

**APLICATIVO PARA LA GESTIÓN DE SOLICITUDES EN EL ÁREA DE
INFRAESTRUCTURA TECNOLÓGICA DE LA UNIVERSIDAD DE PAMPLONA,
SEDE VILLA DEL ROSARIO.**

YURI TATIANA MEDINA VARGAS



UNIVERSIDAD DE PAMPLONA

FACULTAD DE INGENIERÍAS Y ARQUITECTURA

DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y

TELECOMUNICACIONES

INGENIERÍA DE SISTEMAS

VILLA DEL ROSARIO

2016

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y TELECOMUNICACIONES
INGENIERÍA DE SISTEMAS.

TRABAJO PRESENTADO PARA OPTAR POR ÉL TÍTULO DE
INGENIERO DE SISTEMAS

TÍTULO: APLICATIVO PARA LA GESTIÓN DE SOLICITUDES EN EL ÁREA DE INFRAESTRUCTURA TECNOLÓGICA DE LA
UNIVERSIDAD DE PAMPLONA, SEDE VILLA DEL ROSARIO.

FECHA DE INICIO DEL TRABAJO: AGOSTO DE 2016
FECHA DE TERMINACIÓN DEL TRABAJO: DICIEMBRE DE 2016

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA SUSTENTAR:

YURI TATIANA MEDINA VARGAS

AUTOR

MSC. JOHEL ENRIQUE RODRIGUEZ

DIRECTOR

ING-ESP. JOHANN LEONARDO LATORRE JAIMES

DIRECTOR DEL PROGRAMA

JURADO CALIFICADOR:

PhD.

PRESIDENTE

MSc.

OPONENTE

ING-ESP JOHANN LEONARDO LATORRE JAIMES

SECRETARIO

DEDICATORIA

Dedico Este Logro De Mi Vida A Mi DIOS EL ES EL MERECEDOR DE LA HONRA Y LA GLORIA, El Es Mi Motor Y Gracias A El Esta Nueva Meta Se Cumplió.

Dedico Este Triunfo A Mi Mami... Mi Motor de vida, La Persona Por La Cual he salido adelante la guerrera de mi casa...

A mi papa y a mis hermanos Julián y Jonathan por su apoyo.

Dedico con mucha gratitud este triunfo al ingeniero, hermano, amigo y parcerero HAIDER MIRANDA, por su apoyo incondicional a lo largo de esta carrera, pero sobre todo a lo largo de mi vida, me enseñaste que vivimos para servir.

Dedico también este triunfo a Alicia Lopera e Isabel cristina por enseñarme a creer en mí y por su apoyo incondicional.

A mis amigos Sergio y Yajaira López por su ayuda inmensa en este proceso.

A mis amigos Michael Angarita, Jessica duarte y familia, Marial del Mar Botello, Laura, Angélica, Jessica daza, etc.

A mi tía Marina Penagos y mi prima Gina Blanco, por su apoyo incondicional.

AGRADECIMIENTOS

Le agradezco Primeramente a papito Dios por haberme acompañado y guiado a lo largo de mi carrera, por ser mí Fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes, Experiencias, porque he llegado hasta aquí porque él me lo ha permitido y por él seguiré adelante.

Le doy gracias a mis padres Flor y Marcos por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo a mi mamá por ser un excelente ejemplo de vida a seguir.

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial a los ingenieros Johel Rodriguez, director de esta investigación y el ing. Jeffrey Balmaceda codirector, Anderson Florez, Yajaira Lizcano y Haider Miranda gracias por la orientación, el seguimiento y la supervisión continúa de la misma, pero sobre todo por la motivación y el apoyo recibido a lo largo de estos años.

A mis hermanos por ser parte importante de mi vida y representar la unidad familiar. A Haider miranda por ser un ejemplo de desarrollo profesional a seguir y por haberme apoyado en las buenas y en las malas, sobre todo por su paciencia y amor incondicional.

A Alicia por ser una excelente amiga, por motivarme a seguir adelante en los momentos de desesperación y sobre todo por hacer de su familia, una familia para mí.

A mis amigos Michael y Jessy por confiar y creer en mí y haber estado apoyándome en cada momento de este proceso y etapa de mi vida.

A mi tía Marina Penagos y prima Gina blanco por su apoyo a lo largo de mi carrera por su motivación y por creer en mí, por brindarme su mano en momentos difíciles y porque ustedes son mi familia.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos.

A todos ellos, muchas gracias.

Son muchas las personas que han formado parte de mi vida profesional a

Las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y

Compañía en los momentos más difíciles de mi vida. Algunas están aquí

Conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde

Estén quiero darles las gracias por formar parte de mí, por todo lo que me

Han brindado y por todas sus bendiciones.

Resumen

Actualmente en la Universidad de Pamplona (Villa del Rosario) no tiene ningún aplicativo para controlar y gestionar las solicitudes hechas por el personal administrativo, la manera que utilizan los usuarios para hacer sus solicitudes es de forma verbal, manual o telefónicamente, para así comunicarse con la persona encargada del área de infraestructura tecnológica.

Pero estas necesidades que se presentan en la Universidad de Pamplona, sede Villa del Rosario varían de acuerdo al crecimiento en los avances de la infraestructura tecnológica, haciendo que sea interactiva entre los usuarios que la acceden, actualmente se presentan varios problemas en el manejo y control de las solicitudes que hace el personal administrativo, ya que ellos son los encargados de manejar los diferentes sistemas de la universidad, estos problemas conllevan a una pérdida de tiempo en el proceso de gestión y solución de las solicitudes porque el personal no conocen las actividades que realiza el área de infraestructura tecnológica o no tienen un medio establecido para realizar el proceso de hacer sus solicitudes.

El objetivo de este proyecto consiste en desarrollar un aplicativo para controlar las solicitudes que se realizan al área de infraestructura tecnológica de Universidad de Pamplona, sede Villa Del Rosario, este será un aplicativo que permitirá organizar, identificar, gestionar y dar solución en un menor tiempo y de forma eficiente a los requerimientos y solicitudes que presentan el personal administrativo de la universidad, manteniendo la integridad entre los aspectos importantes de esta organización y así poder brindar una mejor atención en el desarrollo en las diferentes actividades de la Universidad de Pamplona, sede Villa del Rosario.

Palabras claves: control de solicitudes, jboss, netbeans, java, aplicación.

Abstract

Currently at the University of Pamplona (Villa del Rosario) has no application to control and manage requests made by the administrative staff, the way users use to make their applications is verbally, manually or by telephone, in order to communicate with the person in charge of the area of infrastructure.

But these needs that arise at the University of Pamplona, home Villa del Rosario vary according to the growth in advances in technology infrastructure, making it interactive between users that access, several problems are now presented in the management and control requests made by the administrative staff, since they are responsible for managing the different systems of the university, these problems lead to a waste of time in the process of management and resolution of applications because the staff do not know the activities takes the area of technological or infrastructure have no means established for the process of making their applications.

The objective of this project is to develop an application to handle requests that are made to the area of technology infrastructure University of Pamplona, home Villa Del Rosario, this will be an application that allows organizing, identify, manage and resolve in less time and efficiently to the needs and requests present the administrative staff of the university, maintaining the integrity among the important aspects of this organization so we can provide better care in development in the various activities of the University of Pamplona, home Villa Rosary.

Keywords: control applications, jboss, netbeans java application.

Contenido

	Pág.
Introducción	19
1. Descripción del problema	20
1.1 Planteamiento del problema	20
1.2 Justificación	20
1.3 Objetivos	21
1.4 Acotaciones	21
2. Referentes Teóricos	22
2.1 Antecedentes	22
2.2 Marco conceptual	24
2.2.1 Conceptualización de las variables	24
2.2.1.1 Control de las solicitudes	24
2.2.1.1.1 Control	24
2.2.1.1.2 Seguimiento	24
2.2.1.2 Eficiencia en la atención y solución	24
2.2.1.2.1 Eficiencia	24
2.2.2Aplicativo	25
2.2.3 Ciclo De Vida Clásico.	27
2.2.3.1 Metodología Scrum	28
2.2.3.1.1 Características De Metodología Scrum.	29
2.2.3.1.2 Scrum emplea las siguientes prácticas de la gestión ágil.	30
2.2.4 Ingeniería De Requerimiento	31
2.2.4.1 Obtención De Requerimientos.	31
2.2.4.2 Análisis Y Especificación De Requerimientos.	32
2.2.4.3 Requerimientos.	33
2.2.4.3.1 Tipos de requerimientos.	34

2.2.4.3.1.1 Tipos de requerimientos de acuerdo con la audiencia.	34
2.2.4.3.1.2 Tipos De Requerimientos De Acuerdo Con Su Característica	34
2.2.4.4 Análisis Y Diseño De Requerimientos.	35
2.2.5 Diseño	36
2.2.5.1 Lenguaje unificado de modelado (uml).	36
2.2.5.1.1 Diagrama de casos de usos:	37
2.2.5.1.2 Diagramas de Clases	39
2.2.5.1.3 Diagrama de actividad	40
2.2.5.1.4 Diagrama de secuencia	42
2.2.5.2 Modelo entidad y relación	44
2.2.6 Herramientas De Diseño	44
2.2.7 Construcción	45
2.2.7.1 Base de datos	45
2.2.7.2 Patrón de arquitectura	46
2.2.7.2.1 Arquitectura de la aplicación	46
2.2.7.2.2 Modelo-Vista-Control (MVC).	48
2.2.7.3 Interfaz gráfica del usuario	49
2.2.7.4 Herramientas para la construcción	49
2.2.7.4.1 Lenguaje de programación	50
2.2.7.4.1.1 Java.	50
2.2.7.4.1.2 Html5	50
2.2.7.4.2 Entorno de desarrollo	51
2.2.7.4.2.1 Netbeans.	51
2.2.7.4.2.2 JDK.	52
2.2.7.4.2.3 Apache Tomcat	53
2.2.7.4.3 Sistema de Gestión de Base de Datos.	54
2.2.7.4.3.1 PostgreSQL	54

2.2.7.4.4 Frameworks	55
2.2.7.4.4.1 JBoss Seam	56
2.2.7.4.4.1.1 Richfaces	59
2.2.7.4.5 Servidor Jboss.	59
3 Diseño Y Construcción De La Aplicación Web	60
3.1 Requerimientos	60
3.2 Análisis Del Sistema	62
3.2.1 Descripción De Requisitos	62
3.2.1.1 Requerimientos funcionales	62
3.2.1.2 Requerimientos no funcionales	63
3.2.1.3 Entregables	64
3.3 Implementación de metodología Scrum	67
4. Desarrollo Del Proyecto	69
4.1 Fase de Análisis	69
4.1.1 Documento de análisis	69
4.1.1.1 Historias de usuario	69
4.1.1.2 Diagrama de casos de uso	76
4.1.1.2.1 Especificaciones de los casos de uso	77
4.2 Diseño De La Base De Datos	98
4.2.1 Entidades.	98
4.2.2 Relaciones	99
4.2.3 Diagrama de clases	102
4.2.4 Diagrama de objetos	102
4.2.4 Modelo de despliegue.	103
4.2.6 Diagrama De Secuencia	104
4.3 Diseño De Interfaz Abstracta	114
5 Fase De Implementación.	117

5.1 Implementación Del Producto Con La Metodología Scrum	117
5.2 Planificación	117
5.3 Ejecución De La Iteración.	117
5.4 Reportes	117
5.4.1 Sprint Review Meeting	117
5.4.2 Sprint Retrospective	118
5.5 Retrospectiva	118
5.5.1 Descripción del software	118
6 Cronograma De Actividades	119
7 Fase de Pruebas	121
7.1 Documento de Pruebas	121
7.2 Historial de Revisiones	121
7.3 Plan de Pruebas	121
7.3.1 Introducción	122
7.3.2 Ámbito de las Pruebas	122
7.3.3 Lista de Ideas de las Pruebas.	122
7.3.4 Enfoque de las Pruebas	123
7.3.5 Herramientas para las Pruebas	124
7.3.6 Casos de Prueba	124
7.3.7 Entregables	144
7.3.8 Resultados/Observaciones para el Ciclo de Prueba	144
7.3.9 Criterio para el Inicio y Fin del Plan de Pruebas	144
8. Implantación	145
8.1 Inicio del Funcionamiento del Sistema	145
8.2 Capacitación de Personal	146
Anexos	147
Conclusiones	161

Recomendaciones	162
Referencias Bibliográficas	163

Lista De Tablas

	Pág.
Tabla 1. Requisitos de Usuario	61
Tabla 2. Requerimientos funcionales.	62
Tabla 3. Requerimientos no funcionales.	63
Tabla 4: Entregables	64
Tabla 5: Manuales.	65
Tabla 6: Documentación	66
Tabla 7: Historia de usuario 1	69
Tabla 8: Historia de usuario 2	69
Tabla 9: Historia de usuario 3	70
Tabla 10: Historia de usuario 4	70
Tabla 11: Historia de usuario 5	71
Tabla 12: Historia de usuario 6	71
Tabla 13: Historia de usuario 7	72
Tabla 14: Historia de usuario 8	72
Tabla 15: Historia de usuario 9	73
Tabla 16: Historia de usuario 10	73
Tabla 17: Historia de usuario 11	74
Tabla 18: Historia de usuario 12	74
Tabla 19: Historia de usuario 13	75
Tabla 20: Historia de usuario 14	75
Tabla 21. Caso de uso Iniciar sesión	77
Tabla 22. Caso de uso Finalizar Sesión	78
Tabla 23. Caso de uso Cambiar contraseña	79

Tabla 24. Caso de uso agregar usuario	80
Tabla 25. Caso de uso agregar incidencia	82
Tabla 26. Caso de uso agregar dependencia.	82
Tabla 27. Caso de uso listar usuarios	83
Tabla 28. Caso de uso listar incidencias	84
Tabla 29. Caso de uso listar dependencias	85
Tabla 30. Caso de uso editar usuario.	86
Tabla 31. Caso de uso editar incidencia	87
Tabla 32. Caso de uso Editar dependencia	88
Tabla 33. Caso de uso ver usuario	89
Tabla 34. Caso de uso ver incidencia	90
Tabla 35. Caso de uso ver dependencia	91
Tabla 36. Caso de uso crear solicitud.	92
Tabla 37. Caso de uso listar solicitud.	93
Tabla 38. Caso de uso cerrar solicitud	94
Tabla39. Caso de uso calificar solicitud.	95
Tabla 40. Caso de uso agregar seguimiento a solicitud.	96
Tabla 41. Caso de uso generar informe	97
Tabla 42. Implementación de la arquitectura de 3 niveles	115
Tabla 43. Cronograma de Actividades	119
Tabla 44 Historial de Revisiones	121
Tabla 45 Enfoque de las Pruebas.	123
Tabla 46 Hardware	124
Tabla 47 Casos de Prueba 1	124

Tabla 48 Casos de Prueba 2	125
Tabla 49 Casos de Prueba 3	126
Tabla 50 Casos de Prueba 4	127
Tabla 51 Casos de Prueba 5	128
Tabla 52 Casos de Prueba 6	129
Tabla 53 Casos de Prueba 7	130
Tabla 54 Casos de Prueba 8	131
Tabla 55 Casos de Prueba 9	132
Tabla 56 Casos de Prueba 10	133
Tabla 57 Casos de Prueba 11	134
Tabla 58 Casos de Prueba 12	135
Tabla 59 Casos de Prueba 13	136
Tabla 60 Casos de Prueba 14	137
Tabla 61 Casos de Prueba 15	138
Tabla 62 Casos de Prueba 16	139
Tabla 63 Casos de Prueba 17	140
Tabla 64 Casos de Prueba 18	141
Tabla 65 Casos de Prueba 19	142
Tabla 66 Casos de Prueba 20	143
Tabla 67 Lista de Entregables de Pruebas	144
Tabla 68 Nivel Prioridad	149
Tabla 69 Tipo de Incidencia	149

Lista De Figuras

	Pág.
Figura 1: Ingeniería Del Software.	27
Figura 2. Metodología Scrum.	28
Figura 3: Obtención de requerimientos.	32
Figura 4: Análisis especificación de requerimientos.	33
Figura 5: Diagrama de casos de uso.	38
Figura 6: Diagrama de clases	40
Figura 7: Diagrama de actividades	42
Figura 8: Diagrama de secuencias	43
Figura 9: Modelo De Tres Capas	47
Figura 10: Esquema del Patrón MVC	48
Figura 11: La estructura de directorios del JDK/J2SE.	53
Figura 12: Arquitectura de una aplicación Seam	58
Figura 13. Ilustración 1 Estructura Desglosada de Trabajo – EDT	67
Figura 14: Diagrama de casos de uso	76
Figura 15: Base De Datos	99
Figura 16: Relación Usuario - Solicitud	100
Figura 17: Relación Solicitud – Seguimiento.	100
Figura 18: Relación Dependencia - Solicitud.	101
Figura 19: Relación incidencia - Solicitud.	101
Figura 20. Diagrama de clases	102
Figura 21. Diagrama de objetos	102
Figura 22. Modelo despliegue	103
Figura 23 Diagrama de secuencia, iniciar sesión	104

Figura 24 Diagrama de secuencia, finalizar sesión	104
Figura 25 Diagrama de secuencia, cambiar contraseña.	105
Figura 26 Diagrama de secuencia, crear usuario	105
Figura 27 Diagrama de secuencia, crear dependencia	106
Figura 28 Diagrama de secuencia, crear incidencia	106
Figura 29 Diagrama de secuencia, listar usuarios	107
Figura 30 Diagrama de secuencia, listar dependencia	107
Figura 31 Diagrama de secuencia, listar incidencia	108
Figura 32 Diagrama de secuencia, ver usuario	108
Figura 33 Diagrama de secuencia, ver dependencia	109
Figura 34 Diagrama de secuencia, ver incidencia	109
Figura 35 Diagrama de secuencia, editar usuario	110
Figura 36 Diagrama de secuencia, editar dependencia	110
Figura 37 Diagrama de secuencia, editar incidencia	111
Figura 38 Diagrama de secuencia, crear solicitud	111
Figura 39 Diagrama de secuencia, listar solicitud	112
Figura 40 Diagrama de secuencia, agregar seguimiento	112
Figura 41 Diagrama de secuencia, cerrar solicitud	113
Figura 42 Diagrama de secuencia, calificar solicitud	113
Figura 43 Descripción del diseño de la interfaz de la página de inicio de la Aplicación	115
Figura 44 Ingreso Al Sistema	146
Figura 45 Darle Click en Iniciar Sesión	146
Figura 46 Digite su Usuario y Clave	147
Figura 47 Página de Inicio	147

Figura 48 Crear Solicitudes	148
Figura 49 Llenar Los Espacio Requeridos	148
Figura 50 Nivel Prioridad	149
Figura 51 Tipo de Incidencia	149
Figura 52 Descripción Breve de la Solicitud	150
Figura 53 Mensaje Exitoso de la Creación de la Solicitud	151
Figura 54 Al darle Guardar	151
Figura 55 Listar las Solicitudes Pendientes	152
Figura 56 Lista de solicitudes	152

Anexos

	Pág.
Anexo 1. Manual del usuario	145
Anexo 2. Manual de recepción de solicitudes en el área de infraestructura tecnológica	154

Introducción

Podemos definir que este aplicativo sea el punto central de contacto entre el departamento de Infraestructura tecnológica y las diferentes áreas de la Universidad de Pamplona, donde podrán encontrar solución a los problemas relacionados con hardware y software o de recibir respuestas a sus consultas. La idea de diseñar este software es reducir los pasos que se realizaban, como el desplazamiento del personal técnico para la solución de los incidentes en las diferentes sucursales, causando demoras y pérdidas en las áreas comerciales.

La gran necesidad de crear este aplicativo para la Universidad de Pamplona, se debe a que no existe una forma sistematizada de tener la información clara y concisa de los equipos y software de la universidad, no se puede tener un control sobre el estado de los mismos y poder atenderlos, de una manera eficiente y rápida que permita mejorar los procesos en cada punto área de la universidad.

La propuesta de este proyecto de grado es mostrar cómo será el prototipo inicial del aplicativo para el Área de Infraestructura Tecnológica, como una herramienta tecnológica que mejore los procesos de administración de equipos, levantamiento de incidencia, soporte y solución de los diferentes que se presenten en la universidad (computadores, impresoras, servidores, etc.) y que además se pueda llevar a través de una base de datos un historial de servicios.

1. Descripción del problema

1.1 Planteamiento del problema

En la Universidad de Pamplona el avance y desarrollo tecnológico ha ido de la mano con los cambios que suceden en el entorno de esta organización educativa, lo que origina una oportunidad y la necesidad de adaptación de los sistemas a dicho cambio en el crecimiento de la infraestructura tecnológica, estos cambios en los sistemas de información que se manejan a través de sus desarrollos internos genera que sus usuarios presenten solicitudes con respecto a fallas, daños o problemas en dichos sistemas, estas solicitudes se hacen de manera personal, telefónica, verbal o por medio de correo electrónico, haciendo de este un proceso largo y demorado, también que no tenga un buen control y seguimiento, lo que afecta el funcionamiento de cada proceso asignado a los usuarios, ya sea el área administrativa o estudiantil, ya que no se tiene un método o lugar fijo y establecido para realizar dichas peticiones lo que hace que algunas solicitudes se pierdan y no sean atendidas.

En la universidad de Pamplona Sede Villa Del Rosario, en el área de infraestructura tecnológica no se cuenta con una aplicación que permita tener un control y seguimiento de las solicitudes realizadas por los usuarios.

1.2 Justificación

Con el problema expuesto, en este proyecto se plantea desarrollar una aplicación que sea capaz de controlar, organizar, gestionar y llevar a cabo un mejor procesamiento de estas solicitudes, obteniendo así respuestas en menor tiempo y de manera eficiente para brindar una mejor atención a las solicitudes de los servicios requeridos al área de infraestructura tecnológica, que hacen los usuarios administrativos de la universidad, además se quiere almacenar la

información detallada de equipos (hardware y software), historiales de soportes de todas las solicitudes y llevar un control específico de los principales factores que causan los problemas, dando soluciones en corto tiempo y prestando un servicio de calidad.

El objetivo principal de este proyecto es el aportar al área de infraestructura tecnológica un aplicativo que ayude a controlar el procedimiento de gestión de las solicitudes hechas y alcanzar la eficiencia en la prestación de servicios.

1.3 Objetivos

Objetivo general. Controlar la gestión de solicitudes realizadas por el personal administrativo al área de infraestructura tecnológica de la Universidad de Pamplona, sede Villa del Rosario.

Objetivos específicos. Mejorar el proceso de solicitudes, minimizando el tiempo y facilitándolo al momento de diligenciar la solicitud por medio del aplicativo.

Gestionar los requerimientos técnicos de las solicitudes recibidas por parte del área de infraestructura tecnológica de la universidad de pamplona sede Villa del Rosario.

Generar informes de gestión de forma ágil desde el aplicativo con la información almacenada en la base de datos.

1.4 Acotaciones

El diseño está limitado solo al área de infraestructura tecnológica, universidad de Pamplona sede Villa Del Rosario.

El diseño está limitado al acceso de herramientas libres.

2. Referentes Teóricos

2.1 Antecedentes

Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación, 19(4), 413–418. Se presenta un marco de trabajo basado en las prácticas de SCRUM, aplicado para gestionar el diseño de las distintas versiones del proyecto del Trabajo Final de Aplicación, documento que explicita un producto tecnológico a desarrollar para la titulación de grado. En la propuesta se aplica SCRUM desde la concepción de la idea, en el proceso de elaboración del proyecto y finalizando con su presentación para su aprobación formal.

Edison x, Siza r. - José r, Guanotasig c. Diseño e implementación de un sistema web utilizando java y empleando la metodología scrum, para la gestión administrativa de la junta parroquial de guaytacama perteneciente al cantón latacunga de la provincia de cotopaxi. Actualmente en cualquier empresa o institución se desenvuelve una cierta cantidad de tareas que se apoyan en la gestión administrativa de la misma. Para que se produzca el correcto desarrollo de la gestión administrativa es necesario que la misma contribuya en una mejora para la eficiencia en el seno de la organización. Como complemento imprescindible no podemos ignorar el hecho de que en la actualidad para lograr el desarrollo de una gestión administrativa se debe contar con sistemas de apoyo, que permitan optimizar el trabajo de la persona responsable de la gestión administrativa en general a lo que se refiere al manejo de los recursos económicos, materiales y humanos para el cumplimiento de los objetivos de la institución, lo que facilitara dar un informe concreto de las inversiones de los diferentes proyectos que se están realizando en la parroquia Guaytacama, siendo beneficiados directos la ciudadanía, el diseño e Implementación de un Sistema Web utilizando Java y empleando la Metodología Scrum, para la Gestión

Administrativa se utiliza nuevos avances tecnológicos enmarcado en la utilización de herramientas de programación actualizadas. La adopción del software libre permite que la institución ahorre una considerable suma de dinero para aplicar a otras áreas, de la misma manera que contribuye a vencer la dependencia tecnológica. Y además, al ser libre, la institución, a través de la persona encargada, queda en capacidad de desarrollar las aplicaciones que cada uno de los componentes de la administración necesita en función de sus requerimientos y necesidades, a la vez de establecer sus propios estándares de seguridad.

Andrea, J., & Palencia, R. (2016). *DISEÑO DE UNA APLICACIÓN WEB RESPONSIVE PARA CONTROL EN LA ENTREGA DE FACULTAD DE INGENIERÍAS Y ARQUITECTURA DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA , ELECTRÓNICA , SISTEMAS Y INGENIERÍA DE SISTEMAS*. UNIVERSIDAD DE PAMPLONA.

El, P., Integral, D., & Sinaloa, U. (2009). Instituto politécnico nacional.

RICARDO, G. C. J., & XAVIER, S. R. E. (2012). Diseño E Implementación De Un Sistema Web Utilizando Java Y Empleando La Metodología Scrum, Para La Gestión Administrativa De La Junta Parroquial De Guaytacama Perteneciente Al Cantón Latacunga De La Provincia De Cotopaxi, 115.

Actualmente en cualquier empresa o institución se desenvuelve una cierta cantidad de tareas que se apoyan en la gestión administrativa de la misma. Para que se produzca el correcto desarrollo de la gestión administrativa es necesario que la misma contribuya en una mejora para la eficiencia en el seno de la organización. Como complemento imprescindible no podemos ignorar el hecho de que en la actualidad para lograr el desarrollo de una gestión administrativa se debe contar con sistemas de apoyo, que permitan optimizar el trabajo de la persona responsable de la gestión administrativa en general a lo que se refiere al manejo de los recursos económicos, materiales y humanos para el cumplimiento de los objetivos de la institución, lo que facilitara dar un informe concreto de las inversiones de los diferentes proyectos que se están realizando en la

parroquia Guaytacama, siendo beneficiados directos la ciudadanía, el diseño e Implementación de un Sistema Web utilizando Java y empleando la Metodología Scrum, para la Gestión Administrativa se utiliza nuevos avances tecnológicos enmarcado en la utilización de herramientas de programación actualizadas. La adopción del software libre permite que la institución ahorre una considerable suma de dinero para aplicar a otras áreas, de la misma manera que contribuye a vencer la dependencia tecnológica. Y además, al ser libre, la institución, a través de la persona encargada, queda en capacidad de desarrollar las aplicaciones que cada uno de los componentes de la administración necesita en función de sus requerimientos y necesidades, a la vez de establecer sus propios estándares de seguridad.

2.2 Marco conceptual

2.2.1 Conceptualización de las variables:

2.2.1.1 Control de las solicitudes

2.2.1.1.1 Control. La Real academia española define control como Comprobación o inspección de una cosa; Controlar como Verificar o comprobar el funcionamiento o evolución de una cosa; Solicitud, documento o memorial en que se solicita algo; solicitar, Pedir o pretender una cosa. [1]

2.2.1.1.2 Seguimiento. El diccionario de la lengua española define seguir como: Estar después o detrás de una persona o cosa; acompañar con la vista a un objeto que se mueve; observar algo con atención, estar atento a ello; seguimiento como persecución, acción de seguir; vigilancia, observación detallada

En base a estas definiciones y para esta investigación, el control y seguimiento de las solicitudes es el proceso de comprobar y vigilar que se atiendan todas las solicitudes.

2.2.1.2 Eficiencia en la atención y solución

2.2.1.2.1 Eficiencia.

1. Según Idalberto Chiavenato, eficiencia "significa utilización correcta de los recursos (medios de producción) disponibles. Puede definirse mediante la ecuación $E=P/R$, donde P son los productos resultantes y R los recursos utilizados". (Chiavenato Idalberto (2004)).
2. Para Koontz y Weihrich, la eficiencia es "el logro de las metas con la menor cantidad de recursos". (Koontz Harold y Weihrich Heinz, (2004)).
3. Según Robbins y Coulter, la eficiencia consiste en "obtener los mayores resultados con la mínima inversión".(Robbins Stephen y Coulter Mary (2005)).
4. Para Reinaldo O. Da Silva, la eficiencia significa "operar de modo que los recursos sean utilizados de forma más adecuada". (Oliveira Da Silva Reinald (2002)).

En base a estas definiciones podemos decir que "Eficiencia es el aprovechar los recursos disponibles para la obtención de resultados deseados". Para esta investigación la variable Eficiencia en la atención y solución será determinada por la solución completa y oportuna de cada una de las solicitudes.

2.2.2 Aplicativo.

En la ingeniería software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un

lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. Las páginas Web no son más que ficheros de texto en un formato estándar denominado HTML (HyperText Markup Language). Estos ficheros se almacenan en un servidor web al cual se accede utilizando el protocolo HTTP (HyperText Transfer Protocol), uno de los protocolos de Internet.

Desde el punto de vista funcional una aplicación web es un SI (Sistema de Información) donde una gran cantidad de datos volátiles, altamente estructurados, van a ser consultados, procesados y analizados mediante navegadores. Una de las principales características va a ser su alto grado de interacción con el usuario, y el diseño de su interfaz debe ser claro, simple y debe estar estructurado de tal manera que sea orientativo para cada tipo de usuarios.

En los últimos años, la rápida expansión de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones. En particular esto afecta a la necesidad de que:

La información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior.

Esta información sea compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa (o a aquella parte que les corresponda según su función) en cada momento.

Estas necesidades han provocado un movimiento creciente de cambio de las aplicaciones tradicionales de escritorio hacia las aplicaciones web, que por su idiosincrasia, cumplen a la perfección con las necesidades mencionadas anteriormente. Por tanto, los sitios web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces

de una interacción más o menos sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende, la necesidad de buscar opciones de diseño nuevas que permitan dar con la arquitectura óptima que facilite la construcción de los mismos.

Para utilizar una aplicación web desde una máquina concreta, basta con tener instalado un navegador web en esa máquina, ya sea éste el Internet Explorer de Microsoft, el Mozilla Firefox o cualquier otro navegador. Desde la máquina cliente donde se ejecuta el navegador, se accede a través de la red al servidor web donde está alojada la aplicación y, de esa forma, se puede utilizar la aplicación sin que el usuario tenga que instalarla previamente en su máquina.

2.2.3 Ciclo De Vida Clásico.

El ciclo de vida clásico consta de varias fases secuenciales siguiendo un esquema en cascada con el mismo orden que el esquema general.

Figura 1: Ingeniería Del Software

Fuente: <http://es.slideshare.net/danielmerchan2/13-ingenieria-del-software-16397494>



Los ciclos de vida del software normalmente varían en el alcance del ciclo, dependiendo de hasta donde llegue el proyecto correspondiente, las características de las fases en que se divide el ciclo y de la organización, estructura y sucesión de las etapas, si hay realimentación entre ellas o si tenemos libertad de repetir las.

