

## RAE

1. **TIPO DE DOCUMENTO:** Trabajo de grado para optar por el título de INGENIERO DE SISTEMAS.
2. **TITULO:** APLICATIVO WEB PARA EL APOYO BÁSICO AL CONTROL DE PROYECTOS DE TECNOLOGÍAS DE INFORMACIÓN BASADO EN LAS PRÁCTICAS DEL CUERPO DE CONOCIMIENTO PARA GESTION DE PROYECTOS (PMBOK).
3. **AUTOR:** Sergio Enrique Franco Portilla.
4. **LUGAR:** Bogotá D.C.
5. **FECHA:** Octubre 2016.
6. **PALABRAS CLAVE:** Gestión de proyectos, Desarrollo, Software, PMBOK, ASP.Net, C#, Diseño, Implementación, Control de tareas, MVC, Modelo, Vista, Controlador, SQL.
7. **DESCRIPCIÓN DEL TRABAJO:** El objetivo principal de este proyecto es el diseño de un aplicativo web que sirva como herramienta de apoyo básico de control de proyectos relacionados con Tecnologías de Información; también se presentan aspectos tales como el marco teórico, elección de metodologías de desarrollo, selección de servidores para montar la aplicación en la nube y diagramas UML con la información sobre requerimientos de la aplicación.
8. **LINEAS DE INVESTIGACION:** Tecnologías de Información y Comunicación (TIC),
9. **METODOLOGÍA:** Empírico Analítica, tomando como base los principios consignados en el cuerpo de conocimiento para la gestión de proyectos.
10. **CONCLUSIONES:** Realizar un proyecto implica un gran esfuerzo por parte de todas las personas involucradas en este y por ello es necesaria la existencia de herramientas que faciliten a estas personas un control de este por lo cual requieren aplicativos con interfaces simples con las cuales puedan acceder de una manera rápida a los datos que necesitan y con ello tomar decisiones, anotar sucesos importantes o anomalías que se presenten durante las etapas de estos y así mismo tomar decisiones y aprender lecciones para un futuro.

APLICATIVO WEB PARA EL APOYO BÁSICO AL CONTROL DE PROYECTOS  
DE TECNOLOGIAS DE INFORMACIÓN BASADO EN LAS PRÁCTICAS DEL  
CUERPO DE CONOCIMIENTO PARA GESTION DE PROYECTOS (PMBOK)

SERGIO ENRIQUE FRANCO PORTILLA

UNIVERSIDAD DE SAN BUENAVENTURA  
FACULTAD DE CIENCIAS BASICAS E INGENIERIA  
INGENIERIA DE SISTEMAS

BOGOTA

2016

APLICATIVO WEB PARA EL APOYO BÁSICO AL CONTROL DE PROYECTOS  
DE TECNOLOGIAS DE INFORMACIÓN BASADO EN LAS PRÁCTICAS DEL  
CUERPO DE CONOCIMIENTO PARA GESTION DE PROYECTOS (PMBOK)

SERGIO ENRIQUE FRANCO PORTILLA

Trabajo de grado

UNIVERSIDAD DE SAN BUENAVENTURA  
FACULTAD DE CIENCIAS BASICAS E INGENIERIA  
INGENIERIA DE SISTEMAS

BOGOTA

2016

## CONTENIDO

CONTENIDO.....	4
DEDICATORIA .....	10
AGRADECIMIENTOS .....	11
INTRODUCCIÓN.....	12
1. PLANTEAMIENTO .....	13
1.1. ANTECEDENTES .....	13
1.2. FORMULACION DEL PROBLEMA .....	15
1.3. JUSTIFICACION .....	18
1.4. OBJETIVOS.....	19
1.4.1. Objetivo General.....	19
1.4.2. Objetivos Específicos .....	19
1.5. ALCANCES Y LIMITACIONES .....	20
1.5.1. ALCANCES .....	20
1.5.2. LIMITACIONES .....	20
2. MARCO DE REFERENCIA.....	21
2.1. GESTION DE PROYECTOS.....	21
2.1.1. Características de un proyecto .....	21
2.2. PROYECTOS DE TECNOLOGIAS DE INFORMACION .....	22
2.3. GERENCIA DE PROYECTOS DE TECNOLOGIAS DE INFORMACION.....	24
2.4. PMBOK .....	26
2.4.1. Áreas del Conocimiento del PMBOK .....	27
2.4.2. Fases de un Proyecto.....	29
2.5. TAREAS .....	29
2.6. INCIDENCIAS EN UN PROYECTO .....	30
2.7. RECURSOS DE UN PROYECTO.....	31
2.7.1. Recursos Humanos.....	31
2.7.2. Recursos Físicos .....	31
2.7.3. Recursos Técnicos.....	32

2.7.4.	Recursos Financieros .....	32
2.8.	HITOS DE UN PROYECTO .....	32
2.9.	STAKEHOLDERS O INTERESADOS .....	33
2.10.	RIESGOS.....	34
2.11.	REQUERIMIENTOS.....	36
2.12.	CAMBIOS .....	37
2.13.	METODOLOGIAS DE DESARROLLO .....	37
2.13.1.	Metodologías Tradicionales de Desarrollo .....	37
2.13.2.	Metodologías Agiles de Desarrollo.....	42
2.13.3.	Comparación entre metodologías.....	46
2.14.	INGENIERIA WEB.....	51
2.14.1.	Estratos de la Ingeniería Web .....	53
2.14.2.	Proceso.....	53
2.14.3.	Métodos .....	54
2.14.4.	Herramientas y Tecnología.....	55
2.14.5.	Procesos de la Ingeniería Web .....	55
2.15.	TIPOS DE PRUEBAS.....	56
2.15.1.	APLICABILIDAD DE LAS PRUEBAS .....	57
2.16.	COMPARACION ENTRE SOFTWARE DE GESTION DE PROYECTOS.....	58
2.17.	REQUISITOS DE IMPLEMENTACION .....	62
2.18.	ANALISIS COSTO BENEFICIO.....	66
2.18.1.	Costos tangibles .....	66
2.18.2.	Costos Intangibles.....	68
2.18.3.	Beneficios Tangibles.....	68
2.18.4.	Análisis de Beneficios Intangibles .....	68
2.18.5.	Comparación entre costos y beneficios.....	69
2.18.6.	Tiempo de Amortización del proyecto.....	69
3.	METODOLOGIA .....	70
3.1.	LINEA DE INVESTIGACION .....	70
3.2.	ENFOQUE DE INVESTIGACION .....	70

3.3.	TECNICAS DE RECOLECCION DE INFORMACION .....	70
3.4.	CONSTRUCCION Y ANALISIS DE REQUERIMIENTOS .....	71
4.5.	PROTOTIPOS DE INTERFAZ GRAFICA.....	104
4.6.	DIAGRAMAS DE SECUENCIA.....	105
4.7.	DIAGRAMA DE ROBUSTEZ.....	116
4.8.	DIAGRAMA ENTIDAD RELACION Y BASE DE DATOS.....	119
4.9.	INTERFACES DE LA APLICACIÓN .....	122
3.11.1.	Inicio de Sesión .....	122
3.11.2.	Registrar un usuario.....	123
3.11.3.	Interfaz de inicio .....	123
3.11.4.	Creación de un Proyecto.....	124
3.11.5.	Editar Información del proyecto .....	124
3.11.6.	Información Detallada del proyecto .....	125
3.11.7.	Eliminar Proyecto .....	125
3.11.8.	Componentes del proyecto.....	126
3.11.9.	Tareas .....	126
3.11.10.	Crear Tarea .....	127
3.11.11.	Editar Tarea.....	127
3.11.12.	Detalles de Tarea .....	128
3.11.13.	Borrar Tarea .....	128
3.11.14.	Visualizar y Descargar Archivos Adjuntos.....	129
3.11.15.	Visualización de Tareas Antecesoras .....	130
3.11.16.	Agregar Tarea Antecesora.....	131
3.11.17.	Visualizar interesados del proyecto.....	131
3.11.18.	Crear Interesado para un Proyecto .....	132
3.11.19.	Editar Datos de un interesado.....	132
3.11.20.	Detalles de Interesado.....	133
3.11.21.	Eliminar Interesado .....	133
3.11.22.	Visualización de Hitos.....	134
3.11.23.	Creación de Hitos.....	134

3.11.24.	Editar Hito .....	135
3.11.25.	Detalles de hito .....	135
3.11.26.	Eliminar Hito .....	136
3.11.27.	Visualizar Incidencias .....	136
3.11.28.	Crear Incidencia .....	137
3.11.29.	Editar Incidencia .....	137
3.11.30.	Borrar Incidencia .....	138
3.11.31.	Compartir Proyecto .....	138
3.11.32.	Crear un Recurso .....	139
3.11.33.	Editar un Recurso.....	139
3.11.34.	Eliminar Un Recurso.....	140
3.11.35.	Ingresar Requerimiento.....	140
3.11.36.	Asignar Riesgo .....	141
1.11.	PATRÓN DE ARQUITECTURA .....	141
1.12.	HERRAMIENTAS .....	142
1.13.	TIPO DE APLICACIÓN.....	143
1.13.1.	Servicios a Ofrecer .....	144
2.	CONCLUSIONES .....	145
3.	BIBLIOGRAFIA E INFOGRAFIA.....	147
5.1.	BIBLIOGRAFIA.....	147

## LISTADO DE ILUSTRACIONES

Ilustración 1 Áreas del Conocimiento.....	27
Ilustración 2 Áreas del conocimiento y las fases de un proyecto .....	28
Ilustración 3 Prototipo Inicio de Sesión .....	104
Ilustración 4 Prototipo Registro .....	104
Ilustración 5 Prototipo Ver Proyectos.....	104
Ilustración 6 Prototipo Ventana de Componentes del Proyecto .....	105
Ilustración 7 Prototipo de ventana con paneles .....	105
Ilustración 8 Diagrama de Secuencia para usuarios y Proyectos .....	106
Ilustración 9 Diagrama de secuencia para tareas .....	107
Ilustración 10 Diagrama de secuencia para Hitos.....	108
Ilustración 11 Diagrama de Secuencia para Recursos .....	109
Ilustración 12 Diagrama de Secuencia para Incidencias .....	110
Ilustración 13 Diagrama de Secuencia para Archivos.....	111
Ilustración 14 Diagrama de Secuencia para interesados.....	112
Ilustración 15 Diagrama de secuencia para Requerimientos.....	113
Ilustración 16 Diagrama de secuencia para cambios.....	114
Ilustración 17 Diagrama de secuencia para riesgos .....	115
Ilustración 18 Diagrama de Robustez para Sesión .....	116
Ilustración 19 Diagrama de Robustez para Proyecto.....	116
Ilustración 20 Diagrama de robustez para Tareas .....	117
Ilustración 21 Diagrama de robustez para Incidencias .....	117
Ilustración 22 Diagrama de Robustez para Interesados .....	118
Ilustración 23 Diagrama de robustez para Recursos .....	118
Ilustración 24 Diagrama de robustez para Archivos.....	119
Ilustración 25 Diagrama Entidad-Relación inicial.....	119
Ilustración 26 Relaciones de tabla usuarios.....	121
Ilustración 27 Relaciones Tabla Proyecto.....	121
Ilustración 28 Relaciones tabla tareas .....	122
Ilustración 29 Vista de Inicio de Sesión .....	122
Ilustración 30 Registrar Usuario.....	123
Ilustración 31 Interfaz de inicio para el usuario .....	123
Ilustración 32 Crear Nuevo Proyecto .....	124
Ilustración 33 Editar Proyecto .....	124
Ilustración 34 Ver Detalles de Proyecto .....	125
Ilustración 35 Eliminar Proyecto.....	125
Ilustración 36 Vista con los componentes del proyecto .....	126
Ilustración 37 Ver Tareas.....	126
Ilustración 38 Crear Tarea .....	127
Ilustración 39 Editar Tarea .....	127
Ilustración 40 Detalles de Tarea .....	128
Ilustración 41 Borrar Tarea .....	128
Ilustración 42 Visualizar Archivos .....	129



Ilustración 43 Eliminar Archivo .....	129
Ilustración 44 Agregar Archivo .....	130
Ilustración 45 Tareas Antecesoras.....	130
Ilustración 46 Crear Tarea Antecesora .....	131
Ilustración 47 Ver Interesados de Proyecto .....	131
Ilustración 48 Crear Interesado.....	132
Ilustración 49 Editar Interesado .....	132
Ilustración 50 Ver Detalles de Interesado .....	133
Ilustración 51 Eliminar Interesado.....	133
Ilustración 52 ver Hitos .....	134
Ilustración 53 Crear Hito .....	134
Ilustración 54 Editar Hito.....	135
Ilustración 55 Ver Detalles de Hito.....	135
Ilustración 56 Eliminar Hito .....	136
Ilustración 57 Ver Incidencias .....	136
Ilustración 58 Crear Incidencia.....	137
Ilustración 59 Editar Incidencia .....	137
Ilustración 60 Borrar Incidencia.....	138
Ilustración 61 Compartir Proyecto.....	138
Ilustración 62 Crear Recurso .....	139
Ilustración 63 Editar Recurso .....	139
Ilustración 64 Ingresar Requerimiento .....	140
Ilustración 65 Creación de Riesgo .....	141
Ilustración 66 Patrón de Arquitectura MVC.....	142

## **DEDICATORIA**

El autor dedica este proyecto a las siguientes personas:

A mis padres Enrique Franco y Aleyda Portilla, quienes han sacrificado y dado todo para formar una familia y darnos educación.

A mis hermanos Jair y Johana quienes se tomaron el tiempo para ayudarme y criarme a lo largo de los años.

A Laura María Camargo, quien ha estado allí en los buenos y malos momentos.

## **AGRADECIMIENTOS**

El autor agradece a las siguientes personas:

A Álvaro Gamboa, quien ha participado activamente en la revisión y aspectos técnicos este proyecto.

A Gloria Amparo Contreras, quien ha estado atenta a cualquier cosa que ha surgido en el proyecto.

A mis compañeros de Antss: Diana, Carlos, Juan, Diego y Arthur; quienes me han permitido crecer como desarrollador y obtener una amplia gama de conocimientos que me permitieron llegar a este punto.

A Hector Enriquez, quien me brindó una oportunidad única que me ha llevado lejos.

## INTRODUCCIÓN

La finalidad de este proyecto es mostrar el proceso de desarrollo de un aplicativo web que pueda servir de apoyo a la gestión de proyectos en el área de tecnologías de información tomando en cuenta varios principios existentes en el Project Management Body of Knowledge o PMBOK. De igual forma, tiene como objetivo la entrega de informes sencillos, los cuales permitan a los encargados de un proyecto tomar decisiones que puedan generar cambios en el desarrollo de estos.

En el primer capítulo se establecen los objetivos los cuales servirán para verificar si el aplicativo cumple con las características necesarias para ser usado en un proyecto; posteriormente se expondrá una justificación de este junto con sus alcances y limitaciones.

El segundo capítulo se expone el marco teórico el cual servirá para que el lector pueda tener una referencia de lo que se pretende solucionar con el desarrollo del proyecto y de los conceptos empleados para determinar aspectos importantes para la etapa de desarrollo.

El tercer capítulo mostrara al lector el componente técnico de la aplicación, es decir, cuales son los tipos de aplicaciones que existen, tipos de pruebas, entre otros conceptos.

El cuarto capítulo tiene como finalidad exponer la metodología de desarrollo, el enfoque de la investigación y la propuesta de desarrollo que se ha elegido para realizar la aplicación.

Para finalizar, se expresan las conclusiones a las cuales se la llegado con el desarrollo del documento, la aplicación y los anexos, en caso de que sean necesarios.

## 1. PLANTEAMIENTO

### 1.1. ANTECEDENTES

Dentro de todas las organizaciones, existen áreas de Tecnologías de la Información, las cuales manejan proyectos que comprometen a una empresa y a quienes interactúan con ella, ya sean interesados o patrocinadores; estos requieren que dichos proyectos se ejecuten y se controlen correctamente. Para esto se han creado estándares aceptados mundialmente por empresas y organizaciones como IBM Corporation, British Petroleum, SAP, Siemens, El Departamento de Defensa de Estados Unidos, Pricewaterhouse Coopers, Sprint, entre otras<sup>1</sup>. Capuz Rizo, Gomez-Senent, Torrealba, Ferret, Gomez y Vivanco brindan la siguiente definición sobre lo que es la gestión de dichos proyectos: La gestión de proyectos tiene como misión establecer los objetivos del proyecto, definir la metodología a seguir en su utilización, planificar y programar tareas y recursos, corregir desviaciones y comunicar progresos y resultados.<sup>2</sup>

Es claro que todas las organizaciones planifican proyectos, y es necesario que estos tengan un seguimiento en todas sus fases desde el inicio del proyecto hasta su culminación y también se requiere un seguimiento de las tareas establecidas para su éxito.<sup>3</sup> Para dicho control se han generado aplicaciones por parte de diversas empresas de desarrollo de software; dichos aplicativos facilitan dicho seguimiento y la verificación del cumplimiento de los objetivos del mismo.

Existen diversas herramientas enfocadas a la gestión de proyectos, una de estas es GanttPV, en su página web se puede observar que es un programa basado

---

<sup>1</sup> (Rozemeijer & Wellington Redwood, Frameworks for IT Management-A Pocket Guide. Wilco: Van Haren Publishing, 2007)

<sup>2</sup> (Capuz Rizo, Gomez-Senent Martinez , Torrealba Lopez, Ferret Gisbert, Gomez Navarro, & Vivancos Bono, Cuadernos de Ingenieria de Proyectos III: Direccion, Gestion y Organizacion de Proyectos. Valencia: Servicio de Publicaciones Universidad Politecnica de Valencia, 2000)

<sup>3</sup> (Project Management Institute, Guia de los Fundamentos para la Direccion de Proyectos (Guia del PMBOK) - Cuarta Edicion. Newton Square: Project Management Institute, 2008)

en diagramas de Gantt para el seguimiento de tareas, la ventaja de este programa a pesar de su simplicidad es el soporte multiplataforma<sup>4</sup>; Otra herramienta existente es Clocking IT, la cual es gratuita y tiene como característica principal la generación de diagramas de Gantt interactivos, disponible en varios idiomas, una interfaz simple y flexibilidad en la generación de reportes<sup>5</sup>.

Otra solución informática existente es OpenProject, diseñado por Serena Software Incorporated, este software es una versión gratuita y de código abierto de Microsoft Office Project y ofrece entre sus ventajas el establecimiento de metas, manejo de documentos, establecimiento y recordatorio de metas y fechas importantes, entre otros.<sup>6</sup>

Una aplicación de pago para la gestión de proyectos es Basecamp, es una aplicación que se ejecuta a través de la web y ofrece gestión de recursos y documentos, aunque es de pago también existe una versión gratuita con la limitante que no se puede compartir archivos<sup>7</sup>.

Al observar las aplicaciones descritas anteriormente, junto con sus características y ventajas encontradas dentro de la página web de cada aplicación, se encuentra que los beneficios que se pueden obtener de estas se limitan solo a la realización de diagramas de Gantt, gestión de documentos y establecimiento de metas; por lo cual se hace necesaria la creación de un aplicativo que le permita al usuario tener un apoyo a la gestión de sus proyectos donde pueda tener el control de diversos componentes del proyecto y no únicamente de las tareas e hitos que aunque son parte vital de un proceso también es necesario tomar en cuenta otras entidades y así se tomen decisiones de una manera simple y efectiva.

---

<sup>4</sup> (Christensen, GanttPV, Internet : <<http://www.pureviolet.net/ganttpv/index.html>> , 2009)

<sup>5</sup> (Simonsen & Simonsen, Clocking IT, Internet: <<http://www.clockingit.com/>> , 2003 - 2008)

<sup>6</sup> (Open Project Foundation, Open Project, Internet: <<https://www.openproject.org/>> , 2013)

<sup>7</sup> (Signals LLC, Basecamp, Internet: <<https://basecamp.com/>> ,1999-2013)

## 1.2. FORMULACION DEL PROBLEMA

Dentro de la gestión de proyectos es necesario encontrar alternativas que estén a la medida de los distintos usuarios, empresas y tipos de proyectos que deseen gestionar, por ello, deben existir herramientas que permitan apoyar y tener un control sobre los proyectos; a su vez estas aplicaciones deben ser eficientes y permitirle a quienes interactúan con estas un manejo intuitivo.

Se encuentra la necesidad de desarrollar aplicaciones que permiten a las organizaciones mejorar su desempeño, productividad, competitividad, manejo de tiempos y presupuesto; a su vez que se busca que estas aplicaciones sean de bajo costo o de libre distribución, para no generar problemas en las empresas, esto se debe a que las soluciones existentes están licenciadas y tiene un precio elevado, lo que hace que las organizaciones desistan de su adquisición.

Una de las guías que contiene los conocimientos y prácticas para la gestión de proyectos es el Project Management Body of Knowledge o PMBOK por sus siglas, el cual contiene una serie de fundamentos para la aplicación de buenas prácticas en la ejecución y seguimiento de proyectos<sup>8</sup>; puede ser usado para realizar la gestión de cualquier tipo y cualquier tamaño de proyecto, dicho libro de buenas prácticas se encuentra traducido en los idiomas más importantes: Español, Inglés, Portugués, entre otros. El PMBOK se basa en diversas áreas del conocimiento, las cuales, a su vez aplican varios procesos en cada una de las fases del proyecto, lo que garantiza que cada fase salga lo mejor posible.<sup>9</sup>

A pesar de que un proyecto disponga de una serie de prácticas que permitan una buena ejecución para estos, pueden seguir presentándose problemas, muchos de estos pueden ocurrir por la falta de control o apoyo existente para determinados aspectos del proyecto.

En un estudio realizado por la consultora McKinsey&Company en el año 2012, se obtuvieron datos de aproximadamente 5400 proyectos de TI; muestra que los principales motivos por los cuales tienden a fallar dichos proyectos, radican en que se excede el presupuesto que se tenía planeado para su ejecución o

---

<sup>8</sup> (Project Management Institute, Guia de los Fundamentos para la Direccion de Proyectos (Guia del PMBOK) - Cuarta Edicion. Newton Square: Project Management Institute. 2008)

<sup>9</sup> (Rozemeijer & Wellington, Frameworks for IT Management - A Pocket Guide. Wilco: Van Haren Publishing, 2007)

existen retrasos en las entregas y en los productos, lo que causa que no se entreguen a tiempo, o el resultado final del proyecto termina siendo diferente a lo esperado por los interesados<sup>10</sup>.

Jospeh Gulla afirma que los proyectos suelen fallar por falta de planeación y dirección de los mismos; para estos casos, Gulla considera que el uso de herramientas es importante para el control y solución de dichos problemas; la primera de estas fallas radica en la ausencia de comunicación con los patrocinadores y con los interesados en el proyecto, por lo cual es necesario planificar encuentros y contenidos para cada encuentro. El siguiente punto en la lista consiste en un continuo cambio en los objetivos, no identificar correctamente ni tener presente tiempos de inicio y fin del proyecto y no tener contemplados los riesgos a los que se encuentra expuesto el mismo; también se encuentran como puntos importantes la carencia de metodologías y herramientas, el no involucramiento de los ejecutivos en un proyecto y por último, la falta de adaptación<sup>11</sup>.

La empresa de consultoría Gartner realizó un estudio a más de 150 organizaciones y sus proyectos relacionados con Tecnologías de Información. En este estudio tomaron empresas y proyectos de pequeño, mediano y gran tamaño, encontrando que, entre mayor fuera el tamaño de la organización y mayor fuera el presupuesto que tenían asignado, su tendencia al fracaso se incrementaba. En ese mismo estudio se solicitó que englobaran los proyectos por las siguientes categorías: problemas de funcionalidad, tardanza, problemas de calidad, alto costo, cancelación antes de su lanzamiento y otras razones; se encontró que los porcentajes similares en los 3 tipos de empresa eran por problemas de tiempo y funcionalidad, tal como lo indica la siguiente gráfica<sup>12</sup>.

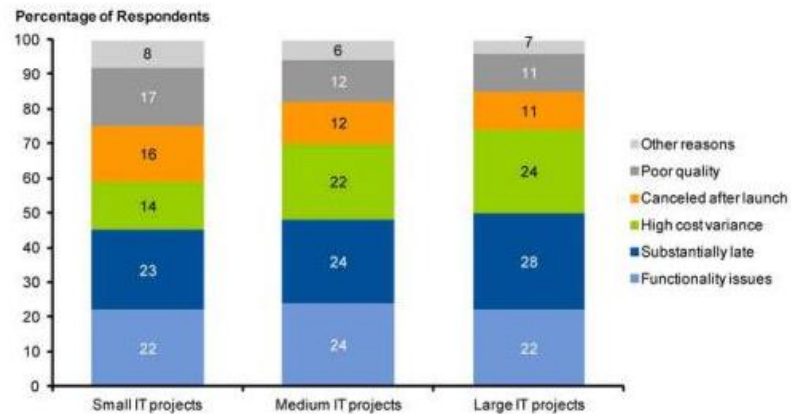
---

<sup>10</sup> (Bloch, Blumberg, & Laartz, Insights & Publications, Internet: < [http://www.mckinsey.com/insights/business\\_technology/delivering\\_large-scale\\_it\\_projects\\_on\\_time\\_on\\_budget\\_and\\_on\\_value](http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value)> 2012)

<sup>11</sup> (Gulla, 2012) Seven Reasons IT Projects Fail, Internet: < [http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project\\_pitfalls/?page=3](http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/?page=3)> , 2012)

<sup>12</sup> (Mieritz, Gartner Survey Shows Why Projects Fail, Internet : < <http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/> > 2012)





**Grafico 1: Relación entre Tamaño de Empresa y Fallos en el Proyecto**

Al observar los datos encontrados en los estudios anteriormente citados se puede encontrar que las fallas principales a la hora de gestionar un proyecto radican en la falla de comunicación entre los distintos actores del proyecto, no tener un control de tiempos para la planeación, realización y aplicación del mismo, ni un control del presupuesto a manejar dentro de este. Así, surge la siguiente pregunta de investigación:

¿Cómo desarrollar una aplicación web que sirva como apoyo básico al control de proyectos en el área de Tecnologías de Información?

### **1.3. JUSTIFICACION**

Un proyecto de Tecnologías de Información puede fallar por diversos motivos, estos suelen terminar en consecuencias fatales para una organización; casos tales como demoras en los tiempos de ejecución, entregas de resultados equivocados o exceder el presupuesto acordado para la ejecución de estos; también se encuentran casos en los que un proyecto fracasa porque el interesado se enfoca en lo que quiere y no en lo que necesita ni tampoco evalúa el modelo de negocio, incluso es necesario establecer roles, criterios y procesos que puedan ser monitoreados y evaluados para que tengan éxito; en otros casos los proyectos no prosperan porque no existe una buena comunicación entre el interesado, los patrocinadores y el equipo de trabajo.

Para un gerente de proyectos es necesario tener una aplicación descentralizada para acceder a la información relacionada con este y tomar decisiones importantes que apoyen la gestión de tareas u otros componentes; esto tomando en cuenta que puede hacerlo en cualquier momento y lugar mediante un dispositivo y una conexión a internet para acceder a la aplicación.

Por estos motivos es necesario el desarrollo de un aplicativo que permita a sus usuarios un apoyo y control a los proyectos que estén a cargo o sean parte.

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

Diseñar un aplicativo web que sirva como herramienta de apoyo básico de control de proyectos relacionados con Tecnologías de Información.

### **1.4.2. Objetivos Específicos**

- Entregar información que sirva de apoyo para los usuarios al momento de tomar decisiones en un proyecto.
- Identificar los diferentes inconvenientes presentados en el proyecto a evaluar.
- Diseñar una interfaz de usuario que permita manejar la aplicación de manera fácil y rápida.

## **1.5. ALCANCES Y LIMITACIONES**

### **1.5.1. ALCANCES**

- Documento escrito con la información necesaria del proyecto.
- Aplicación que será alojada en un servidor, que cumpla con los objetivos planteados en la sección anterior y con los requerimientos que se plantean en el respectivo apartado de este escrito.
- Esta aplicación solo mostrará información, es decir no generara gráficos o diagramas de ninguna clase.

### **1.5.2. LIMITACIONES**

Al escoger C# y ASP.NET como lenguajes de programación principales para el desarrollo de la aplicación, es necesario escoger una herramienta que permita desarrollar la aplicación; para este proyecto se eligió Visual Studio 2013, que tiene como limitación el tipo de licencia que maneja, ya que es un software propietario y en algunos casos es necesario pagar la licencia para su uso.

Por último se tiene como limitante las versiones del explorador en la cuales puede correr la aplicación considerando que versiones anteriores de los exploradores más usados como Google Chrome y Mozilla Firefox presentan problemas de compatibilidad con tecnologías modernas, por lo cual es necesario usar una versión actualizada de dichos programas.

## 2. MARCO DE REFERENCIA

### 2.1. GESTION DE PROYECTOS

Un proyecto según Brown Boveri, puede considerarse como un trabajo no repetitivo que ha de realizarse según unas especificaciones técnicas determinadas con unos costos, inversiones y plazos prefijados, también se define como un trabajo de volumen y complejidad que necesita realizarse entre varios departamentos de la empresa y muchas veces con ayuda de terceros<sup>13</sup>.

El concepto de gestión de proyectos viene del inglés, Project Management, y se define como una serie de técnicas y formas de organización que permite afrontar situaciones complejas que resultan difíciles de solucionar implementando sistemas de dirección clásicos y manteniendo las estructuras básicas funcionales que solo resultan adecuadas para tareas de tipo repetitivo.

#### 2.1.1. Características de un proyecto

- **Trascendencia:** es la característica principal de los proyectos, un proyecto es de gran tamaño, participan diversos departamentos de la empresa y tiene como objetivo final brindarle a la empresa consecuencias de tipo trascendental y trae repercusiones positivas o negativas a corto o largo plazo, es decir que si un proyecto carece de esta particularidad simplemente estaríamos hablando de una tarea rutinaria dentro de la empresa.
- **Utilización de Recursos:** un proyecto está encaminado un cambio para una empresa y por ende exige una inversión importante de tiempo, personal y dinero; a su vez exige una inversión en recursos físicos que son el resultado de realizar una inversión de tipo económico.

---

<sup>13</sup> (Dirección y Gestión de Empresas, Gestión de Proyectos. Málaga: Publicaciones Vertice S.L, 2008)

- **Discontinuidad:** Como ya se ha dicho, un proyecto es una actividad que se realiza de manera esporádica, a su vez cada proyecto es diferente en todo aspecto, desde su tamaño, la cantidad de recursos que se deben invertir, el área a la que está destinada, entre otros. Esto hace que se haga necesario tener mecanismos de gestión específicos.
- **Dinamismo:** Los proyectos al ser tareas poco usuales dentro de la empresa, incorporan nuevos recursos y se van retirando otro conforme avanzan las entregas y las fases.
- **Irreversibilidad:** Durante el ciclo de vida de un proyecto hay decisiones que afectan directamente el progreso del mismo, haciendo que ciertas tareas se conviertan en algo irreversible para la empresa o al menos generan repercusiones más grandes que las que generan las tareas ordinarias dentro de una organización.
- **Influencias Externas:** Los proyectos suelen estar influenciados bajo diversos factores externos, económico, político, social, dependiendo de la importancia de dichos factores, el proyecto tiene más o menos posibilidades de terminar en fracaso.

Si el proyecto es trascendental, su gestión resulta compleja, es inusual y está sometido a fuertes influencias externas, no es extraño que pueda finalizar en fracaso y más frecuentemente quedar lejos de cumplir los objetivos establecidos.

## 2.2. PROYECTOS DE TECNOLOGIAS DE INFORMACION

Las empresas buscan fortalecer su área de tecnologías de información, esto se debe a la frustración que poseen estas referente al progreso en esta area; por lo general los gerentes y ejecutivos del proyecto encuentran 4 problemas que se presentan a lo largo de un proyecto: el retraso en cuanto a los tiempos de entrega del producto, el resultado final no es el esperado por los interesados y patrocinadores, los costos se incrementan por encima de lo esperado y finalmente cuando se termina un proyecto se hace generando estrés a todos los implicados en el desarrollo de este.

Con el fin de evitar estos problemas, los gerentes de TI y las áreas de informática han adoptado prácticas relacionadas con la gerencia de proyectos que si bien ayudan a mitigar estos problemas, no hacen que desaparezcan por completo; A pesar de que se diga que dichas prácticas solo ha servido para añadir algo de papeleo y que no sirve de mucho adoptar estas costumbres, se ha mostrado que una buena gerencia cambia la manera en que las organizaciones pueden afrontar problemas de costos, alcances, comunicación y desarrollo durante la duración de estos.

Algunas organizaciones no han implementado o formalizado la gestión de proyectos de TI dentro de estas, teniendo como principales motivos el miedo a los controles, la falta de conocimiento o experiencia, no existe una organización, cultura o seguimiento de proyectos y se cree que la gerencia de un proyecto exige una gran inversión de tiempo.

Durante el desarrollo de un proyecto se presentan una serie de barreras que enfrentan los proyectos de Tecnologías de Información, una gerencia de proyectos bien llevada permite afrontar los siguientes obstáculos:

- Cambios constantes de requerimientos: este es uno de los problemas que más aqueja a un proyecto, es importante mantener un control de los cambios que sucedan a través del tiempo; esto puede solucionarse haciendo uso de una buena gestión de alcances que pueda asegurar el cumplimiento de requerimientos.
- Expectativas no razonables: este problema tiene que ver con la estimación de tiempos poco realistas lo cual genera falsas expectativas en la gerencia, lo cual obliga al equipo de trabajo a hacer las cosas de manera apresurada; para esto es bueno tener una disciplina de estimación que permita fijar metas viables.
- Falta de Información: Es uno de los problemas más comunes y entorpece los avances del proyecto, por lo que es importante conocer y discutir todos los aspectos de un proyecto con todas las personas que sean necesarias, revisar sus grados de responsabilidad, niveles de impacto y responsabilidad en este; es necesario que durante el desarrollo de un proyecto se tenga un flujo de información constante de todos los aspectos posibles, sean positivos o negativos. Una buena inversión de tiempo en reuniones y apoyo de la gerencia son importantes.

- Resistencia al cambio: Los proyectos relacionados con Tecnologías de Información implican varios cambios en una organización, eso implica que las personas cambie su modo de trabajar, lo cual puede desembocar en una resistencia al cambio si no se tiene el debido apoyo. Para sobrepasar ese obstáculo es necesario que se entienda que las organizaciones funcionan como un todo al igual que un proyecto, y este requiere de liderazgo y establecer objetivos claros que deben manifestarse en apoyo.
- Gerencia de Alcance: Cuando se analizan los fallos de un proyecto, se encuentra que generalmente el equipo no invirtió tiempo para definir los alcances de un proyecto de manera adecuada. El tener una gestión de alcances protege las expectativas del proyecto y si es necesario un cambio se realizaran los reajustes necesarios de tiempos, alcances y costos.

### **2.3. GERENCIA DE PROYECTOS DE TECNOLOGIAS DE INFORMACION**

Dentro de la gestión de un proyecto se involucran cuatro grupos de elementos: recursos humanos, recursos materiales, productos y calendarios; en el área de Tecnologías de Información estos elementos son observados desde el punto de vista del líder de proyecto, del gerente del departamento de sistemas, el gerente del área de tecnologías de información y el punto de vista de la empresa.

En este caso se puede definir la gestión de proyectos como el conjunto de disciplinas que se integran para apoyar eficaz y eficientemente los cuatro niveles de gestión.<sup>14</sup>

Lo primero que se debe analizar son los temas que preocupan a las organizaciones y departamentos de Tecnologías de Información de una empresa, es necesario tomar en cuenta que un gerente de TI debe preocuparse por el día a día de los proyectos y que clase de proyectos se pueden desarrollar para brindarle aportes positivos a la empresa.

---

<sup>14</sup> (Fabregas, Gerencia de Proyectos de Tecnología de Información. Caracas: Editorial CEC, 2005)



Las preocupaciones del día a día tienen tres interrogantes<sup>15</sup>:

- ¿Cómo estamos cumpliendo con el negocio? Es decir si se están cumpliendo los objetivos de la empresa, de qué manera se colabora y que proyectos le están aportando más al negocio.
- ¿Cómo estamos cumpliendo nuestro plan?: como se están cumpliendo los objetivos y como se desarrollan en ejecución del presupuesto
- ¿Cómo estamos cumpliendo los proyectos?: se debe mirar la manera en que avanza el desarrollo de cada producto y cuáles son sus niveles de cumplimiento respecto a fechas y recursos.

Igualmente las preocupaciones con vista al mañana se pueden resumir en tres interrogantes<sup>16</sup>:

- ¿Cómo alinearnos con los objetivos del negocio?: En este caso es necesario entender las direcciones del negocio para diseñar un conjunto de proyectos que apoyen la dirección y el modelo del negocio de una empresa para el cumplimiento de sus objetivos.
- ¿Cómo integrar nuestra cartera de proyectos?: una vez se determinen los proyectos de TI que existan en una organización, es el momento asignar los recursos humanos y materiales; estableciendo la fecha de inicio de cada uno.
- ¿Cómo estructurar un plan viable?: al tener los proyectos seleccionados se debe analizar la viabilidad de estos, identificando los diferentes factores de riesgos y establecer planes de contingencia y mitigación para manejar los riesgos y problemas que se llegasen a presentar.

---

<sup>15</sup> (Fabregas, 2005)

<sup>16</sup> (Fabregas, 2005)

Es necesario considerar la gestión de un proyecto por productos, ya que es bueno darle importancia a los productos, como deben ser desarrollados y su nivel de calidad<sup>17</sup>.

Según Fabregas los entregables también se pueden controlar con más facilidad que los intangibles, ya que las tareas de planificación y control de proyectos pueden simplificarse si se centran en los productos que pueden ser entregados en lugar de centrarlas en las actividades que deben ser cumplidas.

Cuando se planifica por productos en vez de tareas se simplifica la estimación ya que es más sencillo estimar el esfuerzo que implica desarrollar cada producto, consecuentemente es más fácil estimar la duración de las actividades y cada rol de los participantes. En el caso de los productos es más sencillo estimar el desarrollo de uno solo y no una tarea donde pueden ser desarrollados varios elementos de manera parcial al mismo tiempo.

El concepto de gerencia de proyectos por productos es un tema que se ha ido discutiendo recientemente, pero por el momento no existen aplicaciones centradas totalmente en este tipo de gestión, Office Project o PM sirven como herramientas de apoyo para gestión de proyectos pero ninguna posee los medios para generar un apoyo de una gestión por productos.

## **2.4. PMBOK**

Dentro de la gerencia de proyectos existen una serie de disciplinas que integran las actividades asociadas con las fases de un proyecto, para coordinar de manera eficiente los recursos y permitir que los objetivos establecidos sean cumplidos.

En una organización es difícil integrar todas las disciplinas existentes para la gestión de un proyecto, lo cual implica capacitación, documentación, adaptación, divulgación y a la vez incluir estándares y procedimientos para la estimación y formulación de estos.

---

<sup>17</sup> (Fabregas, 2005)

Una de estas guías que permiten el control y gestión de un proyecto es el PMBOK o Project Management Body of Knowledge, esta guía fue creada por PMI que es una organización sin ánimo de lucro que tiene vinculados a más de 125.000 profesionales en todo el mundo.

El PMI maneja un grupo de áreas del conocimiento en las cuales se encuentran un grupo de actividades y procesos de un proyecto.

### 2.4.1. Áreas del Conocimiento del PMBOK

La gerencia de proyectos puede concebirse como un conjunto de disciplinas que orientan las actitudes asociadas a la dirección de proyectos, el PMBOK profundiza en ese concepto y brinda una serie de áreas del conocimiento en la gestión de proyectos.

