea: Definición de servicios REST para la matriz de marco lógico	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción:	

Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de una matriz de marco lógico.

Tabla 4.35 Historia de Usuario 7, Tarea 2 – Definición de servicios REST para la matriz de marco lógico

Fuente: Propia

Tarea	
Historia: Generación de la matriz de marco lógico	
Número historia: 7	Numero tarea: 3
Nombre tarea: Gestión de la primera columna de matriz de marco lógico	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
<p>Descripción:</p> <p>Generar y mostrar el propósito, los componentes y actividades en base a los pasos anteriores, y pedir que se ingrese el fin. Se podrá editar todos los campos antes mencionados a conveniencia del usuario.</p>	

Tabla 4.36 Historia de Usuario 7, Tarea 3 – Gestión de la primera columna de matriz de marco lógico

Fuente: Propia

- Historia de Usuario 8 – Gestión de seguimiento del proyecto

Tarea	
Historia: Gestión de seguimiento del proyecto	
Número historia: 8	Numero tarea: 1
Nombre tarea: Diseño de modelo de datos	
Tipo tarea: Diseño	Puntos estimados: 1
Descripción: Diseñar el modelo para ingresar indicadores, medios de verificación por cada indicador y supuestos, embebidos dentro del modelo de la matriz de marco lógico. El modelo será en formato JSON debido a la naturaleza de mongoDB	

Tabla 4.37 Historia de Usuario 8, Tarea 1 – Diseño del modelo de datos

Fuente: Propia

Tarea

Historia: Gestión de seguimiento del proyecto	
Número historia: 8	Numero tarea: 2
Nombre tarea: Definición de servicios REST para seguimiento de la matriz de marco lógico	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Definición las rutas y generar los servicios REST en el lado del servidor para el acceso de la interfaz de indicadores, medios de verificación supuestos.	

Tabla 4.38 Historia de Usuario 8, Tarea 2 – Definición de servicios REST para seguimiento de la matriz de marco lógico

Fuente: Propia

Tarea	
Historia: Gestión de seguimiento del proyecto	
Número historia: 8	Numero tarea: 3

Nombre tarea: Gestión de indicadores	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Diseñar una interfaz que me permita añadir eliminar indicadores de cada elemento de la matriz de marco lógico	

Tabla 4.39 Historia de Usuario 8, Tarea 3 – Gestión de indicadores

Fuente: Propia

Tarea	
Historia: Gestión de seguimiento del proyecto	
Número historia: 8	Numero tarea: 4
Nombre tarea: Gestión de medios de verificación	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Diseñar una interfaz que me permita añadir eliminar medios de verificación por cada indicador generado en el proyecto.	

Tabla 4.40 Historia de Usuario 8, Tarea 4 – Gestión de medios de verificación

Fuente: Propia

Tarea	
Historia: Gestión de seguimiento del proyecto	
Número historia: 8	Numero tarea: 5
Nombre tarea: Gestión de supuestos	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Diseñar una interfaz que me permita añadir eliminar supuestos en el proyecto.	

Tabla 4.41 Historia de Usuario 8, Tarea 5 – Gestión de supuestos

Fuente: Propia

- Historia de Usuario 9 – Generación de reportes y ayuda al usuario

Tarea

Historia: Generación de reportes y ayuda al usuario	
Número historia: 9	Numero tarea: 1
Nombre tarea: Generar reportes	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
<p>Descripción:</p> <p>Diseñara un formato para los reportes de marco lógico en formatos de pdf, docx.</p> <p>Generar reportes de:</p> <ul style="list-style-type: none"> • Matriz de marco lógico • Avance del proyecto • Generar esquena del proyecto. 	

Tabla 4.42 Historia de Usuario 8, Tarea 4 – Gestión de supuestos

Fuente: Propia

Tarea	
Historia: Generación de reportes y ayuda al usuario	
Número historia: 9	Numero tarea: 2

RELESE PLAN

Nombre tarea: Ayuda al usuario	
Tipo tarea: Diseño y desarrollo	Puntos estimados: 1
Descripción: Generar mensajes de ayuda en cada interfaz del aplicativo, brindando mensajes de ayuda o guías para cada paso del proceso de generación del proyecto	

Tabla 4.43 Historia de Usuario 8, Tarea 4 – Gestión de supuestos

Fuente: Propia

4.1.3 Plan de despliegue (Relese plan)

Se define el cronograma de ejecución de las tareas en iteraciones, cada iteración cuenta como una semana con un total de 8 para la ejecución y finalización de todas las tareas.

Historia de usuario / Tarea		MES 1				MES 2			
		I1	I2	I3	I4	I5	I6	I7	I8
1	Control de usuarios								
1.1	Diseño de modelo de datos	■							
1.2	Definición de servicios REST para usuarios		■						
1.3	Formulario de registro de usuarios					■			
1.4	Formulario para acceso de usuarios registrados					■			
2	Gestión proyectos								
2.1	Diseño de modelo de datos	■							
2.2	Definición de servicios REST para la gestión de proyectos		■						
2.3	Mostrar los proyectos asociados al usuario					■			
2.4	Crear proyectos					■			
2.5	Eliminar proyectos					■			
2.6	Compartir proyectos					■			
2.7	Historial de cambios					■			
3	Análisis de involucrados								
3.1	Diseño de modelo de datos	■							
3.2	Definición de servicios REST para la gestión involucrados			■					

- Tarea 1 Diseño de modelo de datos

Historia de usuario 2 Gestión proyectos:

- Tarea 1 Diseño de modelo de datos

Historia de usuario 3 Análisis de involucrados:

- Tarea 1 Diseño de modelo de datos

Historia de usuario 4 Análisis del problema:

- Tarea 1 Diseño de modelo de datos

Historia de usuario 5 Análisis de objetivos:

- Tarea 1 Diseño de modelo de datos

Historia de usuario 6 Análisis de alternativas:

- Tarea 1 Diseño de modelo de datos

Historia de usuario 7 Generación de la matriz de marco lógico:

- Tarea 1 Diseño de modelo de datos

Historia de usuario 8 Gestión de seguimiento en la matriz de marco lógico:

- Tarea 1 Diseño de modelo de datos

- **Iteración 2**

Historia de usuario 1 Control de usuarios:

- Tarea 2 Definición de servicios REST para usuarios

Historia de usuario 2 Gestión proyectos:

- Tarea 2 Definición de servicios REST para la gestión de proyectos

- **Iteración 3**

Historia de usuario 3 Análisis de involucrados:

- Tarea 2 Definición de servicios REST para la gestión involucrados

Historia de usuario 4 Análisis del problema:

- Tarea 2 Definición de servicios REST para la gestión de problemas

Historia de usuario 5 Análisis de objetivos:

- Tarea 2 Definición de servicios REST para la gestión de objetivos

Historia de usuario 6 Análisis de alternativas:

- Tarea 2 Definición de servicios REST para la gestión de alternativas

- **Iteración 4**

Historia de usuario 7 Generación de la matriz de marco lógico:

- Tarea 2 Definición de servicios REST para la matriz de marco lógico

Historia de usuario 8 Gestión de seguimiento en la matriz de marco lógico:

- Tarea 2 Definición de servicios REST para seguimiento

- **Iteración 5**

Historia de usuario 1 Control de usuarios:

- Tarea 3 Formulario de registro de usuarios
- Tarea 4 Formulario para acceso de usuarios registrados

Historia de usuario 2 Gestión proyectos:

- Tarea 3 Mostrar los proyectos asociados al usuario
- Tarea 4 Crear proyectos
- Tarea 5 Eliminar proyectos
- Tarea 6 Compartir proyectos
- Tarea 7 Historial de cambios

- **Iteración 6**

Historia de usuario 3 Análisis de involucrados:

- Tarea 3 Gestión de involucrados

Historia de usuario 4 Análisis del problema:

- Tarea 3 Gestión del árbol de problemas

Historia de usuario 5 Análisis de objetivos:

- Tarea 3 Gestión del árbol de objetivos

Historia de usuario 6 Análisis de alternativas:

- Tarea 3 Gestión de acciones
- Tarea 4 Gestión de alternativas

- **Iteración 7**

Historia de usuario 7 Generación de la matriz de marco lógico:

- Tarea 3 Gestión de la primera columna de matriz de marco lógico

Historia de usuario 8 Gestión de seguimiento en la matriz de marco lógico:

- Tarea 3 Gestión de indicadores
- Tarea 4 Gestión de medios de verificación
- Tarea 5 Gestión de supuestos

- **Iteración 8**

Historia de usuario 9 Generación de reportes de la matriz de marco lógico:

- Tarea 1 Generar reporte
- Tarea 2 Ayuda al usuario

4.2 ANÁLISIS DE REQUERIMIENTOS

Como se definió en las historias de usuario, ahora mediante la ingeniería de requisitos se clasificara los requerimientos del sistema a desarrollarse en base a las historias de usuario obtenidas.

Mediante el análisis de las historias de usuario obtenidas se puede obtener los requerimientos y clasificarlos en funcionales y no funcionales para mejor ejecución y desarrollo del sistema.

4.2.1 Requerimientos funcionales

- Gestionar el ingreso de usuarios mediante su correo
- Permitir recuperación de contraseñas mediante envío de correo electrónico y comprobando los nombres y apellidos
- Gestionar el ingreso, modificación, eliminación de proyectos de matriz de marco lógico.
- Permitir compartir en edición y no edición los proyectos creados a los diferentes usuarios que estén registrados en el sistema
- Controlar que se cumpla el proceso de la matriz de marco lógico a lo largo de la generación del proyecto
- Gestionar el seguimiento de los proyectos generados mediante evaluaciones intermedias a los indicadores.
- Generar reportes sobre la matriz de marco lógico, el proyecto y el seguimiento.
- Aplicar normas básicas de responsividad web para las interfaces
- Generar ayudas para guiar al usuario en el proceso de generación del proyecto mediante la matriz de marco lógico
- Permitir asociar URL de documentos para validar las evaluaciones intermedias

4.2.2 Requerimientos no funcionales

- El sistema debe ser accesible mediante internet
- Hacer uso de herramientas de software libre

- Almacenamiento de datos en base a nuevas tecnologías NOSQL
- Hacer uso de una arquitectura PASS

4.3 DISEÑO

4.3.1 Iteración 1

En esta iteración se pretende definir el modelo de datos de la aplicación partiendo de un modelo tradicional de entidades a un modelo acorde a MongoDB que se un modelo de objetos JSON.

- **Diagrama de entidades**

En la siguiente imagen se muestra el diagrama de imagen del modelo lógico del aplicativo, hay que tomar en cuenta que es un modelo simplemente de guía para tener de una forma más clara y visual el modelo de datos del sistema, esto por la naturaleza de mongo que no trabaja con tablas al igual que las bases de datos tradicionales.

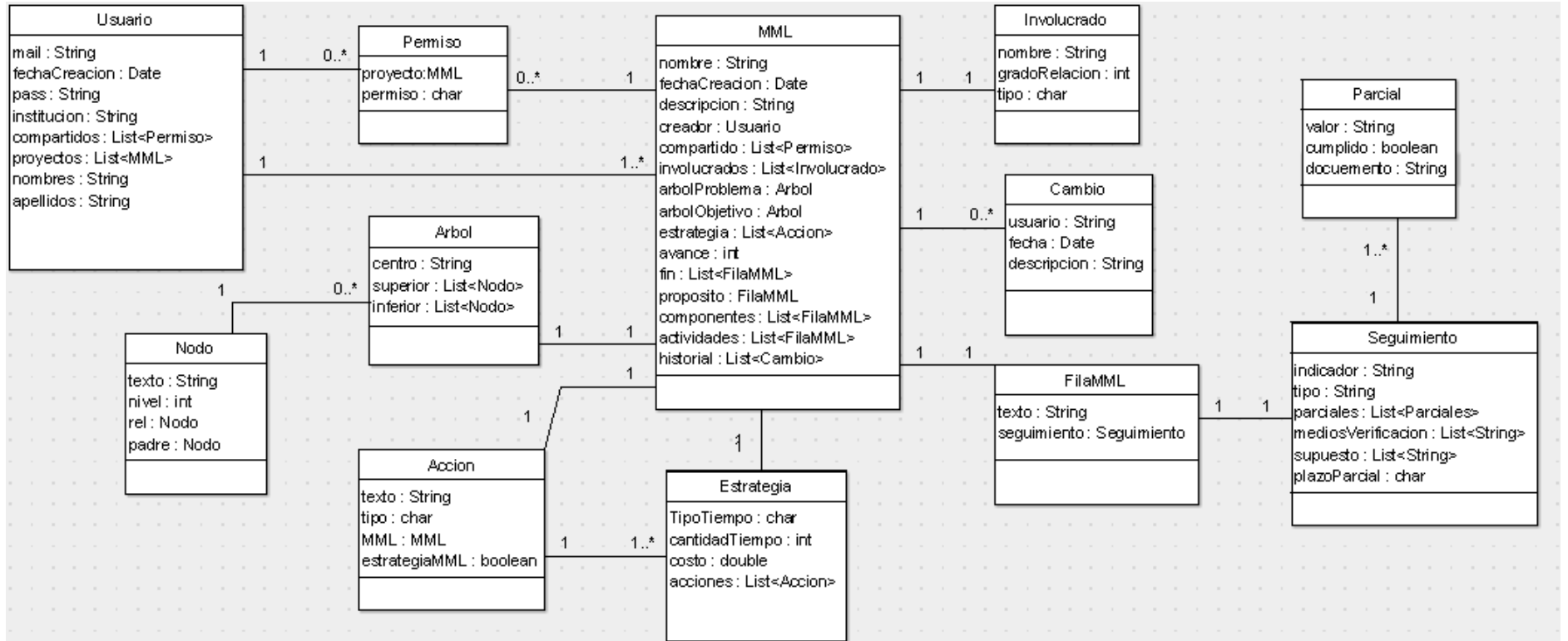


Figura 4.1 Diagrama de entidades

- **Modelos de datos MongoDB**

En base al diagrama de entidades de la **Figura 4.1** se generaran los modelos de datos o colecciones de MongoDB, tomando en cuenta la capacidad de embeber entidades y tratando siempre en lo posible de tener un modelo de datos más eficaz posible, ciertas entidades, gracias a la arquitectura de MongoDB, están embebidas dentro de otras. A continuación se define las principales colecciones del modelo:

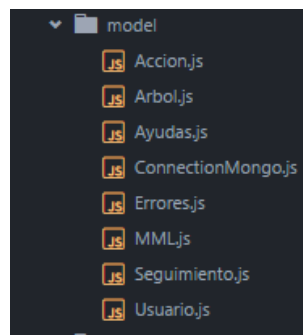


Figura 4.2 Sistema de archivos del modelo de datos

Fuente: Propia

4.3.2 Iteración 2

En la iteración dos se definirán las rutas del lado del cliente de los servicios **REST** su nomenclatura y métodos de acceso de las historias de usuario 1 y 2.

Los servicios **REST** serán por cada colección definida anteriormente, y las rutas en el lado del usuario por cada proceso definido en las tareas.