Otros modelos son

1. Modelos en V.
2. Iterativo.
3. De desarrollo incremental.
4. En espiral.
5. De prototipos

2.2.3.1 Metodología Scrum

Para este proyecto definiremos los conceptos que abarcan todo el diseño del software para satisfacer las necesidades específicas de Infraestructura tecnológica de la Universidad de Pamplona. Partiendo de implementar una metodología mostrada en la figura 1.

Figura 2. Metodología Scrum.

Fuente <http://www.sstecnologia.com.br/>



Scrum es una metodología ágil de desarrollo de proyectos, que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80, fue aplicado por primera vez por Ken Schwaber y Jeff Sutherland, quienes lo documentaron en detalle en el libro *Agile Software Development with Scrum*.

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados Sistemas de software.(RICARDO & XAVIER, 2012).

Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos y se estructura en ciclos de trabajo llamados Sprints. Éstos son iteraciones de 1 a 4 semanas, y se suceden una detrás de otra. Al comienzo de cada Sprint, el equipo multi-funcional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. Al final del Sprint, el equipo lo revisa con los interesados en el proyecto, y les enseña lo que han construido.

El equipo obtiene comentarios y observaciones que se puede incorporar al siguiente Sprint. Scrum pone el énfasis en *Scientia et Technica* Año XIX, Vol. 19, No. 4, Diciembre de 2014. Universidad Tecnológica de Pereira. 415 productos que funcionen al final del Sprint, es decir que estén realizados. (ALFONZO, 2014).

2.2.3.1.1 Características De Metodología Scrum.

1. Enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos prácticos de desarrollo, complementación y demás cuestiones técnicas
2. Hace uso de Equipos auto-dirigidos y auto-organizados
3. Puede ser aplicado teóricamente a cualquier contexto en donde un grupo de gente necesita trabajar junta para lograr una meta común.
4. Desarrollo de software iterativo incremental basado en prácticas ágiles.
5. Iteraciones de treinta días; aunque se pueden realizar con más frecuencia, estas iteraciones, conocidas como Sprint.
6. Dentro de cada Sprint se denomina al Scrum Master al Líder de Proyecto quien llevará a cabo la gestión de la iteración.

Se convocan diariamente un “Scrum Daily Meeting” el cual representa una reunión de avance diaria de no más de 15 minutos con el propósito de tener re alimentación sobre las tareas de los recursos y los obstáculos que se presentan. En la cual se responden preguntas como: ¿Qué has hecho desde el último encuentro? ¿Qué obstáculos hay para cumplir la meta? ¿Qué harás antes del próximo encuentro?

2.2.3.1.2 Scrum emplea las siguientes prácticas de la gestión ágil.

1. **Revisión de las Iteraciones.** Al finalizar cada iteración se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Este es el periodo máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias del producto.
2. **Desarrollo incremental.** Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

3. **Desarrollo evolutivo.** Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos. El desarrollo Scrum va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto.
4. **Auto-organización.** Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles. La gestión predictiva confía la responsabilidad de su resolución al gestor de proyectos.
5. **Colaboración.** Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la auto organización como un control eficaz cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.
6. **Visión general del proceso.** Scrum denomina “sprint” a cada iteración de desarrollo y recomienda realizarlas con duraciones de 30 días. El sprint es por tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

2.2.4 Ingeniería De Requerimiento

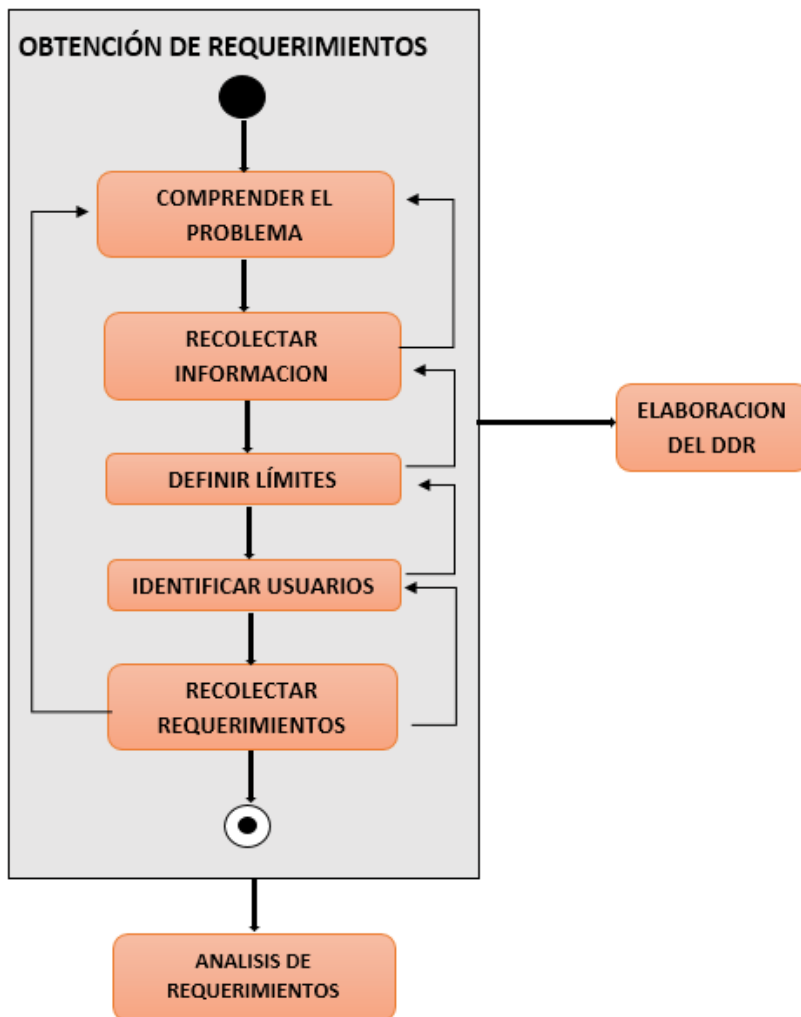
En esta fase se definen y especifican los requerimientos de los clientes y usuarios. Las actividades involucradas en la ingeniería de requerimientos son:

1. Obtención de los requerimientos.
2. Análisis de requerimientos.
3. Especificación de requerimientos.
4. Validación del documento de especificación.
5. Administración de requerimientos.

2.2.4.1 Obtención De Requerimientos. Es indispensable la participación de los usuarios y clientes para la identificación de los requerimientos del sistema. En esta actividad se determina el

dominio de la aplicación, se especifican los servicios que deben proveer el sistema, la funcionalidad requerida del sistema y las restricciones de hardware y software.

Figura 3. Obtención De Requerimientos
Fuente: propia resultado de la investigación.



2.2.4.2 Análisis Y Especificación De Requerimientos. Los requerimientos establecidos en el documento de definición son analizados detalladamente por el equipo de desarrollo y negociados con el cliente y usuarios, para decidir los requerimientos que serán aceptados y definirlos de manera conjunta con el fin de homogenizar su interpretación.

Figura 4. Análisis Y Especificación De Requerimientos

Fuente: propia resultado de la investigación.



2.2.4.3 Requerimientos. Los requerimientos de software se definen como la descripción de las necesidades de los clientes, los servicios que el usuario desea que proporcione el sistema y las restricciones sobre las que el software debe operar.

Estos requerimientos se concentran en el cliente y el problema a resolver en que el resultado del proceso de requerimientos es la base para el diseño, la implementación y la evaluación del software. De esta forma, si no se descubren y especifican todos los requerimientos del sistema en desarrollo, podría conducirnos a la entrega de un producto con alto índice de riesgos y poco funcional.

2.2.4.3.1 Tipos de requerimientos.

2.2.4.3.1.1 Tipos de requerimientos de acuerdo con la audiencia.

Los requerimientos del usuario. Se expresan en lenguaje natural utilizando diagramas fáciles de comprender de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales el sistema debe funcionar; también se conocen como requerimientos del cliente o requerimientos C.

Los requerimientos del sistema. En estos requerimientos se establecen con más detalle los servicios y restricciones del sistema. También se les conoce como especificación funcional, requerimientos del desarrollador o requerimientos D.

La especificación del diseño del software. Es una descripción abstracta del diseño del software que se utiliza como una base para un diseño e implementación detallados

2.2.4.3.1.2 Tipos De Requerimientos De Acuerdo Con Su Característica

Requerimientos funcionales. Son declaraciones de los servicios que proveerá el sistema y comprenden la interacción entre el sistema y su ambiente, la reacción a entradas particulares de datos y su comportamiento en condiciones específicas. En algunos casos también declaran lo que el sistema no debe hacer.

Requerimientos no funcionales. Son especificaciones de las restricciones de los servicios o funciones ofrecidas por el sistema; estas restricciones limitan las elecciones en la solución a un problema. Los requerimientos no funcionales no se refieren directamente a las funciones específicas que entrega el sistema, sino a sus propiedades emergentes.

2.2.4.4 Análisis Y Diseño De Requerimientos. En el proceso de análisis de requerimientos es donde se realiza el levantamiento de la información y su primer paso es involucrar al desarrollador en las necesidades del usuario final.

En este proceso, se debe tener cuidado en manejar requerimientos erróneos, esto causaría problemas muy grande, pues constituyen los elementos de entrada para el proceso de diseño y desarrollo del aplicativo. El objetivo principal no es el levantar el requerimiento, sino que el equipo de desarrollo se familiarice con el problema y definan las mejores prácticas para la creación del programa.

En el análisis se llevan a cabo las siguientes actividades:

1. Priorizar los requerimientos: pueden existir requerimientos más importantes que otros para los clientes y usuarios, por lo que deben clasificarse e implementarse de acuerdo con su prioridad en el sistema.

2. Encontrar dependencias entre requerimientos y etiquetar los requerimientos: con un identificador único, con el fin de poderlos identificar.

3. Resolver conflictos entre los requerimientos: se pueden encontrar conflictos entre requerimientos mediante la revisión de los criterios de calidad que debe cumplir cada requerimiento del sistema.

4. Negociar con flexibilidad: con los demás elementos del equipo que intervienen en el proceso de desarrollo de software, para homogenizar su comprensión, y que de esta forma tanto desarrolladores como usuarios tengan la misma interpretación al momento de leer el documento de requerimientos

2.2.5 Diseño.

El Diseño pone en énfasis en una solución conceptual que satisface los requisitos, en vez de ponerlo en la implementación, por ejemplo los diseños pueden ser implementados como el análisis es más apropiado calificar el término como diseño de objeto o diseño de base de datos.

En esta fase se agregan nuevas clases que proveen de la infraestructura técnica, las cuales son: interfaces de usuario, manejo de bases de datos para almacenar objetos, comunicaciones con otros sistemas. Las clases de dominio del problema del análisis son agregadas en esta fase. El diseño resulta en especificaciones detalladas para la fase de programación. El análisis y el diseño se han resumido en una frase hacer lo correcto.

2.2.5.1 Lenguaje unificado de modelado (uml). El UML (Lenguaje Unificado de Modelado) es una de las herramientas más emocionantes del mundo actual del desarrollo de sistemas, esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas. Es el resultado del trabajo hecho por Grady Booch, James Rumbaugh e Ivar Jacobson. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a los cuales se les conoce como modelo.

En UML, un sistema viene representado por cinco vistas diferentes que lo describen desde diferentes perspectivas. Cada vista se representa mediante un conjunto de diagramas. En UML están presentes las siguientes vistas. (David Sánchez López, 2010).

1. Vista del usuario: Representa el sistema (producto) desde la perspectiva de los usuarios (llamados actores en UML). Vista estructural: modela la estructura estática (clases, objetos y relaciones).

2. Vista del comportamiento: esta parte del modelo del análisis representa los aspectos dinámicos o de comportamiento del sistema.

3. Vista de implementación: los aspectos estructurales y de comportamiento se representan aquí tal y como van a ser implementados.

4. Vista del entorno: aspectos estructurales y de comportamiento en el que el sistema a implementar se representa.

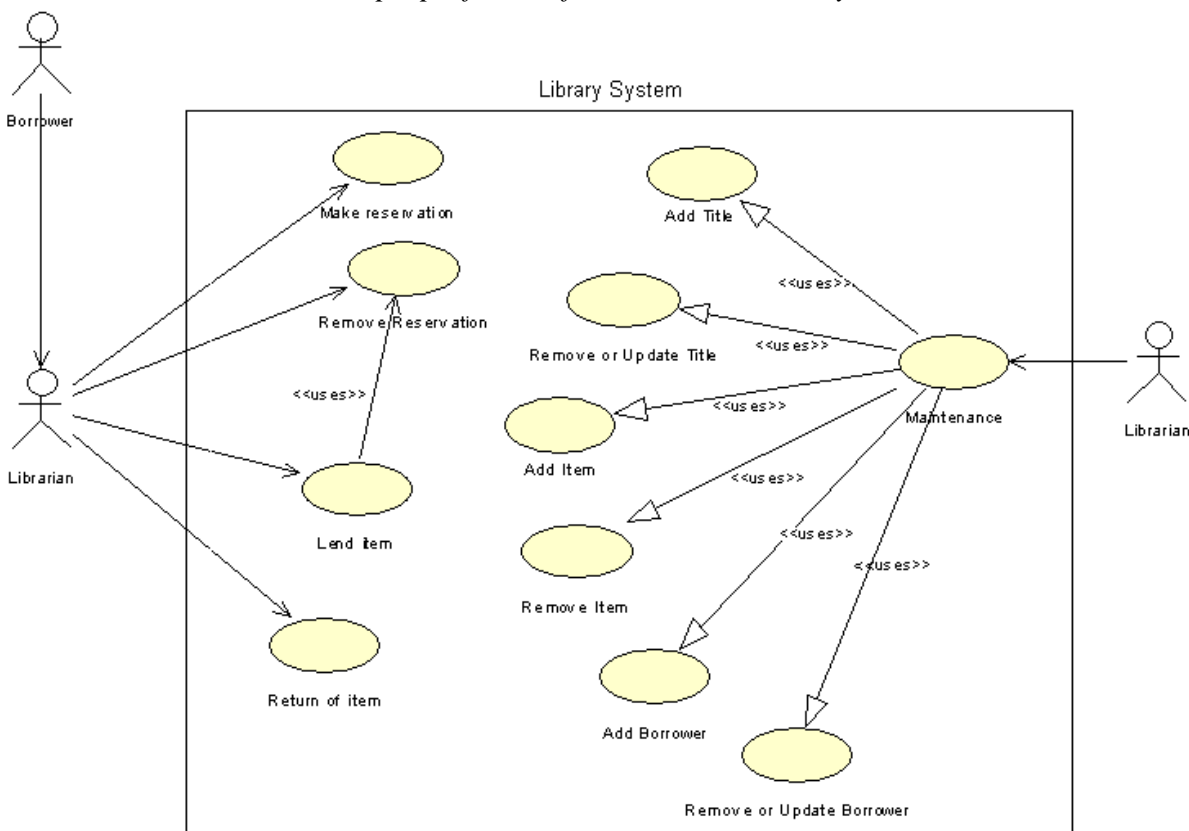
Diagramas UML. Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar.

Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. UML incluye los siguientes diagramas: diagrama de casos de uso, diagrama de clases, diagrama de objetos, diagrama de secuencia, diagrama de colaboración, diagrama de estados, diagrama de actividades, diagrama de componentes y el diagrama de despliegue. A continuación se explican el diagrama de casos de usos y el diagrama de clases:

2.2.5.1.1 Diagrama de casos de usos: El diagrama de casos de usos representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales.

Figura 5: Diagrama de casos

Fuente: <http://profesores.fib.unam.mx/carlos/aydoo/usecase.html>



Los elementos de un diagrama de casos de uso son:

1. Sistema: Un sistema en un diagrama de caso de uso es descrito como una caja; el nombre del sistema aparece arriba o dentro de la caja. Ésta también contiene los símbolos para los casos de uso del sistema.

2. Actores: Un actor es alguien o algo que interactúa con el sistema; es quien utiliza el sistema. Por la frase "interactúa con el sistema" se debe entender que el actor envía a o recibe del sistema unos mensajes o intercambia información con el sistema.

3. Casos de uso: Un caso de uso representa la funcionalidad completa tal y como la percibe un actor. Un caso de uso en UML es definido como un conjunto de secuencias

de acciones que un sistema ejecuta y que permite un resultado observable de valores para un actor en particular.

2.2.5.1.2 Diagramas de Clases. Es una descripción de las clases en un sistema y sus relaciones. No describe el comportamiento dinámico del sistema, por ejemplo el comportamiento de objetos individuales. Éste es el diagrama más común a la hora de describir el diseño de los sistemas orientados a objetos. Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

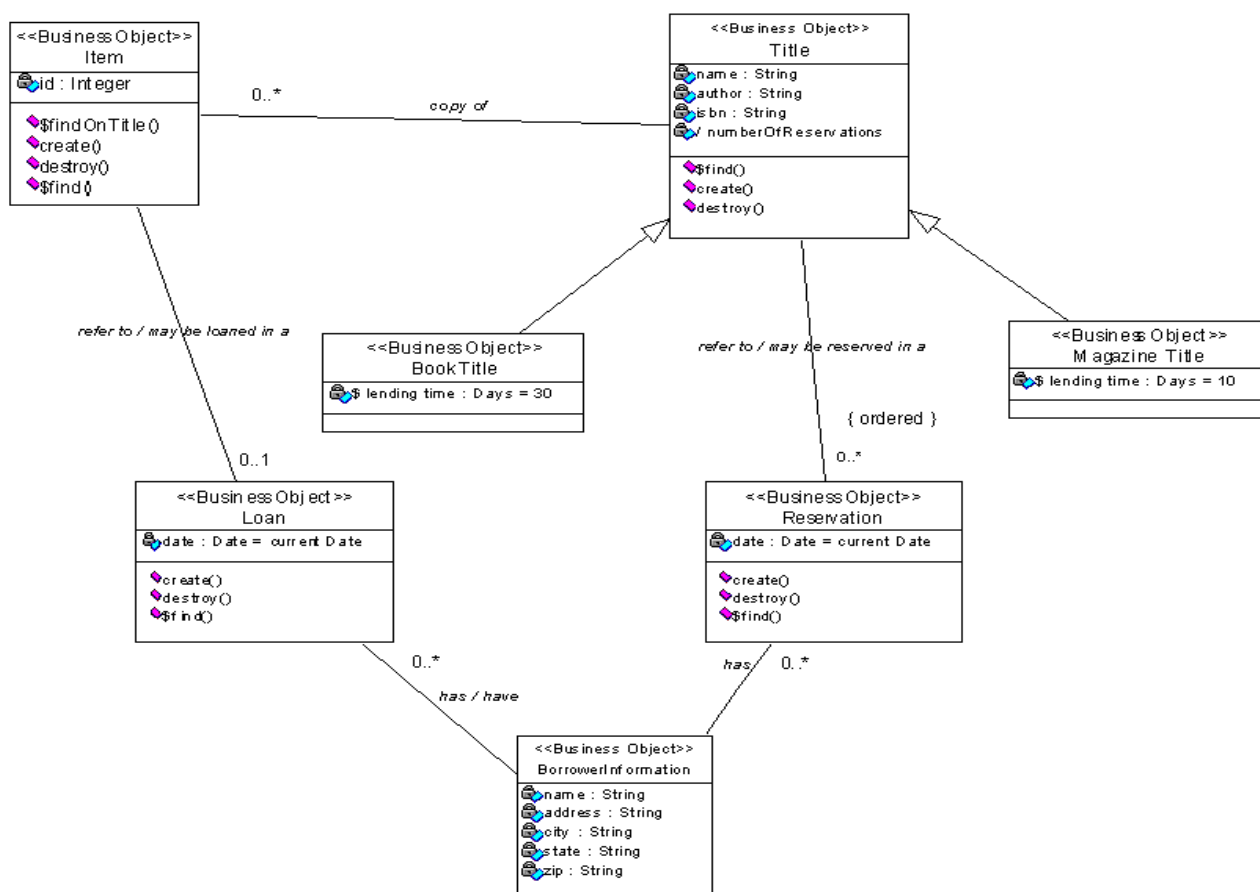
Para poder definir las clases, UML sugiere seis características selectivas que debe utilizar el analista para considerar una clase candidato en el modelo de análisis:

1. **Información retenida.** La clase será útil durante el análisis sólo si la información sobre el mismo ha de ser almacenada, transformada, analizada o manejada en algún otro modo. La información puede referirse a conceptos que deberán estar siempre registrados en el sistema, eventos o transacciones que ocurren en un momento específico.
2. **Sistema externo.** Si se tiene un sistema externo a este sistema, entonces es de interés en la etapa de modelado. Los sistemas externos deberán ser vistos como clases que el sistema contendrá.
3. **Patrones, librerías de clases o componentes.** Si se tienen patrones, librerías de clases o componentes, generalmente éstos son clases candidatos.
4. **Dispositivos que el sistema maneja.** Dispositivos técnicos que maneja el sistema se convertirán en clases que manejarán esos dispositivos.

5. **Partes organizacionales.** Especialmente en modelos de negocio, todas las partes que representan a la organización, serán clases candidatos.

6. **Roles de actores.** Los roles de actores serán vistos como clases, por ejemplo, usuario, operador del sistema, administrador, cliente, etc.[21]

Figura 6: Diagrama de Clases
Fuente propia de la investigación.



2.2.5.1.3 Diagrama de actividad: Un diagrama de actividades muestra el flujo de actividades, siendo un actividad una ejecución general entre los objetos que se está ejecutando en un momento dado dentro de una máquina de estados, el resultado de un actividad es una acción que producen un cambio en el estado del sistema o la devolución de un valor. Las acciones

incluyen llamadas a otras operaciones, envío de señales, creación o destrucción de objetos o simples cálculos.

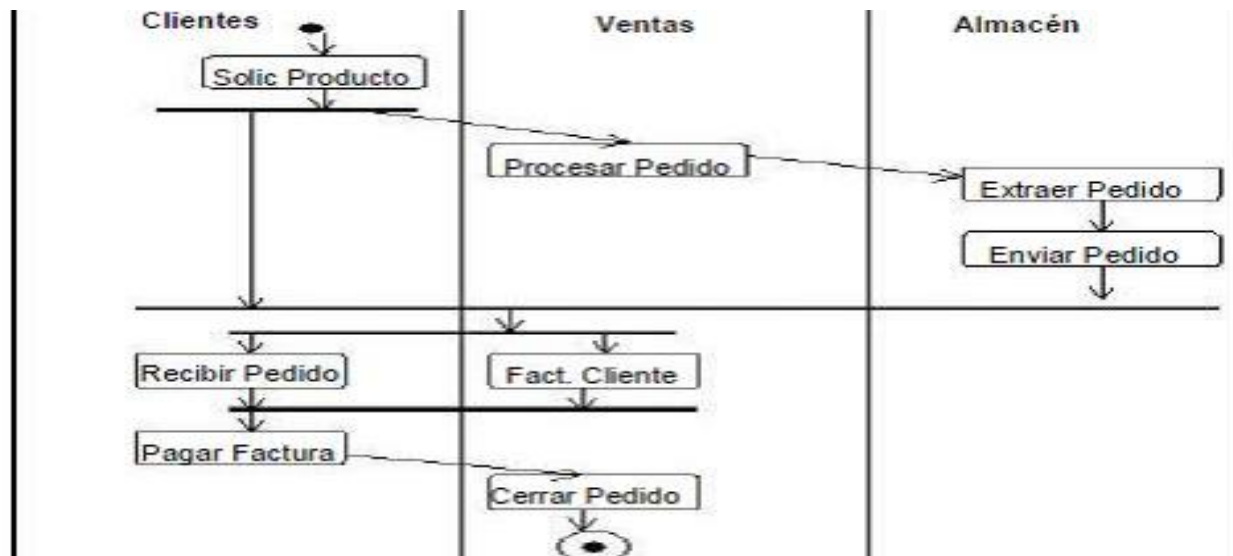
Estados de actividad y estados de acción: a representación de ambos es un rectángulo con las puntas redondeadas, en cuyo interior se representa bien una actividad o bien una acción. La forma de expresar tanto una actividad como una acción, no queda impuesta por UML, se podría utilizar lenguaje natural, una especificación formal de expresiones, un metalenguaje, etc.

Transiciones: Las transiciones reflejan el paso de un estado a otro, bien sea de actividad o de acción. Esta transición se produce como resultado de la finalización del estado del que parte el arco dirigido que marca la transición. Como todo flujo de control debe empezar y terminar en algún momento, podemos indicar esto utilizando dos disparadores de inicio y fin.

Bifurcaciones: Un flujo de control no tiene porqué ser siempre secuencial, puede presentar caminos alternativos. Para poder representar dichos caminos alternativos o bifurcación se utilizará como símbolo el rombo. Dicha bifurcación tendrá una transición de entrada y dos o más de salida.

División y unión: UML representa gráficamente el proceso de división, que representa la concurrencia, y el momento de la unión de nuevo al flujo de control secuencial, por una línea horizontal ancha.

Calles: Cuando se modelan flujos de trabajo de organizaciones, es especialmente útil dividir los estados de actividades en grupos, cada grupo tiene un nombre concreto y se denominan calles.

Figura 7: Diagrama de actividades*Fuente propia de la investigación.*

2.2.5.1.4 Diagrama de secuencia: Modelos la lógica secuencial, en efecto el ordenamiento tiempo de mensajes entre los clasificadores. Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

Línea de Vida: Una línea de vida representa un participante individual en un diagrama de secuencia. Una línea de vida usualmente tiene un rectángulo que contiene el nombre del objeto.

Mensajes: Los mensajes se muestran como flechas. Los mensajes pueden ser completos, perdidos o encontrados; síncronos o asíncronos: llamadas o señales.

Ocurrencia de ejecución: Un rectángulo fino a lo largo de la línea de vida denota la ocurrencia de ejecución o activación de un foco de control. En el diagrama anterior hay tres ocurrencias de ejecución.

Inicio y final de línea de vida: Una línea de vida se puede crear o destruir durante la escala de tiempo representada por un diagrama de secuencia.

Restricciones de tiempo y duración: En forma predeterminada, un mensaje se muestra como una línea horizontal.

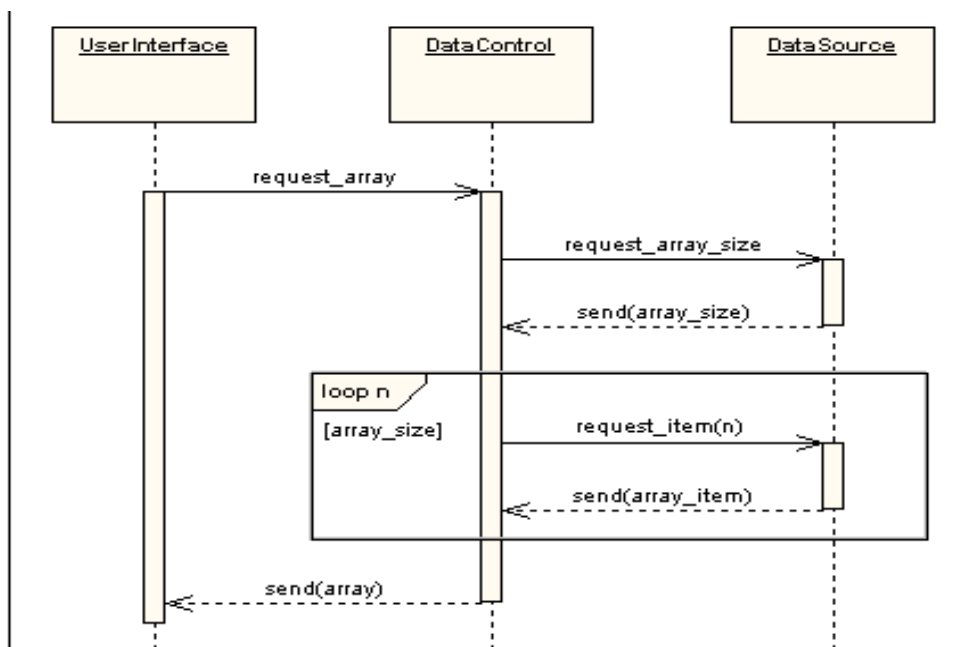
Fragmentos combinados: Se estableció anteriormente que no se espera que los diagramas de secuencia muestren lógicas de procedimientos complejos.

Puerta: Una puerta es un punto de conexión para conectar un mensaje dentro de un fragmento con un mensaje fuera del fragmento.

Descomposición en parte: Un objeto puede tener más de una línea de vida que viene de ésta. Esto permite mensajes de entre e intra objetos para que se muestren en el mismo Diagrama.

Figura 8: Diagrama de Secuencia

Fuente propia de la investigación.



2.2.5.2 Modelo entidad y relación

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chan en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado modelo entidad-relación extendido. (Andrea, J., & Palencia, R. (2016).)

La entidad es un tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso.

La relación es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función [34].

2.2.6 Herramientas De Diseño

Visual Paradigm for UML Visual Paradigm for UML (Visual Paradigm, 2010) es una herramienta CASE con varias opciones para modelado con diagramas UML2 y también es compatible con los requisitos de SysML y diagramas ER diagramas. La herramienta tiene un buen ambiente de trabajo, lo que facilita la visualización y manipulación del proyecto de modelado. Es una herramienta de trabajo y también es compatible específica cambios en el código fuente de algunos lenguajes de programación como C ++ y Java.

Soporta los siguientes tipos de diagrama

1. Diagrama de clases
2. Diagrama de objetos

3. Diagrama de secuencias
4. Diagrama de actividad
5. Diagrama de uso

2.2.7 Construcción

Es la creación del software operativo por medio de la codificación, la verificación y pruebas.

Los límites de este diseño y de la construcción varían dependiendo del proceso del ciclo de vida del software utilizado en el proyecto.

2.2.7.1 Base de datos

Es una colección de datos referentes a una organización estructurada según un modelo de datos de forma que refleja las relaciones y restricciones existentes entre los objetos del mundo real, y consigue independencia, integridad y seguridad de los datos. Lo que debemos tener claro es la diferencia entre Base de Datos y SGBD. La base de datos es el almacenamiento donde residen los datos. El SGBD es el encargado de manipular la información contenida en ese almacenamiento mediante operaciones de lectura/escritura sobre la misma. Además las bases de datos no sólo contendrán las tablas (ficheros) de datos, sino que también almacenará formularios (interfaces para edición de datos), consultas sobre los datos, e informes. El SGBD se encargará de manipular esos datos, controlar la integridad y seguridad de los datos, reconstruir y reestructurar la base de datos cuando sea necesario.

Entre las principales características de los sistemas de base de datos podemos mencionar:

1. Independencia lógica y física de los datos.

2. Redundancia mínima.
3. Acceso concurrente por parte de múltiples usuarios.
4. Integridad de los datos.
5. Consultas complejas optimizadas.
6. Seguridad de acceso y auditoría.
7. Respaldo y recuperación.
8. Acceso a través de lenguajes de programación estándar.

2.2.7.2 Patrón De Arquitectura

2.2.7.2.1 Arquitectura de la aplicación

Diseño. El diseño es una representación significativa de ingeniería de algo que se va a construir. Se puede hacer el seguimiento basándose en los requisitos del cliente, y al mismo tiempo la calidad se puede evaluar y cotejar con el conjunto de criterios predefinidos para obtener un diseño bueno en el contexto de la ingeniería del software, el diseño se centra en cuatro áreas importantes de interés: datos, arquitectura, interfaces y componentes.

