



UNIVERSIDAD
PRIVADA
DEL NORTE

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE
INCIDENCIAS SOCIALES EN LA
MUNICIPALIDAD DISTRITAL
DE FLORENCIA DE MORA”

Tesis para optar el título profesional de:
Ingeniero de Sistemas Computacionales

Autor(es):

Br. Edwin Roy Neyra Herrera
Br. Jorge Luis Angulo Pretel

Asesor:

Mg. José Alberto Gómez Ávila

**Trujillo – Perú
2017**

APROBACIÓN DE LA TESIS

El asesor y los miembros del jurado evaluador asignados, **APRUEBAN** la tesis desarrollada por los Bachilleres **Edwin R. Neyra Herrera y Jorge L. Angulo Pretel**, denominada:

“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE INCIDENCIAS SOCIALES EN LA
MUNICIPALIDAD DISTRITAL DE FLORENCIA DE MORA”

Ing. José Alberto Gómez Ávila
ASESOR

Ing. Laín Jardiel Cárdenas Escalante
JURADO
PRESIDENTE

Ing. Geancarlo Leiva Via
JURADO

Ing. Rolando Javier Berrú Beltrán
JURADO

DEDICATORIA

A:

Dios, por darnos la oportunidad de vivir y por estar con nosotros en cada paso que damos.

Nuestros padres, por darnos la vida, querernos mucho, creer en nosotros y porque siempre nos apoyaron. Gracias por darnos una carrera para nuestro futuro, todo esto se lo debemos a ustedes.

Nuestros hermanos, por estar con nosotros y apoyarnos siempre, los queremos mucho.

Nuestros hijos, que desde que nacieron son el motor de nuestras actividades diarias.

AGRADECIMIENTO

A Dios por protegernos y darnos la oportunidad de demostrar nuestros conocimientos en beneficio de nosotros, de nuestra familia y de los demás seres vivos.

Al Ing. José Alberto Gómez Ávila, asesor de nuestro proyecto de investigación, gracias por su conocimiento y tiempo que nos ha ofrecido a lo largo del planteamiento del proyecto. Ha inculcado en cada uno de nosotros el sentido de responsabilidad y rigor académico que son necesarios para completar nuestra formación profesional.

Al Ing. Juan Orlando Salazar Campos, director de carrera en nuestra universidad de estudios, por darnos la oportunidad de implementar la idea en la que se basa nuestro proyecto de investigación, y por ofrecernos su apoyo en cada necesidad que se nos ha presentado durante este desarrollo.

A las instituciones donde laboramos, por el apoyo en disponibilidad de tiempo para llevar acabo nuestras tareas académicas.

RESUMEN

En la actualidad, en Perú muy pocos son los gobiernos locales que cuentan con un sistema informático móvil/web para la sociedad civil, quizás por razones como costo, tiempo y recursos no logran implementar tan importantes servicios móviles ahora que casi todo ser humano a muy temprana edad ya cuenta con un dispositivo móvil. Además, que los gobiernos locales independientemente de su nivel económico y social sufren problemas sociales como secuestros, robos, sicariato, etc. y problemas de tipo ambiental como deterioro de sus pistas, inundaciones, acumulación de basura, etc.

Es por esto que se realizó el presente proyecto informático para la sociedad civil; que en su primera versión tendrá un alcance sólo a la Municipalidad Distrital de Florencia de Mora. Este proyecto implica un despliegue tecnológico tanto en la parte móvil como en la parte web que servirá de comunicación entre el vecino del distrito y su municipalidad. Y se espera que esta comunicación sea de manera efectiva y eficaz que es el objetivo del presente proyecto.

Los actores del presente proyecto –vecino, municipalidad y equipo técnico- deberán de darle el mejor uso al presente aplicativo móvil/web. Cada uno de ellos pondrá lo mejor de sí desde su propia modalidad de uso y porque no, mejorar constantemente los servicios que brinda. En este sentido el aplicativo será un software de bajo acoplamiento ya que no dependerá de otro software para su buen funcionamiento y estará en la capacidad de poder crecer en funcionalidades a medida que aumente las necesidades de la población.

La aplicación móvil/web tendrá inicialmente cobertura sólo en el Distrito de Florencia de Mora, paralelamente se hará una difusión en las otras municipalidades de la Provincia de Trujillo, generando así conciencia en los pobladores para que ellos pongan de su parte para el mejoramiento de sus respectivas localidades.

Si los gobiernos locales buscan mejorar la calidad de vida de sus pobladores, la calidad de su atención en las diversas gestiones propias de la municipalidad, se hace muy necesario contar con este tipo de aplicación móvil/web. El presente proyecto garantiza esta calidad basada en el estudio de mercado y en la experiencia de este tipo de aplicativos en otros

distritos especialmente de Lima, donde se encuentran ya implementados generando mayor bienestar para la población.

El propósito general de esta alternativa es favorecer inicialmente el desarrollo tecnológico, social, económico y urbano de la población del Distrito de Florencia de Mora. Además, el presente proyecto busca que el producto sea una herramienta útil, segura, amena e interactiva. Además, que los usuarios se sientan identificados y comprometidos con las diferentes funcionalidades que ofrece como son gestionar alertas, consultar novedades, llamadas de emergencia entre otros. Es por este motivo que el sistema informático para la sociedad civil se proyecta a ser la herramienta que todos los pobladores de una comuna necesitan para poder denunciar y comunicar a sus autoridades locales, hechos que atenten contra las buenas costumbres.

ABSTRACT

At present, very few local governments in Peru have a mobile / web computing system for civil society, perhaps reasons such as cost, time and resources fail to implement such important mobile services now that almost every human being at an early age already has a mobile device. In addition, local governments regardless of their economic and social level social problems such as kidnappings, robberies, hysteria, etc. and environmental problems such as the deterioration of their tracks, flooding, accumulation of garbage, etc.

That is why the present computer project was carried out for civil society; which in its first version will have a scope only to the Municipal District of Florence de Mora. This project involves a technological deployment both in the mobile part and in the web part that will serve as communication between the neighbor of the district and its municipality. And this communication is expected to be effective and effective, which is the objective of this project.

The responsables in the present project - neighbor, municipality and technical team - should give the best use to the present mobile / web application. Each one of them will put the best of himself from his own mode of use and why not, constantly improve the services he provides. In this sense the application will be low-coupling software because it will not depend on other software for its smooth operation and will be able to grow in functionality as the needs of the population increase.

The mobile / web application will initially have coverage only in the District of Florence de Mora, in parallel will be made a diffusion in the other municipalities of the Province of Trujillo, thus generating awareness in the settlers so that they do their part to improve their respective locations.

If local governments search to improve the quality of life of their residents, the quality of their attention in the various management of the municipality, it is very necessary to have this type of mobile / web application. The present project guarantees this quality based on the market study and the experience of this type of applications in other districts especially of Lima, where they are already implemented generating more welfare for the population.

The general purpose of this alternative is to initially favor the technological, social, economic and urban development of the population of the District of Florence de Mora. In addition, this project seeks to make the product a useful, safe, entertaining and interactive tool. In addition, users feel identified and committed to the different functionalities offered such as managing alerts, check news, emergency calls and more. It is for this reason that the computer system for civil society is projected to be the tool that all the inhabitants of a commune need to be able to denounce and communicate to their local authorities, facts that violate the good customs.

ÍNDICE DE CONTENIDOS

CAPÍTULO 1.	INTRODUCCIÓN.....	1
1.1.	Realidad problemática	1
1.2.	Formulación del problema	2
1.3.	Justificación.....	2
1.4.	Limitaciones.....	2
1.5.	Objetivos	3
1.5.1.	Objetivo General.....	3
1.5.2.	Objetivos Específicos	3
CAPÍTULO 2.	MARCO TEÓRICO.....	4
2.1.	Antecedentes.....	4
2.2.	Bases Teóricas.....	6
2.3.	Definición de términos básicos	15
CAPÍTULO 3.	HIPÓTESIS	16
3.1.	Formulación de la hipótesis	16
3.2.	Operacionalización de variables	16
CAPÍTULO 4.	PROPUESTA DE APLICACIÓN PROFESIONAL.....	17
4.1.	Modelo del Negocio	17
4.1.1.	Propuesta de Valor.....	17
4.1.2.	Ingresos.....	18
4.1.3.	Estructura de Costos.	18
4.1.4.	Canales de Distribución.....	18
4.1.5.	Relación con el Cliente.....	19
4.1.6.	Segmento de Mercado.	19
4.1.7.	Alianzas.....	19
4.1.8.	Actividades Clave.	19

4.1.9.	Recursos Clave.	19
4.2.	Plan de Desarrollo de Software	20
4.2.1.	Introducción.....	20
4.2.2.	Vista General del Proyecto	21
4.2.3.	Organización del Proyecto.....	26
4.2.4.	Gestión del Proceso	27
4.3.	Modelo de Casos de Uso del Negocio.	31
4.3.1.	Objetivos del Negocio.....	31
4.3.2.	Actores del Negocio.....	31
4.3.3.	Casos de Uso de Negocio y Especificación.....	32
4.3.4.	Relación Casos de Uso de Negocio y Objetivo de Negocio.	34
4.3.5.	Modelo de Caso de Uso del Negocio.....	35
4.4.	Modelo de Objetos del Negocio.	35
4.4.1.	Trabajadores del Negocio.....	35
4.4.2.	Entidades del Negocio	36
4.4.3.	Relación Entidad de Negocio con Trabajador de Negocio.	36
4.4.4.	Realizaciones de Casos de Uso de Negocio.	37
4.4.5.	Diagrama de Entidades de Negocio.	39
4.5.	Glosario.....	39
4.6.	Visión.....	40
4.6.1.	Introducción.....	40
4.6.2.	Posicionamiento	41
4.6.3.	Descripción de Stakeholders (Participantes del Proyecto y Usuarios)	43
4.6.4.	Descripción del producto.	44
4.7.	Modelo de Casos de Uso de Sistema.....	46
4.7.1.	Actores de Sistema.....	46
4.7.2.	Diagrama de Caso de Uso de Sistema.	47
4.8.	Especificaciones Adicionales.....	50

4.9.	Prototipos de Interfaces de Usuario.....	51
4.10.	Diagramas de Secuencia.....	64
4.11.	Diagramas de Estado.....	66
4.12.	Modelo de Análisis y Diseño	68
4.12.1.	Capas de la Arquitectura.....	68
4.12.2.	Diagramas de Clases de Análisis del Sistema.	68
4.12.3.	Patrones de Diseño.....	68
4.12.4.	Diagrama de Clases de Diseño.....	70
4.13.	Modelo de Datos.....	71
4.14.	Modelo de Despliegue.....	72
4.14.1.	Capas con sus Componentes	72
4.14.2.	Diagrama de Componentes	72
4.14.3.	Diagrama de Despliegue.....	73
4.15.	Casos de Prueba.....	73
4.15.1.	Pruebas de Caja Negra.....	73
4.15.2.	Pruebas Unitarias.....	76
4.16.	Gestión de Riesgos.....	98
CAPÍTULO 5. MATERIALES Y MÉTODOS.....		102
5.1.	Tipo de diseño de investigación.....	102
5.2.	Material de estudio.....	102
5.2.1.	Unidad de estudio.....	102
5.2.2.	Población.....	102
5.2.3.	Muestra.....	102
5.3.	Técnicas, procedimientos e instrumentos.....	103
5.3.1.	Para recolectar datos.....	103
5.3.2.	Para procesar datos.....	104
CAPÍTULO 6. RESULTADOS		105
CAPÍTULO 7. DISCUSIÓN		110

CONCLUSIONES	111
RECOMENDACIONES	112
REFERENCIAS	113
ANEXOS.....	115

ÍNDICE DE TABLAS

Tabla No. 1: <i>Ventajas e Inconvenientes de la aplicación móvil nativa</i>	7
Tabla No. 2: <i>Ventajas e Inconvenientes de la aplicación móvil web</i>	8
Tabla No. 3: <i>Ventajas e Inconvenientes de la aplicación móvil híbrida</i>	9
Tabla No. 4: <i>Cuadro detallado de la Operacionalización de las Variables. Elaboración: Propia</i>	16
Tabla No. 5: <i>Roles y responsabilidades del equipo de trabajo</i>	27
Tabla No. 6: <i>Estimación del proyecto</i>	27
Tabla No. 7: <i>Plan de fases del Proyecto</i>	27
Tabla No. 8: <i>Descripción de hitos del proyecto</i>	28
Tabla No. 9: <i>Calendario del proyecto – Fase inicio</i>	29
Tabla No. 10: <i>Calendario del proyecto – Elaboración</i>	30
Tabla No. 11: <i>Sentencias que definen el problema</i>	42
Tabla No. 12: <i>Sentencias que definen la posición del producto</i>	42
Tabla No. 13: <i>Stakeholders del sistema</i>	43
Tabla No. 14: <i>Usuarios del Sistema</i>	44
Tabla No. 15: <i>Características del producto final</i>	45
Tabla No. 16: <i>Costo y precio de hardware, software y servicios del proyecto</i>	46
Tabla No. 17: <i>FURPS del sistema informático</i>	50
Tabla No. 18: <i>Pruebas de caja negra</i>	76
Tabla No. 19: <i>Lista de riesgos del proyecto. (Elaboración equipo de trabajo)</i>	100
Tabla No. 20: <i>Respuestas a los riesgos del proyecto (Elaboración equipo de trabajo)</i> ...	100
Tabla No. 21: <i>Técnicas, procedimientos e instrumentos de recolección de información</i>	104
Tabla No. 22: <i>Técnicas, procedimientos e instrumentos de procesamiento de información. (Elaboración propia)</i>	104

ÍNDICE DE ILUSTRACIONES Y GRÁFICOS

Ilustración No. 1: <i>Objetivos del negocio</i>	31
Ilustración No. 2: <i>Actor del Negocio</i>	31
Ilustración No. 3: <i>Diagrama de Caso de Uso Registrar Incidencia</i>	32
Ilustración No. 4: <i>Diagrama de Caso de Uso Derivar Incidencia</i>	32
Ilustración No. 5: <i>Diagrama de Caso de Uso Generar Reportes</i>	33
Ilustración No. 6: <i>Diagrama de relación entre caso de uso y objetivo del negocio</i>	34
Ilustración No. 7: <i>Diagrama de caso de uso de negocio</i>	35
Ilustración No. 8: <i>Diagrama de trabajadores de negocio</i>	35
Ilustración No. 9: <i>Diagrama de entidades de negocio</i>	36
Ilustración No. 10: <i>Diagrama de relación de entidad y trabajador de negocio</i>	36
Ilustración No. 11: <i>Diagrama de caso de uso de Negocio</i>	37
Ilustración No. 12: <i>Diagrama de actividades – Registro de incidencia</i>	37
Ilustración No. 13: <i>Diagrama de actividades – Derivar incidencia</i>	38
Ilustración No. 14: <i>Diagrama de actividades – Generar reporte</i>	38
Ilustración No. 15: <i>Diagrama de entidades del negocio</i>	39
Ilustración No. 16: <i>Diagrama de entidades del negocio</i>	46
Ilustración No. 17: <i>Diagrama casos de uso de sistema</i>	47
Ilustración No. 18: <i>Pantalla principal del aplicativo móvil</i>	51
Ilustración No. 19: <i>Muestra tipo de incidencia a seleccionar</i>	52
Ilustración No. 20: Muestra tipo de incidencia elegida, la cual puede	52
Ilustración No. 21: <i>Muestra la opción mapa para elegir ubicación de la incidencia</i>	53
Ilustración No. 22: <i>Muestra las alternativas de la opción foto</i>	53
Ilustración No. 23: <i>Muestra agregar datos de la opción inmueble</i>	54
Ilustración No. 24: <i>Muestra agregar datos de la opción persona</i>	54
Ilustración No. 25: <i>Muestra agregar datos de la opción vehículo</i>	55
Ilustración No. 26: <i>Ingresar nombre y teléfono (opcionalmente) de la persona que reporta el hecho y obligatoriamente ingresar información de la incidencia. Luego enviar datos</i> ...55	55
Ilustración No. 27: <i>Enviando datos de la incidencia al sistema web</i>	56
Ilustración No. 28: <i>Datos enviados correctamente al sistema web. A la vez retorna un código de la incidencia para cualquier consulta posterior de parte de la persona que reportó la incidencia</i>	56
Ilustración No. 29: <i>Muestra todas las incidencias enviadas desde un dispositivo móvil</i> ...57	57

Ilustración No. 30: <i>Muestra todas las incidencias enviadas desde la página web de la municipalidad de Florencia de Mora.</i>	58
Ilustración No. 31: <i>Atiende la incidencia.</i>	59
Ilustración No. 32: <i>La incidencia es derivada a una oficina interna de la municipalidad o a un órgano externo a ella.</i>	60
Ilustración No. 33: <i>Se puede hacer una descripción breve de la incidencia antes de derivarse.</i>	61
Ilustración No. 34: <i>Muestra el ícono para generar reporte en Word de la incidencia.</i>	62
Ilustración No. 35: <i>Muestra el reporte de la incidencia con los campos encerrados en círculos de color rojo mostrados en la Pantalla 16.</i>	63
Ilustración No. 36: <i>Diagrama de secuencia general.</i>	64
Ilustración No. 37: <i>Diagrama de secuencia derivar incidencia.</i>	65
Ilustración No. 38: <i>Diagrama de secuencia generar reporte.</i>	65
Ilustración No. 39: <i>Diagrama de estado de incidencia social.</i>	66
Ilustración No. 40: <i>Diagrama de estado de usuario.</i>	66
Ilustración No. 41: <i>Diagrama de estado de entidad externa.</i>	67
Ilustración No. 42: <i>Diagrama de estado de jefe de área.</i>	67
Ilustración No. 43: <i>Capas de arquitectura de sistema.</i>	68
Ilustración No. 44: <i>Diagrama de clases de análisis de sistema.</i>	68
Ilustración No. 45: <i>Diagrama del patrón de diseño – Capa de presentación.</i>	69
Ilustración No. 46: <i>Diagrama del patrón de diseño – Capa de negocio.</i>	69
Ilustración No. 47: <i>Diagrama del patrón de diseño – Capa de acceso a datos.</i>	69
Ilustración No. 48: <i>Diseño de clases del sistema informático.</i>	70
Ilustración No. 49: <i>Diseño de modelo de datos del sistema informático.</i>	71
Ilustración No. 50: <i>Diagrama de capas con sus componente del sistema.</i>	72
Ilustración No. 51: <i>Diagrama de componente del sistema.</i>	72
Ilustración No. 52: <i>Diagrama de despliegue del sistema.</i>	73
Ilustración No. 53: <i>Prueba unitaria a la clase GisColaborador.</i>	78
Ilustración No. 54: <i>Prueba unitaria a la clase GisConfiguración.</i>	79
Ilustración No. 55: <i>Prueba unitaria a la clase GisDenuncias.</i>	81
Ilustración No. 56: <i>Prueba unitaria a la clase GisGrupo.</i>	82
Ilustración No. 57: <i>Prueba unitaria a la clase GisGrupoUsuario.</i>	84
Ilustración No. 58: <i>Prueba unitaria a la clase GisHechoPublico.</i>	85
Ilustración No. 59: <i>Prueba unitaria a la clase GisHechoUsuario.</i>	86
Ilustración No. 60: <i>Prueba unitaria a la clase GisInmueble.</i>	88

Ilustración No. 61: <i>Prueba unitaria a la clase GisPersona</i>	89
Ilustración No. 62: <i>Prueba unitaria a la clase GisPrevencion</i>	91
Ilustración No. 63: <i>Prueba unitaria a la clase GisServicioWeb</i>	92
Ilustración No. 64: <i>Prueba unitaria a la clase GisUsuarioMail</i>	93
Ilustración No. 65: <i>Prueba unitaria a la clase GisVehiculo</i>	95
Ilustración No. 66: <i>Prueba unitaria a la clase GisVerificacionHecho</i>	96
Ilustración No. 67: <i>Prueba en proceso del aplicativo móvil</i>	97
Ilustración No. 68: <i>Prueba terminada del aplicativo móvil</i>	97
Ilustración No. 69: Se genera un código de incidencia.....	109

CAPÍTULO 1. INTRODUCCIÓN

1.1. Realidad problemática

Según Resumen del Informe Regional de Desarrollo Humano 2013-2014 elaborado por el PNUD en el 2013; la inseguridad es un problema compartido por 18 países de la región analizados, incluido Perú, existiendo variaciones entre ellos e incrementándose según estadísticas a un 12%, así mismo los problemas sociales como pésima limpieza pública, deterioro de pistas, parques y jardines en mal estado, alumbrado deficiente, desagüe, etc. varían significativamente de un país a otro, por ejemplo el sistema de limpieza pública y transporte de nuestro hermano país de Chile no es el mismo que en nuestro país lo que constituye para dichos gobiernos incidencias sociales o hechos que suscitan en sus comunidades. Es por esto que en dichos países de la Región y del mundo se vienen implementando ideas creativas basadas en tecnologías tanto web como móvil para la gestión de las incidencias sociales. Y así lo demuestra la Agencia de Noticias IPS Inter Press Services (2014) en su artículo ‘Aplicaciones móviles buscan la seguridad de las mujeres en India’.

En la actualidad, en el Perú muy pocos son los gobiernos locales que cuentan con un sistema informático (web/móvil) para la sociedad civil que colabore con la gestión de los problemas sociales como extorsiones, sicariato, robos, autos mal estacionados, deterioro de pistas, inundaciones, acumulación de basura, caídas de postes etc. Los gobiernos locales independientemente de su nivel económico sufren este tipo de problemática. Municipalidades distritales limeñas como Miraflores, San Isidro, Jesús María, Barranco, etc. cuentan con este tipo de aplicación informática, es decir son gobiernos locales con altos ingresos económicos, mientras que los gobiernos locales de baja solvencia económica se ven limitados a contar con un sistema informático que los ayude a controlar y disminuir en gran medida dichos problemas. La aplicación web/móvil ‘Miraflores en tu corazón’ es una muestra de las pocas aplicaciones que ayudan a gestionar este tipo de incidencias en la sociedad.

En la municipalidad del distrito de Florencia de Mora, Provincia de Trujillo, no existe un sistema informático (web/móvil) que colabore en la gestión (captura, registro y derivación) de la problemática en cuestión. Esto se debe por la baja recaudación tributaria que percibe la municipalidad y/o por la falta de iniciativa de las autoridades locales en éste tipo de inversión tecnológica. Es por esto que es necesario la

implementación de un sistema informático que controle las incidencias sociales vía on-line utilizando la tecnología móvil y que luego gestionará los estados de incidencia.

1.2. Formulación del problema

¿De qué manera la implementación de un Sistema de Información web/móvil influye en la gestión de las incidencias sociales en la Municipalidad Distrital de Florencia de Mora?

1.3. Justificación

La implementación del presente sistema informático permitirá colaborar con los vecinos de la localidad en la gestión de incidencias tales como asaltos, asesinatos, extorsiones, secuestros, acumulación de basura en horarios y lugares no debidos, deterioro de las pistas, áreas verdes descuidadas, lugares recreativos en mal estado, etc. Esto creará un valor agregado a la gestión municipal, aumentando las vías de comunicación entre el municipio y la comunidad, y así los vecinos se sientan más atendidos por sus autoridades locales. El aplicativo informático servirá como antecedente para la implantación en otras municipalidades distritales dentro de la Provincia de Trujillo y a la vez permitirá la aplicación de metodologías de desarrollo de software para casos sociales y profundizar los conocimientos en relación a éste tema. En lo que respecta a la parte académica para los integrantes que realizamos dicho Proyecto seguiremos adquiriendo experiencia en el desarrollo de sistemas informáticos de proyección social y posibilitará encausar futuras investigaciones de éste tipo.

1.4. Limitaciones

Las limitaciones para el presente Proyecto en lo que respecta al conocimiento es en la parte de desarrollo con Android para lo cual estamos llevando un curso acelerado y asesoramiento de un Profesional del área. Otra limitación que estamos teniendo es la poca información sobre éste tipo de Proyecto tanto en tesis como en proyectos de investigación. Y, por último; una limitación importante es el tiempo ya que nos encontramos laborando en distintas organizaciones públicas y privadas.

1.5. Objetivos

1.5.1. Objetivo General

Mejorar la gestión de incidencias sociales en la Municipalidad Distrital de Florencia de Mora mediante el desarrollo de un sistema informático.

1.5.2. Objetivos Específicos

- Reducir el tiempo de registro de las incidencias sociales.
- Reducir el tiempo de derivación de las incidencias sociales.
- Reducir el tiempo de procesamiento de las incidencias sociales.

CAPÍTULO 2. MARCO TEÓRICO

2.1. Antecedentes

- Jorge Fernández (2014), para optar el grado de Ingeniero de Sistemas en la Universidad Politécnica de Valencia, decidió implantar un sistema de gestión de incidencias para el departamento de informática de una empresa y así mejorar el proceso, dicho proyecto lo denominó “Implantación de un Sistema de Gestión de Incidencias”. Durante años se ha utilizado un gestor de correo como vía de comunicación entre los usuarios de los sistemas y el departamento de informática. Sin embargo, para el investigador existen soluciones más adecuadas que facilitan la resolución de problemas, acortando los tiempos de espera y mejorando el rendimiento. Tras un análisis, el investigador decidió implantar Kayako Fusion, que es un sistema de tickets y a través de diversas funciones, mejorar la gestión y el funcionamiento de una empresa en lo que se refiere a la resolución de incidencias, problemas, fallos y dudas que puedan surgir en cualquier momento a los trabajadores, facilitando mucho la tramitación de las consultas y las respuestas.