A continuación se definen las rutas de los servicios en el lado del servidor:

- **Servicios REST de usuario**

Ruta	Métodos	Datos	Sesión
APIMML/usuario/register	POST	{usuario}	{}
APIMML/usuario/update	PUT	{usuario}	{userID}
APIMML/usuario/log	POST	{mail, pass}	{}
APIMML/usuario/logout	GET	{}	{}
APIMML/usuario/verificar	POST	{mail}	{userID}
APIMML/usuario/	GET	{}	{userID}

Tabla 4.45 Servicios REST de usuario

Fuente: Propia

- **Servicios REST de MML**

Ruta	Métodos	Datos	Sesión
APIMML/mml/new	POST	{mml}	{userID}
APIMML/mml/:id/update	PUT	{mml}	{userID}
APIMML/mml/:id	GET	{}	{userID}
APIMML/mml/	GET	{}	{userID}
APIMML/mml/:id/share	POST	{userID}	{userID}
APIMML/mml/:id/delete	DELETE	{mml}	{userID}
APIMML/mml/:id/activate	POST	{mml}	{userID}

Tabla 4.46 Servicios REST de MML

Fuente: Propia

A continuación se definen las rutas en el lado del cliente que son las que se verán directamente en la barra de direcciones:

- **Rutas de usuario**

Ruta	Descripción
/#/login	Página de acceso al sistema
/#/register	Página de registro del sistema
/#/profile	Página que muestra la información del usuario
/#/about	Página que muestra la información acerca del aplicativo

Tabla 4.47 Rutas de usuario

Fuente: Propia

- **Rutas de gestión de proyectos**

Ruta	Descripción
/#/mml	Página que muestra los proyectos asociados, tanto como los compartidos.
/#/mml/:id	Página que muestra un proyecto según el id y su

estado.

Tabla 4.48 Rutas de usuario

Fuente: Propia

4.3.3 Iteración 3

En la iteración dos se definirán las rutas del lado del cliente de los servicios **REST** su nomenclatura y métodos de acceso de las historias de usuario 1 y 2.

Los servicios **REST** serán por cada colección definida anteriormente, y las rutas en el lado del usuario por cada proceso definido en las tareas.

A continuación se definen las rutas de los servicios en el lado del servidor:

- **Servicios REST de árbol**

Ruta	Métodos	Datos	Sesión
APIMML/arbol	POST	{idMML}	{userID}
APIMML/arbol/:id	PUT	{idMML, arbol}	{userID}
APIMML/arbol/:id	GET	{idMML}	{userID}

Tabla 4.49 Rutas de usuario

Fuente: Propia

A continuación se definen las rutas en el lado del cliente que son las que se verán directamente en la barra de direcciones:

- **Rutas de gestión involucrados**

Ruta	Descripción
/#/mml/:id/involucrados	Página que gestiona los involucrados de un proyecto según su id

Tabla 4.50 Rutas de gestión de involucrados

Fuente: Propia

- **Rutas de gestión del problema**

Ruta	Descripción
/#/mml/:id/AProblemas	Página que gestiona el árbol de problemas de un proyecto según su id

Tabla 4.51 Rutas de gestión del problema

Fuente: Propia

- **Rutas de gestión de objetivos**

Ruta	Descripción
/#/mml/:id/AObjetivos	Página que gestiona el árbol de problemas de un proyecto según su id

Tabla 4.52 Rutas de gestión de objetivos

Fuente: Propia

4.3.4 Iteración 4

En la iteración dos se definirán las rutas del lado del cliente de los servicios **REST** su nomenclatura y métodos de acceso de las historias de usuario 1 y 2.

Los servicios **REST** serán por cada colección definida anteriormente, y las rutas en el lado del usuario por cada proceso definido en las tareas.

A continuación se definen las rutas de los servicios en el lado del servidor:

- **Servicios REST de acción**

Ruta	Métodos	Datos	Sesión
------	---------	-------	--------

APIMML/accion	POST	{accion}	{userID}
APIMML/accion	GET	{idMML}	{userID}
APIMML/acción/:id	GET	{idAccion}	{userID}
APIMML/accion/:id	PUT	{idAccion}	{userID}
APIMML/acción/:id	DELETE	{idAccion}	{userID}

Tabla 4.53 Rutas de usuario

Fuente: Propia

- **Servicios REST de seguimiento**

Ruta	Métodos	Datos	Sesión
APIMML/seguimiento	POST	{idMML, seguimiento}	{userID}
APIMML/seguimiento	GET	{idMML}	{userID}
APIMML/seguimiento/:id	PUT	{idSeg}	{userID}

APIMML/seguimiento/:id DELETE {idSeg} {userID}

Tabla 4.54 Rutas de gestión de involucrados

Fuente: Propia

A continuación se definen las rutas en el lado del cliente que son las que se verán directamente en la barra de direcciones:

- **Rutas de gestión de alternativas**

Ruta	Descripción
/#/mml/:id/Estrategia	Página que muestra la estrategia del proyecto por su id

Tabla 4.55 Rutas de gestión de alternativas

Fuente: Propia

- **Rutas de generación de la matriz**

Ruta	Descripción
/#/mml/:id/proyecto	Página que muestra para edición el resultado del proceso de generación para ser guardado o modificado.

Tabla 4.56 Rutas de generación de la matriz

Fuente: Propia

- **Rutas de seguimiento**

Ruta	Descripción
/#/mml/:id/indicadores	Muestra todos los indicadores de seguimiento de un proyecto
/#/mml/:id/seguimiento	Permite agregar evidencias de seguimiento y avance del proyecto verificando el estado de los indicadores.

Tabla 4.57 Rutas de gestión del problema

Fuente: Propia

4.3.5 Iteración 5

En la iteración 5 se procede a la elaboración de los diseños de interfaces de para la gestión y manejo de usuarios y proyectos, como se indica en las historias de usuario 1 y 2.

- **Mockups gestión usuarios**

En la siguiente imagen se ve el formulario de acceso al sistema, esta es la primera pantalla que mostrara el sistema.

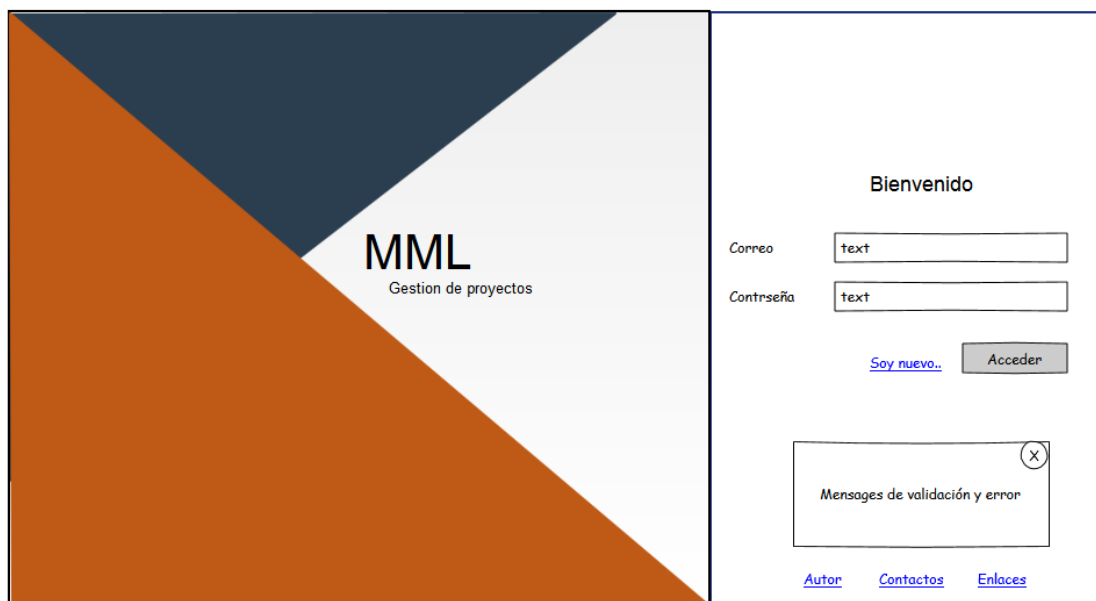


Figura 4.3 Formulario acceso de usuarios

Fuente: Propia

En la siguiente imagen podemos ver el diseño del formulario de registro de nuevos usuarios al sistema.

MML
Gestion de proyectos

Registro de usuarios

Correo

Datos Nombres Apellidos

Institución

Contraseña

Confirmar

CAPTCHA

[Ya estoy registrado](#)

Messages de validación y error

[Autor](#) [Contactos](#) [Enlaces](#)

Figura 4.4 Formulario registro de usuarios

Fuente: Propia

En el siguiente diseño se muestra el perfil del usuario, sus datos y la gestión de los mismos, permitiéndole gestionar sus datos personales y sus claves de acceso.

The image shows a user profile form layout within a blue-bordered container. At the top is an orange navigation bar labeled "Barra de navegacion". The form is divided into two main sections:

- Cuenta:** Contains a "Correo" field with the value "Correo@asdasdas.com", a "Password" field with a blue link "Cambiar password", and a sub-section for password changes with "Anterior" and "Nuevo" fields (both masked with asterisks) and a "Confirmar" button.
- Datos:** Contains three text input fields labeled "Nombres", "Apellidos", and "Direccion", each with a "text" placeholder, and a "Guardar cambios" button at the bottom.

Figura 4.5 Formulario perfil de usuario

Fuente: Propia

- **Mockups gestión proyectos**

El siguiente diseño nos muestra la pantalla principal de gestión de proyectos en la cual se podrá ver todos los proyectos generados, así como los que sean compartido a nosotros y se podrá gestionar los mismos.

MML Usuario Salir Gestión proyectos

Titulo
Fecha Nuevo

Descripcion

Porcentaje y estado

Eliminar Ver Colaboradores

Mis proyectos

MenuItem	Fecha	0%
MenuItem		
MenuItem		
MenuItem		

Proyectos compartidos

MenuItem	Fecha	0%
MenuItem		
MenuItem		
MenuItem		

Papeleria

MenuItem	Fecha	0%
MenuItem		
MenuItem		
MenuItem		

Mensajes del sistema X

Figura 4.6 Formulario gestión de proyectos

Fuente: Propia

En la siguiente imagen se ve el diseño de la ventana modal que permite confirmar la eliminación de un proyecto.



Figura 4.7 Formulario eliminar proyecto

Fuente: Propia

En la siguiente imagen vemos el diseño previo de la ventana modal que permitirá gestionar los colaboradores de un proyecto.

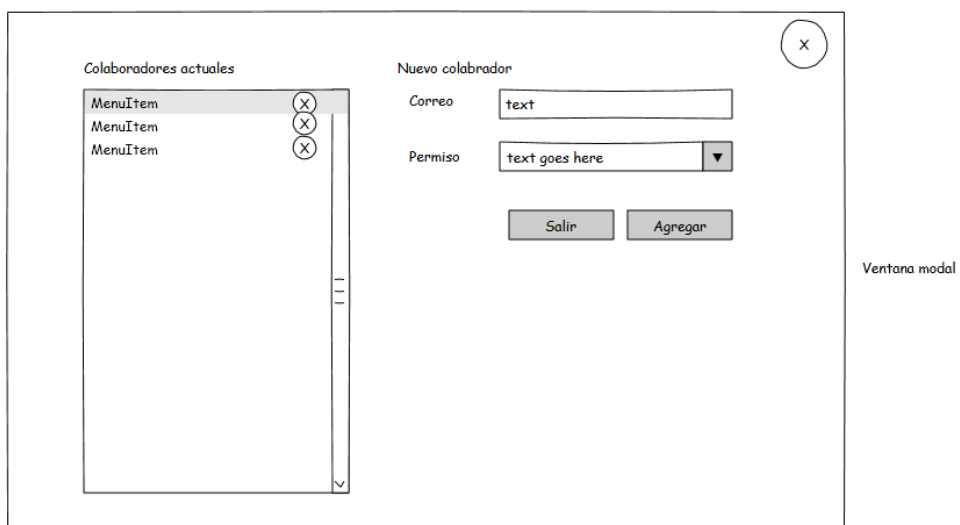


Figura 4.8 Formulario compartir proyecto

Fuente: Propia

El diseño que se muestra a continuación muestra al interfaz que gestionara un proyecto del usuario, mostrando los pasos a resolver para la generación del mismo, su avance e historial de cambios.

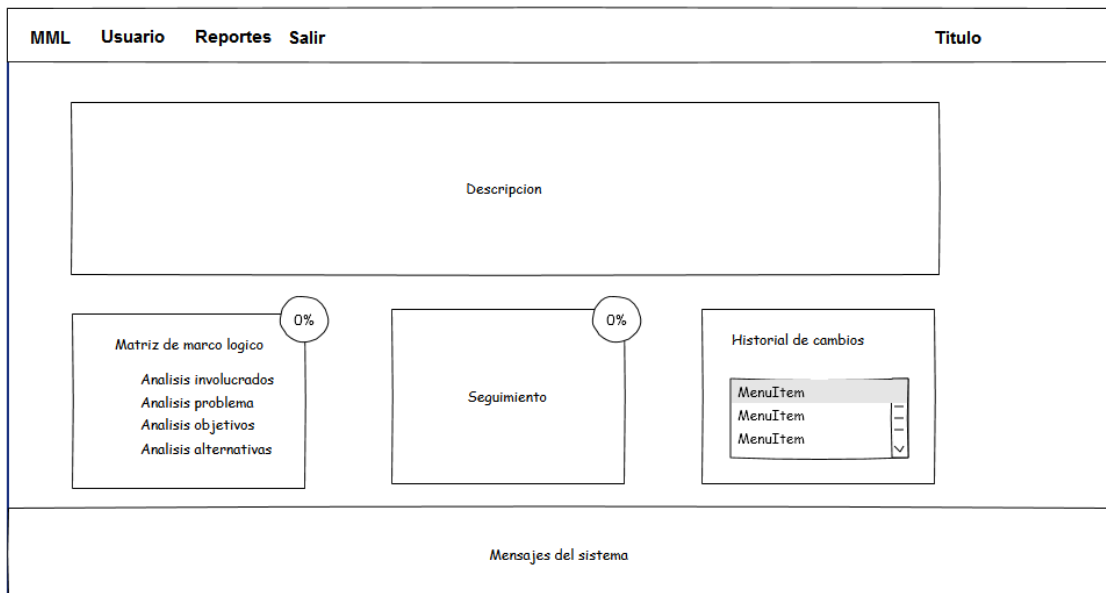


Figura 4.9 Formulario ver proyecto

Fuente: Propia

4.3.6 Iteración 6

En la siguiente iteración se trabajara en las historias de usuario 3, 4, 5 y 6 que son parte de la elaboración inicial del proyecto. Ac continuación se muestran los diseños previos de los formularios y páginas de cada historia.

- **Mockups análisis involucrados**

En la siguiente imagen se muestra el diseño de la interfaz que gestionar a los involucrados del proyecto.

Involucrados

En esta primera etapa se identificaran a las persona o instituciones relacionadas al proyecto, esas pueden estar a favor o en contra en base a un grado de relación.

Nuevo

Nombre

Tipo

Finalizar

Crear

Nro	Involucrado	Tipo	Grado	Acciones
1	n1	Persona	10	editar elim
1	n1	Persona	10	editar elim
1	n1	Persona	10	editar elim

Figura 4.10 Formulario de gestión de involucrados

Fuente: Propia

En el siguiente diseño se ve la ventana modal que permitirá agregar nuevos involucrados al proyecto.

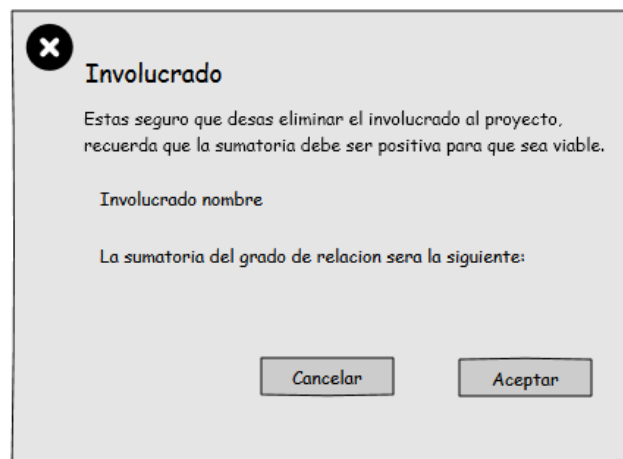


Un formulario de edición con el título "Involucrado" y un ícono de cerrar (X) en la esquina superior izquierda. El formulario contiene un campo de texto etiquetado "Nombre", un menú desplegable etiquetado "Tipo", un control deslizante horizontal con un círculo en el centro, y dos botones al fondo: "Cancelar" a la izquierda y "Editar" a la derecha.