2.2.7.2.1.1 Arquitectura de la aplicación. La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

A continuación describiremos el tipo de Arquitectura empleado para el desarrollo de esta Aplicación Web.

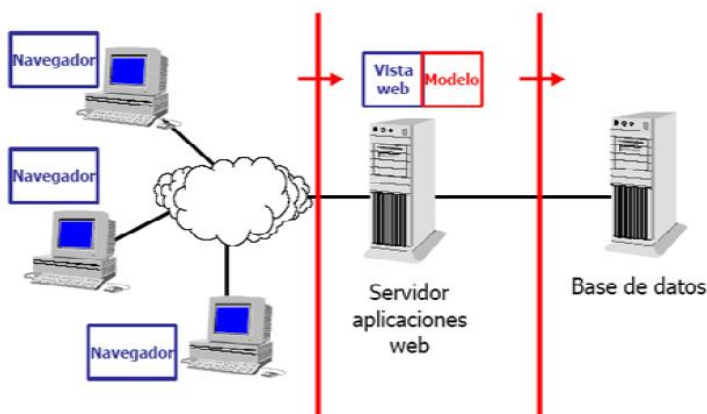
Arquitectura de 3 niveles. Lo que se conoce como arquitectura en capas es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del

desarrollo, tales como las cuestiones de presentación, lógica de negocio, mecanismos de almacenamiento, etc.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. (MONSALVE MAYER, & ORTEGA RAFAEL, 2012)

Figura 9: Modelo De Tres Capas

Fuente: SÁNCHEZ GONZÁLEZ, Carlos. ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras. Coruña: Universidad de Coruña, 2004.



En el presente proyecto la primera capa se encuentra conformada por los diferentes navegadores web desde los cuales el usuario final accede a la aplicación, la segunda capa la compone el servidor de aplicaciones J-BOSS 7.1.1GA el cual se encarga de almacenar la lógica del negocio, por último en la 3 capase ubica el motor de base de datos PostgreSQL, el cual se encarga del modelo de datos de la aplicación.

Todo modelo de arquitectura del software va muy ligado a un patrón de diseño específico para facilitar la construcción del mismo y el cual se encuentra ubicado en segunda capa de la

arquitectura de diseño es decir en el servidor de aplicaciones. Para el desarrollo de nuestra aplicación tomamos como base el patrón de diseño Modelo-Vista –Controlador.

2.2.7.2.2 Modelo-Vista-Control (MVC). Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

1. El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
2. La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
3. El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La figura 13 muestra el Esquema del patrón MVC, en la misma se puede observar la subdivisión que se realiza de cada uno de los niveles. El Modelo está compuesto por Abstracción de Base de Datos y Acceso a los Datos, La Vista está compuesta por el Layout, la lógica de la vista y el Template, El Controlador está compuesto por el controlados Frontal y las Acciones.

Figura 10. Esquema del Patrón MVC

Fuente propia de la investigación.



2.2.7.3 Interfaz gráfica del usuario

Es un método para facilitar la interacción del usuario con el ordenador a través de un conjunto de imágenes, objetos pictóricos y texto. Surge como evolución de la línea de comandos de los primeros sistemas operativos.

La interfaz gráfica está diseñada para brindar al usuario final la manera ágil de encontrar y recordar el uso de las opciones que más le interesan, aprendiendo de forma rápida y permitiendo concentrarse en los análisis de la información y no en el manejo de la aplicación.

(Andrea & Palencia, 2016)

Componentes de la interfaz gráfica son:

1. Ventana de trabajo y despliegue de mapas
2. Panel de control
3. Menús desplegables
4. Administrador de componentes
5. Ventanas de información emergentes
6. Barra de control de nivel de escala del mapa
7. Barra de estado de servicios de mapas
8. Ventana de herramientas organizada por componentes

2.2.7.4 Herramientas para la construcción

Las herramientas de desarrollo de software que se utilizaran en este proyecto serán, lenguaje de programación java, Frameworks JBoss Seam, bases de datos postgresql, entornos de desarrollo netbeans, servidor Jboss.

2.2.7.4.1 Lenguaje de programacion

2.2.7.4.1.1 Java.

El lenguaje para la programación en Java, es un lenguaje orientado a objeto, de una plataforma independiente. Maneja un estándar global en el desarrollo de aplicaciones tanto de escritorio como aplicaciones web. Este lenguaje brinda las APIs necesarias para desarrollar un programa de computadora que permite interactuar tanto con el usuario, como con la lógica del proceso que se quiere modelar.

Maneja un estándar global en el desarrollo de aplicaciones tanto de escritorio como aplicaciones web. Este lenguaje brinda las APIs necesarias para desarrollar un programa de computadora que permite interactuar tanto con el usuario, como con la lógica del proceso que se quiere modelar.

La principal característica de Java es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código que se genera bytecodes es interpretado por una máquina virtual.

Finalmente, Java posee un gestor de seguridad con el que poder restringir el acceso a los recursos del sistema.

2.2.7.4.1.2 Html5

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas. Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C Haider miranda (2016).

Frameworks

“Los frameworks o marcos de trabajo, son esquemas (un esqueleto, un patrón) para el desarrollo y/o implementación de una aplicación. Definen, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Los frameworks incluyen programas de apoyo, compiladores, bibliotecas de código, interfaz de programación de aplicaciones (API) y un conjunto de herramientas que reúnen a todos los diferentes componentes para permitir el desarrollo de un proyecto o solución.”

2.2.7.4.2 Entorno de desarrollo

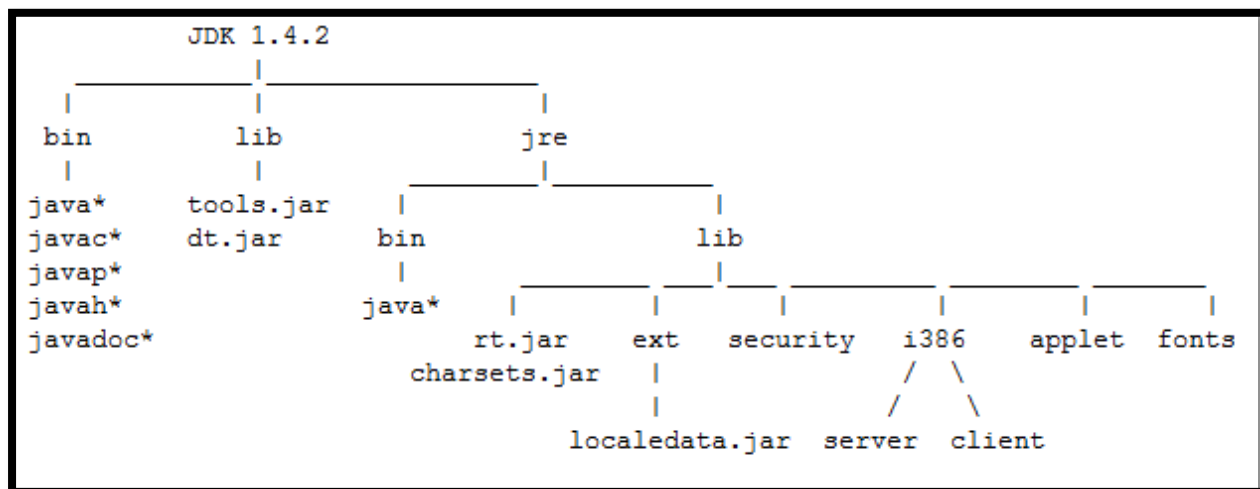
2.2.7.4.2.1 Netbeans. Es una herramienta de entorno integrado libre, muy popular en la comunidad de desarrolladores Java. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas, Ofrece herramientas muy útiles en la edición de código que permite la identificación y depuración de errores, sintaxis resaltada, autocompletado de código, análisis de tipeo, asistente para la configuración de archivos XML, verificación de sintaxis, un compilador, un intérprete, navegador de clases, entre otros.. Está escrito en Java. Pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

2.2.7.4.2.2 JDK. La JDK es la Plataforma de Desarrollo JAVA. Se trata de la suma de todo lo que contiene la JRE (Java Runtime Environment) más una serie de herramientas de desarrollo como un compilador, debugger, compactador, documentador, etc. necesarios para desarrollar applets y aplicaciones de escritorio. En la JDK también se encuentran herramientas de seguridad y de despliegue de aplicaciones.(RICARDO & XAVIER, 2012). La estructura de directorios del JDK/J2SE es la siguiente:

Figura 11: La estructura de directorios del JDK/J2SE

Fuente: <http://javabasico.osmosislatina.com/curso/objetos.htm>



2.2.7.4.2.3 Tomcat. Apache Tomcat (también llamado JakartaTomcat o simplemente Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServerPages (JSP) de Sun Microsystems.

El criterio utilizado para la elección de este contenedor de aplicaciones se ha basado en dos motivos fundamentales:

Tomcat es la implementación oficial de referencia propuesta por Sun para la especificación de contenedor de aplicaciones. La versión 4.1.24 utilizada para el desarrollo del proyecto soporta la especificación 2.3 de Servlets y 1.2 de JSP's. Al utilizar Tomcat sabemos que nuestro código es completamente estándar y debería ser portable a cualquier contenedor de aplicaciones que cumpliera con los estándares J2EE.

Como el resto de la tecnología utilizada en <e-Aula> Tomcat es software de libre distribución.

2.2.7.4.3 Sistema de Gestión de Base de Datos.

Se define un sistema de gestión de base de datos (SGBD) como un conjunto coordinado de programas, procedimientos, lenguajes, entre otros. Que suministra, a los usuarios dependiendo de su rol, los medios necesarios para recuperar, manipular y describir los datos almacenados en la base de datos, manteniendo su integridad, seguridad y confidencialidad.

2.2.7.4.3.1 PostgreSQL. El Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una serie de funciones que nos permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos.

PostgreSQL está ampliamente considerado como el sistema de bases de datos (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980, de código abierto más avanzado del mundo y soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas.

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad como: (A) Linux. (B) UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64). (C) Windows.

Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos del SQL 2008, incluyendo INTEGER, numérico, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C / C + +, Java, Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, y la documentación que actualmente existe es realmente excepcional.

PostgreSQL ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema.

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de la información de la Base de Datos.

Características aportan potencia y flexibilidad adicional: (A) Restricciones. (B) Disparadores. (C) Reglas. (D) Integridad transaccional.

2.2.7.4.4 Frameworks.

El concepto de framework (cuya traducción aproximada sería “marco de trabajo”) es un esquema (un esqueleto o patrón) para el desarrollo y/o la implementación de una aplicación, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

El uso de un framework permite acelerar el proceso de desarrollo, la capacidad de reutilización del código así se obtiene mayor productividad y un tiempo de mercado breve en el desarrollo de aplicaciones, además promueve buenas prácticas de desarrollo como el uso de patrones.

Para el desarrollo de este proyecto se utilizó el Framework SEAM, el cual se describe a continuación:

2.2.7.4.4.1 JBoss Seam. JBoss Seam es una plataforma de integración de tecnologías y desarrollo para la construcción de aplicaciones de generación Web 2.0, unificando e integrando tecnologías tales como Ajax, JSF, EJB3, Java Portlets y Gestión del Proceso de Negocio (BPM), que tiene como objetivo facilitar el desarrollo de aplicaciones JEE. Es un proyecto Open Source con una comunidad abierta y aunque cuenta con el respaldo de JBoss (que es una división de Red Hat), su funcionalidad no está ligada al servidor de aplicaciones de este. El desarrollo de Seam, utilizando JSF y EJB que son las tecnologías con más soporte, es muy ágil ya que reduce el nivel de configuración necesario para la integración y aprovecha al máximo las ventajas de cada una de las tecnologías haciendo al proyecto más estable, legible, predecible y mantenible.

Seam ha sido diseñado desde cero para eliminar la complejidad en los niveles de arquitectura y API. Le permite a los desarrolladores ensamblar aplicaciones Web complejas usando Plain Old Java Objects (POJOs), clases Java compuestas simplemente por atributos privados y métodos públicos GET y SET; y mediante el uso de anotaciones, widgets de componentes UI y muy poco XML. Tiene soporte para un nuevo contexto de variables conocido como Conversación y la posibilidad de gestionar estados (states) en la aplicación declarándolos en la programación, sobre esto se profundizará en el apartado siguiente. Seam presenta una experiencia de usuario más sofisticada, y al mismo tiempo elimina errores comunes encontrados en aplicaciones Web tradicionales. (C. E. El, 2011)

Se tiene como su característica más importante a la habilidad de integrar tecnologías de diferentes ámbitos a través de la inyección de dependencias. A eso se le agrega el tema de los contextos, tenemos más contextos lo que no quiere decir que ocupemos más memoria, si no que todo lo contrario vamos a tener más opciones para elegir donde guardar nuestras instancias y éstas van a permanecer en memoria el tiempo indicado.

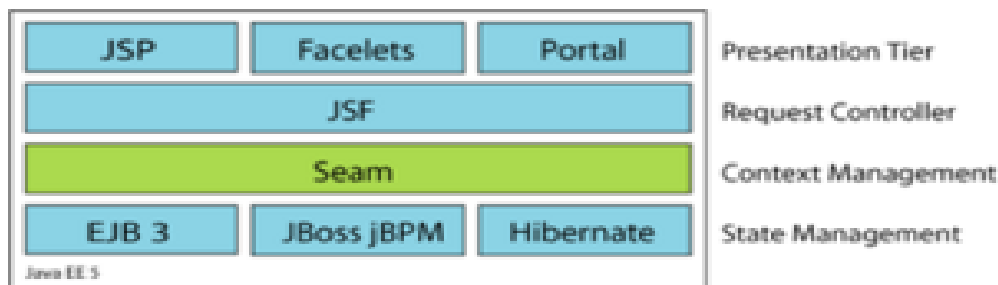
Los objetivos principales que persigue un framework son: (A) Acelerar el proceso de desarrollo. (B) Reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Arquitectura de Aplicaciones Web con Seam. El aspecto más relevante de Seam es la forma en la que integra el uso de varias tecnologías ya existentes para la creación de aplicaciones web en Java para facilitar la implementación del patrón MVC de una forma que resulta más intuitiva para desarrollador y más rápida de programar, pero sin perder la potencia y las características que provee JEE.

A través de la figura 3 se muestra la arquitectura de una aplicación.

Figura 12: Arquitectura de una aplicación Seam

Fuente propia de la investigación.



Seam aprovecha las capacidades de extensión de JSF y utiliza EJB3 para producir la integración de la siguiente forma:

El desarrollador puede crear componentes del modelo (los componentes Seam) que van a vivir en el servidor como componentes Seam, pero que van a poder ser accedidos directamente desde la capa de presentación, como si fueran JSF Managed Beans. Seam se encargará de controlar su ciclo de vida, dependiendo del contexto al que pertenezcan.

El método para crear dichos componentes es simplemente crear un POJO con la funcionalidad deseada, o un Enterprise Java Bean, y añadirle las anotaciones java necesarias para que el servidor sepa qué tipo de componente quiere y sus otras características. Los Spring Beans también se pueden convertir en componentes Seam.

Cada componente se enlaza a uno de los contextos disponibles en el framework. En este sentido, el concepto de contexto de Seam es similar al concepto de ámbito de aplicación, pero además, Seam se encargará de crear o destruir los componentes, dependiendo de la necesidad que de éste se tenga, y de cómo haya sido configurado. Existen varios contextos en Seam: (A)

Stateless context.(B) Event (i.e., request) context.(C) Page context.(D) Conversation context.(E) Session contextg.(F) Business process context.(G) Application context.

2.2.7.4.4.1.1 Richfaces. RichFaces: es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

Son características de RichFaces las siguientes: (A) • Se integra perfectamente en el ciclo de vida de JSF. (B) Incluye funcionalidades Ajax, de modo que nunca vemos el JavaScript y tiene un contenedor Ajax propio. (C) Contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas nuestras necesidades. (D) Soporta facelets. (E) Soporta css themes o skins, (F) Es un proyecto open source, activo y con una comunidad también activa.

RichFaces permite definir (por medio de etiquetas de JSF) diferentes partes de una página JSF que se desee actualizar con una solicitud Ajax, proporcionando así varias opciones para enviar peticiones Ajax al servidor.

2.2.7.4.5 Servidor Jboss. JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en Abril del 2006, fundada por Marc Fleury, el creador de la primera versión de JBoss. El proyecto está apoyado por una red mundial de

colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios.
(Sánchez, n.d.)

Las características destacadas de JBoss incluyen: (A) Producto de licencia de código abierto sin coste adicional. (B) Cumple los estándares. (C) Confiable a nivel de empresa. (D) Incrustable, orientado a arquitectura de servicios. (E) Flexibilidad consistente. (F) Servicios del middleware para cualquier objeto de Java. (G) Ayuda profesional 24x7 de la fuente. (H) Soporte completo para JMX.

3. Diseño Y Construcción De La Aplicación Web

En este capítulo, se darán a conocer los requerimientos para desarrollar el aplicativo para gestión de solicitudes del área de infraestructura tecnológica de la Universidad De Pamplona Sede Villa Del Rosario, que se obtuvieron por medio de una entrevista con el administrador del área.

3.1 Requerimientos

Partiendo desde de la necesidad de crear un sistema que ayudara a gestionar y controlar las solicitudes que se realizan al área de infraestructura tecnológica de la Universidad De Pamplona Sede Villa Del Rosario, se plantea la idea de realizar una Aplicación web que ayudara con este proceso controlando las solicitudes.

Para tener una mejor visión del proyecto a desarrollarse, logramos recopilar la suficiente información por medio del administrador del área de infraestructura tecnológica de la Universidad De Pamplona, Sede Villa Del Rosario.

Utilizando técnicas de investigación, realizando un minucioso análisis de las herramientas tecnológicas a utilizar, teniendo reuniones con el administrador del área.

En base a las técnicas mencionadas anteriormente aplicamos lo siguientes aspectos:

1. Relatos de administrador (Entrevistas y Observación Reuniones).
2. Entregas pequeñas muy frecuentes (Reuniones)
3. Reuniones de pie, todos los días.
4. El sistema será implantado en área de infraestructura tecnológica de la Universidad De Pamplona, Sede Villa Del Rosario.
5. El sistema será manejado por el administrador y auxiliar del área de infraestructura tecnológica de la Universidad De Pamplona, Sede Villa Del Rosario.

Tabla 1. Requisitos de Usuario

REQUISITOS DE USUARIO	
Fecha de Solicitud: 16/09/2016	
1. ESPECIFICACIÓN DETALLADA DE REQUISITOS.	
No.	Descripción del requisito
R1	El sistema debe permitir iniciar sesión por medio del código y contraseña del usuario o administrador.
R2	El sistema debe validar el inicio de sesión del usuario o administrador.
R3	El sistema debe permitir la gestión de usuarios (CRUD).
R4	El sistema debe permitir la gestión de dependencia (CRUD).
R5	El sistema debe permitir la gestión de incidencia (CRUD).
R6	El sistema debe permitir el registro de solicitudes.
R7	El sistema debe permitir el registro de seguimientos a las solicitudes por el administrador
R8	El sistema debe permitir el cierre y calificación de las solicitudes por el usuario.

R9	El sistema mostrara las solicitudes abiertas y cerradas del usuario y administrador.
R10	El sistema mostrara gráficamente el índice de satisfacción mensual.
R11	El sistema mostrara gráficamente el total de solicitudes abiertas y cerradas.
R12	El sistema mostrara gráficamente el total de solicitudes por dependencia.
R13	El sistema mostrara gráficamente el total de solicitudes por incidencia.

3.2 Análisis Del Sistema

3.2.1 Descripción De Requisitos

La especificación de requisitos describe el comportamiento esperado de la aplicación una vez desarrollada. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software.

3.2.1.1 Requerimientos funcionales. Un requisito funcional define el comportamiento interno de la aplicación: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica.

Tabla 2. Requerimientos funcionales.

R1 Validar inicio de sesión de usuario por contraseña
R2 Registrar una solicitud
R3
R4
R5 Registra un Acción asociada a una Hallazgo y sus respectiva(s) prueba(s)
R6 Actualizar los datos de una Acción
R7 Eliminar una Acción
R8 Registrar una nueva zona
R9 Modificar datos de una zona
R10 Eliminar una Zona
R11 Registrar Empleado asociado a un Proceso
R12 Modificar Empleado
R13 Eliminar un Empleado
R14 Registrar Proceso
R15 Modificar Proceso
R16 Eliminar Proceso

R17 Registrar Perfil Sanitario
R18 Modificar Perfil Sanitario
R19 Eliminar Perfil Sanitario
R20 Registrar una clase de Perfil Sanitario
R21 Modificar una clase de Perfil Sanitario
R22 Eliminar una clase de Perfil Sanitario
R23 Consultar Hallazgos
R24 Consultar Pruebas
R25 Consultar Acciones
R26 Consultar Zonas
R27 Consultar Procesos
R28 Consultar Perfiles Sanitario
R29 Consultar Clases de un Perfil Sanitario
R30 Consultar Empleados
R31 Registra Prueba
R32 Eliminar Prueba
R33 Modificar Prueba

3.2.1.2 Requerimientos no funcionales. Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.

Tabla 3. Requerimientos no funcionales.

R1: permitir fácil navegación a través del sitio web.
R2: controlar la redundancia de los datos al realizar un registro.
R3: mantener la seguridad y privacidad referente a los datos (información).
R4: proporcionar una interfaz atractiva y fácil de manejar.

R5: minimizar el tiempo de respuesta.
R6: garantizar compatibilidad con navegadores de uso común.
R7: garantizar capacidad de operar en arquitectura hardware de 32 ó 64 bits.
R8: definir una aplicación orientada a la WEB.
R9: utilizar el idioma castellano para los mensajes y textos en la interfaz.
R10: desarrollar manual de usuario para la aplicación.
R11: considerar el crecimiento esperado en el volumen de datos.
R12 Debe permitir el mantenimiento y la escalabilidad.

3.2.1.3 Entregables

1) Aplicación. Incluye los ejecutables para el servicio web.

Tabla 4: Entregables

Id	Sub-Entregable	Descripción	Criterio De Aceptación
App-01	MODULO TECNICO	Este módulo es el encargado de la administración y gestión de los usuarios, incidencias y dependencia.	La ingeniera de prueba Tatiana Medina debe certificar por medio de un acta que las pruebas de validación tengan un resultado satisfactorio y que la funcionalidad es aplicada teniendo en cuenta la historia de usuario
App-02	MODULO MAESTRO	Este módulo es el encargado de listar las incidencias, dependencias y usuarios.	La ingeniera de prueba Tatiana Medina debe certificar por medio de un acta que las pruebas de validación tengan un resultado satisfactorio y que la funcionalidad es aplicada teniendo en cuenta la historia de usuario

App-03	MODULO DE SOLICITUDES	Este módulo es el encargado de la administración y gestión de las solicitudes y los informes.	La ingeniera de prueba Tatiana Medina debe certificar por medio de un acta que las pruebas de validación tengan un resultado satisfactorio y que la funcionalidad es aplicada teniendo en cuenta la historia de usuario
--------	-----------------------	---	---

2). Manuales

Tabla 5: Manuales.

Id	Sub-Entregable	Descripción	Criterio De Aceptación
man-01	Manual de Usuario	Documento en pdf con los pantallazos de la app web donde indica el correcto funcionamiento de la aplicación.	Debe incluir el 100% de las pantallas que tiene la app web.
man-02	Manual de Instalación	Documento en pdf con un instructivo de cómo se debe desplegar la aplicación web (Servicio) en el servidor y sus respectivos datos, Además como instalar la app web como acceso directo en los equipos.	Debe ser paso a paso de tal forma que cualquier persona externa al desarrollo con conocimientos de software y servidores web y de base de datos pueda instalar correctamente el contexto.

3). Documentación

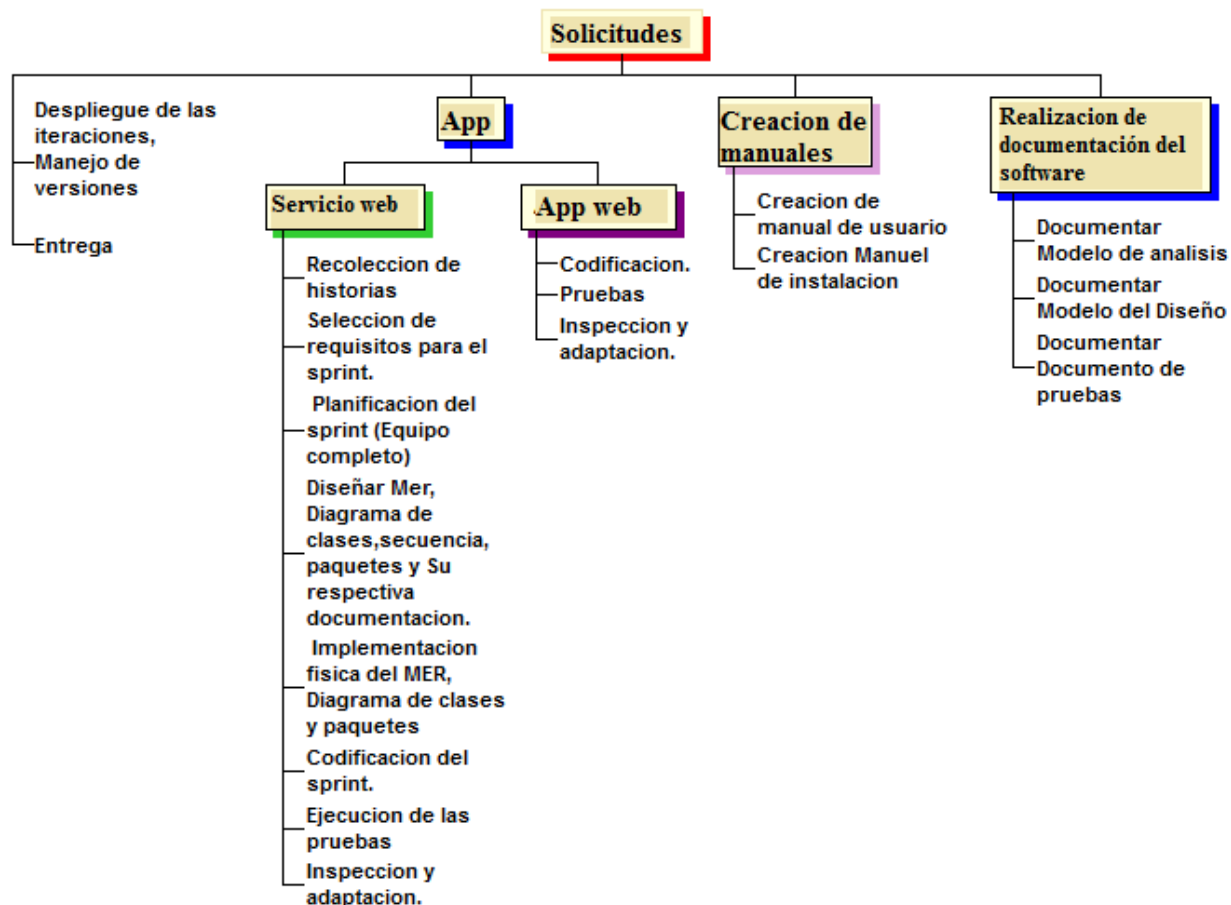
Tabla 6: Documentación

Id	Sub-Entregable	Descripción	Criterio De Aceptación
doc-01	Modelo de Análisis	Comprende los requisitos de usuario y artefactos del producto y metodología generados durante la etapa de análisis, entre estos se encuentran las historias de usuario, diagramas de casos de uso y planificación del sprint,	Los diagramas correspondientes al producto deben ser presentados en UML y las historias de usuario y planificación del sprint de acuerdo a lo planteado por SCRUM.
doc-02	Modelo del Diseño	Comprende los artefactos para el diseño del producto entre estos se encuentra: * Diagramas de secuencia * Diagrama de casos de uso * Diagramas de clases * Diagrama de actividades * Modelo Entidad Relación	Los diagramas correspondientes al diseño del producto se deben implementar en UML para cada sprint.
doc-03	Documento de las pruebas	Documento en pdf plan de pruebas donde se evidencian las pruebas de validación realizadas a los casos de uso de la aplicación	Todos los casos de uso deben ser validados con resultado satisfactorio.

Estructura desglosada de trabajo.

Figura 13: Ilustración 1 Estructura Desglosada de Trabajo – EDT

Fuente propia de la investigación.



3.3 Implementación de metodología Scrum

Para el desarrollo del producto se tiene establecido el uso de algunas de las actividades de la metodología SCRUM debido que el tiempo para la entrega del producto se reduce a un aproximado de ocho semanas, y puesto que es la metodología de desarrollo ágil que más se ajusta a las necesidades del proyecto. Para el producto se establecen las actividades tomadas de acuerdo al tipo de ciclo de vida establecido por la misma metodología de desarrollo como lo es la

iterativa incremental, dichas actividades conforman cada una de las tres etapas de SCRUM de la siguiente manera:

Planificación de la iteración:

1. Recolección de historias.
2. Selección de requisitos para el sprint.
3. Planificación del sprint.

Ejecución de la iteración:

1. Diseñar MER, Diagrama de clases, secuencia, paquetes y su respectiva documentación.
2. Implementación física del MER, diagrama de clases, secuencia, paquetes.
3. Codificación del sprint.
4. Ejecución de las pruebas.

Inspección y adaptación: Después de implementado el producto software se construirán los manuales de usuario y de instalación, seguido de esto se recopilaran los artefactos generados durante el desarrollo establecidos por la metodología para la creación de la documentación del software, por último se realizará la socialización del nuevo sistema web haciendo su respectiva entrega; su implementación que se aclara es parcial (debido a que SCRUM como metodología ágil es más completa y se tomarán de ella sus tópicos más principales en base a las necesidades del proyecto).

4. Desarrollo Del Proyecto

4.1 Fase de Análisis

4.1.1 Documento de análisis

4.1.1.1 Historias de usuario

Historia de Usuario	
Número: 1	Usuario: Usuario Administrativo
Nombre historia: Autenticación del Usuario	
Prioridad en negocio: 1	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Los datos a mostrar del usuario son personales, para acceder a la aplicación se debe autenticar con el usuario y contraseña.	
Validación: El usuario debe iniciar sección con su usuario y contraseña. Ningún usuario puede acceder a los datos del otro.	

Tabla 7: Historia de usuario 1

Tabla 8: Historia de usuario 2

Historia de Usuario	
Número: 2	Usuario: Usuario Administrativo
Nombre historia: Cambiar Contraseña del Usuario Administrativo	
Prioridad en negocio: 2	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Los datos a mostrar del usuario son personales, al acceder a la aplicación el usuario administrativo puede cambiar la contraseña asignada.	
Validación:	

El usuario debe iniciar sección con su usuario y contraseña para realizar el cambio. Ningún usuario puede acceder a los datos del otro.

Tabla 9: Historia de usuario 3

Historia de Usuario	
Número: 3	Usuario: Usuario Administrativo
Nombre historia: Crear solicitud	
Prioridad en negocio: 3	Riesgo en desarrollo: Media
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Crear las solicitudes desde la página web para poder tener registro de las solicitudes realizadas por el usuario al área de infraestructura tecnológica.	
Validación: La solicitud debe tener los parámetros establecidos para que se genere un registro real y concreto.	