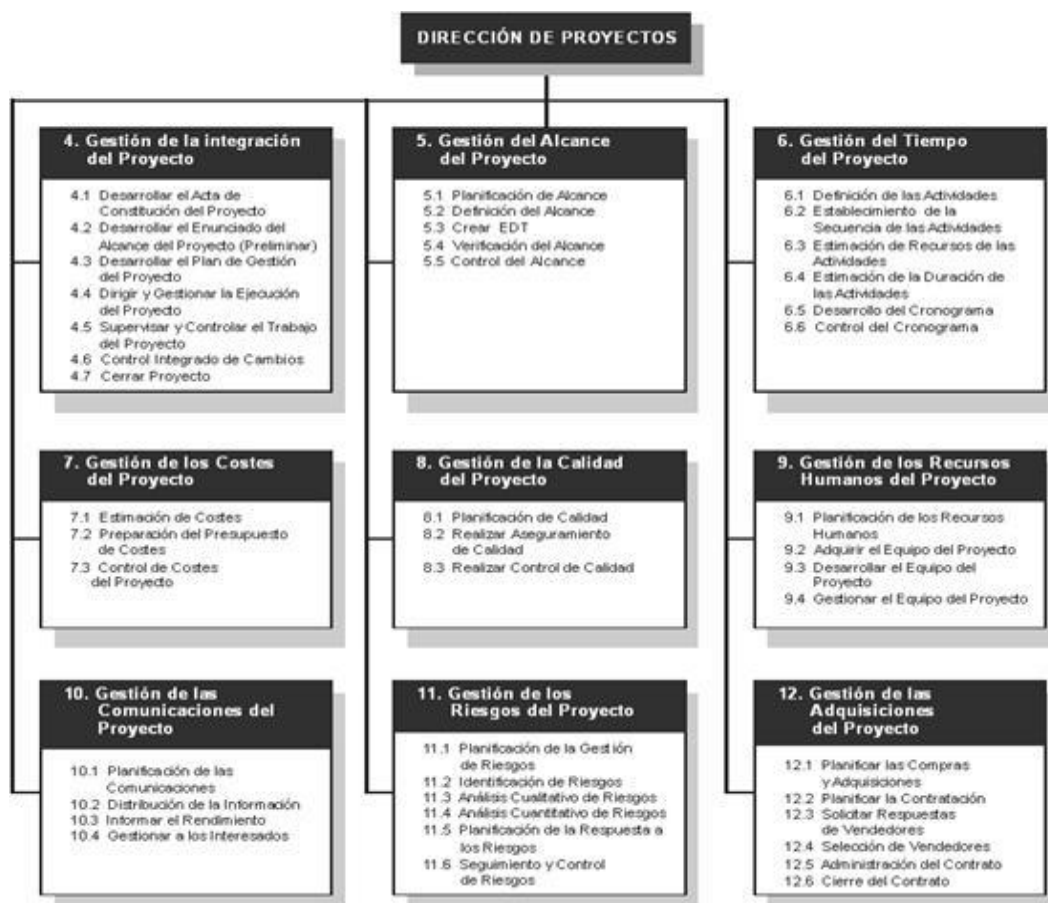


Ilustración 1 Áreas del Conocimiento

Para cada área del conocimiento existen una serie de tareas y entregables que permiten a las personas implicadas en el proyecto tener una mejor visión de este.

Dentro de estas áreas del conocimiento se pueden englobar ciertas tareas o entregables dentro de cada una de las fases de desarrollo de un proyecto, tal como ilustra la siguiente imagen:

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring & Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Execution	4.4 Monitor and Control Project Work 4.5 Perform Integrated Change Control	4.6 Close Project or Phase
5. Project Scope Management		5.1 Collect Requirements 5.2 Define Scope 5.3 Create WBS		5.4 Verify Scope 5.5 Control Scope	
6. Project Time Management		6.1 Define Activities 6.2 Sequence Activities 6.3 Estimate Activity Resources 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Estimate Costs 7.2 Determine Budget		7.3 Control Costs	
8. Project Quality Management		8.1 Plan Quality	8.2 Perform Quality Assurance	8.3 Perform Quality Control	
9. Project Human Resource Management		9.1 Develop Human Resource Plan	9.2 Acquire Project Team 9.3 Develop Project Team 9.4 Manage Project Team		
10. Project Communications Management	10.1 Identify Stakeholders	10.2 Plan Communications	10.3 Distribute Information 10.4 Manage Stakeholder Expectations	10.5 Report Performance	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses		11.6 Monitor and Control Risks	
12. Project Procurement Management		12.1 Plan Procurements	12.2 Conduct Procurements	12.3 Administer Procurements	12.4 Close Procurements

Ilustración 2 Áreas del conocimiento y las fases de un proyecto

## **2.4.2. Fases de un Proyecto**

Las fases de un proyecto son divisiones dentro del mismo proyecto, donde es necesario ejercer un control adicional para gestionar eficazmente la conclusión de un entregable mayor. Las fases de un proyecto suelen comportarse de manera secuencial, pero en determinadas situaciones de un proyecto pueden superponerse. Por su naturaleza de alto nivel, las fases de un proyecto constituyen un elemento del ciclo de vida de un proyecto<sup>18</sup>.

Existen cinco fases para el desarrollo de un proyecto, la fase de iniciación, la fase planeación, la fase de ejecución, la fase de seguimiento y control, y la fase de cierre; dentro del PMBOK y la gestión de proyectos en general, cada una de dichas fases cuenta con sus propios procedimientos para tener en cuenta al momento de proceder hacia otro nivel del proyecto, para cada uno de los procesos existen entradas y salidas claras para cada uno.

Generalmente cada fase del proyecto termina con una revisión de sus entregables, para determinar su compleción y su aceptación, cada revisión significa el fin de una fase y el comienzo de otra y es un buen momento para revisar si se puede continuar con el proyecto, que cambios deben efectuarse de manera inmediata o si es necesario replantear el proyecto completamente.

## **2.5. TAREAS**

Una tarea es una actividad que es asignada a una persona dentro del proyecto o responsable; dicha actividad debe tener un tiempo específico en el que debe ser realizada que consiste de una fecha de inicio y una fecha de finalización. Las tareas pueden tener otras como antecesoras a esa es decir que esa tarea no puede empezar hasta que las otras de las que esta depende no finalicen correctamente por lo cual puede afectar el tiempo de desarrollo del proyecto y generar demoras en caso de que no finalicen a tiempo.

---

<sup>18</sup> (Project Management Institute, Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK) - Cuarta Edición. Newton Square: Project Management Institute. 2008)

Las tareas pueden tener recursos asignados que son utilizados para que esta sea completada; estas también pueden tener como producto final un documento que puede servir como entrada a otra tarea. Es una buena práctica desglosar el proyecto en la mayor cantidad de tareas posibles, ya que esto permite a todo el equipo darse cuenta de varias cosas que pueden pasar desapercibidas al dejar tareas que a su vez estén compuestas de otras.

## **2.6. INCIDENCIAS EN UN PROYECTO**

La gestión de incidencias en un proyecto puede ser la clave para el éxito o el fracaso de un proyecto. Una incidencia se define como todos los hechos que son anómalos o inesperados durante la ejecución de las tareas que corresponden a un proyecto y que producen desviaciones en la planificación; se pueden tomar como ejemplos los retrasos en las entregas de un software, enfermedades, entre otras.

Es importante que se conozcan las tareas que se ven afectadas por esta clase de sucesos con el fin de realizar una evaluación de costos de la misma; para ello es necesario un estudio cuidadoso de todas las actividades que puedan verse implicadas de manera directa o indirecta.

El tipo de solución que se deba implementar para solucionar un suceso anómalo puede ser planteada por el jefe de proyectos, este puede proponer las soluciones que crean necesarias, como por ejemplo incorporar más personas al proyecto, asumir los retrasos acontecidos, realizar tareas en un plazo de tiempo menor al previsto, etc.

De acuerdo a estos sucesos encontrados se deberán reajustar los tiempos y revisar la planificación del proyecto.

El registro de una incidencia tiene como objetivos el tener en cuenta aquellos hechos que afectan de manera negativa el progreso de un proyecto para que la aparición de estos sucesos sea cada vez menor o no vuelva a producirse y también tiene como finalidad que las personas que correspondientes o causantes de estas se les asuman los costos adquiridos.

Cuando se registra una de estos sucesos anómalos, se realiza esta acción para dejar constancia de todos los hechos que han dificultado el desarrollo normal de un proyecto, esto con la finalidad de documentar los retrasos y las causas de estos y permitir que aquellas personas encargadas de tomar medidas correctivas lo hagan y puedan disminuir o evitar estos sucesos anómalos.

Para esto se crea un registro de estos actos que permite un acceso rápido a todos los hechos que han afectado al proyecto.<sup>19</sup>

## **2.7. RECURSOS DE UN PROYECTO**

Los recursos para un proyecto, son un componente vital para el desarrollo de una tarea ya que permiten el desarrollo de estas, aumentando la productividad y cumpliendo los objetivos; estos recursos se pueden dividir en los siguientes:

### **2.7.1. Recursos Humanos**

Es el tipo de recurso más importante, prácticamente de este depende el éxito o el fracaso de un proyecto, esto exige que la selección de estos recursos sea la más rigurosa posible, estos se caracterizan por tener potencial de desarrollo, ideas, creatividad, habilidades, sentimientos, experiencias, conocimientos, entre otros.

### **2.7.2. Recursos Físicos**

Este recurso también es importante, ya que es necesaria la existencia de objetos tangibles que puedan ser usados por el personal para el desarrollo del proyecto; ejemplos de estos serían los edificios,

---

<sup>19</sup>(Gestion de Proyectos,Ministerio de Administraciones Publicas, 2015)

bodegas, almacenes, maquinas, muebles, insumos, materiales, entre otros. Es aconsejable que estos recursos estén inventariados al momento de su uso; el manejo de estos recursos también debe estar acorde a los manuales de procesos que maneje la empresa o a las políticas internas; igualmente deben mantener al día la documentación y las garantías de estos.

### **2.7.3. Recursos Técnicos**

Están relacionados con las tecnologías que se usan o se necesitan adquirir para el funcionamiento del proyecto, en pocas palabras estos recursos engloban todo el conocimiento que se pueda adquirir o sea necesario para el desarrollo del proyecto.

### **2.7.4. Recursos Financieros**

Hace referencia a los recursos monetarios que deben ser usados o invertidos y han sido presupuestados para la financiación del proyecto, siempre teniendo en cuenta la viabilidad del proyecto<sup>20</sup>.

## **2.8. HITOS DE UN PROYECTO**

En el momento de planificar un proyecto es necesario establecer los hitos o puntos finales de una actividad, en estos debe existir una salida formal como un entregable que debe ser presentado; este entregable no debe ser un documento extenso, en su lugar debe ser un documento corto donde se expliquen los logros de las actividades; estos hitos deben representar el final de una etapa del proyecto<sup>21</sup>.

Los hitos sirven como un indicativo de que el proyecto ha llegado a cierta etapa, por ejemplo cuando se habla de construcción generalmente no se habla tanto del cronograma que se ha establecido pero si se habla mucho

---

<sup>20</sup> (Institucion Universitaria de Envigado, 2016)

<sup>21</sup> (Sommerville, 2005)



de los hitos como la construcción de los cimientos, o poner la infraestructura eléctrica.

Las fechas que se colocan en los hitos generalmente son una fecha estimativa o a veces puede que no tengan una fecha en concreto para su realización; pero aun así es necesario revisar cuando el hito no se ha cumplido o si se ha retrasado en comparación de la fecha estimada.

Los hitos le permiten a los interesados conocer de manera más simple el estado en que se encuentra el proyecto ya que sirven como herramienta de comunicación entre las partes permitiendo conocer el estado en que se encuentra este sin recurrir a la revisión de todas y cada una de las tareas que han establecido.

Los hitos permiten el seguimiento por parte del equipo partiendo desde el punto en que se pueden ver como pequeños proyectos, e incluso se puede desglosar el presupuesto en partes más pequeñas.

## **2.9. STAKEHOLDERS O INTERESADOS**

En términos generales un Stakeholder es una persona, un grupo o una organización que se ve afectada de manera directa o indirecta con el éxito o fracaso de un proyecto; estos pueden resultar afectados de manera positiva o negativa los resultados del proyecto.

Un interesado puede ejercer control sobre el equipo de trabajo y el proyecto con el fin de obtener resultados que pueda ser satisfactorios para ellos. El equipo de trabajo debe estar en la capacidad de identificar a todos o al menos la mayoría de stakeholder que afecten de cualquier manera, internos, externos, etc; con el fin de determinar los requisitos del proyecto y las expectativas que estos puedan tener asegurando el éxito del proyecto.

Un interesado tiene un nivel de autoridad y responsabilidad que puede cambiar según avance el proyecto; para un equipo es necesario identificarlos en todo momento y tener en cuenta el poder que pueden ejercer, la capacidad que tienen para acelerar o retrasar una tarea.

Existen interesados de todo tipo, desde el líder de la empresa, pasando por un patrocinador que necesita ver su inversión en buenas manos, los clientes, los usuarios, los distintos departamentos que existan en una organización, vendedores, socios, gerentes, organizaciones gubernamentales, ministerios, consultores entre otros<sup>22</sup>.

Generalmente es aconsejable definir la mayor cantidad o todos los stakeholders en las etapas iniciales del proyecto, ya que muchos de estos pueden ser determinantes en fases avanzadas, igualmente cada uno de estos interesados pueden llegar a aportar gran cantidad de información en el momento de establecer los requisitos<sup>23</sup>.

## **2.10. RIESGOS**

Un riesgo es un evento que en caso de presentarse tiene un efecto sobre los objetivos del proyecto; estos pueden ser positivos o negativos; estos últimos pueden causar problemas como aumento en los costos, retrasos, baja calidad, impacto ambiental, daño material, daño personal, pérdidas económicas, entre otras. Es necesario que dichos riesgos sean mitigados o controlados para que tenga mínimos o nulos efectos.

Los riesgos positivos y negativos se conocen como oportunidades y amenazas; estos pueden aceptarse si se encuentran dentro de la tolerancia<sup>24</sup>.

Los riesgos pueden ser clasificados en las siguientes categorías:

- Riesgos del proyecto.
- Riesgos técnicos
- Riesgos del negocio
- Riesgos del mercado

---

<sup>22</sup> (Project Management Institute, 2008)

<sup>23</sup> (Sols Rodríguez-Candela, Fernández Fernández, & Romero Yacobi, 2013).

<sup>24</sup> (Project Management Institute, 2008)

- Riesgo estratégico
- Riesgo de ventas
- Riesgo de dirección
- Riesgo de presupuesto
- Riesgo operacional
- Riesgo externo
- Riesgo de calendario

Dependiendo de su facilidad de detección, pueden ser clasificados en riesgos conocidos, riesgos predecibles y riesgos impredecibles.

Dentro de la gestión de riesgos es necesaria la identificación de estos riesgos para su posterior análisis, se puede hacer en un listado para medir la probabilidad y el impacto y después ordenarlos según la prioridad que presenten.

Al tener identificados los riesgos que se puedan presentar en un proyecto, es necesario generar un plan para tratar con cada riesgo el paso siguiente es la ejecución de dicho plan para terminar con una comprobación y monitoreo para revisar que estos riesgos se hayan eliminado efectivamente y después realizar una nueva revisión para detectar nuestros riesgos.

En el momento de responder a un riesgo se debe seleccionar una de las siguientes estrategias:<sup>25</sup>

- Eliminación: se busca eliminar la amenaza quitando la causa que pueda provocarla.
- Transferencia: se traslada las consecuencias de este riesgo de un tercero junto con la responsabilidad de responder ante este.
- Mitigación: se busca minimizar las consecuencias que pueda producir el riesgo dentro de los límites aceptables.
- Aceptación: sucede cuando se decide no actuar cuando se presente el riesgo.

---

<sup>25</sup> (Escuela de Organización Industrial, 2016)

## 2.11. REQUERIMIENTOS

El PMBOK define como un requerimiento o requisito como una condición que debe existir en un producto o servicio para satisfacer una especificación propuesta de manera formal.<sup>26</sup>

Son todas aquellas características observables que los interesados y clientes desean que estén incluidas dentro del sistema, estos pueden ser necesidades y expectativas de las personas que están relacionadas con el proyecto.

La gestión de requerimientos tiene como propósito verificar que el proyecto cumpla con las expectativas de los clientes y de los interesados; también sirve como un vínculo que relaciona lo que desea el cliente y lo que deben hacer los equipos para cumplir esos deseos.

Un requerimiento se establece:

- Restricciones que un usuario o cliente impone.
- Funcionalidades que deben incluirse en el sistema para que el proyecto satisfaga una necesidad.

De igual manera debe cumplir las siguientes características:

- Debe ser interpretado de una sola manera, es decir no se puede prestar para ambigüedades.
- Al momento de implementarlo debe arrojar resultados, es decir debe ser verificable.
- La formulación de este debe ser lo más clara posible.
- Debe ser viable en cuanto a presupuesto, tiempo y recursos.
- La existencia de este requerimiento debe ser necesaria.
- Para su comprensión no debería depender del conocimiento de otros requerimientos.
- Se especifica tomando en cuenta todas las posibles salidas y condiciones que se puedan presentar<sup>27</sup>.

---

<sup>26</sup> (Project Management Institute, 2008)

<sup>27</sup> (Ibañez, 2016)

## **2.12. CAMBIOS**

Dentro del desarrollo de un proyecto, se pueden emitir solicitudes de cambio que permiten ajustar los alcances de un proyecto, los requisitos, el producto o los costos de este. Estos pueden impactar en los entregables; estos pueden incluir las siguientes acciones:

- Acción correctiva: pretende realinear el desempeño del trabajo del proyecto con el plan para la dirección de este.
- Acción preventiva: es una actividad que pretende asegurar el desempeño del proyecto para que este alineado con el plan para la dirección de este.
- Reparación de defectos: Actividad orientada a cambiar un componente de un producto con el cual el cliente no se encuentra conforme.<sup>28</sup>

## **2.13. METODOLOGIAS DE DESARROLLO**

### **2.13.1. Metodologías Tradicionales de Desarrollo**

Desde que se inició el desarrollo de software, se generó al tiempo una necesidad de mejorar y adaptar los procesos para realizar software con una eficacia mucho mayor; para esto se utilizaron conocimientos y procedimientos existentes en otras áreas del conocimiento y fueron adaptadas a la disciplina del desarrollo de software lo que tuvo como resultado un proceso manejado por etapas lineales y secuenciales que produjeron una mejora necesaria para el desarrollo de nuevo software.<sup>29</sup>

Estas metodologías tradicionales se enfocan en entregar una documentación exhaustiva del proyecto y en seguir de manera rigurosa la planeación del proyecto definida en sus fases iniciales,

---

<sup>28</sup> (Project Management Institute, 2008)

<sup>29</sup> (Figuerola, Solis, & Cabrera)

generalmente su implementación es algo costosa, no es muy eficiente en entornos cuya condición es dinámica y se enfoca en la revisión administrativa de la documentación.

A continuación se presentan las metodologías tradicionales más usadas:

- **Rational Unified Process (RUP):** Provee un acercamiento a la asignación de tareas, tiene como finalidad que produzca software de alta calidad que tenga en cuenta las necesidades de los clientes y los usuarios finales; Esta metodología es creada y soportada por Rational Software, estos trabajan muy de cerca con clientes y consultores para mantener sus procesos actualizados y mejorados de manera continua y se basa en 6 prácticas de desarrollo efectivo<sup>30</sup>.
  - Desarrollo iterativo
  - Manejo de requerimientos
  - Uso de arquitecturas basadas en componentes
  - Usa un modelo de software visual
  - Verifica la calidad de software
  - Realiza control de cambios

De igual manera maneja 4 fases en su ciclo de desarrollo:

- Fase de Concepción
- Fase de Elaboración
- Fase de Construcción
- Fase de Transición

Es necesario que cada fase tenga una meta definida, cuando se llega al final de estas se deben tomar decisiones y se debe verificar que las metas pensadas se hayan cumplido; cada fase

---

<sup>30</sup> (Rational, The Software Development Company, 2001)

tiene sus propias salidas en forma de entregables o procesos que permiten progresar en el desarrollo.

Esta metodología usa el lenguaje unificado de modelado o UML para expresar requerimientos del desarrollo y procesos.

Esta metodología tiene las siguientes ventajas:

- Evaluación de objetivos en cada una de sus fases lo cual permite el cambio de estos si es necesario.
- Es intuitivo, ya que solo contiene pasos necesarios para el desarrollo.
- Realiza seguimiento en cada fase.

Esta metodología también presenta algunas desventajas:

- Puede ser muy flexible para algunos proyectos
- El cliente tiene que ser muy claro en lo que quiere, para así mismo brindar una explicación detallada de lo que busca para obtener un alcance del proyecto.

- **Microsoft Solution Framework (MSF):** Es un acercamiento adaptable para brindar soluciones tecnológicas de manera rápida, usando pocas personas y con menos riesgos, buscando resultados de alta calidad; esta metodología permite a los gestores de proyectos tener una herramienta de ayuda ante un posible fracaso del proyecto.

MSF se enfoca principalmente en unir las metas del negocio con la tecnología, estableciendo roles y responsabilidades y la finalidad de un proyecto de manera clara; también implementa hitos de manera iterativa; maneja los riesgos de manera proactiva y responde a los cambios efectivamente<sup>31</sup>.

Esta metodología maneja los siguientes principios:

---

<sup>31</sup> (Microsoft Press, 2015)

- Fomenta la comunicación Abierta
- El trabajo gira en torno a una visión compartida
- Empodera a los miembros del equipo
- Establece responsabilidad compartida
- Entrega un valor incrementado
- Se mantiene ágil, adaptándose al cambio
- Invierte en calidad
- Aprende de todas las experiencias

Esta metodología hace que sus usuarios tengan un cambio de mentalidad para maximizar el éxito; se busca que quienes sigan los pasos de esa mentalidad la apliquen dentro y fuera de sus trabajos.

- Fomenta un equipo de pares.
- Se enfoca en el valor del negocio
- Mantiene una perspectiva de soluciones
- Siente orgullo por el trabajo que desempeñan
- Aprenden de manera continua
- Internalizan la calidad del servicio
- Practican la buena ciudadanía
- Cumplen con sus compromisos

Esta metodología maneja cinco fases:

- Visión y alcances
- Planificación
- Desarrollo
- Estabilización
- Implantación

Tiene las siguientes ventajas

- Permite a los miembros del equipo apropiarse del proyecto.
- Hace que el equipo se unifique detrás de una visión común del proyecto

De igual manera posee las siguientes desventajas:



- Puede llegar a ser útil como un modelo complementario a otras metodologías existentes que tener el peso suficiente para ser una metodología.

**Metodología Win Win en el Modelo Espiral:** Este método es una extensión de la metodología en espiral, se ejecuta de la misma manera que espiral, pero la diferencia es que el tiempo que se invierte en la identificación de requerimientos se usa para que el equipo de desarrollo y el cliente puedan negociar los requerimientos que serán incluidos en la iteración que se encuentren; se denomina Win Win o gana – gana porque el cliente termina con un producto que satisface los requerimientos que había solicitado<sup>32</sup>.

Esta metodología tiene las siguientes fases

- Requerimientos y Análisis
- Diseño
- Codificación
- Pruebas
- Mantenimiento

Tiene las siguientes ventajas:

- Usa la metodología en espiral, la cual es bien aceptada en diversos lugares.

De igual manera posee las siguientes desventajas:

- Es limitado cuando el desarrollo está orientado a una plataforma o un sistema orientado a transacciones.
- El proyecto completo de muy largo, costoso y complicado.
- El equipo carece de suficiente experiencia.

**ICONIX:** Esta metodología se encuentra en un punto entre RUP y la Extreme Programming, hace uso de UML mientras mantiene la trazabilidad de los requerimientos y a su vez permite que los casos de

---

<sup>32</sup> (Choudhury, 2015)

uso sean concretos, específico y entendibles para que el equipo de desarrollo tenga un mejor desempeño. Tiene un enfoque minimalista que hacia el área entre los casos de uso y el código. Se enfatiza en lo que necesita que pase en el punto del ciclo de desarrollo que se va a iniciar, en algún punto cuando se empieza el caso de uso también se necesita hacer un buen análisis y un buen diseño<sup>33</sup>.

Esta metodología posee las siguientes ventajas:

- Al ser un punto intermedio entre RUP y XP, se permite el uso de diagramas UML para establecer etapas del diseño y desarrollo.
- Es una vía para saber que sucede entre la formulación del caso de uso y el código.

### **2.13.2. Metodologías Ágiles de Desarrollo**

En una reunión celebrada en estados unidos durante el año 2001 se reunió un grupo de 17 expertos en la industria del software; entre estos se encontraban creados de metodologías de software y tenían como objetivo principal trazar nuevos principios que permitieran a los desarrolladores realizar software de una manera rápida y que tuviera tolerancia a los cambios que pudiera surgir durante un proyecto, durante esta reunión se redactó lo que se conoce como Manifiesto Ágil, el cual engloba la filosofía de los métodos de programación ágil.

El manifiesto ágil se puede resumir en que, importa construir un buen equipo, también es vital construir software de calidad que documentación, se solicita la constante colaboración del cliente y se debe estar en la capacidad de responder a cambios en lugar de apegarse a un plan; estas características se unen a 12 principios básicos que maneja la programación ágil<sup>34</sup>.

---

<sup>33</sup> (Rosenberg & Scott, 2001)

<sup>34</sup> (Canos, Letelier, & Penades, Metodologías Ágiles en el Desarrollo de Software, 2003, p. 1 - 8)

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva. III.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- El software que funciona es la medida principal de progreso.
- Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Las metodologías de desarrollo ágil poseen las siguientes ventajas<sup>35</sup>:

- Son útiles cuando los grupos de trabajo son reducidos, es decir, están conformados de 10 personas o menos.
- Se centra en el producto final y no en la documentación, es decir genera solo la documentación necesaria y únicamente cuando sea el momento de tomar una decisión importante.
- Posee adaptabilidad en caso de que sea necesario realizar cambios durante cualquiera de las fases del desarrollo o en cualquier momento del proyecto.
- Aplica un proceso de mejoramiento continuo, lo que garantiza un mejor desarrollo y un software de calidad
- Existe flexibilidad en las normas de trabajo, ya que según las metodologías pesadas las normas son impuestas de manera externa, en las metodologías ágiles se hace lo opuesto.
- Hace que el cliente sea parte del equipo de desarrollo.

Dentro de las metodologías ágiles se encuentran las siguientes:

- **XP:** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente

---

<sup>35</sup> (Orjuela Duarte & Rojas, Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo, En: Revista Avances en Sistemas e Informática, 2008, p 159 - 171)

adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico<sup>36</sup>.

- **SCRUM:** es una metodología desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. La cual define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. La primera característica consiste en que el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.
- **Crystal:** se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).
- **DSDM:** se define como el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad,

---

<sup>36</sup> (Canos, Letelier, & Penades, Metodologías Ágiles en el Desarrollo de Software, 2003, p 1- 8)

estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación<sup>37</sup>.

- **ASD:** su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.
- **FDD:** se define como un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.
- **LD:** Definida por Bob Charette a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios<sup>38</sup>.

### 2.13.3. Comparación entre metodologías

Para este caso se realiza una comparación entre las metodologías, primero entre metodologías tradicionales y después se comparan las

---

<sup>38</sup> (Canos, Letelier, & Penades, Metodologías Ágiles en el Desarrollo de Software, 2003, p 1- 8)

metodologías ágiles para obtener dos metodologías y entre esas dos opciones escoger aquella que permita un desarrollo de la aplicación que cumpla con estándares de calidad y documentación para este proyecto.

A continuación se presenta un cuadro comparativo con las metodologías tradicionales que se me han mencionado anteriormente en este documento.

Característica	RUP	MSF	Win Win	Iconix
Enfoque de desarrollo	Iterativo	Iterativo	Incremental	Iterativo
Tiempo de iteraciones recomendado	1 mes	1 mes	1 mes	1 a 2 semanas
Equipo de proyecto	Medianos	Medianos	Medianos a grandes	Todos los tamaños
Comunicación en equipo	Basada en documentación	informal	informal	Basada en documentos
Tamaño del proyecto	Medianos y grandes	Todos los tamaños	Todos los tamaños	Todos los tamaños
Involucramiento del cliente	Se involucra dando detalles claros	involucrado	involucrado	involucrado
Documentación del proyecto	La documentación es importante	La documentación es importante	La documentación es importante	La documentación es importante
Ventajas	<p>Evaluación de objetivos en cada una de sus fases lo cual permite el cambio de estos si es necesario.</p> <p>Es intuitivo, ya que solo contiene pasos necesarios para el desarrollo.</p>	<p>Permite a los miembros del equipo apropiarse del proyecto.</p> <p>Hace que el equipo se unifique detrás de una visión común del proyecto</p>	<p>Usa la metodología en espiral, la cual es bien aceptada en diversos lugares.</p>	<p>Al ser un punto intermedio entre RUP y XP, se permite el uso de diagramas UML para establecer etapas del diseño y desarrollo.</p> <p>Es una vía para saber que sucede entre la</p>

	Realiza seguimiento en cada fase.			formulación del caso de uso y el código.
Desventajas	<p>Puede ser muy flexible para algunos proyectos</p> <p>El cliente tiene que ser muy claro en lo que quiere, para así mismo brindar una explicación detallada de lo que busca para obtener un alcance del proyecto.</p>	<p>Puede llegar a ser útil como un modelo complementario a otras metodologías existentes que tener el peso suficiente para ser una metodología.</p>	<p>Es limitado cuando el desarrollo está orientado a una plataforma o un sistema orientado a transacciones.</p> <p>El proyecto completo de muy largo, costoso y complicado.</p> <p>El equipo carece de suficiente experiencia.</p>	Puede tener mucha documentación

**Table 1 Comparación Metodologías Tradicionales**

En el siguiente cuadro se aprecia la comparación entre metodologías Ágiles

Característica	XP	Scrum	DSDM
Enfoque de desarrollo	Incrementos iterativos	Incrementos iterativos	Iterativo
Tiempo de iteraciones recomendado	Una a 6 semanas	2 a 4 semanas	80% de la solución en un 20% del tiempo
Equipo de proyecto	Equipos pequeños	Todos los tamaños	Todos los tamaños
Comunicación en equipo	Informal con reuniones diarias	Informal con reuniones diarias	Basado en la documentación
Tamaño del proyecto	Proyectos muy pequeños	Todos los tipos	Todos los tipos
Involucramiento del cliente	Cliente involucrado	Cliente involucrado a través del rol de dueño del producto	Cliente involucrado a través de liberaciones frecuentes



Documentación del proyecto	Solo documentación básica	Solo documentación básica.	Documentación apenas existente.
Ventajas	<p>Espacio de trabajo abierto.</p> <p>Cliente como parte del equipo.</p> <p>Prácticas muy bien definidas.</p> <p>Retroalimentación.</p>	Alto nivel de comunicación y colaboración.	<p>Enfoque de prioridad hacia los requerimientos.</p> <p>Gestión de proyecto eficiente.</p>
Desventajas	<p>Documentación débil</p> <p>Falta de disciplina.</p> <p>La presencia del cliente es obligatoria.</p>	<p>Documentación débil.</p> <p>Pobre control sobre el proyecto.</p>	Documentación compleja

**Table 2 Comparación de Metodologías Ágiles 1**

Característica	FDD	ASD	Crystal
Enfoque de desarrollo	Iterativo	iterativo	Incremental
Tiempo de iteraciones recomendado	Dos días a dos semanas	Cuatro a ocho semanas	Depende de la metodología
Equipo de proyecto	Muchos miembros, más de un equipo	Equipos muy pequeños, de 5 a 9 miembros	Todos los tamaños dependiendo del método de la familia.
Comunicación en equipo	Basado en documentación	Informal cara a cara	Informal cara a cara
Tamaño del proyecto	Proyectos muy complejos	Proyectos pequeños	Todo tipo de proyectos dependiendo del método
Involucramiento del cliente	Cliente a través de reportes	Cliente a través de liberaciones	Cliente a través de liberaciones incrementales
Documentación del proyecto	Documentación es importante	Solo documentación básica	Solo documentación básica
Ventajas	Reportes y documentación permiten multitarea	Desarrollo prioritario de los componentes de alto riesgo.	Metodologías que se ajustan al tipo de proyecto y el tamaño

		Importancia del ciclo de aprendizaje.	
Desventajas	Pertenencia individual del código  No aplicable a proyectos pequeños	Métodos de documentación pobre	Coordinación ineficiente de equipos muy grandes.

**Table 3 Comparación Metodologías Ágiles 2**

Lo visto en los cuadros anteriores permite desechar aquellas metodologías cuyos grupos sugeridos son muy grandes, ya que este proyecto será desarrollado por una persona y el asesor del proyecto, lo cual sería un total de dos integrantes, por lo que se necesita una metodología de desarrollo sea beneficiosa para grupos muy pequeños; lo cual dejaría como posibles opciones a ICONIX, Scrum, DSDM y ASD.

También es necesario considerar la cantidad de documentación que se debe emplear para este proyecto lo cual descarta a la metodología Scrum por tener una documentación débil y a la DSDM por ser muy compleja; solo dejando a ASD e ICONIX.

Aunque la metodología ASD tiene prioridad sobre los componentes de alto riesgo, ICONIX posee un diseño que permite el desarrollo de los componentes, usando los diagramas UML necesarios para una etapa de requerimientos y diseño lo suficientemente robusta para que el desarrollo sea mucho más sencillo para el equipo.

En este caso la mejor elección es la metodología ICONIX por las siguientes razones:

- Es una metodología que se encuentra en un punto medio entre las ágiles y las tradicionales, permitiendo un desarrollo relativamente rápido gracias a sus iteraciones de 1 a 2 semanas pero con una documentación lo suficientemente robusta para comprender, revisar, modificar y mejorar aspectos de la aplicación.
- Al tener una serie de diagramas UML como el de casos de uso, el de robustez y el de secuencia se tiene una manera de trabajar más sencilla.

- Al darle importancia al desarrollo y a la documentación por igual se tiene una herramienta que puede simplificar la construcción de un software.

## 2.14. INGENIERIA WEB

En la mayoría de aplicaciones web se pueden encontrar las siguientes características:

- **Intensidad de red:** la aplicación reside en la red y tiene como misión satisfacer las necesidades de una amplia gama de usuarios o clientes, un aplicativo de este tipo puede residir en la red o en una intranet.
- **Concurrencia:** la aplicación debe ser capaz de soportar una gran cantidad de usuarios accediendo al programa al mismo tiempo.
- **Carga Impredecible:** El número de usuarios que acceden a la aplicación web puede variar conforme a los días.
- **Desempeño:** se debe procurar que el usuario no se quede demasiado tiempo esperando para acceder a la aplicación web, ya que puede optar por usar otro aplicativo.
- **Disponibilidad:** la aplicación debe encontrarse disponible en todo momento 24 horas al día, 7 días a la semana, 365 días al año sin importar el lugar del mundo donde se necesite acceder, ya que pueden existir momentos en los cuales las aplicaciones se encuentren en mantenimiento y los usuarios no puedan acceder, por lo que se hace necesario hallar solución a un problema de ese tipo.
- **Gobernada por datos:** Las aplicaciones web son usadas como aplicativos intermediarios entre el usuario y la base de datos para acceder a dichas bases de datos y así brindarle a los clientes una visión interactiva y legible de los datos.

- **Sensibilidad al contenido:** el aplicativo debe tener una presentación estética de buena calidad y también debe presentar sus contenidos con un buen aspecto visual.
- **Evolución continua:** las aplicaciones web se diferencian de las aplicaciones de escritorio por la cantidad de actualizaciones que estas reciben, es decir un programa convencional tiene un cronograma de lanzamiento de sus parches y actualizaciones, en cambio los aplicativos web deben actualizarse más seguido.
- **Inmediatez:** Una aplicación web tiene cierto tiempo para ser comercializado, entonces los ingenieros web deben ser capaces de aplicar metodologías y herramientas que se ajustan a las necesidades y tiempos de entrega para aplicativos web.
- **Seguridad:** Es necesario implementar medidas de seguridad que permitan la limitación de acceso a la aplicación y a los datos, de manera que se pueda proteger la confidencialidad de los datos.
- **Estética:** La correcta distribución de los elementos dentro de la aplicación web.

Dentro del desarrollo web se pueden encontrar las siguientes categorías de aplicaciones web:

- **Informativo:** ofrece navegación y acceso a enlaces simples.
- **Descarga:** se puede descargar la información proveniente de los servidores
- **Personalizable:** el usuario puede personalizar el contenido de la aplicación dependiendo de las necesidades que se presenten.
- **Interacción:** hay una comunicación entre usuarios, por medio de salas de chat, mensajería, anuncios, entre otros.
- **Entrada del usuario:** El usuario llena varios formularios con el fin de realizar un proceso de comunicación.

- **Orientada a transacciones:** El usuario hace una solicitud que la aplicación debe cumplir.
- **Orientada a servicios:** la aplicación le proporciona un servicio al usuario.
- **Portal:** la aplicación lleva al usuario hacia otros servicios fuera del dominio de la aplicación.
- **Acceso a una base de datos:** el usuario puede consultar una base de datos y extraer la información que necesite.
- **Almacén de datos:** el usuario consulta una gran cantidad de bases de datos y extrae la información.

#### 2.14.1. Estratos de la Ingeniería Web

De acuerdo con Pressman, el desarrollo de sistemas y aplicaciones basados en la web incorpora modelos de proceso especializados, métodos de ingeniería del software adaptados a las características de desarrollo de aplicaciones Web y un conjunto de importantes tecnologías habilitadoras. Los procesos, métodos y tecnologías proporcionan un enfoque en estratos de la Ingeniería web que es conceptualmente idéntico a los estratos de la ingeniería de software<sup>39</sup>.

#### 2.14.2. Proceso

Los modelos de procesos de la ingeniería web adoptan la filosofía de las metodologías ágiles de desarrollo, dichas metodologías incorporan un enfoque riguroso con rápidos ciclos de desarrollo.

El hecho de que estos ciclos deban ser rápidos garantiza que el análisis hecho durante cada fase, se haga de manera consciente, que se deba realizar un desarrollo y debe tener un enfoque organizado de prueba, pero aun así debe existir un proceso que cumpla con las siguientes características:

---

<sup>39</sup> (Pressman, Ingeniería del Software, Un Enfoque Práctico. México D.F: Mc Graw Hill, 2006)

- Adopte los cambios.
- Aliente la creatividad y la independencia del equipo de desarrollo y fortalezca la interacción con los interesados en la aplicación.
- Construya un sistema compuesto de pequeños equipos de desarrollo.
- Subraye el desarrollo evolutivo o incremental mediante cortos ciclos de desarrollo.

### 2.14.3. Métodos

En la ingeniería web hay un conjunto de labores que le permiten al ingeniero comprender, caracterizar y construir un aplicativo web de alta calidad, los métodos de la ingeniería web se pueden categorizar de la siguiente manera<sup>40</sup>:

- **Métodos de Comunicación:** definen el enfoque que facilita la comunicación entre los ingenieros web y los demás participantes de la aplicación web, es importante durante la fase de recolección de información y establecimiento de requerimientos.
- **Métodos de Análisis de Requisitos:** es la base que permitirá comprender mejor la composición de la aplicación, funciones, interacción con el usuario y todo aquello que el usuario requerirá o usará durante su funcionamiento.
- **Métodos de Diseño:** abarcan una serie de técnicas de diseño que abordan el contenido, la aplicación y la arquitectura de información, así como el diseño de interface y navegación de la aplicación.