Figura 4.11 Formulario editar involucrado

Fuente: Propia

En la imagen que se muestra a continuación se ve la ventana de confirmación para poder eliminar involucrados del proyecto.



Una ventana de confirmación con el título "Involucrado" y un ícono de cerrar (X) en la esquina superior izquierda. El texto principal dice: "Estas seguro que desas eliminar el involucrado al proyecto, recuerda que la sumatoria debe ser positiva para que sea viable." Debajo de esto, se muestra "Involucrado nombre" y "La sumatoria del grado de relacion sera la siguiente:". Al fondo hay dos botones: "Cancelar" a la izquierda y "Aceptar" a la derecha.

Figura 4.12 Formulario eliminar involucrado

Fuente: Propia

- **Mockups análisis del problema**

La siguiente imagen muestra el diseño de la interfaz que se encargara de la gestión del árbol de problemas.

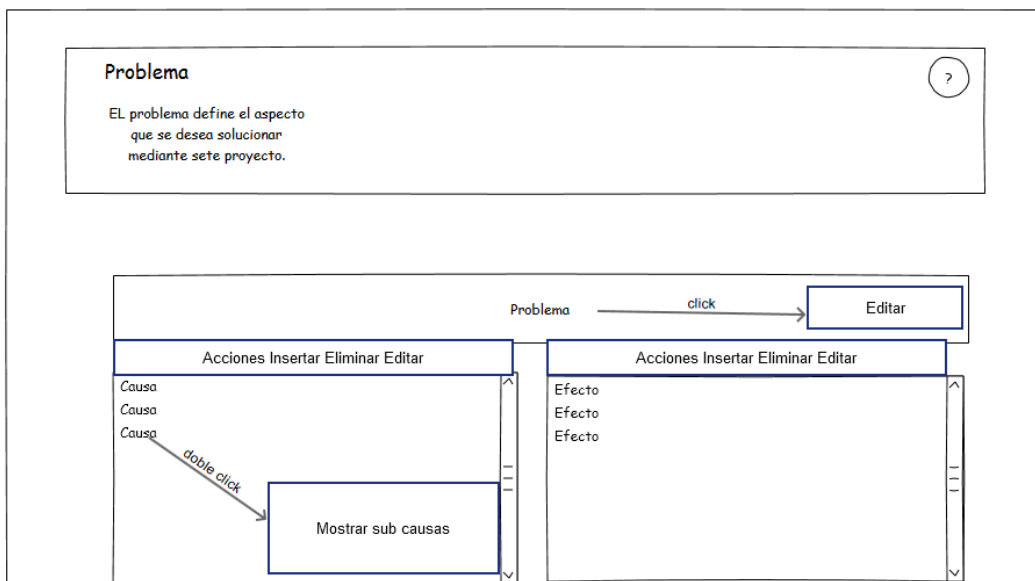


Figura 4.13 Formulario gestión del problema

Fuente: Propia

En la siguiente imagen se ve el diseño de la ventana modal que me permitirá agregar nodos de causa – efecto al proyecto.

El formulario muestra dos columnas: 'Causa' y 'Efecto'. En la columna 'Causa', hay un campo de texto con el valor 'Causa' y un menú desplegable 'Padre' con una flecha hacia abajo. En la columna 'Efecto', hay un campo de texto con el valor 'Efecto' y un campo de texto 'Padre(No editable)'. Una línea con una flecha apunta desde el campo 'Causa' hacia el campo 'Efecto'. En la parte inferior del formulario, hay dos botones: 'Cancelar' y 'Aceptar'. En la esquina superior izquierda hay un botón de cerrar (X) y en la superior derecha un botón de ayuda (?).

Figura 4.14 Formulario ver edición e inserción de una causa-efecto

Fuente: Propia

- **Mockups análisis de objetivos**

En la imagen que a continuación se muestra se ve el diseño de la interfaz que permitirá la gestión del árbol de objetivos.

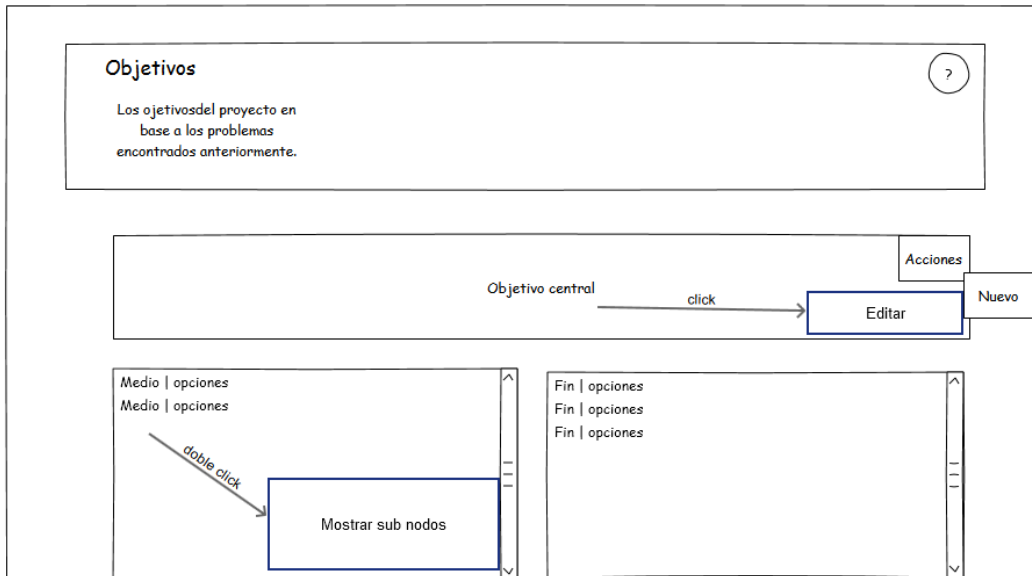


Figura 4.15 Formulario gestión de objetivos

Fuente: Propia

En el diseño que a continuación se muestra se ve la ventana modal que permite agregar nodos al árbol e objetivos.

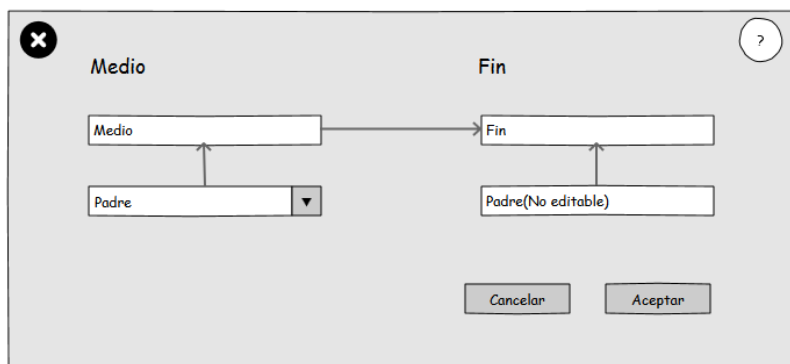


Figura 4.16 Formulario ver edición e inserción de medio-fin

Fuente: Propia

- **Mockups análisis de alternativas**

A continuación se ve el diseño de la interfaz que permite la gestión de alternativas del proyecto mostrando las diferentes estrategias generadas.

Gestion de alternativas

En esta etapa final se elegira un conjunto de acciones que representan la estrategia optima que solucionara el proyecto.

Estrategias sugeridas

Nro	NumAcciones	Eliminar	Principal	Costo Est	TiempoEst
Nro	NumAcciones	Eliminar	<input checked="" type="radio"/>		
Nro	NumAcciones	Eliminar	<input type="radio"/>		

Nro	Accion	Opciones
1	Accion asd asd as da sd	+ -
2	Accion asd asd as da sd	+ -
3	Accion asd asd as da sd	+ -

Opciones

Principal

Agregar accion

Figura 4.17 Formulario de gestión de alternativas

Fuente: Propia

En la siguiente imagen se ve la ventana modal que permitirá la generación de las acciones en base a cada medio del árbol de objetivos.

El formulario muestra los siguientes campos:

- Causa:** Campo de texto con el valor "text".
- Medio:** Campo de texto con el valor "text".
- Acción:** Campo de texto con el valor "text".
- Excluyente con:** Menú desplegable.
- Tiempo estimado:** Campo de texto con el valor "text".
- Día | mes:** Menú desplegable.
- Costo estimado:** Campo de texto con el valor "text".
- Siguiente:** Botón de acción.

Figura 4.18 Formulario de generación de acciones por cada medio

Fuente: Propia

4.3.7 Iteración 7

En la siguiente iteración se trabajara en las historias de usuario 7 y 8 que son parte de la generación del proyecto y el seguimiento del mismo.

- **Mockups generación del proyecto (primera columna de la matriz)**

En la imagen a continuación se muestra la interfaz para la generación del proyecto, mostrando la primera fila de la matriz y permitiendo la gestión de indicadores.

Matriz de marco logico Siguiete

Fin * FIN 1 * FIN 2 * FIN 3	<input checked="" type="checkbox"/> Seguir	Muestra los indicadores, medios de verificación, supuestos	
Proposito Proposito definido	<input checked="" type="checkbox"/> Seguir		
Componetnes * Componente 1 * Componente 2 * Componente 3	<input checked="" type="checkbox"/> Seguir		
Actividades * Actividad 1 * Actividad 2 * Actividad 3	<input checked="" type="checkbox"/> Seguir		

Figura 4.19 Primera columna de la matriz.

Fuente: Propia

En la siguiente imagen muestra las ventanas modales para la gestión de indicadores por cada elemento de la primera fila de la matriz de marco lógico.

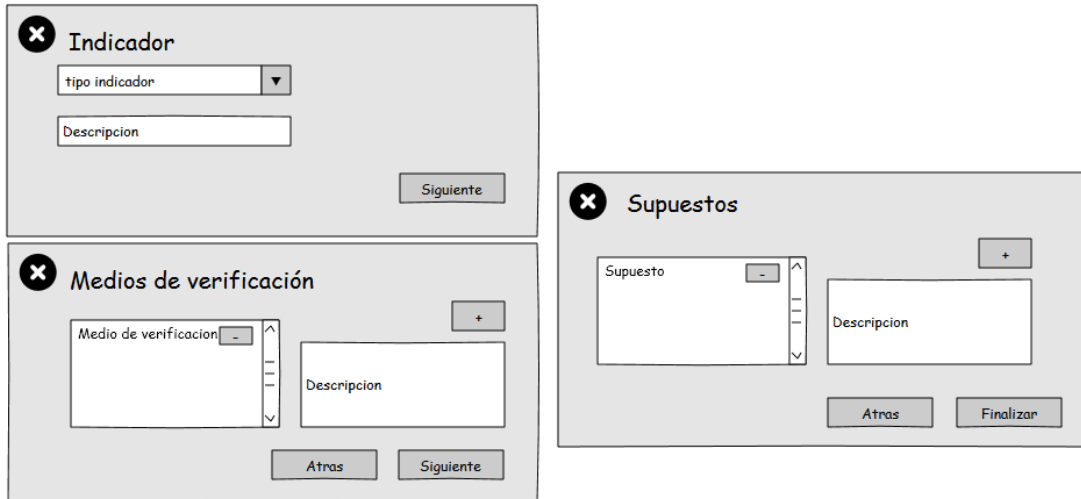


Figura 4.20 Gestión de indicadores, medios de verificación y supuestos.

Fuente: Propia

En la siguiente imagen se muestra el diseño de la gestión de los resúmenes narrativos del proyecto.

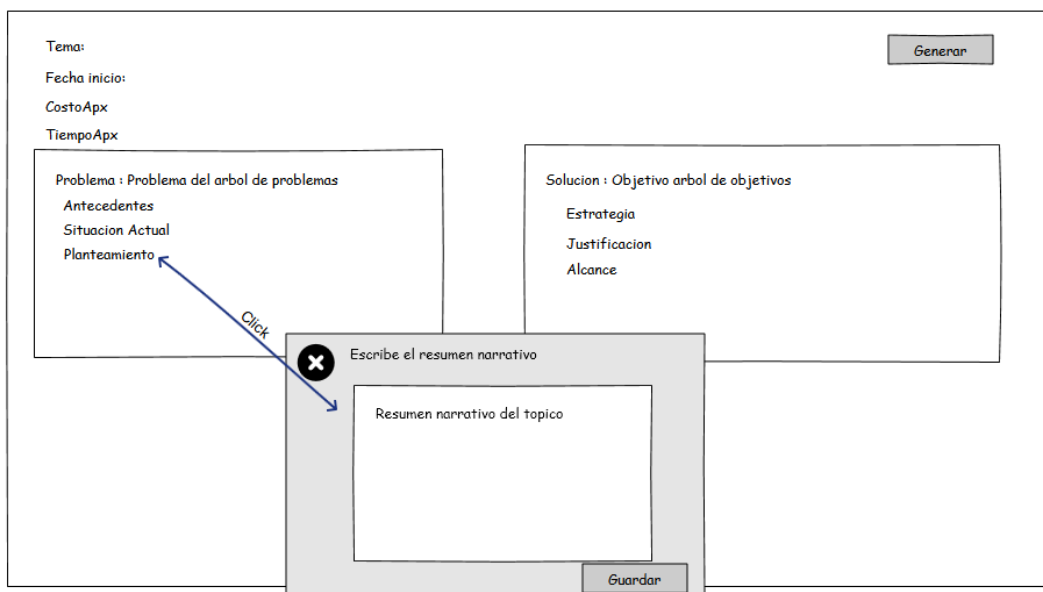


Figura 4.21 Gestión de los resúmenes narrativos del proyecto.

Fuente: Propia

- **Mockups seguimiento**

En la siguiente imagen se muestra el diseño de la página que mostrara los indicadores del proyecto una vez generado y su estado.

FIN	PROPOSITO	COMPONETES	ACTIVIDADES	
FIN	INDICADOR	MEDIOS VERIFICACION	SUPUESTOS	EVALUACIONES
FIN	INDICADOR	MEDIOS VERIFICACION	SUPUESTOS	EVALUACIONES
FIN	INDICADOR	MEDIOS VERIFICACION	SUPUESTOS	EVALUACIONES
FIN	INDICADOR	MEDIOS VERIFICACION	SUPUESTOS	EVALUACIONES
FIN	INDICADOR	MEDIOS VERIFICACION	SUPUESTOS	EVALUACIONES

Figura 4.22 Muestra los indicadores, medios de verificación y supuestos de cada componente de la matriz, así como las ultimas evaluaciones.

Fuente: Propia

En la siguiente imagen se muestra el formulario para la gestión de evaluaciones intermedias del proyecto, para verificar su avance.

Evaluación intermedia

Aquí se agregaran las revisiones o avances de los indicadores del proyecto

FIN
PROPOSITO
COMPONETES
ACTIVIDADES

Indicador | Evaluaciones | última evaluación | agregar evaluación | cumplido

Evaluaciones

Fecha	Medio Verificación	Resultado	documentos
Fecha	Medio Verificación	Resultado	ver
Fecha	Medio Verificación	Resultado	ver
Fecha	Medio Verificación	Resultado	ver
Fecha	Medio Verificación	Resultado	ver

Figura 4.23 Formulario para realizar seguimiento de cada indicador mediante evaluaciones.

Fuente: Propia

En el siguiente diseño se muestra la ventana modal que permitirá agregar nuevas evaluaciones a un indicador específico.