Después de la implantación del sistema se generó una base de conocimientos y recursos para los usuarios a la hora de resolver dudas que tengan y fallos que puedan cometer. Se minimizó el número de consultas realizadas al departamento de sistemas, ya que, mediante la base de conocimientos, cada usuario podría resolver gran parte de sus dudas por sí mismo. Y también los ayudaría a resolver fallos habituales, encontrando estrategias para no volverlos a cometer.

Esta investigación nos dio una idea clara en qué consiste la gestión de incidencias y cuál debería ser su proceso. En nuestro caso se trata de gestionar incidencias sociales del distrito de Florencia de Mora la cual describimos en el alcance.

- Javier Fombona (2012) realizó una monografía en la Universidad de Oviedo y la Facultad de Formación del Profesorado denominado “Gestión de Incidencias Informáticas” donde menciona que la dificultad principal que posee dicha universidad es que no poseían informes y experiencias contrastadas orientadas sobre buen uso, servicio y mantenimiento de los recursos informáticos,

generando así incapacidad de gestión de dichos activos. Este trabajo *ex post facto* intenta descubrir algunos rasgos de tal complejidad, a través del estudio de las incidencias informáticas más representativas que tuvieron lugar en la Universidad de Oviedo y en una de sus facultades (Formación del Profesorado y Educación). También pretende aportar pautas para tomar mejores decisiones en este ámbito y difundir las actuaciones significativas de uso de estas tecnologías en los centros de educación superior.

Para ello se analizan las incidencias recogidas durante tres cursos sucesivos en la Universidad de Oviedo, institución que acoge a unas treinta mil personas repartidas en cuatro campus. Los resultados obtenidos destacan que, aunque aumenta el número de dispositivos informáticos, la cantidad de incidencias generadas permanece constante. Esto apunta a que los usuarios saben emplearlos mejor. La mayoría de los problemas en cuanto a la institución universitaria como a la facultad, están relacionados con el software, por lo que el autor sugiere la implementación de potentes servicios centralizados de actualización y mantenimiento de los programas utilizados.

Acorde con este estudio, tomamos en consideración el concepto de centralización de la información que para nuestro proyecto sería el registro de incidencias sociales en el sistema web de la municipalidad para luego ser derivado a las áreas pertinentes.

- El Programa de las Naciones Unidas para el Desarrollo (PNUD) emitió un informe regional de Desarrollo Humano 2013-2014 denominado "Seguridad Ciudadana con rostro humano: diagnóstico y propuestas para América Latina. Lo que llevó al PNUD a dedicar éste informe es la gran violencia, con más de 100,000 homicidios registrados por año y los niveles altos de mala salud. Los costos humanos y sociales de ésta problemática delictiva y mala salud son demasiado altos.

La seguridad ciudadana es un tema muy preocupante en todo nivel del estado - gobierno local, regional y central- sobre todo en campañas políticas. Para ello la PNUD apoya el desarrollo de políticas nacionales integrales de seguridad ciudadana y convivencia, la creación de observatorios de violencia y reformas legales para el control de armas o el diseño de planes locales de seguridad. Y los resultados son muy prometedores en muchos países.

Basados en el antecedente de la PNUD que apoya mediante reformas, planes y políticas para la disminución de la inseguridad ciudadana, nosotros autores del presente plan de tesis proponemos el cómo dar cuenta de dichos problemas de inseguridad ciudadana y a la vez de problemas ambientales a las autoridades pertinentes para su posible solución.

2.2. Bases Teóricas

2.2.1. Incidencia.

DeConceptos.com afirma que la palabra incidencia, proviene en su etimología del vocablo latino ‘incidencia’, utilizado para hacer referencia a distintas situaciones y en distintos campos de la vida como en la salud, computación, medicina, etc. Puede referirse a un hecho que acontece mientras está ocurriendo un negocio u otra situación, relacionada con ello. Por ejemplo, se está celebrando una compra venta de mercaderías y ocurre un asalto, quitándole el dinero a quien debía pagarlas. La incidencia del hurto sobre la operación comercial es tan intensa que puede llegar a dañarlo, por la imposibilidad de pago de la víctima.

En lo que respecta a las incidencias dentro de un distrito hay de varios tipos, pero lo que se va a tratar en el presente proyecto son incidencias de tipo seguridad ciudadana como asalto, robo, extorsión, sicariato, asesinatos, maltrato a la persona, etc. Y de tipo ambiental como deterioro de pistas, aniegos, jardines en mal estado, postes caídos, acumulación de basura, etc.

2.2.2. Gestión de Incidencias.

El proceso de la gestión de incidencias comienza con la captura de dicha incidencia que puede ser una foto, un vídeo o un texto, luego con su registro y finalmente la solución total o parcial de la misma por parte de la municipalidad o institución externa a la institución edil. ITIL®Foundation sostiene que es frecuente que existan múltiples incidencias concurrentes, por lo que es necesario determinar un nivel de prioridad para la resolución de las mismas. Sigue sosteniendo ITL®Foundation:

La priorización se basa esencialmente en dos parámetros:

- **Impacto:** determina la importancia de la incidencia dependiendo de cómo ésta afecta al número de usuarios afectados.
- **Urgencia:** depende del tiempo máximo de demora que una institución externa a la municipalidad solucione la incidencia. En éste caso las instituciones podría ser Sedalib, Hidrandina, Segat, etc.

También se deben tener en cuenta factores auxiliares tales como el tiempo de resolución esperado y los recursos necesarios: las incidencias “sencillas” se tramitarán cuanto antes. Dependiendo de la prioridad, se asignarán los recursos necesarios para la resolución de la incidencia.

La prioridad del incidente puede cambiar durante su ciclo de vida. Por ejemplo, se pueden encontrar soluciones temporales que restauren aceptablemente los niveles de servicio y que permitan retrasar el cierre del incidente sin graves repercusiones.

2.2.3. Aplicaciones Móviles.

Lance Talent (2016) afirma que las aplicaciones móviles en la actualidad no paran de crecer. Son muchas las empresas públicas y privadas tanto pequeñas como grandes en adquirir una aplicación que cumpla en dar respuesta a sus necesidades. Sin embargo, la mayoría desconocen qué tipos de aplicaciones móviles existen y cuál es la mejor para ellos. Y para conocer más los tipos de aplicaciones móviles existentes en el mercado a continuación explicaremos de manera general sobre ellos:

➤ **Aplicación Móvil Nativa.**

Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Acceso completo al dispositivo • Mejor experiencia del usuario • Visibilidad en APP Store • Envío de notificaciones o “avisos” a los usuarios • La actualización de la app es constante 	<ul style="list-style-type: none"> • Diferentes habilidades / idiomas / herramientas para cada plataforma de destino • Tienden a ser más caras de desarrollar • El código del cliente no es reutilizable entre las diferentes plataformas

Tabla No. 1: *Ventajas e Inconvenientes de la aplicación móvil nativa*

Una aplicación nativa es la que se desarrolla de forma específica para un determinado sistema operativo, llamado *Software Development Kit* o SDK. Cada una de las plataformas, Android, iOS o Windows Phone, tienen un sistema diferente, por lo que si quieres que tu app esté disponible en todas las plataformas se deberán de crear varias apps con el lenguaje del sistema operativo seleccionado.

Por ejemplo:

- Las apps para iOS se desarrollan con lenguaje Objective-C
- Las apps para Android se desarrollan con lenguaje Java
- Las apps en Windows Phone se desarrollan en .Net

La descarga e instalación de estas apps se realiza siempre a través de las tiendas de aplicaciones (app store de los fabricantes).

➤ **Aplicación Móvil Web**

Ventajas	Inconvenientes
<ul style="list-style-type: none"> • El mismo código base reutilizable en múltiples plataformas • Proceso de desarrollo más sencillo y económico • No necesitan ninguna aprobación externa para publicarse (a diferencia de las nativas para estar visibles en app store) • El usuario siempre dispone de la última versión • Pueden reutilizarse sitios “responsive” ya diseñados 	<ul style="list-style-type: none"> • Requiere de conexión a internet • Acceso muy limitado a los elementos y características del hardware del dispositivo • La experiencia del usuario (navegación, interacción..) y el tiempo de respuesta es menor que en una app nativa • Requiere de mayor esfuerzo en promoción y visibilidad

Tabla No. 2: *Ventajas e Inconvenientes de la aplicación móvil web*

Una aplicación web es desarrollada en lenguajes como HTML, Javascript y CSS. La principal ventaja con respecto a la nativa es la posibilidad de programar independiente del sistema operativo en el que se usará la

aplicación. De esta forma se pueden ejecutar en diferentes dispositivos sin tener que crear varias aplicaciones.

Las aplicaciones web se ejecutan dentro del propio navegador web del dispositivo a través de una URL.

La gran diferencia con una aplicación nativa (además de los inconvenientes que se muestran en la tabla) es que no necesita instalación por lo que no están visibles en app store.

Las apps web móviles son siempre una buena opción si nuestro objetivo es adaptar la web a formato móvil.

➤ **Aplicación Móvil Híbrida.**

Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Es posible distribuirla en las tiendas de iOS y Android. • Instalación nativa pero construida con JavaScript, HTML y CSS • El mismo código base para múltiples plataformas • Acceso a parte del hardware del dispositivo 	<ul style="list-style-type: none"> • Experiencia del usuario más propia de la aplicación web que de la app nativa • Diseño visual no siempre relacionado con el sistema operativo en el que se muestre

Tabla No. 3: Ventajas e Inconvenientes de la aplicación móvil híbrida

Una aplicación híbrida es una combinación de las dos anteriores, se podría decir que recoge lo mejor de cada una de ellas. Las apps híbridas se desarrollan con lenguajes propios de las aplicaciones móvil web, es decir, HTML, Javascript y CSS por lo que permite su uso en diferentes plataformas, pero también dan la posibilidad de acceder a gran parte de las características del hardware del dispositivo. La principal ventaja es que a pesar de estar desarrollada con HTML, Java o CSS, es posible agrupar los códigos y distribuirla en app store.

2.2.4. Sistema de información web

Un sistema de información web es un sistema de información que usa tecnologías de internet para brindar información y servicios a una persona u

organización pública y privada, y al que se puede acceder a través de la world wide web (www).

(Ye, y otros, 2014) Afirma que a medida que internet fue ganando espacio como herramienta para vender, no fue suficiente utilizarla como medio publicitario, se pensó que pudiera realizar transacciones. Por ello, hoy en día las nuevas tecnologías de la información aplicadas son una herramienta que usan las organizaciones pequeña, mediana y grande para brindar una mejor atención y tener un mejor nivel competitivo.

2.2.4.1. Metodologías de desarrollo web

Existen varias metodologías para el desarrollo web, las cuales se pueden agrupar en dos grupos. Por un lado, tenemos las metodologías tradicionales, que usan procesos bien definidos.

Básicamente las metodologías tradicionales buscan dividir en etapas el proceso de desarrollo de una manera secuencial, siendo de alto grado de ordenamiento y disciplina, pero cuando se está desarrollando un proyecto se resiste al cambio. La resistencia al cambio no se da en el otro grupo de metodologías conocidas como metodologías ágiles, las cuales son una buena elección cuando el proyecto en el que se está trabajando presenta requisitos desconocidos o variables.

Conforme a (Álvarez García, y otros, 2012) las metodologías ágiles basan su fundamento en la adaptabilidad de los procesos de desarrollo y se centran en dos aspectos fundamentales: retrasar las decisiones y la planificación adaptativa.

2.2.4.1.1. Comparación de las metodologías

Metodología RUP

De acuerdo a (Sidnei Wazlawick, 2014) define RUP (Rational Unified Process) como una secuencia de pasos necesarios para el desarrollo y/o mantenimiento de gran cantidad de sistemas, en diferentes áreas de aplicación diferentes organizaciones, diferentes medios de competencia y en proyectos de tamaños variables (desde el más básico al más complejo).

Actualmente es propiedad de International Business Machines (IBM) y está basado en un enfoque disciplinado de asignación de tareas y responsabilidades dentro de una organización de desarrollo con la finalidad de asegurar la obtención de un software de alta calidad que satisfagan la necesidad de los usuarios finales dentro de un calendario y tiempo predecible.

Metodología XP

(Pérez, y otros, 2011) afirman que la metodología XP (Xtreme Programming: programación extrema) es una metodología ágil basada en una serie de valores y de prácticas de buenas maneras con el objetivo de aumentar la productividad a la hora de desarrollar programas. Las características fundamentales del método son: desarrollo iterativo e incremental, pruebas unitarias continuas, programación en parejas y la refactorización y simplicidad en el código.

Metodología SCRUM

Según (Letelier, 2011), Scrum es una metodología de desarrollo muy simple, requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Scrum es una metodología ágil, y como tal:

- Es un modo de desarrollo de carácter adaptable más que predictivo.
- Orientado a las personas más que a los procesos.
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Comienza con la visión general del producto, especificando y dando detalle a las funcionalidades de las partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un periodo de tiempo breve (30 días).

2.2.5. Selección de la metodología a usar en el presente proyecto.

Para la elección de la metodología a usar en el presente proyecto se consultó a varios especialistas y docentes del tema para que nos den una visión más clara de las metodologías usadas en desarrollo de sistemas web. Los especialistas coincidieron que deberíamos tomar ciertos criterios para la elección de una metodología, dichos criterios deberían ser: información, conocimiento, flexibilidad, compatibilidad, requerimientos y tipo de desarrollo.

Reunido el equipo de trabajo, se discutió las metodologías consideradas en el punto 2.2.4.1. bajo los criterios expuestos en el párrafo anterior. Y también se tomó en cuenta la metodología más usada en el transcurso de nuestra carrera profesional, llegando a la conclusión que para el desarrollo del proyecto “Sistema Informático para la Gestión de Incidencias Sociales en la Municipalidad Distrital de Florencia de Mora” se hará uso de la metodología RUP.

2.2.6. Metodología RUP.

Rigoberto (2012) define que el Proceso Unificado Racional, Rational Unified Process en inglés, y sus siglas RUP, es un proceso de desarrollo de software y constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. La metodología RUP no es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí.

- RUP es explícito en la definición de software y su trazabilidad, es decir, contempla desde el análisis de requerimientos hasta la implementación y pruebas.
- RUP identifica claramente a los profesionales (actores) involucrados en el desarrollo del software y sus responsabilidades en cada una de las actividades.

Fases de desarrollo del software:

- Inicio
- Elaboración
- Construcción
- Transición

RUP para el desarrollo de software moderno que junto con UML trata de mejorar el desarrollo de software no solo con una serie de pasos establecidos si no combinando varios modelos, esto dependiendo de las necesidades de la empresa que lo solicite. Para el desarrollo del sistema informático del presente proyecto utilizaremos ésta metodología por ser flexible y estar acorde a los requerimientos y necesidades de la problemática antes señalada. Además, es una metodología que hemos empleado en gran parte de nuestra carrera universitaria.

2.2.7. Android.

Nieto Alejandro (2011) sostiene que Android es un sistema operativo diseñado inicialmente pensado para dispositivos móviles, basado en el núcleo Linux, sistema operativo libre, gratuito y multiplataforma.

El sistema operativo android permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como GPS, llamadas, agenda, etc.) de una forma sencilla en un lenguaje de programación muy conocido como es Java.

Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hace que una de las cosas más importantes de este sistema operativo sea la cantidad de aplicaciones disponibles, que aumentan la experiencia del usuario en el manejo de su dispositivo móvil.

Una de las mejores características de este sistema operativo es que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Y esto lo hace muy popular entre fabricantes y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos.

Para el presente proyecto, se eligió android por las bondades mencionadas en el párrafo anterior y porque conocemos más este sistema operativo a través de los cursos recibidos en la Universidad Privada del Norte.

2.2.8. Android Studio.

Android Studio es el IDE desarrollado por Google especialmente para crear aplicaciones Android. Sus herramientas y su entorno gráfico permiten ver la apariencia exacta de tu aplicación mientras la desarrollas (Carlos Solis (formador). 2014).

2.2.9. NetBeans IDE.

Manuel T. Remon (2013) sostiene que NetBeans IDE es una aplicación de código abierto diseñada para el desarrollo de aplicaciones entre las distintas plataformas.

Con NetBeans se puede:

- Crear interfaces gráficas de forma visual.
- Desarrollar aplicaciones web.
- Crear aplicaciones compatibles con teléfonos móviles.

Cabe recalcar que usaremos éste IDE usaremos para el desarrollo del presente sistema informático.

2.2.10. UML

Según Rober C. Martin (2004) El Lenguaje Unificado de Modelado (UML) es una notación gráfica para dibujar diagramas de conceptos de software. Se puede utilizar para dibujar diagramas de un dominio del problema, un diseño del software propuesto o una implementación de un software ya completado. Fowler describe estos tres niveles diferentes como *Conceptual, de Especificación y de Implementación*.

Los tipos de diagramas son:

- Diagramas de Clases.
- Diagramas de Objetos.
- Diagramas de Secuencia.

- Diagramas de Colaboración.
- Diagramas de Estado.

Los diagramas mostrados son suficientes para la mayoría de los propósitos.

2.3. Definición de términos básicos

Sistema Informático: Un sistema informático es el resultado de la interrelación de hardware, software y recurso humano que permitirá en este caso capturar, registrar y derivar incidencias ocurridas en un determinado lugar.

Gestión de incidencias sociales: Es el proceso por el cual se registra una incidencia en el sistema web de la municipalidad para luego ser derivada al área correspondiente para su posible solución. Éste tipo de incidencias sociales son de dos tipos: incidencias medio ambientales e incidencias de inseguridad ciudadana.

Captura: Grabar vídeo, tomar foto o escribir texto sobre una incidencia a través de un dispositivo móvil.

Registro: Informe de los tipos de incidencias a través de fotos, vídeos o texto, captada por un dispositivo móvil.

Derivación: Remitir de acuerdo al tipo de incidencia capturada y registrada a través del dispositivo móvil al área que le compete darle una solución debida que puede ser limpieza pública, seguridad ciudadana, áreas verdes, sedalib, hidrandina, etc.

CAPÍTULO 3. HIPÓTESIS

3.1. Formulación de la hipótesis

La implementación de un sistema informático web/móvil influye positivamente en la gestión de incidencias sociales mejorando los tiempos de registro, derivación y procesamiento de incidencias en la Municipalidad Distrital de Florencia de Mora.

3.2. Operacionalización de variables

VARIABLE	DEFINICIÓN CONCEPTUAL	DIMENSIONES	INDICADORES
Sistema Informático	Un sistema informático es el resultado de la interrelación de hardware, software y recurso humano que permitirán almacenar y procesar información para cumplir un objetivo.	Tiempo	$\frac{\text{Tiempo promedio requerido para registrar incidencia}}{\text{Tiempo total actual de registro de incidencia}}$
Gestión de incidencias sociales	Es el seguimiento que se le hace a una determinada incidencia social desde su captura hasta su derivación.	Tiempo	$\frac{\text{Tiempo promedio requerido para derivar incidencia}}{\text{Tiempo total actual de derivación de incidencia}}$
		Procesamiento de información	$\frac{\text{Tiempo promedio de procesamiento de información.}}{\text{Tiempo total actual de procesamiento de información}}$

Tabla No. 4: Cuadro detallado de la Operacionalización de las Variables. Elaboración: Propia

CAPÍTULO 4. PROPUESTA DE APLICACIÓN PROFESIONAL

4.1. Modelo del Negocio

Después del análisis sobre la idea de negocio, y evaluar las funcionalidades principales que la solución informática proporcionará a los usuarios, se ha planteado el siguiente modelo canvas de negocio para la solución del presente proyecto.

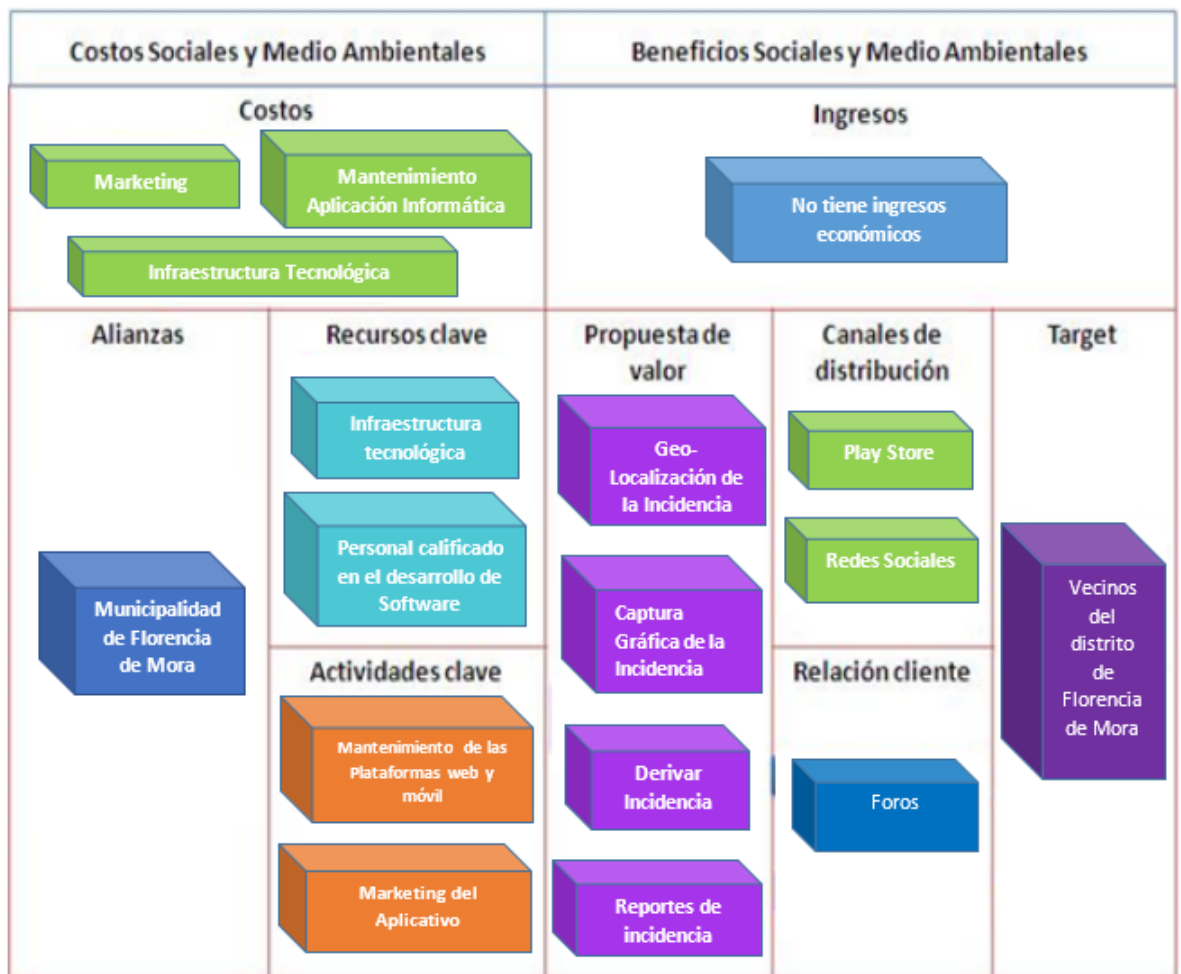


Gráfico No. 1: Modelo Canvas de la solución informática.

4.1.1. Propuesta de Valor.

La aplicación web/móvil enfoca su solución al vecino para reportar incidencias. Dicha aplicación genera valor en el modelo de negocio mediante las siguientes características:

- Geo-localización de la incidencia.- permite al vecino indicar mediante la aplicación de google maps la ubicación exacta de la incidencia ocurrida.
- Captura gráfica de la incidencia.- Mediante la cámara del dispositivo móvil permite capturar la imagen de la incidencia.
- Derivar incidencia.- Deriva la incidencia según su tipo al organismo competente.
- Reportes de incidencia.- Reportes de incidencia para la alta dirección que le permita tomar decisiones y reporte de incidencias al área competente como historial.

4.1.2. Ingresos.

Este modelo de negocio no genera ingresos económicos ya que se trata de un modelo de proyección a la sociedad. En todo caso se trata de ingresos intangibles como son la satisfacción de parte del vecino que registra la incidencia y de parte de la municipalidad que trata de dar solución a dicho hecho reportado.

4.1.3. Estructura de Costos.

Los costos en el presente modelo de negocio provienen de la estimación en los servicios de soporte tecnológico y de promoción.

- Mantenimiento de la aplicación informática.- Costo referido al recurso humano del proyecto.
- Marketing.- Costo referido a campañas de promoción para el conocimiento de la aplicación.
- Infraestructura Tecnológica.- Costo referido al soporte de infraestructura en tecnologías de información, computadoras personales, dominio y hosting.