Tabla 10: Historia de usuario 4

Historia de Usuario	
Número: 4	Usuario: Usuario Administrativo
Nombre historia: Listar solicitudes	
Prioridad en negocio: 4	Riesgo en desarrollo: Alta
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Las solicitudes que se han registrado deben mostrarse en una lista	
Validación:	

<p>Deben listarse las solicitudes propias del usuario administrativo, deben listarse las solicitudes abiertas o que estan vigentes y las solicitudes cerradas o solucionadas.</p>

Tabla 11: Historia de usuario 5

Historia de Usuario	
Número: 5	Usuario: Usuario Administrativo
Nombre historia: Cerrar solicitud	
Prioridad en negocio: 5	Riesgo en desarrollo: Media
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Las solicitudes que se han registrado deben ser cerradas cuando ya estén solucionadas.	
Validación: Una vez se haya cerrado la solicitud, y ante la transacción exitosa se debe actualizar la lista de solicitudes para verificar que ya no aparezca en las solicitudes abiertas.	

Tabla 12: Historia de usuario 6

Historia de Usuario	
Número: 6	Usuario: Usuario Administrativo
Nombre historia: Calificar solicitud	
Prioridad en negocio: 6	Riesgo en desarrollo: Media
Puntos estimados: 8	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción:	

Las solicitudes que se han cerrado deben ser calificadas cuando ya estén solucionadas.
Validación: Una vez se haya cerrado la solicitud, y ante la transacción exitosa se debe actualizar la lista de solicitudes para verificar que ya no aparezca en las solicitudes abiertas.

Tabla 13: Historia de usuario 7

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre historia: Autenticación del Administrador	
Prioridad en negocio: 7	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Los datos a mostrar del usuario son personales, para acceder a la aplicación se debe autenticar con el usuario y contraseña.	
Validación: El administrador debe iniciar sección con su usuario y contraseña.	

Tabla 14: Historia de usuario 8

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre historia: Cambiar Contraseña del Administrador	
Prioridad en negocio: 8	Riesgo en desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción:	

Los datos a mostrar del Administrador son personales, al acceder a la aplicación el Administrador administrativo puede cambiar la contraseña asignada por el administrador.
Validación: El Administrador debe iniciar sección con su usuario y contraseña para realizar el cambio.

Tabla 15: Historia de usuario 9

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre historia: Crear Usuario	
Prioridad en negocio: 9	Riesgo en desarrollo: Media
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Crear los usuarios administrativos desde la página web para poder que estos usuarios puedan tener acceso al aplicativo.	
Validación: Los usuarios deben tener los parámetros establecidos para que se genere un registro real y concreto.	

Tabla 16: Historia de usuario 10

Historia de Usuario	
Número: 10	Usuario: Administrador
Nombre historia: Crear Dependencia	
Prioridad en negocio: 10	Riesgo en desarrollo: Media
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	

<p>Descripción:</p> <p>Crear las dependencias desde la página web para poder tener un registro establecido en el aplicativo y así poder seleccionar el área correspondiente.</p>
<p>Validación:</p> <p>Las dependencias deben tener los parámetros establecidos dentro de la universidad para que se genere un registro real y concreto.</p>

Tabla 17: Historia de usuario 11

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre historia: Crear Incidencia	
Prioridad en negocio: 11	Riesgo en desarrollo: Media
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
<p>Descripción:</p> <p>Crear las incidencias desde la página web para poder tener un registro establecido en el aplicativo y así poder seleccionar la incidencia correspondiente al proceso de la solicitud.</p>	
<p>Validación:</p> <p>Las incidencias deben tener los parámetros establecidos dentro del área de infraestructura tecnológica para que se genere un registro real y concreto de acorde a los servicios prestados.</p>	

Tabla 18: Historia de usuario 12

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre historia: Listar solicitudes	
Prioridad en negocio: 12	Riesgo en desarrollo: Alta
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
<p>Descripción:</p> <p>Las solicitudes que se han registrado por los usuarios administrativos deben mostrarse en una lista, para así dar solución.</p>	

Validación:
Deben listarse las solicitudes propias del usuario administrativo, deben listarse las solicitudes abiertas o que estan vigentes y las solicitudes cerradas o solucionadas.

Tabla 19: Historia de usuario 13

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre historia: agregar Seguimiento	
Prioridad en negocio: 13	Riesgo en desarrollo: Alta
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Las solicitudes solucionadas por el administrador del área de infraestructura tecnológica se le agrega el respectivo seguimiento describiendo el procedimiento realizado para que el usuario administrativo revise y proceda a cerrar la solicitud si está totalmente solucionada,	
Validación: Deben listarse las solicitudes abiertas del administrador y verificar que la solicitud a la cual se dio solución no esté en la lista de solicitudes a realizar o abiertas.	

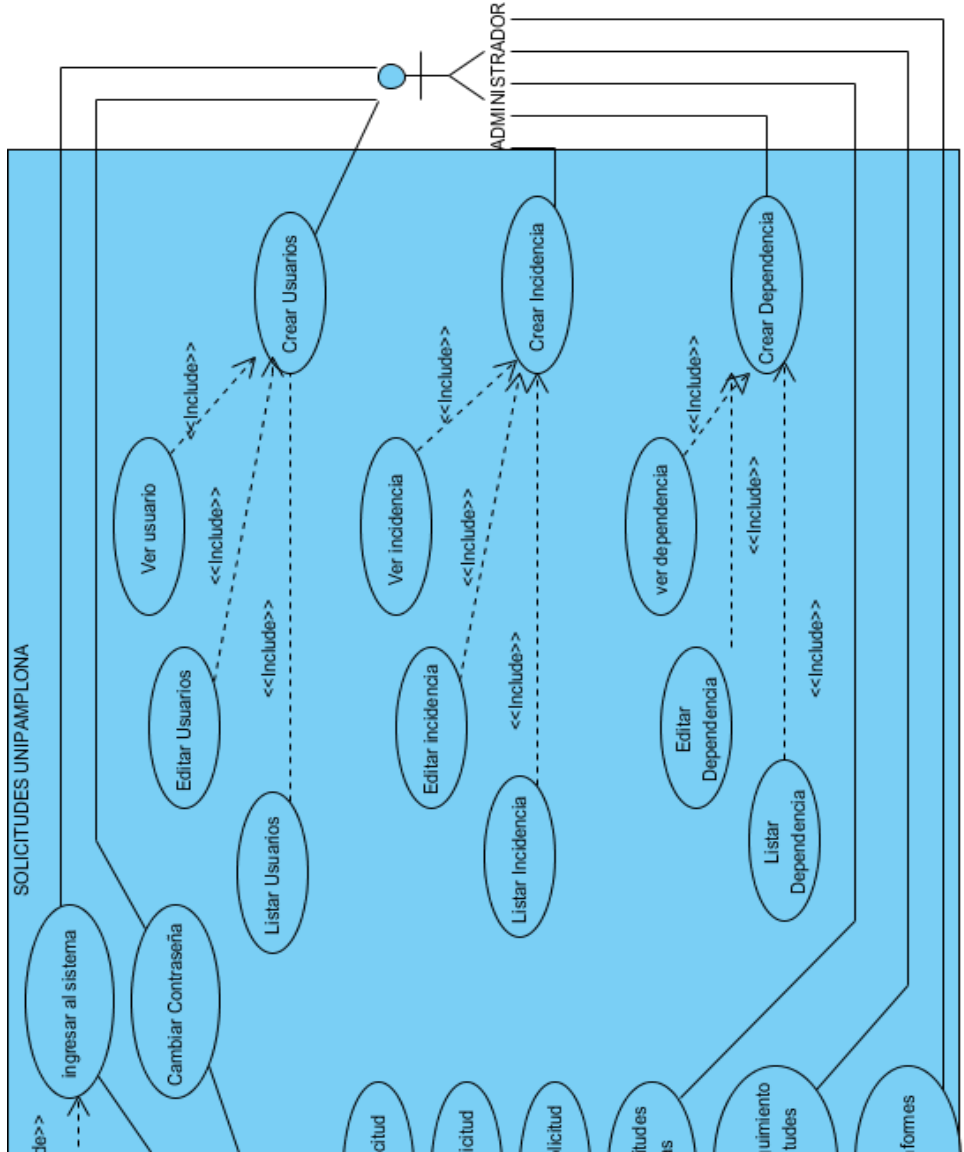
Tabla 20: Historia de usuario 14

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre historia: Generar Informes	

Prioridad en negocio: 14	Riesgo en desarrollo: Alta
Puntos estimados: 24	Iteración asignada: 1
Programador responsable: Tatiana Medina	
Descripción: Se generan informes de acuerdo a los procesos realizados solo los pueden ver el administrador del aplicativo.	
Validación: Se selecciona el periodo que quiere ver y se revisan según los datos y las solicitudes.	

4.1.1.2 Diagrama de casos de uso

Figura 14: Diagrama de casos de uso
Fuente propia de la investigacion.



4.1.1.2.1 Especificaciones de los casos de uso

Tabla 21. Caso de uso iniciar sesión

Nombre:	Ingresar al sistema
Actores:	Administradores, Sistema
Pre condición:	El sistema Esté en funcionamiento, administrador y usuario administrativo este creado.
Post condición:	Administrador y usuario administrativo pueda acceder al sistema.
Descripción:	<ol style="list-style-type: none"> 1. Administrador y usuario administrativo, llena los campos de usuario y contraseña respectivamente. 2. El sistema consulta el administrador y usuario administrativo mediante el usuario y la contraseña digitados. 3. El sistema carga la información del administrador o usuario administrativo que está iniciando la sesión. 4. El Administrador o usuario administrativo inicia la sesión. 5. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. El campo de usuario este vacío. 1.2. El campo de contraseña este vacío. 1.3. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. Administrador o usuario administrativo no este registrado. 2.2. El usuario está mal escrito. 2.3. La contraseña está mal escrita. 2.4. Regresar al paso 1

Tabla 22. Caso de uso finalizar sesión

Nombre:	Finalizar Sesión
Actores:	Administrador, Sistema
Pre condición:	El administrador o usuario administrador haya iniciado la sesión.
Post condición:	El administrador o usuario administrador pueda finalizar sesión.
Descripción:	<ol style="list-style-type: none"> 1. El administrador o usuario administrador seleccione el menú de usuario. 2. El administrador o usuario administrador busca la opción cerrar sesión y da click sobre ella. 3. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de usuario. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción cerrar sesión. 2.2. Regresar al paso 1

Tabla 23. Caso de uso cambiar contraseña

Nombre:	Cambiar contraseña
Actores:	Administrador, Sistema
Pre condición:	El administrador o usuario haya iniciado la sesión.
Post condición:	El administrador o usuario pueda cambiar contraseña.
Descripción:	<ol style="list-style-type: none"> 1. El administrador o usuario seleccione el menú de usuario. 2. El administrador o usuario busca la opción cambiar contraseña y da click sobre ella. 3. El administrador o usuario modifica su contraseña y da click sobre aceptar. 4. El sistema envía un mensaje confirmando su modificación. 5. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de usuario. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción cambiar contraseña. 2.2. Regresar al paso 1 3. <ol style="list-style-type: none"> 3.1. No modifique contraseña 3.2. Regresar al paso 2 4. <ol style="list-style-type: none"> 4.1. No confirme cambio de contraseña 4.2. Regresar al paso 3

Tabla 24. Caso de uso agregar usuario

Nombre:	Crear usuario
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado la sesión, Que exista al menos un tipo de usuario creado.
Post condición:	El administrador pueda agregar un usuario.
Descripción:	<ol style="list-style-type: none"> 1. El administrador seleccione el menú de usuario administrativo. 2. El administrador busca la opción crear usuario y da click sobre ella. 3. El administrador debe ingresar el nombre, usuario, tipo de usuario (administrador, usuario administrativo) y contraseña. 4. El sistema confirma que no hayan campos vacíos. 5. El sistema confirma si el usuario se creó con éxito. 6. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de usuario. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción crear usuario. 2.2. Regresar al paso 1 3. <ol style="list-style-type: none"> 3.1. Debe completar los campos que son obligatorios. 3.2. Regresar al paso 3. 4. <ol style="list-style-type: none"> 4.1. Debe completar los campos vacíos. 4.2. Regresar al paso 4. 5. <ol style="list-style-type: none"> 5.1. No confirme la creación de usuario 5.2. Regresar al paso 2

Tabla 25. Caso de uso agregar incidencia

Nombre:	Crear incidencia
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado la sesión.
Post condición:	El administrador pueda agregar una incidencia.
Descripción:	<ol style="list-style-type: none"> 7. El administrador seleccione el menú de incidencia. 8. El administrador busca la opción crear incidencia y da click sobre ella. 9. El administrador debe ingresar el nombre. 10. El sistema confirma que no hayan campos vacíos. 11. El sistema confirma si el usuario se creó con éxito. 12. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 6. <ol style="list-style-type: none"> 6.1. No seleccione el menú de incidencia. 6.2. Regresar al paso 1 7. <ol style="list-style-type: none"> 7.1. No de click sobre la opción crear incidencia. 7.2. Regresar al paso 1 8. <ol style="list-style-type: none"> 8.1. Debe completar los campos que son obligatorios. 8.2. Regresar al paso 3. 9. <ol style="list-style-type: none"> 9.1. Debe completar los campos vacíos. 9.2. Regresar al paso 4. 10. <ol style="list-style-type: none"> 10.1. No confirme la creación de incidencia 10.2. Regresar al paso 2

Tabla 26. Caso de uso agregar dependencia.

Nombre:	Crear dependencia
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado la sesión.
Post condición:	El administrador pueda agregar una dependencia.
Descripción:	<p>13. El administrador seleccione el menú de dependencia.</p> <p>14. El administrador busca la opción crear dependencia y da click sobre ella.</p> <p>15. El administrador debe ingresar el nombre.</p> <p>16. El sistema confirma que no hayan campos vacíos.</p> <p>17. El sistema confirma si el usuario se creó con éxito.</p> <p>18. Fin caso de uso</p>
Alternativas:	<p>11.</p> <p> 11.1. No seleccione el menú de incidencia.</p> <p> 11.2. Regresar al paso 1</p> <p>12.</p> <p> 12.1. No de click sobre la opción crear incidencia.</p> <p> 12.2. Regresar al paso 1</p> <p>13.</p> <p> 13.1. Debe completar los campos que son obligatorios.</p> <p> 13.2. Regresar al paso 3.</p> <p>14.</p> <p> 14.1. Debe completar los campos vacíos.</p> <p> 14.2. Regresar al paso 4.</p> <p>15.</p> <p> 15.1. No confirme la creación de dependencia</p> <p> 15.2. Regresar al paso 2</p>

Tabla 27. Caso de uso listar usuarios

Nombre:	Listar usuarios
Actores:	Administrador, Sistema
Pre condición:	Administrador haya iniciado la sesión, que los usuarios estén creados.
Post condición:	Mostrar un listado de usuarios.
Descripción:	<ol style="list-style-type: none"> 1. El administrador seleccione el menú de usuario. 2. El administrador busca la opción listar usuario y da click sobre ella. 3. El sistema muestra un listado de los usuarios existentes. 4. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de usuario. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción listar usuarios. 2.2. Regresar al paso 1

Tabla 28. Caso de uso listar incidencias

Nombre:	Listar incidencias
Actores:	Administrador, Sistema
Pre condición:	Administrador haya iniciado la sesión.
Post condición:	Mostrar un listado de incidencias.
Descripción:	<ol style="list-style-type: none"> 5. El administrador seleccione el menú de incidencias. 6. El administrador busca la opción listar incidencias y da click sobre ella. 7. El sistema muestra un listado de las incidencias existentes. 8. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 3. <ol style="list-style-type: none"> 3.1. No seleccione el menú de incidencias. 3.2. Regresar al paso 1 4. <ol style="list-style-type: none"> 4.1. No de click sobre la opción listar incidencias. 4.2. Regresar al paso 1

Tabla 29. Caso de uso listar dependencias

Nombre:	Listar dependencias
Actores:	Administrador, Sistema
Pre condición:	Administrador haya iniciado la sesión.
Post condición:	Mostrar un listado de dependencias.
Descripción:	<ol style="list-style-type: none"> 9. El administrador seleccione el menú de dependencias. 10. El administrador busca la opción listar dependencias y da click sobre ella. 11. El sistema muestra un listado de las dependencias existentes. 12. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 5. <ol style="list-style-type: none"> 5.1. No seleccione el menú de dependencias. 5.2. Regresar al paso 1 6. <ol style="list-style-type: none"> 6.1. No de click sobre la opción listar dependencias. 6.2. Regresar al paso 1

Tabla 30. Caso de uso editar usuario.

Nombre:	Editar usuario
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión, que los usuarios estén creados.
Post condición:	El administrador pueda editar usuario.
Descripción:	<ol style="list-style-type: none"> 1. El administrador seleccione el menú de usuario. 2. El administrador busca la opción listar usuarios y da click sobre ella. 3. El administrador de click sobre el botón editar de algún usuario. 4. El administrador modifique los datos del usuario (nombre, usuario, tipo de usuario y contraseña), y da click sobre aceptar. 5. El sistema confirma su modificación 6. Fin caso de uso.
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de usuario. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción listar usuario. 2.2. Regresar al paso 1 3. <ol style="list-style-type: none"> 3.1. No de click sobre la opción editar 3.2. Regresar al paso 2 4. <ol style="list-style-type: none"> 4.1. Regresar al paso 2 5. <ol style="list-style-type: none"> 5.1. Regresar al paso 4

Tabla 31. Caso de uso editar incidencia

Nombre:	Editar incidencia
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión, que las incidencias estén creados.
Post condición:	El administrador pueda editar incidencia.
Descripción:	<ol style="list-style-type: none"> 7. El administrador seleccione el menú de incidencia 8. El administrador busca la opción listar incidencias y da click sobre ella. 9. El administrador de click sobre el botón editar de algún incidencia. 10. El administrador modifique los datos del incidencia (nombre), y da click sobre aceptar. 11. El sistema confirma su modificación 12. Fin caso de uso.
Alternativas:	<ol style="list-style-type: none"> 6. <ol style="list-style-type: none"> 6.1. No seleccione el menú de incidencia. 6.2. Regresar al paso 1 7. <ol style="list-style-type: none"> 7.1. No de click sobre la opción listar incidencias. 7.2. Regresar al paso 1 8. <ol style="list-style-type: none"> 8.1. No de click sobre la opción editar 8.2. Regresar al paso 2 9. <ol style="list-style-type: none"> 9.1. Regresar al paso 2 10. <ol style="list-style-type: none"> 10.1. Regresar al paso 4

Tabla 32. Caso de uso Editar dependencia

Nombre:	Editar dependencia
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión, que las dependencias estén creados.
Post condición:	El administrador pueda editar dependencia.
Descripción:	<p>13. El administrador seleccione el menú de dependencia</p> <p>14. El administrador busca la opción listar dependencia y da click sobre ella.</p> <p>15. El administrador de click sobre el botón editar de algún dependencia.</p> <p>16. El administrador modifique los datos del dependencia (nombre), y da click sobre aceptar.</p> <p>17. El sistema confirma su modificación</p> <p>18. Fin caso de uso.</p>
Alternativas:	<p>11.</p> <p>11.1. No seleccione el menú de dependencia.</p> <p>11.2. Regresar al paso 1</p> <p>12.</p> <p>12.1. No de click sobre la opción listar dependencia.</p> <p>12.2. Regresar al paso 1</p> <p>13.</p> <p>13.1. No de click sobre la opción editar</p> <p>13.2. Regresar al paso 2</p> <p>14.</p> <p>14.1. Regresar al paso 2</p> <p>15.</p> <p>15.1. Regresar al paso 4</p>

Tabla 33. Caso de uso Ver usuario

Nombre:	Ver usuario
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión, que los usuarios estén creados.
Post condición:	El administrador pueda ver usuario.
Descripción:	<ol style="list-style-type: none"> 1. El administrador seleccione el menú de usuario. 2. El administrador busca la opción listar usuarios y da click sobre ella. 3. El administrador de click sobre el botón ver de algún usuario. 4. El sistema muestra la información del usuario. 5. Fin caso de uso.
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de usuario. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción listar usuario. 2.2. Regresar al paso 1 3. <ol style="list-style-type: none"> 3.1. No de click sobre la opción ver 3.2. Regresar al paso 2 4. <ol style="list-style-type: none"> 4.1. Regresar al paso 2

Tabla 34. Caso de uso ver incidencia.

Nombre:	Ver incidencia
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión, que las incidencias estén creados.
Post condición:	El administrador pueda ver incidencia.
Descripción:	<ol style="list-style-type: none"> 6. El administrador seleccione el menú de incidencia. 7. El administrador busca la opción listar incidencias y da click sobre ella. 8. El administrador de click sobre el botón ver de algún incidencia. 9. El sistema muestra la información de la incidencia. 10. Fin caso de uso.
Alternativas:	<ol style="list-style-type: none"> 5. <ol style="list-style-type: none"> 5.1. No seleccione el menú de incidencia. 5.2. Regresar al paso 1 6. <ol style="list-style-type: none"> 6.1. No de click sobre la opción listar incidencia. 6.2. Regresar al paso 1 7. <ol style="list-style-type: none"> 7.1. No de click sobre la opción ver 7.2. Regresar al paso 2 8. <ol style="list-style-type: none"> 8.1. Regresar al paso 2

Tabla 35. Caso de uso ver dependencia

Nombre:	Ver dependencia
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión, que las dependencias estén creados.
Post condición:	El administrador pueda ver dependencia.
Descripción:	<ol style="list-style-type: none"> 11. El administrador seleccione el menú de dependencia. 12. El administrador busca la opción listar dependencias y da click sobre ella. 13. El administrador de click sobre el botón ver de algún dependencia. 14. El sistema muestra la información de la dependencia. 15. Fin caso de uso.
Alternativas:	<ol style="list-style-type: none"> 9. <ol style="list-style-type: none"> 9.1. No seleccione el menú de dependencia. 9.2. Regresar al paso 1 10. <ol style="list-style-type: none"> 10.1. No de click sobre la opción listar incidencia. 10.2. Regresar al paso 1 11. <ol style="list-style-type: none"> 11.1. No de click sobre la opción ver 11.2. Regresar al paso 2 12. <ol style="list-style-type: none"> 12.1. Regresar al paso 2

Tabla 36. Caso de uso crear solicitud.

Nombre:	Crear solicitud
Actores:	Usuario, Sistema
Pre condición:	El usuario haya iniciado sesión.
Post condición:	El usuario pueda agregar una solicitud.
Descripción:	<ol style="list-style-type: none"> 1. El usuario seleccione el menú de solicitudes. 2. El usuario busca la opción crear solicitud y da click sobre ella. 3. El usuario debe ingresar la descripción de la solicitud, seleccionar la prioridad, seleccionar la dependencia y seleccionar la incidencia. 4. El sistema confirma que no hayan campos vacíos. 5. El sistema confirma si la solicitud se creó con éxito. 6. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de solicitudes. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción crear solicitud. 2.2. Regresar al paso 1 3. <ol style="list-style-type: none"> 3.1. Debe completar los campos que son obligatorios. 3.2. Regresar al paso 3. 4. <ol style="list-style-type: none"> 4.1. Debe completar los campos vacíos. 4.2. Regresar al paso 4. 5. <ol style="list-style-type: none"> 5.1. No confirme la creación de la solicitud. 5.2. Regresar al paso 4

Tabla 37. Caso de uso listar solicitud.

Nombre:	Listar solicitudes
Actores:	Administrador, Sistema
Pre condición:	Administrador o usuario haya iniciado la sesión, que las solicitudes estén creadas.
Post condición:	Mostrar un listado de solicitudes.
Descripción:	<ol style="list-style-type: none"> 1. El administrador o usuario seleccione el menú de solicitudes. 2. El administrador o usuario busca la opción listar solicitudes y da click sobre ella. 3. El sistema muestra un listado de las solicitudes existentes. 4. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de solicitudes. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción listar solicitudes. 2.2. Regresar al paso 1

Tabla 38. Caso de uso cerrar solicitud

Nombre:	Cerrar la Solicitud
Actores:	Usuario, Sistema
Pre condición:	El usuario haya iniciado la sesión, que la solicitud esta creada.
Post condición:	El usuario pueda listar solicitudes.
Descripción:	<ol style="list-style-type: none"> 4. El usuario seleccione el menú de solicitud, listar solicitudes. 5. El usuario busca la solicitud y luego la opción cerrar solicitud y da click sobre ella. 6. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 3. <ol style="list-style-type: none"> 3.1. No seleccione el menú de solicitud. 3.2. Regresar al paso 1 4. <ol style="list-style-type: none"> 4.1. No de click sobre la opción cerrar solicitud. 4.2. Regresar al paso 1

Tabla 39. Caso de uso calificar solicitud.

Nombre:	Calificar la Solicitud
Actores:	Usuario, Sistema
Pre condición:	El usuario haya iniciado la sesión, que la solicitud esta creada.
Post condición:	El usuario pueda listar solicitudes.
Descripción:	<ol style="list-style-type: none"> 7. El usuario seleccione el menú de solicitud, listar solicitudes. 8. El usuario busca la solicitud y luego la opción cerrar solicitud se debe de calificar y da click sobre ella. 9. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 5. <ol style="list-style-type: none"> 5.1. No seleccione el menú de solicitud. 5.2. Regresar al paso 1 6. <ol style="list-style-type: none"> 6.1. No de click sobre la opción cerrar solicitud y calificar solicitud. 6.2. Regresar al paso 1

Tabla 40. Caso de uso agregar seguimiento a solicitud.

Nombre:	Agregar seguimiento a solicitud
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión.
Post condición:	El administrador pueda agregar un seguimiento.
Descripción:	<ol style="list-style-type: none"> 1. El administrador seleccione el menú de solicitudes. 2. El administrador busca la opción solicitudes busca la solicitud, agregar seguimiento y da click sobre ella. 3. El administrador debe ingresar la descripción del seguimiento. 4. El sistema confirma que no hayan campos vacíos. 5. El sistema confirma si el seguimiento se creó con éxito. 6. El sistema envía correo al usuario. 7. Fin caso de uso.
Alternativas:	<ol style="list-style-type: none"> 1. <ol style="list-style-type: none"> 1.1. No seleccione el menú de solicitudes. 1.2. Regresar al paso 1 2. <ol style="list-style-type: none"> 2.1. No de click sobre la opción crear seguimiento. 2.2. Regresar al paso 1 3. <ol style="list-style-type: none"> 3.1. Debe completar los campos que son obligatorios. 3.2. Regresar al paso 3. 4. <ol style="list-style-type: none"> 4.1. Debe completar los campos vacíos. 4.2. Regresar al paso 4. 5. <ol style="list-style-type: none"> 5.1. No confirme la creación del seguimiento. 5.2. Regresar al paso 2

Tabla 41. Caso de uso generar informe

Nombre:	Generar informes
Actores:	Administrador, Sistema
Pre condición:	El administrador haya iniciado sesión.
Post condición:	El administrador pueda generar informe.
Descripción:	<ol style="list-style-type: none"> 1. El administrador seleccione el menú de solicitudes. 2. El administrador busca la opción informe de solicitudes y da click buscar. 3. El administrador debe seleccionar el periodo a generar. 4. El sistema muestra las tablas estadísticas. 5. Fin caso de uso
Alternativas:	<ol style="list-style-type: none"> 6. <ol style="list-style-type: none"> a. No seleccione el menú de solicitudes. b. Regresar al paso 1 7. <ol style="list-style-type: none"> a. No de click sobre la opción informe de solicitudes. b. Regresar al paso 1 8. <ol style="list-style-type: none"> a. Debe seleccionar el periodo. b. Regresar al paso 3. 9. <ol style="list-style-type: none"> a. Debe dar click en buscar. b. Regresar al paso 4.

4.2 Diseño De La Base De Datos

En esta fase lograremos darle el aspecto, la forma al sistema tomando en cuenta: (A) Diseños agradables.(B) Entorno amigable. (C) Creación de la BBDD. (D) Crear soluciones rápidas puntuales para reducir el riesgo. (E) No agregar funcionalidad antes de tiempo. (F) Cambiar donde y cuando sea posible.

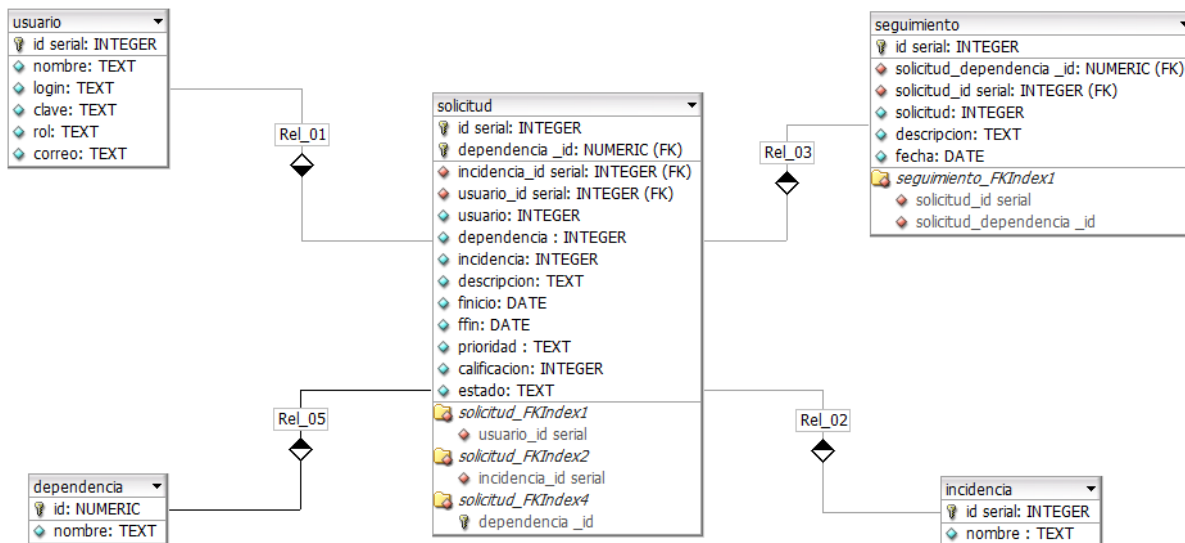
4.2.1 Entidades. Para la creación de la BBDD de nuestro sistema comenzaremos a elaborar las entidades tomando en cuenta que son objetivos concretos, que presentan interés para el sistema y sobre los que se coge información, cada entidad está constituida por uno o más atributos.

Por ejemplo, la entidad “Usuario” podría tener los atributos: nombre, apellido_usu, login, clave, correo. Perfil.

Dentro de nuestra Base de Datos utilizaremos las siguientes entidades:

1. Usuario
2. Solicitud
3. Incidencia
4. Dependencia
5. Seguimiento

Figura 15: BASE DE DATOS
Fuente: propia de la investigación.



4.2.2 Relaciones. Las relaciones representan asociaciones en el mundo real entre una o más entidades. Las relaciones se caracterizan por su nombre, el grado (número de entidades que participan en la relación), el tipo de cardinalidad (número máximo de ejemplares de una entidad asociados a una combinación de ejemplares de las otras entidades de la relación, que pueden ser 1 ó N). Los tipos de cardinalidad de asignación son:

1. Una-Una (1:1)
2. Una-Muchas (1:N)
3. Muchas-Una(N:1)
4. Muchas-Muchas(N:M)

En nuestro caso utilizaremos la cardinalidad de una a muchas entre las entidades ya mencionadas.

Figura 16: Relación Usuario - Solicitud

Fuente: propia de la investigación.