Figura 4.24 Formulario para agregar evaluaciones.

Fuente: Propia

La siguiente imagen muestra la ventana modal que permitirá agregar pruebas a una evaluación.

Figura 4.25 Formulario para agregar enlaces de los documentos o imágenes de pruebas de una evaluación.

Fuente: Propia

En la siguiente imagen se muestra el diseño de la ventana modal que permitirá verificar el estado actual en el que se encuentra el indicador, si está en ejecución, si ya finalizó o si se dio algún supuesto.

El formulario modal, titulado "Estado indicador", contiene los siguientes elementos:

- Un encabezado con un ícono de cerrar (X) y el título "Estado indicador".
- Una pregunta: "Se ha cumplido a su totalidad el indicador".
- Dos botones de opción de radio: "Si" (seleccionado) y "No".
- Una segunda pregunta: "Se dio al gun supuesto que impida su finalización".
- Un campo de texto con el placeholder "text goes here" y un menú desplegable.
- Un campo de texto con el placeholder "Otro supuesto".
- Un botón "Finalizar" ubicado a la derecha del campo de texto inferior.

Figura 4.26 Formulario para verificar el cumplimiento del indicador o su falla.

Fuente: Propia

4.3.8 Iteración 8

En la última iteración se procederá con la generación de reportes y ayuda al usuario para el proceso de la generación de un proyecto, esto dentro de la última historia de usuario con el número 9.

- **Mockups ayuda al usuario**

En la imagen a continuación se ve la ventana emergente que brindara ayuda en cada paso al usuario, dando consejos o guías para la resolución de cada paso o formulario.

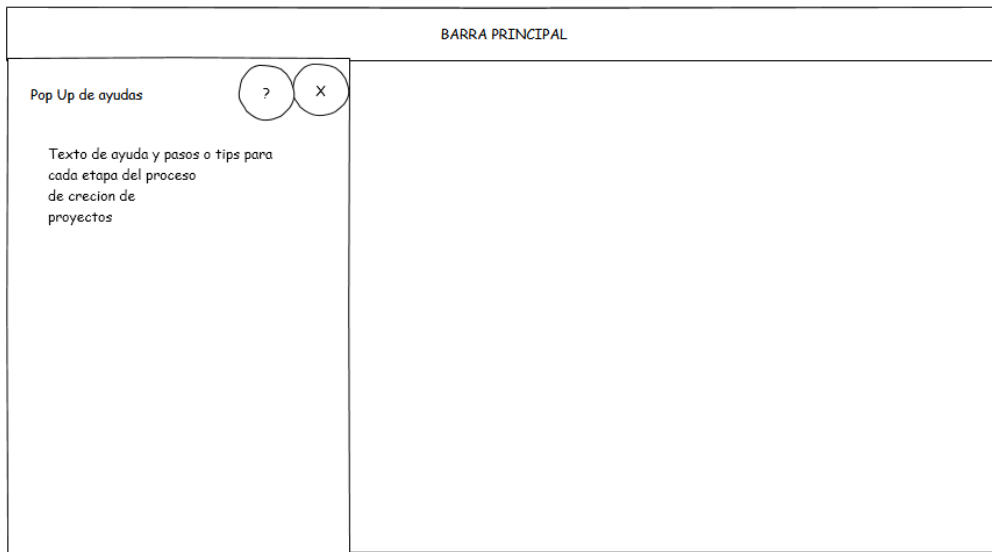


Figura 4.27 Pop up de ayuda al usuario con información de cada paso de la matriz.

Fuente: Propia

En la siguiente imagen se ve las opciones adicionales al perfil de usuario que permiten activar o desactivar las ayudas, así como cambiar el estilo de la aplicación.

BARRA PRINCIPAL

CUETA DE USUARIO

Configuraciones

Activar ayudas ▼

Estilo ▼

Figura 4.28 Configuración de ayudas.

Fuente: Propia

- **Mockups reportes**

En la siguiente imagen se ve el diseño del reporte de la matriz de marco lógico.

FIN			
Texto Texto	Indicadores Texto Texto	Medios de verificación Texto Texto	Supuestos Texto Texto
PROPOSITO			
Texto	Indicadores Texto	Medios de verificación Texto	Supuestos Texto
COMPONETES			
Texto Texto	Indicadores Texto Texto	Medios de verificación Texto Texto	Supuestos Texto Texto
ACTIVIDADES			
Texto Texto	Indicadores Texto Texto	Medios de verificación Texto Texto	Supuestos Texto Texto

Figura 4.29 Reporte de matriz de marco lógico.

Fuente: Propia

En el siguiente diseño se ve el formato del reporte del proyecto.

1. Tema
TEXTO
2. Problema
2.1. Antecedentes
2.2. Situacion Actual
2.3. Planteamiento
3. Objetivos
3.1. Principal
3.2. Secundarios
3. Alcance
3. Justificación

Figura 4.30 Reporte del proyecto

Fuente: Propia

En la siguiente imagen se muestra el reporte de seguimiento con sus evaluaciones por cada indicador.

TEMA			
FECHA INICIO			
FECHA ACTUAL			
FILA	INDICADOR	ULTIMA REVISION	ESTADO
TEXT0	TEXT0	TEXT0	TEXT0
REVISIONES			
FECHA		RESULTADO	ESTADO
FECHA		RESULTADO	ESTADO
FILA	INDICADOR	ULTIMA REVISION	ESTADO
TEXT0	TEXT0	TEXT0	TEXT0
REVISIONES			
FECHA		RESULTADO	ESTADO

Figura 4.31 Reporte de seguimiento

Fuente: Propia

4.4 DESARROLLO

4.4.1 Iteración 5

Se comienza con la parte de desarrollo, generando el proyecto y su estructura principal. Además se generan las historias de usuario 1 y 2.

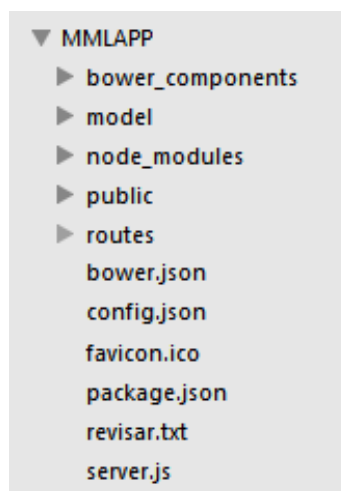


Figura 4.32 Estructura del proyecto

Fuente: Propia

En la siguiente imagen se muestra el resultado final del desarrollo de la interfaz de acceso al sistema.

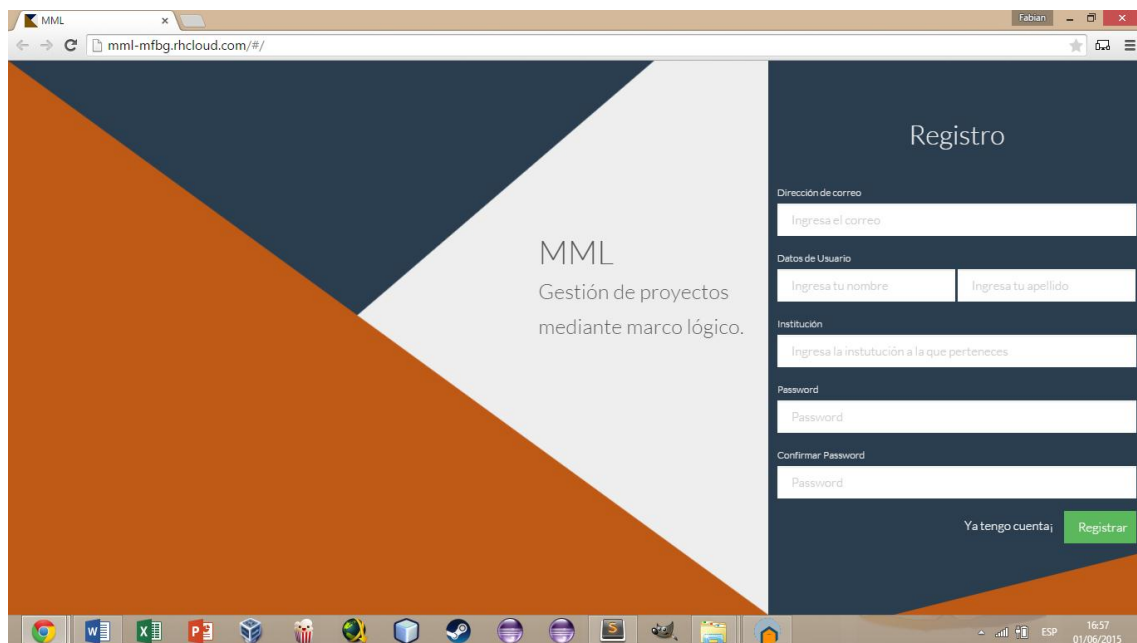


Figura 4.33 Acceso al sistema

Fuente: Propia

La **Figura 4.34** muestra la interfaz de gestión de proyectos para cada usuario ya en su diseño final y ya finalizado.

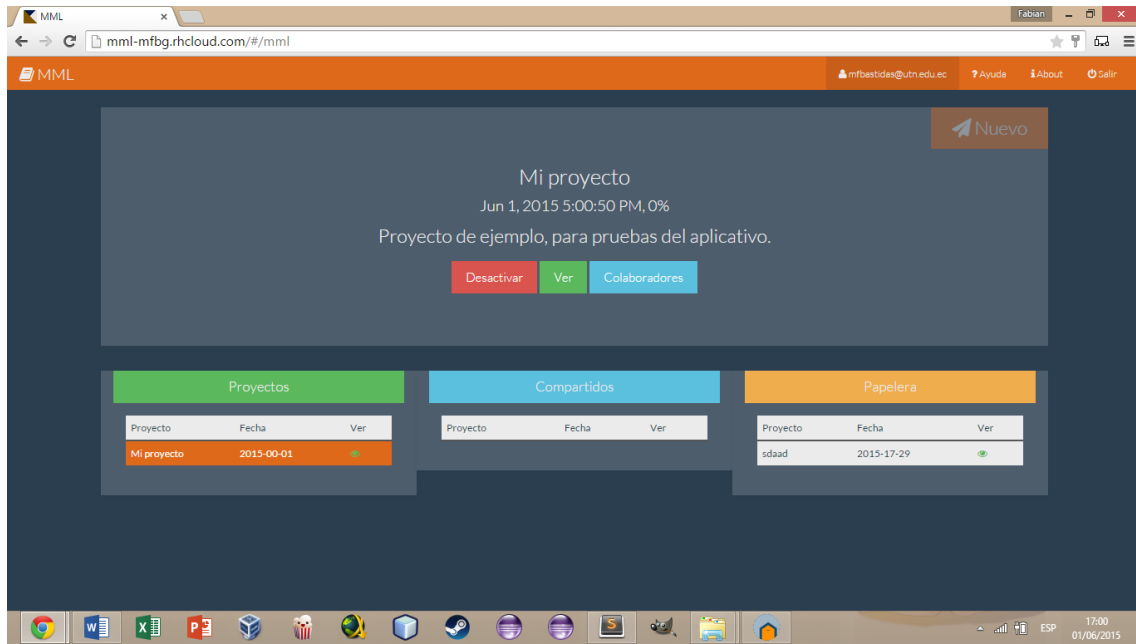


Figura 4.34 Gestión de proyectos

Fuente: Propia

En la siguiente imagen se ve el resultado final de la gestión de proyectos, que nos permite gestionar la elaboración de un proyecto.

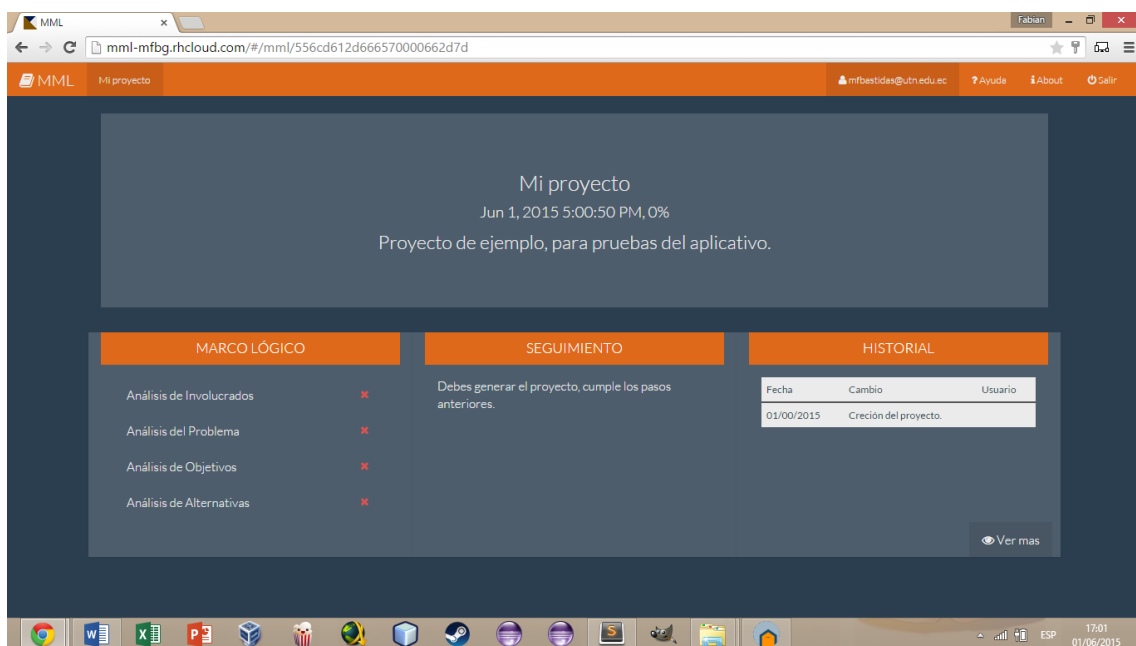


Figura 4.35 Gestión de un proyecto

Fuente: Propia

4.4.2 Iteración 6

En la siguiente iteración se trabajara en las historias de usuario 3, 4, 5 y 6 que son parte de la elaboración inicial del proyecto. A continuación se muestran los diseños previos de los formularios y páginas de cada historia.

Como se nos muestra en la siguiente imagen, que es la interfaz de gestión de usuarios, es el resultado de esta iteración.

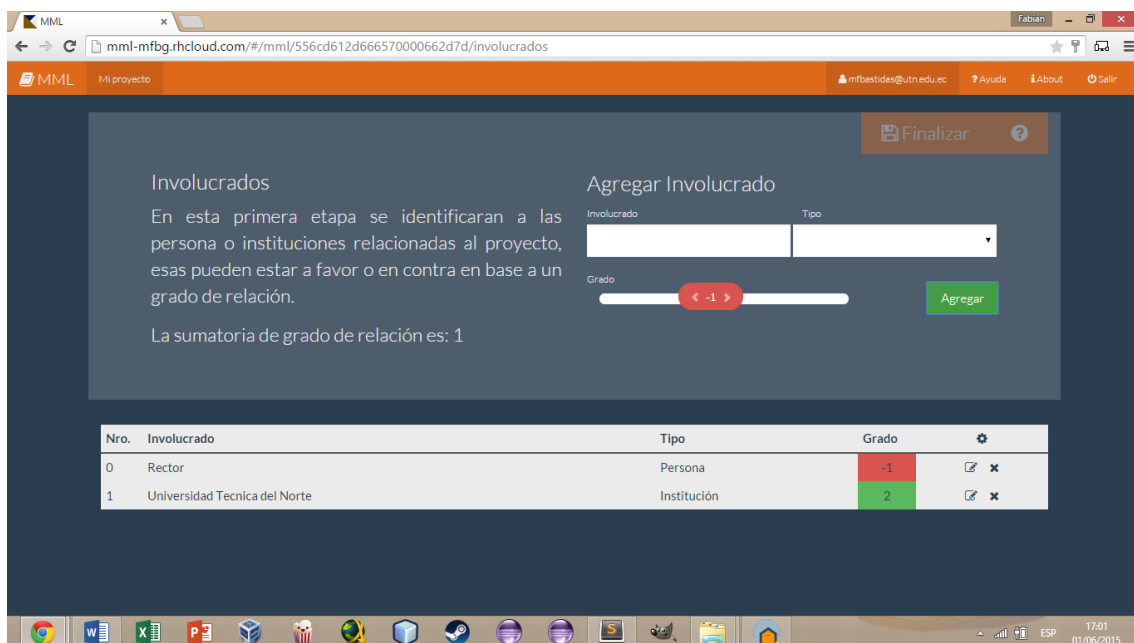


Figura 4.36 Gestión de involucrados

Fuente: Propia

En la siguiente figura nos muestra el resultado de la gestión del problema, esta interfaz nos permite la gestión del árbol de problemas del proyecto.