---

<sup>40</sup> (Pressman, Ingeniería del Software, Un Enfoque Práctico. México D.F: Mc Graw Hill, 2006)

- **Métodos de Prueba:** incorporación de revisiones técnicas formales, pruebas de navegación, pruebas de facilidad de uso, pruebas de seguridad y pruebas de configuración. Aparte de estas pruebas, es necesario aplicar técnicas de gestión de proyectos, técnicas de gestión de configuración de software y de revisión.

#### 2.14.4. Herramientas y Tecnología

Con el pasos de los años se han desarrollado una amplia gama de tecnologías que abarcan un amplio conjunto de descripción de contenido y lenguajes de modelación (por ejemplo, HTML, VRML, XML), lenguajes de programación (por ejemplo, java), recursos de desarrollo basados en componentes (por ejemplo, COBRA, COM, ActiveX, .NET), navegadores, herramientas multimedia, herramientas de autoría de servicio, herramientas de conectividad de bases de datos, herramientas de seguridad, servidores y utilidades de servidor, y herramientas de administración y análisis de sitio<sup>41</sup>.

#### 2.14.5. Procesos de la Ingeniería Web

Comunicación con el Cliente: Dentro del proceso de Ingeniería Web la comunicación con el cliente se caracteriza por medio de dos grandes tareas: el análisis del negocio y la formulación. El análisis de negocio define el contexto empresarial-organizativo para la aplicación. Además se identifican los participantes, se predicen los potenciales cambios en el ambiente o los requisitos del negocio y se define la integración entre el software y otras aplicaciones de negocios, bases de datos y funciones. La formulación es una actividad de recopilación de requisitos que involucra a todos los participantes. El intento es describir el problema que la aplicación habrá de resolver; con el aprovechamiento de la mejor información posible. Además, se intenta identificar áreas de incertidumbre y donde ocurrirán cambios potenciales.

- **Planeación:** se crea el plan del proyecto para el incremento del programa. El plan consiste de una definición de tareas y un calendario

---

<sup>41</sup> (Pressman, Ingenieria del Software, Un Enfoque Practico. Mexico D.F: Mc Graw Hill, 2006)

de plazos respecto al periodo proyectado para el desarrollo del incremental del aplicativo.

- **Modelado:** Las labores convencionales de análisis y diseño de ingeniería del software se adaptan al desarrollo de una aplicación web, se mezclan y luego se funden en la actividad de modelado. El intento es desarrollar análisis rápidos y modelos de diseño que definan requisitos y al mismo tiempo representen un aplicativo que sea satisfactoria.
- **Construcción:** Las herramientas se aplican para construir lo que se ha modelado. Una vez construida los esfuerzos se dirigen a una serie de pruebas rápidas para asegurar que se descubran los errores en el diseño (es decir contenido, arquitectura, interface, navegación).
- **Despliegue:** La aplicación se configura para su ambiente operativo, se entrega a los usuarios finales y luego comienza un periodo de evaluación. la retroalimentación acerca de la evaluación se presenta al equipo y el incremento se modifica conforme se requiera<sup>42</sup>.

## 2.15. TIPOS DE PRUEBAS

Las pruebas se dividen en pruebas de verificación y validación, en ambas se revisa que el sistema se esté construyendo de manera correcta, en el caso de las pruebas de verificación, se prueba que el sistema funcione correctamente, en las pruebas de validación se toma en cuenta que el sistema cumpla con las características deseadas por el usuario. Existen varias técnicas para el desarrollo de pruebas.

**Pruebas de Regresión:** este tipo de pruebas se realizan después de haberse hecho un cambio en el sistema debido a un error dentro de este; se revisa que el sistema funcione normalmente después de haberse introducido los cambios.

---

<sup>42</sup> (Pressman, Ingenieria del Software, Un Enfoque Practico. Mexico D.F: Mc Graw Hill, 2006)



**Pruebas de Operación:** la finalidad de estas pruebas es medir que el sistema funcione dentro de condiciones normales por un largo periodo de tiempo, estas pruebas también son llamadas pruebas de confiabilidad.

**Pruebas basadas en casos de uso:** estas pruebas se realizan con el fin de evaluar los casos de uso presentados, se usa para verificar que el sistema cumpla con las especificaciones descritas en los casos de uso.

**Pruebas de caja negra:** Estas pruebas tienen como propósito evaluar las entradas y salidas del sistema, es decir se evalúa que hace el sistema, pero no se observa como suceden las cosas dentro del sistema; sencillamente se envían unos parámetros de entrada al sistema y se obtienen unas salidas, para cuando el sistema finaliza sus pruebas, se comparan las salidas del sistema con unas salidas esperadas por quien realiza la prueba.

**Pruebas estructurales o pruebas de caja blanca:** Buscan revisar que la estructura interna del programa sea la correcta, es decir, se debe conocer como está constituido el programa de manera interna; para ello se usan pruebas de cobertura, las cuales consisten en probar todos los casos con todos los parámetros y caminos existentes con el fin de hallar combinaciones inusuales que puedan afectar el programa.

**Pruebas de integración:** en este caso se van integrando todos los módulos existentes en el sistema, hasta que todo el sistema esté funcionando correctamente, esto se hace con el fin de revisar que todos los módulos del sistema puedan trabajar bien unos con otros, se tiene la ventaja de que con estas pruebas se pueden encontrar fallos que no se habían contemplado en las pruebas de unidad; para estas pruebas se utilizan las pruebas basadas en casos de uso, se tiene en cuenta el funcionamiento básico, el funcionamiento inusual y el manejo de excepciones.

#### **2.15.1. APLICABILIDAD DE LAS PRUEBAS**

Estas pruebas han sido aplicadas en diversos puntos del desarrollo de la aplicación; en las etapas iniciales se debieron realizar varios cambios a la base de datos y por ende al software por lo que fue necesario aplicar pruebas de regresión después de realizar los ajustes necesarios.

Realizar pruebas de caja blanca y caja negra se realizaron con mucha frecuencia ya que son necesarias para detectar errores imprevistos y también son útiles para probar que se ha programado de manera correcta ya que al conocer las posibles salidas se sabe si el resultado es el esperado o no, si el resultado no era el esperado se revisaba el código mediante puntos de interrupción mirando el comportamiento interno del sistema.

Paralelamente a las pruebas anteriores se realizaban las de casos de uso, ya que permitían observar si se cumplía con cada requerimiento descrito.

Finalmente con las pruebas de integración se hizo la revisión de la aplicación haciendo un posible flujo de la aplicación probar los posibles caminos y fallos que puedan suceder al integrar los módulos que se han desarrollado.

## **2.16. COMPARACION ENTRE SOFTWARE DE GESTION DE PROYECTOS**

En este caso se buscaran las aplicaciones web para gestión de proyectos más usadas y conocidas para hacer una comparación entre todos para revisar los puntos que tienen en común, y obtener datos que permitan saber qué cosas debe tener el software a desarrollar y en aspectos se podría fortalecer para hacerlo diferente a las propuestas analizadas.

El primero que se puede encontrar como la más eficiente y más usada aplicación web para gestión de proyectos es OpenProject; esta herramienta Open Source que es similar a Microsoft Office Project ya que posee funcionalidades parecidas<sup>43</sup>.

Dentro de sus ventajas se pueden encontrar las siguientes:

- Manejo de proyectos e hitos.
- Seguimiento de Bugs.
- Manejo de Incidencias.
- Gestión de documentación.
- Sección de noticias del proyecto.

---

<sup>43</sup> (OpenProject, 2016)

- Es colaborativo.
- Tiene líneas de tiempo interactivas.
- Su estructura lo hace una buena opción para proyectos relacionados con las Tecnologías de Información.
- Reporte de costos.
- Maneja licencia GNU GPL lo que permite a los desarrolladores modificar el código y mejorarlo.
- Maneja los módulos que Microsoft Office Project ofrece.

Como desventajas posee:

- No tiene gestión de recursos
- Maneja licencias de pago que restringen la cantidad de usuarios y proyectos que puede manejar el usuario.
- Puede resultar difícil de manejar para proyectos pequeños o empresas pequeñas.

La siguiente herramienta a analizar es Basecamp; esta aplicación funciona como un organizador de proyecto, donde los involucrados en este se pueden mantener actualizados sobre lo que ocurre y a su vez discutir sobre este<sup>44</sup>.

Ventajas:

- Tablero de discusión e hilos de comentarios.
- Chat en tiempo real.
- Chequeo automático de las tareas que están realizando los usuarios.
- Listas de cosas para hacer.
- Almacenamiento de archivos.
- Agenda centralizada.
- Generación de reportes
- Datos de actividad.
- Es ideal para todos los tamaños de proyecto
- Presenta una interfaz sencilla y amigable con el usuario

---

<sup>44</sup> (Basecamp, 2016)

## Desventajas

- No tiene gestión de incidencias
- No maneja portafolios
- No tiene gestión de costos
- Carece de manejo de tiempos

Otra herramienta existente es Zoho Projects; esta provee a los usuarios de una interfaz intuitiva, permite el uso de una variada cantidad de plantillas en caso de que las establecidas por defecto no cumplen con sus necesidades y facilita la comunicación entre participantes.<sup>45</sup>

## Ventajas

- Variedad de plantillas para que los usuarios puedan escoger la que gusten.
- Creación de hitos, tareas y listas que permiten un mejor seguimiento.
- Permite la colaboración entre los participantes del proyecto.
- Posee herramientas de comunicación como tablero de comunicaciones y chat.
- Agenda con las tareas existentes.
- Posee integración con Google Apps

## Desventajas:

- Posee un módulo de gestión de archivos, que en las licencias gratuitas posee una limitación de tamaño a 10MB, en versiones de pago este límite posee un tope superior.
- La licencia gratuita solo permite un proyecto por usuario, al igual que la desventaja anterior esta disminuye dependiendo del plan que se adquiera.
- Con la versión gratuita la mayoría de ventajas enunciadas en la parte superior no se encuentran presentes.

---

<sup>45</sup> (Zoho Corporation Pvt. Ltd, 2016).

Otra aplicación a comprar es Bitrix24, esta aplicación presenta una licencia gratuita que puede agrupar una cantidad ilimitada de usuarios sin importar la versión que se adquiera.<sup>46</sup>

#### Ventajas:

- La versión gratuita tiene muchas herramientas para usar.
- Tiene una amplia gama de mecanismos de comunicación, que van desde mensajería instantánea hasta video llamadas.
- Permite la creación de tareas predecesoras, mediante el uso de diagramas de Gantt para actualizar las tareas junto con su duración.

#### Desventajas

- El manejo de tantas herramientas puede ser complicado para usuarios que no están acostumbrados.
- Puede que tantas herramientas sean innecesarias para un usuario que busca un manejo más simple de sus proyectos.

La siguiente aplicación que se encuentra en la lista es Freedcamp, esta permite un número ilimitado de usuarios en su licencia gratuita, en caso se quiera pagar una versión mejorada sus precios son bastante bajos cual hace que se convierta en una herramienta ideal para muchos usuarios.<sup>47</sup>

#### Ventajas:

- Su versión gratuita permite cantidad ilimitada de proyectos y usuarios.
- Existe una separación entre tareas y cosas para hacer.
- Posee manejo de calendarios que permite una separación entre tareas, eventos e hitos.
- Tiene un sistema de gestión de archivos.

#### Desventajas:

---

<sup>46</sup> (Bitrix, Inc, 2016).

<sup>47</sup> (Freedcamp Inc., 2016).

- Hay un límite en el almacenamiento de archivos, el cual es de 200MB.
- Para usar los módulos de manejo de incidencias, CRM y Wiki hay que pagar un valor adicional.

Como las aplicaciones que se revisaron anteriormente existen muchas más todas manejan principios parecidos, las diferencias se encuentran en el lenguaje que usaron para su desarrollo, la ausencia, presencia o necesidad de pago de algunos módulos; pero todos se caracterizan por la presencia de las siguientes características.

- Reporte de incidencias.
- Existen tareas.
- Algunos manejan un portafolio de proyectos.
- La cantidad de usuarios puede ser ilimitada.
- Pueden manejar una cantidad ilimitada de proyectos
- La colaboración es importante.
- Es necesaria la comunicación entre los usuarios.
- Es importante la diferencia entre hitos y tareas
- Es opcional la existencia de listas de cosas por hacer para cada usuario.
- Tienen una interfaz intuitiva para el usuario.
- Permiten la visualización de tareas asignadas a un usuario y eventos por medio de un calendario y un listado
- El manejo de recursos aunque no está presente en algunas aplicaciones, también es importante que se considere su implementación.
- Se pueden subir archivos relacionados a una tarea.

## **2.17. REQUISITOS DE IMPLEMENTACION**

Se realiza la búsqueda de varias opciones para el hosting de la aplicación, en este caso se debe observar que la aplicación en un inicio debe poseer un lugar donde el almacenamiento sea gratuito o en su

defecto sea un hosting de bajo costo y a su vez tenga soporte para alojar una aplicación desarrollada en ASP.NET MVC 4.

Para ello se encontraron los siguientes hostings que cumplen lo mencionado en el párrafo anterior, estas páginas están recomendadas por Microsoft en su sitio Microsoft Web Hosting Gallery<sup>48</sup>.

El primer sitio de Hosting que esta en la lista es GearHost, el cual posee un plan gratuito con las siguientes especificaciones<sup>49</sup>:

- Nodo Web reservado
- 10 instancias de trabajador máxima
- Pool de aplicación de 2GB
- Ancho de banda de 1TB
- 100 GB de almacenamiento
- Dominio customizado gratuito
- Tiempo de sesión de 24 horas
- Múltiples Bases de datos SQL Server
- Discos de estado solido
- Soporte de .NET 3.5, 4.5 y 4.6
- SQL server 2012 y 2014

El siguiente sitio en la lista es MyWindowsHosting, el cual tiene un plan de 60 días gratuitos y también dispone de un plan premium, con las siguientes especificaciones:<sup>50</sup>

- 1GB de espacio Web
- 1 sitio
- 1 nombre de dominio
- Ancho de banda de 10GB
- 1 FTP
- Soporte para visual studio
- Puede correr múltiples versiones de .NET

---

<sup>48</sup> (Microsoft, 2016).

<sup>49</sup> (GearHost <sup>TM</sup> Inc., 2016)

<sup>50</sup> (MYWINDOWSHOSTING.COM, 2016)

- 1 base de datos
- Soporte para varias versiones de ASP.NET
- Firewall
- Anti-Virus

Dentro de las posibles opciones, se encuentra SmarterASP.NET, también cuenta con un plan gratuito que tiene una duración de 60 días junto con un plan con mejores especificaciones<sup>51</sup>:

- Espacio web ilimitado
- Múltiples sitios
- Múltiples dominios, pero estos tienen un costo anual
- Ancho de banda ilimitado
- ftp ilimitado
- Autenticación de base de datos
- Soporte para SQL
- Firewall
- Soporte técnico

El siguiente en la lista es MyASP.NET, que maneja las siguientes características en su mejor plan de pago<sup>52</sup>:

- Espacio web ilimitado
- Múltiples sitios
- Múltiples dominios, pero estos tienen un costo anual
- Ancho de banda ilimitado
- ftp ilimitado
- Autenticación de base de datos
- Soporte para SQL
- Firewall
- Soporte técnico
- Anti Virus
- Anti Spam

---

<sup>51</sup> (SmarterASP.NET, 2016)

<sup>52</sup> (myASP.NET, 2016)



En este caso, al comparar todos los sitios hallados anteriormente; se puede deducir que todos cumplen con características similares en cuanto almacenamiento y seguridad, pero la mejor opción que tiene características sobresalientes y es compatible con las tecnologías que se usan para desarrollar la aplicación sería GearHost; ya que es un sitio que posee compatibilidad con aplicaciones desarrolladas en Visual Studio y bases de datos SQL y MySql junto con un precio mensual de 5 Dolares en su plan más económico.

A continuación se presenta un cuadro comparativo con cada una de las propuestas mostradas anteriormente, esta tabla se elabora teniendo en cuenta los planes con mejores prestaciones que presta cada servicio de hosting.

	SmarterASP .NET	MyWindows Hosting	myASP. NET	GearHost
Cuota Mensual	\$7.95	\$7.95	\$7.95	\$25
Instalación	Gratis	Gratis	Gratis	Gratis
Espacio (GB)	Ilimitado	Ilimitado	Ilimitado	100 GB
Dominio	Ilimitado	Ilimitado	ilimitado	Ilimitado
Ancho de Banda	Ilimitado	ilimitado	ilimitado	1 TB
Costo de Dominio	\$10 Anual	\$10 Anual	\$10 Anual	Gratis
Cantidad de Bases de Datos	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Compatibilidad con varias versiones de .NET	Si	Si	Si	Si

**Tabla 1 Comparación técnica de los servicios de hosting**

## **2.18. ANALISIS COSTO BENEFICIO**

### **2.18.1. Costos tangibles**

Para este caso se dividirán los costos en costos de inversión, operación y mantenimiento, esto con el fin de realizar el análisis costo beneficio de la aplicación.

Los costos de inversión tienen que ver con todo lo correspondiente a obtener los equipos, personas e infraestructura necesarias para que el proyecto funcione.

Se tiene un presupuesto inicial de 3.000.000 de pesos, los cuales serán distribuidos en el pago de gastos de única vez, pago de servicios y compra de equipos.

El siguiente costo, es el costo de los equipos, esto consiste en el material de trabajo que se usara para realizar el desarrollo; en este caso será necesario lo siguiente:

- PC portátil, aunque el desarrollador ya cuenta con un equipo con las especificaciones mínimas para correr los programas que se usaran para el desarrollo, el valor de dicho equipo es de 1.400.000 pesos colombianos,
- Impresora, en caso de que sea necesaria la impresión de documentos para presentar, este tiene un costo de 100.000 pesos colombianos.

A continuación se presentan los costos de software, es decir aquellos costos que son generados por las licencias que se deben adquirir para el desarrollo de la aplicación sin caer en problemas legales.

- Licencia de Visual Studio 2013: para esta herramienta, la cual será la piedra angular del

desarrollo existen grupos de licencias gratuitas para usuarios, en este caso se optara por una licencia de ese tipo considerando que solo hay un desarrollador.

- Licencia de SQL Server Express 2012: La versión Express de SQL Server es gratuita.
- Servidor donde alojar la aplicación: como se vio en el punto anterior se elegido el servicio de GearHost para subir en ese lugar el aplicativo; por el momento se usara la versión de prueba de 60 días para montar la aplicación cuando sea necesario y hacer las pruebas correspondientes; después del periodo de prueba se puede optar por pagar el plan de 5 Dólares es decir 15.000 pesos colombianos mensuales.

Los costos de infraestructura representan aquellos costos destinados al espacio físico donde se va a realizar el desarrollo e irán destinados de la siguiente manera.

- Servicios y papelería: Agua, Luz, Internet; aproximadamente se invierte 30.000 pesos colombianos de manera mensual.

Finalmente se presenta un cuadro con la sumatoria de los costos:

Costo	Valor
Pc Portatil	1.400.000
Impresora	100.000
Costos de Servidor	15.000
Servicios	30.000
Total	1.545.000

Table 4 Suma de costos

### 2.18.2. Costos Intangibles

Para los costos intangibles hay que tener en cuenta el tiempo que se debe estar desarrollando, que es un mínimo de 8 horas diarias por 5 días de la semana o un total de 40 horas semanales que deberían distribuirse a lo largo de esta o se pueden aumentar según sea necesario, por un tiempo aproximado de 2 meses que es lo que se tiene pensada la etapa de desarrollo.

### 2.18.3. Beneficios Tangibles

Se puede cobrar una membresía de bajo precio para empezar; considerando que los recursos a usar tienen un costo relativamente bajo; se podría cobrar una licencia con un valor de 15.000 pesos colombianos de manera inicial, lo cual cubre el costo del hosting donde se almacena la aplicación.

La tabla con los beneficios tangibles quedaría de la siguiente manera

Beneficio	Valor
Costo de la membresía	20.000
Total	20.000

Table 5 Lista de Beneficios Tangibles

### 2.18.4. Análisis de Beneficios Intangibles

- Accede a los datos del proyecto desde cualquier dispositivo con internet.
- Tener acceso a las tareas de cada proyecto.
- Cada usuario observa las tareas que debe hacer
- Se realiza un control de incidencias en caso de que ocurra una a lo largo del proyecto.
- Se pueden agregar archivos relacionados a una tarea

#### 2.18.5. Comparación entre costos y beneficios

Costos Cotidianos	Beneficios Cotidianos	Costos de única Vez
15.000	20.000	1.545.000

Al restar los beneficios de los costos se obtiene que con una sola membresía se ganarían 5.000 pesos mensuales, en el peor de los casos que es donde solo se tiene un usuario haciendo uno de la membresía; esta ganancia aumentaría de acuerdo a la cantidad de suscripciones que existan.

#### 2.18.6. Tiempo de Amortización del proyecto

El proyecto tiene una inversión inicial de \$1.545.000 y se tiene estimado que puede generar una ganancia mensual de 100.000 pesos de manera inicial, lo cual puede incrementar de manera mensual dependiendo del aumento de los usuarios que podrían ir desde los 10 usuarios y aumentando se obtendría la siguiente amortización:

$$\text{Amortización} = 1.545.000 / 100.000 = 15.45$$

En caso de que los cálculos sean correctos y se manifieste el peor escenario tomaría un aproximado de 15.45 meses recuperar la inversión inicial del proyecto.

### **3. METODOLOGIA**

#### **3.1. LINEA DE INVESTIGACION**

De acuerdo con los lineamientos establecidos anteriormente por la Universidad de San Buenaventura, Sede Bogotá, la línea de investigación apropiada para la realización de este trabajo es: Tecnologías de Información y Comunicación (TIC)

#### **3.2. ENFOQUE DE INVESTIGACION**

El enfoque de investigación escogido en este caso es Construcción de Software.

#### **3.3. TECNICAS DE RECOLECCION DE INFORMACION**

Para este proyecto se usaron como técnicas de recolección de información, la revisión de documentos y de revisión selectiva, para ser más exacto se tomó en cuenta la información consignada en el PMBOK, creado por el Project Management Institute cuya primera edición fue publicada en 1996, pero se tomó en cuenta la información consignada en la quinta edición publicada en el año 2013.

Se realizó un proceso de comparación para realizar la búsqueda de diversas aplicaciones y tomar en cuenta que ofrecía el mercado para poder brindar una nueva característica que la hiciera diferente de las demás.

Se hizo una búsqueda sobre diversos informes que brindaran información sobre las posibles fallas que podría presentar un proyecto de tecnologías de información, lo cual se ve en la formulación de la problemática.

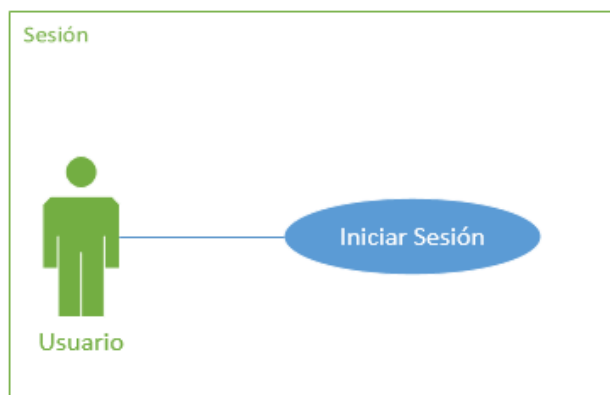
Por último se realizó un proceso de indagación en el cual se realizaban correcciones a un documento y se obtenía nueva información que permitía avanzar en el desarrollo del escrito.

### 3.4.CONSTRUCCION Y ANALISIS DE REQUERIMIENTOS

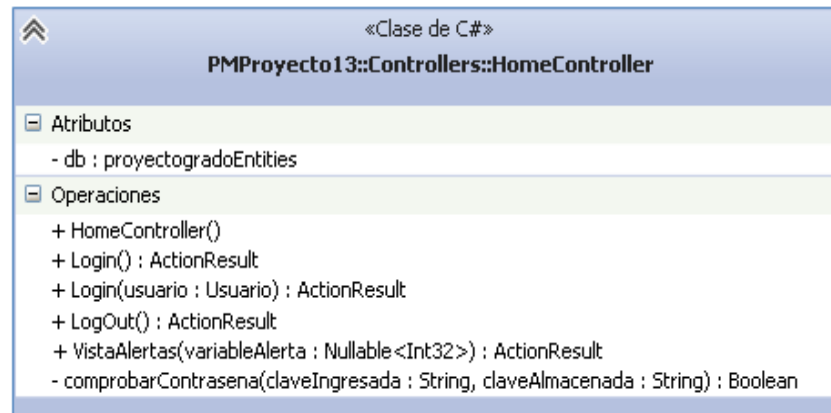
Se presenta el formato para cada requerimiento con sus diagramas de caso de uso y clase.

<b>Identificación</b>	PMB-001
<b>Nombre</b>	Iniciar Sesión
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Debe aparecer una pantalla, en la cual el usuario deberá ingresar un usuario y una contraseña que se generan cuando se registra el usuario.
<b>Precondición</b>	Acceder a la aplicación
<b>Postcondición</b>	Si el usuario tiene proyectos asociados a este, observara un listados con estos y sus respectivas opciones para interactuar
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la URL de la aplicación.</li> <li>2. Esperar a que la aplicación Cargue completamente.</li> <li>3. Ingresar el usuario.</li> <li>4. Ingresar la contraseña.</li> <li>5. Dar clic en “Iniciar Sesión”.</li> </ol>
<b>Camino de Excepción</b>	<p>Que no se pueda Acceder a la aplicación por algún error de acceso.</p> <p>Encontrarse sin una conexión a internet.</p> <p>Carecer de un usuario creado en la aplicación.</p>

#### Diagrama de Caso de Uso

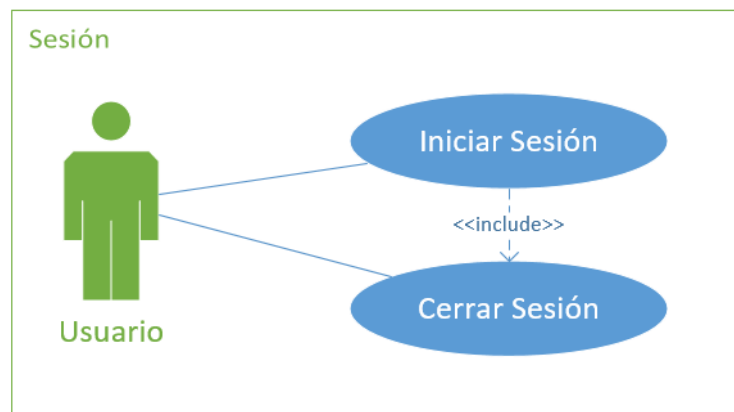


## Diagrama de Clase



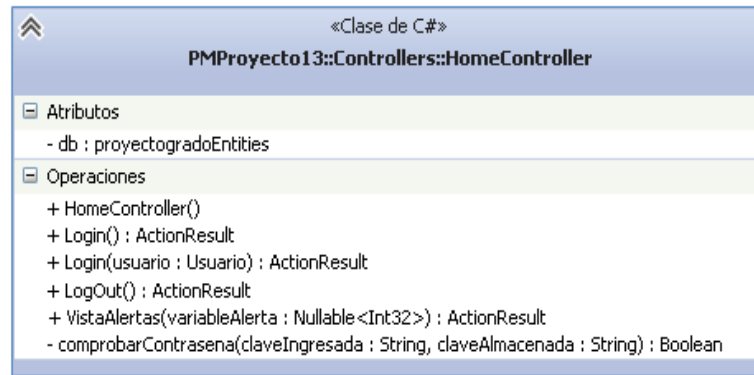
<b>Identificación</b>	PMB-002
<b>Nombre</b>	Cerrar Sesión
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Al finalizar todo el trabajo realizado el usuario puede optar por salir de manera segura de la aplicación.
<b>Precondición</b>	Acceder a la aplicación. Iniciar sesión
<b>Postcondición</b>	Volver a la pantalla de inicio de sesión
<b>Flujo de Eventos</b>	1. Hacer clic en salir. 2. La aplicación cerrara sesión y le retornara a la pantalla de inicio.
<b>Camino de Excepción</b>	No ingresar a la aplicación.

## Diagrama de Caso de Uso



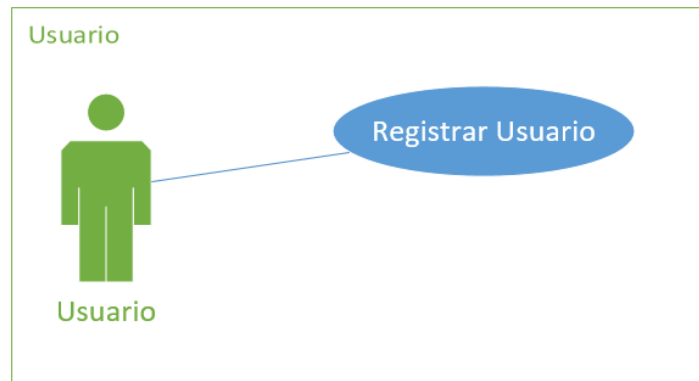


## Diagrama de Clase

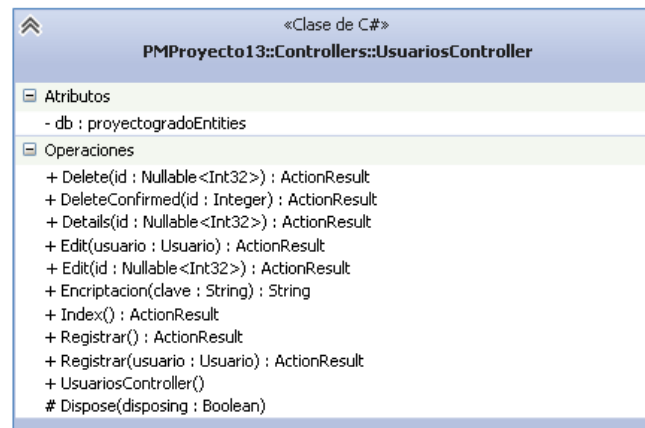


<b>Identificación</b>	PMB-003
<b>Nombre</b>	Registrar Usuario
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Este requerimiento consiste en la creación de un usuario para acceder a la aplicación.
<b>Precondición</b>	Acceder a la aplicación.
<b>Postcondición</b>	El usuario será creado en la base de datos.
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. La pantalla de inicio tendrá los botones de “Iniciar Sesión” y “Registrar Usuario”, se dará clic en la segunda opción.</li> <li>3. Una ventana modal aparecerá con un formulario de datos a llenar.</li> <li>4. Crear el usuario brindando los siguientes datos: Nombre, Apellido, Usuario, Contraseña, Ciudad y Rol en el proyecto.</li> <li>5. Si ingresa los datos de manera correcta, la ventana se cerrará y el usuario será creado en sistema.</li> <li>6. Si se desea se podrá Ingresar usuario y contraseña para usar la aplicación.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña.

## Diagrama de caso de Uso



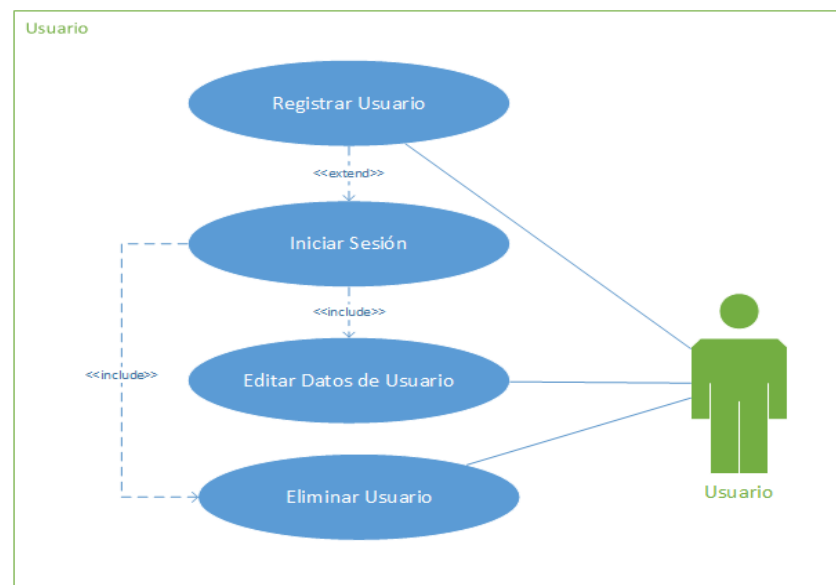
## Diagrama de Clase



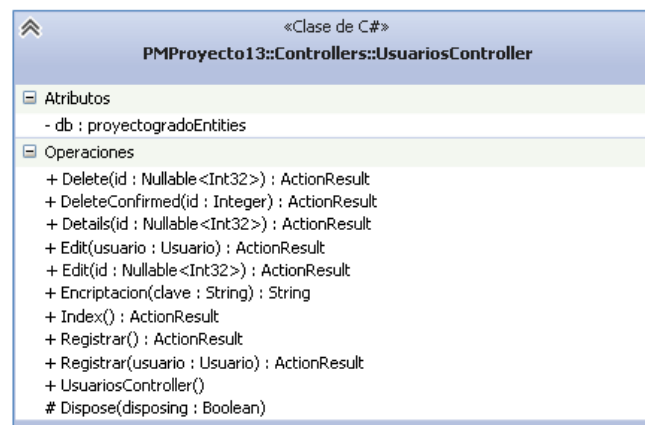
<b>Identificación</b>	PMB-004
<b>Nombre</b>	Gestionar Usuario
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Permite que los datos básicos del usuario puedan ser editados o se pueda eliminar el usuario en caso de que sea necesario.
<b>Precondición</b>	Acceder a la aplicación. Iniciar sesión.
<b>Postcondición</b>	Se guardaran los datos del usuario editados
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar el usuario y la contraseña.</li> <li>3. Si ingresa correctamente el usuario busca la opción de "Editar Datos"</li> </ol>

		<p>4. Se cargara una ventana donde podrá editar datos tales como su contraseña y su nombre de usuario.</p> <p>5. Si el usuario requiere eliminar sus datos del sistema, tendrá la opción para hacerlo; esa acción eliminara todo lo relacionado con este.</p>
<b>Camino de Excepción</b>	<b>de</b>	<p>No Acceder a la aplicación.</p> <p>No poseer un usuario registrado en la aplicación</p>

### Diagrama de caso de uso

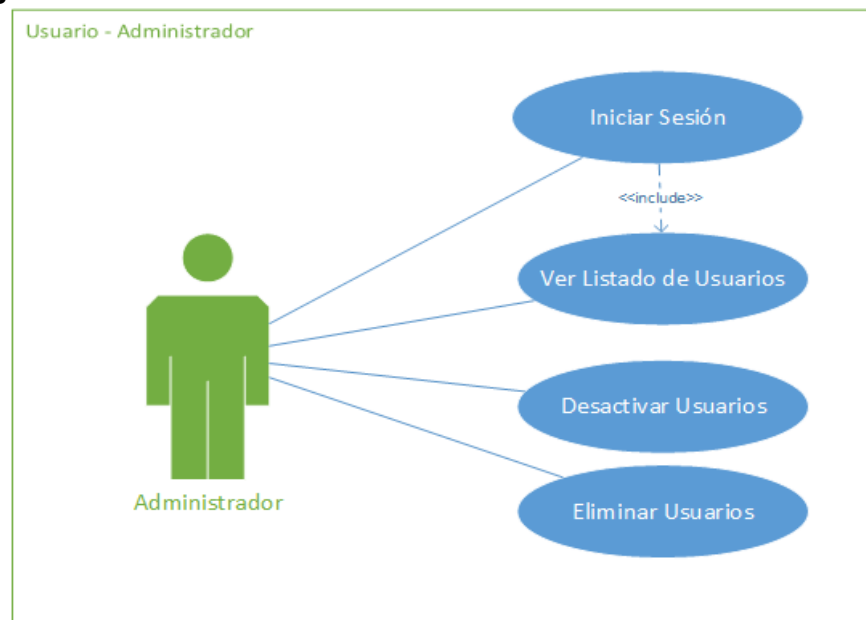


### Diagrama de Clase



<b>Identificación</b>	PMB-005
<b>Nombre</b>	Ver listado de usuarios
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se puede visualizar todo un listado de usuarios para que el administrador de la aplicación tenga un registro y opciones sobre estos.
<b>Precondición</b>	Acceder a la aplicación. Ingresar Usuario y Contraseña.
<b>Postcondición</b>	Cerrar Aplicación
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Entrar al módulo de administración de usuarios.</li> <li>3. Ver el listado de usuarios registrados en la aplicación.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. Carecer de permisos necesarios para esta acción. No tener usuario y contraseña.

#### Diagrama de caso de Uso



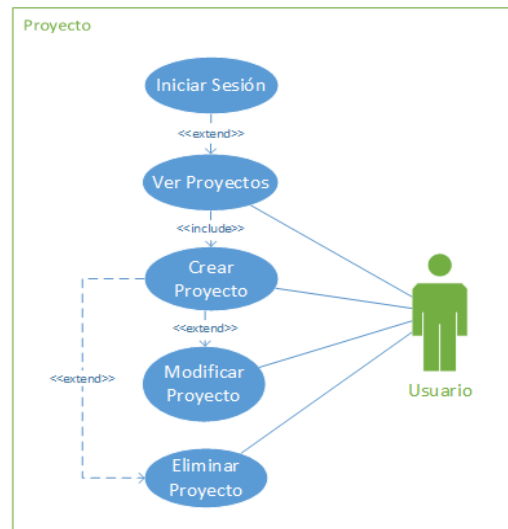
#### Diagrama de clase



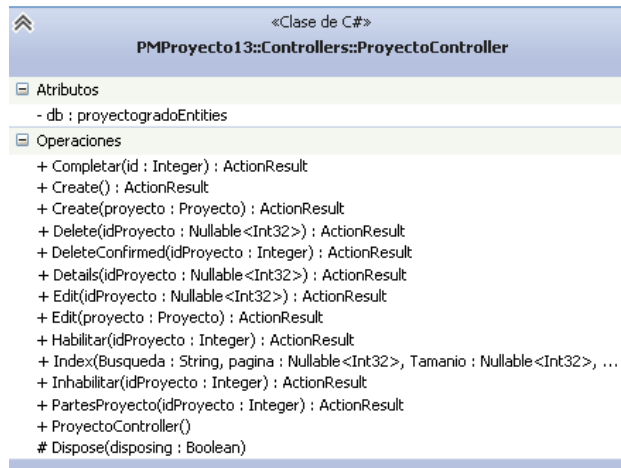
<b>Identificación</b>	PMB-006
<b>Nombre</b>	Gestionar Proyectos.
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Al ingresar a la aplicación, el usuario vera un listado de los proyectos que ha creado, y una serie de acciones que le permitirá ver las tareas relacionadas con este entre otras acciones.
<b>Precondición</b>	Acceder a la aplicación. Ingresar Usuario y Contraseña.
<b>Postcondición</b>	Cerrar Aplicación
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña</li> <li>3. Si el usuario necesita crear un proyecto, lo hará con el botón de “crear proyecto” que desplegara una vista con la información necesaria.</li> <li>4. El usuario vera el listado de los proyectos que se encuentran asociados a este junto con acciones que le permitirán editar la información, borrar es proyecto, compartirlo o acceder a sus componentes.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña.