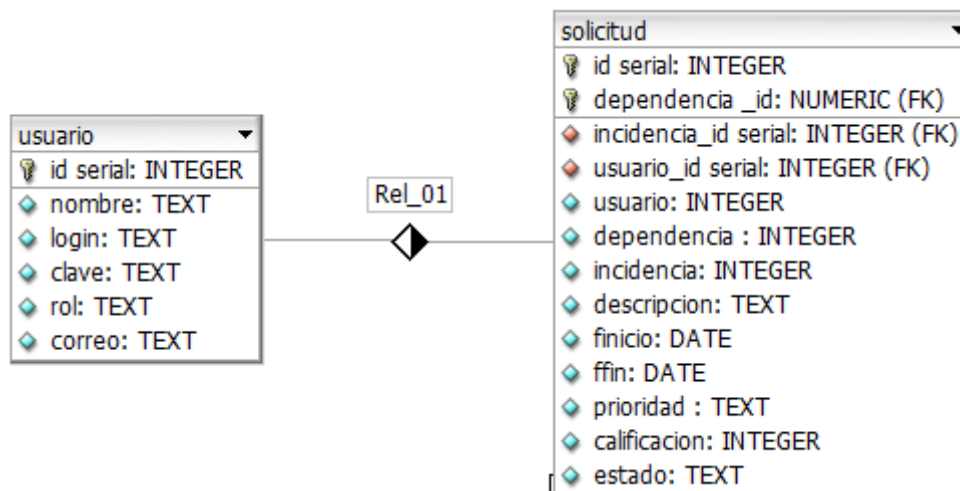


Figura 16. Entre la entidad Usuario (1) y la entidad Solicitud (2) existe una relación de uno a muchas respectivamente.

Figura 17: Relación Solicitud – Seguimiento.

Fuente: propia de la investigación.

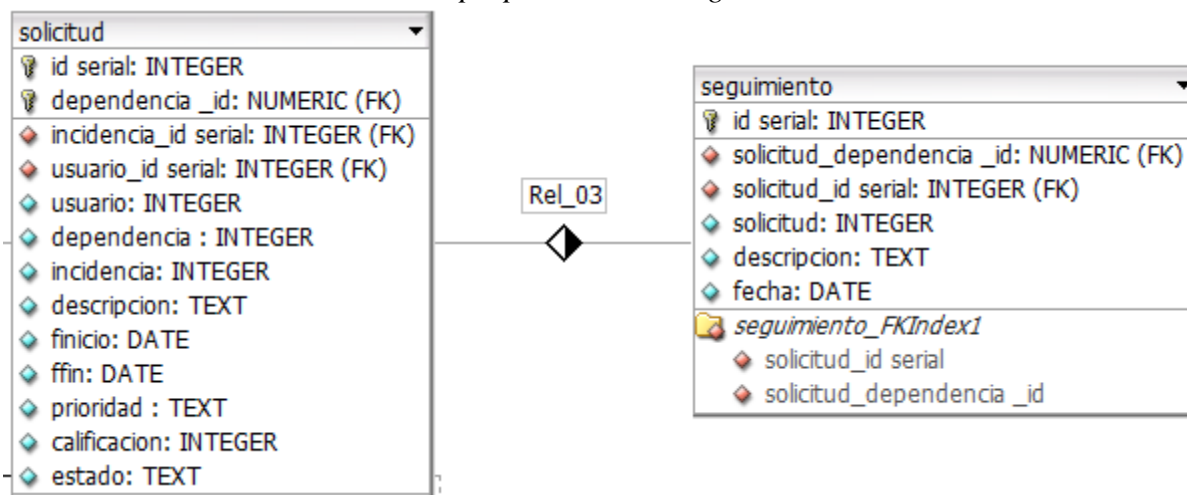


Figura 17. Entre la entidad Solicitud (1) y la entidad seguimiento (2) existe una relación de uno a muchas respectivamente.

Figura 18: Relación Dependencia - Solicitud.

Fuente: propia de la investigación.

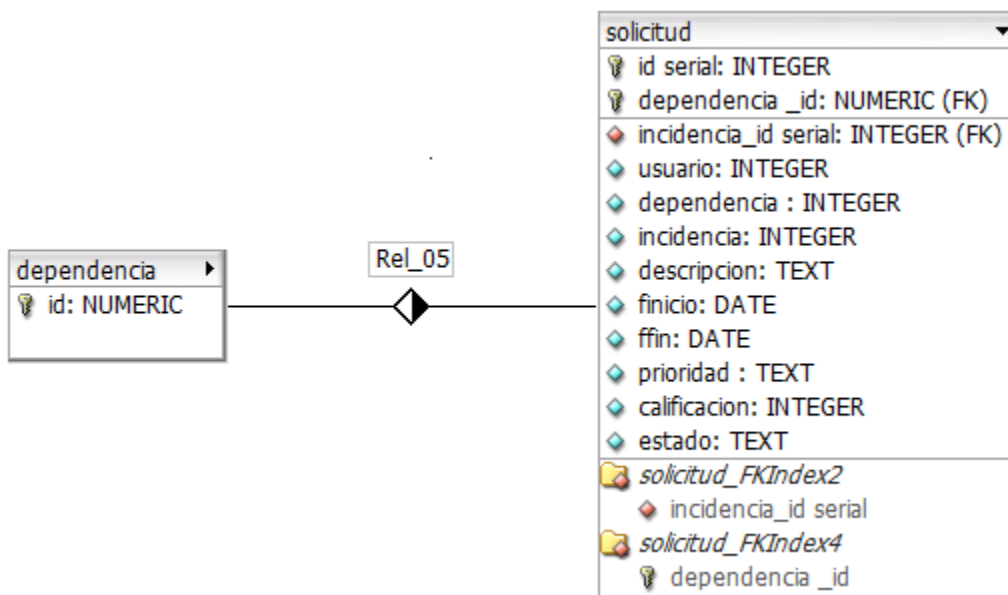


Figura 18. Entre la entidad dependencia (1) y la entidad solicitud (2) existe una relación de uno a muchas respectivamente.

Figura 19: Relación incidencia - Solicitud.

Fuente: propia de la investigación.

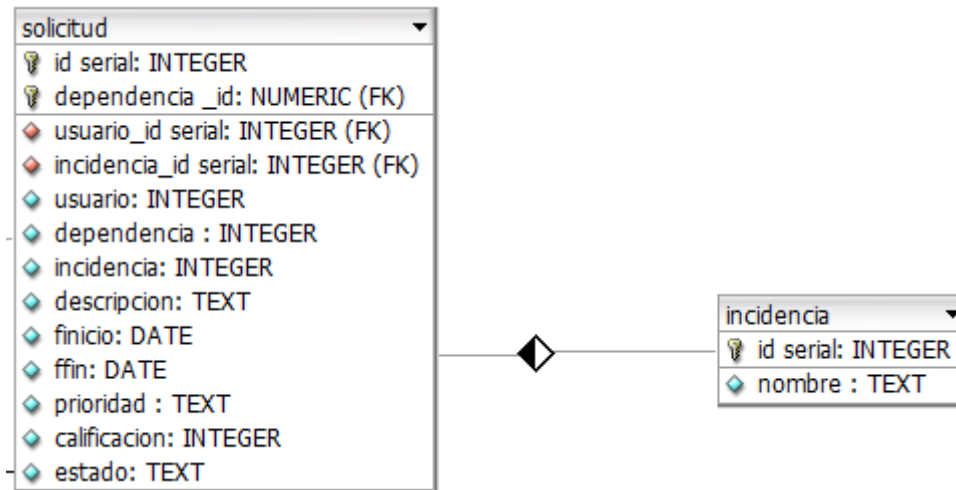
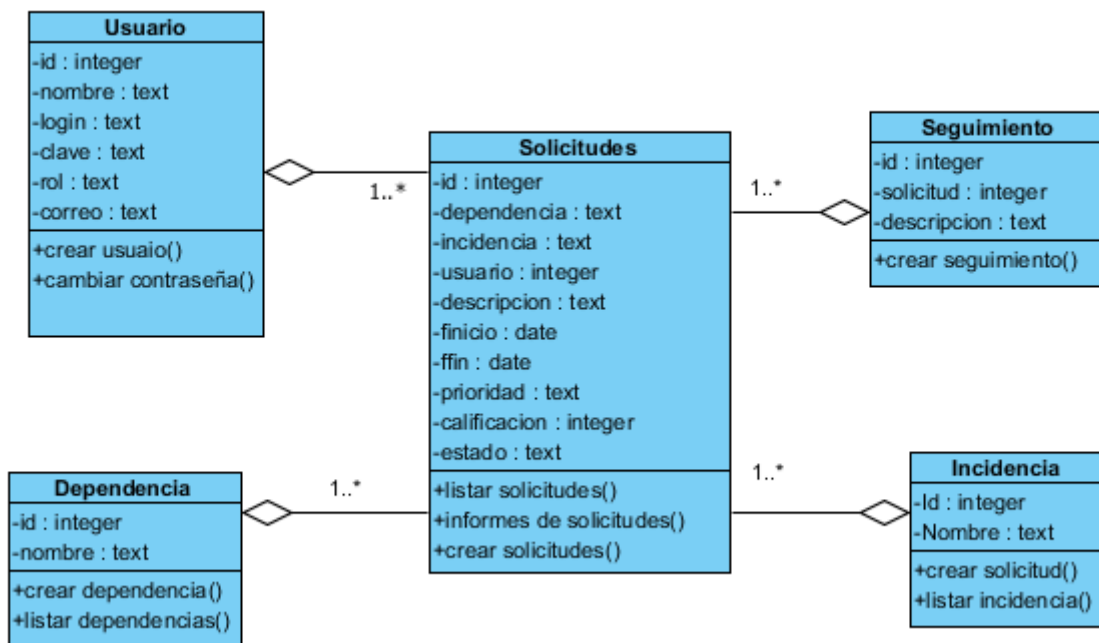


Figura 19. Entre la entidad incidencia (2) y la entidad solicitud (1) existe una relación de uno a muchas respectivamente.

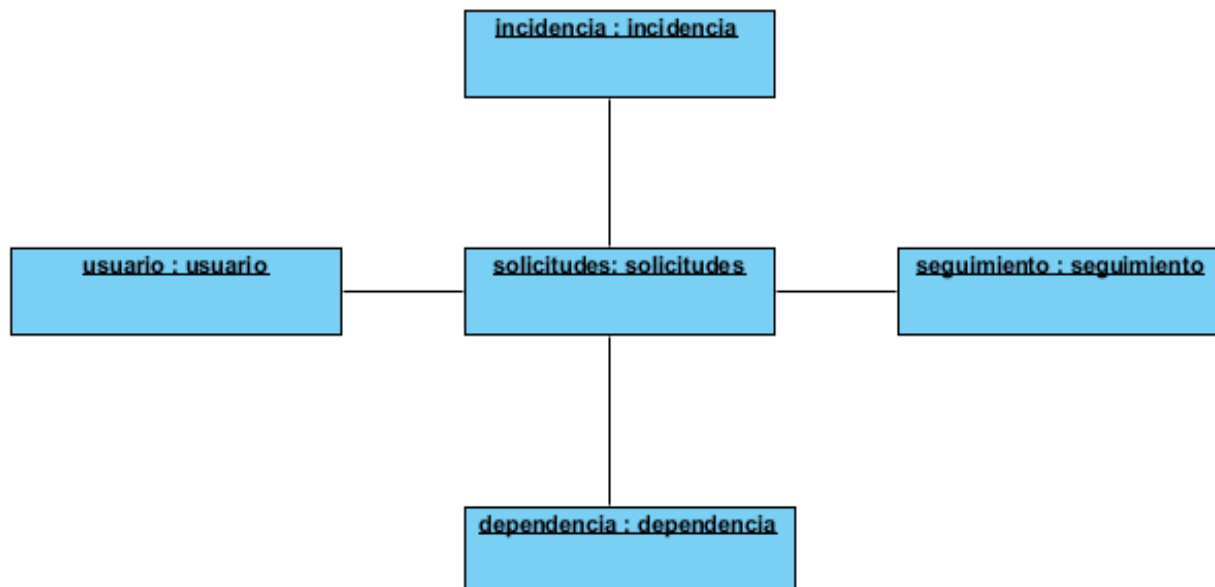
4.2.3 Diagrama de clases

Figura 20. Diagrama de clases
Fuente: propia de la investigación.



4.2.4 Diagrama de objetos

Figura 21. Diagrama de objetos
Fuente: propia de la investigación.

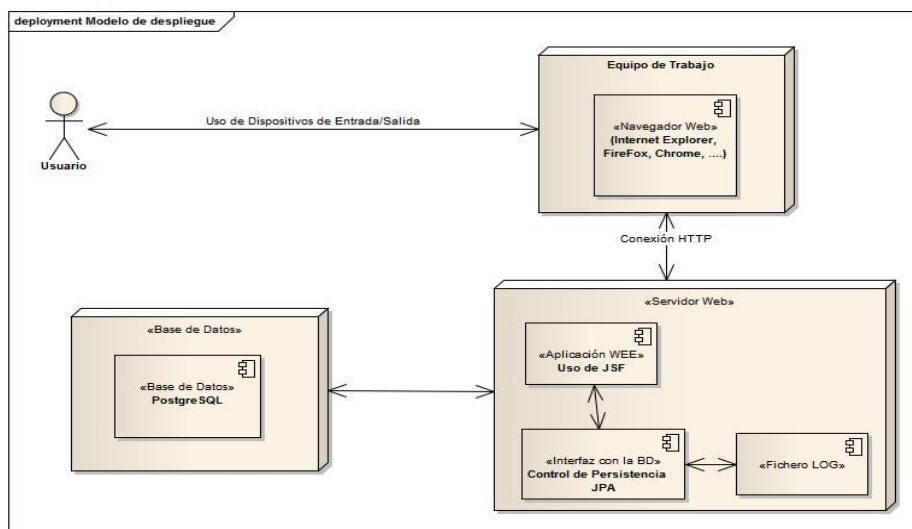


4.2.5 Modelo de despliegue. Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: (A) Dispositivos. (B) Procesadores. (C) Memoria.

Un nodo, es un objeto físico entiendo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Pueden representarse instancias o tipos de nodos que se representa como un cubo 3D en los diagramas de implementación.

Figura 22. Modelo despliegue



4.2.6 Diagrama De Secuencia

Figura 23 Diagrama de secuencia, iniciar sesión

Fuente propia de la investigación.

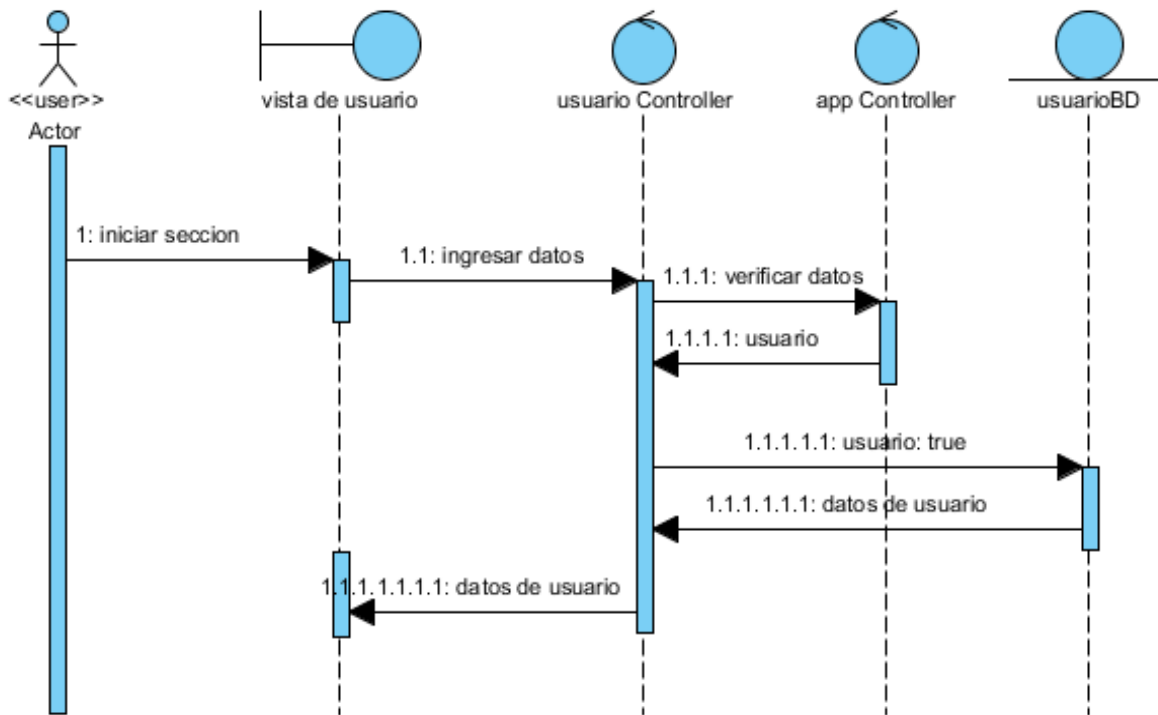


Figura 24 Diagrama de secuencia, finalizar sesión

Fuente propia de la investigación.

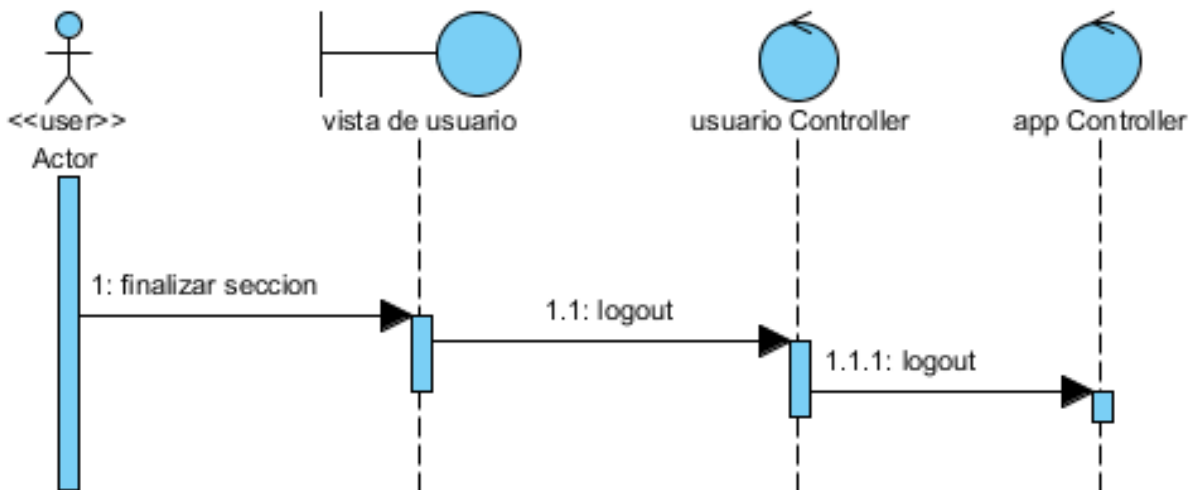


Figura 25 Diagrama de secuencia, cambiar contraseña.

Fuente propia de la investigación.

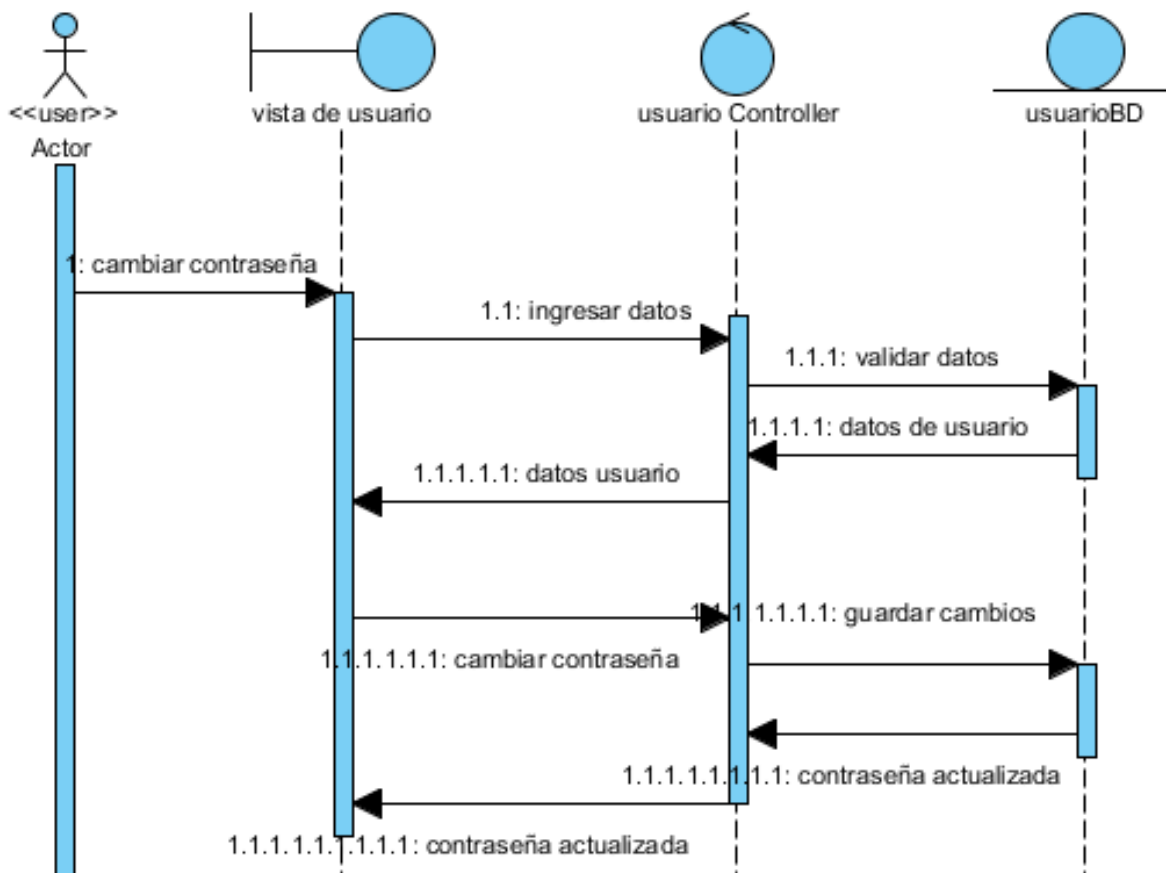


Figura 26 Diagrama de secuencia, crear usuario

Fuente propia de la investigación.

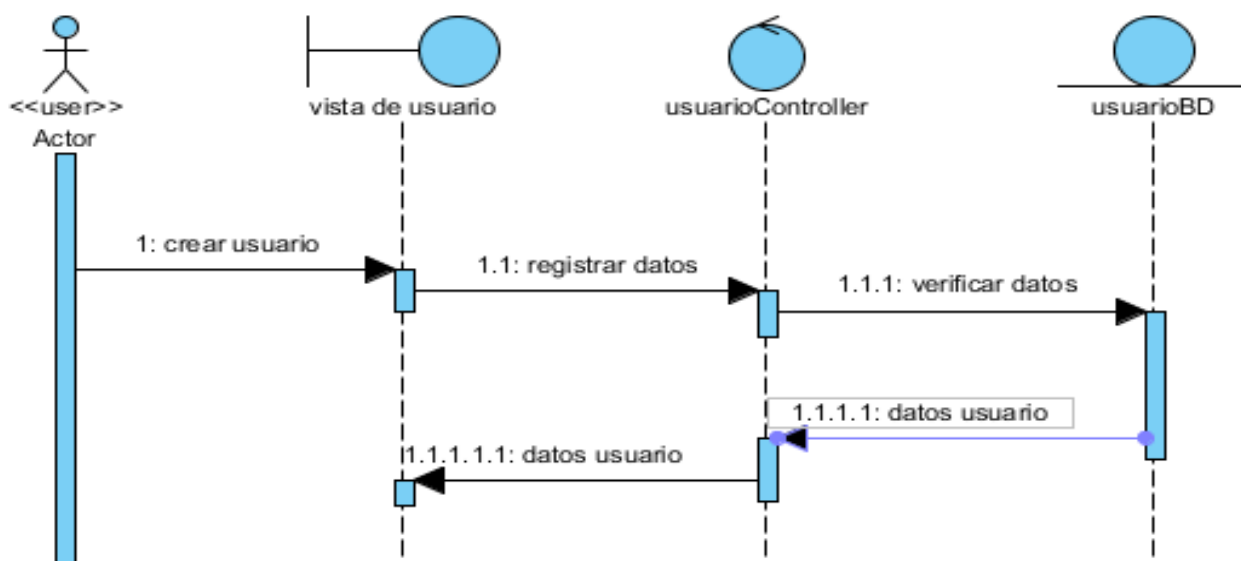


Figura 27 Diagrama de secuencia, crear dependencia

Fuente propia de la investigación.

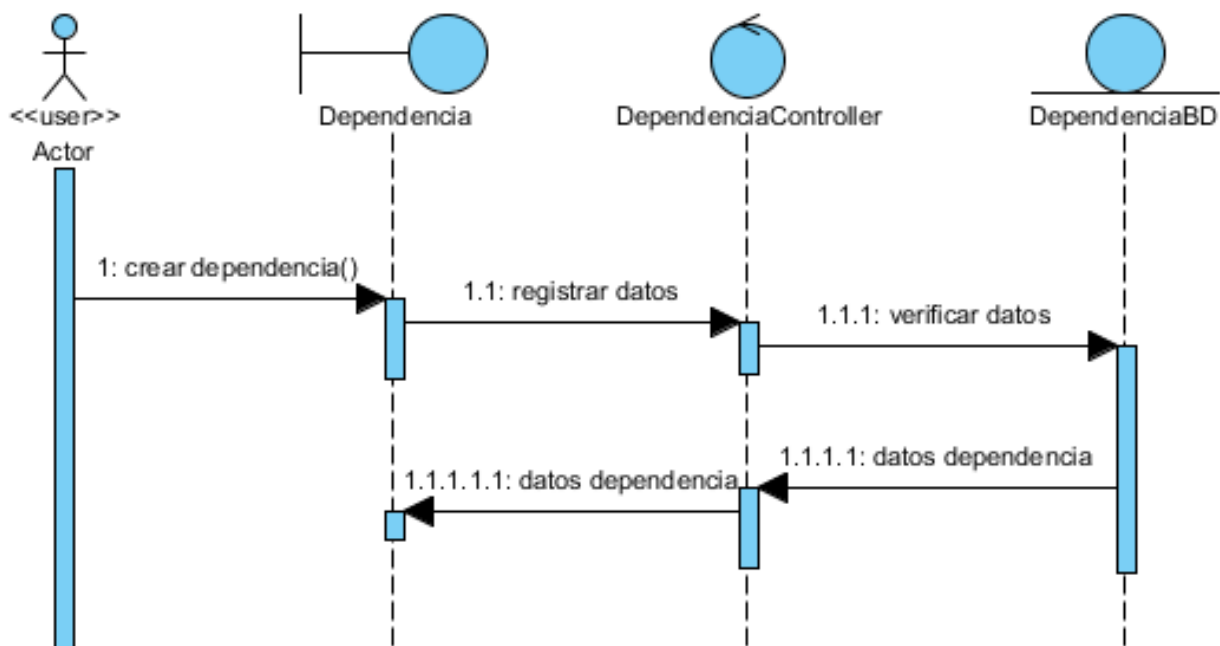


Figura 28 Diagrama de secuencia, crear incidencia

Fuente propia de la investigación.

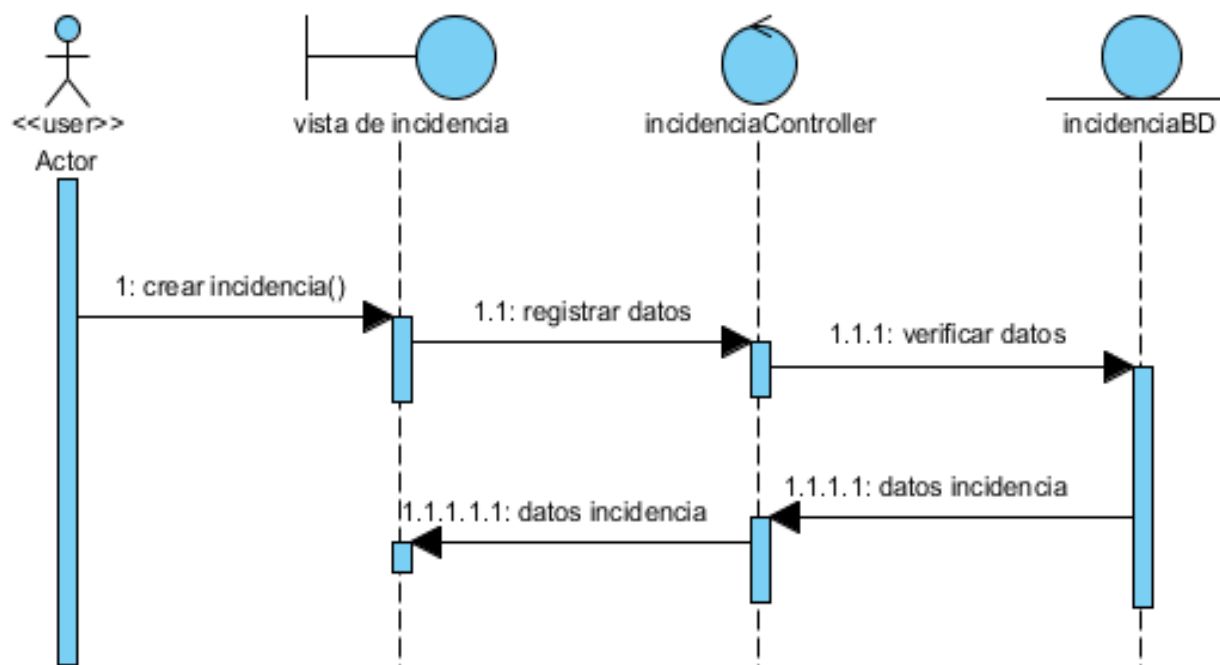


Figura 29 Diagrama de secuencia, listar usuarios

Fuente propia de la investigación.

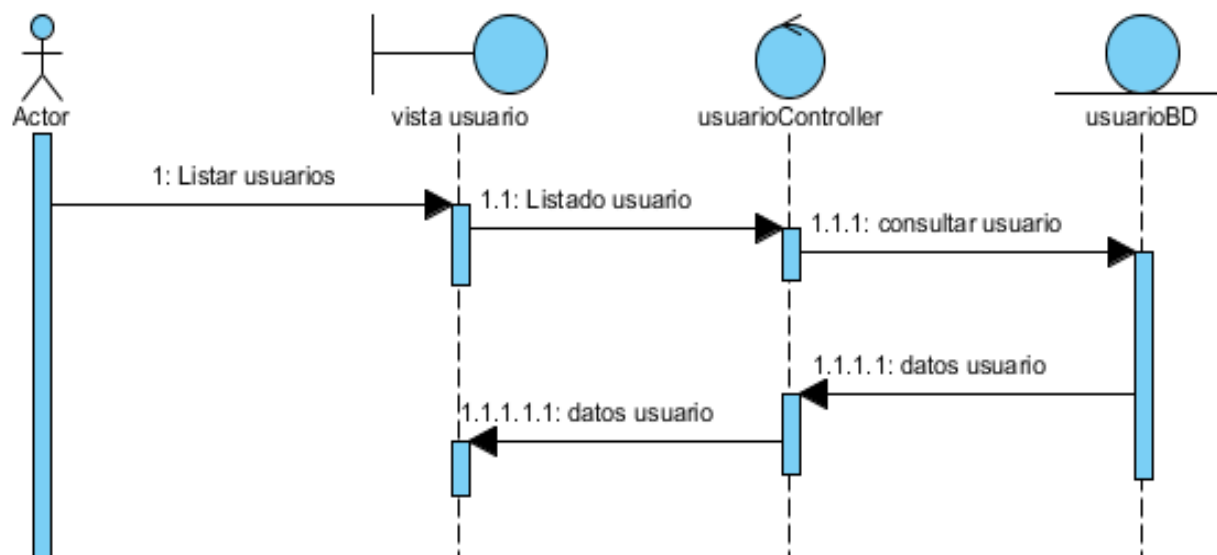


Figura 30 Diagrama de secuencia, listar dependencia

Fuente propia de la investigación.