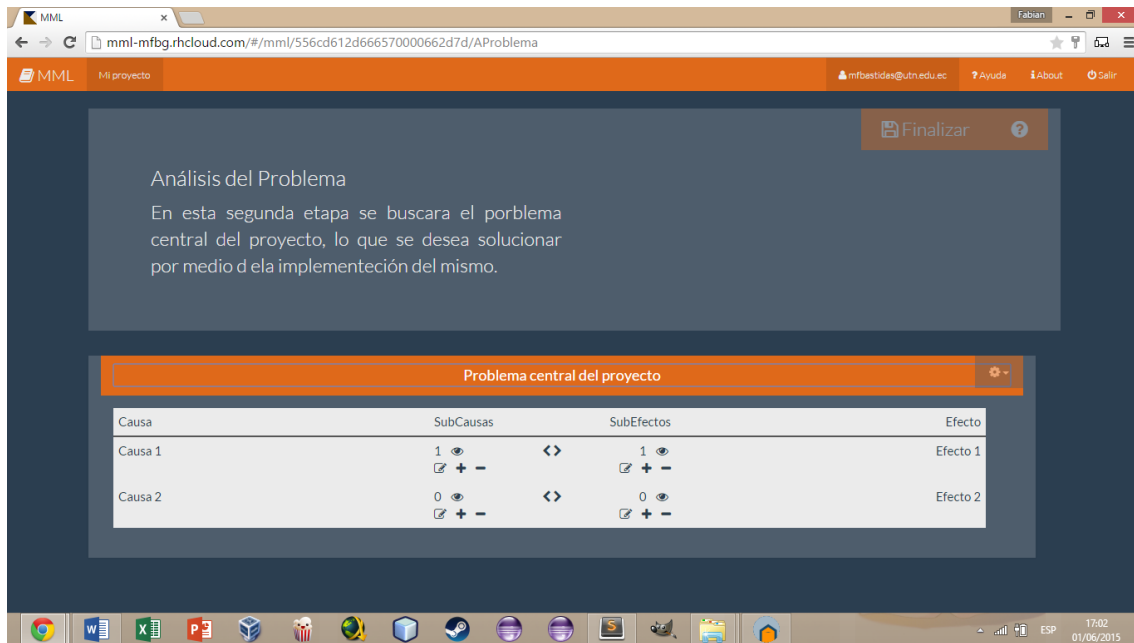


Figura 4.37 Gestión del problema

Fuente: Propia

En la siguiente figura nos muestra el resultado de la gestión de objetivos, esta interfaz nos permite la gestión del árbol de objetivos del proyecto.

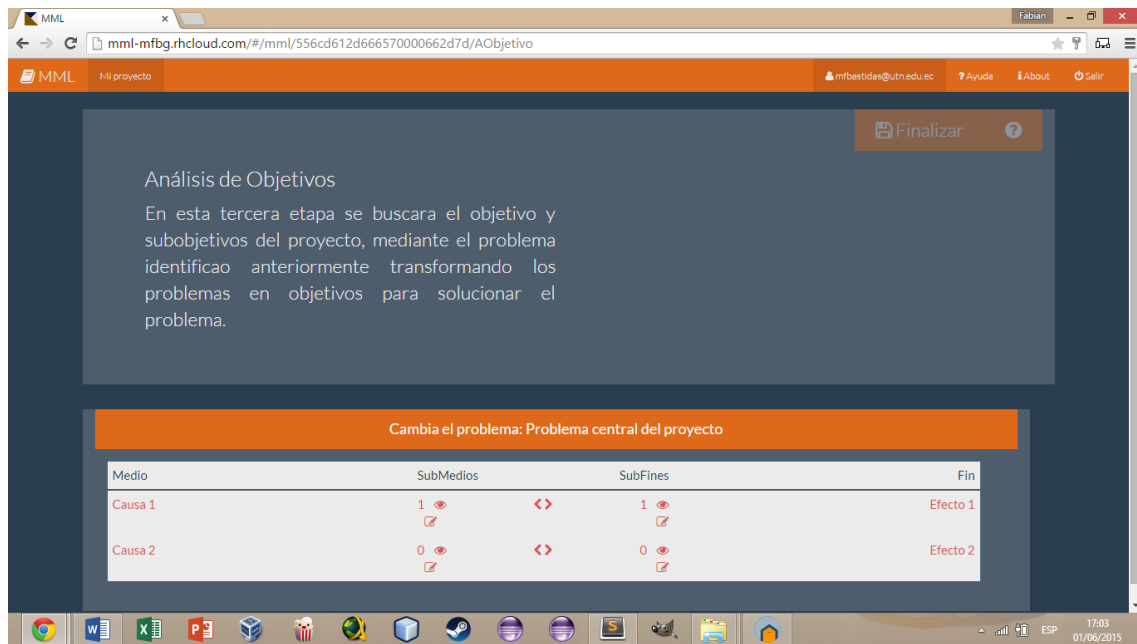


Figura 4.38 Gestión del objetivo

Fuente: Propia

La **Figura 4.38** es la interfaz que permite la gestión de estrategias y acciones del proyecto permitiendo escoger una de las varias combinaciones generadas.

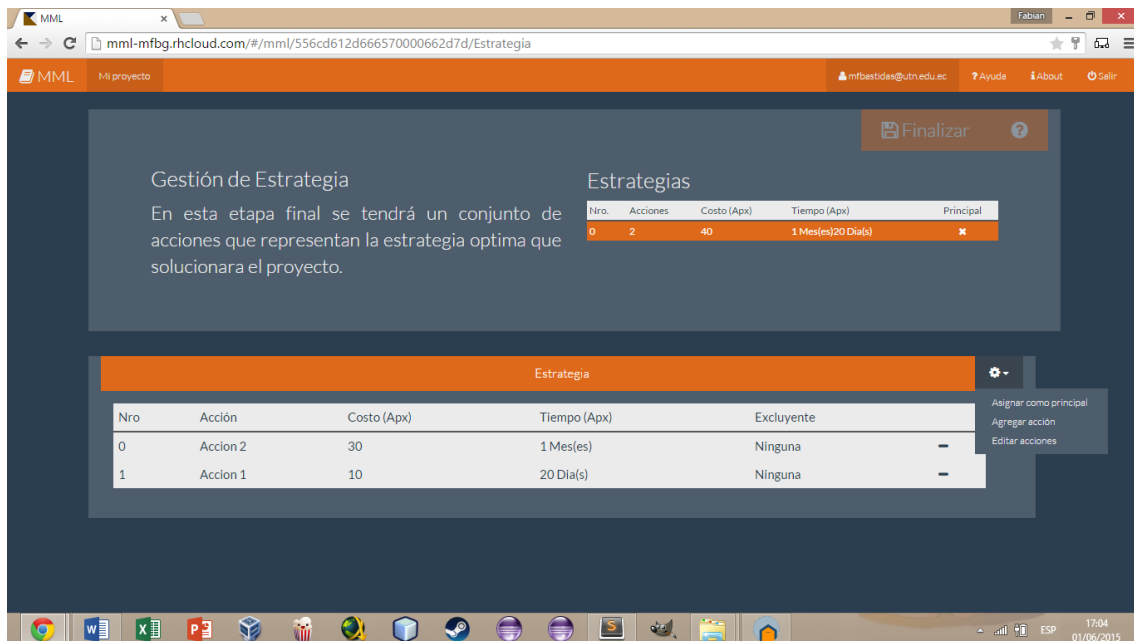


Figura 4.39 Gestión de estrategia

Fuente: Propia

4.4.3 Iteración 7

Se procede con el desarrollo de las historias 7 y 8 que constan de la generación del proyecto así como su seguimiento.

Como vemos a continuación el resultado de esta iteración son las siguientes capturas de pantalla.

En la siguiente imagen se ve la interfaz para la generación del proyecto, que muestra la primera columna de la matriz de marco lógico.

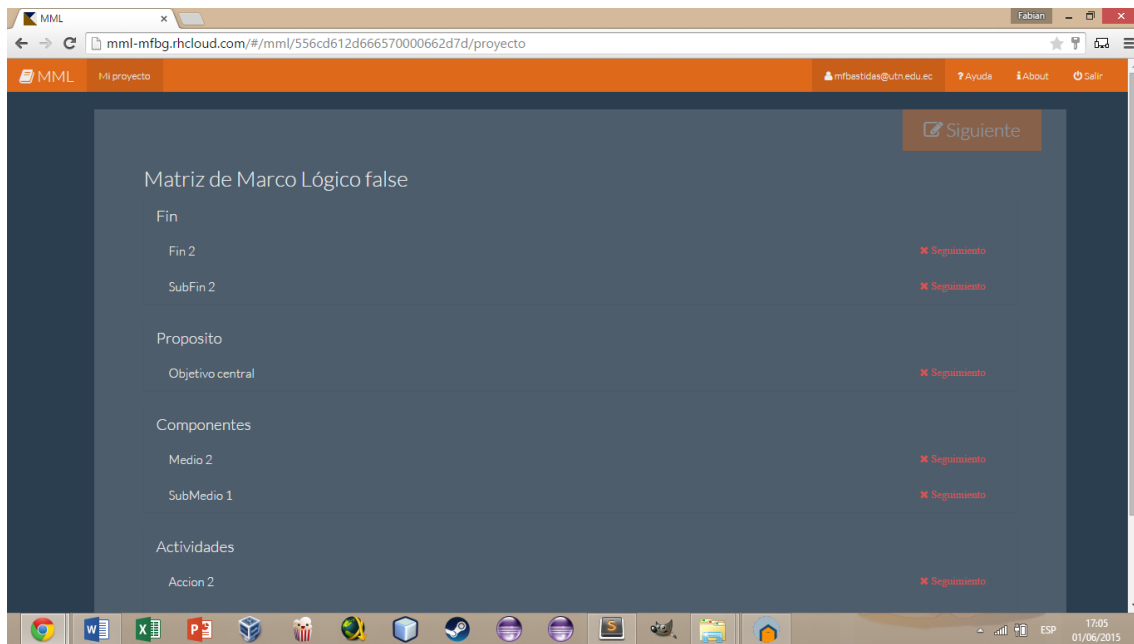


Figura 4.40 Generar proyecto, gestión de seguimiento

Fuente: Propia

En la siguiente imagen se ve la interfaz de gestión de los resúmenes narrativos del proyecto.

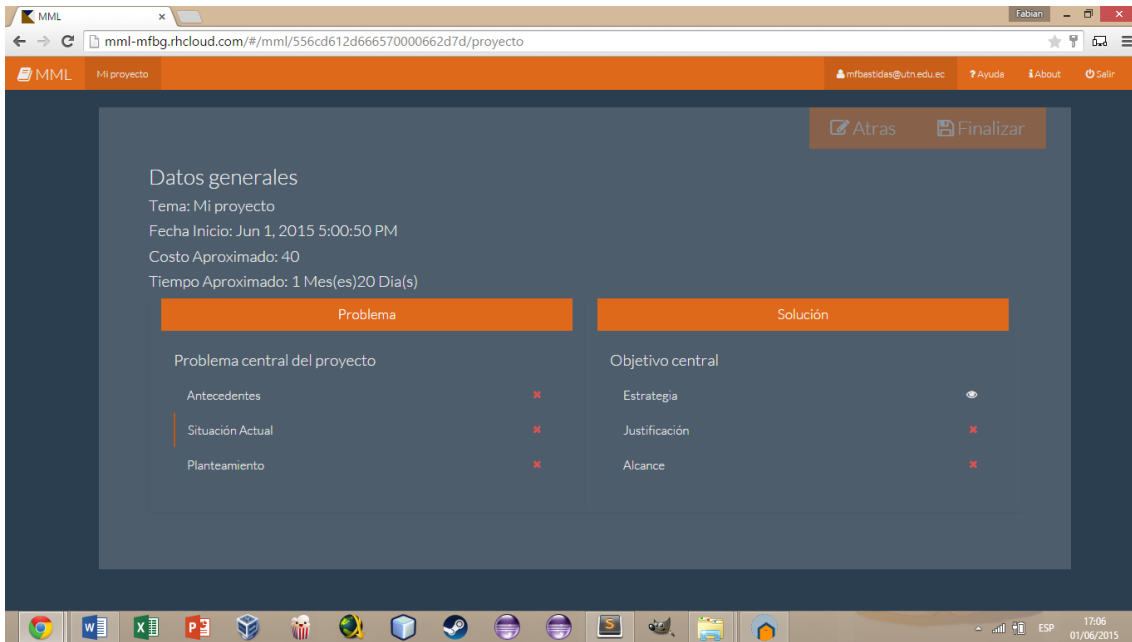


Figura 4.41 Generar proyecto, resúmenes narrativos

Fuente: Propia

La siguiente captura de pantalla nos muestra la página con los indicadores del proyecto una vez ya haya sido generado.

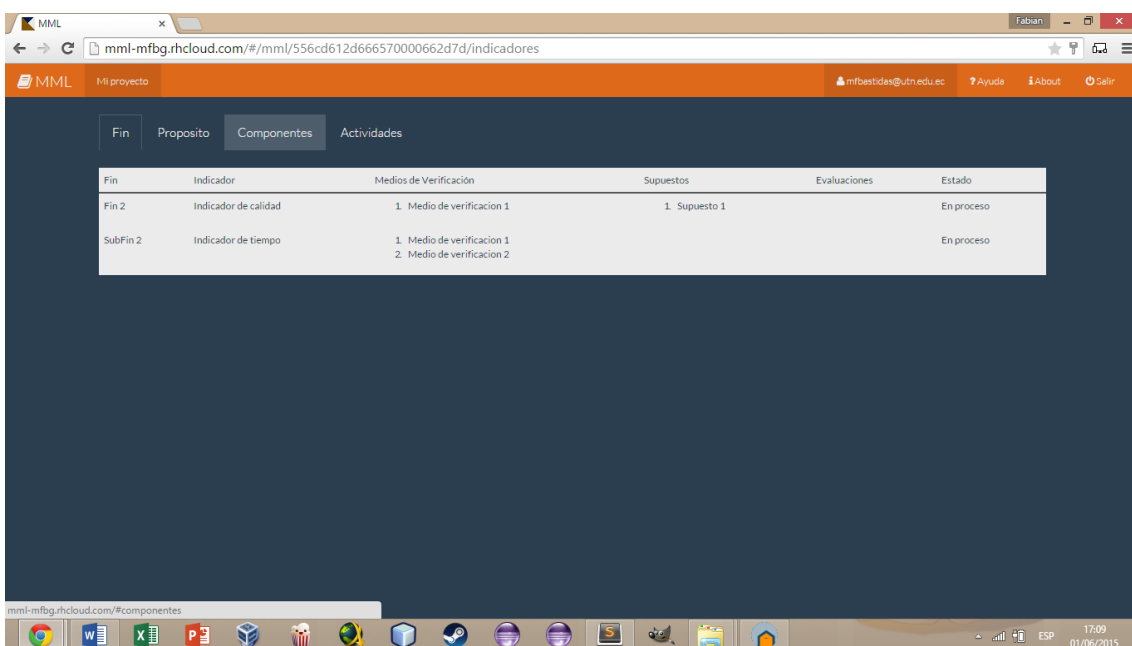


Figura 4.42 Ver indicadores

Fuente: Propia

La imagen que se ve a continuación es la interfaz que nos permite la agregación de evaluaciones a los indicadores ya definidos del proyecto.