	En caso de no tener proyectos creados aparecerá la advertencia de que no se han creado proyectos.
--	---

### Diagrama de Caso de Uso



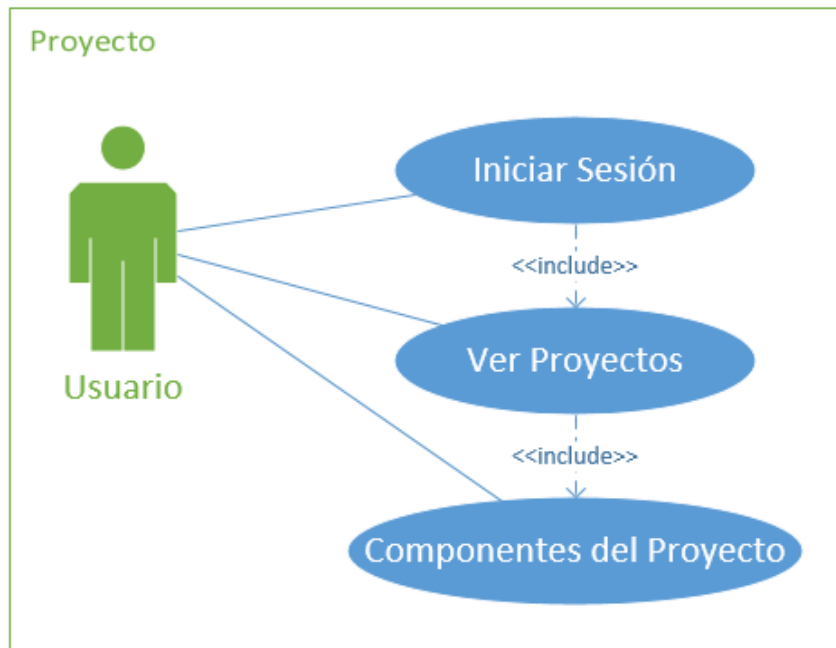
### Diagrama de Clase



<b>Identificación</b>	PMB-007
<b>Nombre</b>	Ver Componentes del Proyecto
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se visualiza en una sola página los objetos que componen el proyecto (Tareas, Hitos, Recursos, incidencias, Interesados)
<b>Precondición</b>	Acceder a la aplicación.

	Tener proyectos existentes en la base de datos.
<b>Postcondición</b>	Ninguna
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar nombre de Usuario y Contraseña.</li> <li>3. Se cargara el listado de todos los proyectos asociados al usuario.</li> <li>4. En la vista que posee el listado de todos los proyectos buscara el botón con las opciones de proyecto y buscara la opción para ver los componentes.</li> <li>5. Se cargara el listado con los objetos que están relacionados con el proyecto.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No tener proyectos creados.

#### Diagrama de caso de uso



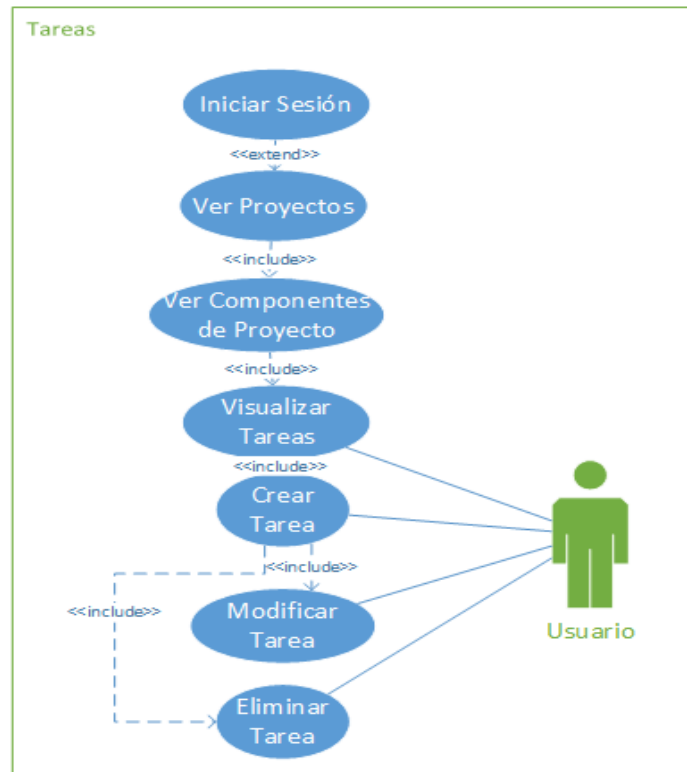
#### Diagrama de clase



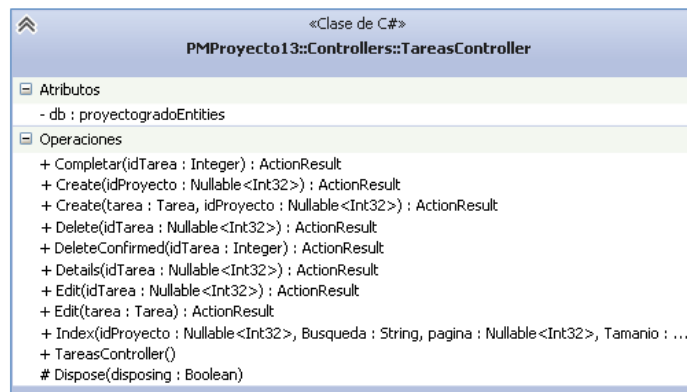


<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No tener los permisos creados No existan proyectos creados.
----------------------------	---

### Diagrama de Caso de Uso



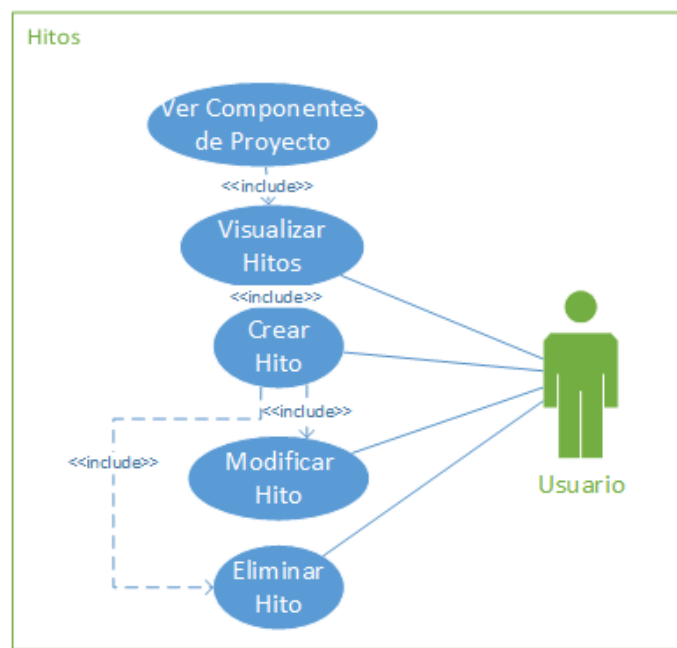
### Diagrama de Clase



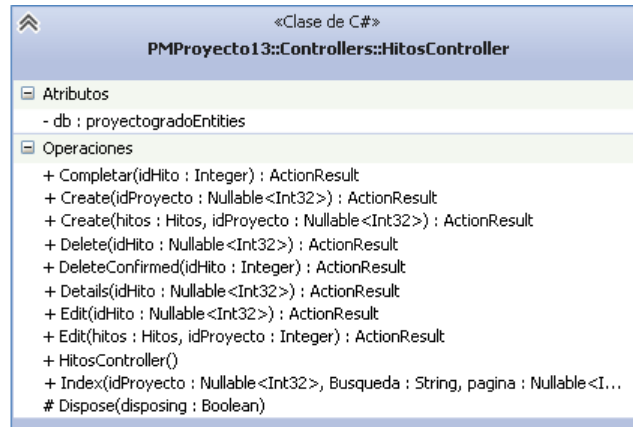
<b>Identificación</b>	PMB-009
<b>Nombre</b>	Gestionar Hitos

<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se visualizaran los hitos relacionados a un proyecto.
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.
<b>Postcondición</b>	Visualizar hitos asociados a un proyecto Cerrar Aplicación
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario ira a la ventana de componentes del proyecto</li> <li>5. Se cargaran los paneles con objetos del proyecto entre ellos un panel relacionado con los hitos.</li> <li>6. En el panel anterior el usuario podrá interactuar con los hitos existentes, crear nuevos o modificarlos.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos creados.

#### Diagrama de caso de uso

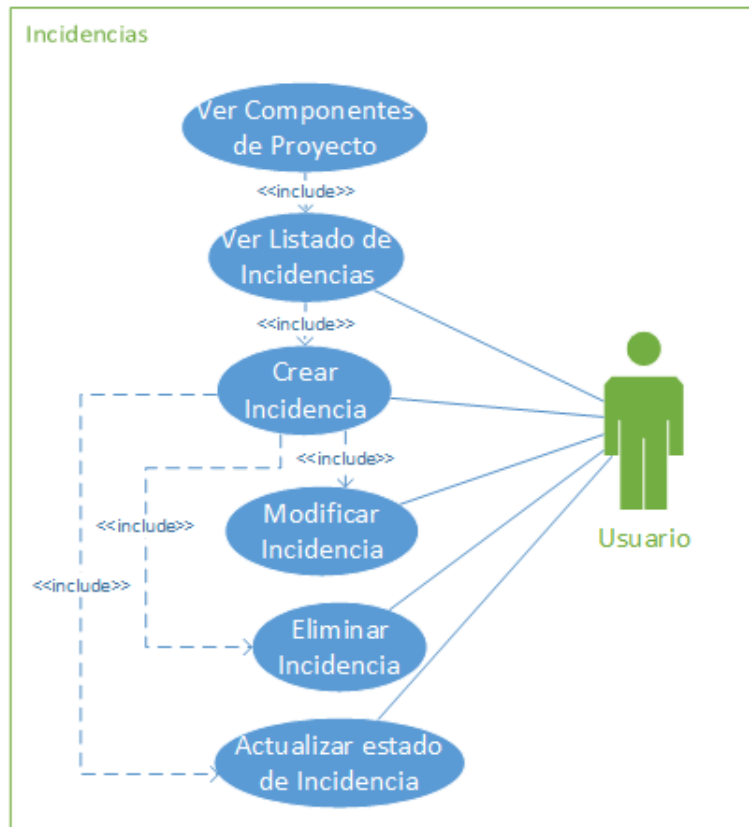


## Diagrama de clase

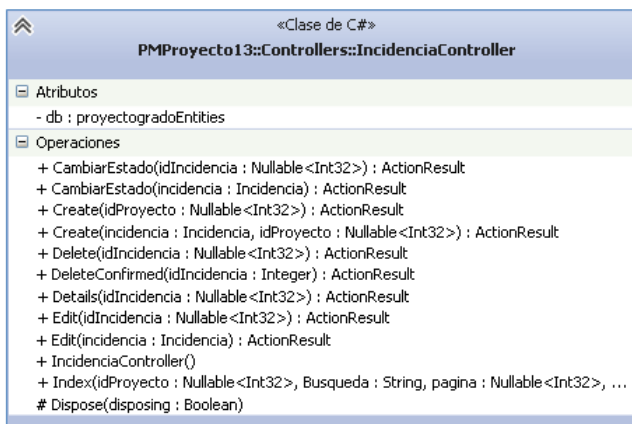


<b>Identificación</b>	PMB-010
<b>Nombre</b>	Ver Listado de Incidencias
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se tiene una interfaz que permite la visualización de las incidencias que se encuentran asociadas a un proyecto.
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.
<b>Postcondición</b>	Visualizar tareas asociadas Cerrar Aplicación
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario hará clic en el botón "Incidencias"</li> <li>5. Se cargara un listado con las incidencias asociadas al proyecto, junto con las opciones para interactuar con estas.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos creados.

## Diagrama de Caso de Uso



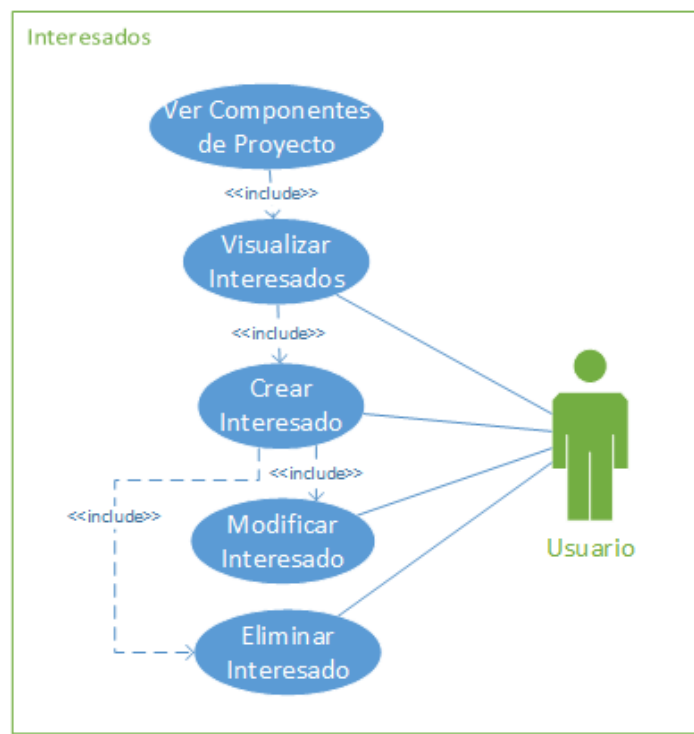
## Diagrama de Clase



<b>Identificación</b>	PMB-011
<b>Nombre</b>	Visualizar Interesados
<b>Tipo</b>	Funcional.

<b>Resumen</b>	Se tiene una ventana donde se encuentra un listado con los interesados de un proyecto.
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.
<b>Postcondición</b>	Visualizar tareas asociadas Cerrar Aplicación
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario hará clic en el botón “Interesados”</li> <li>5. Se cargara un listado con las incidencias asociadas al proyecto en caso de que hayan sido creados previamente, junto con las opciones para interactuar con estas.</li> </ol>
<b>Camino de Excepción</b>	de No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos creados.

#### Diagrama de Caso de uso

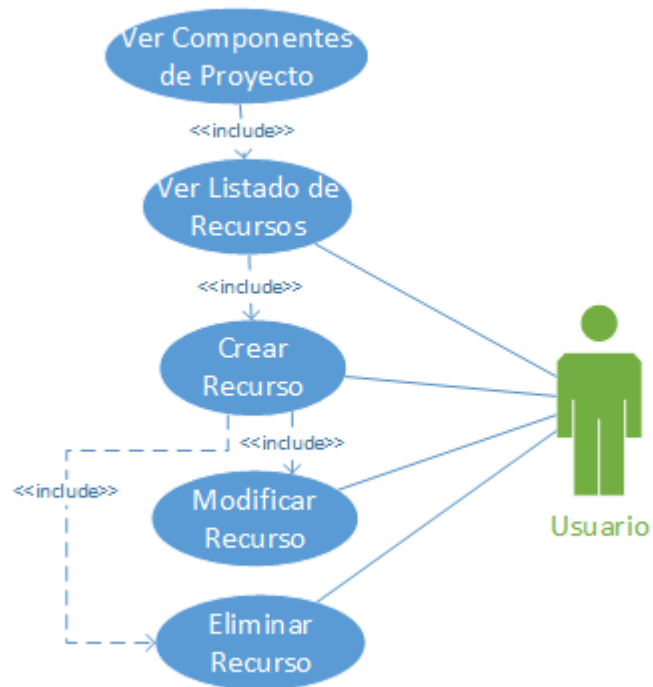


## Diagrama de Clase

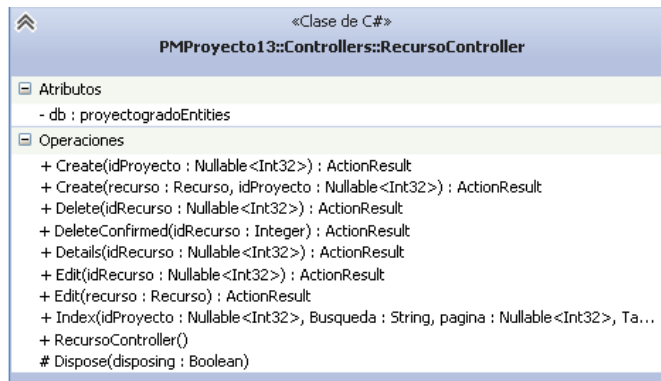


<b>Identificación</b>	PMB-012
<b>Nombre</b>	Ver Listado de Recursos
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Es necesario brindar un listado de los recursos disponibles para un proyecto
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.
<b>Postcondición</b>	Visualizar tareas asociadas Cerrar Aplicación
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario hará clic en el botón "Recursos"</li> <li>5. Se cargara un listado con los recursos asociados al proyecto, junto con las opciones para interactuar con estas.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos creados.
<b>Diagrama de Caso de Uso</b>	

## Recursos

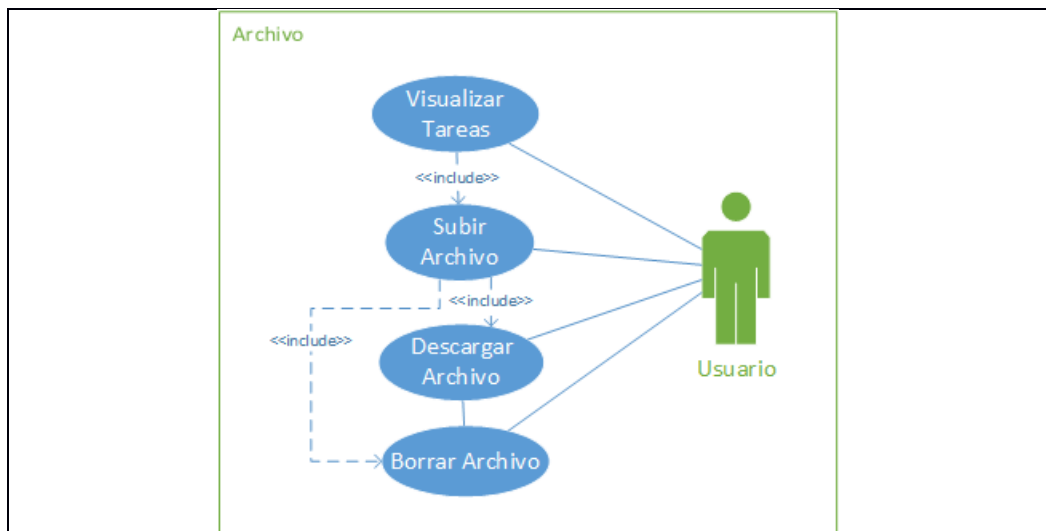


## Diagrama de clase

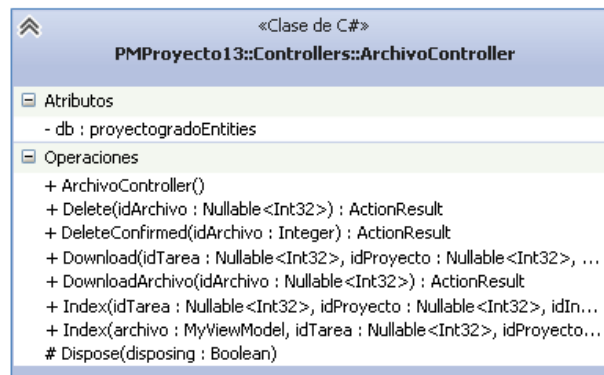


<b>Identificación</b>	PMB-013
<b>Nombre</b>	Subir Archivo
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Este requerimiento permite que los usuarios puedan subir un archivo asociado a una tarea y quede almacenado en base de datos para su posterior descarga en el equipo.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados Existan tareas creadas
<b>Postcondición</b>	El archivo será subido y guardado en la base de datos
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar a la aplicación con un nombre de usuario y contraseña válidos.</li> <li>3. La aplicación cargara un listado con los proyectos asociados a un usuario.</li> <li>4. El usuario hará clic en el botón “Ver Tareas” donde cargara las tareas asociadas a dicha tarea.</li> <li>5. Cargara un listado con las tareas asociadas, allí se encontrara un botón con el nombre “Subir Archivo” al hacer clic este le llevara a la interfaz para subir el archivo.</li> <li>6. El usuario seleccionara el archivo y el sistema cargara y guardara el archivo en la base de datos de la aplicación.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No tener documentos para subir.
<b>Diagrama de caso de uso</b>	

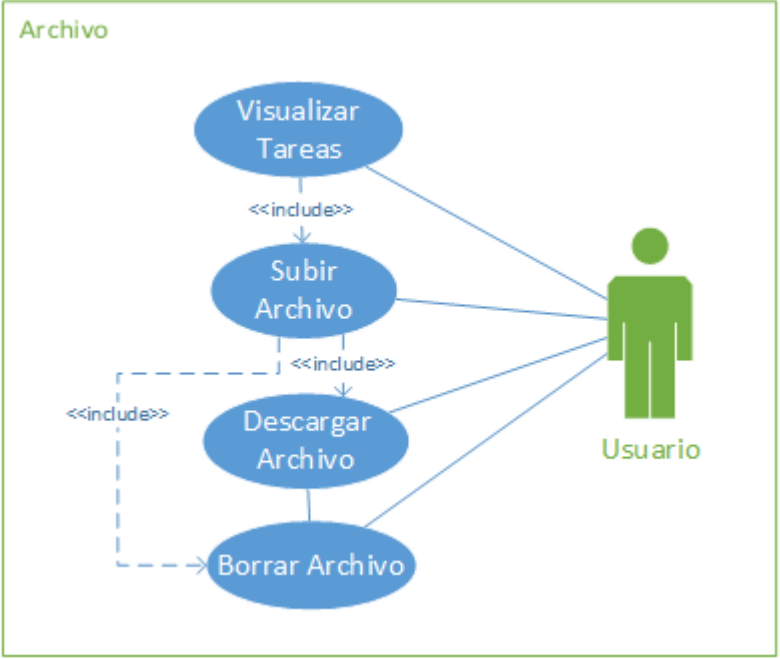


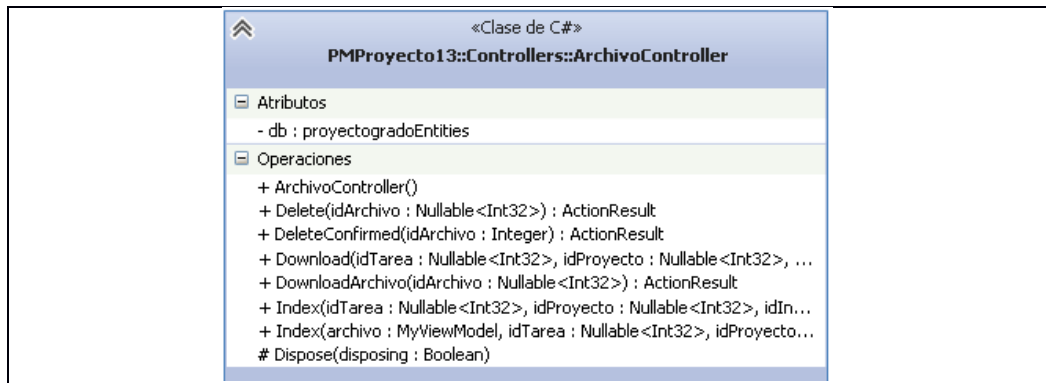


### Diagrama de clase



<b>Identificación</b>	PMB-014
<b>Nombre</b>	Descargar Archivo
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Al tener archivos subidos en la base de datos y asociados a una tarea, los usuarios que puedan ver estos proyectos podrán descargarlos en su equipo y abrirlos.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados Existan tareas creadas Deben existir archivos subidos.
<b>Postcondición</b>	El archivo será subido y guardado en la base de datos

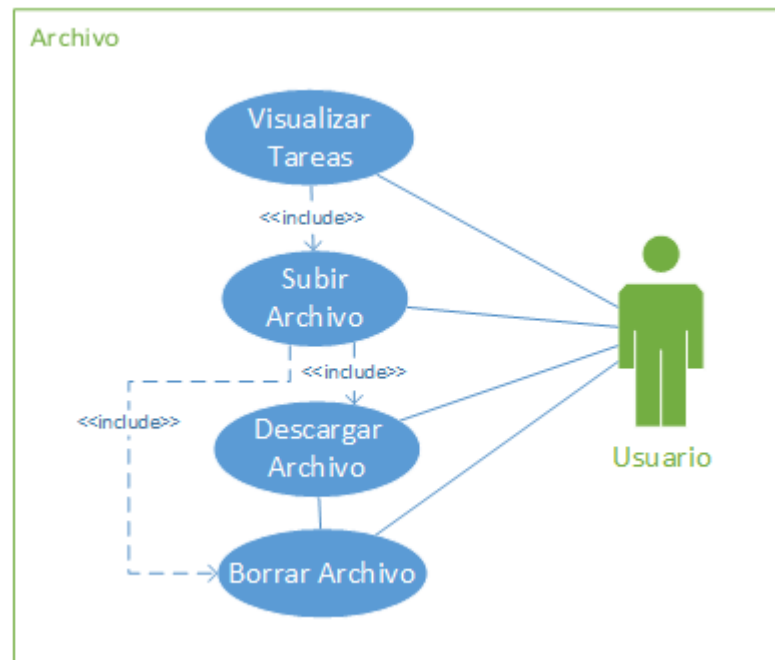
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar a la aplicación con un nombre de usuario y contraseña válidos.</li> <li>3. La aplicación cargara un listado con los proyectos asociados a un usuario.</li> <li>4. El usuario hará clic en el botón “Ver Tareas” donde cargara las tareas asociadas a dicha tarea.</li> <li>5. Cargara un listado con las tareas asociadas, allí se encontrara un botón con el nombre “Ver Archivos” donde se encontraran los archivos asociados a esa tarea.</li> <li>6. El usuario seleccionara hará clic en el botón descargar que se encuentra al lado del nombre del archivo.</li> <li>7. El archivo se descargara en el equipo del usuario.</li> </ol>
<b>Camino de Excepción</b>	de No Acceder a la aplicación. No tener usuario y contraseña. No existan documentos asociados a esa tarea.
<b>Diagrama de caso de uso</b>  <pre> graph TD     subgraph Archivo         VT([Visualizar Tareas])         SA([Subir Archivo])         DA([Descargar Archivo])         BA([Borrar Archivo])     end     Usuario((Usuario))     VT --&gt; SA     SA --&gt; DA     DA --&gt; BA     BA -.-&gt; BA     VT -.-&gt; Usuario     SA -.-&gt; Usuario     DA -.-&gt; Usuario     BA -.-&gt; Usuario     VT -.-&gt; SA : &lt;&lt;include&gt;&gt;     SA -.-&gt; DA : &lt;&lt;include&gt;&gt;     DA -.-&gt; BA : &lt;&lt;include&gt;&gt;     BA -.-&gt; BA : &lt;&lt;include&gt;&gt;         </pre>	
<b>Diagrama de clase</b>	



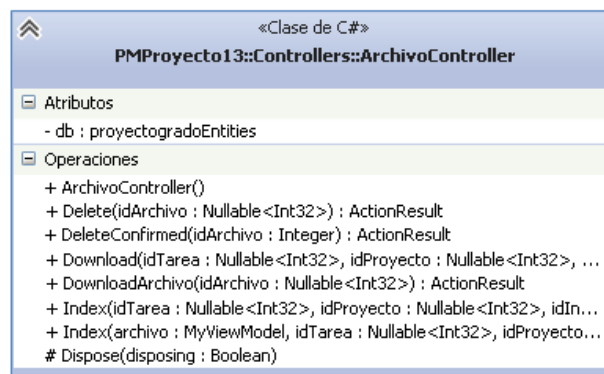
<b>Identificación</b>	PMB-015
<b>Nombre</b>	Borrar Archivo
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Al tener archivos subidos en la base de datos y asociados a una tarea, los usuarios que puedan ver estos proyectos podrán descargarlos en su equipo y abrirlos.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados Existan tareas creadas Deben existir archivos previamente creados.
<b>Postcondición</b>	El archivo será subido y guardado en la base de datos
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar a la aplicación con un nombre de usuario y contraseña válidos.</li> <li>3. La aplicación cargara un listado con los proyectos asociados a un usuario.</li> <li>4. El usuario hará clic en el botón “Ver Tareas” donde cargara las tareas asociadas a dicha tarea.</li> <li>5. Cargara un listado con las tareas asociadas, allí se encontrara un botón con el nombre “Ver Archivos” donde se encontraran los archivos asociados a esa tarea.</li> </ol>

		<p>6. El usuario seleccionara hará clic en el botón borrar al lado del botón de descarga.</p> <p>7. Una ventana modal se ejecutara pidiendo la confirmación de la acción.</p>
<b>Camino</b>	<b>de</b>	No Acceder a la aplicación.
<b>Excepción</b>		No tener usuario y contraseña. No existan documentos asociados a esa tarea.

#### Diagrama de Caso de uso

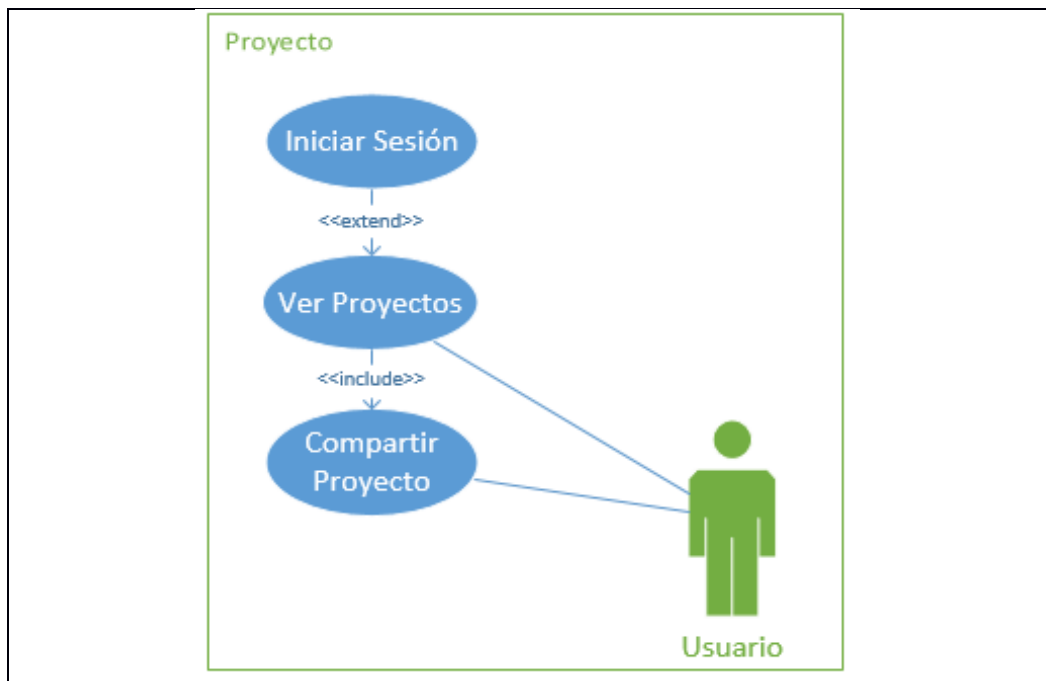


#### Diagrama de clase

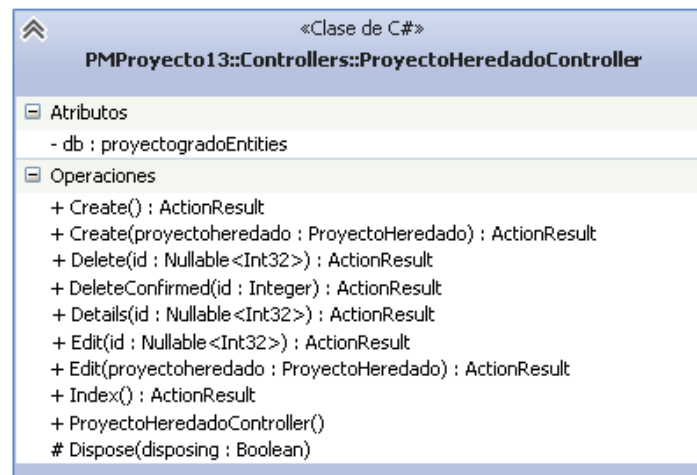


<b>Identificación</b>	PMB-016
<b>Nombre</b>	Compartir Proyecto

<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se puede asociar un proyecto creado a un usuario existente en sistema, este podrá acceder a las mismas áreas a las que accede el mismo usuario que lo compartió, y a su vez se le pueden asignar permisos sobre las operaciones que puede realizar el usuario a quien se le comparte el proyecto.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados
<b>Postcondición</b>	El proyecto será compartido y visualizado por el otro usuario.
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar a la aplicación con un nombre de usuario y contraseña válidos.</li> <li>3. La aplicación cargara un listado con los proyectos asociados a un usuario.</li> <li>4. El usuario hará clic en el botón "Compartir Proyecto".</li> <li>5. Se cargara una ventana con el listado de usuarios existentes en la aplicación, seleccionado aquel con el que desea compartir el proyecto.</li> <li>6. El usuario al cual se compartirá el proyecto podrá visualizarlo.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos. El usuario no exista.
<b>Diagrama de Caso de uso</b>	



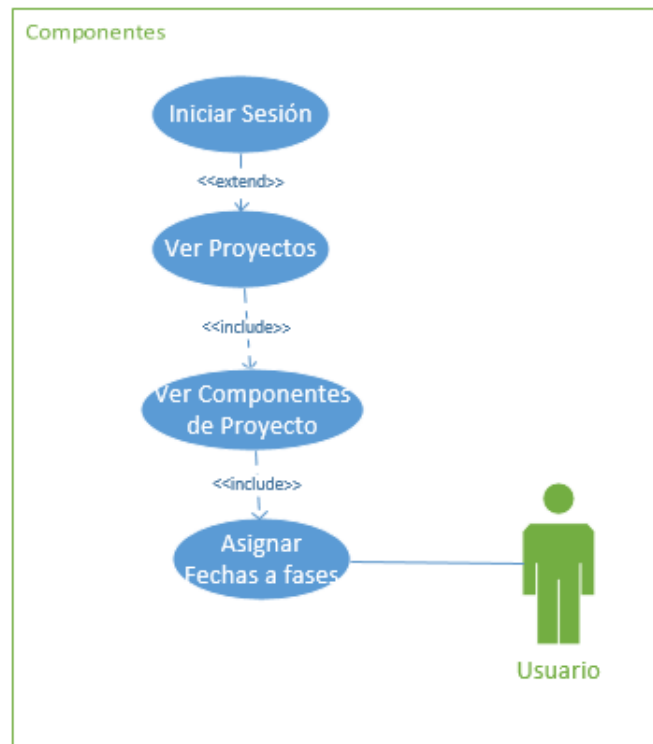
### Diagrama de clase



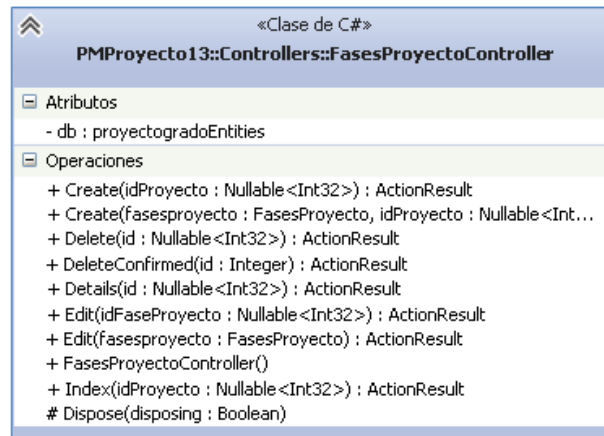
<b>Identificación</b>	PMB-017
<b>Nombre</b>	Agregar Fechas para Fases
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Es necesario brindarle a un proyecto las fechas en las que sucederán sus fases
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados

<b>Postcondición</b>	El proyecto será compartido y visualizado por el otro usuario.
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar a la aplicación con un nombre de usuario y contraseña válidos.</li> <li>3. La aplicación cargara un listado con los proyectos asociados a un usuario.</li> <li>4. El usuario hará clic en el botón "Componentes"</li> <li>5. Se cargara todos los objetos relacionados al proyecto seleccionado.</li> <li>6. Se busca el panel de fases.</li> <li>7. Se crea una nueva fecha de inicio y fin para la fase del proyecto seleccionada.</li> <li>8. Se Guardan los cambios.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos. El usuario no exista.

#### Diagrama de Caso de uso



## Diagrama de clase

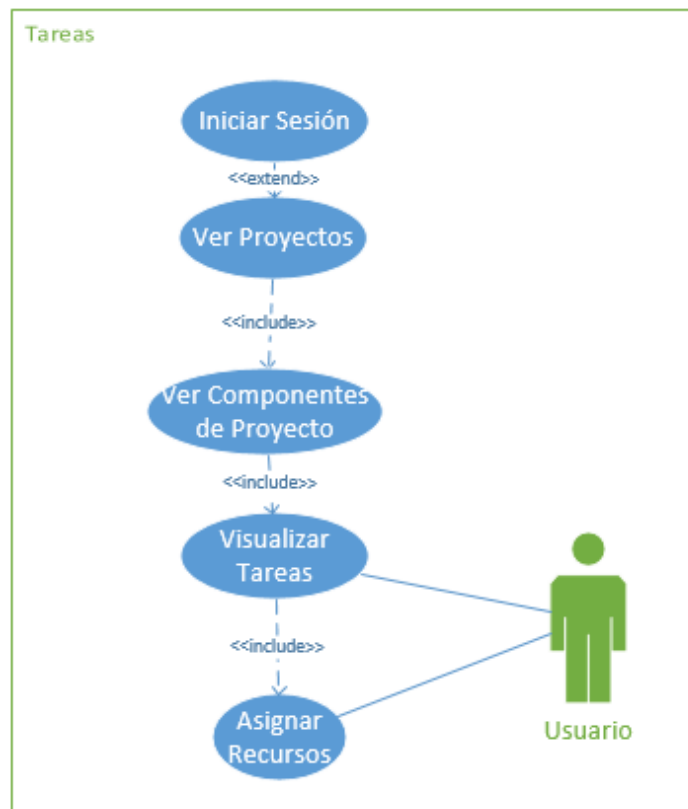


<b>Identificación</b>	PMB-018
<b>Nombre</b>	Asignar Recursos a Tarea
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Un proyecto posee recursos que pueden ser asignados a una tarea.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados Tener recursos creados en la aplicación
<b>Postcondición</b>	Se visualizaran los recursos que estén asignados para esa tarea, restándolos de los recursos que existen.
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar a la aplicación con un nombre de usuario y contraseña válidos.</li> <li>3. La aplicación cargara un listado con los proyectos asociados a un usuario.</li> <li>4. El usuario hará clic en el botón "Componentes"</li> <li>5. Se ingresa al panel de tareas</li> <li>6. Se hace clic en el botón "Asignar recursos"</li> <li>7. Se selecciona el recurso y la cantidad</li> <li>8. Los recursos quedaran asignados</li> </ol>

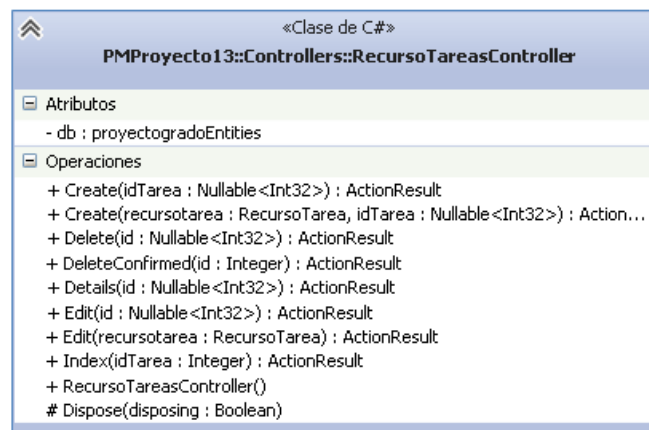


<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos. No existan recursos. No existan tareas.
----------------------------	--

### Diagrama de caso de uso

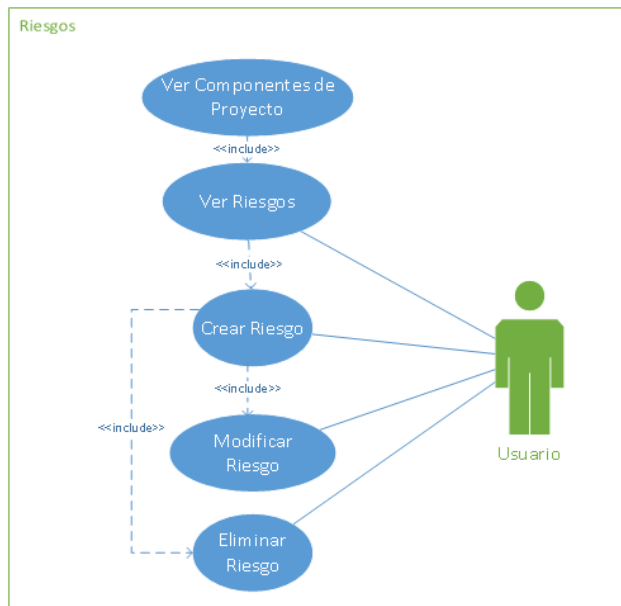


### Diagrama de clase



<b>Identificación</b>	PMB-019
<b>Nombre</b>	Ver Riesgos
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se pueden ingresar los riesgos a los que puede estar sujeto el proyecto
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.
<b>Postcondición</b>	Visualizar el riesgo que se ha ingresado en sistema para interactuar con este.
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario hace clic en el botón "componentes"</li> <li>5. Se cargaran los paneles con los componentes asociados a este proyecto, cargando consigo el listado de riesgos si los hay.</li> </ol>
<b>Camino de Excepción</b>	de No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos creados.