4.1.4. Canales de Distribución.

El aplicativo web/móvil se dará a conocer al vecino a través de las redes sociales y podrá ser descargado desde Play Store.

4.1.5. Relación con el Cliente.

- Foros.

4.1.6. Segmento de Mercado.

La presente aplicación está dirigida al vecino florenciano y a las personas que se encuentren en el distrito que vean y quieran reportar una incidencia tipo seguridad ciudadana y medio ambiental.

4.1.7. Alianzas.

La alianza por el momento es sólo con la municipalidad distrital de Florencia de Mora.

4.1.8. Actividades Clave.

Entre las actividades clave que apoyan este proyecto, se encuentran:

- Mantenimiento de las plataformas web y móvil. - El soporte especializado a ambas plataformas garantizará el acceso permanente del vecino al aplicativo móvil y del administrador web al aplicativo web.
- Marketing del aplicativo. - Hacer conocer el aplicativo a través de las redes sociales, página web de la municipalidad e instituciones aliadas.

4.1.9. Recursos Clave.

Entre los recursos clave que apoyan este proyecto, se encuentran:

Infraestructura tecnológica. - Recurso fundamental de la propuesta de negocio, basado en la implementación de dos soluciones: web y móvil. Esta infraestructura permitirá el soporte de la información para la gestión de incidencias.

Personal calificado en desarrollo de software. - Otro recurso fundamental de la propuesta de negocio, basado en el recurso humano especializado en tecnologías de información para el soporte oportuno y efectivo en la gestión de incidencias.

4.2. Plan de Desarrollo de Software

4.2.1. Introducción

4.2.1.1. Propósito

El objetivo del presente documento es proporcionar la información necesaria para controlar el proyecto en términos de fases e iteraciones.

4.2.1.2. Alcance

La presente investigación aborda la implementación de un sistema informático para facilitar a los vecinos de la comunidad florenciana capturar incidencias sociales ocurridas en su distrito para luego mediante el sistema informático ser registradas y derivadas al área competente para su solución. Éste tipo de incidencias sociales son de dos tipos: incidencias medio ambientales e incidencias de inseguridad ciudadana. Es por ello que la investigación es de carácter experimental.

4.2.1.3. Resumen

El plan de desarrollo de software contiene la siguiente información:

- **Vista General del Proyecto** — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.
- **Organización del Proyecto** — describe la estructura organizacional del equipo de desarrollo.
- **Gestión del Proceso** — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.
- **Planes y Guías de aplicación** — proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

4.2.2. Vista General del Proyecto

4.2.2.1. Propósito, Alcance y Objetivos

El propósito principal del proyecto es lograr una solución informática web/móvil integrada e innovadora, que reporte las incidencias sociales dentro de un distrito, teniendo como objetivo facilitar a los trabajadores de la municipalidad el registro y la derivación de dichas incidencias sociales para su atención oportuna.

El alcance del proyecto está relacionado directamente con los objetivos planteados que se detallan a continuación:

Objetivo General

El objetivo general del proyecto es el diseño e implementación de una solución tecnológica para la gestión de incidencias sociales.

Objetivos Específicos

- Analizar y modelar el proceso de negocio en la municipalidad distrital de Florencia de Mora, identificando necesidades y expectativas.
- Elaborar la documentación necesaria para cada fase del proceso de desarrollo de software propuesta.
- Identificar las funcionalidades que aseguren una mejor comodidad para el vecino dentro del proceso de reporte de la incidencia.
- Implementar el aplicativo móvil en Android para el reporte de las incidencias sociales.
- Implementar el aplicativo web para el uso exclusivo del administrador web de la municipalidad.
- Desarrollar base de datos que integre información tanto de la aplicación móvil, como de la aplicación web a implementar.
- Cumplir con los tiempos establecidos en el cronograma para la implementación del proyecto.

- Garantizar el correcto despliegue de la solución tecnológica y almacenamiento de la información, en un entorno virtual centralizado.

4.2.2.2. Suposiciones y Restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de los requerimientos de la empresa son:

Suposiciones:

- Uso de la metodología RUP
- Uso de tecnología web y móvil
- Arquitectura: Modelo Vista Controlado (MVC)
- Uso de la herramienta Rational Software Architect
- Uso de la herramienta Java

Restricciones:

- Tiempo: 24 semanas.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto.

4.2.2.3. Entregables del Proyecto

La presente lista de entregables constituye la configuración de RUP desde la perspectiva de artefactos, y que se seguirá en este proyecto.

Es preciso indicar y destacar que de acuerdo a la filosofía de RUP, no es una metodología con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada Proyecto. Todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Modelo de Casos de Uso del Negocio

Es el modelo de los procesos de negocio vistas desde la perspectiva de los actores externos (vecinos). Permite situar al sistema en el contexto del distrito de Florencia de Mora haciendo énfasis en los objetivos en este ámbito.

Para el presente proyecto, se cuenta con tres Casos de Uso de Negocio:

- Proceso de Registrar incidencias.
- Proceso de Derivar incidencias
- Proceso de Reportes de incidencias

3) Modelo de Objetos del Negocio

Este modelo describe la realización de cada Caso de Uso del Negocio, estableciendo los actores internos (vecino, jefe de área), la información que estos manipulan y los flujos de trabajo asociados al Caso de Uso del Negocio.

4) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

5) Visión

Este documento define la visión del producto desde la perspectiva del vecino (cliente), especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema de gestión de incidencias sociales.

6) Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema de gestión de incidencias sociales y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

7) Especificaciones de Casos de Uso

Para cada Caso de Uso se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados.

8) Especificaciones Adicionales

Este documento captura todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc.

9) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema de gestión de incidencias sociales y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema.

10) Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

11) Modelo de Datos

Puesto que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases.

12) Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

13) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

14) Lista de Riesgos

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

15) Material de Apoyo al Usuario Final

Se incluye Guía del Usuario.

16) Producto

Los ficheros del producto empaquetados y almacenadas en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva release al final de cada iteración.

Los artefactos del 12 al 16 no se generarán sino hasta llegar a las fases finales del proyecto, sin embargo, se han incluido aquí sólo

para dar una visión global de todos los artefactos que se generarán en el proceso de desarrollo.

4.2.3. Organización del Proyecto

4.2.3.1. Participantes del Proyecto

Las personas que conforman el equipo de trabajo y aquellas que serán responsables de desarrollar y revisar los entregables propuestos son:

- Br. Neyra Herrera Edwin Roy
- Br. Angulo Pretel Jorge Luis

Cada uno de ellos aportará al proyecto según las distintas funciones asignadas.

4.2.3.2. Interfaces externas.

El equipo de investigación estará en contacto con los responsables del municipio tanto en el área medio ambiental como de seguridad ciudadana, con quienes se definirá las interfaces del sistema, verificando que este cumpla con los requisitos exigidos.

4.2.3.3. Roles y Responsabilidades.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.

Analista Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario
Asesor de Tesis	Guía en bases a su conocimiento y experiencia profesional durante todo el tiempo que dure el proyecto al equipo investigador.

Tabla No. 5: Roles y responsabilidades del equipo de trabajo

4.2.4. Gestión del Proceso

4.2.4.1. Estimación del Proyecto

Miembro del Equipo	Tiempo (mes)	Costo (S/.) (mes)	Total (S/.)
Jefe de Proyecto	6	0	0
Analista de Sistemas	6	0	0
Programador	4	300	1200
Asesor de Tesis	6	0	0
			1200

Tabla No. 6: Estimación del proyecto.

4.2.4.2. Plan del Proyecto

Plan de las Fases

Fase	Nro. De Iteración	Duración (semanas)
Inicio	2	06
Elaboración	4	12
Construcción	4	19
Transición	1	04

Tabla No. 7: Plan de fases del Proyecto.

Descripción de los hitos

Descripción	Hito
Fase de Inicio	En esta fase desarrollarán los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente/usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla la arquitectura (incluyendo las partes más relevantes y/o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera reléase de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión, aceptación de la arquitectura del sistema y la implementación de los casos de uso core del negocio marcan el final de esta fase.
Fase de Construcción	El producto se construye en base a 2 iteraciones, cada una produciendo una reléase a la cual se le aplican las pruebas y se valida con el cliente/usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la reléase 2.0, con la capacidad operacional del producto, lista con la realización de las pruebas y desarrollo de los informes de reportes.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla No. 8: *Descripción de hitos del proyecto.*

Calendario del Proyecto

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un

estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios. Cabe indicar que en el presente calendario sólo se incluyen las fases de Inicio y Elaboración, ya que la mayoría de los artefactos son generados muy tempranamente en el proyecto, pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo (año 2017)	Aprobación (año 2017)
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1 02/03	Semana 3 20/03
Requisitos		
Glosario	Semana 1 02/03	Semana 3 20/03
Visión	Semana 5 01/04	Semana 9 28/04
Modelo de Casos de Uso	Semana 5 01/04	Semana 9 28/04
Especificación de Casos de Uso	Semana 5 01/04	Semana 9 28/04
Especificaciones Adicionales	Semana 5 01/04	Semana 9 28/04
Análisis/Diseño		
Modelo de Análisis/Diseño	Semana 5 01/04	Semana 14 01/06
Modelo de Datos	Semana 11 15/05	Semana 18 01/07
Implementación		
Prototipos de Interfaces de Usuario	Semana 14 15/05	Semana 18 01/07
Modelo de Implementación	Semana 14 15/05	Semana 18 01/07
Pruebas		
Casos de Pruebas Funcionales	Semana 18 03/07	Semana 23 09/08
Despliegue		
Modelo de Despliegue	Semana 18 03/07	Semana 23 09/08
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las iteraciones	Semana 1 02/03	Semana 23 09/08
Ambiente	Durante todo el proyecto	

Tabla No. 9: *Calendario del proyecto – Fase inicio.*

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo (año 2017)	Aprobación (año 2017)
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1 02/03	Aprobado
Requisitos		
Glosario	Semana 1 02/03	Aprobado
Visión	Semana 5 01/04	Aprobado
Modelo de Casos de Uso	Semana 5 01/04	Aprobado
Especificación de Casos de Uso	Semana 5 01/04	Revisar en cada iteración
Especificaciones Adicionales	Semana 5 01/04	Revisar en cada iteración
Análisis/Diseño		
Modelo de Análisis/Diseño	Semana 5 01/04	Revisar en cada iteración
Modelo de Datos	Semana 11 15/05	Revisar en cada iteración
Implementación		
Prototipos de Interfaces de Usuario	Semana 14 15/05	Revisar en cada iteración
Modelo de Implementación	Semana 14 15/05	Revisar en cada iteración
Pruebas		
Casos de Pruebas Funcionales	Semana 18 03/07	Revisar en cada iteración
Despliegue		
Modelo de Despliegue	Semana 18 03/07	Revisar en cada iteración
Gestión de Cambios y Configuración		
Durante todo el proyecto		
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las iteraciones	Semana 1 02/03	Revisar en cada iteración
Ambiente		
Durante todo el proyecto		

Tabla No. 10: *Calendario del proyecto – Elaboración.*

4.3. Modelo de Casos de Uso del Negocio.

4.3.1. Objetivos del Negocio.

4.3.1.1. Implementar Sistema Informático para gestionar incidencias.

4.3.1.2. Reducir tiempo de registro de incidencias.

4.3.1.3. Reducir tiempo de derivación de incidencias.

4.3.1.4. Reducir tiempo de procesamiento de incidencias.



Ilustración No. 1: *Objetivos del negocio*

4.3.2. Actores del Negocio.

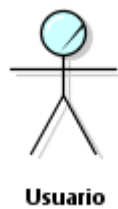


Ilustración No. 2: *Actor del Negocio.*

4.3.3. Casos de Uso de Negocio y Especificación.

4.3.3.1. CU Registrar Incidencia.



Ilustración No. 3: *Diagrama de Caso de Uso Registrar Incidencia.*

Nombre:	Registrar incidencia
Autor:	Roy Neyra
Fecha:	03/07/2017
Descripción:	Caso de uso que describe el proceso de registro de incidencias de parte del vecino (usuario) en la municipalidad distrital de Florencia de Mora.
Actores:	Usuario.
Precondiciones:	1. Haberse producido una incidencia social.
Flujo Normal:	1. Se produce una incidencia en el distrito de Florencia de Mora. 2. El vecino se acerca a la municipalidad a registrar la incidencia social.
Flujo Alternativo:	1. El vecino no se acerca a la municipalidad a registrar la incidencia. 1.1. Se incrementa los problemas sociales en el distrito sin posibles soluciones.
Post-condiciones:	- La incidencia debe llegar al área respectiva para su pronta atención.

4.3.3.2. CU. Derivar Incidencia.

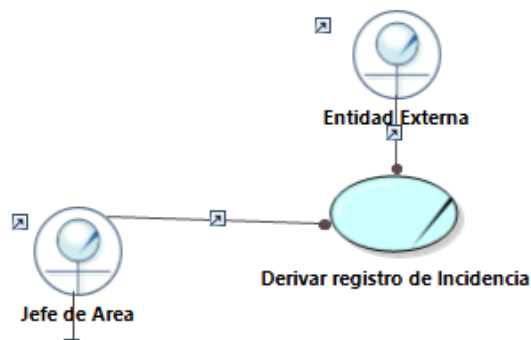


Ilustración No. 4: *Diagrama de Caso de Uso Derivar Incidencia*

Nombre:	Derivar incidencia
Autor:	Roy Neyra
Fecha:	03/07/2017
Descripción:	Caso de uso que describe el proceso de derivar la incidencia registrada por el usuario en el área competente de la municipalidad distrital de Florencia de Mora o a una entidad externa para su posible solución.
Actores:	Jefe de Área
Precondiciones:	Registrar la incidencia.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario se acerca a la Oficina de Mesa de Partes de la municipalidad a registrar la incidencia o el usuario va directamente a la Oficina respectiva a registrar su incidencia. 2. Dependiendo del tipo de incidencia se deriva al Jefe de Área que puede ser de la oficina de seguridad ciudadana y de la Oficina de la oficina de limpieza pública o a una entidad externa a la municipalidad. 3. El Jefe de Área descarta la incidencia si verifica (el asunto se repite en otro documento) que el mismo hecho ha sido registrado por otro usuario.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. El encargado de Mesa de Partes no sabe a quién derivar el registro de la incidencia.
Post-condiciones:	<ul style="list-style-type: none"> - La incidencia debe ser atendido luego de ser derivado.

4.3.3.3. CU Generar Reportes

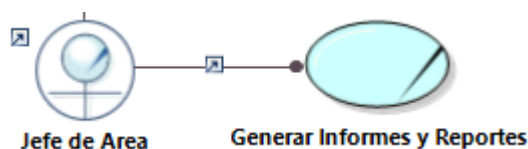


Ilustración No. 5: *Diagrama de Caso de Uso Generar Reportes.*

Nombre:	Generar reportes.
Autor:	Roy Neyra
Fecha:	03/05/2017
Descripción:	Caso de uso que describe el proceso de generar reportes estadísticos para la toma de decisiones a la gerencia municipal.
Actores:	Jefe de Área
Precondiciones:	Haberse derivado la incidencia.

Flujo Normal:

1. La Gerencia o el Concejo Municipal solicita un reporte estadístico de las incidencias registradas en un determinado periodo de tiempo para la toma de decisiones y para cumplir ciertas metas impuestas por el gobierno central.

Flujo Alternativo:

1. No se lleve el control de las incidencias registradas.

Post-condiciones:

- El reporte debe generarse en Excel al área solicitante.

4.3.4. Relación Casos de Uso de Negocio y Objetivo de Negocio.

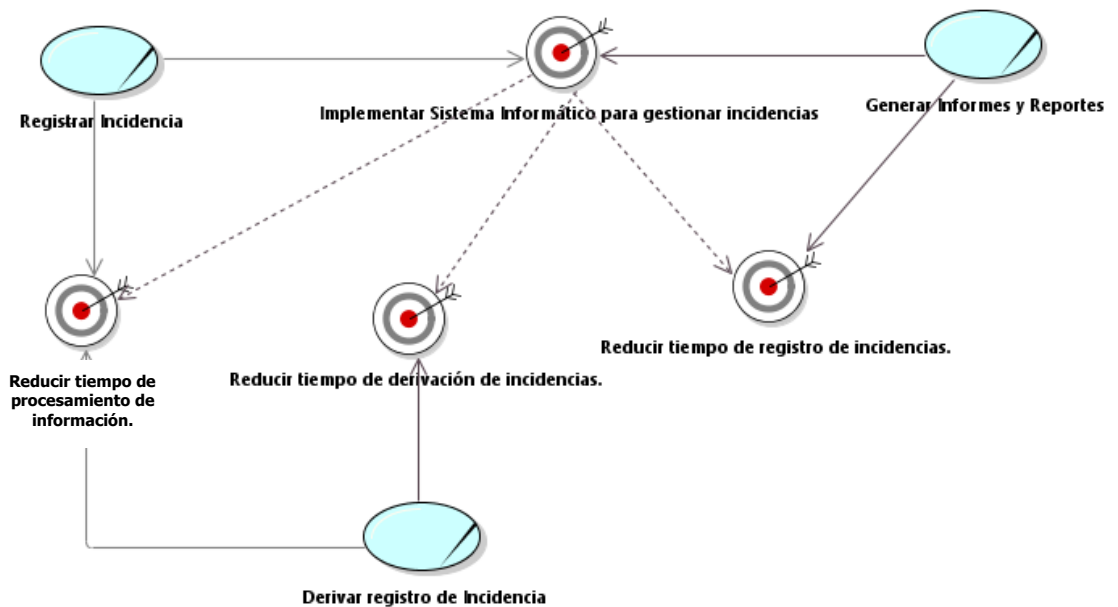


Ilustración No. 6: *Diagrama de relación entre caso de uso y objetivo del negocio*

4.3.5. Modelo de Caso de Uso del Negocio.

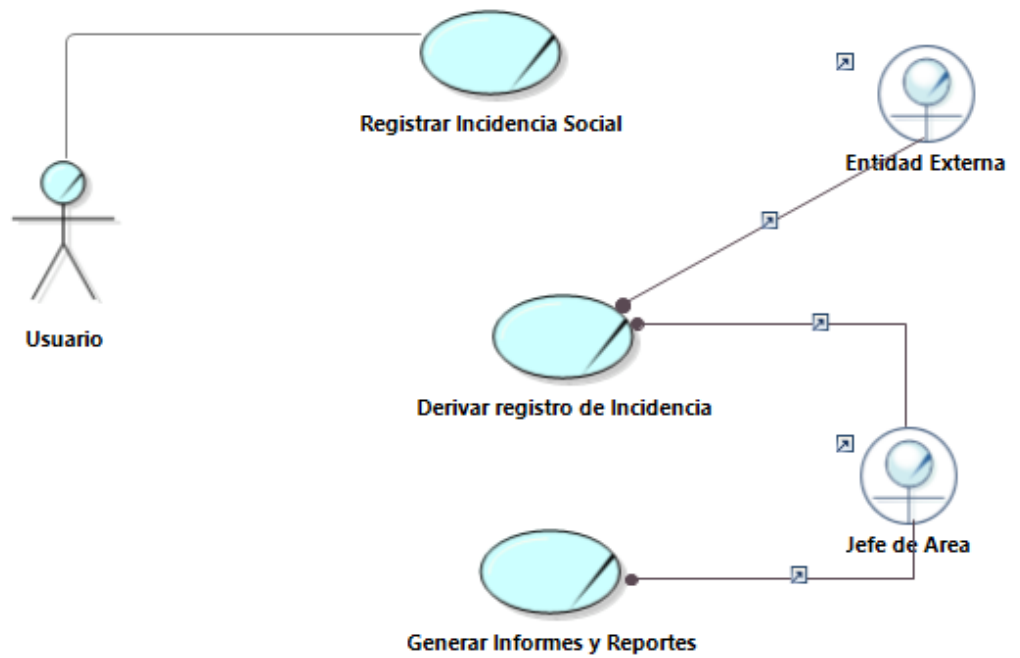


Ilustración No. 7: *Diagrama de caso de uso de negocio*

4.4. Modelo de Objetos del Negocio.

4.4.1. Trabajadores del Negocio.

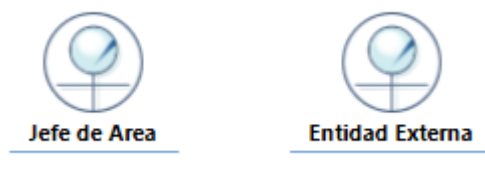


Ilustración No. 8: *Diagrama de trabajadores de negocio*

4.4.2. Entidades del Negocio.



Ilustración No. 9: Diagrama de entidades de negocio

4.4.3. Relación Entidad de Negocio con Trabajador de Negocio.

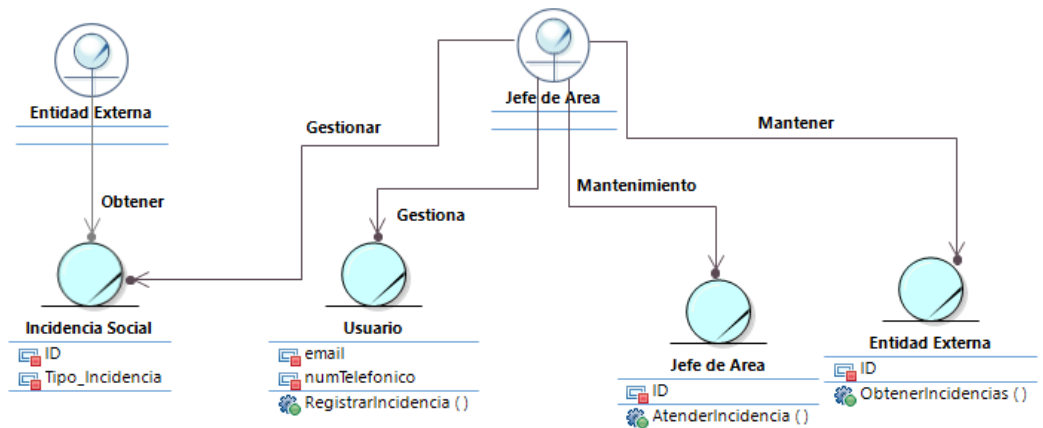


Ilustración No. 10: Diagrama de relación de entidad y trabajador de negocio

4.4.4. Realizaciones de Casos de Uso de Negocio.

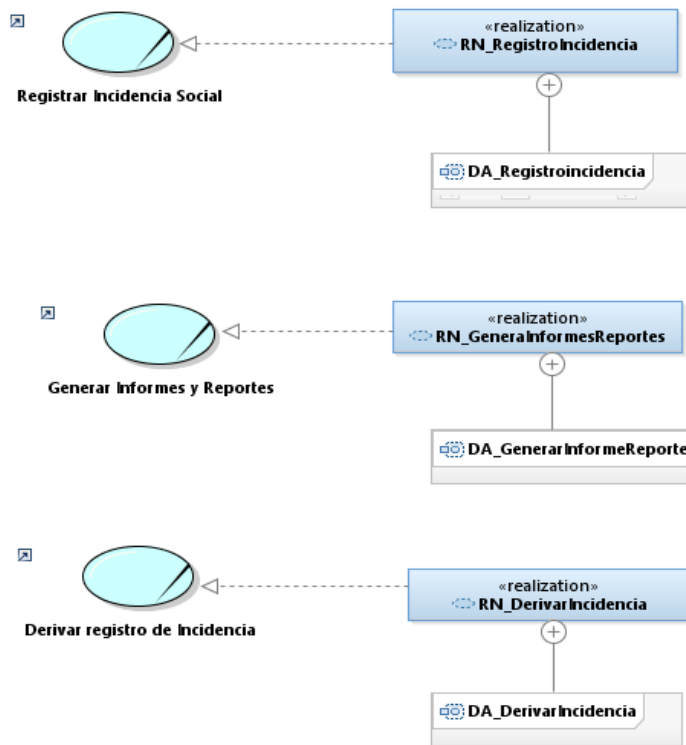


Ilustración No. 11: Diagrama de caso de uso de Negocio

4.4.4.1. Diagrama de Actividades de los Casos de Uso de Realización.

DA_RegistroIncidencia.