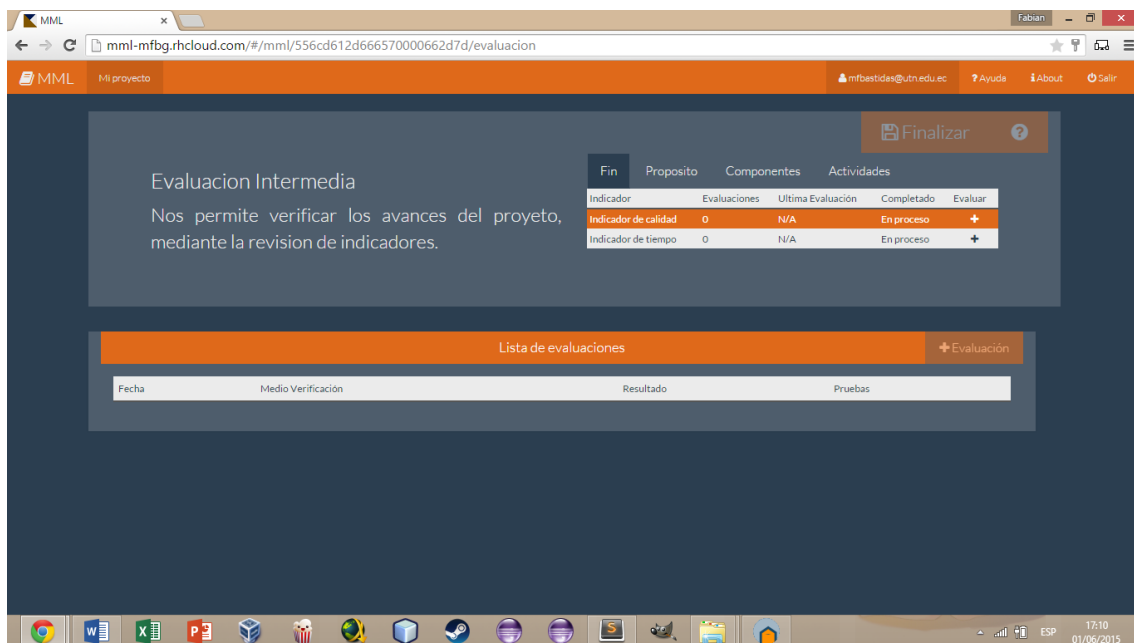


Figura 4.43 Gestión de evaluaciones

Fuente: Propia

4.4.4 Iteración 8

La parte final del desarrollo que consta de la historia de usuario 9. En donde se generan los reportes y la ayuda al usuario.

En la imagen a continuación se muestra la ventana pop up que estará en todas las páginas del sistema brindan información sobre cada paso.

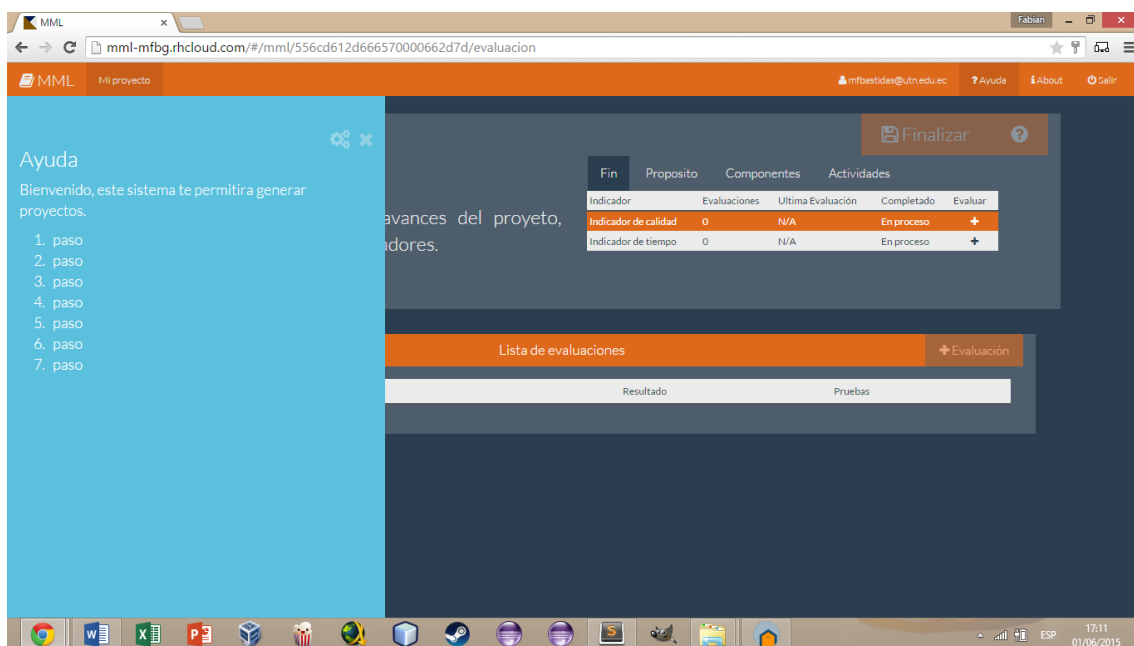


Figura 4.44 Pop up de ayuda

Fuente: Propia

En la **Figura 4.45** se ve las configuraciones de la ayuda al usuario para poder activarla o desactivarla según la necesidad.

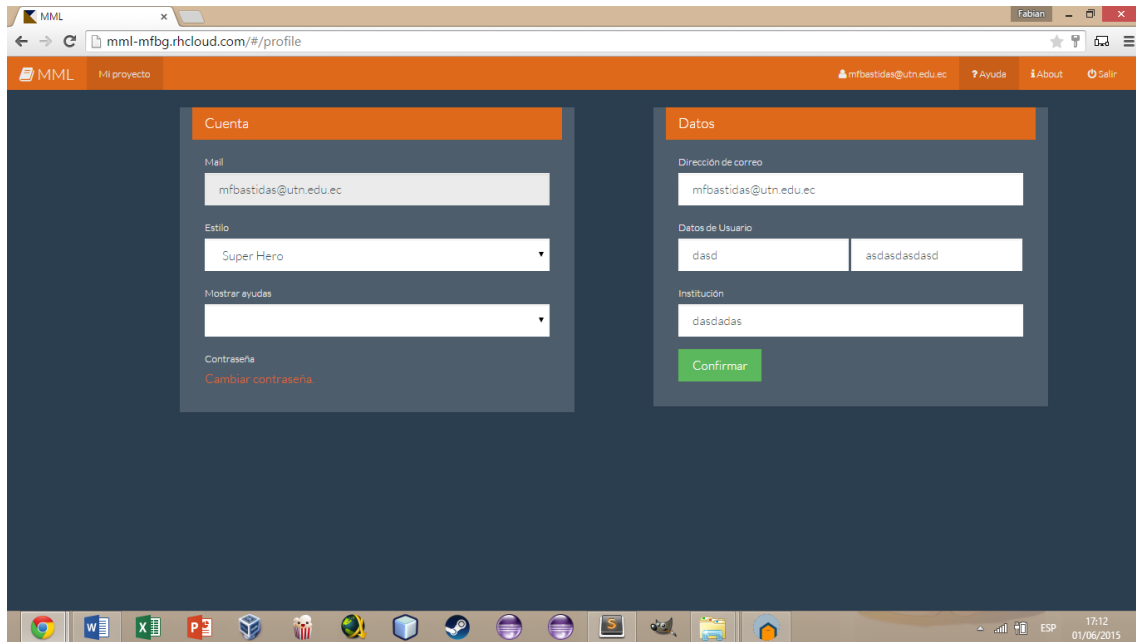


Figura 4.45 Configuración de ayudas

Fuente: Propia

4.5 PRUEBAS

En la siguiente sección se describe las pruebas realizadas sobre el sistema en base a los requerimientos iniciales descritos en las historias de usuario.

4.5.1 Historia de usuario 1 – Control de usuarios

Requisito	Validar campos de correo y clave
------------------	----------------------------------

Datos de entrada	Correo y contraseña
Resultado esperado	Mensaje de error con información de que faltan campos, mensaje para correos inválidos
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.58 Control de usuarios – validar acceso

Fuente: Propia

Requisito	Permitir acceder los usuarios al sistema mediante el correo y una clave
------------------	---

Datos de entrada	Correo y contraseña
Resultado esperado	Redirección a la página de gestión de proyectos
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.59 Control de usuarios – acceso

Fuente: Propia

Requisito	Validar campos para creación de una cuenta
Datos de entrada	Correo, contraseña, nombre, apellido e institución.
Resultados	Mensajes de error en caso de no completar los campos,

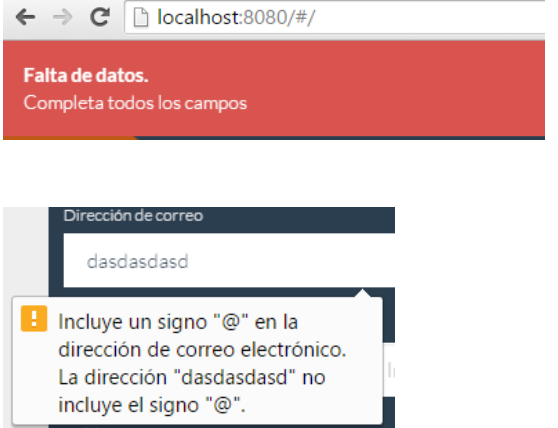
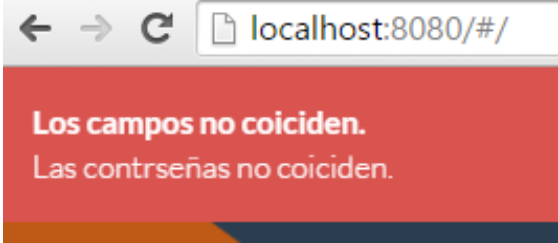
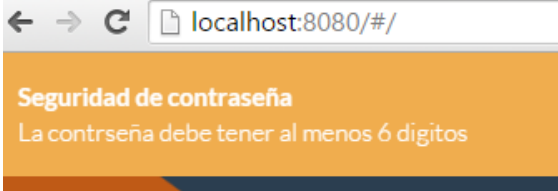
esperado	de correos erróneos, en la verificación de contraseñas y en la longitud de las mismas
Resultados obtenido	  
Revisión	1
Tester	Bastidas Fabian

Tabla 4.60 Control de usuarios – validar registro

Fuente: Propia

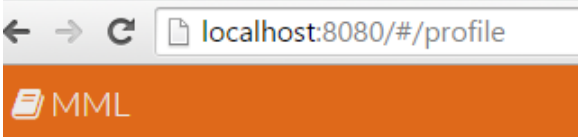
Requisito	Permitir a los nuevos usuarios crear cuentas
Datos de entrada	Correo, contraseña, nombre, apellido e institución.
Resultado esperado	Redirección al perfil de usuario
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.61 Control de usuarios – registro

Fuente: Propia

4.5.2 Historia de usuario 2 – Gestión de proyectos

Requisito	Validar la creación de proyectos
------------------	----------------------------------

Datos de entrada	Tema y descripción
Resultado esperado	Mensaje de campos faltantes
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.62 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

Requisito	Crear un nuevo proyecto
Datos de entrada	Tema y descripción

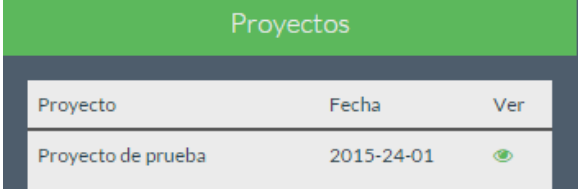
Resultado esperado	Nuevo proyecto en la lista de proyectos propios
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.63 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

Requisito	Desactivar proyectos
Datos de entrada	Proyecto a desactivar
Resultado esperado	Nuevo proyecto en la lista de proyectos desactivados

Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.64 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

Requisito	Eliminar permanentemente proyectos
Datos de entrada	Proyecto a eliminar
Resultado esperado	Eliminación definitiva del proyecto


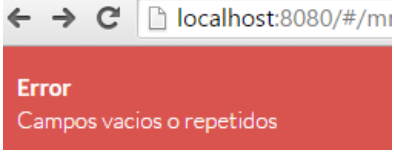
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.65 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

Requisito	Validar compartir proyectos
Datos de entrada	Correo del colaborador y permiso
Resultado esperado	Mensaje de error si no existe ese usuario, mensaje de error si no se completan todos los campos
Resultado obtenido	

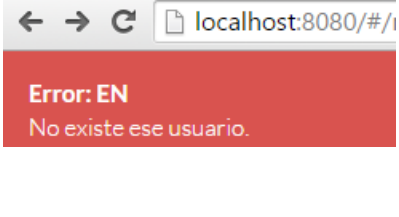
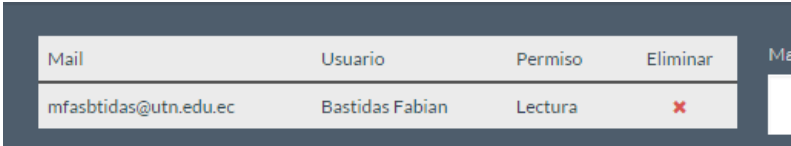
	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.66 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

Requisito	Agregar colaboradores al proyecto
Datos de entrada	Correo del colaborador y permiso
Resultado esperado	Nuevo colaborador en la lista del colaboradores del proyecto
Resultado obtenido	
Revisión	1

Tester	Bastidas Fabian
---------------	-----------------

Tabla 4.67 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

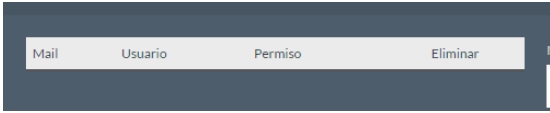
Requisito	Eliminar colaboradores al proyecto
Datos de entrada	Colaborador a eliminar
Resultado esperado	El colaborador desaparece de la lista
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.68 Gestión de proyectos – validar nuevo proyecto

Fuente: Propia

4.5.3 Historia de usuario 3 – Análisis de involucrados

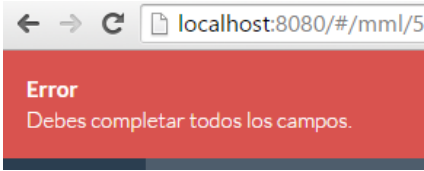
Requisito	Validar ingreso de involucrados
Datos de entrada	Involucrado y tipo
Resultado esperado	Mensaje de error de campos vacíos
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.69 Análisis de Involucrados – validar nuevo involucrado

Fuente: Propia

Requisito	Ingreso de involucrados				
Datos de entrada	Involucrado y tipo				
Resultado esperado	Nuevo involucrado en la lista de involucrados				
Resultado obtenido	<table border="1"> <thead> <tr> <th>Involucrado</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>Universidad Técnica del Norte</td> <td>Institución</td> </tr> </tbody> </table>	Involucrado	Tipo	Universidad Técnica del Norte	Institución
Involucrado	Tipo				
Universidad Técnica del Norte	Institución				
Revisión	1				
Tester	Bastidas Fabian				