#### Diagrama de caso de uso



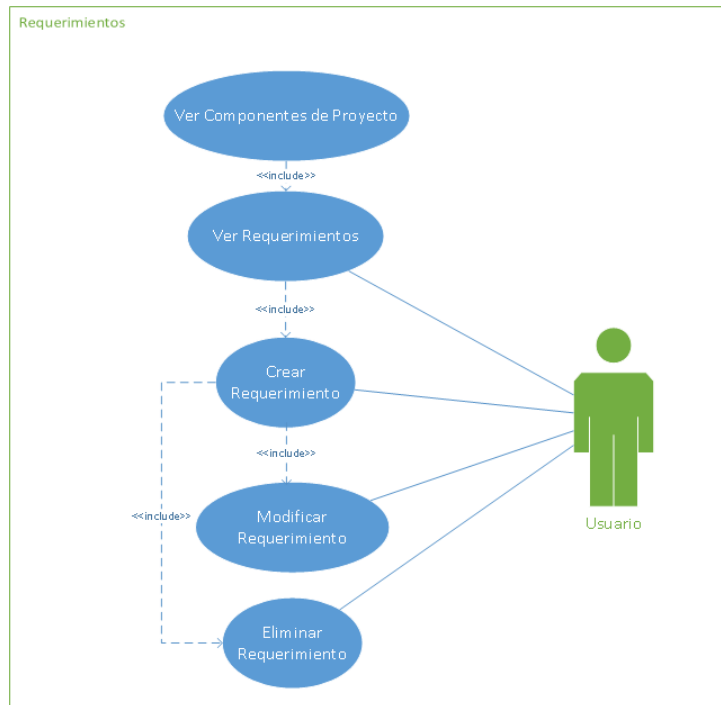
## Diagrama de clase



<b>Identificación</b>	PMB-020
<b>Nombre</b>	Ver Requerimientos
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se pueden ingresar los requerimientos que se soliciten para un proyecto.
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.
<b>Postcondición</b>	Visualizar el requerimiento que se ha creado en sistema
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario hace clic en el botón "componentes"</li> <li>5. Se cargaran los paneles con los componentes asociados a este proyecto, cargando consigo el listado de requerimientos si los hay.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña.

No existan proyectos creados.

## Diagrama de caso de uso



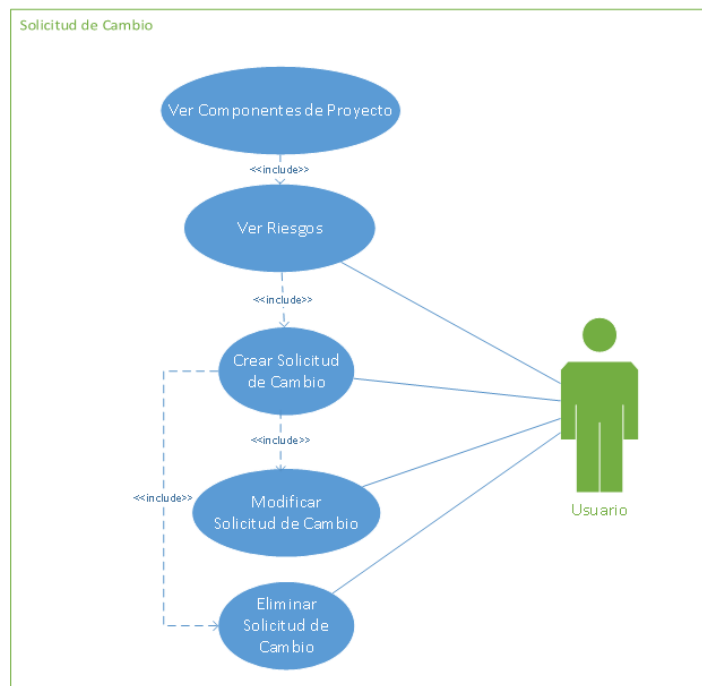
## Diagrama de clase



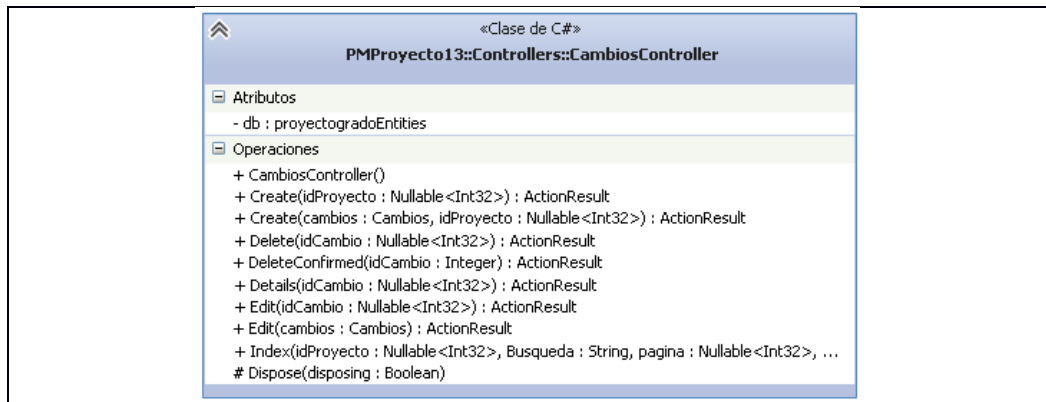
<b>Identificación</b>	PMB-021
<b>Nombre</b>	Ver Cambios del proyecto
<b>Tipo</b>	Funcional.
<b>Resumen</b>	Se pueden ingresar los cambios que se soliciten para un proyecto.
<b>Precondición</b>	Acceder a la aplicación. Tener uno o más proyectos existentes.

<b>Postcondición</b>	Visualizar los cambios para el proyecto que se ha creado en sistema
<b>Flujo de Eventos</b>	<ol style="list-style-type: none"> <li>1. Acceder a la aplicación.</li> <li>2. Ingresar usuario y contraseña.</li> <li>3. Se cargara el listado con todos los proyectos asociados al usuario.</li> <li>4. El usuario hace clic en el botón “componentes”</li> <li>5. Se cargaran los paneles con los componentes asociados a este proyecto, cargando consigo el listado de cambios en caso de que los haya.</li> </ol>
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña. No existan proyectos creados.

#### Diagrama de casos de uso



#### Diagrama de clase



<b>Identificación</b>	PMB-022
<b>Nombre</b>	Diseño de Aplicación
<b>Tipo</b>	No Funcional.
<b>Resumen</b>	El proyecto tiene una plantilla de diseño
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener proyectos creados
<b>Postcondición</b>	El proyecto será compartido y visualizado por el otro usuario.
<b>Flujo de Eventos</b>	1. Acceder a la aplicación. 2. El sistema carga automáticamente la plantilla usada para la aplicación.
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña.

<b>Identificación</b>	PMB-023
<b>Nombre</b>	Buscar
<b>Tipo</b>	No Funcional.
<b>Resumen</b>	Cada módulo de la aplicación tiene una opción para la búsqueda de objetos dentro de cada uno.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener objetos creados
<b>Postcondición</b>	Se muestran los resultados en pantalla según los criterios.

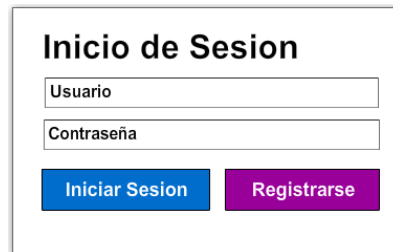
<b>Flujo de Eventos</b>	1. Acceder a la aplicación. 2. El sistema carga con los datos relacionados.
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña.

<b>Identificación</b>	PMB-024
<b>Nombre</b>	Ordenar
<b>Tipo</b>	No Funcional.
<b>Resumen</b>	Cada módulo de la aplicación tiene la capacidad de ordenar los objetos de manera ascendente o descendente.
<b>Precondición</b>	Acceder a la aplicación. Ingresar nombre de usuario y contraseña Tener objetos creados
<b>Postcondición</b>	Se ordenan los objetos según los criterios de ordenamiento.
<b>Flujo de Eventos</b>	1. Acceder a la aplicación. 2. El sistema ordena el modulo según el criterio.
<b>Camino de Excepción</b>	No Acceder a la aplicación. No tener usuario y contraseña.

<b>Identificación</b>	PMB-025
<b>Nombre</b>	Encriptación de contraseñas
<b>Tipo</b>	No Funcional.
<b>Resumen</b>	Cuando se registre un usuario la contraseña con la que ingrese debe estar encriptada para reforzar la seguridad.
<b>Precondición</b>	Registrar el usuario
<b>Postcondición</b>	Se crea un usuario con una contraseña con encriptación.
<b>Flujo de Eventos</b>	1. Registrar el usuario
<b>Camino de Excepción</b>	No Acceder a la aplicación.

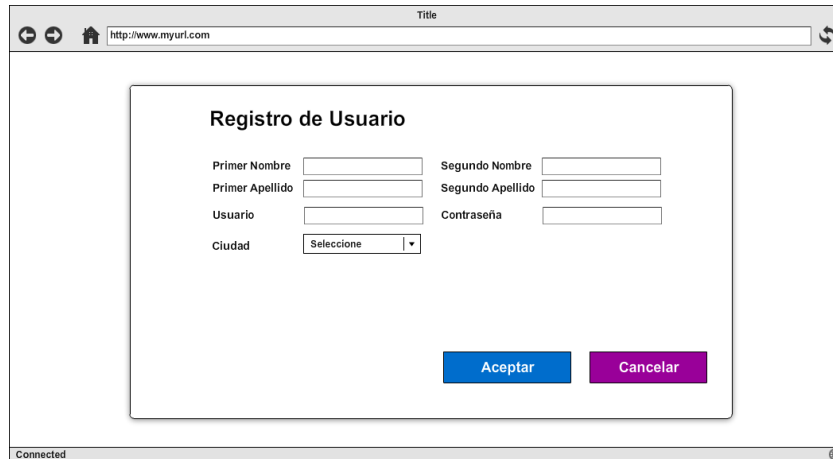
## 4.5. PROTOTIPOS DE INTERFAZ GRAFICA

La metodología ICONIX aconseja tener prototipos de interfaz gráfica que puedan ayudar con la idea que se puede tener del proyecto a desarrollar; a continuación se presentan prototipos de interfaces gráficas para este proyecto.



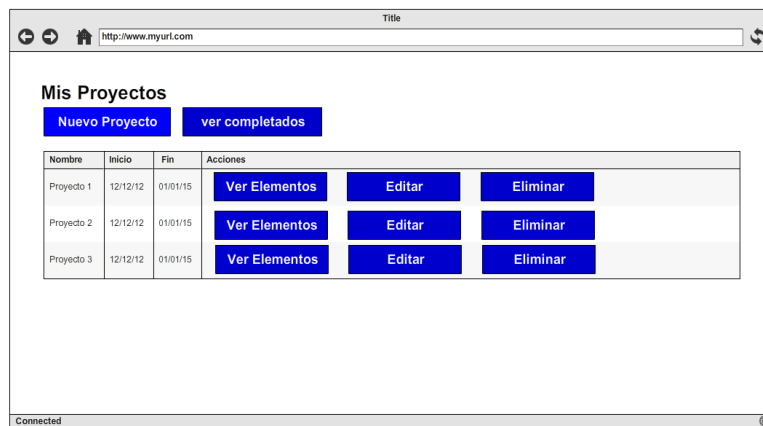
Prototipo de la interfaz de inicio de sesión. El formulario tiene un título "Inicio de Sesión", un campo de texto para "Usuario", un campo de texto para "Contraseña", un botón azul "Iniciar Sesión" y un botón morado "Registrarse".

Ilustración 3 Prototipo Inicio de Sesión



Prototipo de la interfaz de registro de usuario. El formulario está dentro de un navegador web con la URL "http://www.myurl.com". El título es "Registro de Usuario". Los campos incluyen: "Primer Nombre", "Segundo Nombre", "Primer Apellido", "Segundo Apellido", "Usuario", "Contraseña" y un menú desplegable para "Ciudad" con la opción "Seleccione". Hay botones "Aceptar" (azul) y "Cancelar" (morado).

Ilustración 4 Prototipo Registro

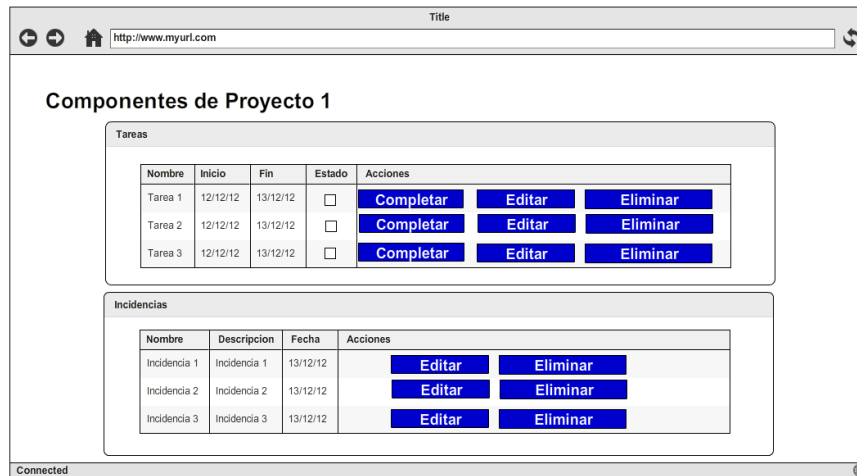


Prototipo de la interfaz de "Mis Proyectos". Muestra un título "Mis Proyectos", dos botones "Nuevo Proyecto" (azul) y "ver completados" (morado), y una tabla de proyectos.

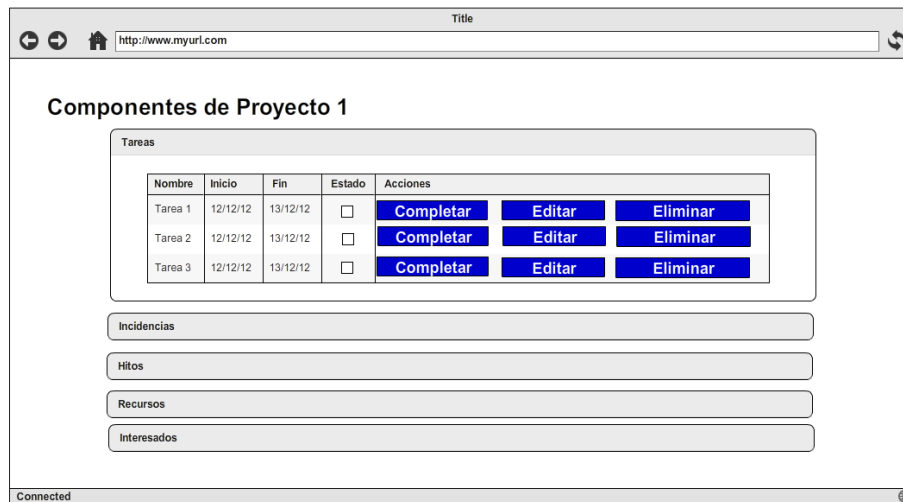
Nombre	Inicio	Fin	Acciones		
Proyecto 1	12/12/12	01/01/15	Ver Elementos	Editar	Eliminar
Proyecto 2	12/12/12	01/01/15	Ver Elementos	Editar	Eliminar
Proyecto 3	12/12/12	01/01/15	Ver Elementos	Editar	Eliminar

Ilustración 5 Prototipo Ver Proyectos





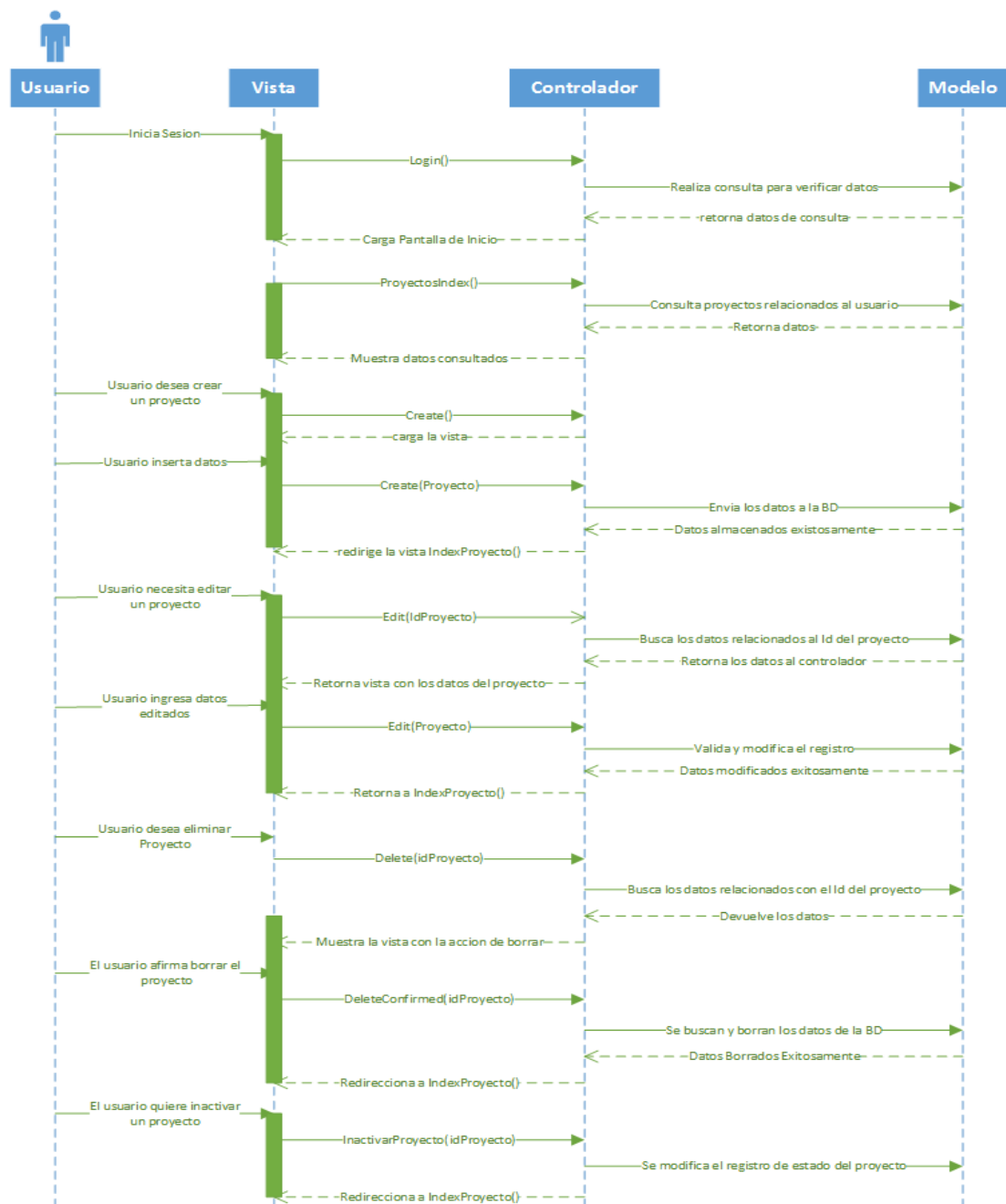
**Ilustración 6 Prototipo Ventana de Componentes del Proyecto**



**Ilustración 7 Prototipo de ventana con paneles**

## 4.6. DIAGRAMAS DE SECUENCIA

Junto con los diagramas de clase, es una buena opción el manejo de diagramas de secuencia, ya que muestra el flujo de eventos y envío de mensajes a cada sitio de la aplicación.