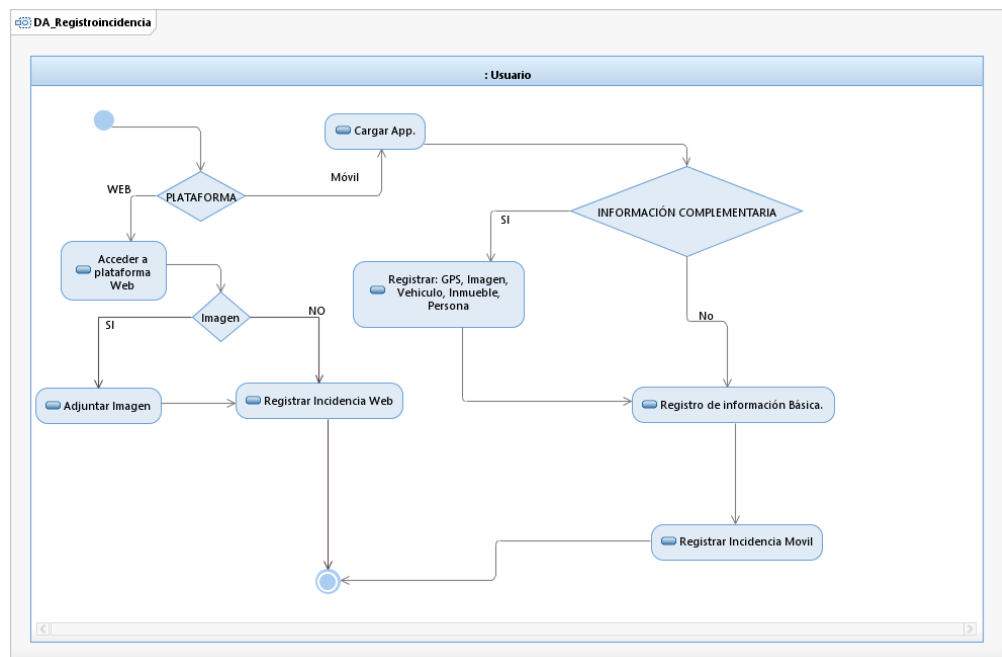


Ilustración No. 12: Diagrama de actividades – Registro de incidencia.

DA_DerivarIncidencia

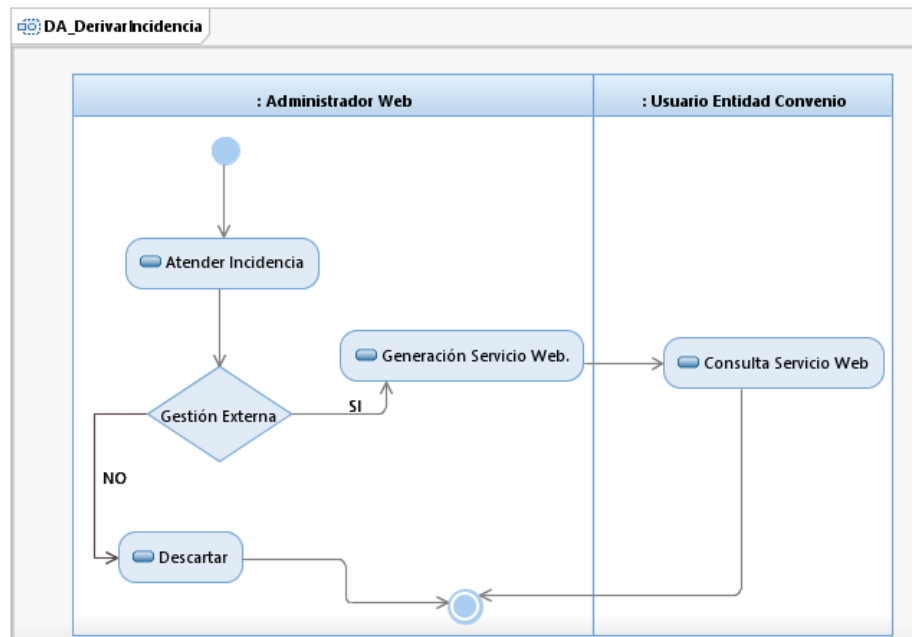


Ilustración No. 13: Diagrama de actividades – Derivar incidencia.

DA_GenerarInformeReporte

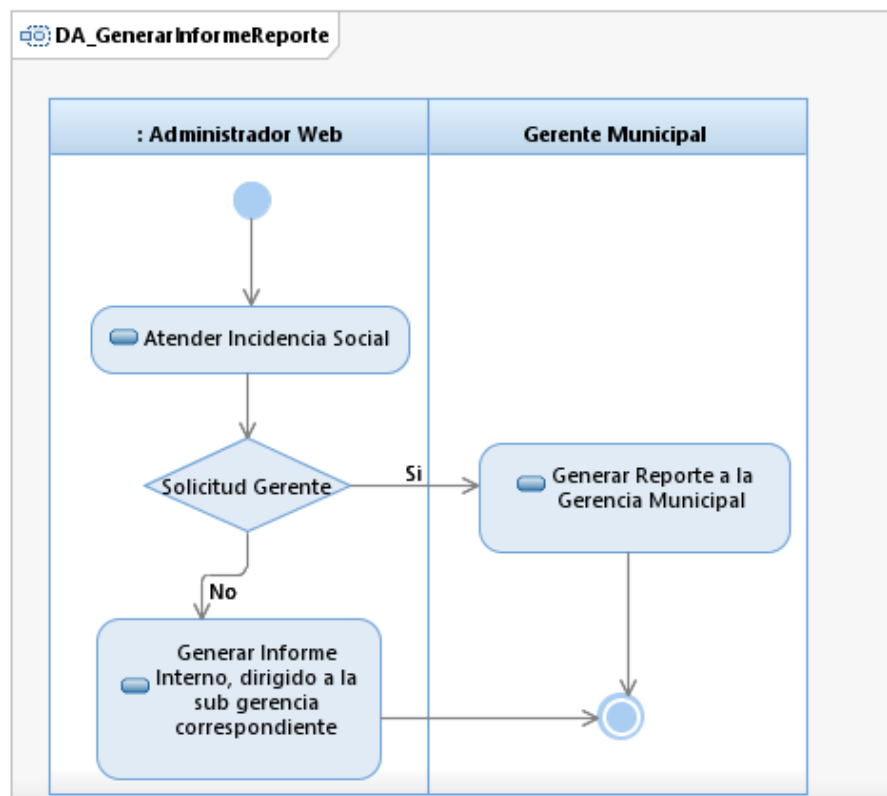


Ilustración No. 14: Diagrama de actividades – Generar reporte

4.4.5. Diagrama de Entidades de Negocio.

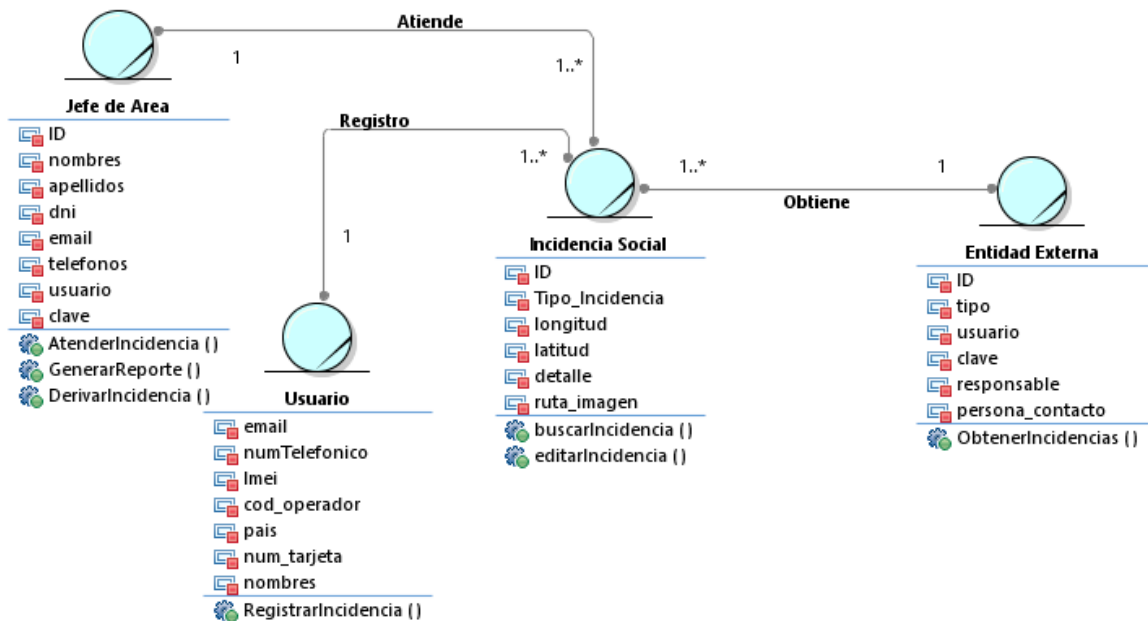


Ilustración No. 15: *Diagrama de entidades del negocio.*

4.5. Glosario.

Sistema Informático: Un sistema informático (móvil/web) es el resultado de la interrelación de hardware, software y recurso humano que permitirá en este caso capturar, registrar y derivar incidencias ocurridas en un determinado lugar.

Gestión de incidencias sociales: Es el proceso por el cual se registra una incidencia en el sistema web de la municipalidad para luego ser derivada al área correspondiente para su posible solución. Éste tipo de incidencias sociales son de dos tipos: incidencias medio ambientales e incidencias de inseguridad ciudadana.

Captura: Grabar vídeo, tomar foto o escribir texto sobre una incidencia a través de un dispositivo móvil.

Registro: Informe de los tipos de incidencias a través de fotos, vídeos o texto, captada por un dispositivo móvil.

Derivación: Remitir de acuerdo al tipo de incidencia capturada y registrada a través del dispositivo móvil al área que le compete darle una solución debida que puede ser limpieza pública, seguridad ciudadana, áreas verdes, sedalib, hidrandina, etc.

Entidad: Es la representación de un objeto del sistema informático, este objeto tiene atributos.

Capas de Arquitectura: Es la distribución de roles y responsabilidades en forma jerárquica. El rol está dado por la interacción con otras capas y las responsabilidades está dada por la función que está desarrollando.

Jefe de Área: Es el Jefe de la Oficina de Seguridad Ciudadana o Jefe de la Unidad de Limpieza Pública.

Entidad Externa: Es la entidad a la cual es derivada la incidencia. Puede ser una oficina interna de la municipalidad (Oficina Seguridad Ciudadana o la Unidad de Limpieza Pública) o a un organismo externo (Sedalib, Hidrandina, Bomberos, Policía Nacional).

4.6. Visión.

4.6.1. Introducción

4.6.1.1. Propósito

El propósito de esta sección es analizar y definir las necesidades del concejo municipal junto a las características de un sistema de gestión de incidencias sociales para la municipalidad distrital de Florencia de Mora. El documento enfatiza la funcionalidad por parte del responsable del proyecto, así como los usuarios finales del sistema.

La funcionalidad se basa principalmente en la gestión de incidencias sociales de tipo medio ambiental y de inseguridad ciudadana que la municipalidad debe realizar en las diferentes zonas del distrito, de manera que todas estas zonas sean atendidas oportunamente y así los vecinos se sientan satisfechos.

4.6.1.2. Alcance

La implementación del sistema informático para la gestión de incidencias sociales en la municipalidad distrital de Florencia de Mora será desarrollada por el equipo investigador de la presente tesis.

La presente investigación aborda la implementación de un sistema informático para facilitar a los vecinos de la comunidad florenciana capturar incidencias sociales ocurridas en su distrito para luego mediante el sistema informático ser registradas y derivadas al área competente para su atención debida. Éste tipo de incidencias sociales son de dos tipos: incidencias medio ambientales e incidencias de inseguridad ciudadana.

4.6.2. Posicionamiento

4.6.2.1. Oportunidad del Negocio.

El sistema informático permitirá a la municipalidad informatizar la gestión de incidencias sociales ocurridas en las diferentes zonas del distrito. Las incidencias serán registradas por los vecinos desde su dispositivo móvil y enviadas a un sistema web de la municipalidad que se encargará de derivar dicha incidencia al área competente para ser atendida de manera oportuna.

4.6.2.2. Sentencia que define el problema

Problema de	Las incidencias sociales por lo general no son comunicadas a las autoridades ediles y cuando lo comunican el vecino tiene que acercarse a la municipalidad y hacer conocer tal incidencia. También lo hacen por teléfono, pero en este caso la atención a la incidencia es casi nula.
Afecta a	Los vecinos del distrito de Florencia de Mora que se ven inmersos en dicha incidencia social.
Impacto del problema	No reportar la incidencia social como por ejemplo pistas deterioradas, caídas de postes, mayor inseguridad ciudadana y por ende mal aspecto del distrito ante los propios vecinos como de sus visitantes.

Una adecuada solución sería	Implementar un sistema informático web-móvil. El sistema móvil lo usarán los vecinos para reportar las incidencias y el sistema web se operará desde el municipio – El vecino al encontrarse frente a una incidencia social; desde su dispositivo móvil capturará dicha incidencia mediante una foto y/o mensaje descriptivo de la incidencia, la cual después enviará al sistema web de la municipalidad. El administrador web al recibir dicha incidencia derivará al área competente para su atención debida.
------------------------------------	--

Tabla No. 11: *Sentencias que definen el problema*

4.6.2.3. Sentencia que define la posición del producto.

Para	<ul style="list-style-type: none"> • Vecinos del distrito de Florencia de Mora.
Quiénes	Operan el dispositivo móvil para reportar incidencias y quienes también mediante el sistema web derivan las incidencias.
El nombre del producto	Es una aplicación web-móvil llamada SISMUNI.
Qué	Gestiona una incidencia social.
No como	Sistema actual, que es totalmente manual.
Nuestro producto	Permite al vecino del distrito de Florencia de Mora, desde su dispositivo móvil reportar una incidencia social ocurrida a su alrededor; tal incidencia puede ser reportada mediante la captura de una imagen del hecho ocurrido acompañado si se quiere de un texto descriptivo del hecho. Una vez reportado el incidente, es registrado en el sistema web de la municipalidad que se encargará dependiendo del tipo de la incidencia al área o institución correspondiente para su debida atención.

Tabla No. 12: *Sentencias que definen la posición del producto.*

4.6.3. Descripción de Stakeholders (Participantes del Proyecto y Usuarios)

Desarrollar un sistema eficiente y de buena calidad que sea apto a las necesidades de los usuarios implica tomar en cuenta los participantes que están inmersos en las funciones de negocio.

Esta sección muestra un resumen sobre los participantes y usuarios involucrados en el proyecto, los problemas que aquejan a la solución propuesta.

4.6.3.1. Resumen de Stakeholders.






Nombre	Descripción	Responsabilidades
 Alcalde	Alcalde distrital de Florencia de Mora.	Supervisa y aprueba el Proyecto.
 Jefe de Seguridad Ciudadana	Encargado de la Oficina de Seguridad Ciudadana del Distrito.	Soluciona los problemas referidos a seguridad ciudadana del distrito.
 Jefe de Limpieza Pública.	Encargado de la Oficina de Limpieza Pública.	Soluciona los problemas referidos a limpieza pública.
 Vecino	Vecino que vive o se encuentra en el distrito de Florencia de Mora.	Reporta la incidencia mediante su dispositivo móvil al sistema web de la municipalidad.
 Entidad Vinculante	Organismo externo a la municipalidad como sedalib, hidrandina, etc.	Soluciona los problemas como caída de postes, incendios, roturas de tubo de desagüe, etc.

Tabla No. 13: Stakeholders del sistema

4.6.3.2. Resumen de Usuarios.



Nombre	Descripción	Responsabilidades
 Vecino	Vecino que vive o se encuentra en el distrito de Florencia de Mora.	Reporta la incidencia mediante su dispositivo móvil al sistema web de la municipalidad.
 Administrador Web	Operador del Sistema Web ubicado en la municipalidad de Florencia de Mora.	Atiende las incidencias sociales que son enviadas por el usuario vecino.

Tabla No. 14: *Usuarios del Sistema.*

4.6.3.3. Entorno de Usuario.

Son dos entornos de usuario, un entorno móvil con sistema operativo Android al que tiene acceso el vecino florenciano para capturar las incidencias -pantallas amigables que indican claramente la función que desempeñan- y un entorno web al que accede el administrador web previo ingreso de su usuario y clave.

4.6.4. Descripción del producto.

4.6.4.1. Perspectiva del Producto.

El producto a desarrollar es un sistema informático web/móvil con la finalidad de automatizar la gestión de incidencias sociales ocurridas dentro del distrito. La parte móvil es la que se encargará de reportar la incidencia social y la parte web la de registrar y derivar la incidencia social al área competente para su pronta atención.

4.6.4.2. Resumen de características.

A continuación, se mostrará un listado con los beneficios que obtendrá el cliente a partir del producto:

Beneficios del Usuario	Características que la apoyan
Mayor facilidad de reportar una incidencia de tipo medio ambiental y de inseguridad ciudadana.	Aplicación móvil atractiva para la vista y que con tres o cuatro click reporta una incidencia.
Mayor rapidez en la atención de la incidencia.	Aplicación web que atiende y deriva la incidencia al área u organismo competente.
Ahorro de tiempo y dinero. El vecino ya no tiene que apersonarse al municipio a reportar una incidencia. Social.	Aplicación móvil que reporta la incidencia desde el lugar ocurrido.
Mejor condición de vida en el aspecto ambiental y seguridad, porque la app pretende dar a conocer y derivar con mayor rapidez una incidencia y poder así ser solucionado a la brevedad posible y estar libre de ese hecho social inapropiado.	Sistema informático web/móvil.

Tabla No. 15: Características del producto final.

4.6.4.3. Suposiciones y dependencias.

El software propuesto deberá ser implementado de la manera más sencilla y entendible por el usuario, suponiendo que el mismo conlleva las características de comunicación, mantenimiento y portabilidad.

4.6.4.4. Costo y Precio.

Producto	Costo (S/.)	Costo Real (S/.)
SOFTWARE		
Ubuntu Server	0	0
Glassfish Server	0	0
Netbeans IDE 8.2	0	0
Android Studio	0	0
MySQL Server Data Base	4000	0
IBM Rational Software Architect	10000	0
HARDWARE		
02 laptops	4600	0
02 Celulares	1000	0
02 Impresoras	380	0
02 usb	70	0
SERVICIOS		
Impresiones de informes	300	0
Servidor Privado Virtual (VPS)	300	300
Pasaje a Lima	320	0
Viáticos a Lima	360	0
Electricidad	240	0
Internet	480	0
OTROS		
Papelería y Suministros	548	548
Imprevistos	200	200
Pasajes	50	50

Tabla No. 16: Costo y precio de hardware, software y servicios del proyecto.

4.7. Modelo de Casos de Uso de Sistema.

4.7.1. Actores de Sistema.

- Usuario
- Administrador Web
- Entidad con Convenio

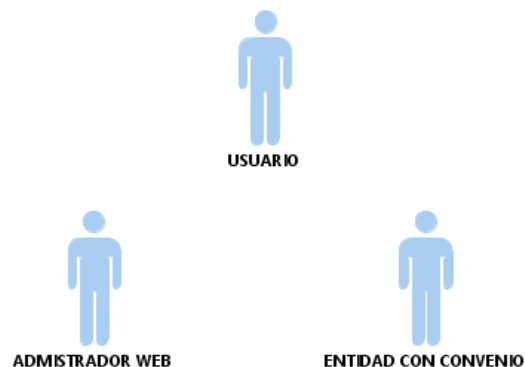


Ilustración No. 16: Diagrama de entidades del negocio

4.7.2. Diagrama de Caso de Uso de Sistema.

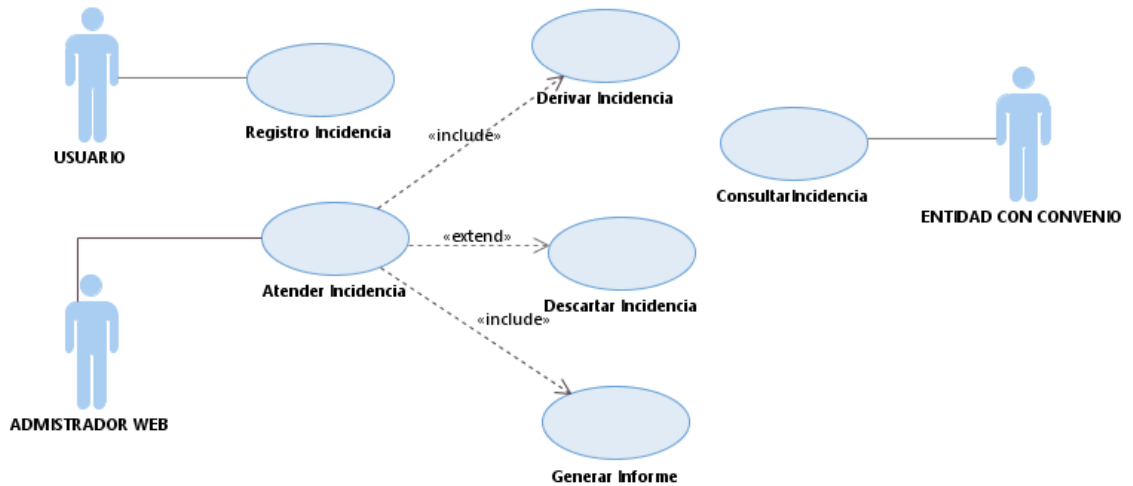


Ilustración No. 17: Diagrama casos de uso de sistema.

Nombre:	Registrar incidencia
Autor:	Roy Neyra
Descripción:	Caso de uso que describe el proceso de reporte de incidencias de parte del vecino (usuario) desde su celular o vía web.
Actores:	Usuario.
Precondiciones:	Haber aceptado los términos y condiciones del uso del aplicativo e iniciar sesión.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se produce una incidencia en el distrito de Florencia de Mora. 2. El usuario accede al aplicativo desde su dispositivo móvil. 3. El sistema muestra una pantalla de bienvenida al sistema y luego pasa a la pantalla principal del aplicativo. 4. El usuario elige opción: ELEGIR TIPO INCIDENCIA SOCIAL. 5. El sistema muestra una lista de tipos de incidencias. 6. El usuario elige el tipo de incidencia a reportar. 7. El sistema muestra: Información de la Incidencia. 8. El usuario debe ingresar información detallada o general de la incidencia que está reportando. 9. El sistema muestra el botón: ENVIAR DATOS. 10. El usuario presiona dicho botón para enviar la incidencia al sistema web de la municipalidad. 11. El sistema devuelve el siguiente mensaje al usuario: GRACIAS + código de la incidencia reportada.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Luego del punto 7 del flujo normal, el sistema muestra 5 opciones adicionales: MAPA, FOTO, INMUEBLE, PERSONA, AUTOS. 2. El usuario la opción MAPA. 3. El sistema muestra la ubicación de la incidencia.

4. El usuario da un click en la ubicación de la incidencia mostrada por google maps.
5. El sistema muestra los botones **GUARDAR y CANCELAR.**
6. El usuario guardar o cancelar la información de la ubicación.
7. El sistema vuelve a la pantalla principal.
8. El usuario elige la opción **FOTO.**
9. El sistema muestra tres opciones: **OPCIÓN MULTI-FOTO, TOMAR FOTO, SELECCIONAR UNA FOTO.**
10. El usuario selecciona una de las tres opciones de FOTO.
11. El sistema muestra los botones **ACEPTAR y BORRAR.**
12. El usuario acepta o borra la captura de la foto.
13. El sistema vuelve a la pantalla principal.
14. El usuario elige la opción **INMUEBLE.**
15. El sistema muestra cajas de texto para ingresar información detallada del inmueble a reportar.
16. El usuario llena dichas cajas de texto.
17. El sistema muestra los botones: **AGREGAR ó CANCELAR.**
18. El usuario según criterio propio agrega o cancela la información.
19. El sistema vuelve a la pantalla principal.
20. El usuario elige la opción **PERSONA.**
21. El sistema muestra cajas de texto para ingresar información detallada de la persona a reportar.
22. El usuario llena dichas cajas de texto.
23. El sistema muestra los botones: **AGREGAR ó CANCELAR.**
24. El usuario según criterio propio agrega o cancela la información.
25. El sistema vuelve a la pantalla principal.
26. El usuario elige la opción **AUTO.**
27. El sistema muestra la opción: **SELECCIONE UNA OPCIÓN.**
28. El usuario elige dicha opción.
29. El sistema muestra tres opciones: **AUTO, CAMIONETA y CAMIÓN.**
30. El usuario elige una de las tres opciones.
31. El sistema muestra cajas de texto para ingresar información detallada del auto a reportar.
32. El usuario llena dichas cajas de texto.
33. El sistema muestra los botones: **AGREGAR ó CANCELAR.**
34. El usuario según criterio propio agrega o cancela la información.
35. El sistema vuelve a la pantalla principal.
36. Luego del punto 7, 13, 19, 25 ó 35 del flujo alternativo se realiza el punto 9 del flujo normal.

Poscondiciones:

- La incidencia debe registrarse en el sistema web.

Nombre:	Atender incidencia
Autor:	Roy Neyra
Descripción:	Caso de uso que describe el proceso de atender la incidencia registrada por el usuario desde un dispositivo móvil o web al órgano competente. Atender la incidencia implica dejar en vista, descartar o derivar la incidencia.
Actores:	Administrador web.
Precondiciones:	Que se registre la incidencia.