Tabla 4.70 Análisis de Involucrados – validar nuevo involucrado

Fuente: Propia

Requisito	Ingreso de involucrados
------------------	-------------------------

Datos de entrada	Involucrado y tipo				
Resultado esperado	Nuevo involucrado en la lista de involucrados				
Resultado obtenido	<table border="1"> <thead> <tr> <th>Involucrado</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>Universidad Técnica del Norte</td> <td>Institución</td> </tr> </tbody> </table>	Involucrado	Tipo	Universidad Técnica del Norte	Institución
Involucrado	Tipo				
Universidad Técnica del Norte	Institución				
Revisión	1				
Tester	Bastidas Fabian				

Tabla 4.71 Análisis de Involucrados – insertar nuevo involucrado

Fuente: Propia

Requisito	Edición de involucrados
Datos de entrada	Involucrado y tipo
Resultado	Cambio de datos de un involucrado

esperado				
Resultado obtenido	<table border="1"> <tr> <td>1</td> <td>Universidad Técnica del Norte</td> <td>Persona</td> </tr> </table>	1	Universidad Técnica del Norte	Persona
1	Universidad Técnica del Norte	Persona		
Revisión	1			
Tester	Bastidas Fabian			

Tabla 4.72 Análisis de Involucrados – insertar nuevo involucrado

Fuente: Propia

Requisito	Eliminación de involucrados
Datos de entrada	Involucrado a eliminar
Resultado esperado	Quitar de la lista de involucrados al involucrado eliminado

Resultado obtenido	<table border="1"> <thead> <tr> <th>Involucrado</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>Decano de la Fica</td> <td>Persona</td> </tr> </tbody> </table>	Involucrado	Tipo	Decano de la Fica	Persona
	Involucrado	Tipo			
Decano de la Fica	Persona				
Revisión	1				
Tester	Bastidas Fabian				

Tabla 4.73 Análisis de Involucrados – insertar nuevo involucrado

Fuente: Propia

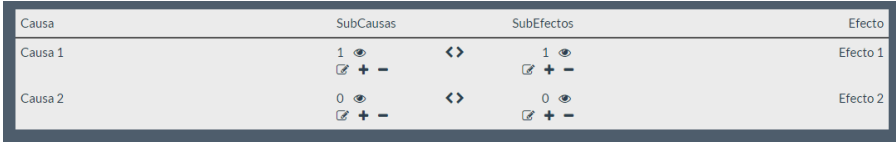
4.5.4 Historia de usuario 4 – Análisis del problema

Requisito	Validar causa - efecto
Datos de entrada	Causa y efecto
Resultado esperado	Mensaje de error que avisa que no se han completado todos los campos

Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.74 Análisis de Problema – validar causa-efecto

Fuente: Propia

Requisito	Insertar causa - efecto
Datos de entrada	Causa y efecto
Resultado esperado	Causa y efecto en la lista de causas y efectos
Resultado obtenido	

Revisión	1
Tester	Bastidas Fabian

Tabla 4.75 Análisis de Problema – insertar causa-efecto

Fuente: Propia


Requisito	Editar causa - efecto
Datos de entrada	Causa y efecto
Resultado esperado	Cambio de causa-efecto en la lista
Resultado obtenido	 <p>The screenshot shows a table with four columns: Causa, SubCausas, SubEfectos, and Efecto. The first row contains 'Alta tasa de mortalidad' under Causa, '1' under SubCausas, '1' under SubEfectos, and 'Mas muertes al año' under Efecto. The second row contains 'Causa 2' under Causa, '0' under SubCausas, '0' under SubEfectos, and 'Efecto 2' under Efecto. Each cell in the SubCausas and SubEfectos columns has a small icon (an eye) and a '+ -' symbol. There are also '<>' symbols between the SubCausas and SubEfectos columns. The table has a light gray background and a dark border.</p>
Revisión	1
Tester	Bastidas Fabian

Tabla 4.76 Análisis de Problema – editar causa-efecto

Fuente: Propia

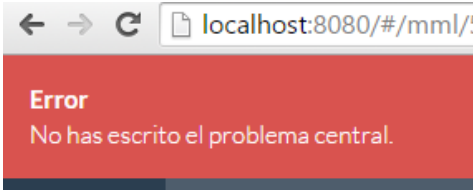
Requisito	Validar problema central antes de guardar
Datos de entrada	Problema central
Resultado esperado	Mensaje de error
Resultado obtenido	 A screenshot of a web browser window showing an error message. The address bar contains 'localhost:8080/#/mml/'. The error message is displayed in a red box with the text 'Error' and 'No has escrito el problema central.'
Revisión	1
Tester	Bastidas Fabian

Tabla 4.77 Análisis de Problema – validar problema central

Fuente: Propia

4.5.5 Historia de usuario 5 – Análisis de objetivos

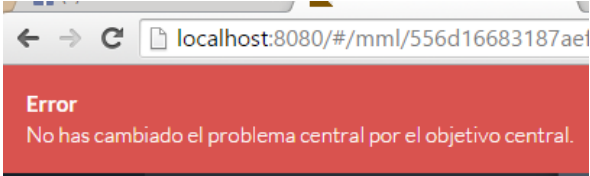
Requisito	Validar cambio de problema a objetivo
Datos de entrada	Objetivo central
Resultado esperado	Mensaje de error que no se ha cambiado el problema por el objetivo
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.78 Análisis de Objetivos – validar objetivo central

Fuente: Propia

Requisito	Validar cambio de causa-efecto a medio-fin
------------------	--

Datos de entrada	Medio y fin
Resultado esperado	Mensaje de error que no se ha cambiado todas las causas-efectos por medios-fines
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.79 Análisis de Objetivos – validar cambio de causa-efecto a medio-fin

Fuente: Propia

4.5.6 Historia de usuario 6 – Análisis de alternativas

Requisito	Validar generación de acciones
Datos de	Acción, excluyente, costo y tiempo

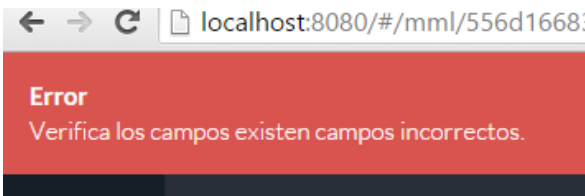
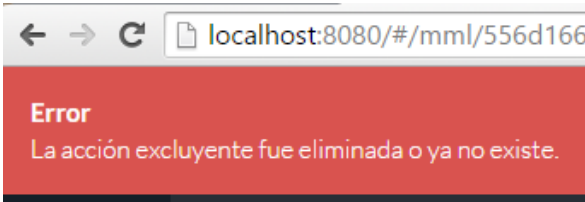
entrada	
Resultado esperado	Mensajes de error si ingresa datos erróneos
Resultado obtenido	 
Revisión	1
Tester	Bastidas Fabian

Tabla 4.80 Análisis de Alternativas – validar generación de acciones

Fuente: Propia

Requisito	Validar la agregación de acciones a una estrategia
Datos de	Acción

entrada	
Resultado esperado	Mensajes de error informando de la acción errónea
Resultado obtenido	 
Revisión	1
Tester	Bastidas Fabian

Tabla 4.81 Análisis de Alternativas – validar objetivo central

Fuente: Propia

4.5.7 Historia de usuario 7 – Generación de la matriz de marco lógico

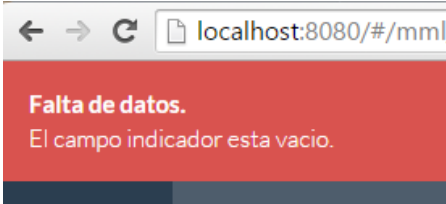
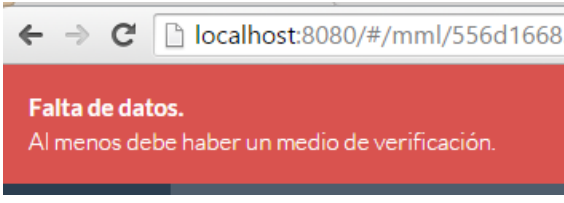
Requisito	Validar agregación de seguimiento a las filas
Datos de entrada	Indicador, medios de verificación y supuestos
Resultado esperado	Mensaje de error que muestre información de los campos faltantes
Resultado obtenido	 
Revisión	1
Tester	Bastidas Fabian

Tabla 4.82 Generación de la matriz de marco lógico – validar objetivo central

Fuente: Propia

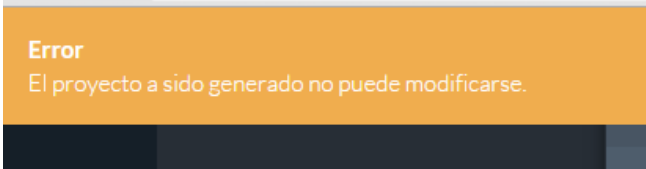
Requisito	Validar no edición de proyecto creado
Datos de entrada	Cambios al proyecto
Resultado esperado	Mensaje de error que mencione que el proyecto ya ha sido generado
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.83 Generación de la matriz de marco lógico – validar no edición de proyecto creado

Fuente: Propia

Requisito	Validar al menos el indicador del objetivo principal
------------------	--

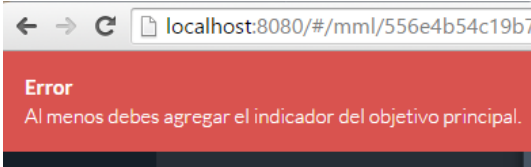
Datos de entrada	Seguimiento
Resultado esperado	Mensaje de error que mencione que el proyecto no tiene ningún indicador
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.84 Generación de la matriz de marco lógico – validar al menos el indicador del objetivo principal

Fuente: Propia

4.5.8 Historia de usuario 8 – Gestión de seguimiento del proyecto

Requisito	No permitir agregar evaluaciones si no se selecciona un indicador
Datos de	Seleccionar indicador

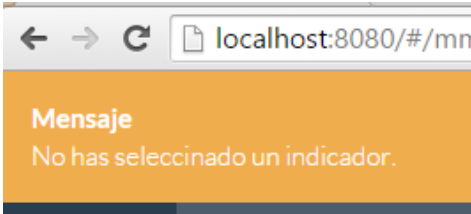
entrada	
Resultado esperado	Mensaje de advertencia
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.85 Generación de seguimiento del proyecto – evaluaciones

Fuente: Propia

Requisito	Validar inserción de evaluaciones
Datos de entrada	Medio de verificación, pruebas y resultado
Resultado	Mensaje de advertencia por los campos faltantes.

esperado	
Resultado obtenido	 
Revisión	1
Tester	Bastidas Fabian

Tabla 4.86 Generación de seguimiento del proyecto – insertar evaluaciones

Fuente: Propia

Requisito	Validar inserción de pruebas
Datos de entrada	Descripción y URL
Resultado	Mensaje de advertencia por los campos faltantes.

esperado	
Resultado obtenido	 
Revisión	1
Tester	Bastidas Fabian

Tabla 4.87 Generación de seguimiento del proyecto – validar insertar pruebas

Fuente: Propia

Requisito	No permitir seguir evaluando después de la finalización de un indicador
Datos de	Nueva evaluación

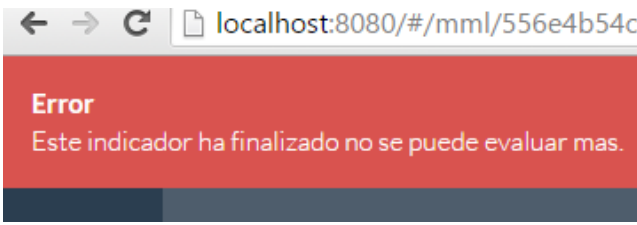
entrada	
Resultado esperado	Mensaje de error
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.88 Generación de seguimiento del proyecto – no insertar más evaluaciones después del fin del indicador

Fuente: Propia

Requisito	Insertar evaluación a un indicador
Datos de entrada	Medio verificación, resultado, pruebas y estado indicador
Resultado	Evaluación en la lista de evaluaciones

esperado	
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.89 Generación de seguimiento del proyecto – insertar indicador

Fuente: Propia

4.5.9 Historia de usuario 9 – Generación de reportes y ayuda

Requisito	Solo acceder cuando el proyecto este generado
Datos de entrada	Seleccionar reporte
Resultado esperado	Mensaje de advertencia que el proyecto no está generado

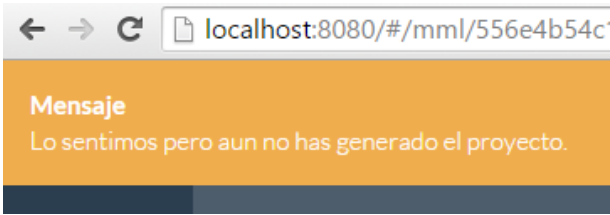
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.90 Generación de reportes y ayuda – Solo acceder cuando el proyecto este generado

Fuente: Propia

Requisito	Descargar PDF
Datos de entrada	Seleccionar descargar reporte
Resultado esperado	Documento PDF con el reporte

Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.91 Generación de reportes y ayuda – Descargar PDF

Fuente: Propia

Requisito	Descargar DOC
Datos de entrada	Seleccionar descargar reporte
Resultado esperado	Documento DOC con el reporte

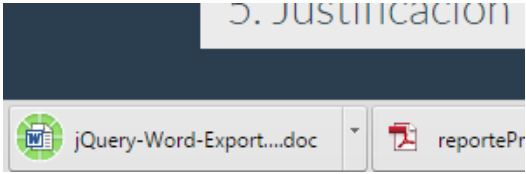
Resultado obtenido	
Revisión	1
Tester	Bastidas Fabian

Tabla 4.92 Generación de reportes y ayuda – Descargar DOC

Fuente: Propia



CAPITULO V



- 8. ANÁLISIS, CONCLUSIONES
Y RECOMENDACIONES
 - 8.1 ANÁLISIS DE IMPACTO
 - 8.2. CONCLUSIONES
 - 8.3. RECOMENDACIONES

5 ANÁLISIS, CONCLUSIONES Y RECOMENDACIONES

5.1 ANÁLISIS DE IMPACTO

5.1.1 Social

En el ámbito social el desarrollo de aplicaciones web en general contribuyen a la sociedad; mediante la globalización y distribución de conocimiento a mejorar su calidad de vida. La gestión de proyectos es un acto social en el que una o varias personas generan soluciones para la sociedad, automatizando y haciendo uso de las nuevas tecnologías, esto a su vez permite una participación más activa de los individuos involucrados en el desarrollo social.

En la siguiente tabla se hace un análisis sobre el impacto de este proyecto sobre la sociedad:

Nivel de impacto Indicadores	3	2	1	0	-1	-2	-3	TOTAL
Inclusión social		X						2
Condiciones de vida		X						2
Contribución social			X					1
TOTAL								5

Tabla 5.1 Tabla de medición de indicadores sociales

Fuente: Propia

Para realizar el cálculo se utiliza la siguiente fórmula, que determina el índice que corresponde a cada impacto.

$$\text{Nivel de Impactos} = \frac{\sum \text{de Impactos}}{\text{Número de Impactos}}$$

$$\text{Impacto social} = \frac{5}{3}$$

$$\text{Impacto social} = 1,66... \approx 2$$

Análisis:

- **Inclusión social**

El sistema de gestión de proyectos mediante matriz de marco lógico al estar sobre la nube y ser accesible mediante internet permite que las personas tener más facilidades de colaboración ayudando a que se pueda trabajar más rápidamente desde sitios más cómodos, brindando un factor de inclusión social que muchas veces se ve mermado por la dificultad de localización o movilidad de las personas.