**Ilustración 8 Diagrama de Secuencia para usuarios y Proyectos**

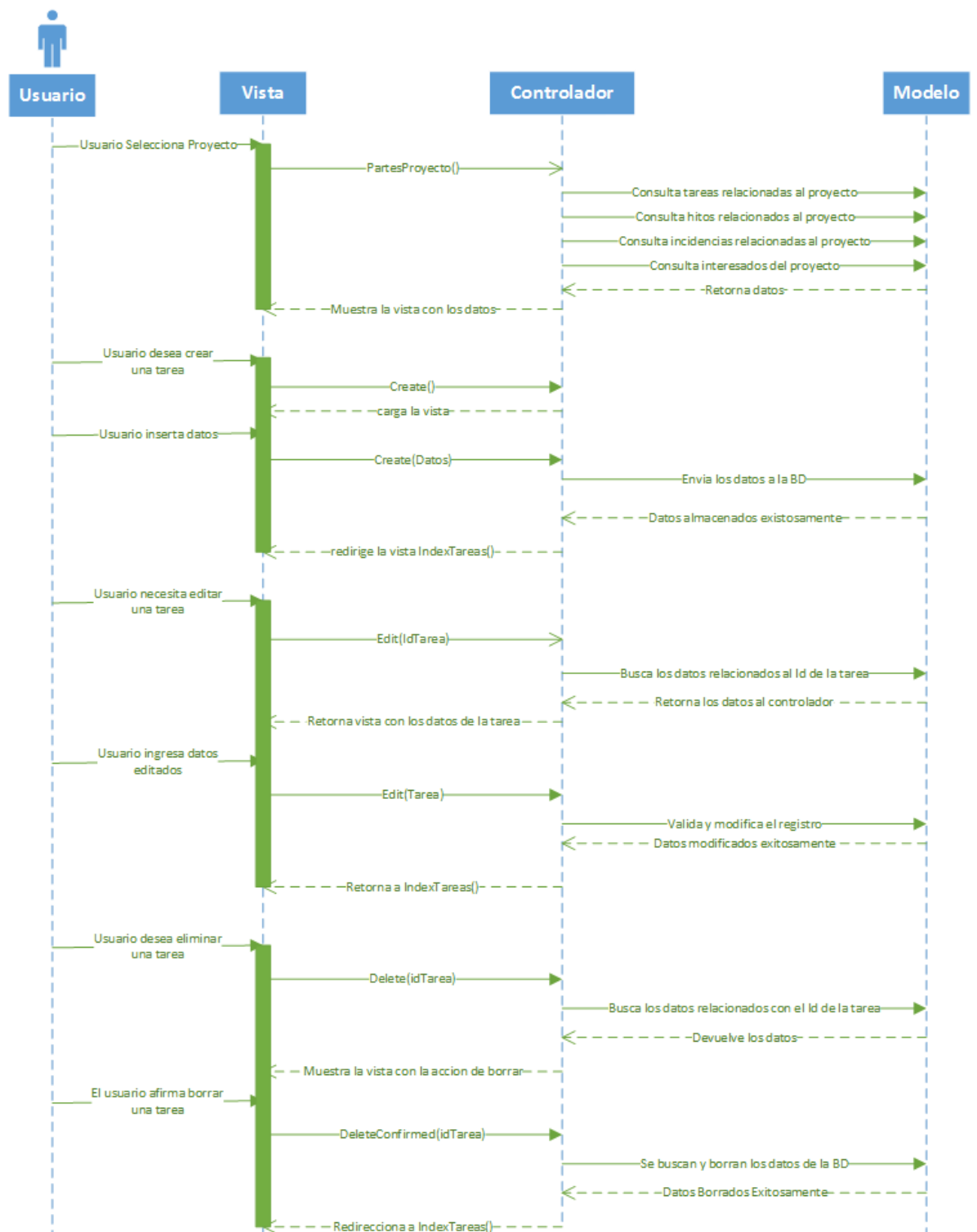


Ilustración 9 Diagrama de secuencia para tareas

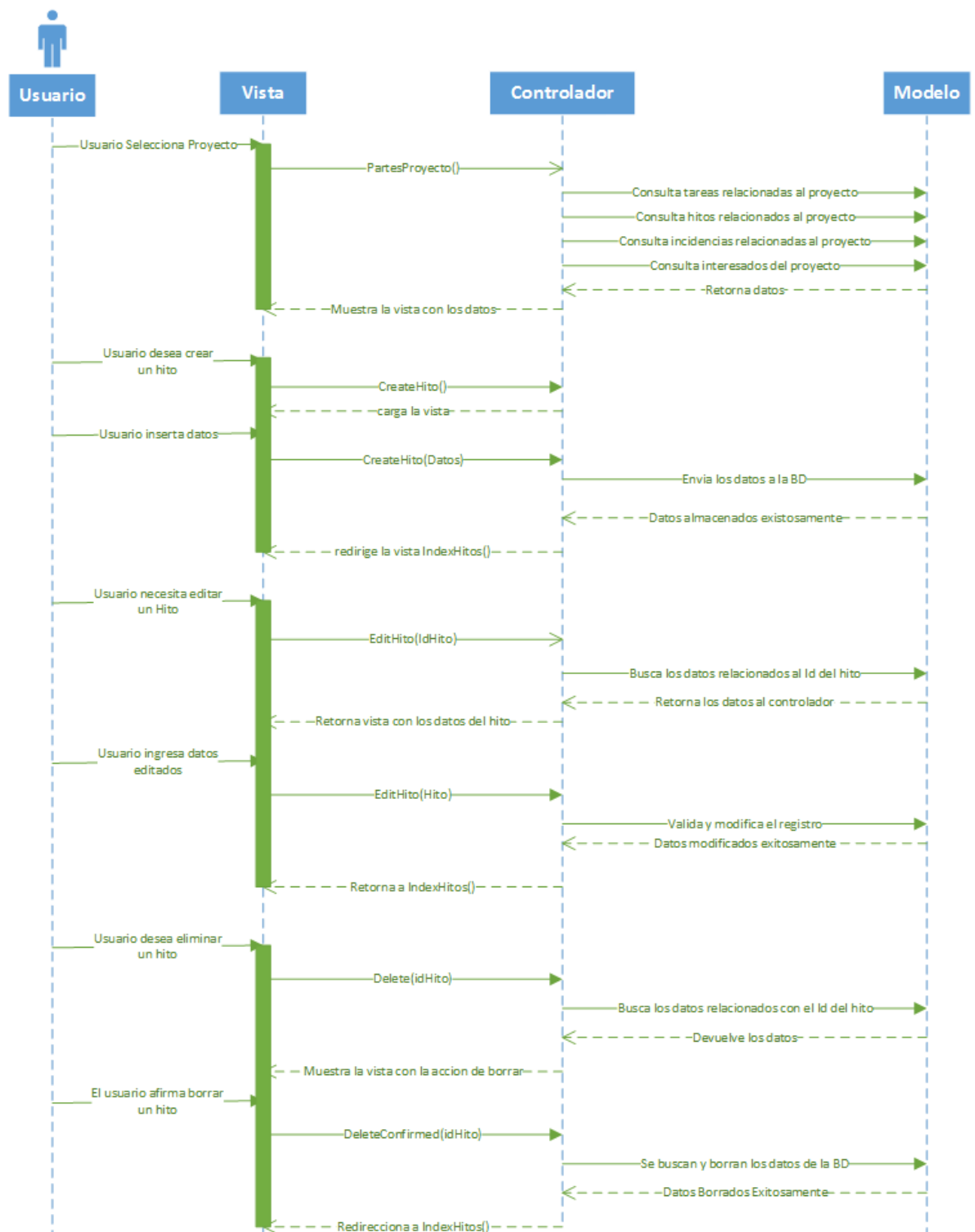


Ilustración 10 Diagrama de secuencia para Hitos

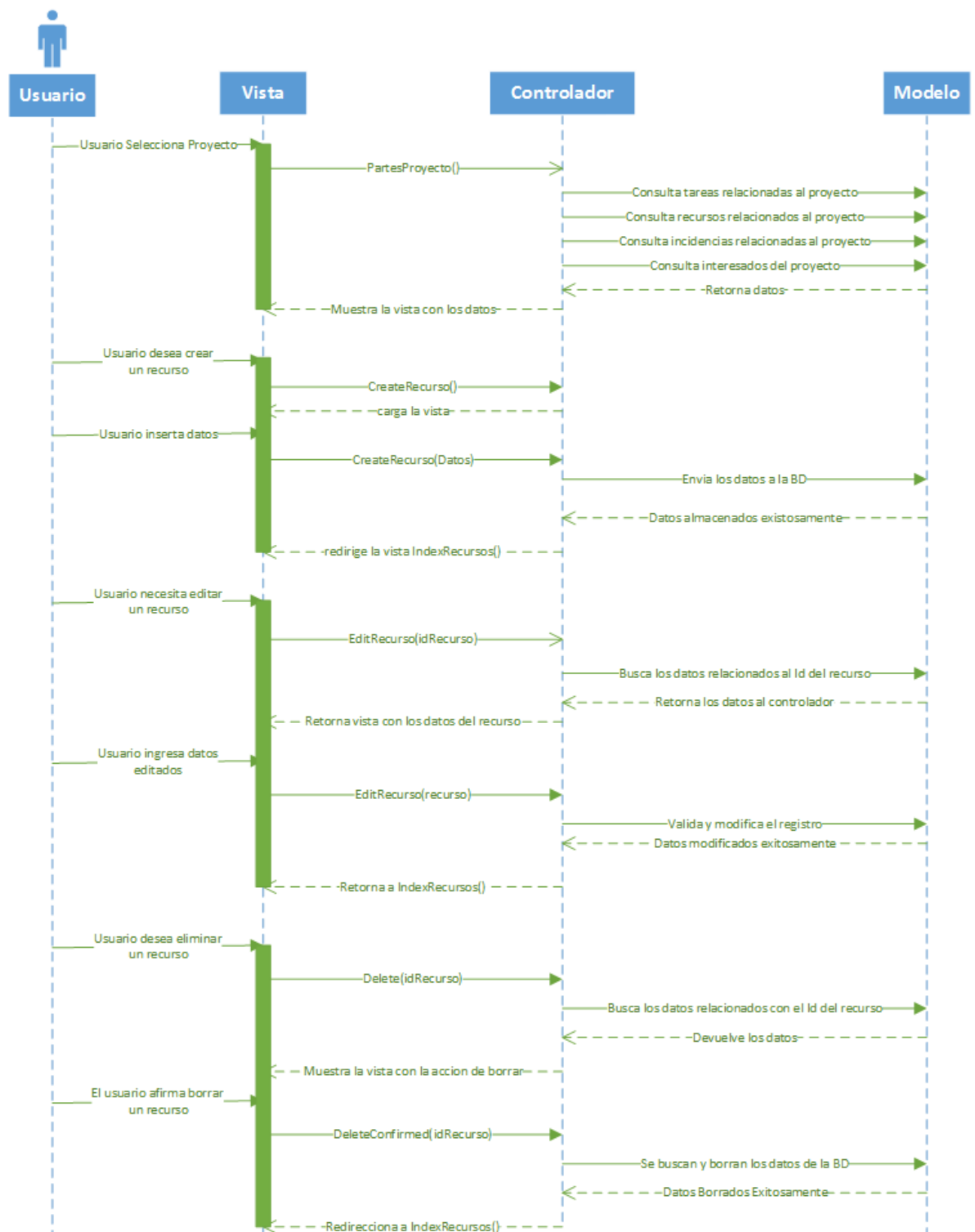


Ilustración 11 Diagrama de Secuencia para Recursos

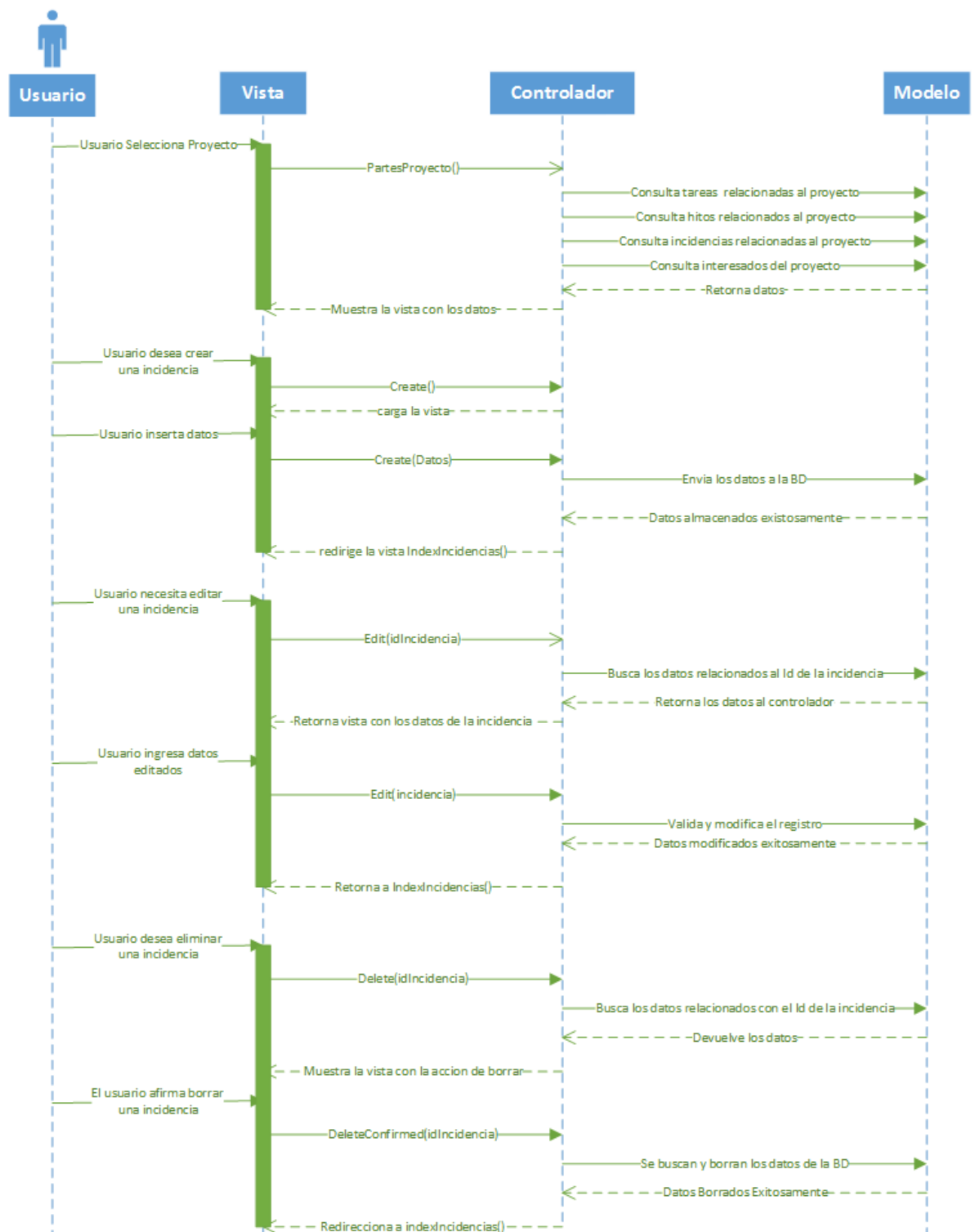


Ilustración 12 Diagrama de Secuencia para Incidencias

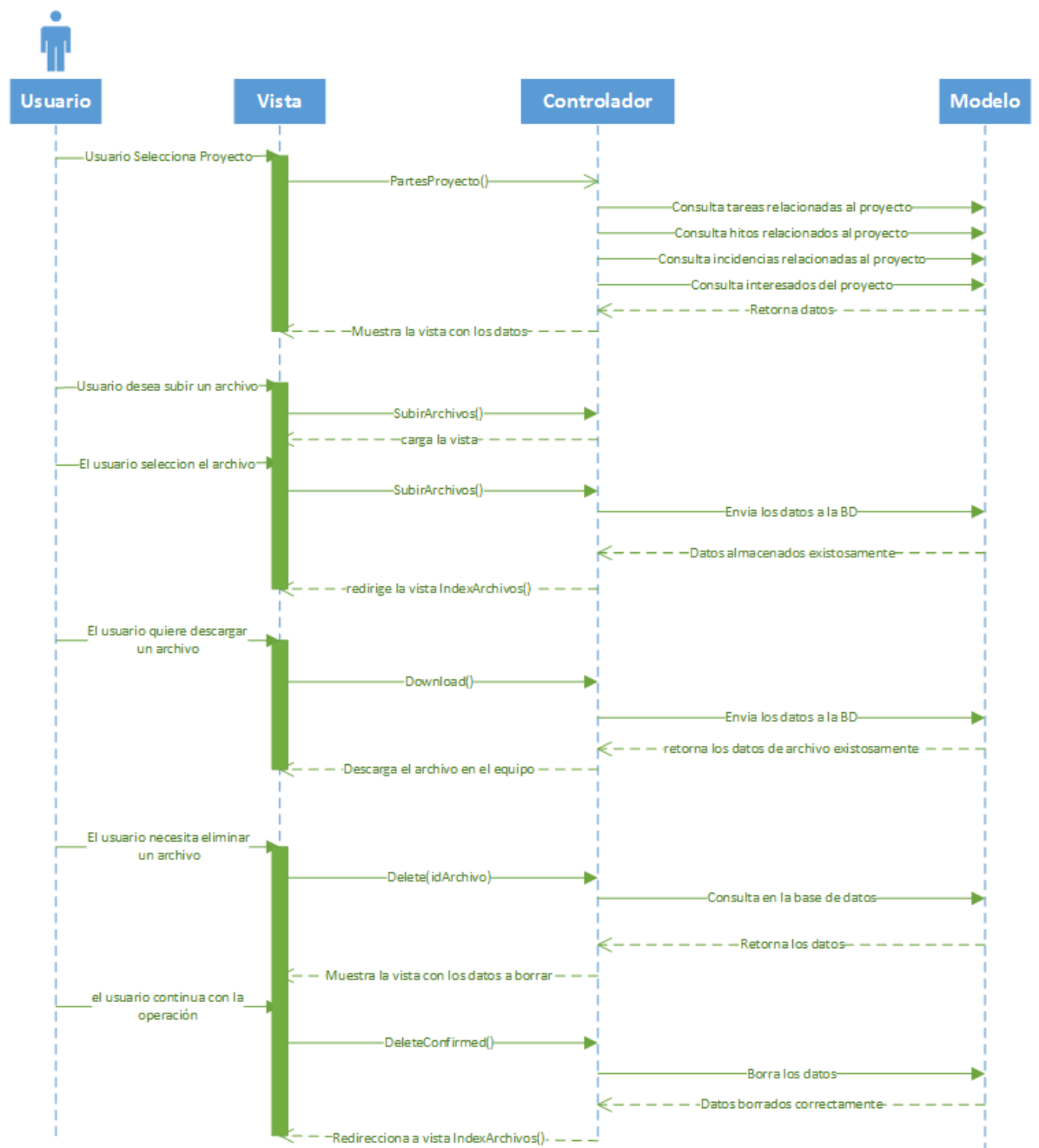


Ilustración 13 Diagrama de Secuencia para Archivos

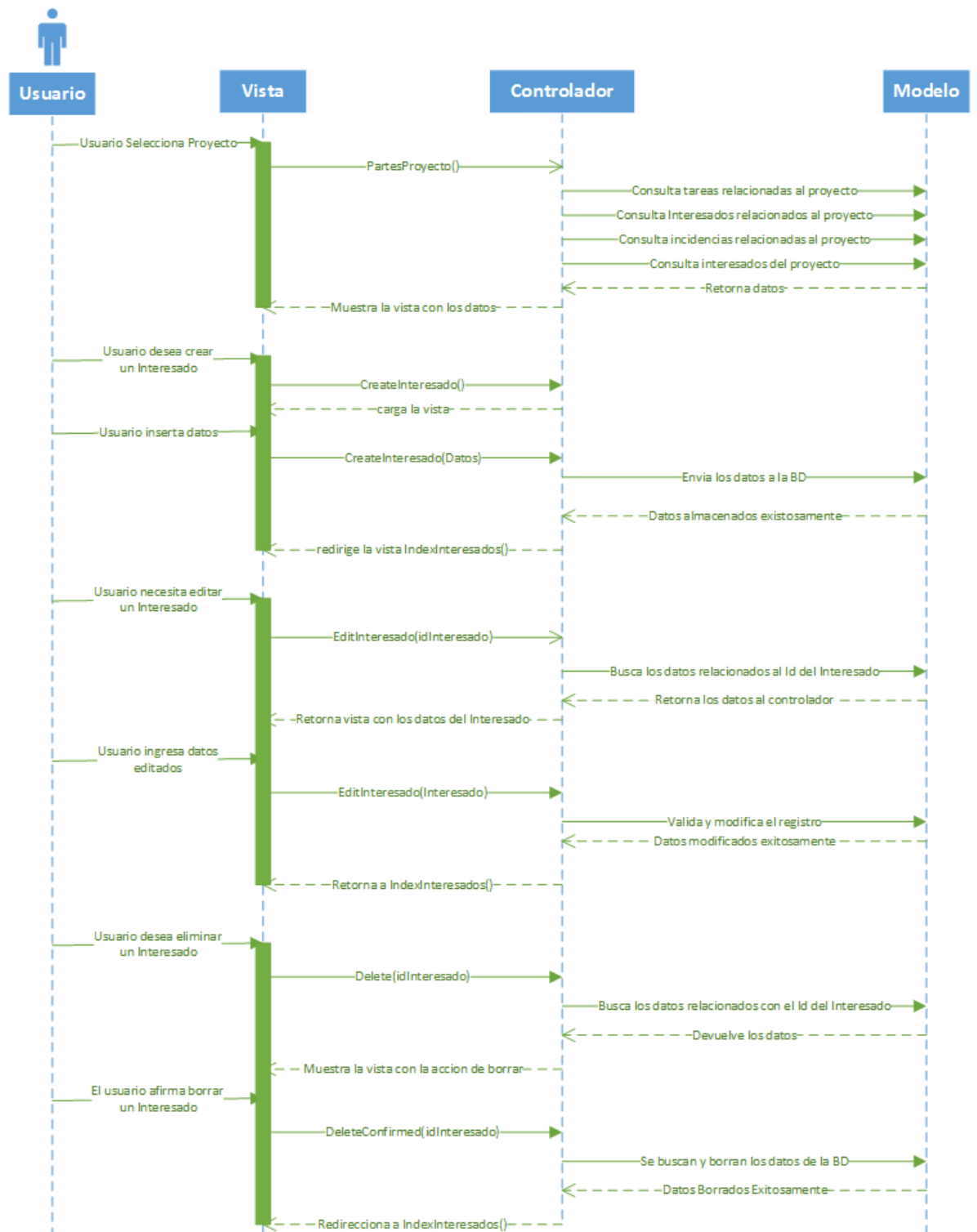


Ilustración 14 Diagrama de Secuencia para interesados



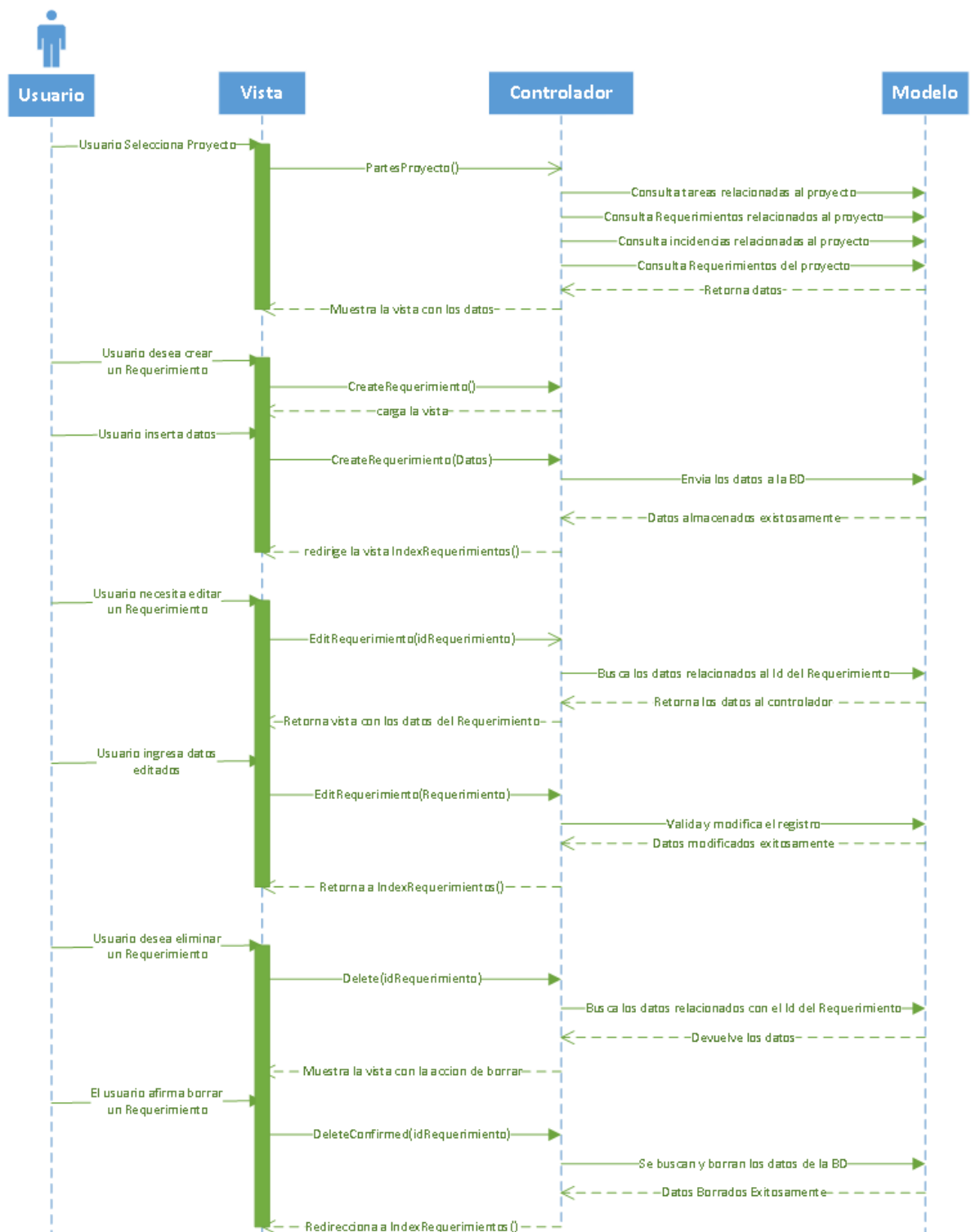


Ilustración 15 Diagrama de secuencia para Requerimientos

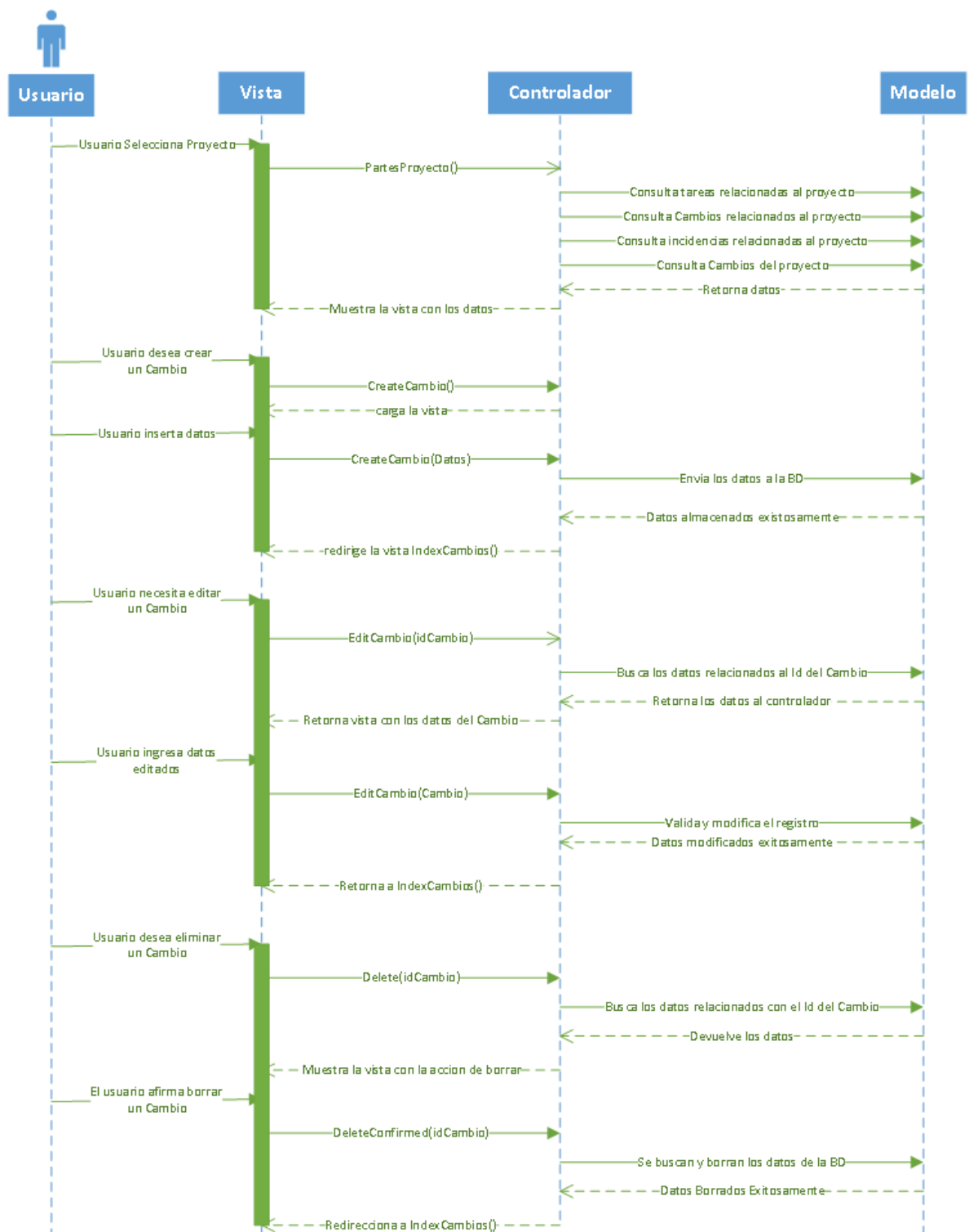


Ilustración 16 Diagrama de secuencia para cambios

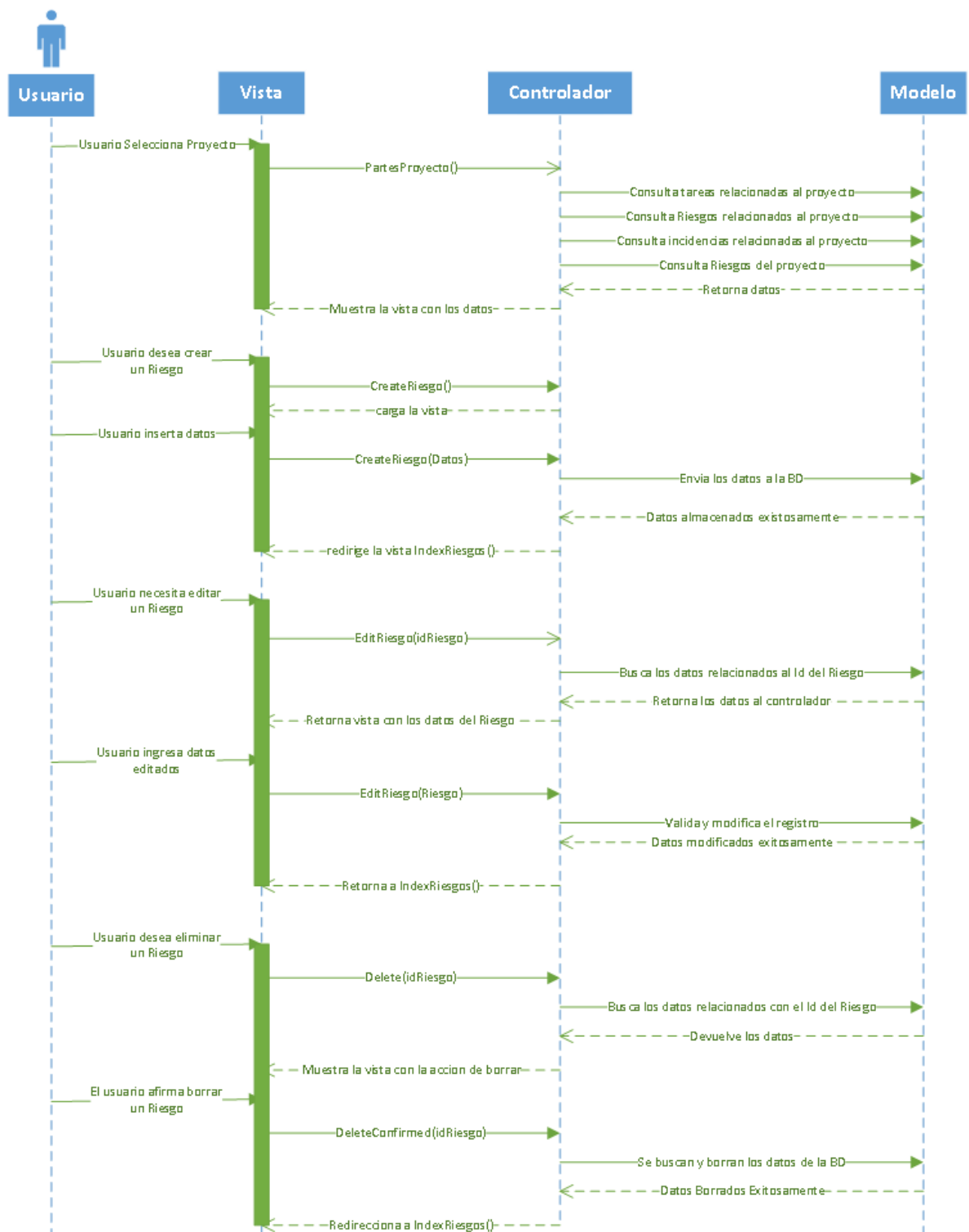


Ilustración 17 Diagrama de secuencia para riesgos

#### 4.7. DIAGRAMA DE ROBUSTEZ

Un diagrama de robustez permite cerrar la brecha entre lo que el sistema tiene que hacer y lo que se debe hacer para cumplir cada cosa que debe realizar la aplicación.

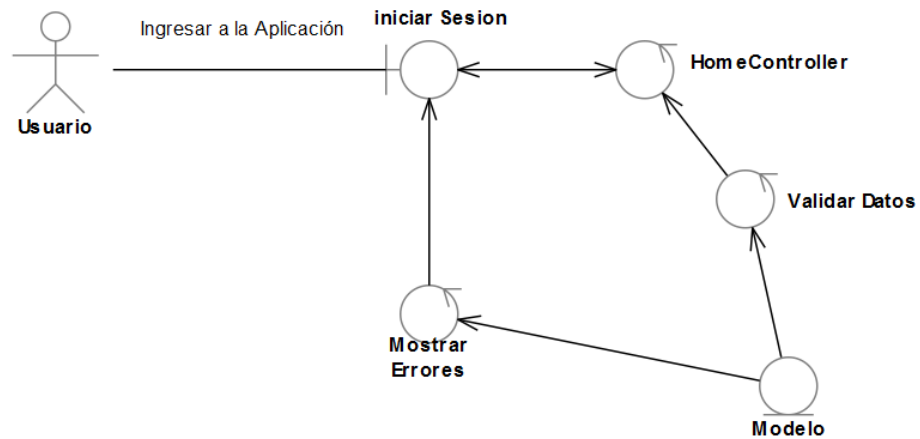


Ilustración 18 Diagrama de Robustez para Sesión

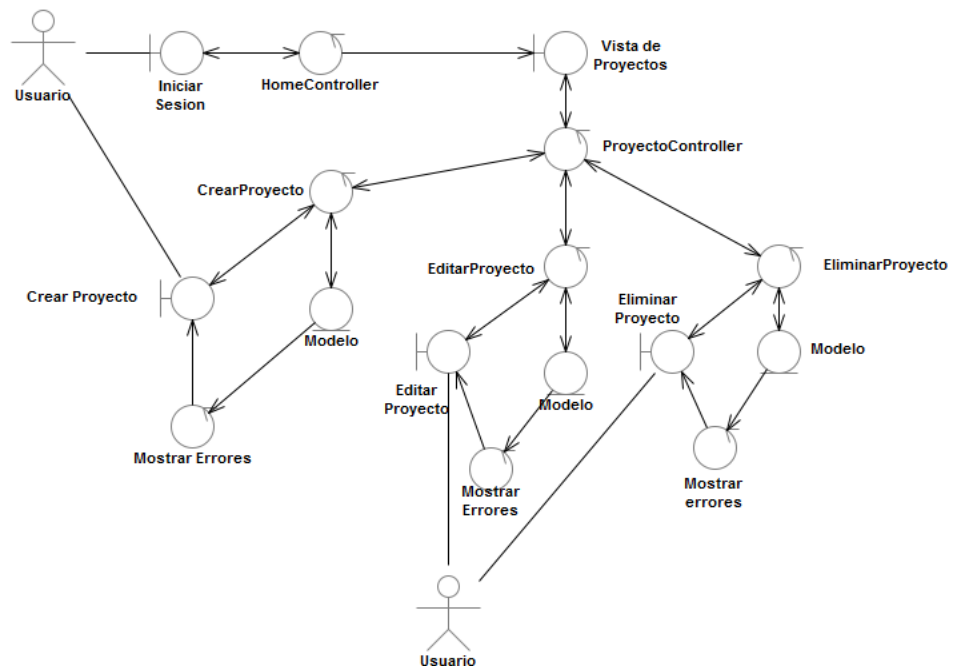


Ilustración 19 Diagrama de Robustez para Proyecto

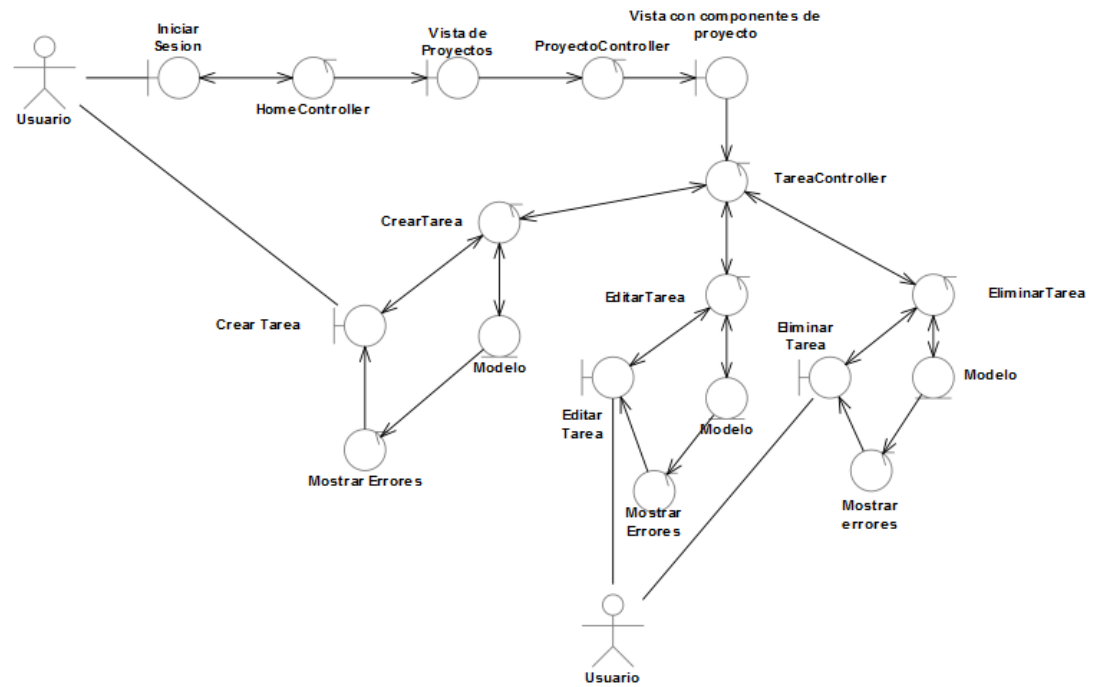


Ilustración 20 Diagrama de robustez para Tareas

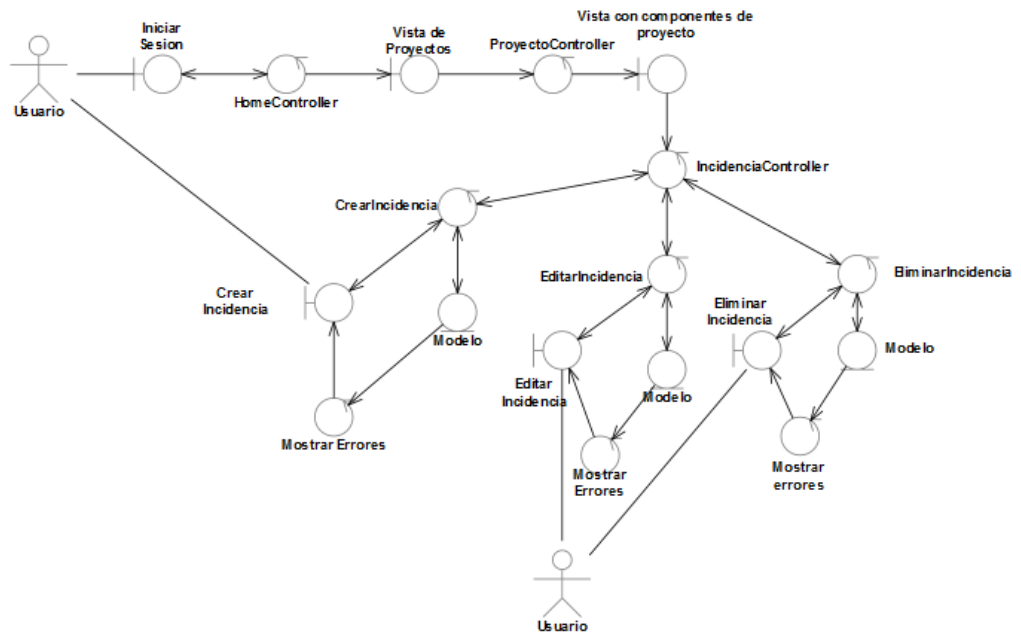


Ilustración 21 Diagrama de robustez para Incidencias

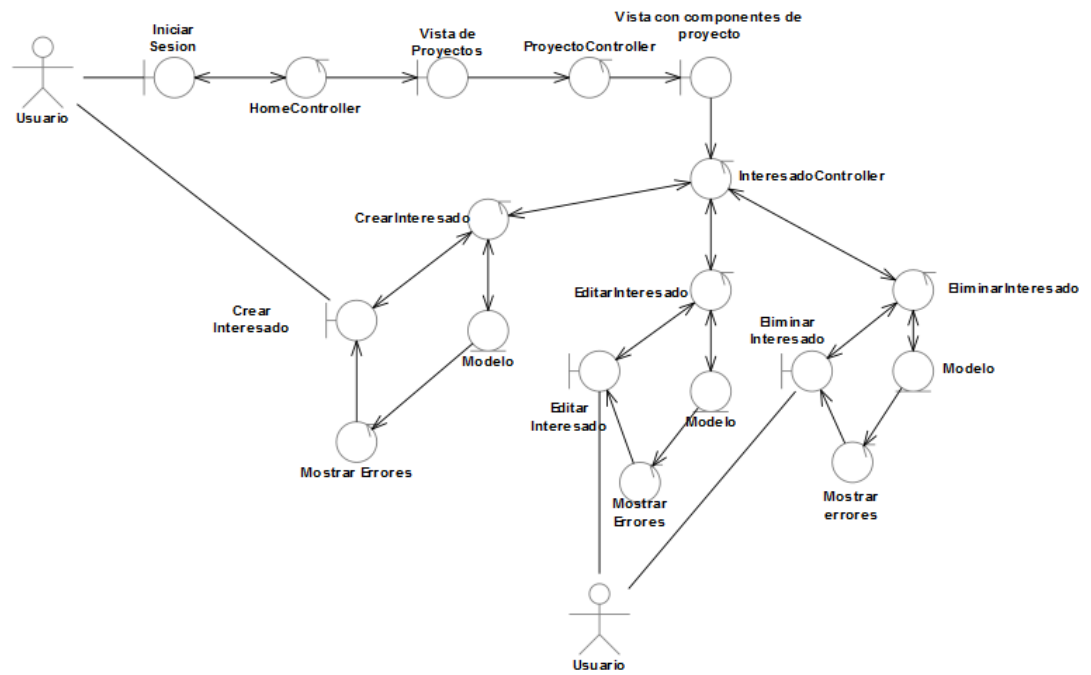


Ilustración 22 Diagrama de Robustez para Interesados

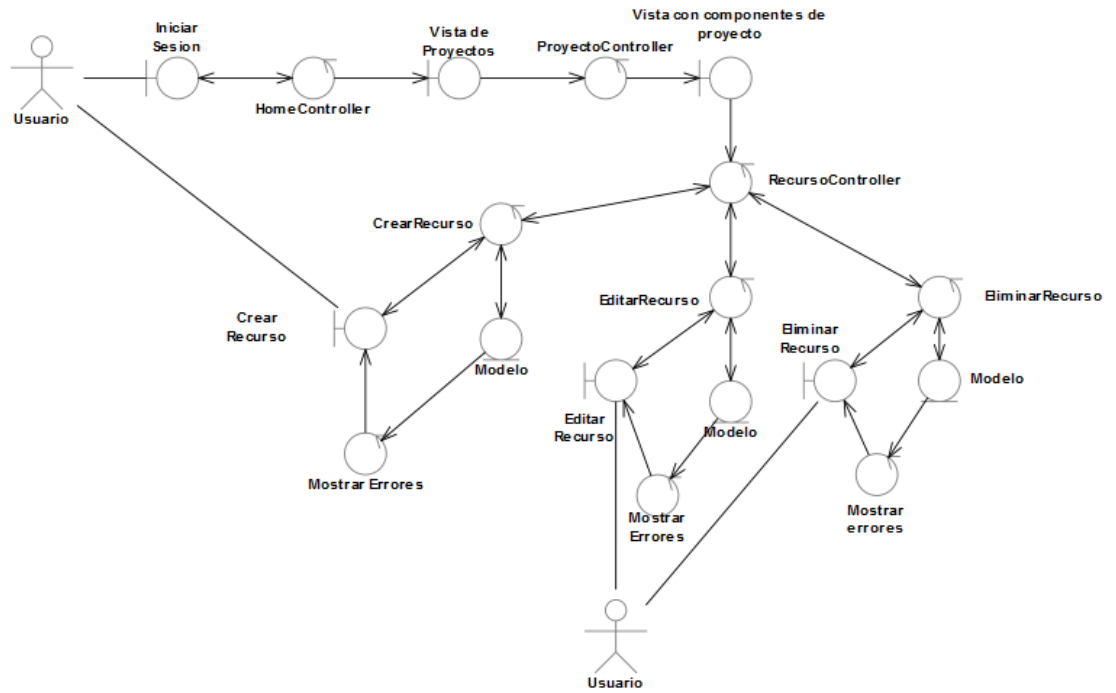


Ilustración 23 Diagrama de robustez para Recursos

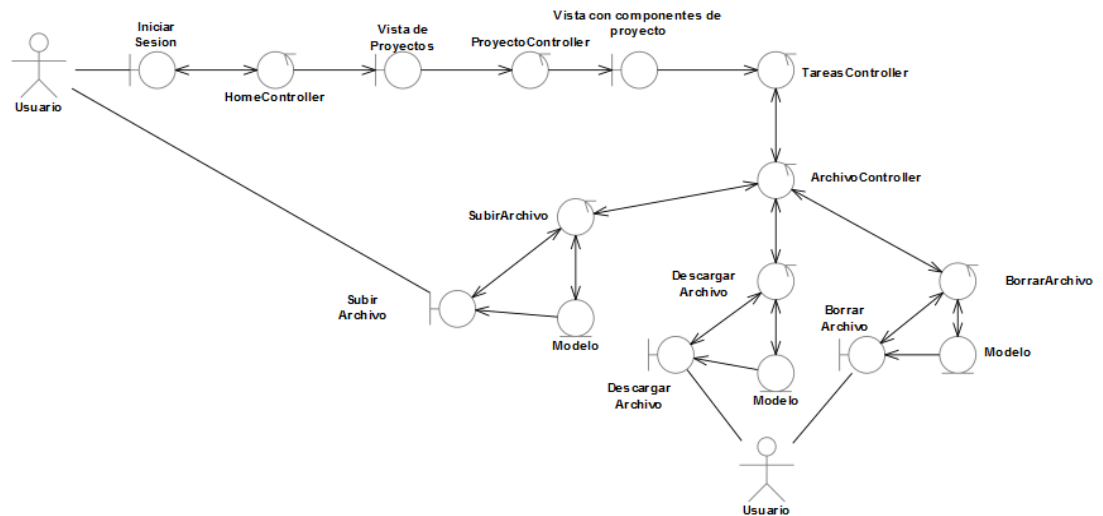


Ilustración 24 Diagrama de robustez para Archivos

#### 4.8. DIAGRAMA ENTIDAD RELACION Y BASE DE DATOS

A continuación se presentan los diagramas entidad relación propuestos al inicio del proyecto y aquel que se presenta como el modelo final del proyecto.

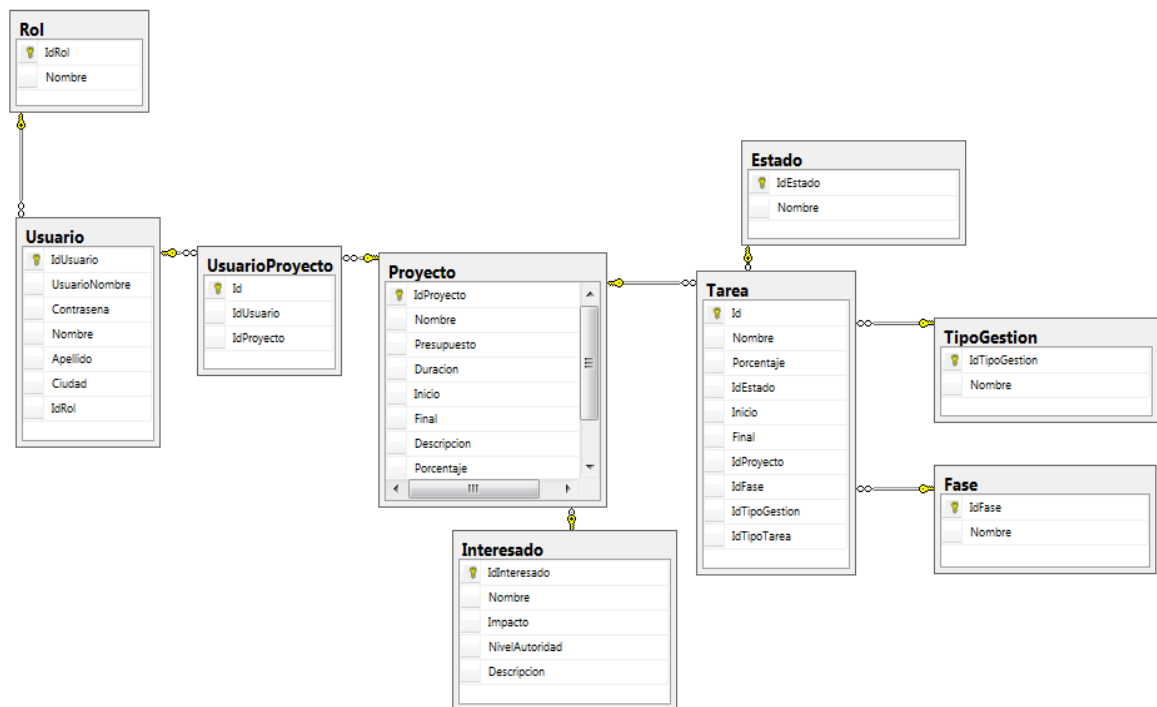


Ilustración 25 Diagrama Entidad-Relación inicial

Se presenta un listado con las tablas que se encuentran en el modelo final de la base de datos:

proyectogrado	dbo	ProyectoHeredado
proyectogrado	dbo	RelevanciaIncidencia
proyectogrado	dbo	RolUsuario
proyectogrado	dbo	TablaLogging
proyectogrado	dbo	Tarea
proyectogrado	dbo	TareasAntecesoras
proyectogrado	dbo	TipoGestion
proyectogrado	dbo	Usuario
proyectogrado	dbo	UsuarioProyecto
proyectogrado	dbo	sysdiagrams
proyectogrado	dbo	Recurso
proyectogrado	dbo	RecursoTarea
proyectogrado	dbo	TipoRecurso
proyectogrado	dbo	Archivo
proyectogrado	dbo	Cambios
proyectogrado	dbo	ImpactoRiesgo
proyectogrado	dbo	ProbabilidadCualitativaRiesgo
proyectogrado	dbo	Requerimientos
proyectogrado	dbo	CategorialInteresado
proyectogrado	dbo	CategoriaProyecto
proyectogrado	dbo	TipoRequerimiento
proyectogrado	dbo	Ciudad
proyectogrado	dbo	TipoRespuesta
proyectogrado	dbo	EstadoHito
proyectogrado	dbo	TipoRiesgo
proyectogrado	dbo	EstadoIncidencia
proyectogrado	dbo	EstadoProyecto
proyectogrado	dbo	Fases
proyectogrado	dbo	FasesProyecto
proyectogrado	dbo	Hitos
proyectogrado	dbo	ImpactoInteresado
proyectogrado	dbo	Riesgos
proyectogrado	dbo	Incidencia
proyectogrado	dbo	Interesado
proyectogrado	dbo	NivelAutoridad
proyectogrado	dbo	NivelInteres
proyectogrado	dbo	Proyecto

Table 6 Tablas que componen la base de datos



A continuación se muestra el diagrama entidad relación de la aplicación, se corta en varias partes para mostrar el diseño de una manera legible.