<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El administrador web carga al sistema web de incidencias sociales. 2. El sistema pide ingresar usuario y password para tener acceso a el. 3. El administrador web ingresa usuario y password correctos. 4. El sistema muestra la pantalla principal del sistema web con las opciones: Denuncias Móvil y Denuncias Web. 5. El administrador web elige una de las dos opciones: Denuncias Móvil y Denuncias Web. 6. El sistema muestra las incidencias reportadas por el usuario.
<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 1. El sistema muestra las incidencias sociales reportadas por el usuario del sistema móvil y web con la opción: ATENDER. 2. El administrador atiende las incidencias presionando la opción atender. 3. El sistema muestra una pantalla con el detalle de la incidencia más los botones: ATENDER y DESCARTAR. <ol style="list-style-type: none"> a. Si el administrador web elige la opción descartar. b. El sistema muestra otra pantalla para indicar el motivo del descarte y luego muestra la opción GRABAR descarte. c. Si el administrador web elige la opción atender. d. El sistema muestra otra pantalla ATENDIENDO INCIDENCIA con opciones de INTERNO y EXTERNO más el botón FINALIZAR ATENCIÓN. e. El administrador web según sea el tipo de incidencia elige interno o externo que viene hacer una incidencia será atendida por una dependencia interna de la municipalidad o un organismo externo de la municipalidad según tipo de incidencia. f. El sistema muestra por cada opción interno o externo un combo de tres opciones. g. El administrador web elige la opción deseada y luego el botón FINALIZAR ATENCIÓN.
<p>Poscondiciones:</p> <ul style="list-style-type: none"> - La incidencia debe ser atendido luego de ser derivado.

Nombre:	Generar reportes.
Autor:	Roy Neyra
Descripción:	Caso de uso que describe el proceso de generar reportes estadísticos para la toma de decisiones a la gerencia municipal o a la oficina competente.
Actores:	Administrador web.
Precondiciones:	Que la incidencia social haya sido derivada.
Flujo Normal:	<ol style="list-style-type: none"> 1. Área usuaria solicita reporte estadístico al administrador web, por fecha o tipo de incidencia. 2. Administrador web genera reporte estadístico en excel de acuerdo a lo solicitado por el área usuaria.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Que no se haya producido incidencia alguna en la fecha o tipo de incidencia solicitada.
Poscondiciones:	<ul style="list-style-type: none"> - El reporte debe generarse en Excel al área solicitante.

4.8. Especificaciones Adicionales.

Sigla	Tipo de requerimiento		Descripción
F	Functional	Funcional	<ul style="list-style-type: none"> Hacer uso de forma nativa del sistema Operativo Android 5.1 o superior. Una intercomunicación entre el producto móvil y el producto web de forma transparente, a través de web services. El envío de datos en forma anónima. Eficiencia en la reducción en tiempo de respuesta a una incidencia registrada.
U	Usability	Facilidad de uso	<ul style="list-style-type: none"> Poseer una curva de aprendizaje lo suficientemente corta (una hora), esto permite a los operadores del sistema poder realizar su trabajo de forma eficiente. Tener controles Web con tecnología HTML5, la cual es fácil e intuitiva. Utilizar colores (azul y blanco) que representen a la institución. Tener documentación de desarrollo de software como: diagramas UML, base de datos, etc.
R	Reliability	Fiabilidad	<ul style="list-style-type: none"> Manejar las incidencias registradas con un ID único para cada incidencia. Si un usuario de la app móvil cierra la pantalla donde está trabajando, no debe perder los datos que ya ha ingresado. Ingresar la fecha y hora exacta de una incidencia. Tener un registro de las atenciones de incidencias.
P	Performance	Rendimiento	<ul style="list-style-type: none"> Debe permitir la atención de una incidencia, reflejándose esta acción en tiempo real. Los tiempos de respuesta no deben ser superior a los 3.5 segundos Permite generar una estadística en tiempo real de los datos.
S	Supportability	Soporte	<ul style="list-style-type: none"> La plataforma web es independiente del sistema operativo, utilizando un navegador web. La app móvil es independiente de la marca del equipo celular, para este caso debe ser con sistema operativo Android 5.1 o superior. Los productos deben ser modulares y debe permitir su mantenimiento en el tiempo. La plataforma web debe ser de fácil configuración, permitiendo registrar los datos más importantes en una base de datos.

Tabla No. 17: *FURPS del sistema informático.*

4.9. Prototipos de Interfaces de Usuario.

Caso de Uso: Registrar incidencias sociales (Móvil)



Su Información siempre es Importante
M. FLORENCIA DE MORA

TRABAJANDO PARA USTED !.

👤 Déjenos su Nombre (Opcional)

☎ Déjenos su Teléfono (Opcional)

SELECCIONE TIPO INCIDENCIA SOCIAL

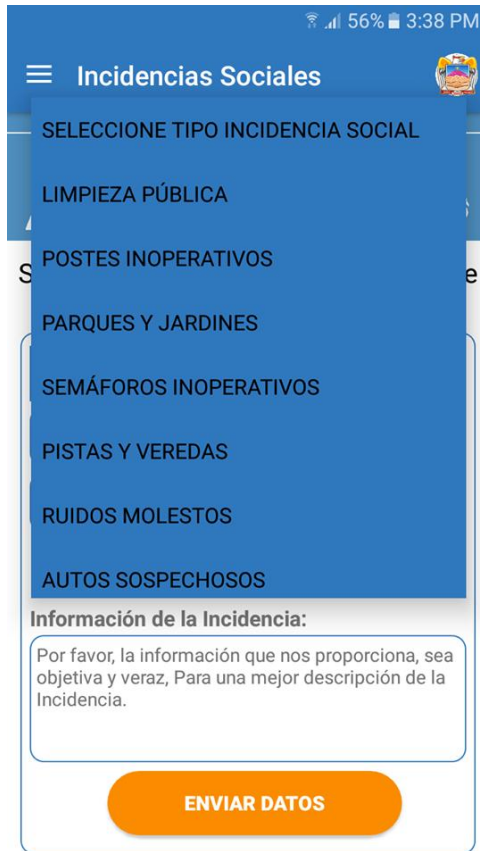
Información de la Incidencia:

Por favor, la información que nos proporciona, sea objetiva y veraz, Para una mejor descripción de la Incidencia.

ENVIAR DATOS

Opciones que pueden acompañar al reporte de una incidencia para hacerla más descriptiva. Son opcionales.

Ilustración No. 18: Pantalla principal del aplicativo móvil



Incidentes Sociales

SELECCIONE TIPO INCIDENCIA SOCIAL

- LIMPIEZA PÚBLICA
- POSTES INOPERATIVOS
- PARQUES Y JARDINES
- SEMÁFOROS INOPERATIVOS
- PISTAS Y VEREDAS
- RUIDOS MOLESTOS
- AUTOS SOSPECHOSOS

Información de la Incidencia:

Por favor, la información que nos proporciona, sea objetiva y veraz, Para una mejor descripción de la Incidencia.

ENVIAR DATOS

Ilustración No. 19: *Muestra tipo de incidencia a seleccionar.*



Incidentes Sociales

USTED PUEDE AGREGAR CUALQUIERA DE LOS SIGUIENTES DATOS

MAPA FOTO INMUEBLE PERSONA AUTOS

Su Información siempre es Importante
M. FLORENCIA DE MORA

TRABAJANDO PARA USTED !.

Déjenos su Nombre (Opcional)

Déjenos su Teléfono (Opcional)

LIMPIEZA PÚBLICA

Información de la Incidencia:

Por favor, la información que nos proporciona, sea objetiva y veraz, Para una mejor descripción de la Incidencia.

ENVIAR DATOS

Ilustración No. 20: *Muestra tipo de incidencia elegida, la cual puede ser acompañada de información sobre ubicación (mapa), foto, inmueble, persona o auto. Además, opcionalmente se puede escribir el nombre y teléfono de la persona que reporta el hecho. Y obligatoriamente escribiremos información de la incidencia.*



Ilustración No. 21: *Muestra la opción mapa para elegir ubicación de la incidencia.*



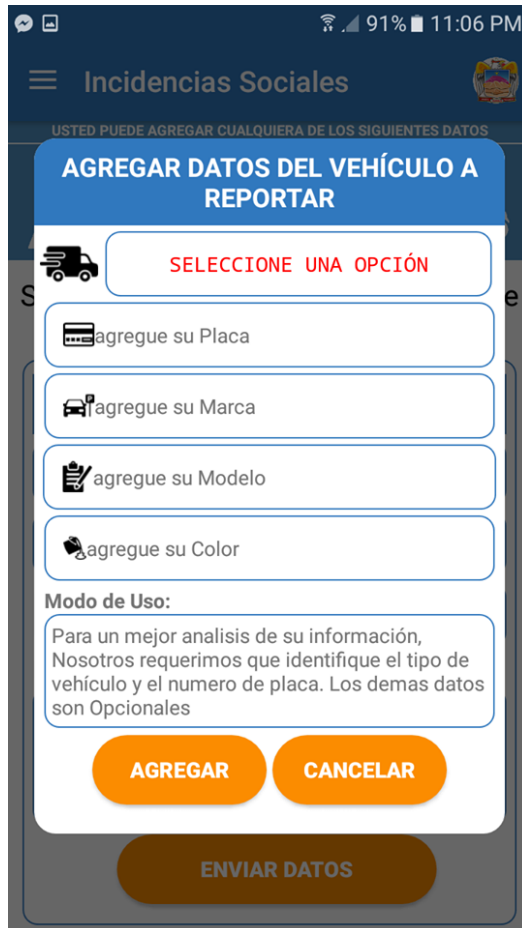
Ilustración No. 22: *Muestra las alternativas de la opción foto.*

The screenshot shows the 'Incidencias Sociales' app interface. At the top, the status bar shows 68% battery and 10:00 AM. The app title is 'Incidencias Sociales'. Below the title, a navigation bar lists options: MAPA, FOTO, INMUEBLE, PERSONA, and AUTOS. A blue modal box is open with the title 'AGREGAR DATOS DEL INMUEBLE A REPORTAR'. It contains two input fields: 'Agregar Dirección' with a house icon and 'Agregar Color' with a color palette icon. Below these is a 'Modo de Uso:' section with the text: 'Para un mejor uso de su información debe proporcionarnos la dirección del inmueble, los demás datos son Opcionales.' At the bottom of the modal are two orange buttons: 'AGREGAR' and 'CANCELAR'. Below the modal, a text area is partially visible with the text 'objetiva y veraz, Para una mejor descripción de la Incidencia.' and a large brown 'ENVIAR DATOS' button at the very bottom.

Ilustración No. 23: *Muestra agregar datos de la opción inmueble.*

The screenshot shows the 'Incidencias Sociales' app interface. At the top, the status bar shows 91% battery and 11:06 PM. The app title is 'Incidencias Sociales'. Below the title, a navigation bar lists options: MAPA, FOTO, INMUEBLE, PERSONA, and AUTOS. A blue modal box is open with the title 'AGREGAR DATOS DE LA PERSONA A REPORTAR'. It contains three input fields: 'agregue su nombre' with a person icon, 'agregue Apellidos' with a group of people icon, and 'agregue sus Apelativos' with a person icon. Below these is a 'Modo de uso:' section with the text: 'Para una mejor información, necesitamos que nos proporcione los nombres y apellidos, así como el apelativo de la persona.' At the bottom of the modal are two orange buttons: 'AGREGAR' and 'CANCELAR'. Below the modal, a text area is partially visible with the text 'Incidencia.' and a large brown 'ENVIAR DATOS' button at the very bottom.

Ilustración No. 24: *Muestra agregar datos de la opción persona.*



Incidentes Sociales

USTED PUEDE AGREGAR CUALQUIERA DE LOS SIGUIENTES DATOS

AGREGAR DATOS DEL VEHÍCULO A REPORTAR

SELECCIONE UNA OPCIÓN

agregue su Placa

agregue su Marca

agregue su Modelo

agregue su Color

Modo de Uso:
Para un mejor analisis de su información, Nosotros requerimos que identifique el tipo de vehículo y el numero de placa. Los demas datos son Opcionales

AGREGAR CANCELAR

ENVIAR DATOS

Ilustración No. 25: Muestra agregar datos de la opción vehículo.



Incidentes Sociales

USTED PUEDE AGREGAR CUALQUIERA DE LOS SIGUIENTES DATOS

MAPA FOTO INMUEBLE PERSONA AUTOS

Su Información siempre es Importante
M. FLORENCIA DE MORA

TRABAJANDO PARA USTED !.

Roy

999103277

LIMPIEZA PÚBLICA

Información de la Incidencia:
Acumulación de desmonte

ENVIAR DATOS

Ilustración No. 26: Ingresar nombre y teléfono (opcionalmente) de la persona que reporta el hecho y obligatoriamente ingresar información de la incidencia. Luego enviar datos.

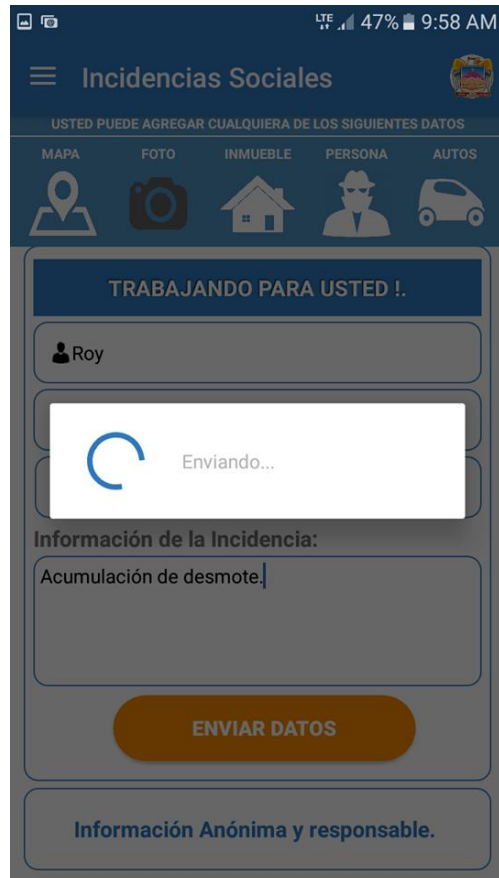


Ilustración No. 27: *Enviando datos de la incidencia al sistema web.*



Ilustración No. 28: *Datos enviados correctamente al sistema web. A la vez retorna un código de la incidencia para cualquier consulta posterior de parte de la persona que reportó la incidencia.*

Caso de Uso: Derivar registros de incidencias (Web)

CENTRAL INFORMÁTICA - MUNICIPALIDAD DE FLORENCIA DE MORA Conectado: admin admin admin




GESTIÓN DE INCIDENCIAS SOCIALES

CENTRAL INFORMÁTICA

GESTIÓN DE INCIDENCIAS SOCIALES

SISTEMA DE DENUNCIAS SOCIALES

(1 of 1) [1] 10

ATENDER	ID	FECHA	EMAIL	NOMBRE	TELEF	IMAGEN	MAPA
VER	9	28/09/2017 11:03:20	incidenciamdfm2017@gmail.com				
ATENDER	8	15/09/2017 10:39:42	jlap.1525@gmail.com			VER	
ATENDER	7	06/09/2017 11:18:34	jlap.1525@gmail.com	Luis		VER	
VER	6	05/09/2017 09:25:55	jlap.1525@gmail.com	Jorge		VER	
ATENDER	5	02/09/2017 03:25:13	jlap.1525@gmail.com	Jorge Angulo	949657123	VER	
ATENDER	4	25/08/2017 10:12:33	incidenciamdfm2017@gmail.com				
VER	3	22/08/2017 12:10:56	incidenciamdfm2017@gmail.com	Roy Neyra	999103277		VER
VER	2	22/08/2017 10:08:05	incidenciamdfm2017@gmail.com				
VER	1	22/08/2017 10:05:36	incidenciamdfm2017@gmail.com			VER	

(1 of 1) [1] 10

Se encontraron un total de: 9 elementos.

LEIDO ATENDER DESCARTAR

Ilustración No. 29: Muestra todas las incidencias enviadas desde un dispositivo móvil.

45.55.188.187/GIP-war/pages/denunciasweb.shtml?idx=2

CENTRAL INFORMÁTICA - MUNICIPALIDAD DE FLORENCIA DE MORA

Conectado: admin admin admin

GESTIÓN DE INCIDENCIAS PÚBLICAS

CENTRAL INFORMÁTICA

GESTION DE INFORMACIÓN PÚBLICA

[Inicio](#)
[Denuncias Móvil](#)
[Denuncias Web](#)

MUNICIPALIDAD DE FLORENCIA DE MORA
 SISTEMA INTEGRADO DE GESTIÓN DE INCIDENCIAS PÚBLICAS.

DENUNCIAS SOCIALES WEB

(1 of 1) 1 10

ATENDER	ID	FECHA	ASUNTO	CORREO	IMAGEN
ATENDER	99	03/07/2017 08:59:05	Caída de postes	edwinroy@hotmail.com	VER

(1 of 1) 1 10

Se encontraron un total de: 1 elementos.

[LEIDO](#)
[ATENDER](#)
[DESCARTAR](#)

Ilustración No. 30: Muestra todas las incidencias enviadas desde la página web de la municipalidad de Florencia de Mora.

45.55.188.187/GIP-war/pages/denuncias.xhtml?idx=1

CENTRAL INFORMÁTICA - MUNICIPALIDAD DE FLORENCIA DE MORA

Conectado: admin admin admin

GESTIÓN DE INCIDENCIAS PÚBLICAS

CENTRAL INFORMÁTICA

GESTION DE INFORMACIÓN PÚBLICA

Inicio

Denuncias Móvil

Denuncias Web

MUNICIPALIDAD DE FLORENCIA DE MORA
SISTEMA INTEGRADO DE GESTIÓN DE INCIDENCIAS PÚBLICAS.

SISTEMA DE DENUNCIAS SOCIALES

ATENDER	ID	FECHA	
ATENDER	188	21/08/2017 10:58:31	inc
ATENDER	187	21/08/2017 10:54:05	inc
ATENDER	186	17/08/2017 01:24:32	inc
VER	185	17/08/2017 12:11:05	inc
ATENDER	184	06/08/2017 05:01:30	
ATENDER	183	04/08/2017 07:19:50	inc
ATENDER	182	04/08/2017 07:10:05	inc
VER	181	04/08/2017 12:53:33	inc
ATENDER	180	15/07/2017 07:21:06	jlap.1525@gmail.com Jorge
ATENDER	179	12/07/2017 08:19:27	jlap.1525@gmail.com jorge

ATENDE DENUNCIA

CÓDIGO: 188
FECHA: 21/08/2017 09:58:31

DETALLE:
Acumulaci^on de desmote.

ATENDER DESCARTAR

DETALLE	TELEF	IMAGEN	MAPA
Acumulaci ^o n de desmote.	999103277	VER	
Acumulaci ^o n de desmote	999103277	VER	
ruidos excesivos			
poste ca ^o do hace un d ^o a.	999103277		VER
Basura acumulada			
poste ca ^o do		VER	
pista malograda	999103277	VER	
ruidos molestos y beben licor			
Sospechoso			
robo		VER	

(1 of 3) 1 2 3 10

Ilustración No. 31: *Atiende la incidencia.*

CENTRAL INFORMÁTICA - MUNICIPALIDAD DE FLORENCIA DE MORA Conectado: admin admin admin

GESTIÓN DE INCIDENCIAS SOCIALES

CENTRAL INFORMÁTICA

GESTIÓN DE INCIDENCIAS SOCIALES

Inicio

Denuncias Móvil

Denuncias Web

MUNICIPALIDAD DE FLORENCIA DE MORA
SISTEMA INTEGRADO DE GESTIÓN DE INCIDENCIAS PÚBLICAS.

SISTEMA DE DENUNCIAS SOCIALES

ATENDER	ID	FECHA
ATENDER	3	22/08/2017 12:10:56
ATENDER	2	22/08/2017 10:08:05
ATENDER	1	22/08/2017 10:05:36

Se encontraron un total de: 3

ATENDER DENUNCIA

CÓDIGO: 3
FECHA: 22/08/2017 11:10:56

DETALLE: Robaron a u... esa cdra.

ATENDIENDO INCIDENCIA

DESTINO FINAL INTERNO EXTERNO

OPCION DISPONIBLE

FINALIZAR ATENCIÓN

NOMBRE	TELEF	IMAGEN	MAPA
Roy Neyra	999103277		VER
			VER

Ilustración No. 32: La incidencia es derivada a una oficina interna de la municipalidad o a un órgano externo a ella.

CENTRAL INFORMÁTICA - MUNICIPALIDAD DE FLORENCIA DE MORA Conectado: admin admin admin

GESTIÓN DE INCIDENCIAS SOCIALES

GESTION DE INCIDENCIAS SOCIALES

- Inicio
- Denuncias Móvil
- Denuncias Web

MUNICIPALIDAD DE FLORENCIA DE MORA
SISTEMA INTEGRADO DE GESTION DE INCIDENCIAS PÚBLICAS.

SISTEMA DE DENUNCIAS SOCIALES			
ATENDER	ID	FECHA	
ATENDER	3	22/08/2017 12:10:56	
ATENDER	2	22/08/2017 10:08:05	
ATENDER	1	22/08/2017 10:05:36	

Se encontraron un total de: 3

NOMBRE	TELEF	IMAGEN	MAPA
Roy Neyra	999103277		VER
			VER

ATENDER DENUNCIA

CÓDIGO: 3
FECHA: 22/08/2017 11:10:56

DETALLE: Robaron a u

ATENDIENDO INCIDENCIA

DESTINO FINAL INTERNO EXTERNO

OPCION DISPONIBLE: OPCION WS 1

Caja de texto: Caída de postes en la Av. 26 de Marzo s.d.r.a 05

SINOPSIS DE LA INCIDENCIA

FINALIZAR ATENCIÓN

Ilustración No. 33: Se puede hacer una descripción breve de la incidencia antes de derivarse.

Caso de Uso: Reporte de registros de incidencias (Web)

The screenshot displays the 'GESTIÓN DE INCIDENCIAS PÚBLICAS' web application. The main interface includes a navigation menu on the left with options like 'Inicio', 'Denuncias Móvil', and 'Denuncias Web'. The central area shows a table of social incidents under the heading 'SISTEMA DE DENUNCIAS SOCIALES'. A modal window titled 'ATENDER DENUNCIA' is open, showing details for incident 188, including a 'Word' icon for reporting. A detailed view of the incident is also visible, showing a table with columns for 'DETALLE', 'TELEF', 'IMAGEN', and 'MAPA'. Red circles highlight the 'ID' and 'FECHA' columns in the incident list, the 'Word' icon in the modal, and the 'IMAGEN' button in the details view.

ATENDER	ID	FECHA	INCIDENCIA
VER	188	21/08/2017 10:58:31	inc
ATENDER	187	11/08/2017 10:54:05	inc
ATENDER	186	17/08/2017 01:24:32	inc
VER	185	17/08/2017 12:11:05	inc
ATENDER	184	06/08/2017 05:01:30	inc
ATENDER	183	04/08/2017 07:19:50	inc
ATENDER	182	04/08/2017 07:10:05	inc
VER	181	04/08/2017 12:53:33	incidenciamdfm2017@gmail.com
ATENDER	180	15/07/2017 07:21:06	jlap.1525@gmail.com Jorge
		12/07/2017	jlap.1525@gmail.com jorge

DETALLE	TELEF	IMAGEN	MAPA
Acumulación de desmote.	999103277	VER	
Acumulación de desmote	999103277	VER	
ruidos excesivos			
poste caído hace un día.	999103277		VER
Basura acumulada			
poste caído		VER	
pista malograda	999103277	VER	
ruidos molestos y beben licor			
Sospechoso			
robo		VER	

Ilustración No. 34: Muestra el ícono para generar reporte en Word de la incidencia.