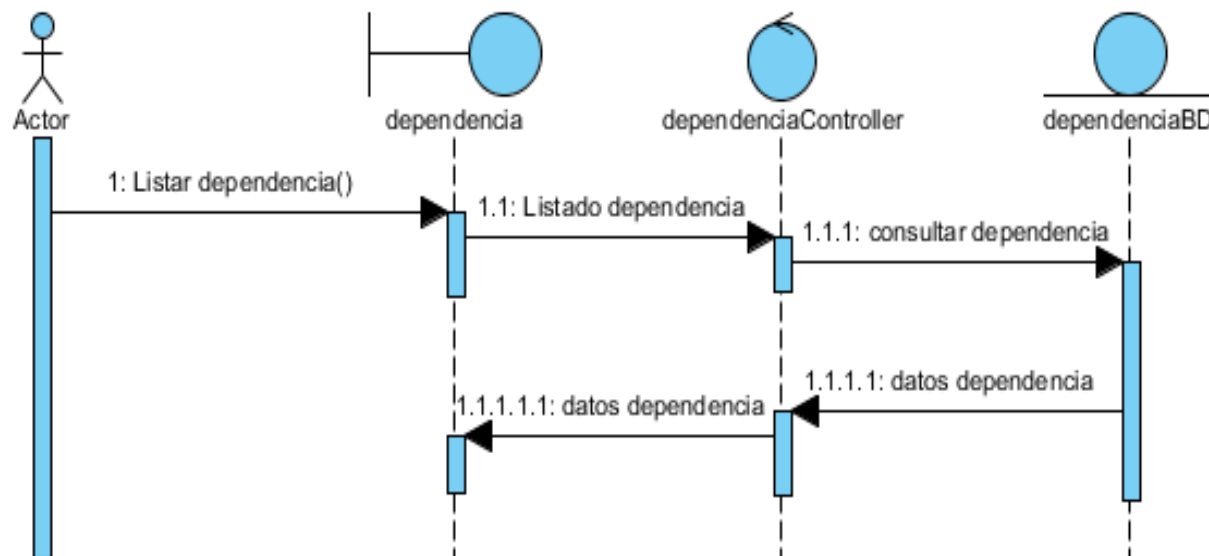


Figura 31 Diagrama de secuencia, listar incidencia

Fuente propia de la investigación.

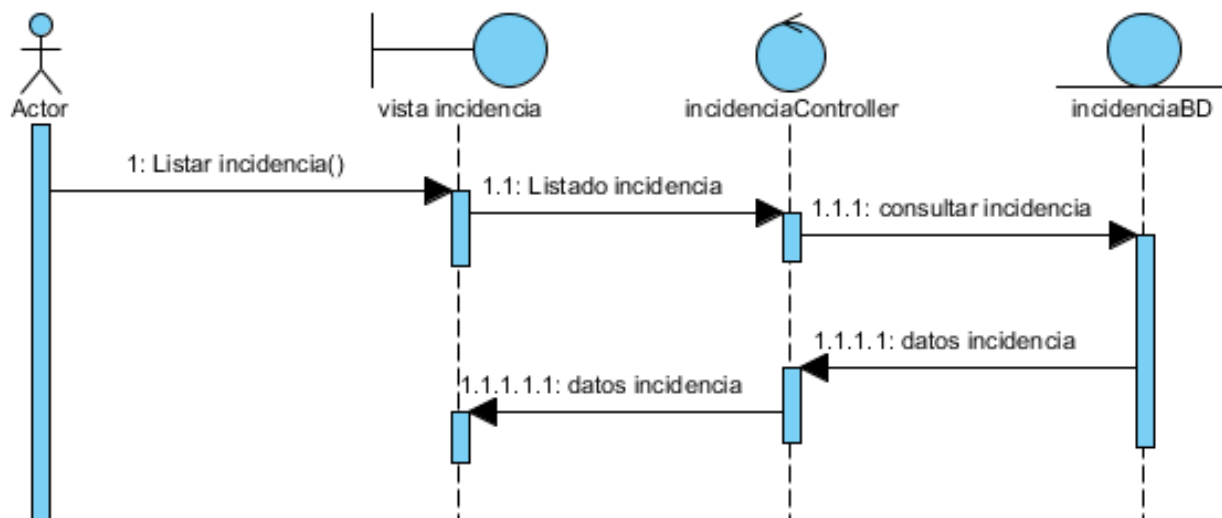


Figura 32 Diagrama de secuencia, ver usuario

Fuente propia de la investigación.

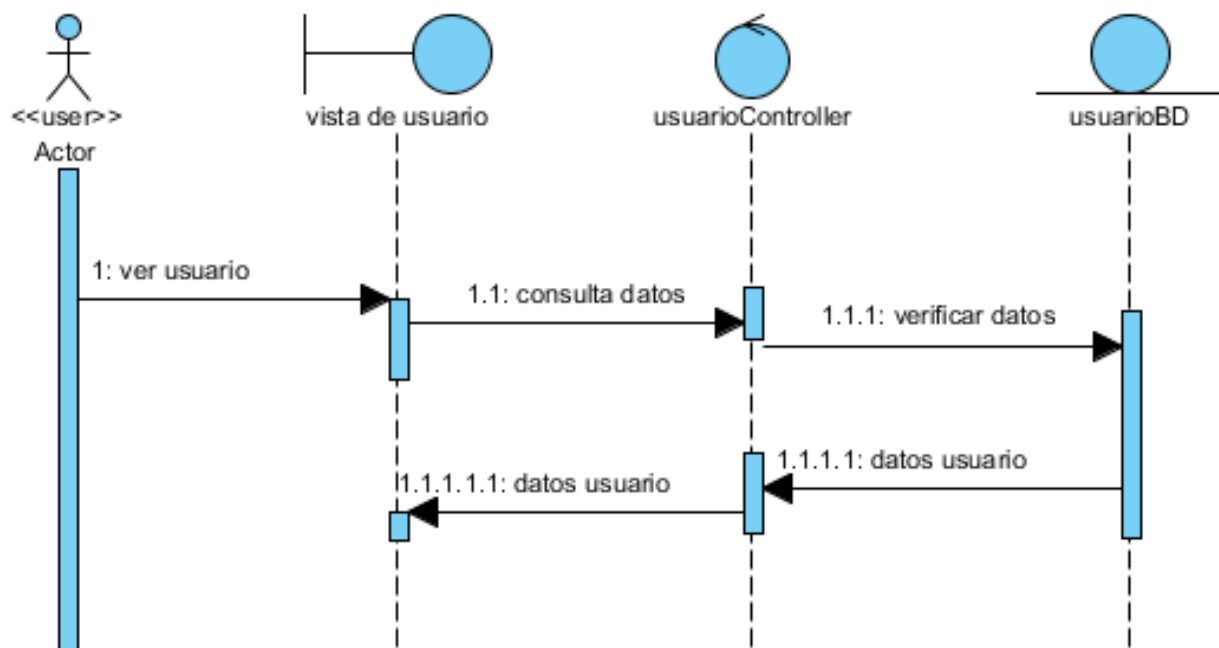


Figura 33 Diagrama de secuencia, ver dependencia

Fuente propia de la investigación.

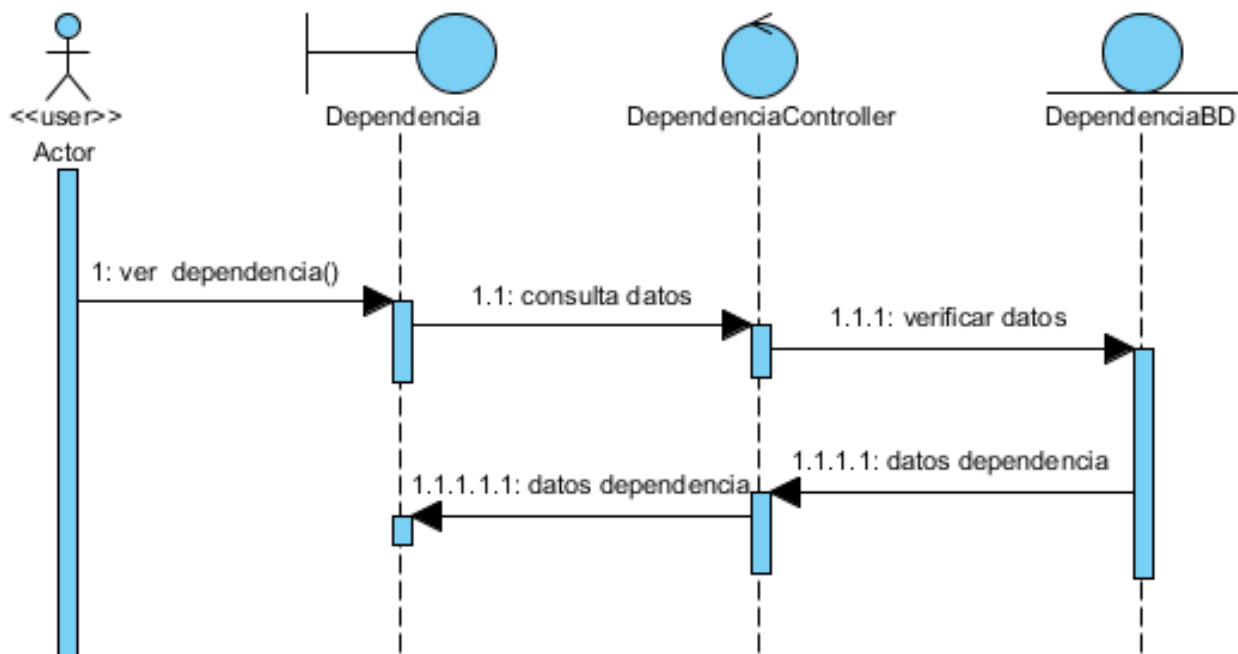


Figura 34 Diagrama de secuencia, ver incidencia

Fuente propia de la investigación.

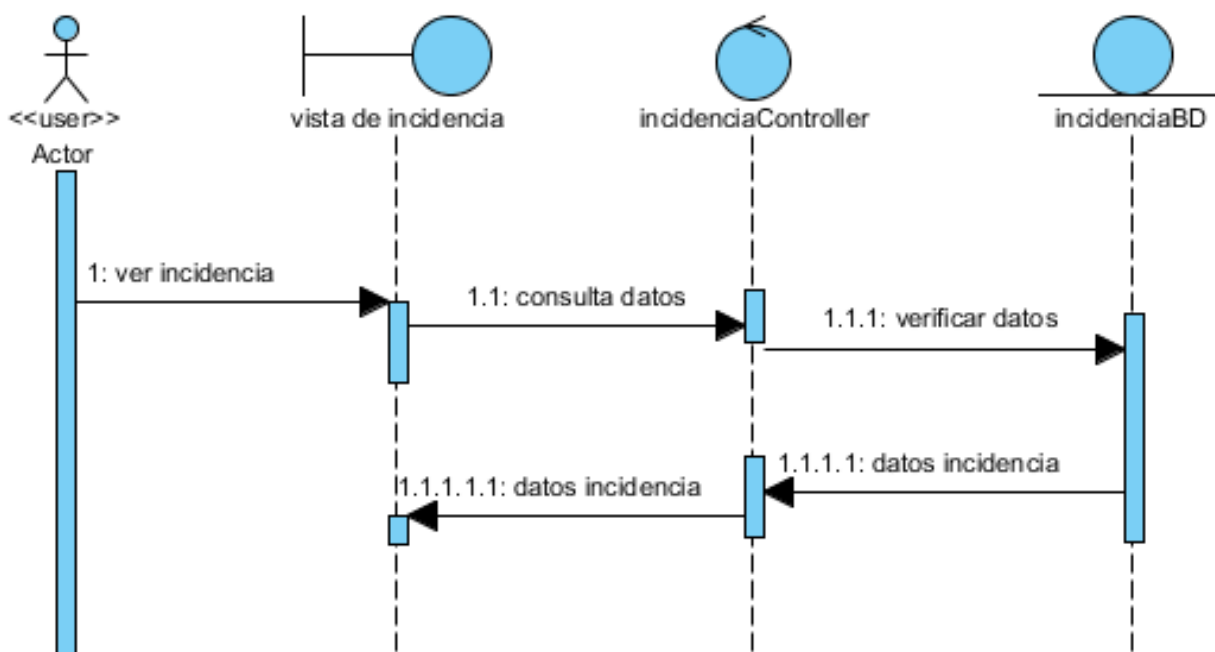


Figura 35 Diagrama de secuencia, editar usuario

Fuente propia de la investigación.

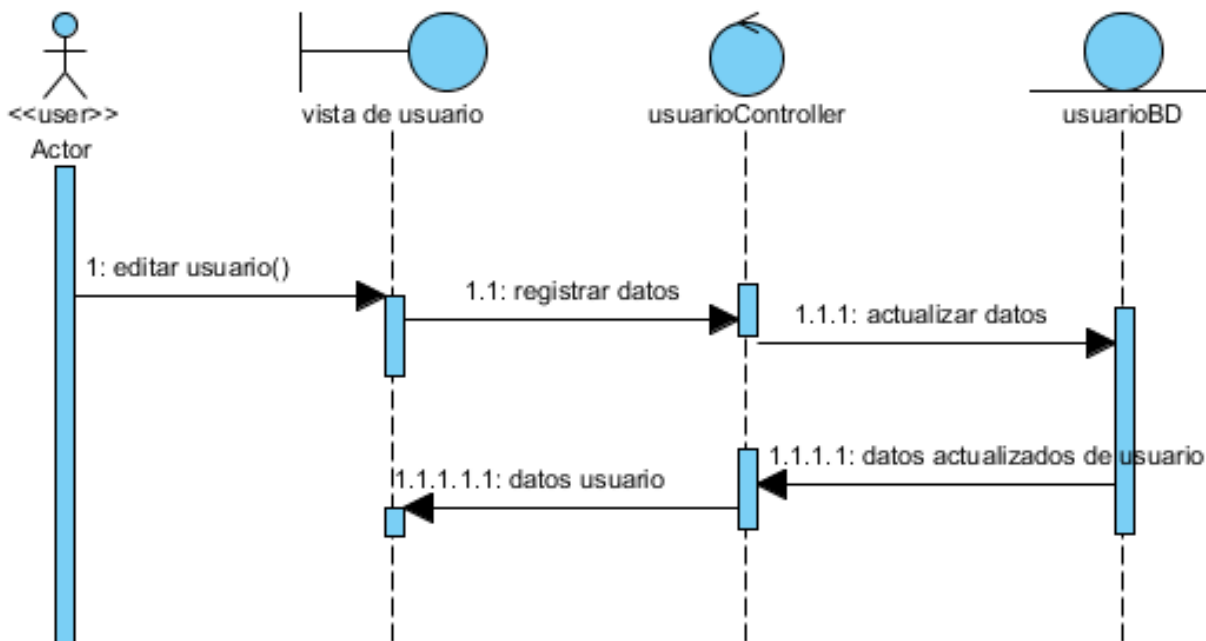


Figura 36 Diagrama de secuencia, editar dependencia

Fuente propia de la investigación.

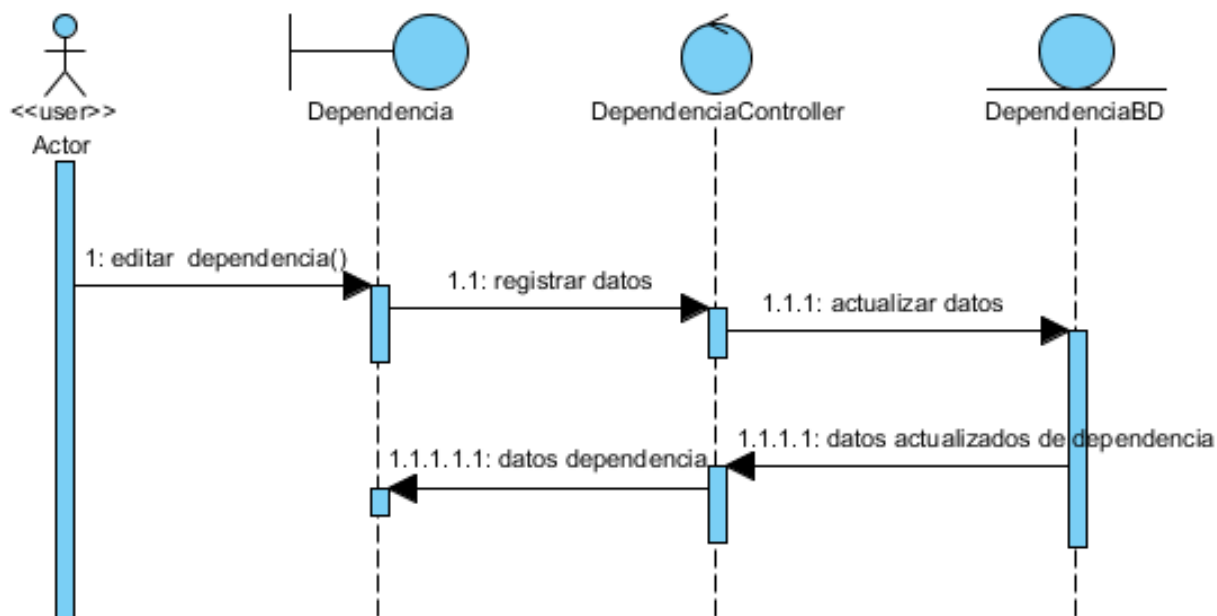


Figura 37 Diagrama de secuencia, editar incidencia
Fuente propia de la investigación.

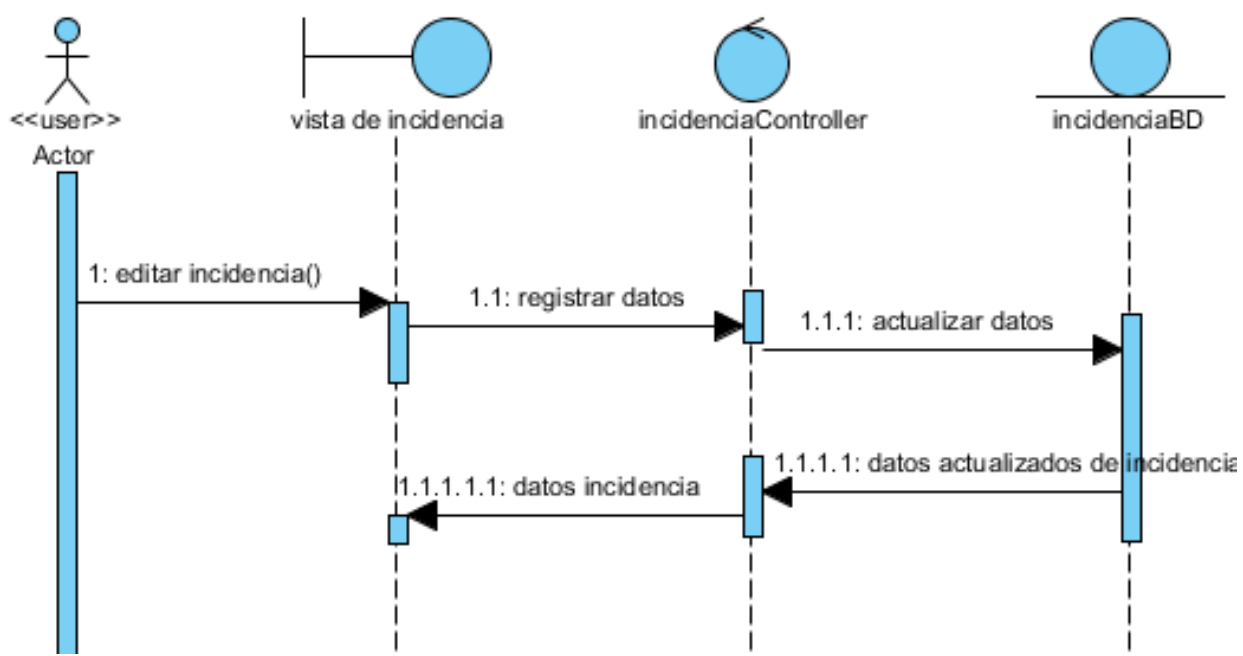


Figura 38 Diagrama de secuencia, crear solicitud
Fuente propia de la investigación.

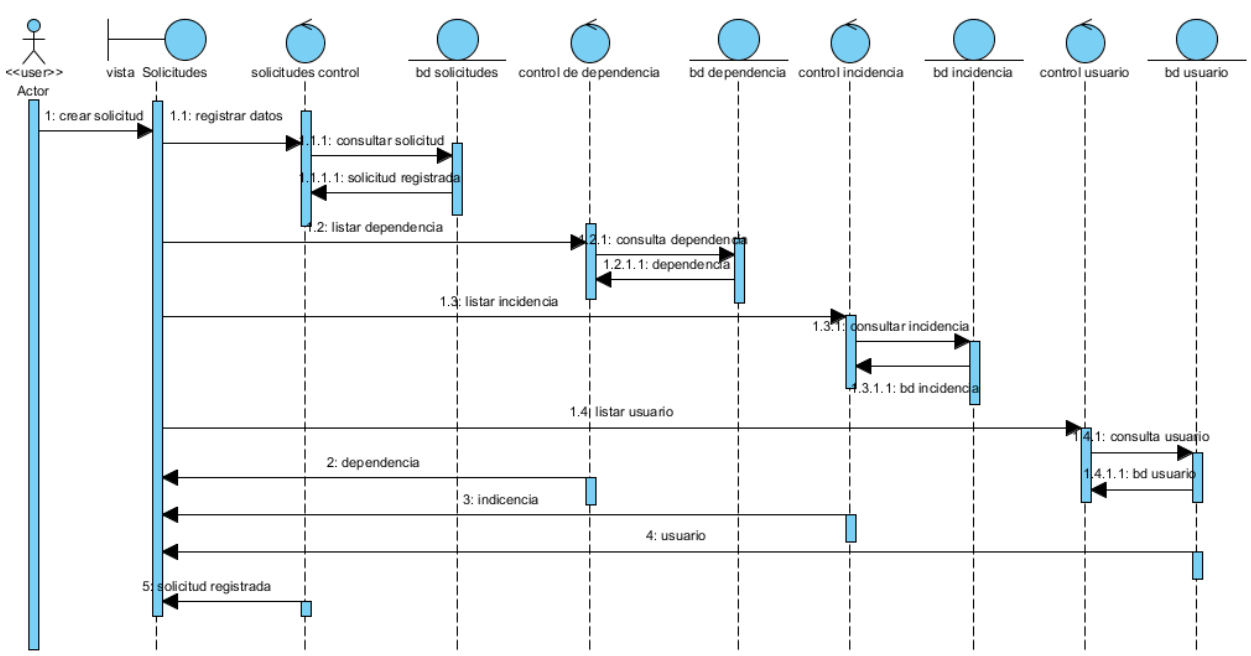


Figura 39 Diagrama de secuencia, listar solicitud

Fuente propia de la investigación.

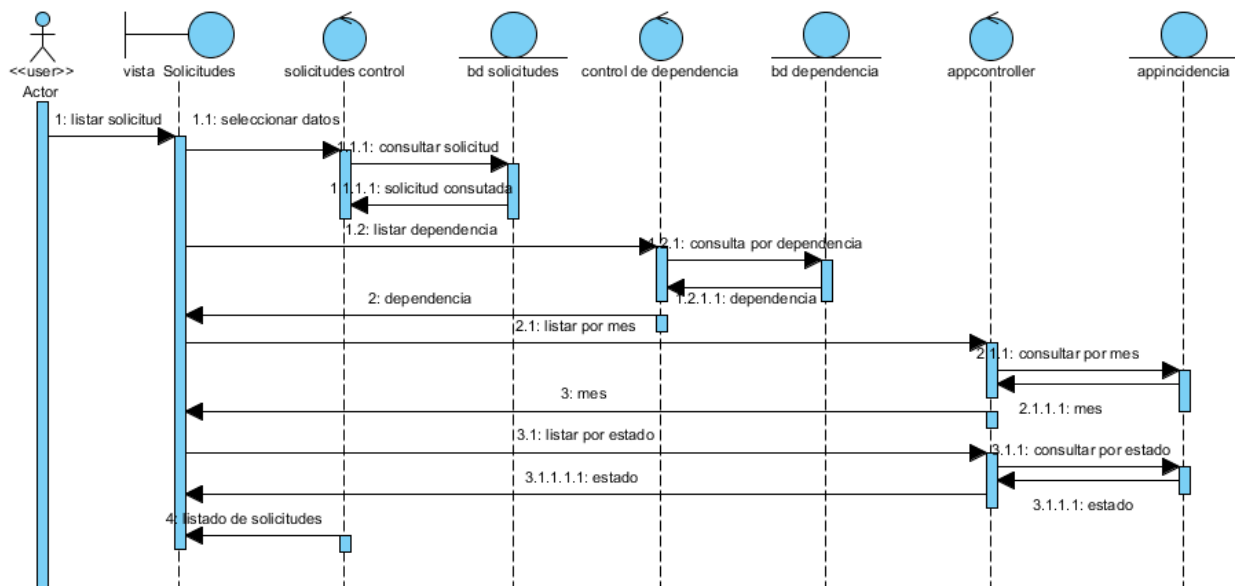


Figura 40 Diagrama de secuencia, agregar seguimiento

Fuente propia de la investigación.

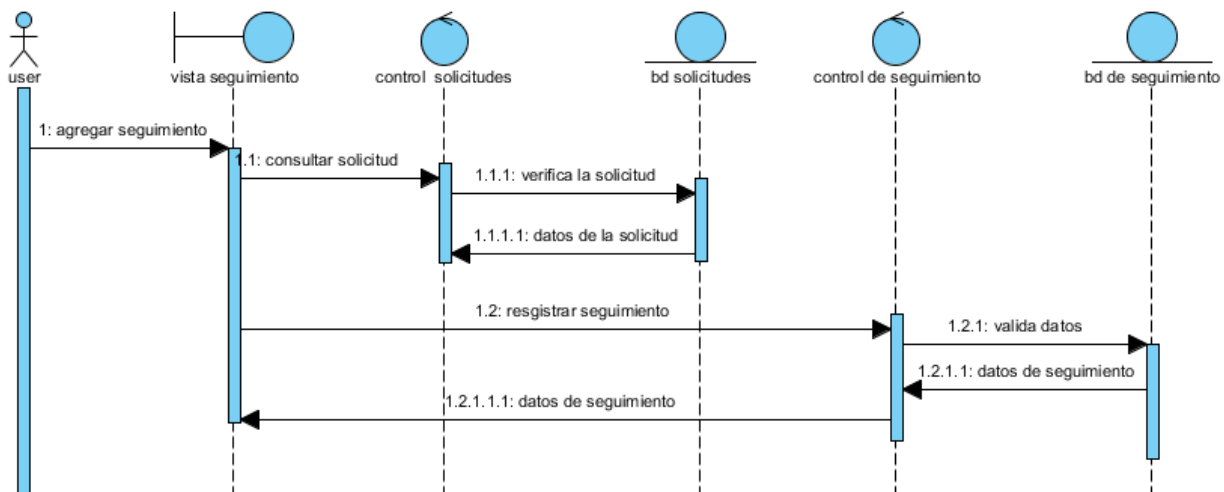


Figura 41 Diagrama de secuencia, cerrar solicitud
Fuente propia de la investigación.

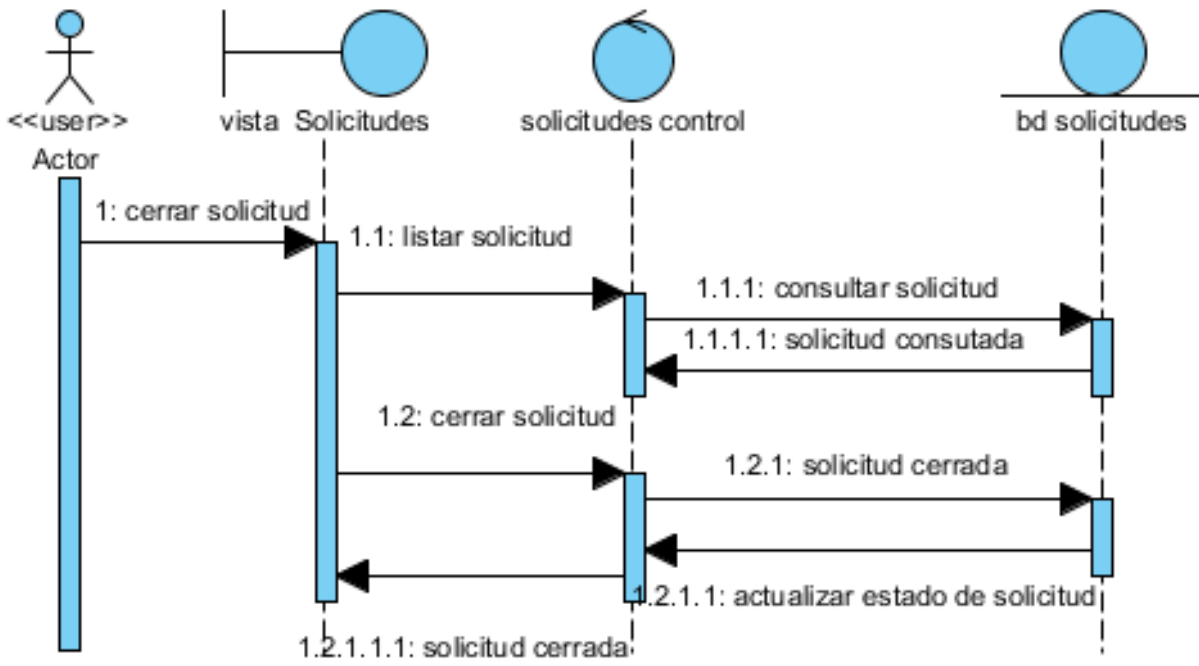
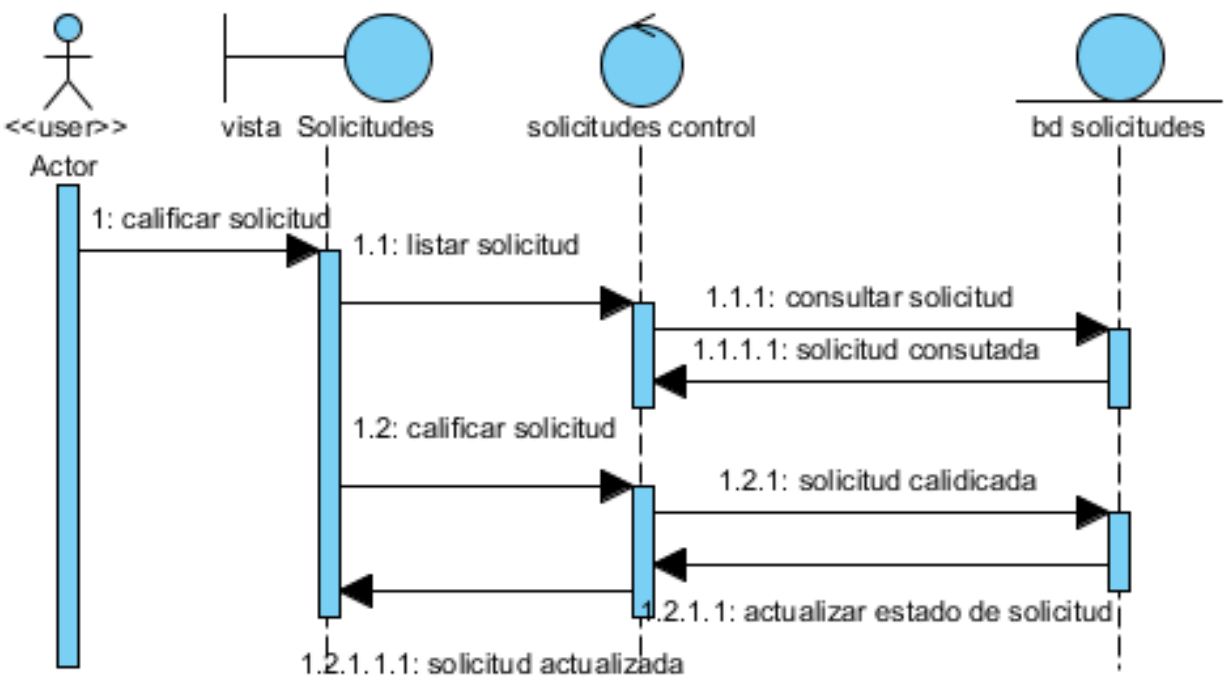


Figura 42 Diagrama de secuencia, calificar solicitud
Fuente propia de la investigación.