- **Condiciones de vida**

El sistema de gestión de proyectos mediante matriz de marco lógico es una herramienta didáctica que cuenta con la ayuda necesaria para que los usuarios puedan generar proyectos de manera más fácil y eficaz, además de hacer uso de la metodología de marco lógico que brinda facilidades y un proceso claro y simple

para la gestión de proyectos; mejorando la calidad de vida y el trabajo en equipo mediante sistemas distribuidos, además permite a las personas hacer mejor uso de su tiempo y a producir proyectos de manera fácil y con estándares internacionales.

- **Contribución social**

Al hacer uso de un estándar de la Comisión Económica para América Latina y el Caribe; contribuye a la producción de proyectos de calidad que generan las personas, aportando a la sociedad proyectos de calidad y un análisis previo mediante la matriz de marco lógico, que permite crear soluciones a problemas actuales de la sociedad a través de un proceso secuencial y fácil de entender.

5.1.2 Económico

Dentro de la economía los proyectos son la base del crecimiento y desarrollo económico, mediante esta herramienta que ayuda a la generación y seguimiento de los mismos; se mejora enormemente el ámbito económico de un grupo social. El uso de esta herramienta accesible y fácil de usar permite reducir el tiempo y mejorar la calidad de la generación y seguimiento de proyectos.

En la siguiente tabla se hace un análisis sobre el impacto económico:

Nivel de impacto Indicadores	3	2	1	0	-1	-2	-3	TOTAL
Reducción de gastos		X						2
Ahorro de tiempo	X							3
Productividad			X					1
TOTAL								6

Tabla 5.2 Tabla de medición de indicadores económicos

Fuente: Propia

$$\text{Nivel de Impactos} = \frac{\sum \text{de Impactos}}{\text{Número de Impactos}}$$

$$\text{Impacto económico} = \frac{6}{3}$$

$$\text{Impacto económico} = 2$$

Análisis:

- **Reducción de gastos**

El gasto implicado en la elaboración de proyectos no es un factor determinante pero si puede ver afectada la calidad del mismo debido al uso de papel que no es un medio cien por ciento fiable para almacenar información, además de los gastos

implicados por la movilización de cada integrante de un grupo, gracias a esta herramienta se pretende reducir los costos de los mencionados anteriormente como es el consumo de insumos de oficina, además al estar sobre una nube en internet permite trabajar sin necesidad de realizar reuniones con el grupo de trabajo en caso de que así el proyecto lo requiera.

- **Ahorro de tiempo**

Cualquier herramienta de software tiene como principio reducir el tiempo de un proceso de negocio, en el caso de la herramienta para gestión de proyectos mediante la matriz de marco lógico no es la excepción, a través de la sistematización y uso del marco lógico que es una metodología para gestión de proyectos de calidad se reduce el proceso de creación de un proyecto generando una documentación adecuada del mismo de manera simple y didáctica en un menor tiempo y colaborando a varias personas a la vez.

- **Productividad**

Mediante proyectos de calidad, estructurados y bien formulados la productividad está garantizada, además de realizar un buen seguimiento tomando en cuenta indicadores para la evaluación de objetivos, esto es lo que la herramienta ofrece; mejorar los proyectos que se generan, asimismo brindar de manera centralizada la verificación, avance y comprobar su desarrollo en tiempo real.

5.1.3 Ambiental

En la actualidad el medio ambiente es un factor casi determinante en cualquier proyecto que se desarrolle ya que puede ser un factor de aceptación definitivo para que el proyecto se cumpla a cabalidad.

El software en relación al medio ambiente presenta puntos a favor como es la reducción del uso de materiales.

En la siguiente tabla se hace un análisis sobre el impacto de este proyecto sobre el ambiente:

Nivel de impacto								
Indicadores	3	2	1	0	-1	-2	-3	TOTAL

Contaminación				X				0
Huella ecológica		X						2
Deforestación		X						2
TOTAL								4

Tabla 5.3 Tabla de medición de indicadores ambientales

Fuente: Propia

Para realizar el cálculo se utiliza la siguiente fórmula, que determina el índice que corresponde a cada impacto.

$$\text{Nivel de Impactos} = \frac{\sum \text{de Impactos}}{\text{Número de Impactos}}$$

$$\text{Impacto social} = \frac{4}{3}$$

$$\text{Impacto social} = 1,33... \approx 1$$

Análisis:

- **Contaminación**

El software genera contaminación tecnológica, además del uso de energía que proviene de fuentes no renovables que contaminan el medio ambiente.

El sistema de gestión de proyectos mediante la matriz de marco lógico está diseñado con una arquitectura PASS, es decir es un servicio único y que esta

sobre una nube, no necesita de discos de instalación, también reduce el uso de hardware mediante servidores virtualizados que son provistos por el servicio de PASS OPENSIFT.

Al estar sobre un PASS si existe gasto energético que es un factor contaminante, pero al ser una empresa a nivel internacional el proveedor del PASS como lo es OPENSIFT garantiza el uso mínimo de energía y siempre tendiendo a las fuentes de energía renovables.

- **Huella ecológica**

Ayuda a la reducción de insumos de oficina, haciendo uso de arquitecturas PASS que reducen el uso de hardware que genera la herramienta. Es mediante esta pequeña contribución al ambiente que se demuestra que se pueden generar proyectos de software más amigables con el medio ambiente sin hacer uso de hardware excesivo.

- **Deforestación**

Mediante la reducción del consumo de papel en todos los pasos de generación y seguimiento de un proyecto se reduce en gran medida la deforestación, ya que el proyecto y todos sus pasos para su elaboración, revisión y seguimiento se almacenan de manera digital.

5.2 CONCLUSIONES

- La gestión de proyectos es una ciencia que contiene sin fin de metodologías y forma para generar proyectos de calidad, la matriz de marco lógico aunque ya antigua ha demostrado que es una de las más eficientes y fáciles de usar mediante sus organizadores gráficos sus metodologías nos brinda una manera fácil y simple de generar proyectos de calidad.
- La matriz de marco lógico también es una buena forma de hacer seguimiento y verificar el cumplimiento de las fases de un proyecto mediante sus indicadores brindando una manera útil y eficaz de revisar los avances de un proyecto y mantenernos al tanto de su desarrollo.
- Gracias a los métodos didácticos como el árbol de problemas y árbol de objetivos se garantiza una correcta identificación de componentes de un proyecto así como una correcta relación entre componentes de un proyecto.
- Actualmente existen gran cantidad de nuevas tecnologías en el mercado que nos permiten realización de grandes aplicaciones más rápidamente y eficientemente.
- Gracias al crecimiento del internet y de sus tecnología ahora es muy fácil poder tener aplicaciones montadas en la red sin tanta complicación, esto gracias a los **PAAS** que nos ofrecen una manera simple de tener un espacio de desarrollo para nuestras aplicaciones sin tener que preocuparnos de configuraciones innecesarias, a veces largas y tediosas que nos quitan tiempo y complican el despliegue de las aplicaciones.

- Los **PAAS** son una gran forma de desplegar aplicaciones en el internet para los programadores que no tiene los recursos necesarios para tener sus propias estructuras de despliegue como centros de datos.
- Dentro de los nuevos sistemas de almacenamiento existen varias opciones, **MongoDB** es un gigante actualmente en lo que a almacenamiento **no SQL** se refiere, brinda gran velocidad en consultas y presenta un modelo de datos basado en **BJSON** simple que puede ser cambiado y modelado fácilmente con los cambios que se presente.
- **MongoDB** no define esquemas de datos específicos o fijos estos pueden cambiar drásticamente unos de otros mejorando su escalabilidad.
- JavaScript Ancio como lenguaje de scripting, su uso se basó netamente en mejorar las páginas web, pero gracias a su versatilidad y paradigma de programación es un lenguaje que actualmente sirve para programar cosas más allá que simples mejoras a páginas web.
- Gracias a **NodeJS javascript** brinda las funcionalidad es de un lenguaje de programación dando las oportunidades de realizar grandes cosas con este, aun joven, lenguaje de programación.
- El manejo de concurrencia en **javascript** con **NodeJS** es más limpio y fácil debido a su paradigma de programación orientada a eventos, un poco complejo de entender este nuevo formato de programación pero a su vez presenta cualidades que nos son útiles en el desarrollo de aplicaciones en tiempo real.
- Requiere cierto nivel de entendimiento en el manejo de concurrencia pasar de un lenguaje de bloqueo a uno de no bloqueo como lo es **javascript**.

- **NodeJS** presenta gran cantidad de módulos o componentes desarrollados por la comunidad y bajo licencias gratuitas que facilitan el desarrollo de aplicaciones y mejoran el alcance y la extensibilidad de este lenguaje de programación.
- **AngularJS** un conjunto de librerías de **javascript** que permiten pasara gran cantidad de trabajo del servidor al lado del cliente, nos brinda una clara y eficiente forma de mejorar la velocidad de nuestras aplicaciones restando ciertas tareas del servidor y pasándolas al lado del cliente.
- **AngularJS** permite organizar de mejor manera el código del lado del cliente esto gracias a su modelo **MVC** (Modelo Vista Controlador) en el lado del cliente.
- Gracias a las librerías de **AngularJS** el desarrollo de las vistas se lo realiza más rápidamente y eficazmente.
- **NodeJS** dentro del desarrollo de aplicaciones web nos presenta grandes ventajas como lo es la portabilidad ya que no requiere de un servidor de aplicaciones este ya está embebido dentro del motor de **NodeJS**, a diferencia de otros lenguajes que requieren de un servidor de aplicaciones para que las diferentes aplicaciones puedan ser desplegadas.
- El paquete o conjunto de tecnologías **MEAN** (MongoDB Express AngularJS NodeJS) en una clara y eficaz herramienta para el desarrollo de aplicaciones en tiempo real que por su naturaleza y uso estricto de **javascript** y **JSON** como principal lenguaje hacen que este conjunto este bien integrado y brinde una simple y mejor manera de generar aplicaciones web.

- Los servicios REST son una forma más rápida y simple, pero menos segura de generar arquitecturas orientadas a servicios.
- La metodologías XP es una manera rápida de generar documentación y fácil para el desarrollador sin exceso de procesos repetitivos y tediosos sobre la documentación.
- La metodología XP es una excelente metodología para sistemas computacionales pequeños y medianos.

5.3 RECOMENDACIONES

- Enseñar el uso de metodologías de generación y seguimiento de proyecto para que los estudiantes tengan la oportunidad de generar proyectos por si mismos de manera independiente.
- La matriz de marco lógico es una excelente metodología de gestión de proyectos ya que nos brinda una forma bien clara y definida de identificar el problema y su solución, por ende se recomienda su uso en la elaboración de proyectos e emprendimiento ya que además es un formato exigido por parte de la **SENESCYT**.
- Nuevas tecnologías nacen cada día pero se des actualizan rápidamente, estar siempre al día de lo que viene dentro del mudo de la informática siempre es necesario para poder estar al día con el futuro de la carrera.
- El conjunto de herramientas **MEAN** (MongoBD Express AngularJS NodeJS) es una herramienta muy completa para el desarrollo web y existe una gran comunidad, hacer uso de esta herramienta para la generación de aplicaciones es muy factible para emprendedores.

- El aprender el uso de javascript requiere cierto nivel de practica por eso es muy recomendable realizar cursos de javascript y entender sobre el código de no bloqueo antes de empezar a realizar aplicaciones en este lenguaje.
- La metodología XP es rápida y menos laboriosa al momento de generar documentación se centra en el desarrollo rápido y menos complicado, es recomendada usarla para proyectos medianos y pequeños que no requieran extensa documentación.
- Se recomienda la instalación de servicios **PAAS** dentro de la universidad para brindar facilidad a los estudiantes de tener plataformas donde puedan desplegar aplicaciones o proyectos de investigación de manera más fácil y dentro de la propia universidad.
- El uso de servicios en la nube es una manera simple de tener acceso a una infraestructura que a veces es muy costosa o inalcanzable para pequeñas empresas pero a un costo reducido y usando solo lo que se necesita, por eso es una muy buena forma de tener un propio centro de datos o aplicaciones sobre el internet y con un costo menos que la implementación del mismo.

BIBLIOGRAFÍA

Aldunate, E., & Córdoba, J. (2011). *Formulación de programas con la metodología de marco lógico*. Santiago de Chile: Naciones Unidas.

Beck, K., & Andres, C. (2005). *Extreme Programming Explained*.

Belqasmi, F. G., & Fu, C. (2011). RESTful web services for service provisioning in next-generation networks: a survey. *Communications Magazine, IEEE*, 66-73.

Brown, E. (2014). *Web Development with Node and Express*. O'Reilly Media.

Camacho, H., Cámara, L., Cascante, R., & Sainz, H. (s.f.). *El Enfoque del marco lógico*. CIDEAL.

Chodorow, K. (2013). *MongoDB: The Definitive Guide*. O'Reilly Media.

Clements, D. M. (2014). *Node Cookbook*. Packt Publishing.

CodeSchool. (2015). *codeschool*. Obtenido de <http://campus.codeschool.com/courses/shaping-up-with-angular-js>

Eric, R., & Jim R., W. (2012). *Seven Databases in seven weeks*. Pragmatic Programmers.

Fielding, R., & Reschke, J. (s.f.). *Engineering Task Force (IETF)* . Obtenido de <https://tools.ietf.org/html/rfc7230>

FORMULARIO PARA PRESENTACIÓN DE PROYECTOS DE I+D. (2013). 13.

- Gauchat, J. D. (2012). *El Gran Libro de HTML5, CSS3 y Javascript*. Barcelona: marcombo.
- GoogleInc. (2014). *angularjs*. Obtenido de <https://angularjs.org>
- Gourley, D., & Totty, B. (2002). *HTTP The Definitive Guide*. O'Reilly.
- Joskowicz, J. (2008). *Reglas y Prácticas en eXtreme Programming*.
- Joyanes, A. L. (2010). Computación en Nube (Cloud Computing) y Centros de Datos: la nueva revolución industrial. ¿Cómo cambiará el trabajo en organizaciones y empresas? *Sociedad y Utopía: Revista de Ciencias Sociales*, 36.
- Joyent. (2014). *nodejs.org*. Obtenido de <http://nodejs.org/>
- mongodb. (2014). *mongodb*. Obtenido de <http://docs.mongodb.org/manual>
- Mongoose. (2011). *mongoosejs*. Obtenido de <http://mongoosejs.com/>
- Nunn, B. (2014). *Building a Responsive Website with Bootstrap*. Packt Publishing.
- Ortegón, E., Pacheco, J., & Prieto, A. (2005). Metodología del marco lógico. 124.
- Pollack, G., & Souza, C. (2015). *Code School*. Obtenido de <https://www.codeschool.com/courses/real-time-web-with-node-js>
- Posso Yépez, M. Á. (2011). *Proyectos, tesis y marco lógico : planes e informes de investigación*. Noción.
- Pousty, S., & Miller, K. (2014). *Getting Started with OpenShift*. O'Reilly Media.

Rodriguez, A. (2008). <http://www.ibm.com/>. Obtenido de <http://www.ibm.com/developerworks/library/ws-restful/>

Sanchez, N. (2007). El marco lógico. Metodología para la planificación, seguimiento y evaluación de proyectos. *Visión Gerencial*.

Shore, J., & Warden, S. (2007). *The art of agile development*. O'Reilly Media.

Shyam, S., & Brad, G. (2014). *AngularJS: Up and Running*. O'Reilly Media.

W3C. (2015). [w3schools](http://www.w3schools.com/angular/angular_intro.asp). Obtenido de http://www.w3schools.com/angular/angular_intro.asp

Webber, J., Parastatidis, S., & Robinson, I. (2010). *REST in Practice*. O'Reilly .

ANEXOS

- Manual de usuario
- Manual Técnico
- Glosario de términos
- Documentación del sistema