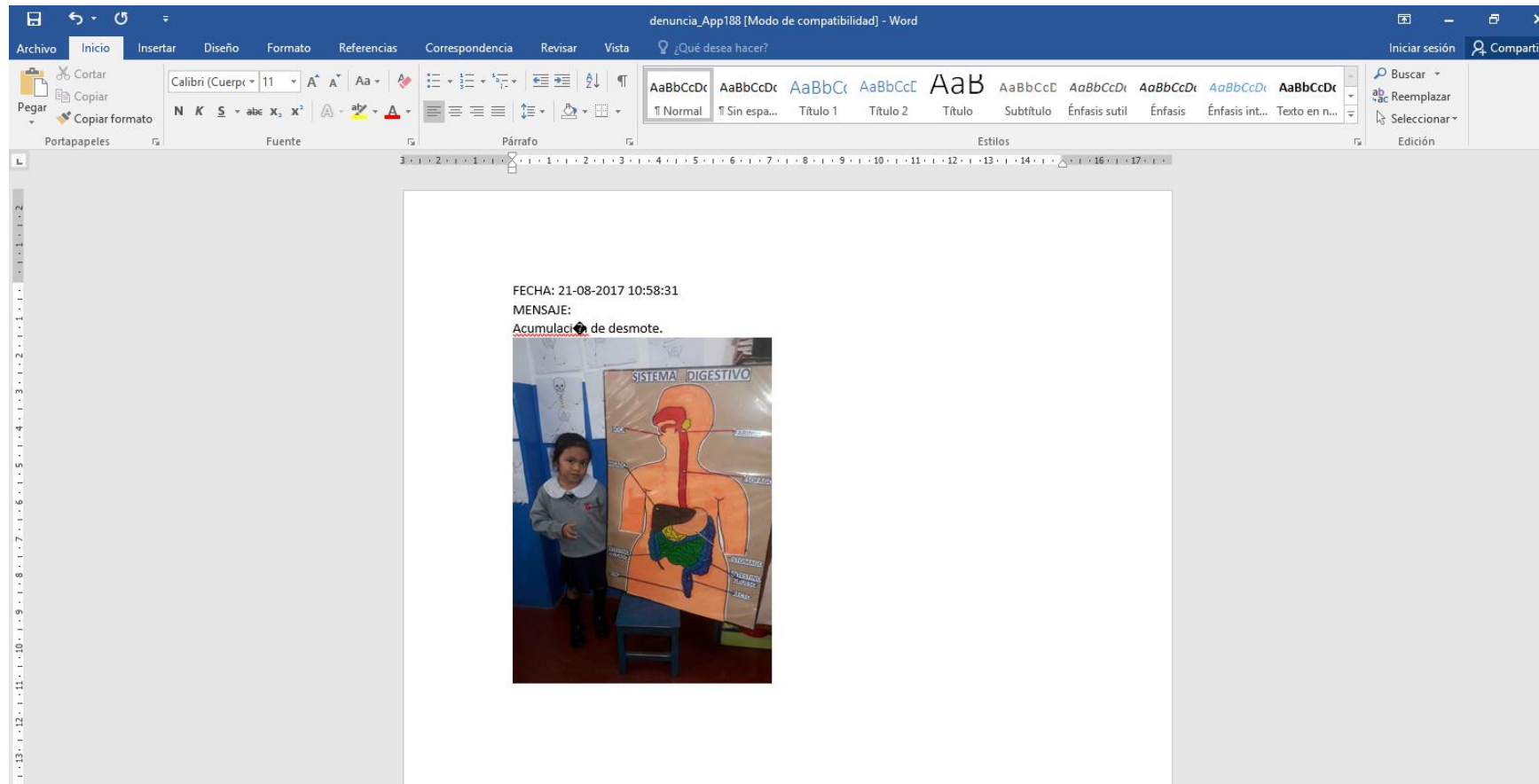


Ilustración No. 35: Muestra el reporte de la incidencia con los campos encerrados en círculos de color rojo mostrados en la Pantalla 16.

4.10. Diagramas de Secuencia.

Diagrama de secuencia: Registrar incidencia.

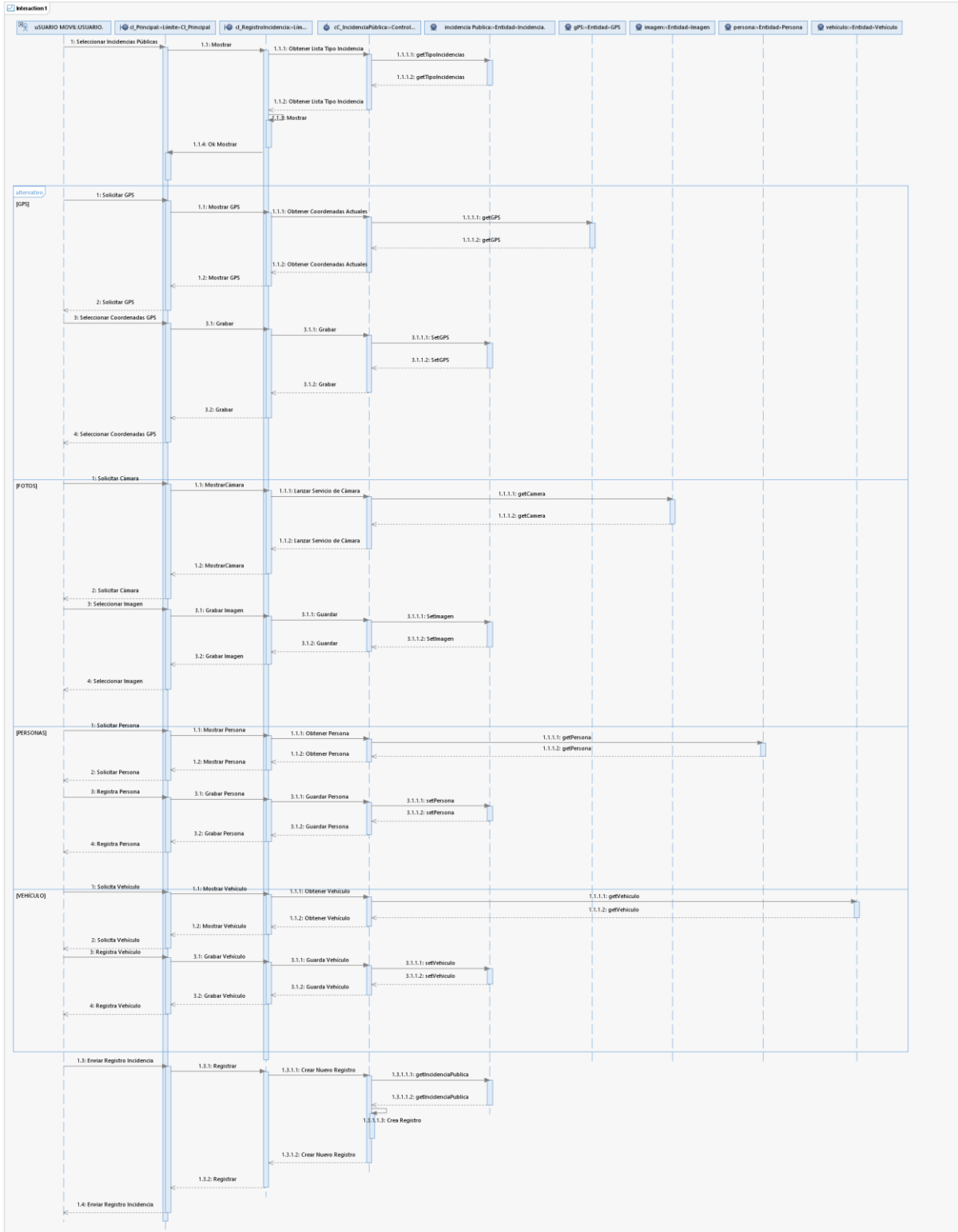


Ilustración No. 36: Diagrama de secuencia general.

Diagrama de Secuencia: Derivar incidencia.

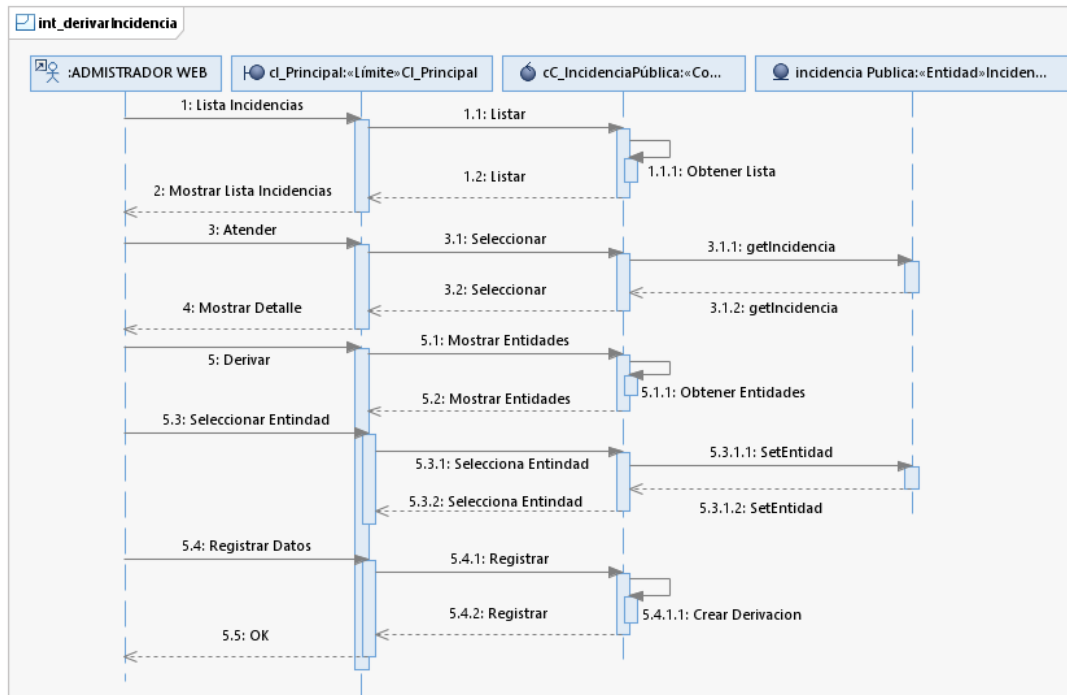


Ilustración No. 37: Diagrama de secuencia derivar incidencia.

Diagrama de Secuencia: Generar Reporte.

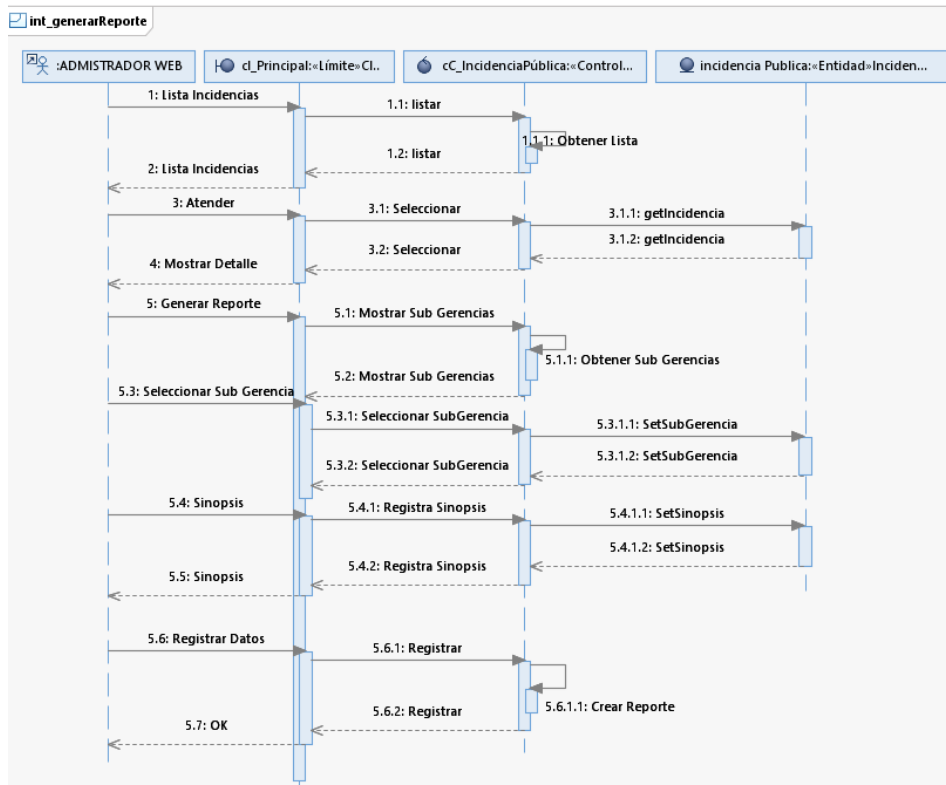


Ilustración No. 38: Diagrama de secuencia generar reporte.

4.11. Diagramas de Estado.

Diagrama de estado Incidencia Social.

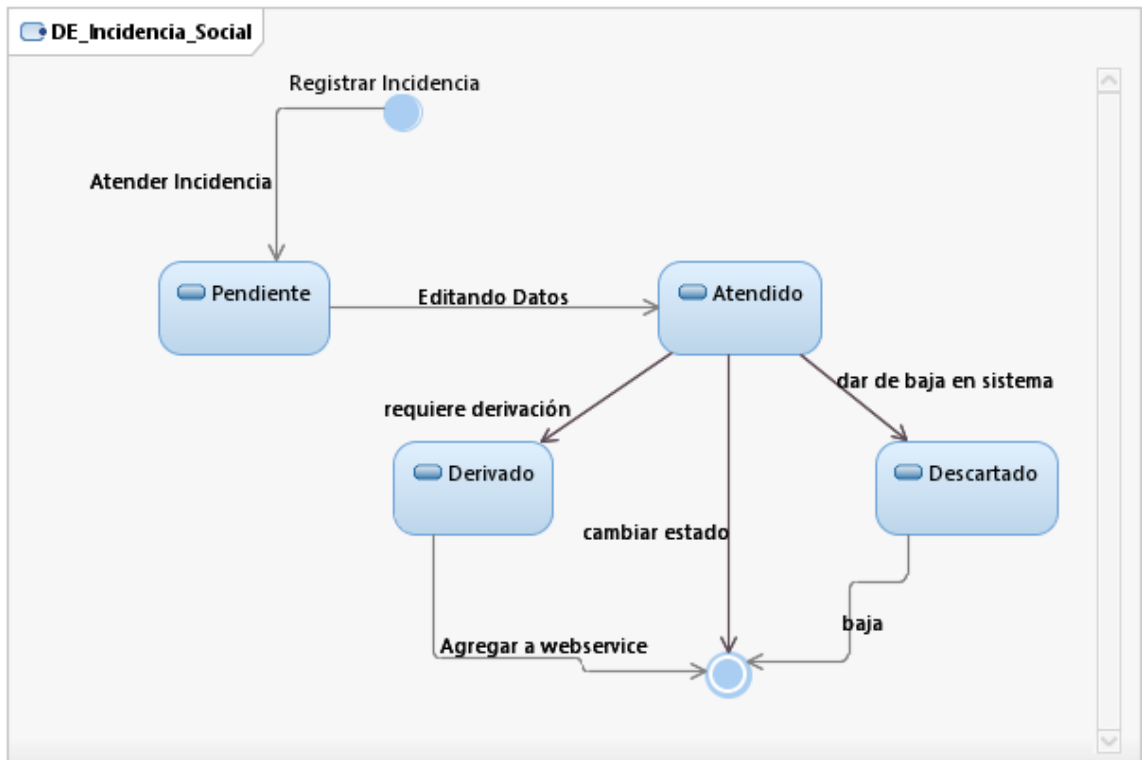


Ilustración No. 39: *Diagrama de estado de incidencia social.*

Diagrama de estado Usuario.

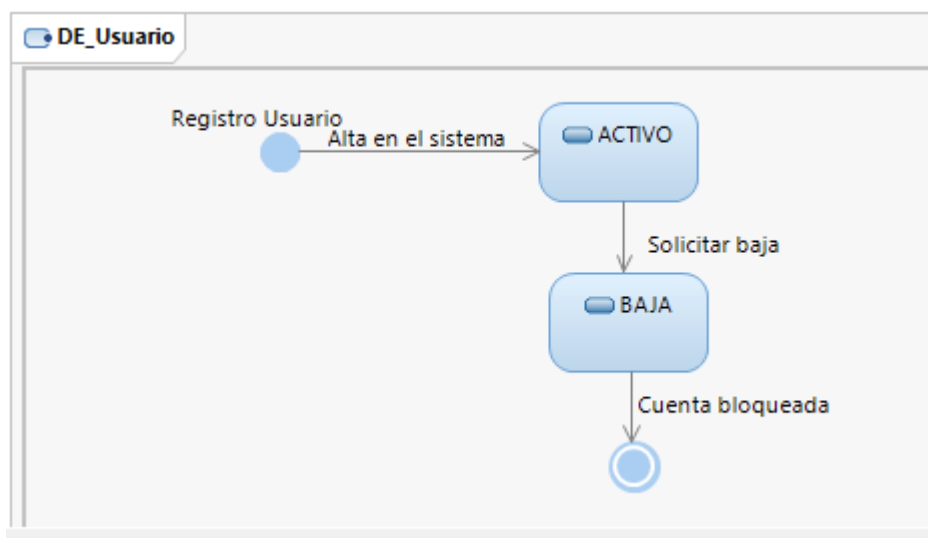


Ilustración No. 40: *Diagrama de estado de usuario.*

Diagrama de estado Entidad Externa.



Ilustración No. 41: *Diagrama de estado de entidad externa.*

Diagrama de estado Jefe Área.

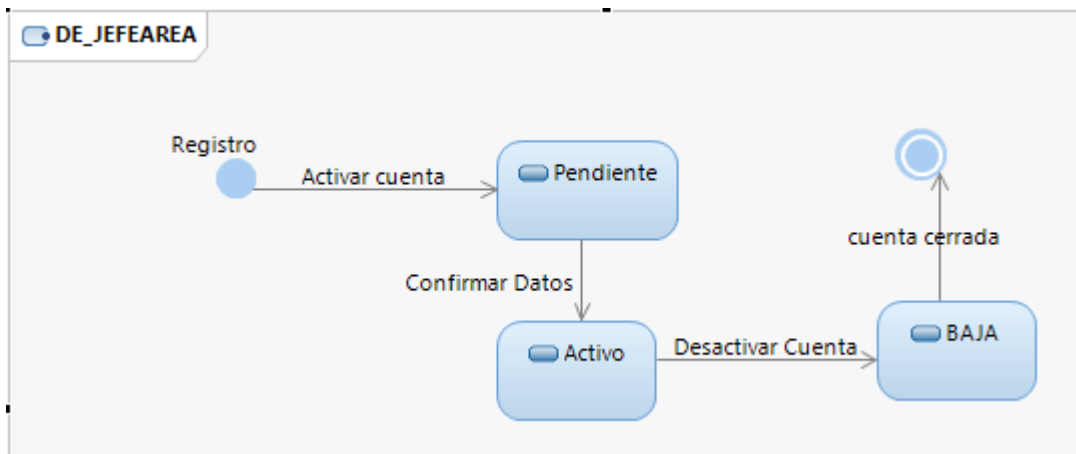


Ilustración No. 42: *Diagrama de estado de jefe de área.*

4.12. Modelo de Análisis y Diseño

4.12.1. Capas de la Arquitectura.

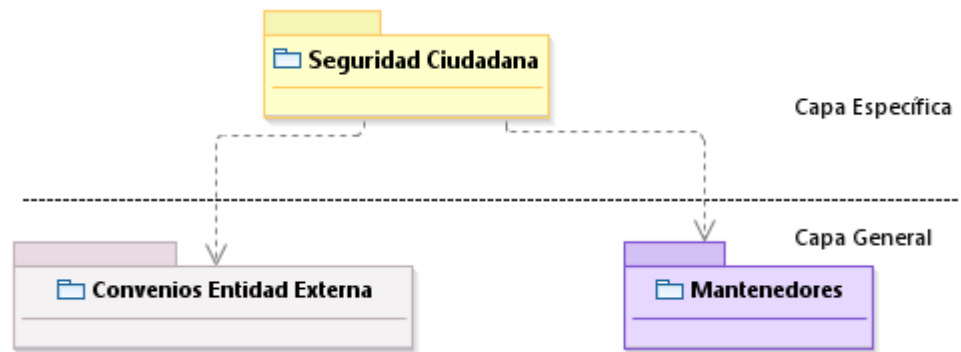


Ilustración No. 43: *Capas de arquitectura de sistema.*

4.12.2. Diagramas de Clases de Análisis del Sistema.

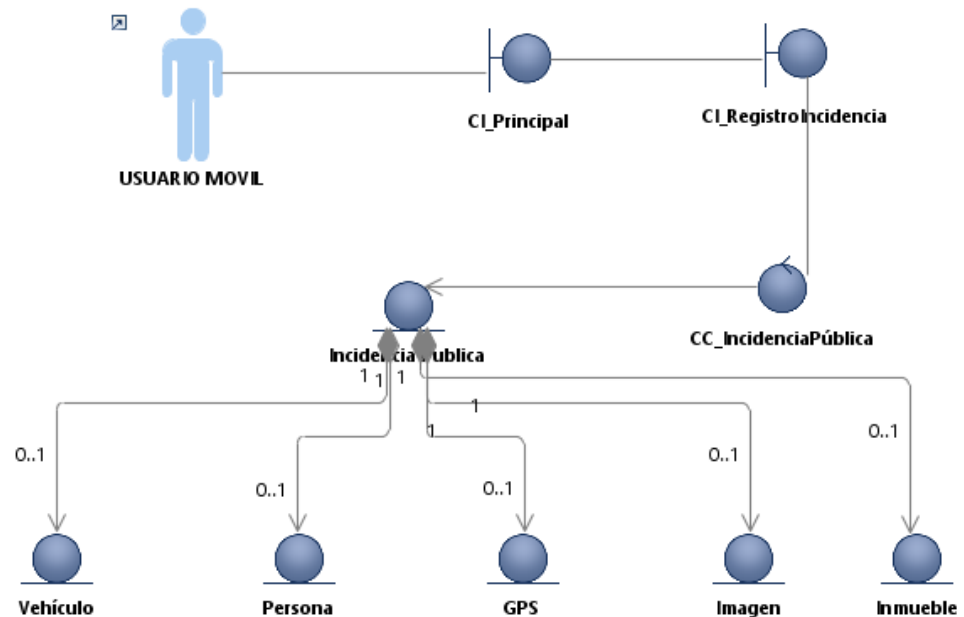


Ilustración No. 44: *Diagrama de clases de análisis de sistema.*

4.12.3. Patrones de Diseño

Composite View (Capa de presentación)

Usamos el patrón composite view cuando deseamos utilizar múltiples subvistas, cada una de las subvistas pueden ser incluidas, de esta forma se crea una página con un layout que puede ser manejado de forma independiente de su contenido.

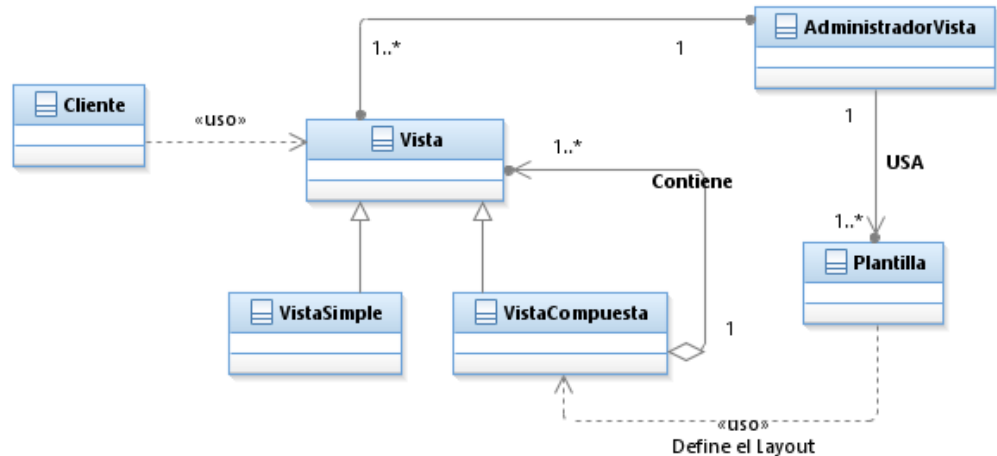


Ilustración No. 45: Diagrama del patrón de diseño – Capa de presentación.

Session Facade (Capa de negocio)

El uso de Session Facade permite encapsular los componentes de la capa de negocio y exponer de forma granular un servicio a todos los clientes remotos.

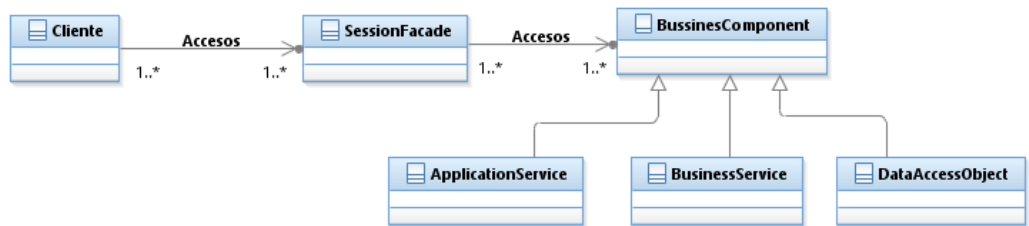


Ilustración No. 46: Diagrama del patrón de diseño – Capa de negocio.

Data Access Object (Capa de acceso a datos)

Se usa Data access Object para abstraer y encapsular todos los accesos a un almacenamiento persistente, Data Access object maneja la conexión con el data source para obtener y almacenar datos.

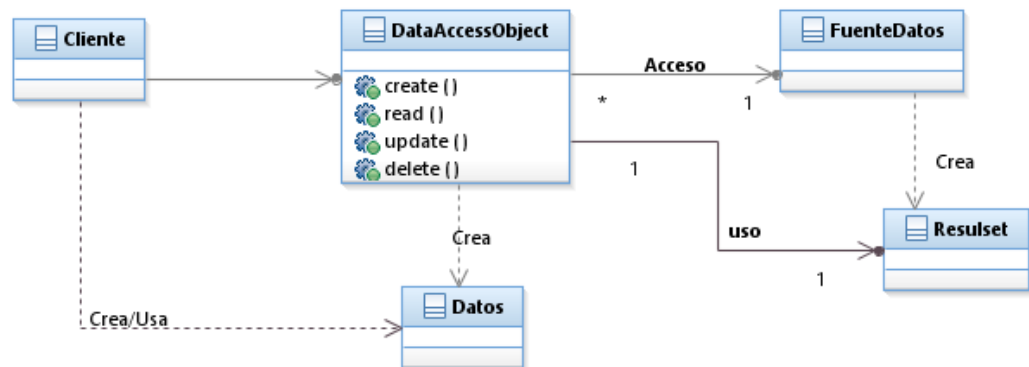


Ilustración No. 47: Diagrama del patrón de diseño – Capa de acceso a datos.