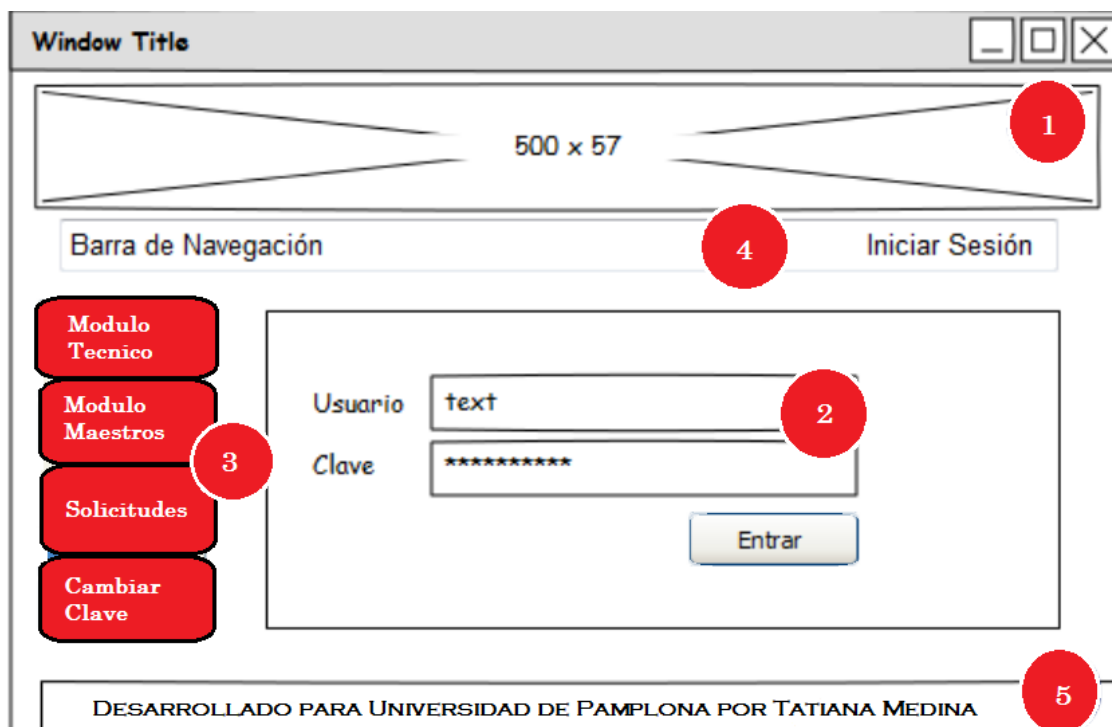
4.3 Diseño De Interfaz Abstracta

Para el diseño de la interfaz de usuario de la Aplicación Web SOLICITUDES, se tuvo en cuenta un diseño minimalista y que sea amigable con el usuario final, evitando caer en un diseño navegacional intuitivo, es decir, que el usuario tenga que adivinar cuál es el flujo a seguir para realizar determinada tarea. El diseño del Home page, donde se puede apreciar que la interfaz se divide en 3 secciones principales:

1. **Banner:** El cual contiene el nombre del aplicativo, logo de la Universidad de Pamplona y el Ciadti
2. **Panel de Inicio:** Para iniciar sesión, además una vez ingresado a la aplicación es donde se carga el contenido.
3. **Menú:** Al ingresar aparece el menú pero sin accesos, una vez ingrese se habilitan las opciones.
4. **Barra de Navegación:** Para Iniciar Sesión y Cerrar Sesión
5. **Pie de Página**

Figura 43. Descripción del diseño de la interfaz de la página de inicio de la Aplicación

Fuente propia de la investigación.



Implementación del patrón MVC. El tabla 1 muestra la forma como se implementó el patrón de diseño en la aplicación de Solicitudes.

Tabla 42. Implementación de la arquitectura de 3 niveles

Nivel	SubNivel	Locación	Archivos
Modelo	Abstracción de Base de Datos	solicitudes\main	Los archivos que componen este nivel son las clases con terminación en .class que realizan el acceso a los datos de la base de datos.
Modelo	Acceso a Datos	solicitudes\hot(List)	Los archivos que componen este nivel son las clases llamadas list, las cuales se encargan de listar los registros de las tablas de la base de datos, y entregárselas a los entitysHome para realizar las

			diferentes operaciones.
Modelo	Acción	solicitudes\hot (entityHome)	Por cada tabla existe dos clases una con terminación en Home, conocidos como ActionsBeans, encargados de manejar la lógica del negocio. Además contiene otras clases encargadas de realizar validaciones y controlar los datos.
Modelo	Layout	solicitudes\view\layout	Encapsula las partes de la vista que son comunes a todas los templates de la aplicación. Header y Footer.
Modelo	Estilo de la vista	solicitudes\view\stylesheet	Contiene los archivos que controlan el estilo de la aplicación.
Modelo	Interfaces	solicitudes\view	Archivos .xhtml los cuales contienen las interfaces de diseño. Por cada .xhtml existe su respectivos .xml el cual controla las reglas y permisos de cada interfaz.

5. Fase De Implementación.

5.1 Implementación Del Producto Con La Metodología Scrum

Sprint 001

5.2 Planificación

En esta etapa de la metodología se crearan las actividades que se van a desarrollar por el grupo de trabajo, al final de esta se obtendrá un cronograma donde se asignara el trabajo por tramos no mayores a 16 horas y se estimara el esfuerzo de cada una de estas. Para el desarrollo del producto se utilizarán del lado del servidor (jboss) tecnologías para la utilización de API tales como Netbeans y en lenguaje Java.

Estas tecnologías nos permiten iniciar una App que puede crecer con el tiempo y que puede mejorar el uso de estos componentes.

Integrantes. Tatiana Medina Vargas.

5.3 Ejecución De La Iteración.

Integrantes. Tatiana Medina Vargas.

5.4 Reportes

5.4.1 Sprint Review Meeting

Trabajo Completado: Se completaron todas las actividades propuestas en el sprint Backlog a cabalidad, se termina la ejecución con un producto funcionando de acuerdo a las historias de usuario.

Problemas Encontrados: Hubieron actividades con errores de estimación pero a las cuales se les asigno poco tiempo de esfuerzo, como hubieron algunas a las que se les asigno

tiempo excedente, el cual se pudo usar para dar un balance al tiempo faltante de las actividades de mayor costo que fueron las que incurrieron en la falta de tiempo.

Adaptaciones Necesarias: Ninguna.

5.4.2 Sprint Retrospective. El buen diseño fue crucial para el buen desarrollo en la implementación del producto, los casos de usuario fueron fáciles de entender e interpretar por lo que hubo poca ambigüedad, esto favoreció que durante este sprint no ocurrieran cambios y poder entregar el producto en la fecha establecida. Hubo problemas con la estimación de los tiempos los cuales nos dan una base más sólida para la definición de estos criterios para proyectos futuros, donde hubo actividades desfasadas y otras con poco tiempo asignado.

5.5 Retrospectiva

5.5.1 Descripción del Software

El software que se entrega presenta una completa adaptación del proceso de solicitudes que nos describieron al inicio, haciendo una comunicación completa con la información de las solicitudes y arrojando los datos actualizados. De igual forma estos datos son presentados al administrador del área para ser interpretados y procesados por medio de la aplicación. Otra de la funcionalidad del Aplicación es que hacer un control de las solicitudes, dentro de este incluye el registro de las solicitudes de cada dependencia, la observación de las solicitudes procesadas, El registro de de nuevas incidencias, la consulta de informe estadístico el cual lo hace el administrador del área para llevar así sus reportes y por último el cierre y calificación de la solicitud el cual es importante para medir la trazabilidad de satisfacción en cuanto a la solución a su solicitud dada por el área.

El App se encuentra disponible en la funcionalidad planteada. Con las pruebas completamente realizadas.

Requerimientos del Sistema

- a. Tener acceso a internet.
- b. Tener navegadores instalados en el equipo.
- c. Tener código y contraseña asignada para el uso del App.

Reportes

Sprint Review Meeting

En la reunión con el equipo y los usuarios el día 1 de noviembre de 2016, se determinó que la funcionalidad fue aplicada y aportaron como sugerencia las siguientes validaciones.

Validaciones a tener en cuenta:

- No debe registrarse un usuario más de una vez en un mismo evento.
- La fecha de cierre de la solicitud debe tomarse por el sistema en el servicio web.

6 Cronograma De Actividades

El cronograma de actividades fue realizado según las actividades establecidas con el director, según la responsabilidad y el orden con el que maneje el cronograma de actividades y se mantengan en las fechas establecidas para cada actividad, el trabajo de grado tiene estipulada una duración de 6 meses.

Tabla 43. Cronograma de Actividades

No	Nombre de tarea	Duración	Comienzo	Fin
1)	Aplicativo	26 días	15/09/2016	11/10/2016
2)	Servicio de web	26 días	15/09/2016	11/10/2016
3)	Recolección de historias	1 día	15/09/2016	15/09/2016
4)	Selección de requisitos para el sprint.	1 día	16/09/2016	16/09/2016
5)	Planificación del sprint	1 día	17/09/2016	17/09/2016
6)	Diseñar MER, Diagrama de clases, secuencia, paquetes y Su respectiva documentación.	5 días	18/09/2016	22/09/2016
7)	Implementación física del MER, Diagrama de clases y paquetes	1 día	23/09/2016	23/09/2016
8)	Codificación del sprint.	15 días	24/09/2016	24/09/2016
9)	Ejecución de las pruebas	1 día	25/09/2016	25/09/2016
10)	Inspección y adaptación.	1 día	26/09/2016	25/09/2016
11)	Creación de Manuales	3 días	27/09/2016	30/09/2016
12)	Creación de manual de usuario	1 día	01/10/2016	01/10/2016
13)	Creación Manual de instalación	2 días	02/10/2016	03/10/2016
14)	Realización Documentación del software	5 días	04/11/2016	08/12/2016
15)	Documentar Modelo de análisis	3 días	05/11/2016	08/11/2016
16)	Documentar Modelo del Diseño	3 días	09/11/2016	12/11/2016
17)	Documentar Documento de pruebas	2 días	13/11/2016	15/11/2016
18)	Recomendaciones	1 día	16/11/2016	16/11/2016
19)	conclusiones	1 día	17/11/2016	17/11/2016

8 Fase de Pruebas

7.1 Documento de Pruebas

Descripción del entregable: En la colección formada por los casos de prueba y procedimientos de prueba. Este artefacto incluye el propósito de las pruebas, que elemento se va a probar, las herramientas a utilizar y con qué recursos, así como el documento que va hacer entregado. Al tener el resultado de las pruebas se puede comparar lo obtenido con lo esperado.

8.2 Historial de Revisiones

Tabla 44. Historial de Revisiones

Versión	Fecha	Autor	Descripción
1.0	02/11/2016	Tatiana Medina	Prueba Sprint 1

7.3 Plan de Pruebas

7.3.1 Introducción

Alcance. Pruebas realizadas para la aplicación web para la gestión y control de las solicitudes del área de infraestructura tecnológica de la Universidad de Pamplona, sede Villa del Rosario, en este plan se evidenciaran las pruebas de validación para determinar si cumplen o no con lo definido en las historias de usuario.

Propósito de la Evaluación

1. Encontrar tantos errores como sea posible.
2. Determinar que el funcionamiento de los casos de uso cumplen con lo establecido en las historias de usuario.

Motivadores de la prueba

1. Requerimientos Funcionales.

7.3.2 Ámbito de las Pruebas

Dentro del Ámbito

1. Prueba del inicio de sesión.
2. Prueba de creación de solicitudes.
3. Prueba de creación de usuarios.
4. Prueba de creación de dependencias.
5. Prueba de creación de incidencias.
6. Prueba del listado de solicitudes abiertas.
7. Prueba del listado de solicitudes cerradas.
8. Prueba del listado de usuarios.
9. Prueba del listado de incidencias.
10. Prueba del listado de dependencias.
11. Prueba de Crud de dependencia.
12. Prueba de Crud de incidencia.
13. Prueba de Crud de usuarios.
14. Prueba de cambio de contraseña de usuario
15. Prueba del informe de solicitudes.

7.3.3 Lista de Ideas de las Pruebas.

Las pruebas serán identificadas siguiendo la técnica de generación de casos de prueba a través de los casos de uso, detallando los siguientes pasos:

1. Para cada caso de uso, se identifican los caminos posibles, permitiendo establecer los escenarios.

2. Para cada uno de los caminos, se identifican los conjuntos de valores de entrada y precondiciones, al igual que el resultado esperado.
3. Se hace, a través de una tabla, un resumen por cada caso de uso que muestre los distintos caminos posibles con sus entradas y salidas.

Los recursos utilizados para la identificación de las pruebas se mencionan a continuación:

1. El documento de especificación de requerimientos del software.
2. El documento de arquitectura de software.
3. Generación de pruebas de sistema a partir de la especificación funcional.
4. Mejora de la calidad de los requisitos mediante la generación de pruebas.
5. Especificación e implementación de casos de prueba.

7.3.4 Enfoque de las Pruebas

T-00: Pruebas de Validación

Tabla 45 Enfoque de las Pruebas.

Objetivo:	Verificar que los casos de uso corresponden a con lo especificado en las historias de usuario.
Descripción:	Proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su cometido.
Técnicas:	Se definirán los flujos y el comportamiento de cada uno de estos, en los casos de uso que requieren formularios se evaluarán los tipos de caracteres incluidos en estos.
Fases:	<ol style="list-style-type: none"> 1. Se identifica el caso de uso a probar 2. Se definen los flujos de información 3. se prueban cada uno de los flujos establecidos 4. Se determina el éxito o fallo del caso de uso.
Entorno de prueba:	Conexión a internet en cualquier lugar, El usuario debe estar registrado en la base de datos de prueba.
Hardware:	Servidor jboss de pruebas con Tomcat.
Software:	Es necesario tener instalado una aplicación. <ul style="list-style-type: none"> • Tomcat • Postgres • Netbeans • Navegadores.

Configuración:	La prueba se realizara con los datos arrojados desde la aplicativo web también tenga un funcionamiento correcto.
Criterios de Éxito:	<ul style="list-style-type: none"> • Todas las funciones arrojen un mensaje de salida • Cada función realice la validación correspondiente para su ejecución.
Consideraciones Especiales:	Ninguna.

7.3.5 Herramientas para las Pruebas

Tabla 46 Hardware

Recurso	Cantidad	Descripción	Perfil del usuario
Computador	1	Computador para que sea el servidor con la aplicación web instalada y configurada, cuyas características del equipo sean altas en rendimiento.	Administrador
Computador	1	Computador con conexión y acceso a internet.	Usuario administrativo de la universidad

7.3.6 Casos de Prueba

Tabla 47 Casos de Prueba 1

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 1	Tipo(s) de Pruebas(s): VALIDACION ID	
Requerimiento: 1	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Ingreso al sistema	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016

Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el móvil 2. Usuario registrado en la base de datos. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	El nombre del usuario consta de letras	Ytmv	Acceder al aplicativo según el perfil administrador o usuario.	Acceso al aplicativo según el perfil usuario o administrador.
2	Debe ingresar usuario y contraseña	Ytmv 1234	Acceder al aplicativo web	Usuario y clave correctos
3	Si ingresa solo el usuario	Ytmv	Acceder al aplicativo web	Acceso negado ingrese de nuevo
4	Si ingresa solo contraseña	1234	Acceder al aplicativo web	Acceso negado ingrese de nuevo
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 48 Casos de Prueba 2

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 2	Tipo(s) de Pruebas(s): VALIDACION ID	
Requerimiento: 2	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:2	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Cambiar Contraseña.	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación:	Fecha de Ejecución:

	02/12/2016	02/12/2016		
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe estar en el módulo de cambiar contraseña 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Debe ingresar usuario y contraseña	Ytmv 1234	Acceder al aplicativo según el perfil administrador o usuario.	Acceso al aplicativo según el perfil usuario o administrador.
2	Cambiar la contraseña	5467	Acceder al aplicativo e ir al módulo de cambiar contraseña.	Contraseña modificada.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 49 Casos de Prueba 3

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 3	Tipo(s) de Pruebas(s): VALIDACION ID	
Requerimiento: 3	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Crear solicitud.	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016

Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de solicitudes. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo de solicitudes, en crear solicitud.	N/A	Se abre formulario a diligenciar; se diligencia de acuerdo a los parámetros de la solicitud.	La solicitud fue creada exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 50 Casos de Prueba 4

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 4	Tipo(s) de Pruebas(s): VALIDACION ID	
Requerimiento: 4	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Cerrar solicitud.	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016
Condición(es) para que se ejecute el Caso de Prueba:		
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 		

<ol style="list-style-type: none"> 2. Usuario registrado en la base de datos. 3. Solo lo puede hacer el usuario administrativo de la universidad. 4. Debe de estar en el módulo de solicitudes. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo de solicitudes y luego solicitudes abiertas.	N/A	Buscar la solicitud a cerrar, la que ya le hayan solucionado y que sea satisfecha la solución.	La solicitud fue cerrada exitosamente y no aparece en el listado de solicitudes abiertas.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 51 Casos de Prueba 5

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 5	Tipo(s) de Pruebas(s): VALIDACION ID	
Requerimiento: 5	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Calificar solicitud.	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016
Condición(es) para que se ejecute el Caso de Prueba:		
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Solo lo puede hacer el usuario administrativo de la universidad. 		

4. Debe de estar en el módulo de solicitudes.				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo de solicitudes y luego solicitudes abiertas.	N/A	Buscar la solicitud a cerrar, después del cierre se debe dar la calificación de satisfacción de esa solicitud.	La solicitud fue calificada exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 52 Casos de Prueba 6

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 6	Tipo(s) de Pruebas(s): VALIDACION	
Requerimiento: 6	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Listar solicitudes.	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016
Condición(es) para que se ejecute el Caso de Prueba:		
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de solicitudes. 4. Debe de estar en el menú listado de solicitudes. 		

Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo de solicitudes y luego listas de solicitudes	N/A	Se listan las solicitudes en las cuales se debe seleccionar el estado (abierto o cerrado)	Las solicitudes se muestran en pantalla según la selección.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 53 Casos de Prueba 7

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1	
ID Caso de Uso: 7	Tipo(s) de Pruebas(s): VALIDACION	
Requerimiento: 7	Ambiente de Prueba: Oficina desarrollo.	
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.	
ID/Nombre Caso de Prueba: Crear usuarios.	Nombre del Probador: Tatiana Medina	
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016
Condición(es) para que se ejecute el Caso de Prueba:		
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, usuarios. 		

Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, usuarios y crear usuario.	N/A	Se abre formulario a diligenciar con los datos del usuario a crear.	El usuario fue creado exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 54 Casos de Prueba 8

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1			
ID Caso de Uso: 8	Tipo(s) de Pruebas(s): VALIDACION			
Requerimiento: 8	Ambiente de Prueba: Oficina desarrollo.			
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.			
ID/Nombre Caso de Prueba: Crear dependencia.	Nombre del Probador: Tatiana Medina			
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016		
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, dependencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido

1	Seleccionar el módulo técnico, dependencia y crear dependencia.	N/A	Se abre formulario a diligenciar con los datos de la dependencia a crear.	La dependencia fue creada exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 55 Casos de Prueba 9

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 9		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 9		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: Crear incidencia.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, incidencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo	N/A	Se abre formulario a diligenciar con los datos	La incidencia fue creada exitosamente.

técnico, incidencia y crear incidencia.		de la incidencia a crear.	
Criterios de Aprobación del Caso de Prueba:			
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>			
Fecha de Aprobación del Caso de Prueba: 02/11/2016			

Tabla 56 Casos de Prueba 10

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1			
ID Caso de Uso: 10	Tipo(s) de Pruebas(s): VALIDACION			
Requerimiento: 10	Ambiente de Prueba: Oficina desarrollo.			
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.			
ID/Nombre Caso de Prueba: editar dependencia.	Nombre del Probador: Tatiana Medina			
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016		
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, editar dependencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, dependencia y	N/A	Se abre formulario a editar con los nuevos datos de la dependencia.	La dependencia fue editada exitosamente.

	editar dependencia.			
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 57 Casos de Prueba 11

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 11		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 11		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: ver dependencia.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, ver dependencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, ver dependencia	N/A	Se abre formulario de datos de la dependencia.	Datos de dependencia exitosamente.
Criterios de Aprobación del Caso de Prueba:				

Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>
Fecha de Aprobación del Caso de Prueba: 02/11/2016

Tabla 58 Casos de Prueba 12

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes	Nivel de Prueba: 1			
ID Caso de Uso: 12	Tipo(s) de Pruebas(s): VALIDACION			
Requerimiento: 12	Ambiente de Prueba: Oficina desarrollo.			
ID/Nombre Escenario:	Autor del Caso de Prueba: Tatiana Medina.			
ID/Nombre Caso de Prueba: ver incidencia.	Nombre del Probador: Tatiana Medina			
Versión del Caso de Prueba: 1.0	Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016		
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, listar incidencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, listar incidencia.	N/A	Se abre listas de incidencia	Muestra el listado de incidencia
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 59 Casos de Prueba 13

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 13		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 13		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: editar incidencia.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 5. Conexión a internet desde el computador 6. Usuario registrado en la base de datos. 7. El administrador del aplicativo es el único que tiene acceso a este módulo. 8. Debe de estar en el módulo técnico, editar incidencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, incidencia y editar incidencia.	N/A	Se abre formulario a editar con los nuevos datos de la incidencia.	La incidencia fue editada exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 60 Casos de Prueba 14

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 14		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 14		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: ver usuarios.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 5. Conexión a internet desde el computador 6. Usuario registrado en la base de datos. 7. El administrador del aplicativo es el único que tiene acceso a este módulo. 8. Debe de estar en el módulo técnico, ver usuarios. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, ver usuarios	N/A	Se abre formulario de datos de usuarios.	Datos de usuarios exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 61 Casos de Prueba 15

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 15		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 15		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: editar usuarios.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. El administrador del aplicativo es el único que tiene acceso a este módulo. 4. Debe de estar en el módulo técnico, editar usuarios. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo técnico, usuario y editar usuario.	N/A	Se abre formulario a editar con los nuevos datos del usuario.	El usuario fue editado exitosamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 62 Casos de Prueba 16

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 16		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 16		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: Listar usuarios.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de maestro. 4. Debe de estar en el menú listado de usuarios. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo maestro y luego listo de usuarios.	N/A	Se listan los usuarios	Los usuarios se muestran en pantalla
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 63 Casos de Prueba 17

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 17		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 17		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: Listar dependencia.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de maestro. 4. Debe de estar en el menú listado de dependencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo maestro y luego listo de dependencia.	N/A	Se listan las dependencias.	El listado de dependencia se muestran en pantalla
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 64 Casos de Prueba 18

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 18		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 18		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: Listar incidencia.		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condicón(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de maestro. 4. Debe de estar en el menú listado de incidencia. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo maestro y luego listo de incidencia.	N/A	Se listan las incidencias.	El listado de incidencia se muestran en pantalla
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <input checked="" type="checkbox"/> Fallo: <input type="checkbox"/>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 65 Casos de Prueba 19

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 19		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 19		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: Informe de solicitudes		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de maestro. 4. Debe de estar en el menú solicitudes y luego en Informe de solicitudes. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo maestro, solicitudes y luego Informe de solicitudes.	N/A	Se listan el Informe de solicitudes.	El Informe de solicitudes se muestran en pantalla
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

Tabla 66 Casos de Prueba 20

ID/Nombre/Sistema/Proyecto: Aplicación web de solicitudes		Nivel de Prueba: 1		
ID Caso de Uso: 20		Tipo(s) de Pruebas(s): VALIDACION		
Requerimiento: 20		Ambiente de Prueba: Oficina desarrollo.		
ID/Nombre Escenario:		Autor del Caso de Prueba: Tatiana Medina.		
ID/Nombre Caso de Prueba: agregar seguimiento a Solicitudes		Nombre del Probador: Tatiana Medina		
Versión del Caso de Prueba: 1.0		Fecha de Creación: 02/12/2016	Fecha de Ejecución: 02/12/2016	
Condición(es) para que se ejecute el Caso de Prueba:				
<ol style="list-style-type: none"> 1. Conexión a internet desde el computador 2. Usuario registrado en la base de datos. 3. Debe de estar en el módulo de maestro. 4. Debe de estar en el menú solicitudes. 5. Debe listar las solicitudes de solicitudes cuyo estado sea abierto. 6. Debe dar en ver y luego en agregar seguimiento. 				
Para la Ejecución del Caso de Prueba:				
Nro. Paso Flujo	Condición	Valor(es)	Resultado Esperado	Resultado Obtenido
1	Seleccionar el módulo maestro, solicitudes y luego listar solicitudes abiertas.	N/A	Se listan la solicitudes abiertas se selecciona una y se agrega el seguimiento el cual es la descripción del procedimiento o solución dada.	Seguimiento creado satisfactoriamente.
Criterios de Aprobación del Caso de Prueba:				
Decisión de Aprobación del Caso de Prueba: Aprobó: <u> x </u> Fallo: <u> </u>				
Fecha de Aprobación del Caso de Prueba: 02/11/2016				

7.3.7 Entregables

Lista de Entregables de Pruebas (Véase el Anexo 1).

Tabla 67 Lista de Entregables de Pruebas

Entregables	Descripción
Plan de pruebas	Documento con la especificación de las pruebas y sus respectivos casos de prueba.

7.3.8 Resultados/Observaciones para el Ciclo de Prueba. Todas las pruebas han arrojado los resultados esperados por tanto cumplen con las especificaciones en las historias de usuario y casos de uso.

Aprobado Ciclo de Prueba por:

Cliente

Probador

7.3.9 Criterio para el Inicio y Fin del Plan de Pruebas

Criterios de Inicio

1. Funcionalidad de la aplicación terminada al 100%
2. Probador dedicado de lleno a las pruebas.

Criterios de Fin. En todos los casos de prueba deben arrojar un resultado satisfactorio o aprobado.

Criterios de Suspensión y Retomo de Actividades. Ninguno.

8. Implantación

8.1 Inicio del Funcionamiento del Sistema

Requisitos mínimos

Se necesita uno de los siguientes sistemas operativos con Internet

- ✓ Microsoft® Windows® 7 (ultimate) (www.microsoft.com/security).
- ✓ Microsoft® Windows® 7 (Home o Professional)
- ✓ Familia de Microsoft® Windows®.

Requisitos mínimos de hardware

Escenario	CPU necesaria	RAM necesaria
Cliente	INTEL CORE 2 QUAD*	2 GB**
Servidor	INTEL CORE 5 *	4 GB**

O la CPU mínima requerida para ejecutar el sistema operativo, lo que sea mayor.

O la RAM mínima necesaria para ejecutar el sistema operativo, lo que sea mayor.

Instalación del Sistema:

El sistema fue instalado de la siguiente manera:

El servidor web y de base de datos se encuentra en el centro de servidores y los clientes están en cada dependencia de la universidad de Pamplona, Sedes Villa del Rosario conectados mediante la red de comunicaciones.

Ya que clientes con diferentes usuarios, se ha controlado el acceso a las páginas y a los datos mediante el ingreso de usuario y contraseña, por lo tanto cada uno de los usuarios podrá realizar

únicamente las actividades que se les han sido permitidas después del análisis de permisos de usuarios.

8.2 Capacitación de Personal

El personal colaboró durante todo el proceso de realización del sistema, especialmente durante las pruebas, por lo tanto ya estaban familiarizados con el manejo del sistema y solamente fue necesario realizar al final una reunión con todos los usuarios para indicarles cuales serían sus funciones y responsabilidades en el manejo del sistema y por supuesto se realizaron algunas pruebas para que los usuarios estén totalmente seguros de las actividades que van a desarrollar y se puedan satisfacer todos sus dudas.

Además se entregó a cada empleado involucrado un manual de usuario para que se ayude en caso de tener alguna dificultad.

8.3 Evidencia fotográfica de capacitación



Anexos

Anexo 1.Manual del usuario

MANUAL PARA SOLICITUDES EN INFRAESTRUCTURA TECNOLÓGICA

✓ Ingreso Al Sistema

1.Para ingresar abra su navegador y escriba la dirección:

<http://infravilla.unipamplona.edu.co:8080/solicitudes>

Figura 44 Ingreso Al Sistema



Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

2.Darle Click en Iniciar Sesión

Figura 45 Darle Click en Iniciar Sesión

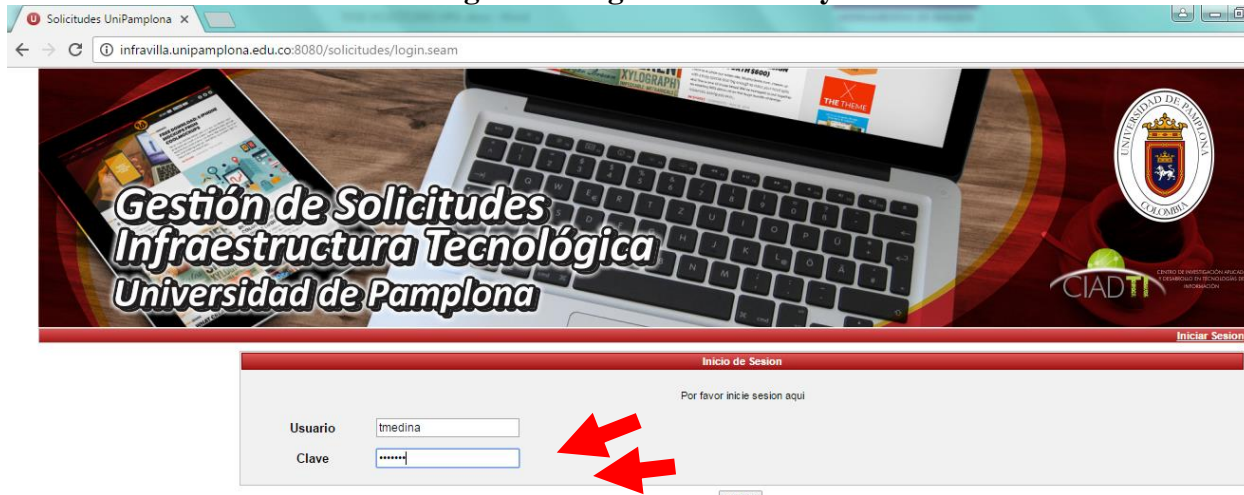


Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

3. Digite su Usuario y Clave (son requeridos), y darle click en “Entrar”

Figura 46 Digite su Usuario y Clave



Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación.