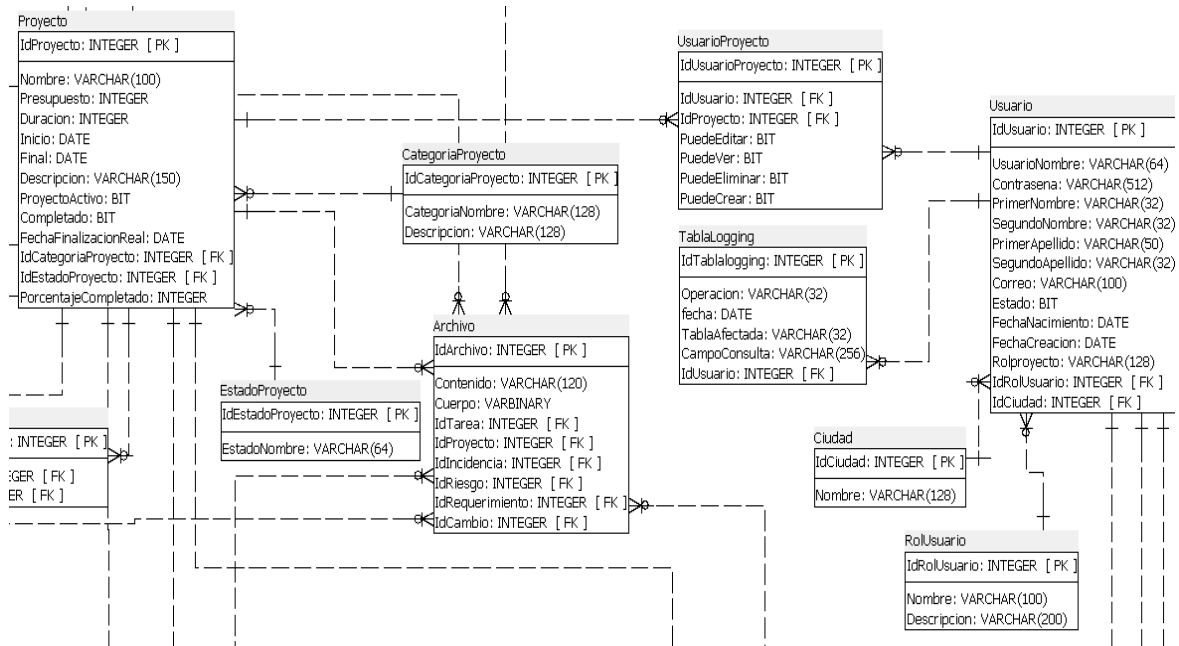


Ilustración 26 Relaciones de tabla usuarios

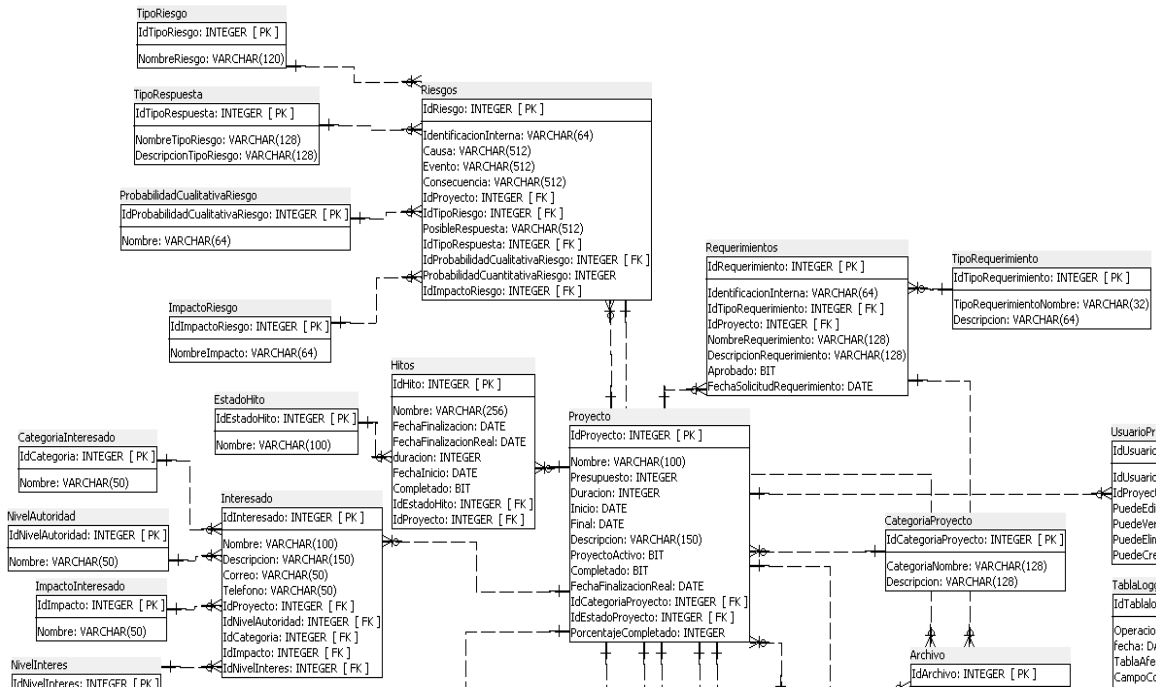


Ilustración 27 Relaciones Tabla Proyecto

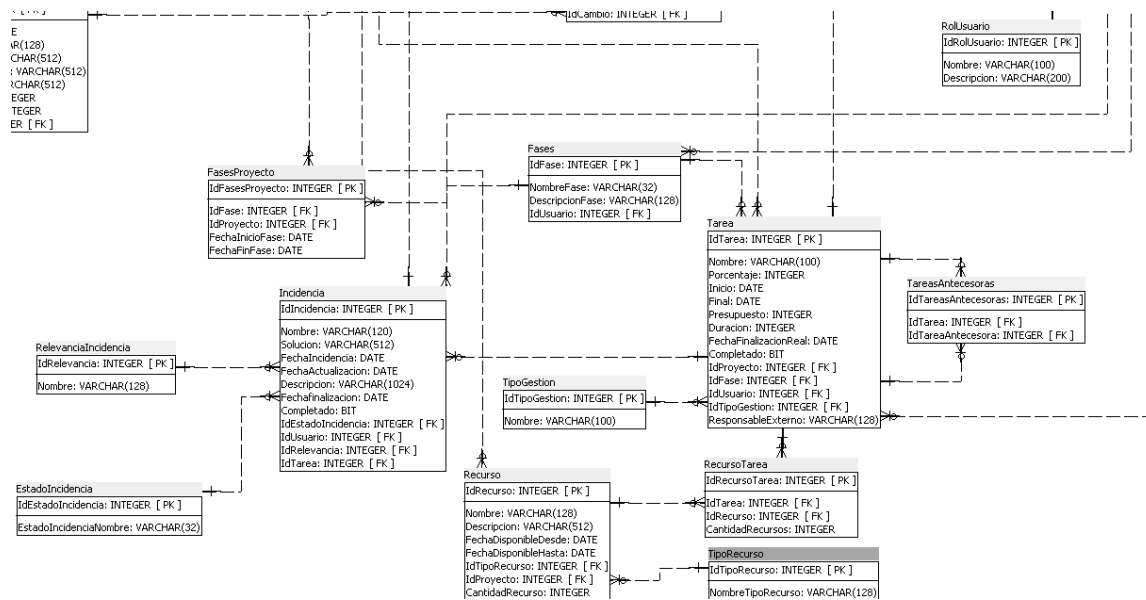


Ilustración 28 Relaciones tabla tareas

## 4.9. INTERFACES DE LA APLICACIÓN

### 3.11.1. Inicio de Sesión

En esta interfaz el usuario ingresa su nombre de usuario y su contraseña para ingresar a la aplicación.

Inicio de Sesión - Taskmaster

Iniciar Sesión

Registrar Usuario

Ilustración 29 Vista de Inicio de Sesión

### 3.11.2. Registrar un usuario

Esta es la interfaz que permitirá la creación de un usuario en caso de que este no exista dentro de la aplicación.

Registro de Usuario

Información Los campos con \* son obligatorios

\*Nombre de Usuario: Usuario

\*Primer Nombre: Primer Nombre

\*Primer Apellido: Primer Apellido

\*Correo: Correo

\*Fecha de Nacimiento: Fecha

\*Contraseña: Contraseña

Segundo Nombre: Segundo Nombre

Segundo Apellido: Segundo Apellido

\*Ciudad: Seleccione

Crear Usuario

Ilustración 30 Registrar Usuario

### 3.11.3. Interfaz de inicio

Esta es la pantalla que visualizará el usuario cuando ingrese al sistema, aquí podrá buscar entre todos los proyectos que el haya creado, ordenarlos por nombre y por fechas inicio o fin, en caso de que sea necesario editar sus datos, cerrar sesión, revisar si tiene proyectos inactivos y completos; para finalmente interactuar con las diversas opciones de la aplicación.

Taskmaster - Gestión de Proyectos | Bienvenido: Sergio Franco | Editar Datos | Cerrar Sesión

Mis Proyectos Activos e Incompletos

Crear Proyecto | Ver Proyectos Inactivos | Ver Proyectos Completados

Buscar

Nombre	Inicio	Fin								
Proyecto 12	10/07/2016 4:14:00 p. m.	6/11/2016 4:14:00 p. m.	Editar	Detalles	Borrar	Componentes	Compartir	Archivos	Desactivar	
proyecto1	21/06/2016 9:58:00 a. m.	31/12/2016 9:58:00 a. m.	Editar	Detalles	Borrar	Componentes	Compartir	Archivos	Desactivar	
proyecto2	21/06/2016 10:47:00 a. m.	16/10/2016 10:47:00 a. m.	Editar	Detalles	Borrar	Componentes	Compartir	Archivos	Desactivar	

© Sergio Enrique Franco Portilla - 2016

Ilustración 31 Interfaz de inicio para el usuario

### 3.11.4. Creación de un Proyecto

The screenshot shows a modal window titled "Crear Nuevo Proyecto" (Create New Project) overlaid on a sidebar menu. The sidebar menu has a "Mis Proyectos" (My Projects) section with a "Crear Proyecto" (Create Project) button. The modal form contains the following fields:

- \*Nombre:** Text input field with placeholder "Nombre del Proyecto".
- \*Presupuesto:** Text input field with placeholder "Presupuesto".
- Descripción:** Text area with placeholder "Descripción del Proyecto".
- \*Categoría:** Dropdown menu with "Seleccione" (Select) as the current value.
- \*Estado:** Dropdown menu with "Seleccione" (Select) as the current value.
- \*Fecha de Inicio:** Text input field with placeholder "Fecha de Inicio".

An orange banner at the top of the modal reads: "Información Los campos con \* son obligatorios" (Information The fields with \* are mandatory).

Ilustración 32 Crear Nuevo Proyecto

### 3.11.5. Editar Información del proyecto

Esta pantalla permite la edición de los datos de un proyecto, estos cambios son almacenados en la base de datos.

The screenshot shows a modal window titled "Editar Proyecto" (Edit Project) overlaid on the same sidebar menu as the previous image. The modal form contains the following fields:

- \*Nombre:** Text input field with the value "Proyecto 12".
- \*Presupuesto:** Text input field with the value "12000".
- Descripción:** Text area with the value "Proyecto 12".
- \*Categoría:** Dropdown menu with "Conocimiento de la Realidad" (Knowledge of Reality) as the current value.
- \*Estado:** Dropdown menu with "En Progreso" (In Progress) as the current value.
- \*Fecha de Inicio:** Text input field with the value "10/07/2016 16:14 PM".

An orange banner at the top of the modal reads: "Información Los campos con \* son obligatorios" (Information The fields with \* are mandatory).

Ilustración 33 Editar Proyecto

3.11.6. Información Detallada del proyecto

Esta vista permite que los usuarios vean los detalles completos del proyecto que han seleccionado

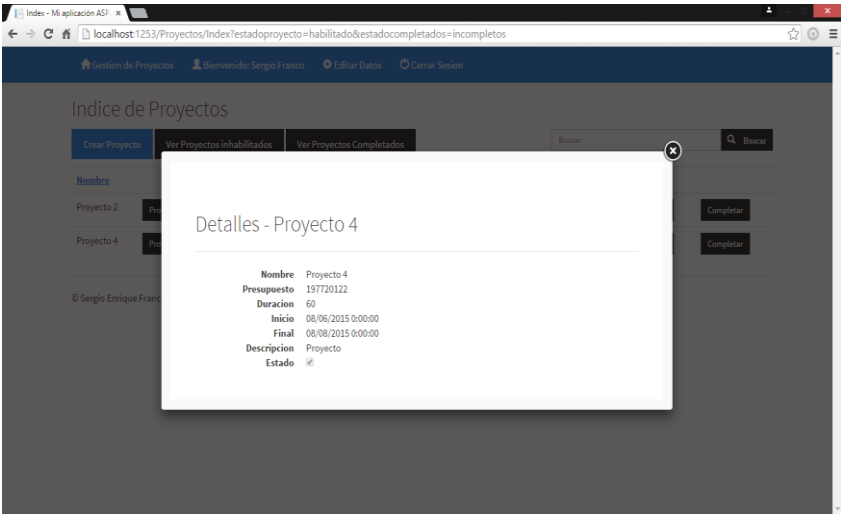


Ilustración 34 Ver Detalles de Proyecto

3.11.7. Eliminar Proyecto

Esta ventana permite eliminar un proyecto de la base de datos, siempre y cuando no tenga tareas u otros datos asociados a este.

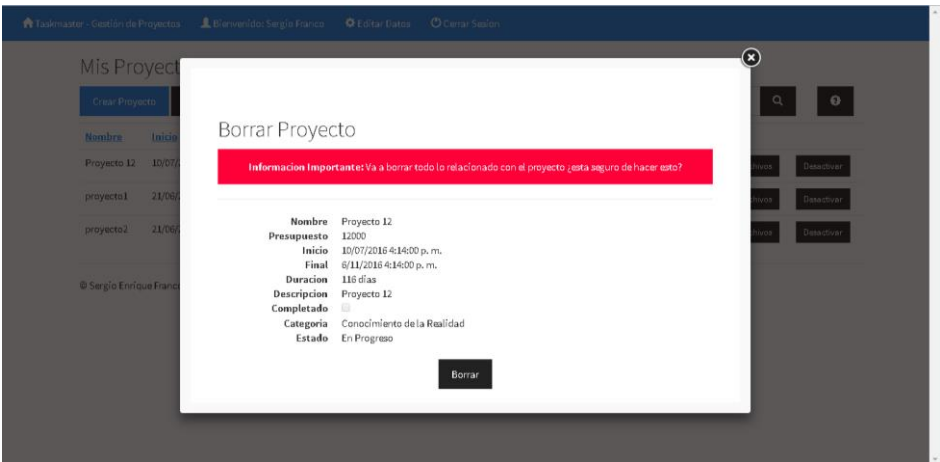


Ilustración 35 Eliminar Proyecto

### 3.11.8. Componentes del proyecto

Esta vista permite ver los objetos relacionados a un proyecto: Tareas, Incidencias, Recursos, interesados e Hitos

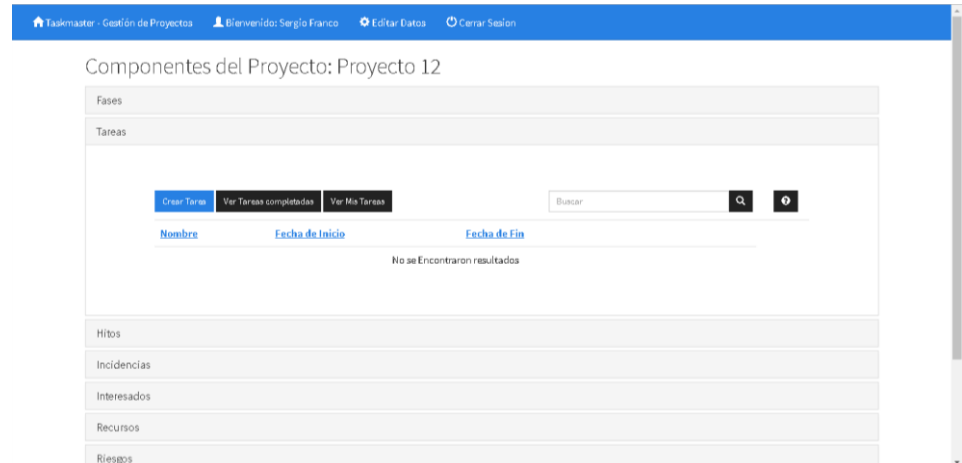


Ilustración 36 Vista con los componentes del proyecto

### 3.11.9. Tareas

En la vista de componentes se puede realizar la gestión de las tareas existentes en el proyecto



Ilustración 37 Ver Tareas

### 3.11.10. Crear Tarea

Esta ventana muestra la creación de una tarea para un proyecto previamente seleccionado.

The screenshot shows the 'Crear Tarea' dialog box. It has a title bar with a close button. Below the title is an orange information bar that reads 'Información: Los campos con \* son obligatorios'. The form contains the following fields:

- \*Nombre:** A text input field with the placeholder 'Nombre de la Tarea'.
- Descripción:** A larger text input field with the placeholder 'Descripción de la tarea'.
- \*Porcentaje:** A text input field with the placeholder 'Porcentaje'.
- \*Presupuesto:** A text input field with the placeholder 'Presupuesto'.
- \*Fase:** A dropdown menu with the placeholder 'Seleccione'.
- ¿Es un responsable externo?:** A checkbox.

Ilustración 38 Crear Tarea

### 3.11.11. Editar Tarea

Se cargará una ventana con los datos de la tarea seleccionada, para su edición.

The screenshot shows the 'Editar Información de Tarea 1' dialog box. It has a title bar with a close button. Below the title is an orange information bar that reads 'Información: Los campos con \* son obligatorios'. The form contains the following fields:

- \*Nombre:** A text input field with the value 'Tarea 1'.
- Descripción:** A larger text input field with the value 'Tarea 1'.
- \*Porcentaje:** A text input field with the value '10'.
- \*Presupuesto:** A text input field with the value '1200'.
- \*Fase:** A dropdown menu with the value 'Inicio'.
- Responsable:** A dropdown menu with the value 'sfranco'.
- \*Tipo de:** A dropdown menu with the value 'Gestión de Integración'.

Ilustración 39 Editar Tarea

3.11.12. Detalles de Tarea

Esta vista permite ver toda la información correspondiente a una tarea.

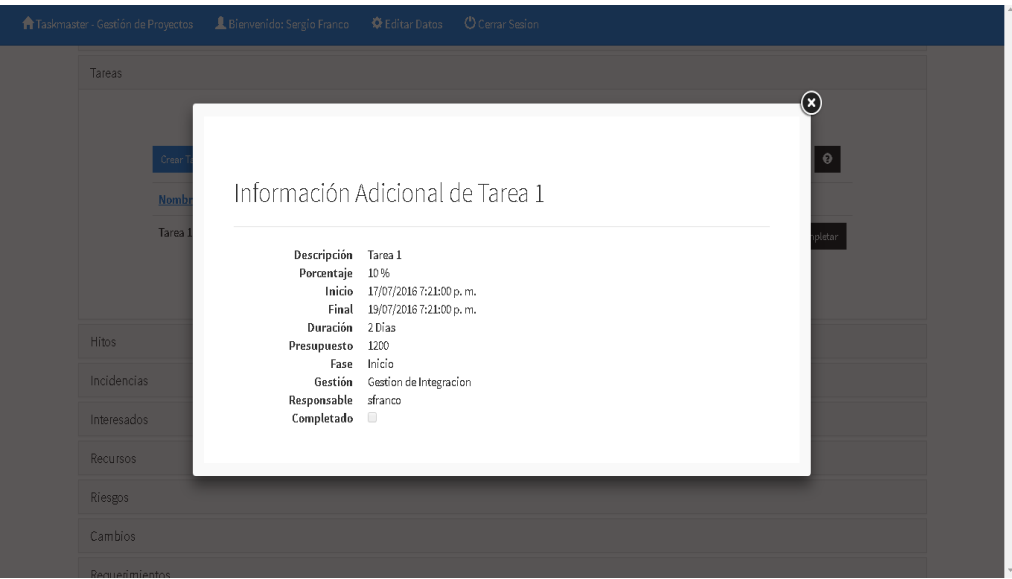


Ilustración 40 Detalles de Tarea

3.11.13. Borrar Tarea

Esta ventana muestra los datos de la tarea y la opción de borrarla de la base de datos.

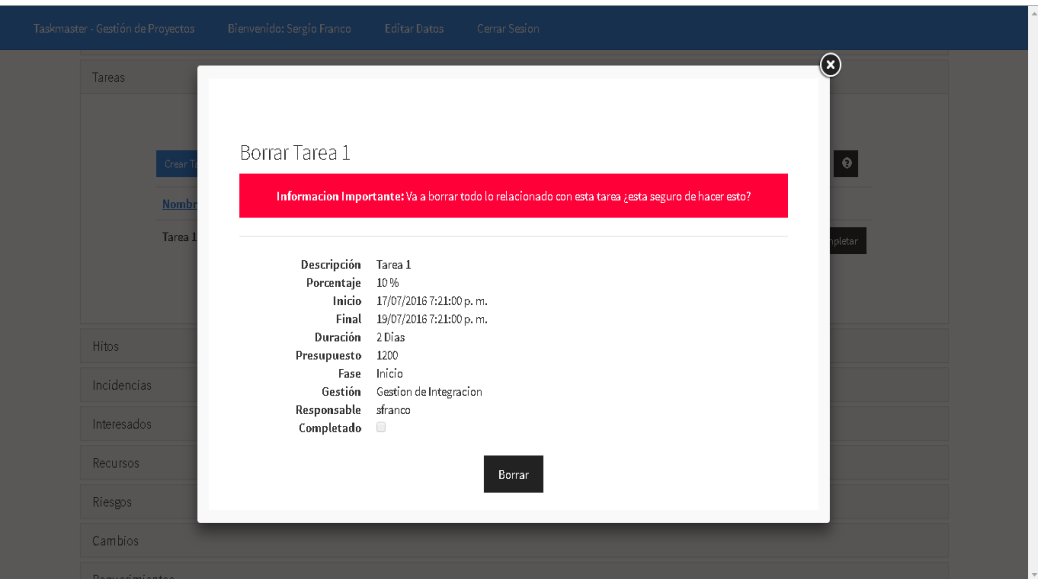


Ilustración 41 Borrar Tarea



### 3.11.14. Visualizar y Descargar Archivos Adjuntos

Esta ventana permite ver los archivos que se encuentren relacionados a una tarea para descargarlos, borrarlos o añadir uno nuevo, según la necesidad del usuario.

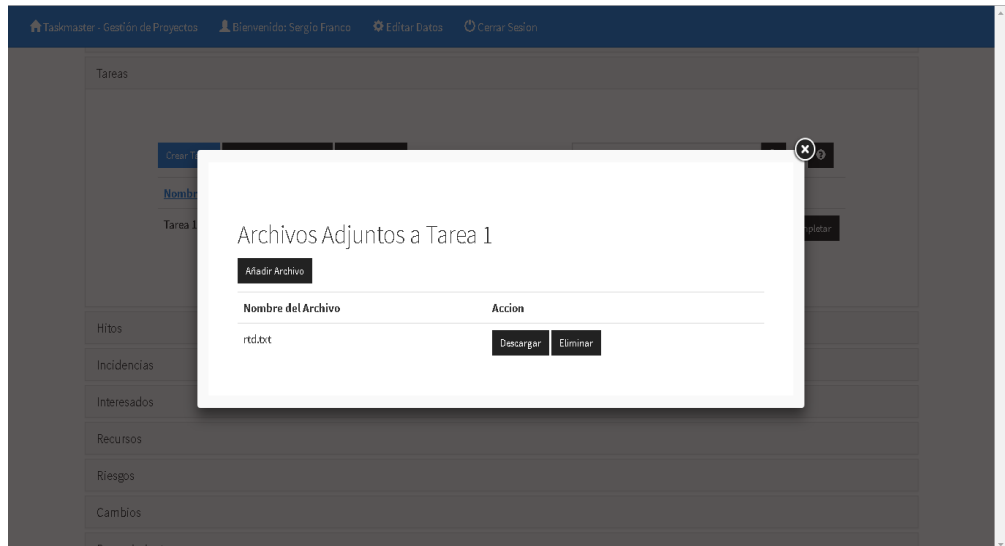


Ilustración 42 Visualizar Archivos

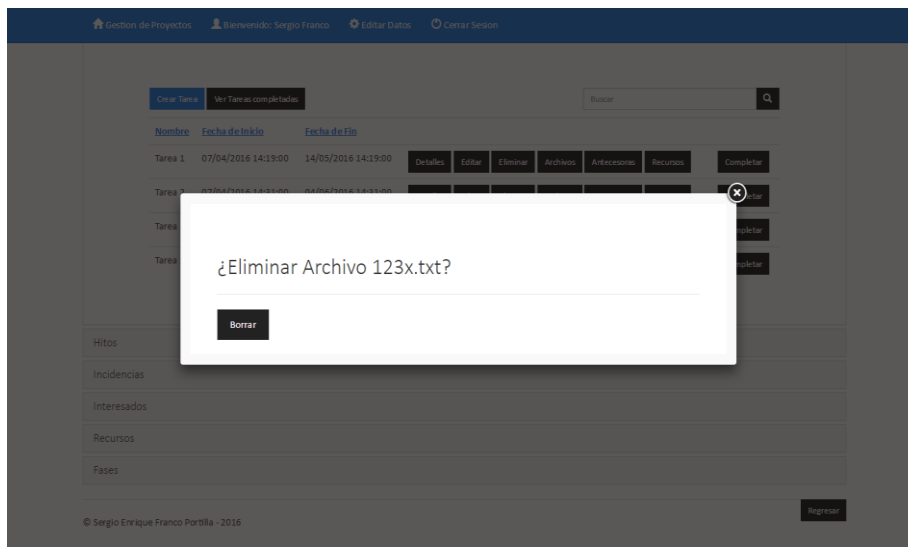
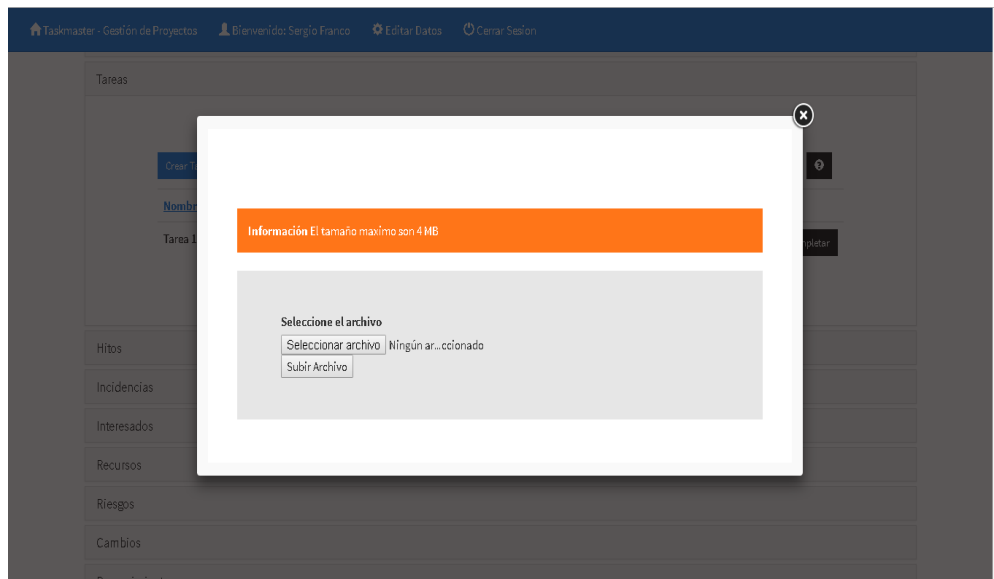


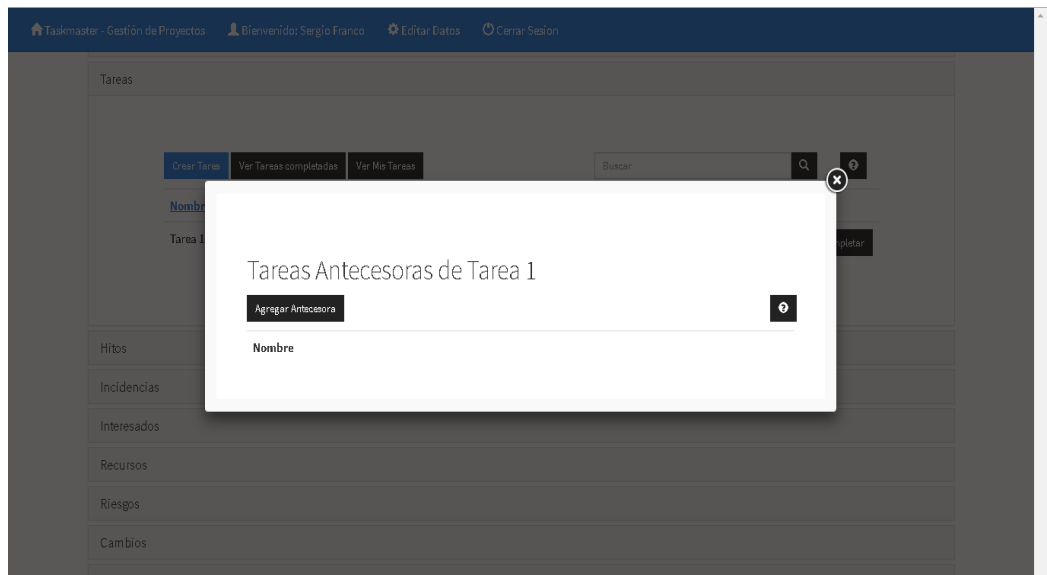
Ilustración 43 Eliminar Archivo



**Ilustración 44 Agregar Archivo**

### 3.11.15. Visualización de Tareas Antecesoras

Esta ventana permite ver un listado de las tareas antecesoras de otra tarea.



**Ilustración 45 Tareas Antecesoras**

### 3.11.16. Agregar Tarea Antecesora

The screenshot shows a web application interface with a dark blue header. The header contains the text 'Taskmaster - Gestión de Proyectos', a user profile 'Bienvenido: Sergio Franco', and links for 'Editar Datos' and 'Cerrar Sesión'. The main content area is titled 'Tareas'. A modal window titled 'Asignar Tarea Antecesora' is open in the center. Inside the modal, there is an orange banner with the text 'Información Los campos con \* son obligatorios'. Below this, there is a label '\*Tarea Antecesora' next to a dropdown menu with the text 'Seleccione'. At the bottom right of the modal is a 'Guardar' button. The background of the application shows a sidebar with various menu items like 'Hitos', 'Incidencias', 'Interesados', 'Recursos', 'Riesgos', 'Cambios', and 'Requerimientos'.

Ilustración 46 Crear Tarea Antecesora

### 3.11.17. Visualizar interesados del proyecto

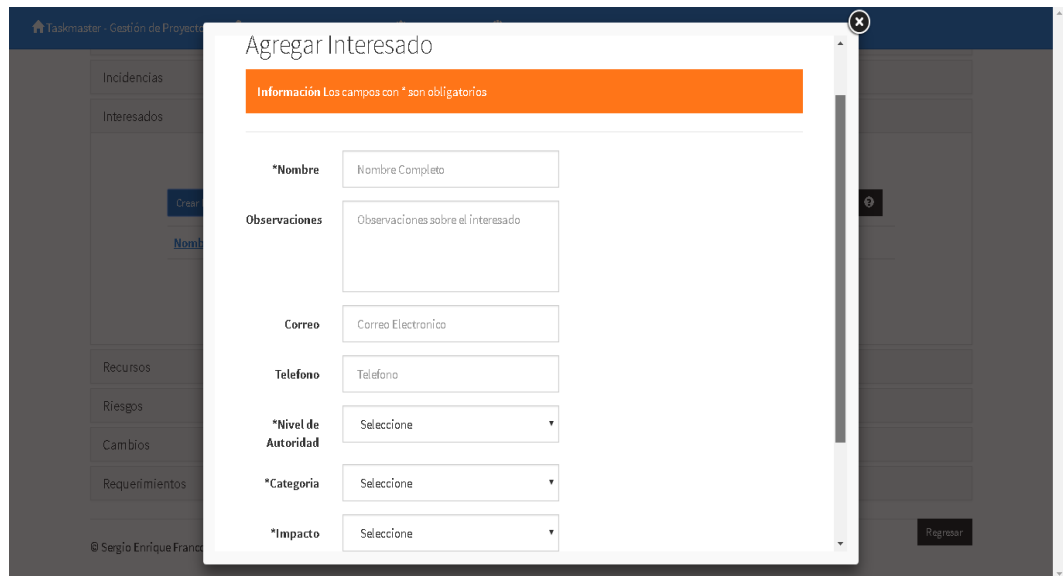
En esta vista se tiene un listado de los interesados relacionados con el proyecto.

The screenshot shows the 'Interesados' (Interested Parties) view in the Taskmaster application. The header is blue and contains the same navigation elements as the previous screenshot. The main content area has a light gray background. At the top, there is a section titled 'Interesados'. Below this, there is a 'Crear Interesado' button and a search bar with the text 'Buscar'. Below the search bar, there are three columns labeled 'Nombre', 'Correo', and 'Telefono'. Below these columns, there is a message 'No se Encontraron resultados'. At the bottom of the page, there is a 'Regresar' button and a copyright notice '© Sergio Enrique Franco Portilla - 2016'.

Ilustración 47 Ver Interesados de Proyecto

### 3.11.18. Crear Interesado para un Proyecto

Esta vista permite crear a un interesado relacionado con un proyecto.

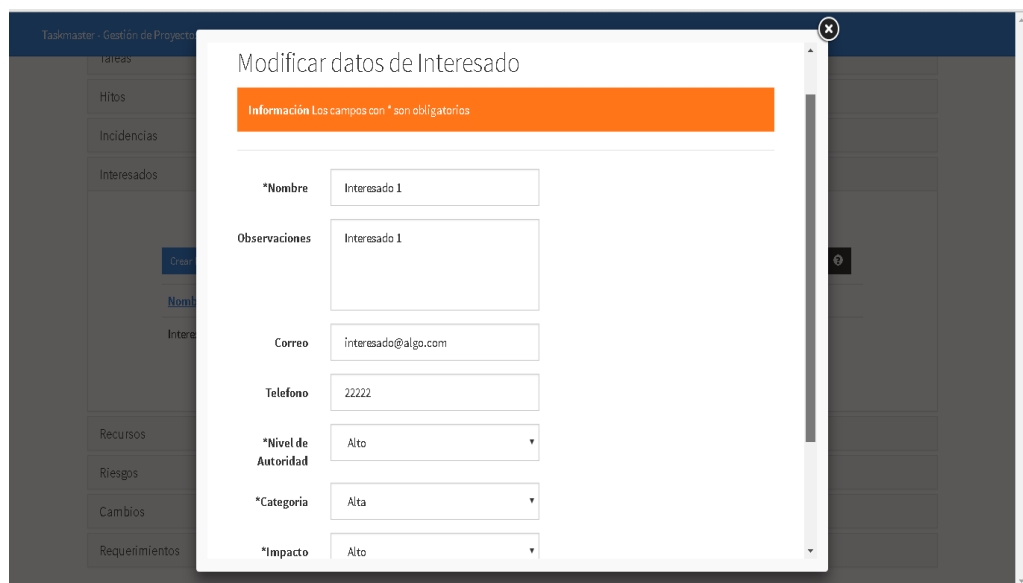


The screenshot shows a web application interface with a sidebar on the left containing menu items: Incidencias, Interesados, Recursos, Riesgos, Cambios, and Requerimientos. The main content area displays a modal window titled "Agregar Interesado". At the top of the modal is an orange banner with the text "Información Los campos con \* son obligatorios". Below this, the form includes several fields: a text input for "\*Nombre" (containing "Nombre Completo"), a text area for "Observaciones" (containing "Observaciones sobre el interesado"), a text input for "Correo" (containing "Correo Electrónico"), a text input for "Telefono" (containing "Telefono"), and three dropdown menus for "\*Nivel de Autoridad", "\*Categoria", and "\*Impacto", all showing "Seleccione". A "Regresar" button is located at the bottom right of the modal.

Ilustración 48 Crear Interesado

### 3.11.19. Editar Datos de un interesado

Esta ventana permite editar los datos relacionados al interesado de un proyecto.



The screenshot shows the same web application interface as the previous one, but with a modal window titled "Modificar datos de Interesado". It also features an orange banner with the text "Información Los campos con \* son obligatorios". The form fields are pre-filled with data: "\*Nombre" is "Interesado 1", "Observaciones" is "Interesado 1", "Correo" is "interesado@algo.com", "Telefono" is "22222", and the dropdowns for "\*Nivel de Autoridad", "\*Categoria", and "\*Impacto" are all set to "Alto". A "Regresar" button is visible at the bottom right of the modal.

Ilustración 49 Editar Interesado

### 3.11.20. Detalles de Interesado

Muestra la información detallada de un interesado.

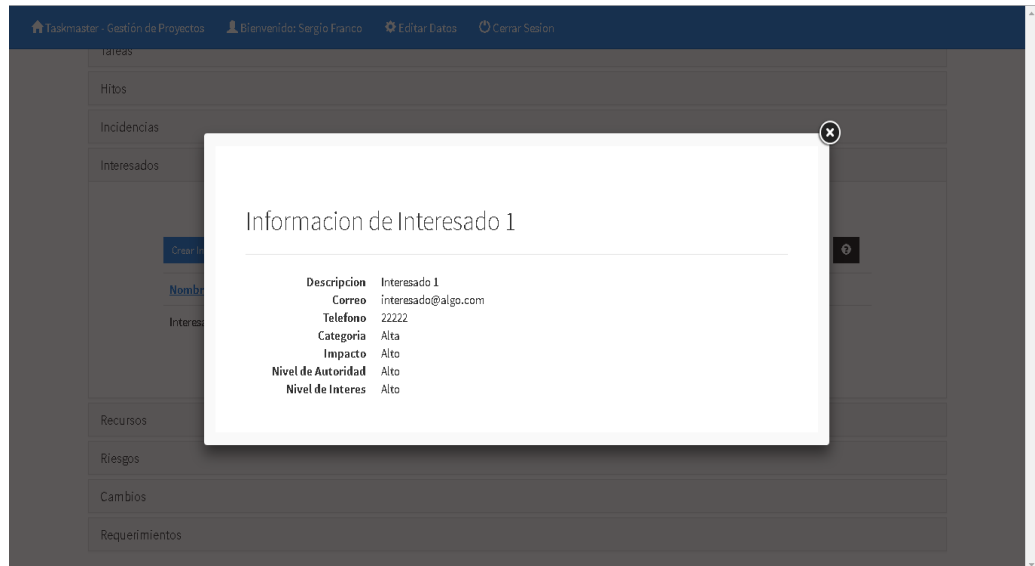


Ilustración 50 Ver Detalles de Interesado

### 3.11.21. Eliminar Interesado

Elimina la información relacionada a un interesado del proyecto.

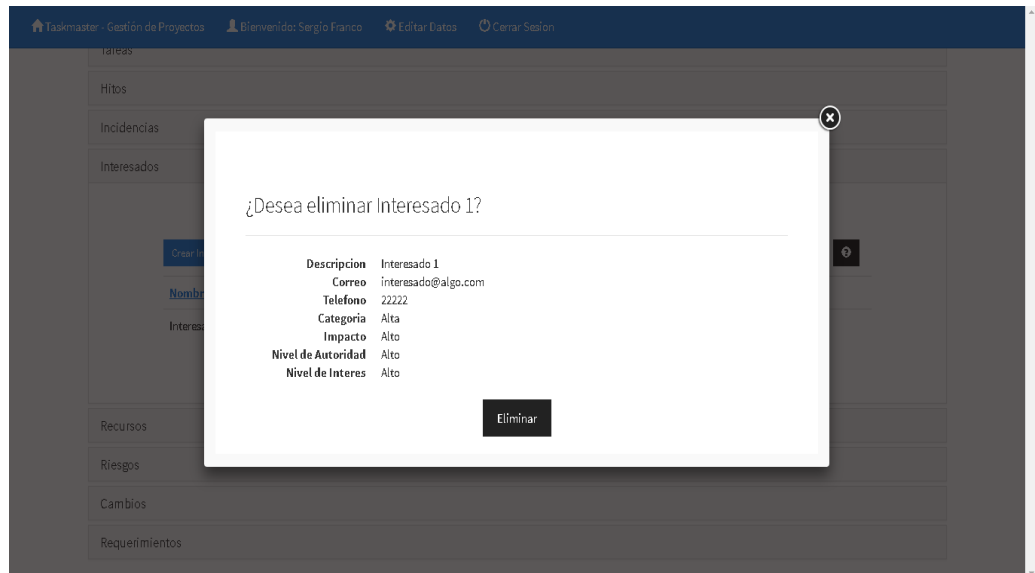


Ilustración 51 Eliminar Interesado

### 3.11.22. Visualización de Hitos

Permite visualizar los hitos relacionados con el proyecto.

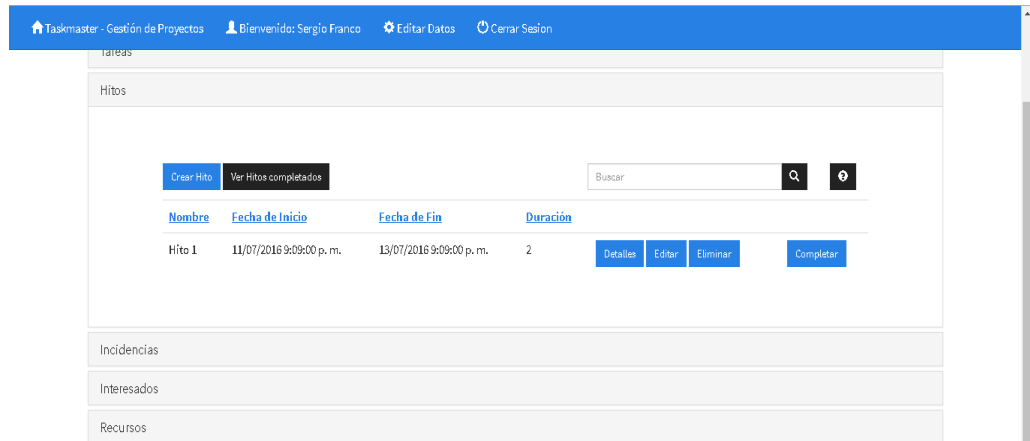


Ilustración 52 ver Hitos

### 3.11.23. Creación de Hitos

Esta vista permite la creación de un hito relacionado con un proyecto

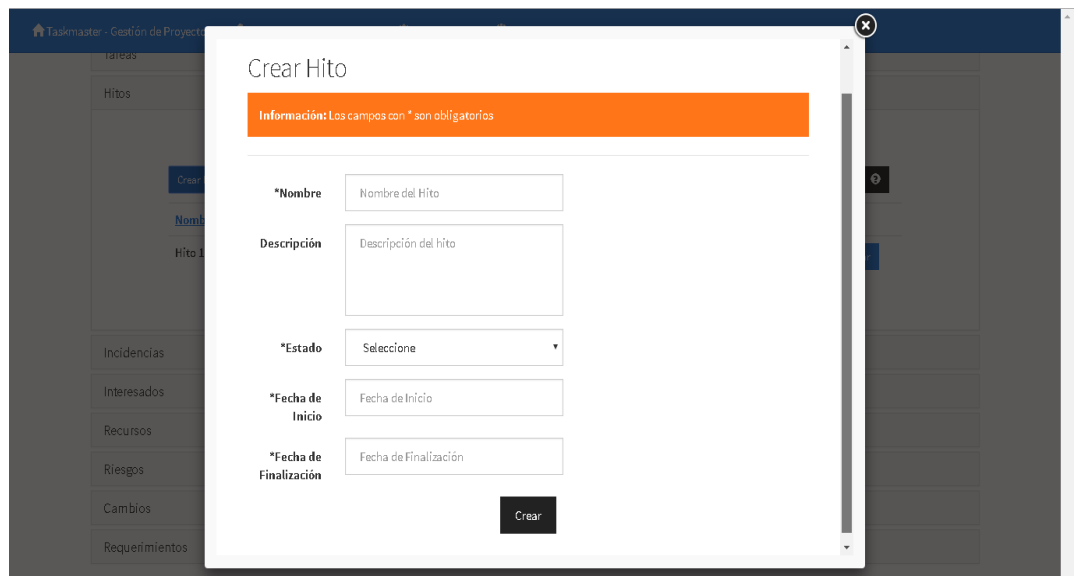


Ilustración 53 Crear Hito

### 3.11.24. Editar Hito

Permite editar los datos de un hito y guardarlos en base de datos.