4.13. Modelo de Datos.

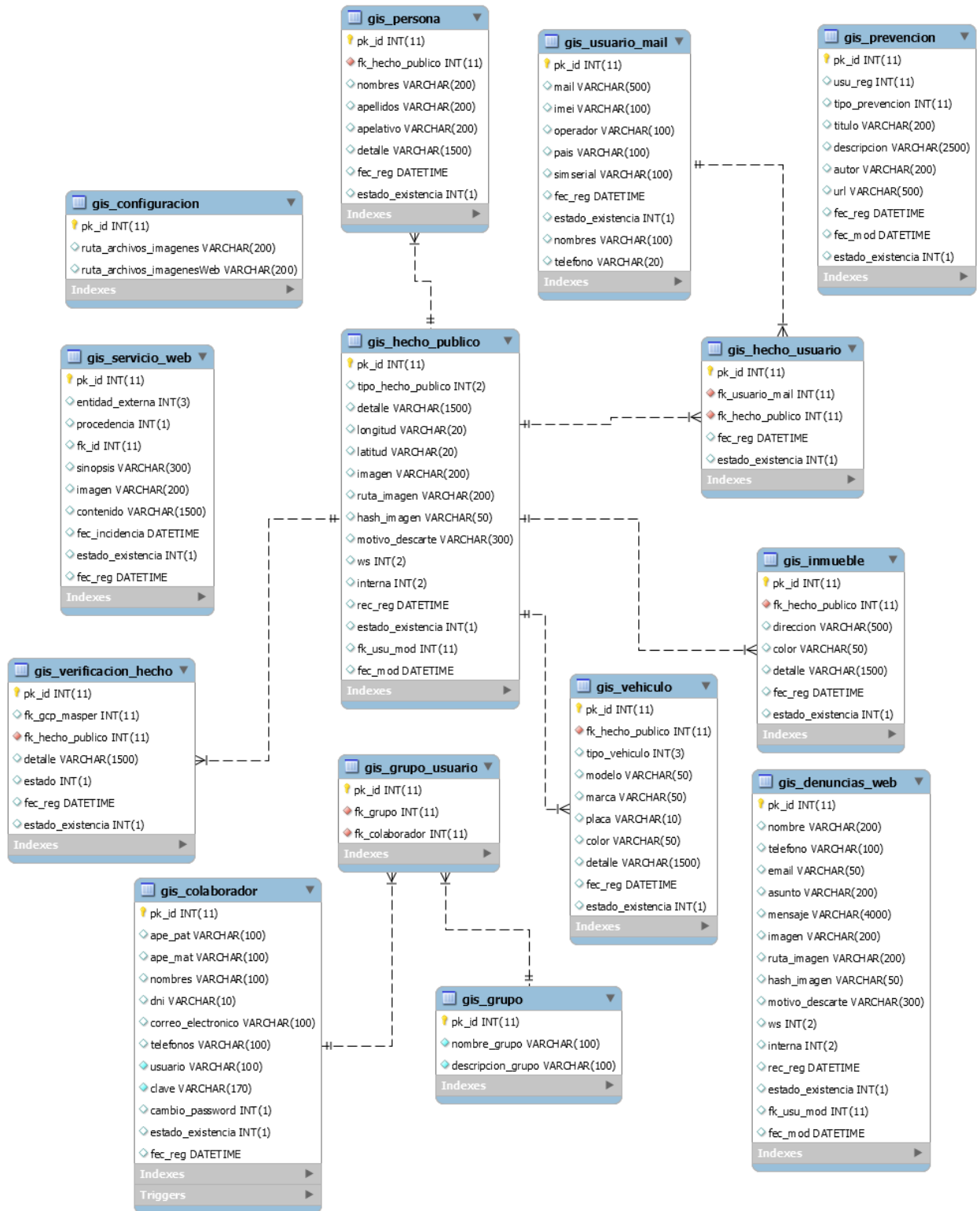


Ilustración No. 49: Diseño de modelo de datos del sistema informático

4.14. Modelo de Despliegue.

4.14.1. Capas con sus Componentes

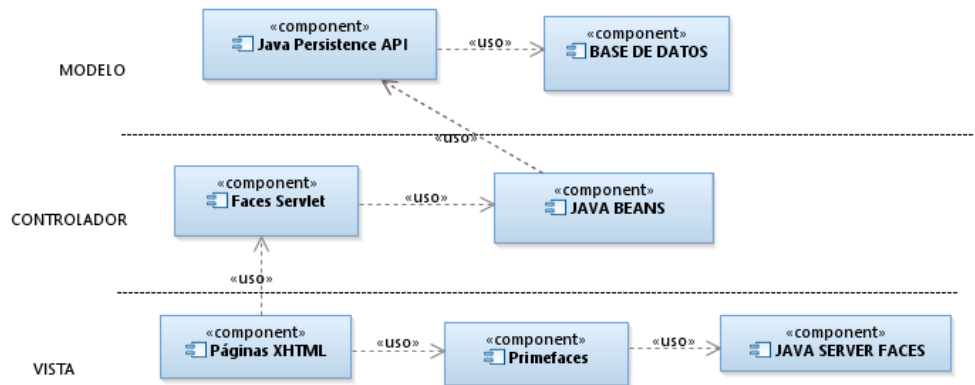


Ilustración No. 50: Diagrama de capas con sus componente del sistema.

4.14.2. Diagrama de Componentes

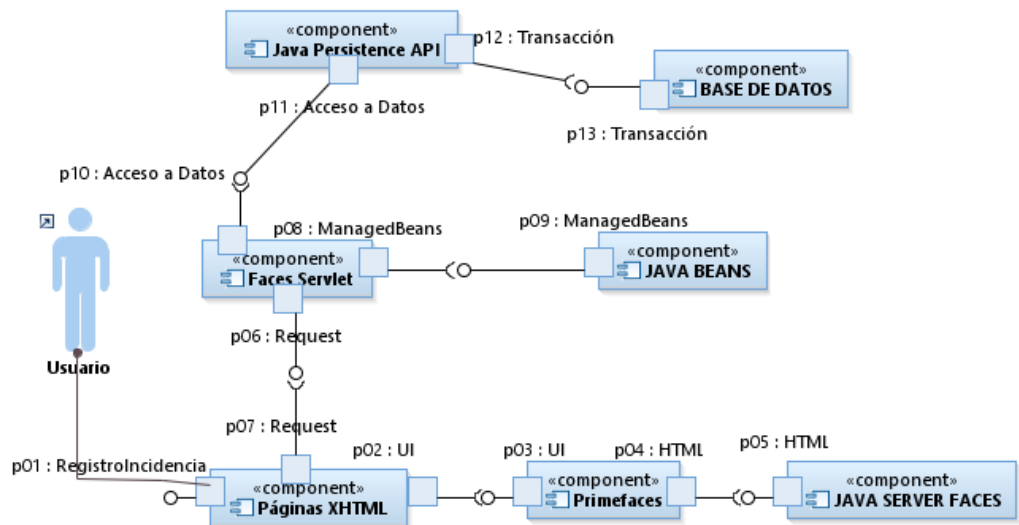


Ilustración No. 51: Diagrama de componente del sistema.

4.14.3. Diagrama de Despliegue

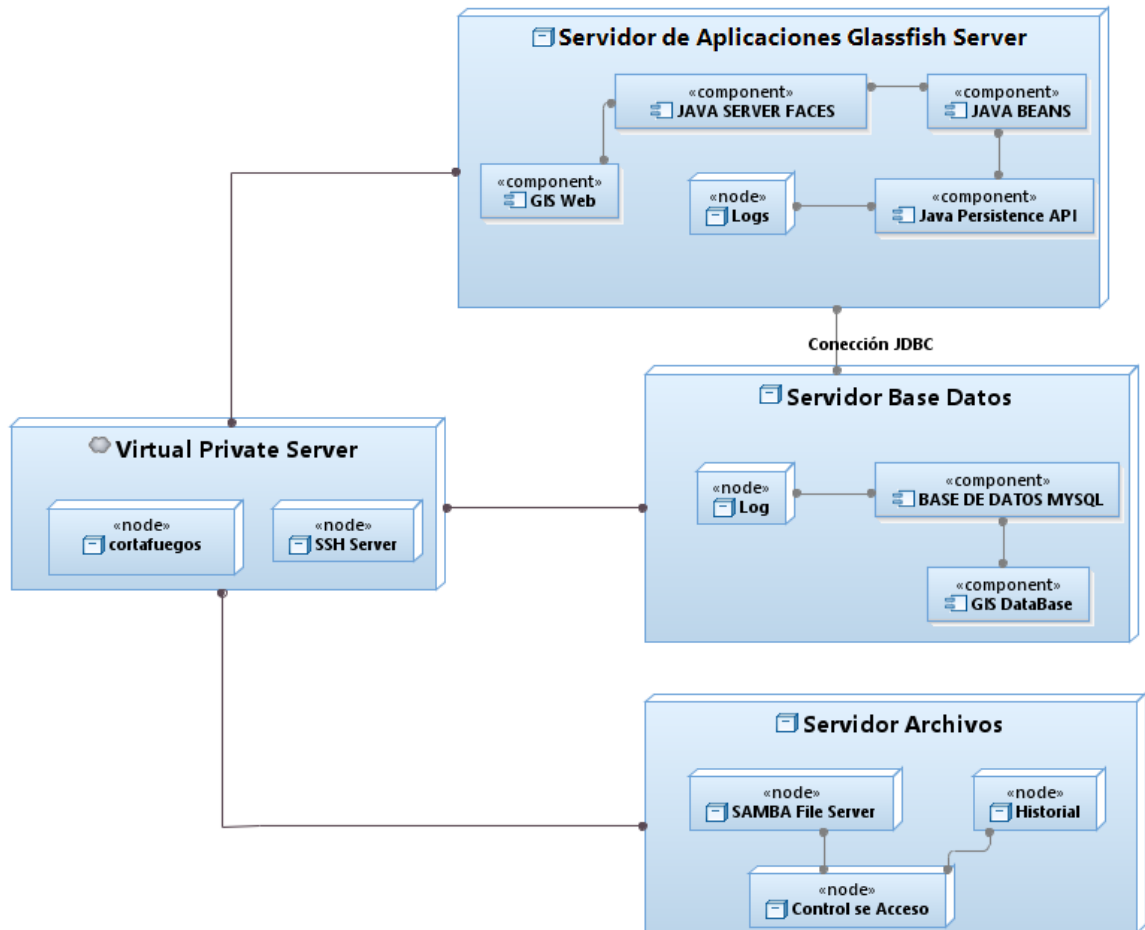


Ilustración No. 52: *Diagrama de despliegue del sistema.*


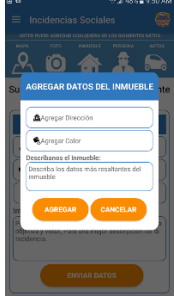


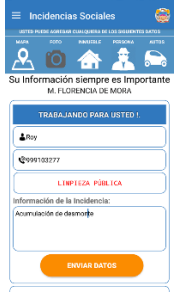
4.15. Casos de Prueba.

4.15.1. Pruebas de Caja Negra.

Es una técnica de pruebas de software la cual toma en cuenta las entradas que recibe y las respuestas o salidas que produce sin tomar en cuenta su funcionamiento interno.

Caso de Uso: Registrar incidencia (Móvil)

CASO	DATOS INGRESADOS	RESULTADO ESPERADO	RESULTADO OBTENIDO	ESTADO
Buscar tipo de incidencia	Presiona botón “SELECCIONE TIPO DE INCIDENCIA”	Muestra pantalla en azul los tipos de incidencia.		Ok
Selecciona tipo de incidencia	Elige cualquier tipo de incidencia.	Muestra en letras rojas la incidencia seleccionada.		Ok.
Ubicar la incidencia.	Elige opción MAPA.	Muestra mapa para indicar lugar de la incidencia.		Ok.
Grabar lugar de la incidencia.	Selecciona GUARDAR.	Guarda ubicación de la incidencia en el sistema web.		Ok.
Capturar incidencia con cámara.	Selecciona FOTO.	Muestra 03 opciones: - OPCIÓN MULTI-FOTO. - TOMAR FOTO. - SELECC. UNA FOTO		Ok.

<p>Grabar foto tomada o seleccionada.</p>	<p>Selecciona MULTI-FOTO, FOTO o SELECCIONA FOTO.</p>	<p>Se activa la cámara o permite elegir una foto de la galería. Y a la vez se registra en el sistema web.</p>		<p>Ok.</p>
<p>Describir inmueble.</p>	<p>Selecciona INMUEBLE.</p>	<p>Permite ingresar y grabar detalles del inmueble a reportar los cuales se verán reflejados en el sistema web.</p>		<p>Ok.</p>
<p>Describir persona.</p>	<p>Selecciona PERSONA.</p>	<p>Permite ingresar y grabar detalles de la persona a reportar los cuales se verán reflejados en el sistema web.</p>		<p>Ok.</p>
<p>Describir auto.</p>	<p>Selecciona AUTOS.</p>	<p>Permite ingresar y grabar detalles del auto(s) a reportar los cuales se verán reflejados en el sistema web.</p>		<p>Ok.</p>
<p>Registrar nombre y nro. de teléfono del que reporta la incidencia.</p>	<p>- No se ingresa datos. - Se ingresa datos</p>	<p>Como es opcional, se ingrese datos o no, se envía la incidencia al sistema web.</p>		<p>Ok</p>


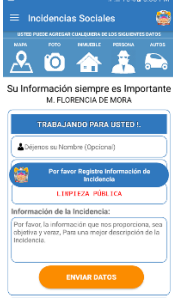
<p>Registrar información de la incidencia.</p>	<p>Se registra información de la incidencia.</p>	<p>Se envía correctamente la información al sistema web mostrando el sgte. mensaje: Logo de la muni + código de la incidencia + Gracias.</p>		<p>Ok</p>
<p>Registrar información de la incidencia.</p>	<p>No se registra información de la incidencia.</p>	<p>No se envía correctamente la información al sistema web mostrando el sgte. Mensaje: Logo de la muni + Por favor Registre Información de Incidencia.</p>		<p>Ok</p>

Tabla No. 18: Pruebas de caja negra.

4.15.2. Pruebas Unitarias.

Para este tipo de pruebas se ha utilizado Junit, una herramienta que trae integrada Netbeans, con la cual podemos comprobar el correcto funcionamiento de un método, función y las clases, para este caso vamos a comprobar las clases.

Clase Colaborador

```

package bc;

import be.GisColaborador;
import java.util.Date;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisColaboradorFacadeTest {

    public GisColaboradorFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisColaborador entity = new GisColaborador();
        entity.setApePat("admin");
        entity.setApeMat("admin");
        entity.setNombres("admin");

        entity.setDni("12345678");
        entity.setUsuario("admin02");
        entity.setClave("8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918");
        entity.setCambioPassword(1);
        entity.setEstadoExistencia(1);
        entity.setFecReg(new Date());

        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisColaboradorFacadeLocal instance = (GisColaboradorFacadeLocal) container.getContext().lookup("java:global/classes/GisColaboradorFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisColaborador entity = new GisColaborador(1);
        entity.setApePat("admin");
        entity.setApeMat("admin");
        entity.setNombres("admin");
        entity.setDni("12345678");
        entity.setUsuario("admin");
        entity.setClave("8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918");
        entity.setCambioPassword(1);
        entity.setEstadoExistencia(1);
        entity.setFecReg(new Date());

        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisColaboradorFacadeLocal instance = (GisColaboradorFacadeLocal) container.getContext().lookup("java:global/classes/GisColaboradorFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");

        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    }
}

```

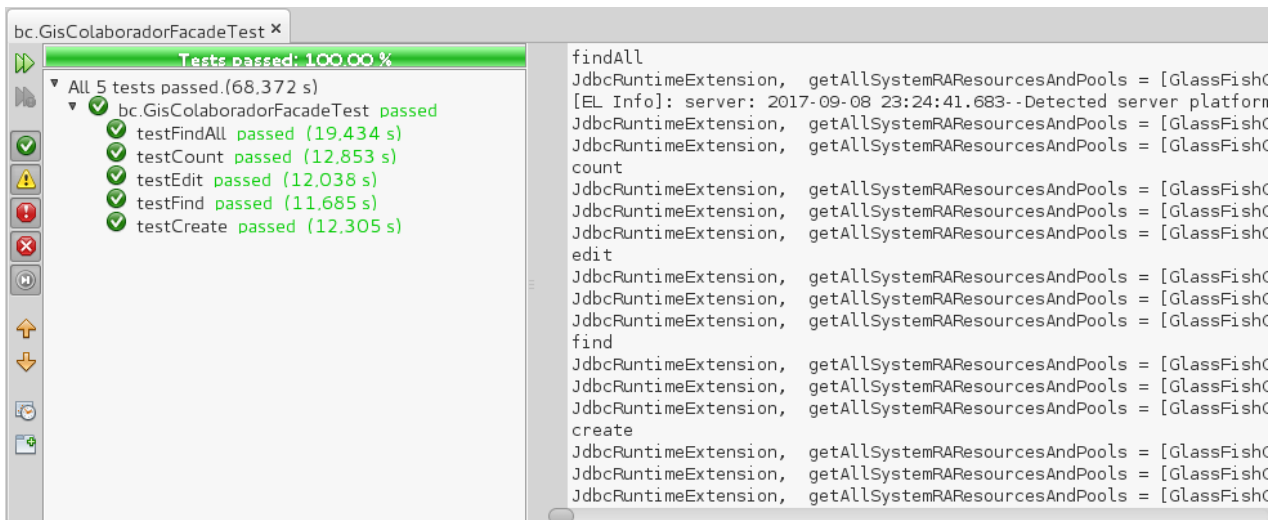
```

        GisColaboradorFacadeLocal instance = (GisColaboradorFacadeLocal) container.getContext().lookup("java:global/classes/GisColaboradorFacade");
        GisColaborador expResult = new GisColaborador(1);
        GisColaborador result = instance.find(id);
        assertEquals(expResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisColaboradorFacadeLocal instance = (GisColaboradorFacadeLocal) container.getContext().lookup("java:global/classes/GisColaboradorFacade");
        List<GisColaborador> expResult = new LinkedList<>();
        expResult.add(new GisColaborador(1));
        expResult.add(new GisColaborador(2));
        expResult.add(new GisColaborador(3));
        expResult.add(new GisColaborador(4));
        expResult.add(new GisColaborador(5));
        List<GisColaborador> result = instance.findAll();
        assertEquals(expResult, result);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisColaboradorFacadeLocal instance = (GisColaboradorFacadeLocal) container.getContext().lookup("java:global/classes/GisColaboradorFacade");
        int expResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expResult, result);
        container.close();
    }
}

```



bc.GisColaboradorFacadeTest x

Tests passed: 100.00 %

- All 5 tests passed. (68.372 s)
 - bc.GisColaboradorFacadeTest passed
 - testFindAll passed (19.434 s)
 - testCount passed (12.853 s)
 - testEdit passed (12.038 s)
 - testFind passed (11.685 s)
 - testCreate passed (12.305 s)

findAll
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
[EL Info]: server: 2017-09-08 23:24:41.683--Detected server platform
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
count
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
edit
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
find
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
create
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC

Ilustración No. 53: Prueba unitaria a la clase *GisColaborador*.

Clase Configuración

```

package bc;

import bc.GisConfiguracion;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisConfiguracionFacadeTest {

    public GisConfiguracionFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisConfiguracion entity = new GisConfiguracion();
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisConfiguracionFacadeLocal instance = (GisConfiguracionFacadeLocal) container.getContext().lookup("java:global/classes/GisConfiguracionFacade");
        instance.create(entity);
        container.close();
    }

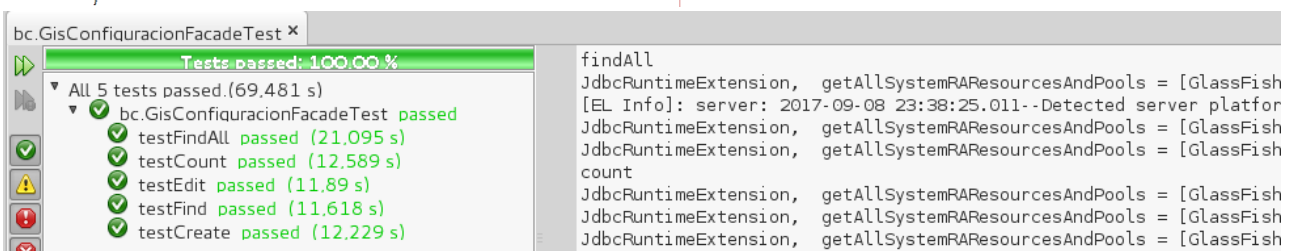
    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisConfiguracion entity = new GisConfiguracion(1);
        entity.setRutaArchivosImagenes("localhost/imagenes/");
        entity.setRutaArchivosImagenesWeb("localhost/imagenesWeb/");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisConfiguracionFacadeLocal instance = (GisConfiguracionFacadeLocal) container.getContext().lookup("java:global/classes/GisConfiguracionFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisConfiguracionFacadeLocal instance = (GisConfiguracionFacadeLocal) container.getContext().lookup("java:global/classes/GisConfiguracionFacade");
        GisConfiguracion expectedResult = new GisConfiguracion(1);
        GisConfiguracion result = instance.find(id);
        assertEquals(expectedResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisConfiguracionFacadeLocal instance = (GisConfiguracionFacadeLocal) container.getContext().lookup("java:global/classes/GisConfiguracionFacade");
        List<GisConfiguracion> expectedResult = new LinkedList<>();
        expectedResult.add(new GisConfiguracion(1));
        expectedResult.add(new GisConfiguracion(2));
        List<GisConfiguracion> result = instance.findAll();
        assertEquals(result, expectedResult);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisConfiguracionFacadeLocal instance = (GisConfiguracionFacadeLocal) container.getContext().lookup("java:global/classes/GisConfiguracionFacade");
        int expectedResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expectedResult, result);
        container.close();
    }
}

```



bc.GisConfiguracionFacadeTest x

Tests passed: 100.00 %

- ▼ All 5 tests passed. (69.481 s)
 - bc.GisConfiguracionFacadeTest passed
 - testFindAll passed (21.095 s)
 - testCount passed (12.589 s)
 - testEdit passed (11.89 s)
 - testFind passed (11.618 s)
 - testCreate passed (12.229 s)

findAll
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
[EL Info]: server: 2017-09-08 23:38:25.011--Detected server platfor
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
count
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish

Ilustración No. 54: Prueba unitaria a la clase GisConfiguración.

Clase DenunciasWeb

```

package bc;

import bc.GisDenunciasWeb;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisDenunciasWebFacadeTest {

    public GisDenunciasWebFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisDenunciasWeb entity = new GisDenunciasWeb();
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisDenunciasWebFacadeLocal instance = (GisDenunciasWebFacadeLocal)container.getContext().lookup("java:global/classes/GisDenunciasWebFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisDenunciasWeb entity = new GisDenunciasWeb(100);
        entity.setNombre("ARGOS");
        entity.setTelefono("22143245");
        entity.setEmail("argos@argos.com");

        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisDenunciasWebFacadeLocal instance = (GisDenunciasWebFacadeLocal)container.getContext().lookup("java:global/classes/GisDenunciasWebFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 100;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisDenunciasWebFacadeLocal instance = (GisDenunciasWebFacadeLocal)container.getContext().lookup("java:global/classes/GisDenunciasWebFacade");
        GisDenunciasWeb expectedResult = new GisDenunciasWeb(100);
        GisDenunciasWeb result = instance.find(id);
        assertEquals(expectedResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisDenunciasWebFacadeLocal instance = (GisDenunciasWebFacadeLocal)container.getContext().lookup("java:global/classes/GisDenunciasWebFacade");
        List<GisDenunciasWeb> expectedResult = new LinkedList<>();
        expectedResult.add(new GisDenunciasWeb(1));
        expectedResult.add(new GisDenunciasWeb(2));
        expectedResult.add(new GisDenunciasWeb(3));
        expectedResult.add(new GisDenunciasWeb(4));

        expectedResult.add(new GisDenunciasWeb(4));
        List<GisDenunciasWeb> result = instance.findAll();
        assertEquals(result, expectedResult);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisDenunciasWebFacadeLocal instance = (GisDenunciasWebFacadeLocal)container.getContext().lookup("java:global/classes/GisDenunciasWebFacade");
        int expectedResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expectedResult, result);
        container.close();
    }
}

```

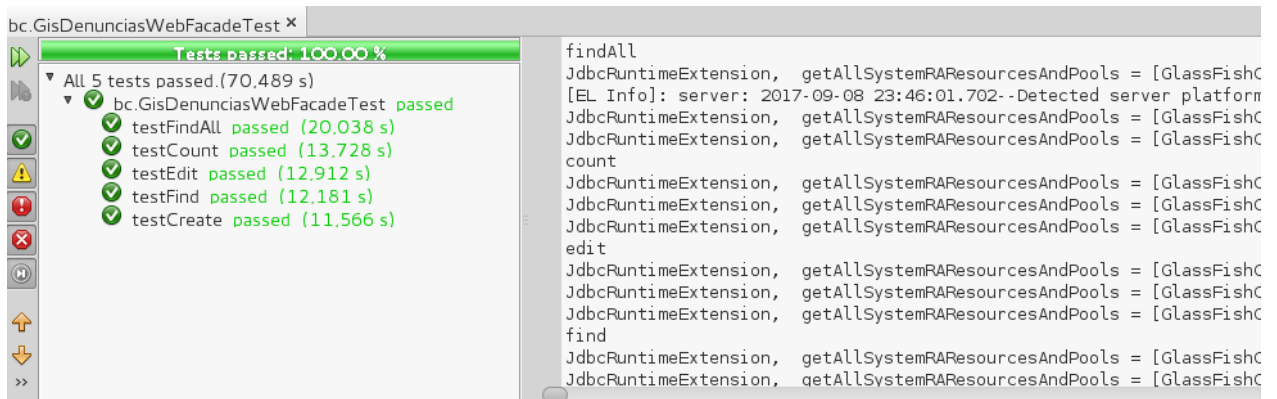



Ilustración No. 55: Prueba unitaria a la clase *GisDenuncias*.