4. Página de Inicio* (Esta es la Interfaz para los USUARIOS; El menú de los Administradores del Sistemas poseen más ítem de acuerdo a las necesidades de cada uno)

Figura 47 Página de Inicio



Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

✓ Creación De Solicitudes

1. Para crear solicitudes, en el menú Procesos, ítem Sistemas, ítem [Crear Solicitud](#)

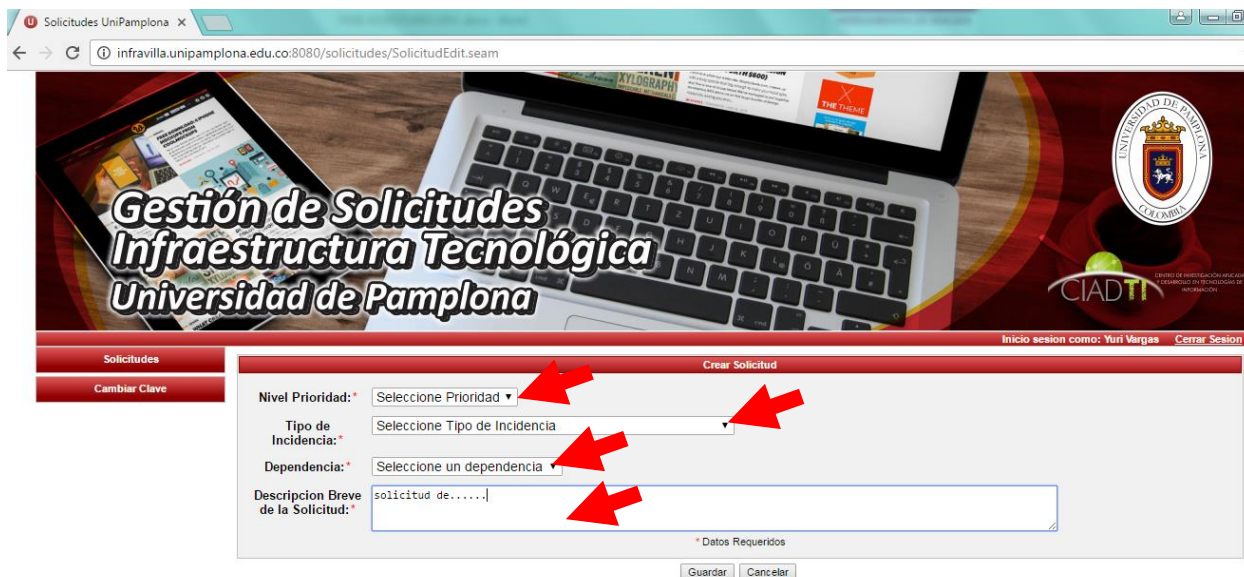
Figura 48 Crear Solicitudes



Todos Los Derechos Reservados TATIANAMEDINA
Fuente propia de la investigación.

2. Se deben llenar los espacio requeridos*, para finalizar dar Click en Guardar

Figura 49 Llenar Los Espacio Requeridos



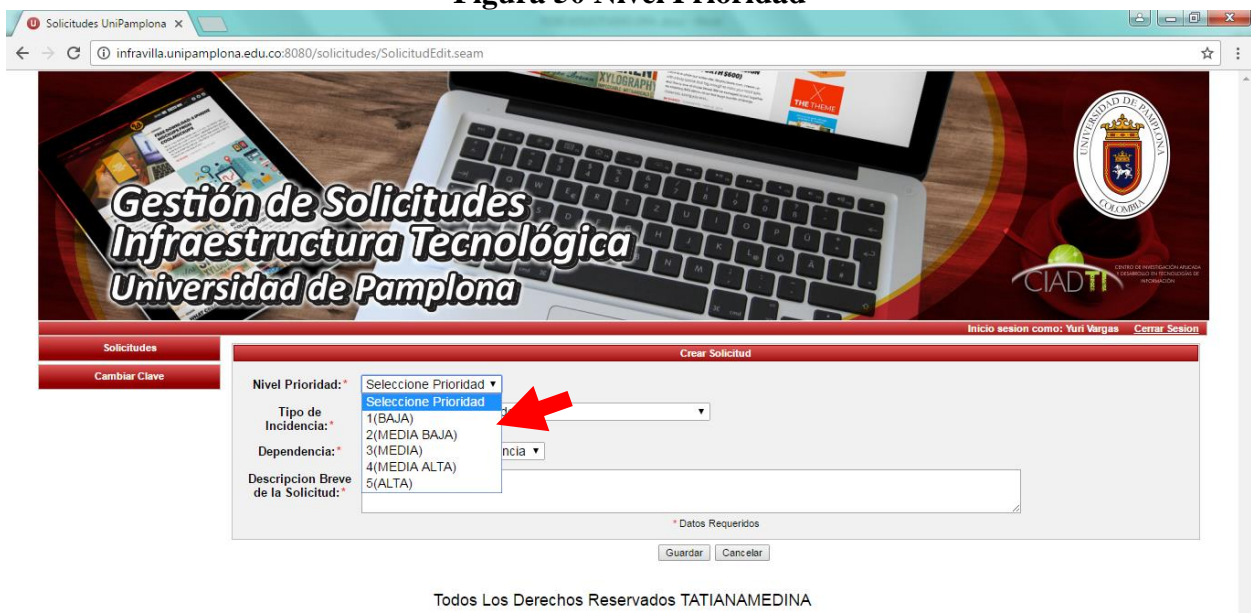
Todos Los Derechos Reservados TATIANAMEDINA
Fuente propia de la investigación

3. Nivel Prioridad, Clasifique y Seleccione la Prioridad de su Solicitud:

Tabla 68 Nivel Prioridad

1	BAJA
	MEDIA
2	BAJA
	MEDIA
3	MEDIA
	MEDIA
4	ALTA
	MEDIA
5	ALTA

Figura 50 Nivel Prioridad



Todos Los Derechos Reservados TATIANAMEDINA
Fuente propia de la investigación

4. Tipo de Incidencia, Clasifique y Seleccione el tipo de Incidencia de sus Solicitud:

Tabla 69 Tipo de Incidencia

Instalación Hardware
Instalación Software
Capacitación sobre Uso
Mantenimiento Correctivo
Mantenimiento Preventivo
Construcción Punto de Red
Traslado de Equipo
Información
Otros

Figura 51 Tipo de Incidencia

Solicitudes UniPamplona X

infravilla.unipamplona.edu.co:8080/solicitudes/SolicitudEdit.seam

**Gestión de Solicitudes
Infraestructura Tecnológica
Universidad de Pamplona**

Inicio sesión como: Yuri Vargas Cerrar Sesión

Solicitudes

Cambiar Clave

Crear Solicitud

Nivel Prioridad:* Seleccione Prioridad

Tipo de Incidencia:* Seleccione Tipo de Incidencia

Dependencia:* Cambio de Equipos

Descripción Breve de la Solicitud:*

Instalación Hardware

Mantenimiento Correctivo De Equipo De Cómputo

Mantenimiento Preventivo De Equipos De Cómputo

Otros

Registro De Mac De Equipos Móviles Y De Cómputo

Restablecimiento De Contraseña De Correo Institucional

Traslado De Equipos De Cómputo

Cancelar

TATIANAMEDINA

Fuente propia de la investigación

5.Descripción Breve de la Solicitud: Realizar una Breve y Concisa descripción de la Solicitud

6.

Figura 52 Descripción Breve de la Solicitud

Solicitudes UniPamplona X

infravilla.unipamplona.edu.co:8080/solicitudes/SolicitudEdit.seam

**Gestión de Solicitudes
Infraestructura Tecnológica
Universidad de Pamplona**

Inicio sesión como: Yuri Vargas Cerrar Sesión

Solicitudes

Cambiar Clave

Crear Solicitud

Nivel Prioridad:* 4(MEDIA ALTA)

Tipo de Incidencia:* Creación De Punto De Red

Dependencia:* Edufísica

Descripción Breve de la Solicitud:* solicitud de punto de red en la oficina de educación física

* Datos Requeridos

Guardar Cancelar

Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

7. Mensaje Exitoso de la Creación de la Solicitud **Successfully created** (Creado con Éxito)

Figura 53 Mensaje Exitoso de la Creación de la Solicitud



Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

8. Al darle Guardar el sistema lo traslada hacia la Interfaz donde podrá ver todas las Solicitudes Pendientes

Figura 54 Al darle Guardar



Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

✓ **PARA VER LAS SOLICITUDES PENDIENTES**

1. Para Listar las Solicitudes Pendientes, en el menú Solicitudes, ítem [Lista Solicitud](#)

Figura 55 Listar las Solicitudes Pendientes

Inicio sesión como: Yuri Vargas Cerrar Sesión

Busqueda Solicitudes

Estado: Seleccione Estado

* Datos Requeridos

Buscar

Listado de Solicitudes(0)

No Hay Ninguna Solicitud.

Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación

- Lista de solicitudes dependiendo del estado ya sea abierto o cerrado.

Figura 56 Lista de solicitudes

Inicio sesión como: Yuri Vargas Cerrar Sesión

Busqueda Solicitudes

Estado: Abierto

* Datos Requeridos

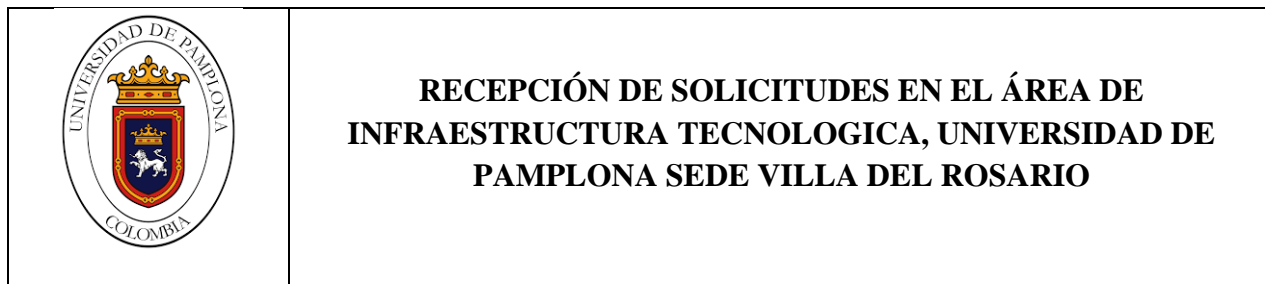
Buscar

Listado de Solicitudes(4)

No	Fecha	Dependencia	Usuario	Estado	Prioridad	Incidencia	Descripcion	Calificacion	Opciones
16	10/10/2016	Talento Humano	Yuri Vargas	Abierto	1(BAJA)	Instalación Hardware	INSTALAR PORTATIL		Ver
17	10/10/2016	Laboratorio de Geología	Yuri Vargas	Abierto	3(MEDIA)	Creación De Punto De Red	hoy		Ver
18	10/10/2016	Laboratorio de Informatica	Yuri Vargas	Abierto	1(BAJA)	Creación De Punto De Red	yaaaaaaaa		Ver
35	21/11/2016	Consultorio Jurídico	Yuri Vargas	Abierto	3(MEDIA)	Cambio de Equipos	cambiar equipo de consultorio jurídico.		Ver

Todos Los Derechos Reservados TATIANAMEDINA

Fuente propia de la investigación



RECEPCIÓN DE SOLICITUDES EN EL ÁREA DE INFRAESTRUCTURA TECNOLÓGICA, UNIVERSIDAD DE PAMPLONA SEDE VILLA DEL ROSARIO

1. OBJETIVO

Gestionar las solicitudes del personal administrativo de la universidad al el área de infraestructura tecnológica.

2. ALCANCE

Solicitudes direccionadas al área de infraestructura tecnológica en cuanto a adquisición e instalación de software, adquisición y configuración de equipos nuevos y los mantenimientos correctivos.

3. POLÍTICA

- ◆ Las solicitudes de los usuarios *se le debe hacer seguimiento cuando no se pueda solucionar.*
- ◆ No se atenderán solicitudes que no sean diligenciadas en la plataforma **Gestión de Solicitudes Infraestructura Tecnológica**
<http://infravilla.unipamplona.edu.co:8080/solicitudes/SolicitudList.seam>
- ◆ Las solicitudes pueden ser realizadas por los usuarios administrativos de la Universidad de Pamplona sede Villa del Rosario.

4. MANTENIMIENTO

Director Administrativo del área de Infraestructura Tecnológica.

5. REVISION


Director Administrativo del área de Infraestructura Tecnológica.

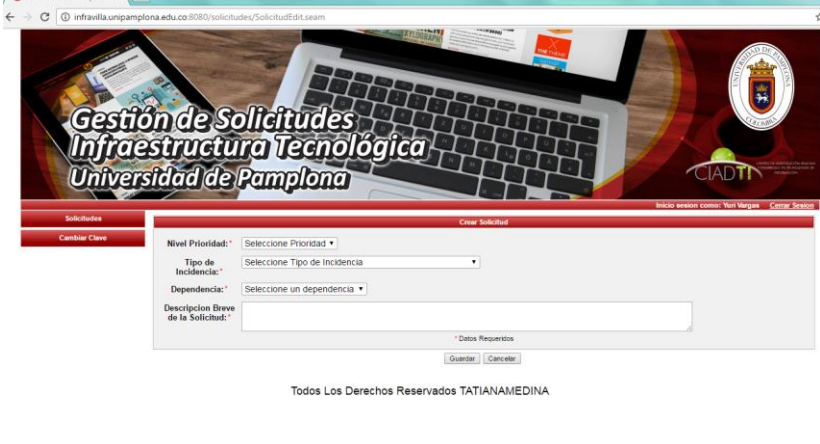
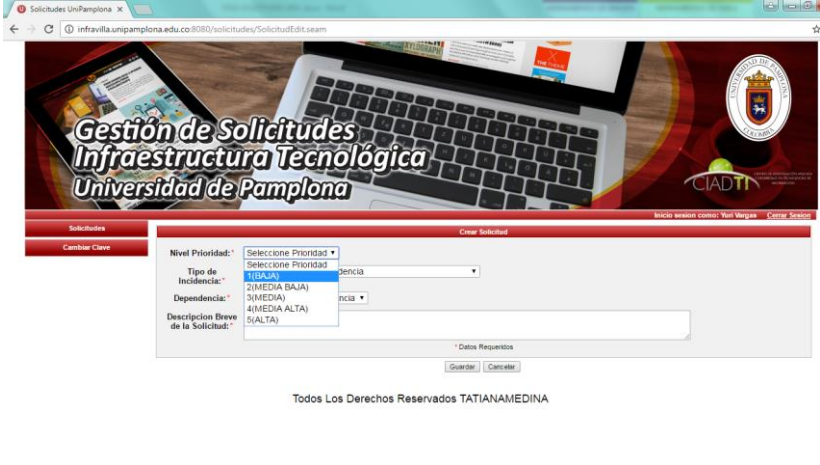
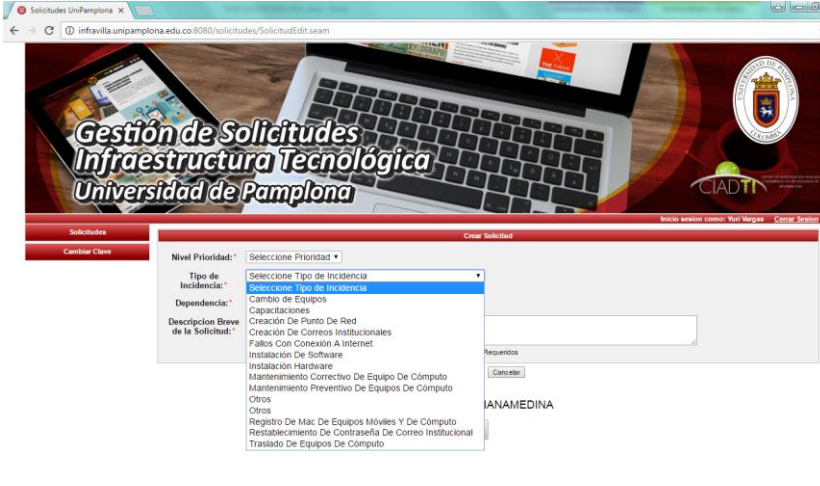
6. RESPONSABLE

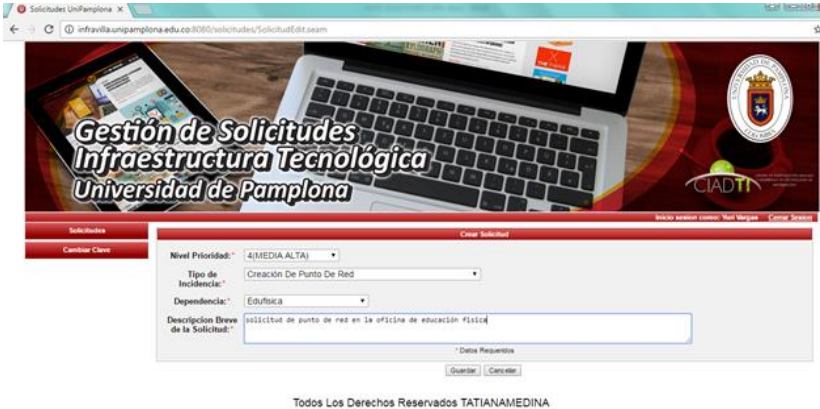
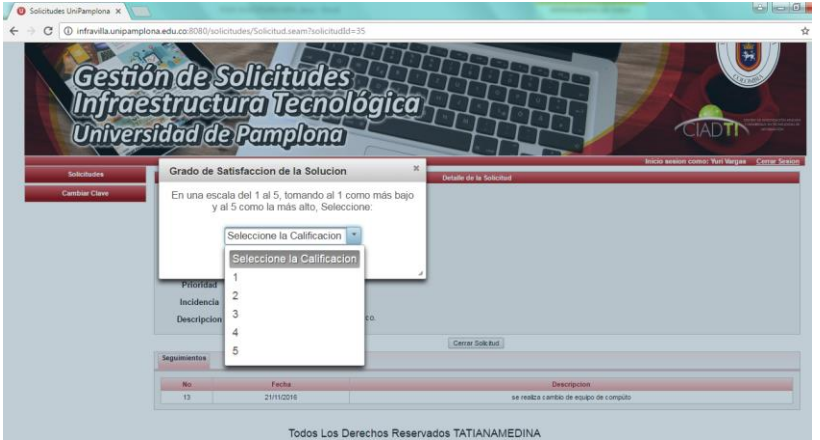
Director Administrativo del área de infraestructura tecnológica.

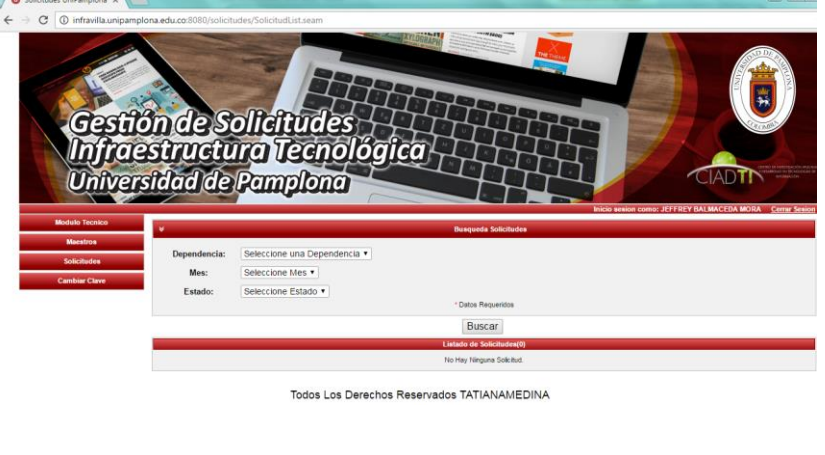

7. DESARROLLO

7.1 SOLICITUDES

PASO	RESPONSABLE	ACTIVIDAD
1.	Usuarios de Equipos Terminales	<p>Ingresar al Gestión de Solicitudes Infraestructura Tecnológica</p> <p>http://infravilla.unipamplona.edu.co:8080/solicitudes/SolicitudLlist.seam con la clave y usuario asignado (solicitar al área de servidores).</p> 
2.	Usuarios de Equipos	Ingresar a proceso – sistemas – crear solicitud.

	<p>Terminales</p>	
<p>3.</p>	<p>Usuarios de Equipos Terminales</p>	<p>Seleccionar el nivel de prioridad según sea el caso de la solicitud.</p> 
<p>4.</p>	<p>Usuarios de Equipos Terminales</p>	<p>Seleccionar el tipo de incidencia o solicitud.</p> 

<p>5.</p>	<p>Usuarios de Equipos Terminales</p>	<p>Realizar una Descripción Breve de la Solicitud</p> 
<p>6.</p>	<p>Usuarios de Equipos Terminales</p>	<p>Revisar el cierre de la solicitud y el grado de satisfacción tras la finalización de la misma.</p> 
<p>7.</p>	<p>Director Administrativo del área de infraestructura tecnológica.</p>	<p>Revisar las solicitudes que los usuarios diligencian en el aplicativo de Gestión de Solicitudes Infraestructura Tecnológica.</p> <p>http://infravilla.unipamplona.edu.co:8080/solicitudes</p>

		
<p>8.</p>	<p>Director Administrativo del área de infraestructura tecnológica.</p>	<p>Ingresar al aplicativo de Gestión de Solicitudes Infraestructura Tecnológica http://infravilla.unipamplona.edu.co:8080/solicitudes y agregar un seguimiento a la solicitud junto con un plan de acción y una fecha de entrega.</p> 
<p>9.</p>	<p>Director Administrativo del área de infraestructura tecnológica.</p>	<p>Revisa los informes estadísticos de las solicitudes.</p>

		<p>The screenshot shows a web interface for 'Solicitudes Unipamplona'. The main content area displays 'Estadística de Solicitudes' for the year 2016. There are two bar charts:</p> <ul style="list-style-type: none"> Índice de Satisfacción en Solicitudes: A bar chart showing satisfaction scores from 0.00 to 5.00 across months. Data points are visible for September (approx. 2.5), October (approx. 4.0), and November (approx. 4.5). Solicitudes Abiertas/Cerradas: A stacked bar chart showing the number of open (red) and closed (green) requests from August to December. Open requests peak in October (approx. 15) and November (approx. 12). Closed requests are also present in these months.
<p>10.</p>	<p>Director Administrativo del área de infraestructura tecnológica.</p>	<p>Ya sea por dependencia, incidencia, solicitudes abiertas y cerradas o por satisfacción.</p> <p>The screenshot shows a web interface for 'Solicitudes Unipamplona' with two stacked bar charts:</p> <ul style="list-style-type: none"> Solicitudes por Tipo de Incidencia: A stacked bar chart showing the number of requests by type (e.g., Incidencia, Problemas, etc.) across months. Data points are visible for September and October. Solicitudes por Dependencia: A stacked bar chart showing the number of requests by department (e.g., Infraestructura, TI, etc.) across months. Data points are visible for September and October.

10. Conclusiones

La Aplicación Web SOLICITUDES entregada, es una herramienta de fortalecimiento del proceso de Gestión y Control de solicitudes, en el que se mejora el proceso de atención a las solicitudes, minimizando el tiempo y facilitándolo al momento de diligenciar la solicitud por medio del aplicativo y teniendo registro de las acciones aplicadas para la solución de cada solicitud.

Al culminar esta investigación de tipo aplicada, se cumplió minuciosamente al final del sprint con cada uno de los requerimientos reportados en la etapa de registro de historias de usuario, se denota que con los productos obtenidos en el desarrollo del proyecto se cumplieron a cabalidad con los objetivos planteados en el inicio del proyecto.

El uso de metodologías o procedimientos estándares para el análisis, diseño, implementación y prueba de la misma, que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba permitió la obtención de una aplicación eficiente y eficaz.

La captura correcta de los requerimientos a partir de las entrevistas con Usuario, y su eficiente análisis permitió la satisfacción de las necesidades que se presentaron.

La aplicación fue desarrollada en lenguajes java, html5 usando el framework de Jboss Seam, con el gestor base de datos PostgreSQL, aplicando la arquitectura de software Modelo Vista Controlador para el desarrollo del sistema de control y usando metodología Scrum para el desarrollo del software.

Recomendaciones

El acceso a la aplicación se realizará solo para usuarios del área administrativa de la Universidad de Pamplona, sede Villa del Rosario los cuales estén activos en planta, estos usuarios serán registrados por el ingeniero Jeffrey Balmaceda persona encargada del area.

El acceso a la aplicación se realizará mediante un navegador WEB. Cabe destacar que esta aplicación es compatible con “Mozilla Firefox” y “Google Chrome”.

Se recomienda una reestructuración de los procesos del negocio a fin ampliar las funcionalidades de la aplicación y solucionar algunas necesidades que se presenten.

Referencias Bibliográficas

10. Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación, 19(4), 413–418.
11. Andrea, J., & Palencia, R. (2016). Diseño de una aplicación web responsive para control en la entrega de facultad de ingenierías y arquitectura departamento de ingenierías eléctrica, electrónica, sistemas e ingeniería de sistemas. Universidad de pamplona.
12. El, P., Integral, D., & Sinaloa, U. (2009). Instituto politécnico nacional.
13. Bases de Datos, Recuperado de:
<https://si.ua.es/es/documentos/documentacion/office/access/teoria-de-bases-de-datos.pdf>
14. Chiavenato Idalberto. (2004), Introducción a la Teoría General de la Administración, Séptima Edición, McGraw-Hill Interamericana, Pág. 52.
15. Definición de Control, Recuperado de: <http://dle.rae.es/?id=AeYZ09V>
16. Definición de Eficiencia, Recuperado de:
<http://www.promonegocios.net/administracion/definicion-eficiencia.html>
17. Documentación sobre jboss, Recuperado de:
<https://www.alferez.es/documentos/Jboss.pdf>
18. Documentación sobre jboss, Recuperado de:
<https://www.codejobs.biz/es/blog/2013/09/30/que-es-jboss>
19. (El, Integral, & Sinaloa, 2009). Diccionario de la lengua española, Vigésima segunda edición.
20. El, P., Integral, D., & Sinaloa, U. (2009). Instituto politécnico nacional.

21. SSTecnología, Figura 2. Metodología Scrum; **Error! Marcador no definido.** Recuperado de <http://www.sstecnologia.com.br/>
22. Figura 1: Ingeniería Del Software Recuperado de: <http://es.slideshare.net/danielmerchan2/13-ingenieria-del-software-16397494>
23. Fuente <http://www.sstecnologia.com.br>
24. Figura 4. Diagrama De Casos De Uso, Recuperado de: <http://profesores.fib.unam.mx/carlos/aydoo/usecase.html>
25. Figura 5. Diagrama De Casos de clases, Recuperado de: <http://profesores.fib.unam.mx/carlos/aydoo/clases.html>
26. Figura 6. Diagrama De actividades, Recuperado de: http://datateca.unad.edu.co/contenidos/200609/exeuml/leccin_21_diagramas_de_actividades.html
27. Figura 8. Diagrama De secuencias, Recuperado de: http://www.sparxsystems.com.ar/resources/tutorial/uml2_sequencediagram.html
28. Figura 9. Arquitectura de jboss Seam, Recuperado de: <http://osl2.uca.es/wikiCE/index.php/Seam>
29. Figura 10. Modelo de 3 capas, Recuperado de: <http://oness.sourceforge.net/proyecto/html/ch03s02.html>
30. Figura 10. Modelo de 3 capas, Recuperado de: FRECHOSO, Miguel Ángel. La arquitectura modelo - vista – controlador como base de la plataforma de servicio. Madrid: Cubenube, 2011.
31. Haider Miranda (2016), APLICACIÓN MÓVIL PARA DIAGNOSTICAR LOS FACTORES DE DESERCIÓN ESTUDIANTIL EN LA UNIVERSIDAD DE

PAMPLONA A TRAVÉS DE MÉTODOS DE RECUPERACIÓN DE INFORMACIÓN Y NUEVAS TECNOLOGÍAS.

32. Koontz Harold y Weihrich Heinz. (2004), Administración Un Perspectiva Global, 12a. Edición, de, McGraw-Hill Interamericana, Pág. 14.
33. (MONSALVE MAYER, & ORTEGA RAFAEL, 2012), MEJORAMIENTO DEL SITIO WEB DEL EGRESADO DE LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER.
34. Modelo entidad – relación, recuperado de:
<http://cursos.aiu.edu/Base%20de%20Datos/pdf/Tema%203.pdf>
35. Recuperado de: <https://www.infoq.com/articles/jboss-seam>.
36. Recuperado de:
http://fcasua.contad.unam.mx/apuntes/interiores/docs/98/4/informatica_4.pdf
37. El, P., Integral, D., & Sinaloa, U. (2009). Instituto politécnico nacional.
38. RICARDO, G. C. J., & XAVIER, S. R. E. (2012). Diseño E Implementación De Un Sistema Web Utilizando Java Y Empleando La Metodología Scrum, Para La Gestión Administrativa De La Junta Parroquial De Guaytacama Perteneciente Al Cantón Latacunga De La Provincia De Cotopaxi, 115.
39. Robbins Stephen y Coulter Mary. (2005), Administración, Octava Edición, Pearson Educación, Págs. 7.
40. Teorías de la Administración, de Oliveira Da Silva Reinaldo, International Thomson Editores, S.A. de C.V., 2002, Pág. 20.
41. Recuperado de:
<http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4278/87>

42. Recuperado de: <https://Jonathan08cruz.Wordpress.Com/2010/05/17/Analisisdisenooo/>

43. Visual Paradigm for UML, Recuperado de:

[http://www.dpi.ufv.br/projetos/geoprofile/tutoriais/Visual Paradigm for UML Tutorial_english.pdf](http://www.dpi.ufv.br/projetos/geoprofile/tutoriais/Visual_Paradigm_for_UML_Tutorial_english.pdf)

44. *Figura 9: Modelo De Tres Capas. SÁNCHEZ GONZÁLEZ, Carlos. ONess: un proyecto open source para el negocio textil mayorista desarrollado con tecnologías open source innovadoras. Coruña: Universidad de Coruña, 2004.*

45. Chiavenato Idalberto (2004), *Introducción a la Teoría General de la Administración* McGraw-Hill Interamericana, Pág. 52, Séptima Edición.

46. Koontz Harold y Weihrich Heinz, (2004) *Administración, Un Perspectiva Global*, de, McGraw-Hill Interamericana, 12a. Edición, Pág. 14.

47. Robbins Stephen y Coulter Mary (2005), *Administración*, Pearson Educación, Octava Edición, Págs. 7

48. Oliveira Da Silva Reinald (2002), *Teorías de la Administración*, International Thomson Editores S.A. de C.V. Pág. 20.