The screenshot shows a web application interface with a sidebar menu on the left containing items like 'Tareas', 'Hitos', 'Incidentes', 'Interesados', 'Recursos', 'Riesgos', 'Cambios', and 'Requerimientos'. The main content area displays a modal window titled 'Editar Hito'. Inside the modal, there is an orange banner with the text 'Información: Los campos con \* son obligatorios'. Below this, the form fields are as follows:

Hitos	
*Nombre	Hito 1
Descripción	Descripción del hito
*Estado	Sin Completar
*Fecha de Inicio	11/07/2016 21:09 PM
*Fecha de Finalización	13/07/2016 21:09 PM

At the bottom right of the modal is a button labeled 'Guardar Cambios'.

Ilustración 54 Editar Hito

### 3.11.25. Detalles de hito

The screenshot shows the same web application interface. A modal window titled 'Información Adicional del Hito' is displayed over the 'Hitos' section. The modal contains the following information:

Nombre	Hito 1
FechaInicio	11/07/2016 9:09:00 p. m.
FechaFinalizacion	13/07/2016 9:09:00 p. m.
duracion	2 + Días
Completado	<input type="checkbox"/>
Nombre	Sin Completar

Ilustración 55 Ver Detalles de Hito

### 3.11.26. Eliminar Hito

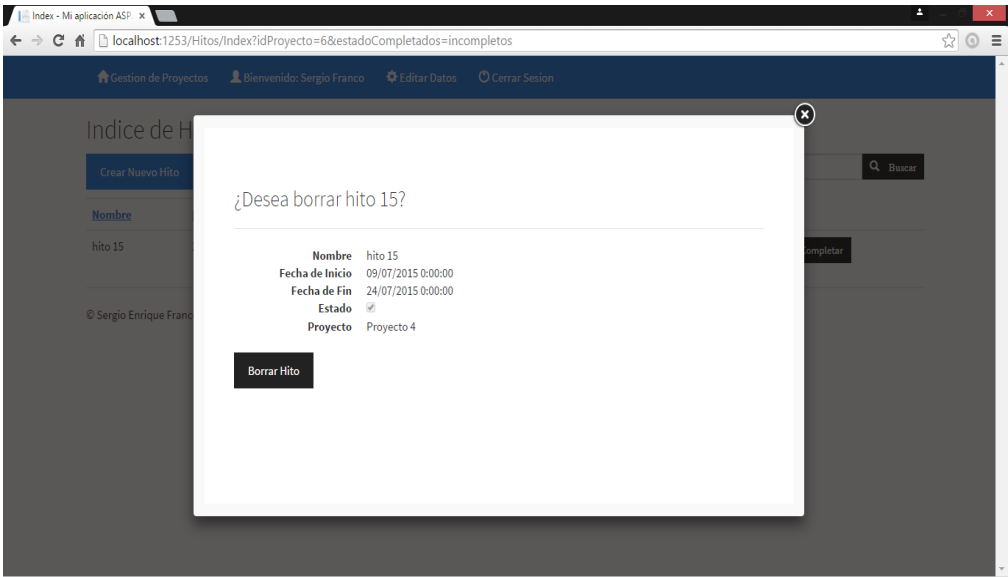


Ilustración 56 Eliminar Hito

### 3.11.27. Visualizar Incidencias

Esta ventana muestra un listado con las incidencias ocurridas en el proyecto.

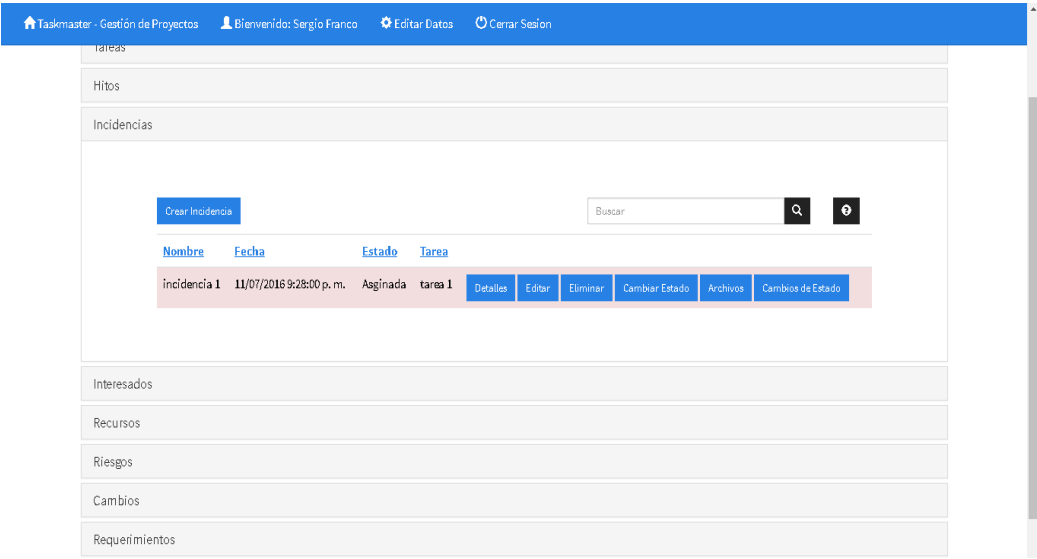
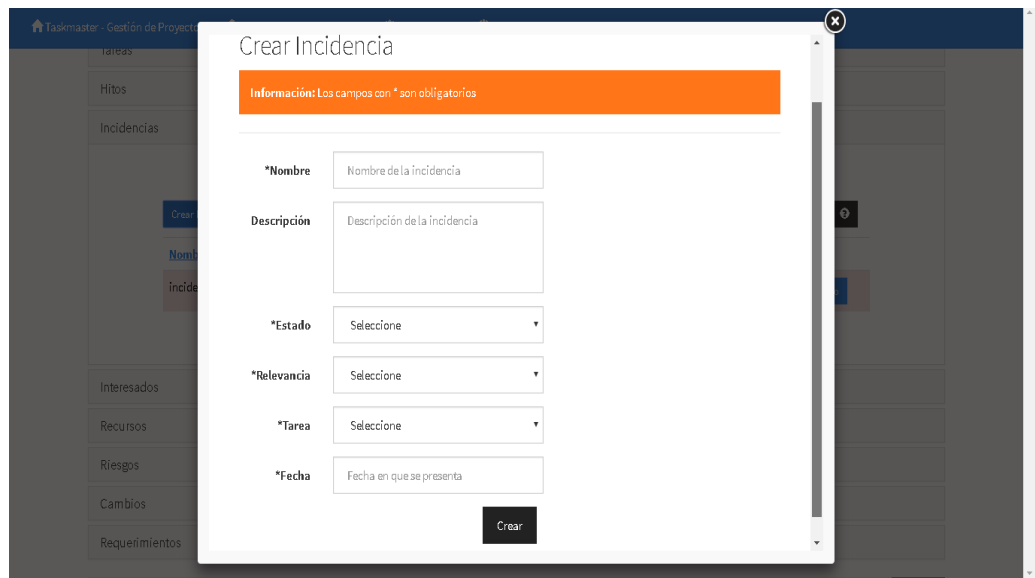


Ilustración 57 Ver Incidencias



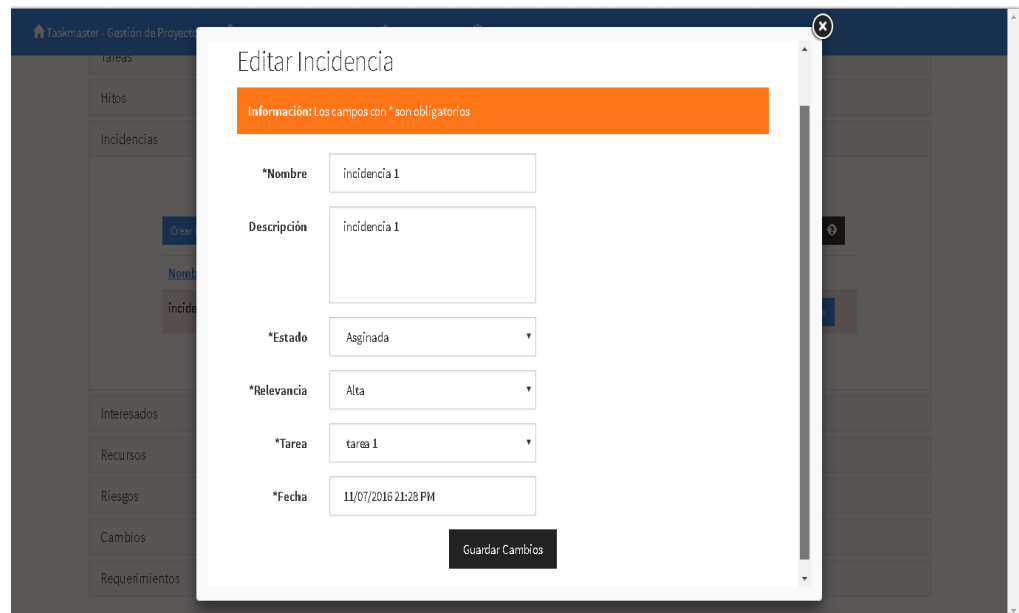
### 3.11.28. Crear Incidencia



The screenshot shows a web application interface with a sidebar on the left containing menu items: Tareas, Hitos, Incidencias, Interesados, Recursos, Riesgos, Cambios, and Requerimientos. The 'Incidencias' menu item is highlighted. A modal window titled 'Crear Incidencia' is open in the center. It features an orange header bar with the text 'Información: Los campos con \* son obligatorios'. Below this, there are six form fields: '\*Nombre' (text input), 'Descripción' (text area), '\*Estado' (dropdown menu), '\*Relevancia' (dropdown menu), '\*Tarea' (dropdown menu), and '\*Fecha' (text input). A 'Crear' button is located at the bottom right of the modal.

Ilustración 58 Crear Incidencia

### 3.11.29. Editar Incidencia



The screenshot shows the same web application interface as the previous one. The modal window is now titled 'Editar Incidencia'. It has the same orange header bar with the text 'Información: Los campos con \* son obligatorios'. The form fields are: '\*Nombre' (text input with value 'incidencia 1'), 'Descripción' (text area with value 'incidencia 1'), '\*Estado' (dropdown menu with value 'Asignada'), '\*Relevancia' (dropdown menu with value 'Alta'), '\*Tarea' (dropdown menu with value 'tarea 1'), and '\*Fecha' (text input with value '11/07/2016 21:28 PM'). A 'Guardar Cambios' button is located at the bottom right of the modal.

Ilustración 59 Editar Incidencia

3.11.30. Borrar Incidencia

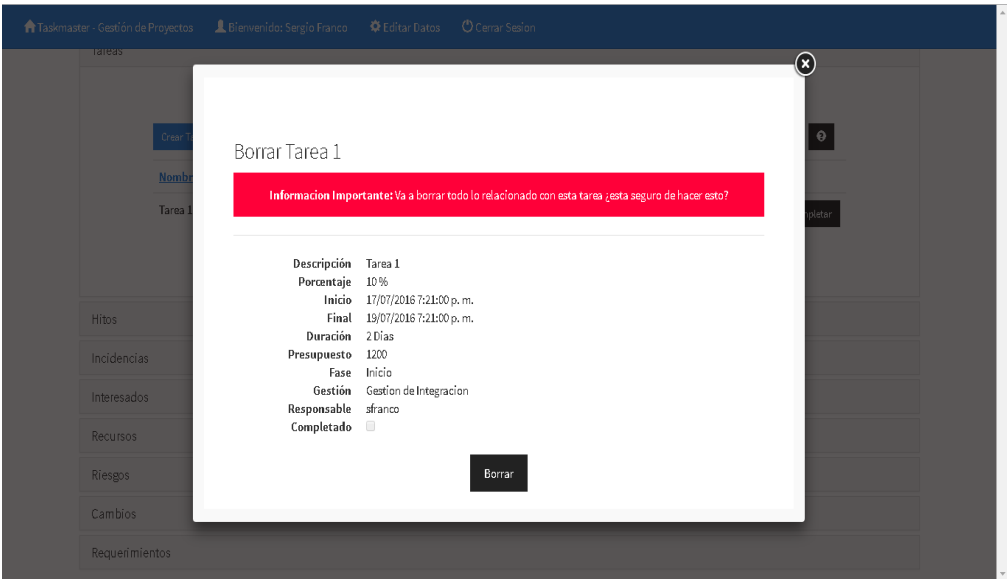


Ilustración 60 Borrar Incidencia

3.11.31. Compartir Proyecto

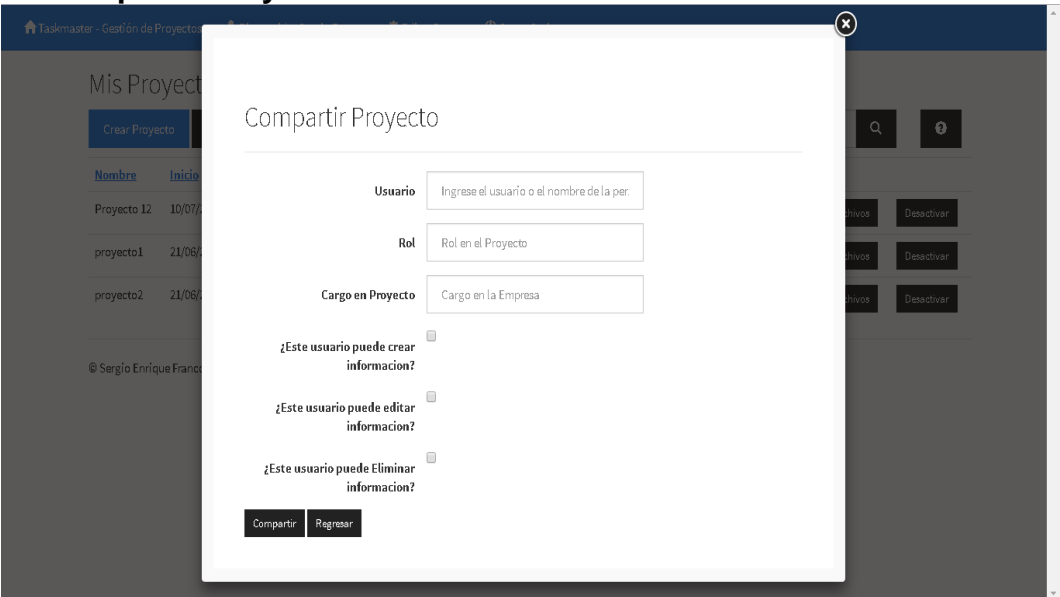


Ilustración 61 Compartir Proyecto

### 3.11.32. Crear un Recurso

The screenshot shows a web application interface with a sidebar on the left containing menu items: Incidencias, Interesados, Recursos, Riesgos, Cambios, and Requerimientos. The main content area displays a modal window titled 'Crear Recurso'. Inside the modal, there is an orange header bar with the text 'Información Los campos con \* son obligatorios'. Below this, the form contains the following fields: 'Nombre' (text input with 'Recurso'), 'Descripción' (text area with 'Observaciones sobre el interesado'), 'Tipo de Recurso' (dropdown menu with 'Seleccione'), 'Cantidad' (text input with 'Cantidad'), '\*Desde' (text input with 'Disponible Desde'), and '\*Hasta' (text input with 'Disponible Hasta'). At the bottom right of the modal is a 'Crear Recurso' button. The background shows a partial view of the sidebar and a 'Regresar' button.

Ilustración 62 Crear Recurso

### 3.11.33. Editar un Recurso

The screenshot shows the same web application interface as the previous one, but with a modal window titled 'Editar Recurso Recurso 1'. The form contains the following fields: 'Nombre' (text input with 'Recurso 1'), 'Descripción' (text area with 'Recurso 1'), 'Tipo de Recurso' (dropdown menu with 'Recurso Humano'), 'Cantidad' (text input with '1'), '\*Desde' (text input with '11/07/2016 21:17 PM'), and '\*Hasta' (text input with '13/07/2016 21:17 PM'). At the bottom right of the modal is a 'Guardar Cambios' button. The background shows a partial view of the sidebar and a 'Regresar' button.

Ilustración 63 Editar Recurso

3.11.34. Eliminar Un Recurso

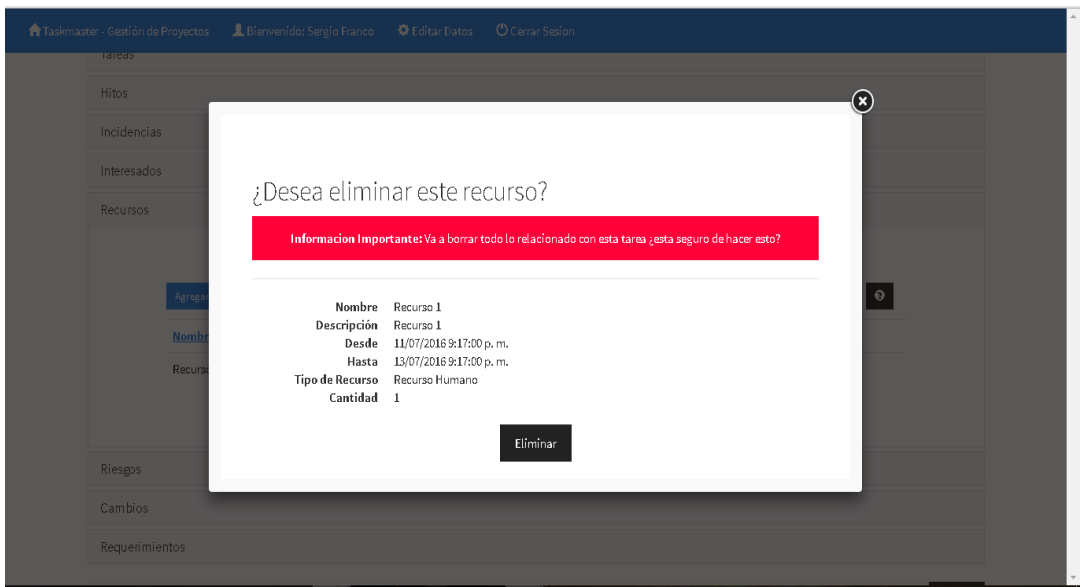


Ilustración 1 Eliminar Recurso

3.11.35. Ingresar Requerimiento

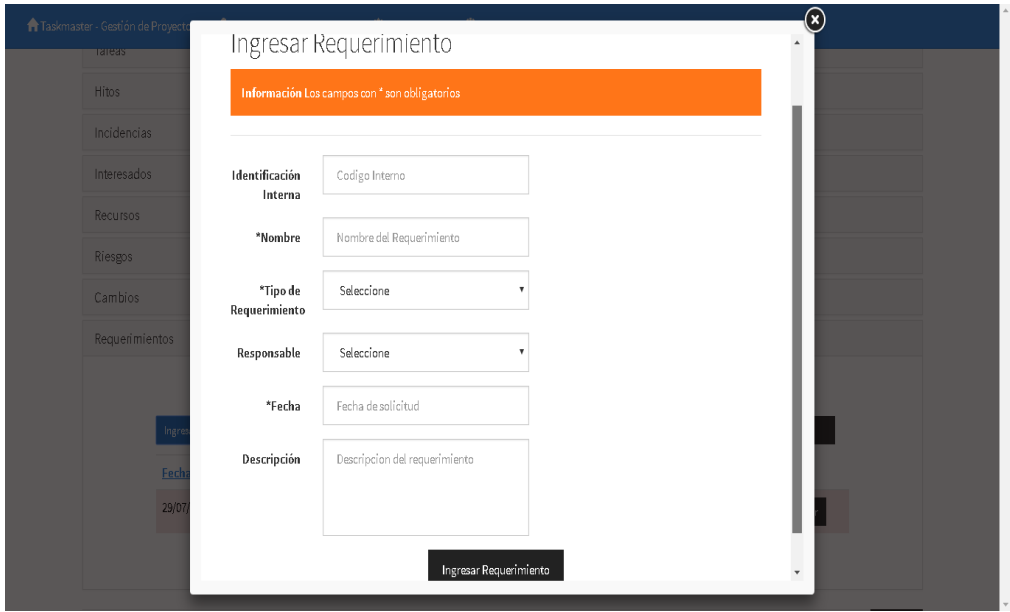
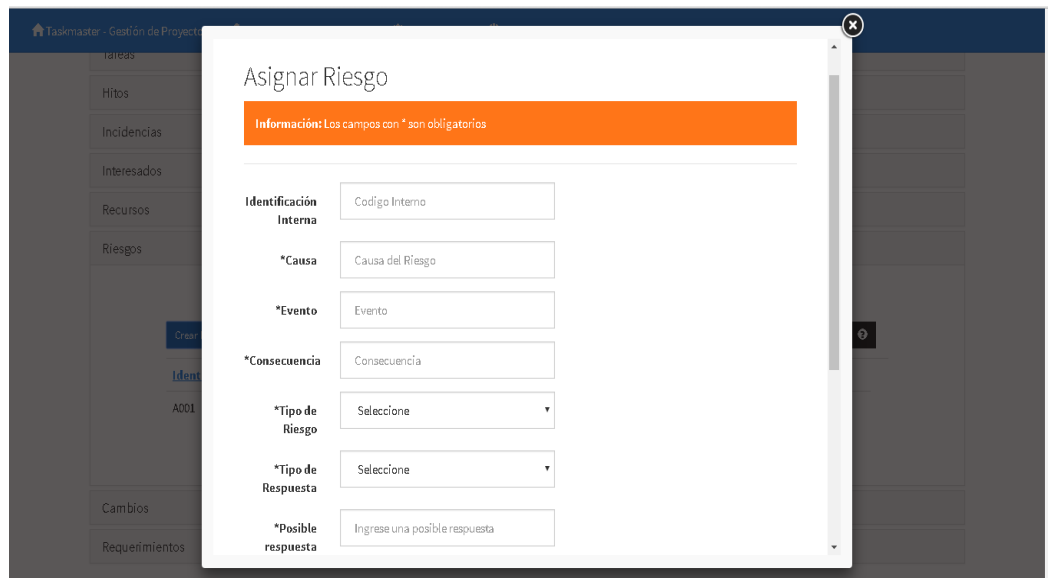


Ilustración 64 Ingresar Requerimiento

### 3.11.36. Asignar Riesgo



Taskmaster - Gestión de Proyectos

Asignar Riesgo

Información: Los campos con \* son obligatorios

Identificación Interna

Código Interno

\*Causa

Causa del Riesgo

\*Evento

Evento

\*Consecuencia

Consecuencia

\*Tipo de Riesgo

Seleccione

\*Tipo de Respuesta

Seleccione

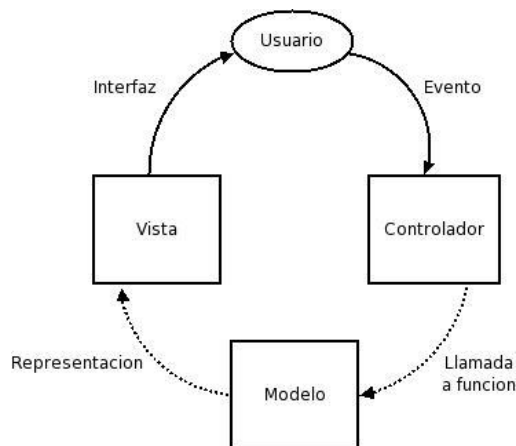
\*Posible respuesta

Ingrese una posible respuesta

Ilustración 65 Creación de Riesgo

## 1.11. PATRÓN DE ARQUITECTURA

Se hará uso de un patrón de arquitectura de Modelo-Vista-Controlador, esta elección se realiza porque permite un desarrollo ágil y los cambios que se realizan no afectan a otras vistas y controladores dentro de la aplicación.



**Ilustración 66 Patrón de Arquitectura MVC**

En este caso cada controlador es independiente, tiene sus propios métodos y sus propias vistas; en estos casos la lógica que se encuentra dentro de los controladores puede ser modificada de tal manera que no afecta la manera en que se muestran las vistas relacionadas.

Como ventaja se encuentra que el desarrollo se separa por capas, lo cual permite al desarrollador enfocarse en la programación de los controladores, ya que el modelo se carga en archivos independientes con sus propios tipos de datos; en el controlador se puede crear todos los métodos necesarios tanto para la lógica como para generar las vistas y por último se encuentran las clases que permiten mostrarle los resultados finales al usuario.

## **1.12. HERRAMIENTAS**

Para el desarrollo de la aplicación es necesario seleccionar las herramientas que se usaran para programar en el software, dichas herramientas y lenguajes de programación son:

- Microsoft Visual Studio 2013: Es un entorno de desarrollo integrado que permite el desarrollo de aplicaciones de manera robusta para Windows, aplicaciones web y aplicaciones para dispositivos móviles. Esta herramienta tiene compatibilidad con diversos lenguajes de programación como Visual Basic, C#, C++, a la vez posee soporte para HTML, JavaScript, CSS, etc. Este IDE posee varias ediciones, cada una se acomoda a las diferentes

necesidades de los desarrolladores: Professional, Ultimate, Premium y Test.

- SQL Server Express 2012: Es un sistema de gestión de base de datos relacionales, su función principal es la conexión, recepción y envío de datos por otras aplicaciones que pueden correr ya sea en el mismo equipo donde se desarrolla una aplicación o también puede obedecer a solicitudes hechas en otros equipos conectados a una red; al igual que con Visual Studio, Microsoft ofrece diversas versiones que se pueden ajustar a las necesidades del usuario y la finalidad del desarrollo.
- SQL Power Architect: Es una herramienta que permite el modelado del diagrama entidad relación para exportarlo hacia SQL Server.
- C#: Este lenguaje será usado para programar la lógica de las vistas y traer la información de la base de datos.
- SQL: lenguaje en el cual estará programada la base de datos.
- HTML: lenguaje de etiquetas que permitirá la visualización de los elementos del aplicativo web.
- Javascript: Permitirá funcionalidades adicionales que HTML, SQL y C# no permiten.
- JSON: permitirá el uso de funciones y métodos que no pueden ser accedidos de manera tradicional.
- CSS y Bootstrap: este lenguaje será usado para cuestiones estéticas de la aplicación, permitiendo personalización y características visuales amigables para el usuario.

### **1.13. TIPO DE APLICACIÓN**

La aplicación que se desarrolla es de tipo web; ya que el usuario accede desde un explorador web, también se podrá adaptar a tamaños y

resoluciones de distintos dispositivos que puedan acceder a internet, debido a que se usara tecnología responsiva que permitirá dichas acciones.

#### **1.13.1. Servicios a Ofrecer**

La aplicación ofrece los siguientes servicios:

- Creación de proyectos, a los cuales se les puede crear una serie de tareas.
- Creación de usuarios dentro del sistema para un uso personal.
- Compartir proyectos entre otros usuarios de la aplicación brindándole permisos de acceso y manipulación.
- Subir y descargar archivos relacionados con las tareas, proyectos, incidencias, requerimientos y riesgos.
- Listado con los interesados de un proyecto.
- Gestión de incidencias del proyecto.
- Gestión de recursos de un proyecto.
- Ingresar los requerimientos de un proyecto.
- Gestión de cambios de un proyecto.
- Ingresar los riesgos de un proyecto.



## 2. CONCLUSIONES

Tener en cuenta las fechas de inicio y de fin de un proyecto, sus tareas, entre otros elementos que dependan de este tipo de datos, ya que las validaciones dentro de la lógica de programación de los controladores influye en obtener o no errores en su ejecución; en caso de que los métodos creados en los controladores, pueden suceder fallos que no permitirán que la aplicación corra con normalidad.

El patrón de diseño MVC es una manera bastante útil de concebir el desarrollo de una aplicación, debido a la manera en que sus capas están concebidas; permitiendo que los cambios generados se puedan realizar más rápido, en caso de que el modelo sufra cambios solo es cuestión de modificar los datos y volver a subir el modelo diseñado; si los controladores son modificados, ni la vista ni el modelo sufren problemas agregando la lógica que el desarrollador desee; y las vistas pueden implementar varios lenguajes como CSS y Javascript permitiendo que las aplicaciones ganen más atractivo y funcionalidad.

Usar Frameworks y entornos de desarrollo se han convertido en una herramienta de uso práctico para muchos desarrolladores, ya que usar uno de estos programas significa ahorrarse tiempos de desarrollo e implementación de cosas tales como la conexión a la base de datos, o la carga de librerías que antes se debían hacer manualmente y ahora de manera automática se pueden precargar y modificar a gusto; permitiendo que los programadores se enfoquen en el desarrollo de la lógica de la aplicación.

Para desarrollar una aplicación es importante entorno de desarrollo que se use para trabajar porque dependiendo del lenguaje y el programa que se escoja pueden variar las librerías que se usan, la manera de ejecutar los scripts, también puede aumentar o disminuir el acceso a determinados pluggins que pueden mejorar el aplicativo a nivel visual y de implementación brindando mejores aspectos y métodos para usar.

Realizar un proyecto implica un gran esfuerzo por parte de todas las personas involucradas en este y por ello es necesaria la existencia de herramientas que faciliten a estas personas un control de este por lo cual requieren aplicativos con interfaces simples con las cuales puedan acceder de una manera rápida a los datos que necesitan y con ello tomar decisiones, anotar

sucesos importantes o anomalías que se presenten durante las etapas de estos y así mismo tomar decisiones y aprender lecciones para un futuro.

Poseer una herramienta que permita tener en control varios aspectos de un proyecto es útil, ya que se puede actuar con mayor rapidez en el momento de tomar una decisión que dependa de varios criterios.

El desarrollar una aplicación con una interfaz simple le permite al usuario ver los datos que necesita de una manera más fácil, ya que no tiene que lidiar con funcionalidades que desconoce, ni navegar por varias pestañas o sentirse confundido por la gran cantidad de información que pueda recibir en una sola pestaña.

### 3. BIBLIOGRAFIA E INFOGRAFIA

#### 5.1. BIBLIOGRAFIA

Bloch, M., Blumberg, S., & Laartz, J. (Octubre de 2012). *Insights & Publications*. Recuperado el 14 de Marzo de 2015, de Insights & Publications: [http://www.mckinsey.com/insights/business\\_technology/delivering\\_large-scale\\_it\\_projects\\_on\\_time\\_on\\_budget\\_and\\_on\\_value](http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value)

Canos, J., Letelier, P., & Penades, M. (2003). Metodologías Ágiles en el Desarrollo de Software. 1-8.

Capuz Rizo, S., Gomez-Senent Martinez , E., Torrealba Lopez, A., Ferret Gisbert, P., Gomez Navarro, T., & Vivancos Bono, J. L. (2000). *Cuadernos de Ingenieria de Proyectos III: Direccion, Gestion y Organizacion de Proyectos*. Valencia: Servicio de Publicaciones Universidad Politecnica de Valencia.

Direccion y Gestion de Empresas. (2008). *Gestion de Proyectos*. Malaga: Publicaciones Vertice S.L.

Fabregas, L. (2005). *Gerencia de Proyectos de Tecnología de Información*. Caracas: Editorial CEC, SA.

Figueroa, R., Solis, C., & Cabrera, A. (s.f.). Metodologias Tradicionales Vs Metodologias Agiles. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

Escuela de Organización Industrial. (01 de Enero de 2016). *GESTIÓN DE RIESGOS en Gestión de proyectos*. Obtenido de GESTIÓN DE RIESGOS en Gestión de proyectos: [http://www.eoi.es/wiki/index.php/GESTI%C3%93N\\_DE\\_RIESGOS\\_en\\_Gesti%C3%B3n\\_de\\_proyectos](http://www.eoi.es/wiki/index.php/GESTI%C3%93N_DE_RIESGOS_en_Gesti%C3%B3n_de_proyectos)

Gulla, J. (Febrero de 2012). *IMB Systems Magazine*. Recuperado el 14 de Marzo de 2015, de Seven Reasons IT Projects Fail: [http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project\\_pitfalls/?page=3](http://www.ibmsystemsmag.com/power/Systems-Management/Workload-Management/project_pitfalls/?page=3)

Institucion Universitaria de Envigado. (01 de 01 de 2016). *Los Recursos*. Obtenido de Los Recursos: <http://www.iue.edu.co/documents/emp/gestiorecursos.pdf>

Mieritz, L. (1 de Junio de 2012). *Gartner Survey Shows Why Projects Fail*. Recuperado el 14 de Marzo de 2015, de Gartner Survey Shows Why Projects Fail: <http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/>

Miguez Gonzales, M. (2007). Analisis del uso de Conceptos de Publico, Stakeholder y Constituent en el Marco Teorico de las Relaciones Publicas. 183-197.

Ministerio de Administraciones Publicas. (15 de 06 de 2015). *Portal de Administracion Electronica*. Obtenido de Portal de Administracion Electronica:  
[http://administracionelectronica.gob.es/pae\\_Home/dms/pae\\_Home/documentos/Documentacion/Metodologias-y-guias/Metricav3/METRICA\\_V3\\_Gestion\\_de\\_Proyectos.pdf](http://administracionelectronica.gob.es/pae_Home/dms/pae_Home/documentos/Documentacion/Metodologias-y-guias/Metricav3/METRICA_V3_Gestion_de_Proyectos.pdf)

Orjuela Duarte, A., & Rojas, M. (2008). Las Metodologias de Desarrollo Agil como una Oportunidad para la Ingenieria del Software Educativo. *Revista Avances en Sistemas e Informatica*, 159-171.

Pearson Education, Informit. (01 de 01 de 2016). *Introduction to the ICONIX Process of Software Modeling*. Obtenido de Introduction to the ICONIX Process of Software Modeling:  
<http://www.informit.com/articles/article.aspx?p=167902>

Pressman, R. (2006). Ingenieria del Software, Un Enfoque Practico. En R. Pressman, *Ingenieria del Software, Un Enfoque Practico* (págs. 502-503). Mexico D.F: Mc Graw Hill.

Project Management Institute. (2008). *Guia de los Fundamentos para la Direccion de Proyectos (Guia del PMBOK) - Cuarta Edicion*. Newton Square: Project Management Institute.

Sols Rodriguez-Candela, A., Fernandez Fernandez, I., & Romero Yacobi , J. (2013). *La Gestión Integral de Proyectos*. Madrid: Pontificia Universidad Comillas.

Sommerville, I. (2005). *Ingeniería del Software Septima Edición*. Madrid: Pearson Education.

## 5.2. INFOGRAFIA

Basecamp. (01 de 01 de 2016). *Basecamp*. Obtenido de Basecamp: <https://basecamp.com/>

Bitrix, Inc. (01 de 01 de 2016). *Bitrix24 - Your company. United*. Obtenido de Bitrix24 - Your company. United.: <https://www.bitrix24.com/>

Choudhury, A. (13 de 03 de 2015). *Software Development Life Cycle (sdlc) Tutorials*. Obtenido de Software Development Life Cycle (sdlc) Tutorials: <https://www.sdlc.ws/win-win-spiral-model/>

Christensen, B. (2009). *GanttPV*. Recuperado el 9 de 10 de 2013, de GanttPV: <http://www.pureviolet.net/ganttpv/index.html>

Freedcamp Inc. (01 de 01 de 2016). *Freedcamp*. Obtenido de Freedcamp: <https://freedcamp.com/>

GearHost <sup>TM</sup> Inc. (01 de 01 de 2016). *CloudSite Pricing*. Obtenido de CloudSite Pricing: <https://www.gearhost.com/pricing>

Microsoft. (01 de 01 de 2016). *Buscar hospedaje web...* Obtenido de Buscar hospedaje web...: <https://hosting.asp.net/hosting/home>

Microsoft Press. (13 de 03 de 2015). *Microsoft Solutions Framework (MSF) Overview*. Obtenido de Microsoft Solutions Framework (MSF) Overview: <https://msdn.microsoft.com/en-us/library/jj161047%28v=vs.120%29.aspx?f=255&MSPPErr=-2147217396>

myASP.NET. (01 de 01 de 2016). *Shared ASP.Net Hosting Plans*. Obtenido de Shared ASP.Net Hosting Plans: [http://www.myasp.net/hosting\\_plans](http://www.myasp.net/hosting_plans)

MYWINDOWSHOSTING.COM. (01 de 01 de 2016). *Shared Windows Hosting Plan*. Obtenido de Shared Windows Hosting Plan: [http://www.mywindowshosting.com/hosting\\_plans](http://www.mywindowshosting.com/hosting_plans)

Open Project Foundation. (2013). *Open Project*. Recuperado el 9 de 10 de 2013, de Open Project: <https://www.openproject.org/>

OpenProject. (01 de 01 de 2016). *Open Project*. Obtenido de Open Project: <https://www.openproject.org/>

Rational, The Software Development Company. (Noviembre de 2001). *IBM*. Obtenido de IBM: [https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\\_bestpractices\\_TP026B.pdf](https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf)

Rosenberg, D., & Scott, K. (12 de 10 de 2001). *Introduction to the ICONIX Process of Software Modeling*. Obtenido de Introduction to the ICONIX Process of Software Modeling: <http://www.informit.com/articles/article.aspx?p=167902>

Rozemeijer, E., & Wellington Redwood, Q. (2007). Frameworks for IT Management - A Pocket Guide. En E. Rozemeijer, & Q. Wellington Redwood, *Frameworks for IT Management - A Pocket Guide* (págs. 135-139). Wilco: Van Haren Publishing.

Signals LLC. (1999-2013). *Basecamp*. Recuperado el 9 de 10 de 2013, de Basecamp: <https://basecamp.com/>

Simonsen, E., & Simonsen, E. (2003 - 2008). *Clocking IT*. Recuperado el 8 de 10 de 2013, de Clocking IT: <http://www.clockingit.com/>

SmarterASP.NET. (01 de 01 de 2016). *Superior Shared ASP.NET Hosting*. Obtenido de Superior Shared ASP.NET Hosting: [http://www.smarterasp.net/hosting\\_plans](http://www.smarterasp.net/hosting_plans)

Zoho Corporation Pvt. Ltd. (01 de 01 de 2016). *Zoho* . Obtenido de Zoho: <https://www.zoho.com/>