Clase Grupo

```
package bc;

import bc.GisGrupo;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisGrupoFacadeTest {

    public GisGrupoFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisGrupo entity = new GisGrupo();
        entity.setNombreGrupo("test");
        entity.setDescripcionGrupo("desc");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoFacadeLocal instance = (GisGrupoFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoFacade");

        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisGrupo entity = new GisGrupo(1);
        entity.setNombreGrupo("administrador");
        entity.setDescripcionGrupo("");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoFacadeLocal instance = (GisGrupoFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoFacade");
        instance.edit(entity);
        container.close();
    }
}
```

```

@Test
public void testFind() throws Exception {
    System.out.println("find");
    Object id = 1;
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisGrupoFacadeLocal instance = (GisGrupoFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoFacade");
    GisGrupo expResult = new GisGrupo(1);
    GisGrupo result = instance.find(id);
    assertEquals(expResult, result);
    container.close();
}

@Test
public void testFindAll() throws Exception {
    System.out.println("findAll");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisGrupoFacadeLocal instance = (GisGrupoFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoFacade");
    List<GisGrupo> expResult = new LinkedList<>();
    expResult.add(new GisGrupo(1));
    expResult.add(new GisGrupo(2));
    expResult.add(new GisGrupo(3));
    List<GisGrupo> result = instance.findAll();

    assertEquals(result, expResult);
    container.close();
}

@Test
public void testCount() throws Exception {
    System.out.println("count");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisGrupoFacadeLocal instance = (GisGrupoFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoFacade");
    int expResult = instance.findAll().size();
    int result = instance.count();
    assertEquals(expResult, result);
    container.close();
}
}

```

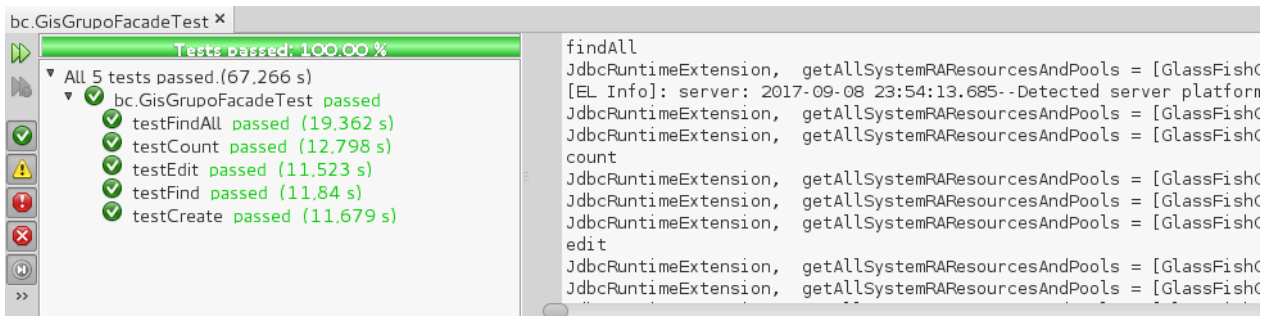


Ilustración No. 56: Prueba unitaria a la clase `GisGrupo`.

Clase GrupoUsuario

```

package bc;

import bc.GisColaborador;
import bc.GisGrupo;
import bc.GisGrupoUsuario;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisGrupoUsuarioFacadeTest {

    public GisGrupoUsuarioFacadeTest() {
    }
    @BeforeClass
    public static void setUpClass() {
    }
    @AfterClass
    public static void tearDownClass() {
    }
    @Before
    public void setUp() {
    }
    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisGrupoUsuario entity = new GisGrupoUsuario();
        entity.setGisColaborador(new GisColaborador(1));
        entity.setGisGrupo(new GisGrupo(1));

        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoUsuarioFacadeLocal instance = (GisGrupoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoUsuarioFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisGrupoUsuario entity = new GisGrupoUsuario(1);
        entity.setGisColaborador(new GisColaborador(1));
        entity.setGisGrupo(new GisGrupo(1));
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoUsuarioFacadeLocal instance = (GisGrupoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoUsuarioFacade");
        instance.edit(entity);
        container.close();
    }

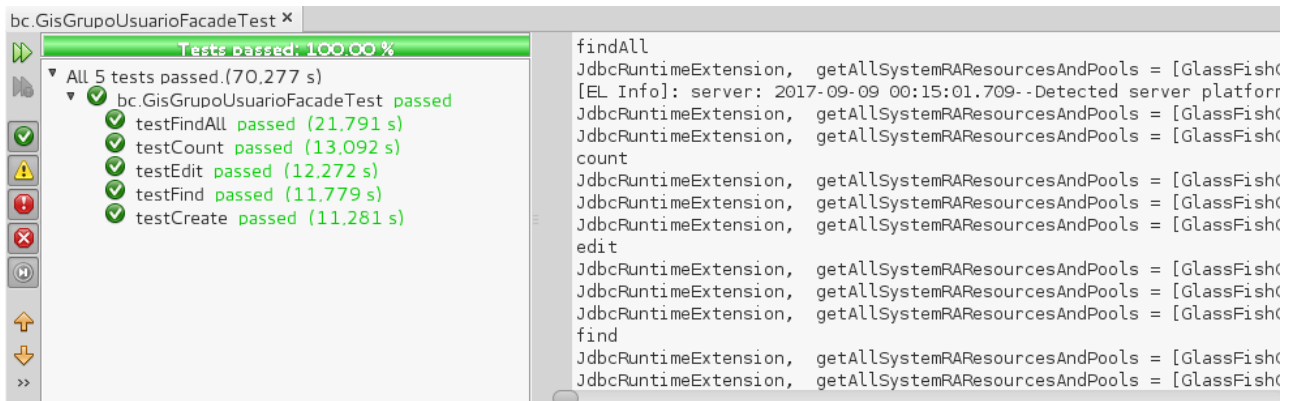
    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoUsuarioFacadeLocal instance = (GisGrupoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoUsuarioFacade");
        GisGrupoUsuario expResult = new GisGrupoUsuario(1);
        GisGrupoUsuario result = instance.find(id);
        assertEquals(expResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoUsuarioFacadeLocal instance = (GisGrupoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoUsuarioFacade");
        List<GisGrupoUsuario> expResult = new LinkedList<>();
        expResult.add(new GisGrupoUsuario(1));
        expResult.add(new GisGrupoUsuario(2));

        List<GisGrupoUsuario> result = instance.findAll();
        assertEquals(result, expResult);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisGrupoUsuarioFacadeLocal instance = (GisGrupoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisGrupoUsuarioFacade");
        int expResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expResult, result);
        container.close();
    }
}

```



```

bc.GisGrupoUsuarioFacadeTest X
Tests passed: 100.00 %
All 5 tests passed.(70.277 s)
  bc.GisGrupoUsuarioFacadeTest passed
    testFindAll passed (21.791 s)
    testCount passed (13.092 s)
    testEdit passed (12.272 s)
    testFind passed (11.779 s)
    testCreate passed (11.281 s)
  
```

Ilustración No. 57: Prueba unitaria a la clase *GisGrupoUsuario*.

Clase HechoPublico

```

package bc;

import bc.GisHechoPublico;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisHechoPublicoFacadeTest {

    public GisHechoPublicoFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisHechoPublico entity = new GisHechoPublico();
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisHechoPublicoFacadeLocal instance = (GisHechoPublicoFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoPublicoFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisHechoPublico entity = new GisHechoPublico(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisHechoPublicoFacadeLocal instance = (GisHechoPublicoFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoPublicoFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisHechoPublicoFacadeLocal instance = (GisHechoPublicoFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoPublicoFacade");
        GisHechoPublico expResult = new GisHechoPublico(1);
        GisHechoPublico result = instance.find(id);
        assertEquals(expResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisHechoPublicoFacadeLocal instance = (GisHechoPublicoFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoPublicoFacade");
        List<GisHechoPublico> expResult = new LinkedList<>();
        expResult.add(new GisHechoPublico(1));
        expResult.add(new GisHechoPublico(2));
        List<GisHechoPublico> result = instance.findAll();
        assertEquals(result, expResult);
        container.close();
    }
}
  
```

```

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisHechoPublicoFacadeLocal instance = (GisHechoPublicoFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoPublicoFacade");
        int expectedResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expectedResult, result);
        container.close();
    }
}

```

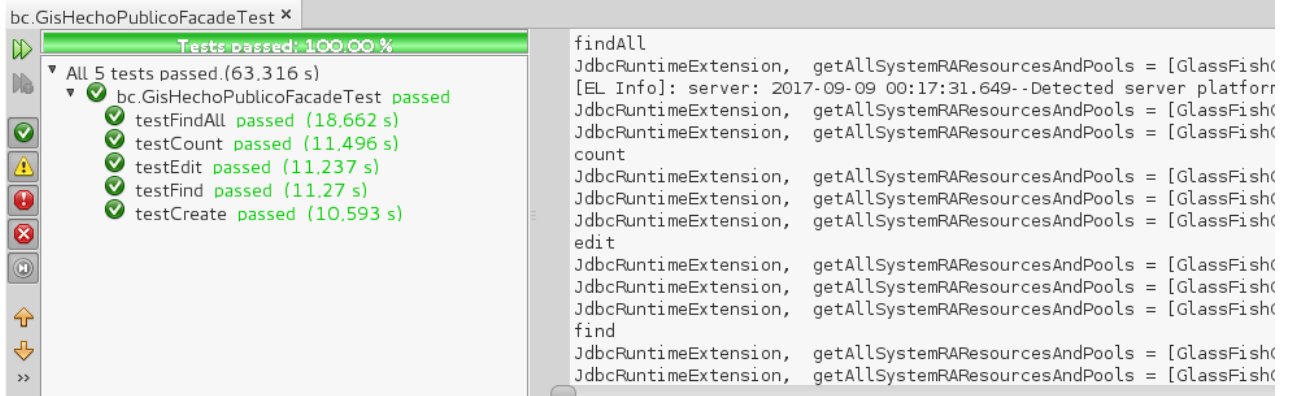


Ilustración No. 58: Prueba unitaria a la clase *GisHechoPublico*.

Clase HechoUsuario

```

package bc;

import bc.GisHechoPublico;
import bc.GisHechoUsuario;
import bc.GisUsuarioMail;
import java.util.Date;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisHechoUsuarioFacadeTest {

    public GisHechoUsuarioFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisHechoUsuario entity = new GisHechoUsuario();
        entity.setGisUsuarioMail(new GisUsuarioMail(1));

        entity.setGisHechoPublico(new GisHechoPublico(1));
        entity.setFecReg(new Date());
        entity.setEstadoExistencia(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisHechoUsuarioFacadeLocal instance = (GisHechoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoUsuarioFacade");
        instance.create(entity);
        container.close();
    }
}

```

```

@Test
public void testEdit() throws Exception {
    System.out.println("edit");
    GisHechoUsuario entity = new GisHechoUsuario(1);
    entity.setGisUsuarioMail(new GisUsuarioMail(1));
    entity.setGisHechoPublico(new GisHechoPublico(1));
    entity.setFechaReg(new Date());
    entity.setEstadoExistencia(1);

    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisHechoUsuarioFacadeLocal instance = (GisHechoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoUsuarioFacade");
    instance.edit(entity);
    container.close();
}

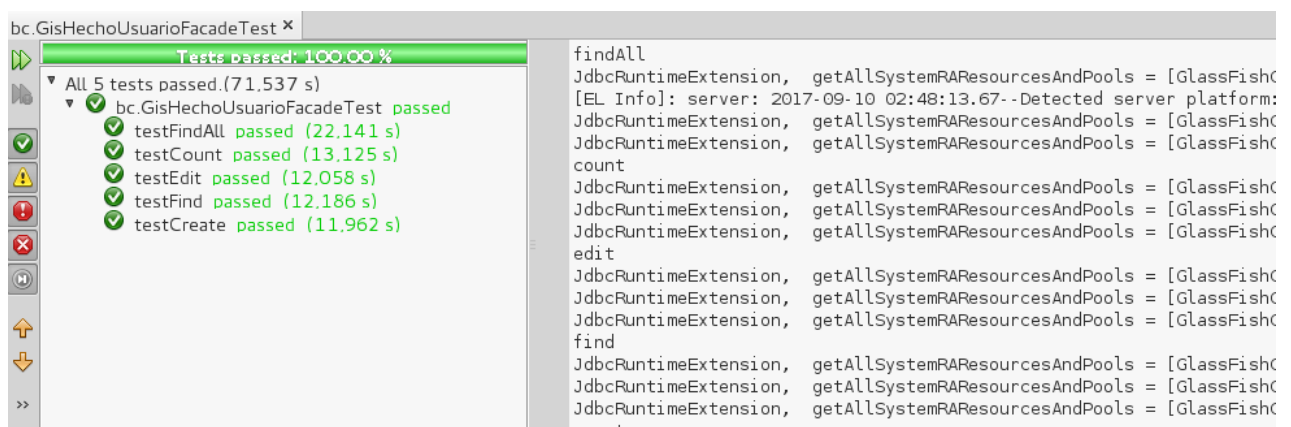
@Test
public void testFind() throws Exception {
    System.out.println("find");
    Object id = 1;
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisHechoUsuarioFacadeLocal instance = (GisHechoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoUsuarioFacade");
    GisHechoUsuario expectedResult = new GisHechoUsuario(1);
    GisHechoUsuario result = instance.find(id);
    assertEquals(expectedResult, result);
    container.close();
}

@Test
public void testFindAll() throws Exception {
    System.out.println("findAll");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisHechoUsuarioFacadeLocal instance = (GisHechoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoUsuarioFacade");
    List<GisHechoUsuario> expectedResult = new LinkedList<>();
    expectedResult.add(new GisHechoUsuario(1));
    expectedResult.add(new GisHechoUsuario(2));
    List<GisHechoUsuario> result = instance.findAll();
    assertEquals(result, expectedResult);
    container.close();
}

@Test
public void testFindAll() throws Exception {
    System.out.println("findAll");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisHechoUsuarioFacadeLocal instance = (GisHechoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoUsuarioFacade");
    List<GisHechoUsuario> expectedResult = new LinkedList<>();
    expectedResult.add(new GisHechoUsuario(1));
    expectedResult.add(new GisHechoUsuario(2));
    List<GisHechoUsuario> result = instance.findAll();
    assertEquals(result, expectedResult);
    container.close();
}

@Test
public void testCount() throws Exception {
    System.out.println("count");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisHechoUsuarioFacadeLocal instance = (GisHechoUsuarioFacadeLocal) container.getContext().lookup("java:global/classes/GisHechoUsuarioFacade");
    int expectedResult = instance.findAll().size();
    int result = instance.count();
    assertEquals(expectedResult, result);
    container.close();
}
}

```



bc.GisHechoUsuarioFacadeTest ×

Tests passed: 100.00 %

- All 5 tests passed. (71.537 s)
 - bc.GisHechoUsuarioFacadeTest passed
 - testFindAll passed (22.141 s)
 - testCount passed (13.125 s)
 - testEdit passed (12.058 s)
 - testFind passed (12.186 s)
 - testCreate passed (11.962 s)

findAll
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 [EL Info]: server: 2017-09-10 02:48:13.67- -Detected server platform:
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 count
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 getAllSystemRAResourcesAndPools = [GlassFishC
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 edit
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 getAllSystemRAResourcesAndPools = [GlassFishC
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 find
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 getAllSystemRAResourcesAndPools = [GlassFishC
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
 create

Ilustración No. 59: Prueba unitaria a la clase GisHechoUsuario.

Clase Inmueble

```

package bc;

] import be.GisHechoPublico;
import be.GisInmueble;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
- import static org.junit.Assert.*;

public class GisInmuebleFacadeTest {

]     public GisInmuebleFacadeTest() {
-     }
-     @BeforeClass
-     public static void setUpClass() {
-     }
-     @AfterClass
-     public static void tearDownClass() {
-     }
-     @Before
-     public void setUp() {
-     }
-     @After
-     public void tearDown() {
-     }

-     @Test
-     public void testCreate() throws Exception {
-         System.out.println("create");
-         GisInmueble entity = new GisInmueble();
-         entity.setEstadoExistencia(1);
-         entity.setGisHechoPublico(new GisHechoPublico(1));
-         EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();

-         GisInmuebleFacadeLocal instance = (GisInmuebleFacadeLocal) container.getContext().lookup("java:global/classes/GisInmuebleFacade");
-         instance.create(entity);
-         container.close();
-     }

-     @Test
-     public void testEdit() throws Exception {
-         System.out.println("edit");
-         GisInmueble entity = new GisInmueble(1);
-         entity.setEstadoExistencia(1);
-         entity.setGisHechoPublico(new GisHechoPublico(1));

-         EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
-         GisInmuebleFacadeLocal instance = (GisInmuebleFacadeLocal) container.getContext().lookup("java:global/classes/GisInmuebleFacade");
-         instance.edit(entity);
-         container.close();
-     }

-     @Test
-     public void testFind() throws Exception {
-         System.out.println("find");
-         Object id = 1;
-         EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
-         GisInmuebleFacadeLocal instance = (GisInmuebleFacadeLocal) container.getContext().lookup("java:global/classes/GisInmuebleFacade");
-         GisInmueble expResult = new GisInmueble(1);
-         GisInmueble result = instance.find(id);
-         assertEquals(expResult, result);
-         container.close();
-     }

-     @Test
-     public void testFindAll() throws Exception {
-         System.out.println("findAll");
-         EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
-         GisInmuebleFacadeLocal instance = (GisInmuebleFacadeLocal) container.getContext().lookup("java:global/classes/GisInmuebleFacade");
-         List<GisInmueble> expResult = new LinkedList<>();
-         expResult.add(new GisInmueble(1));
-         expResult.add(new GisInmueble(2));
    }
}

```

```

        expResult.add(new GisInmueble(3));
        expResult.add(new GisInmueble(4));
        List<GisInmueble> result = instance.findAll();
        assertEquals(result, result);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisInmuebleFacadeLocal instance = (GisInmuebleFacadeLocal) container.getContext().lookup("java:global/classes/GisInmuebleFacade");
        int expResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expResult, result);
        container.close();
    }
}

```

bc.GisInmuebleFacadeTest x

Tests passed: 100.00 %

- All 5 tests passed.(76.253 s)
 - bc.GisInmuebleFacadeTest passed
 - testFindAll passed (22.964 s)
 - testCount passed (13.937 s)
 - testEdit passed (13.551 s)
 - testFind passed (13.097 s)
 - testCreate passed (12.641 s)

findAll
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 [EL Info]: server: 2017-09-10 02:59:48.765--Detected server platform
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 count
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 edit
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 find
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
 create
 JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish

Ilustración No. 60: Prueba unitaria a la clase GisInmueble.

Clase Persona

```

package bc;

import be.GisHechoPublico;
import be.GisPersona;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisPersonaFacadeTest {

    public GisPersonaFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisPersona entity = new GisPersona();
        entity.setGisHechoPublico(new GisHechoPublico(2));
        entity.setNombres("Daniel");
        entity.setApellidos("gato");
    }
}

```



```

        entity.setApelativo("xav");
        entity.setEstadoExistencia(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPersonaFacadeLocal instance = (GisPersonaFacadeLocal) container.getContext().lookup("java:global/classes/GisPersonaFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisPersona entity = new GisPersona(1);
        entity.setGisHechoPublico(new GisHechoPublico(2));
        entity.setNombres("Daniel");
        entity.setApellidos("gato");
        entity.setApelativo("xav");
        entity.setEstadoExistencia(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPersonaFacadeLocal instance = (GisPersonaFacadeLocal) container.getContext().lookup("java:global/classes/GisPersonaFacade");
        instance.edit(entity);
        container.close();
    }

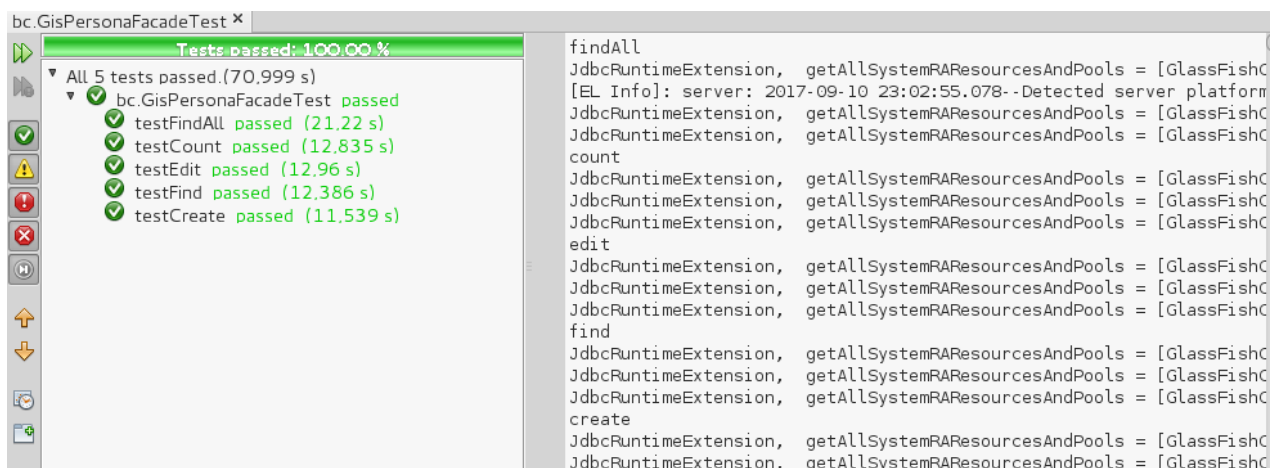
    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPersonaFacadeLocal instance = (GisPersonaFacadeLocal) container.getContext().lookup("java:global/classes/GisPersonaFacade");
        GisPersona expResult = new GisPersona(1);
        GisPersona result = instance.find(id);
        assertEquals(expResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");

        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPersonaFacadeLocal instance = (GisPersonaFacadeLocal) container.getContext().lookup("java:global/classes/GisPersonaFacade");
        List<GisPersona> expResult = new LinkedList<>();
        expResult.add(new GisPersona(1));
        expResult.add(new GisPersona(2));
        expResult.add(new GisPersona(3));
        expResult.add(new GisPersona(4));
        expResult.add(new GisPersona(5));
        expResult.add(new GisPersona(6));
        expResult.add(new GisPersona(7));
        List<GisPersona> result = instance.findAll();
        assertEquals(result, expResult);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPersonaFacadeLocal instance = (GisPersonaFacadeLocal) container.getContext().lookup("java:global/classes/GisPersonaFacade");
        int expResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expResult, result);
        container.close();
    }
}

```



bc.GisPersonaFacadeTest x

Tests passed: 100.00 %

▼ All 5 tests passed.(70.999 s)

- bc.GisPersonaFacadeTest passed
- testFindAll passed (21.22 s)
- testCount passed (12.835 s)
- testEdit passed (12.96 s)
- testFind passed (12.386 s)
- testCreate passed (11.539 s)

findAll
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
[EL Info]: server: 2017-09-10 23:02:55.078 - Detected server platform
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
count
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
edit
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
find
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
create
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFishC

Ilustración No. 61: Prueba unitaria a la clase GisPersona.

Clase Prevencion

```

package bc;

import bc.GisPrevencion;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisPrevencionFacadeTest {

    public GisPrevencionFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisPrevencion entity = new GisPrevencion();
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPrevencionFacadeLocal instance = (GisPrevencionFacadeLocal) container.getContext().lookup("java:global/classes/GisPrevencionFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisPrevencion entity = new GisPrevencion(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPrevencionFacadeLocal instance = (GisPrevencionFacadeLocal) container.getContext().lookup("java:global/classes/GisPrevencionFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPrevencionFacadeLocal instance = (GisPrevencionFacadeLocal) container.getContext().lookup("java:global/classes/GisPrevencionFacade");
        GisPrevencion expResult = new GisPrevencion(1);
        GisPrevencion result = instance.find(id);
        assertEquals(expResult, result);
        container.close();
    }

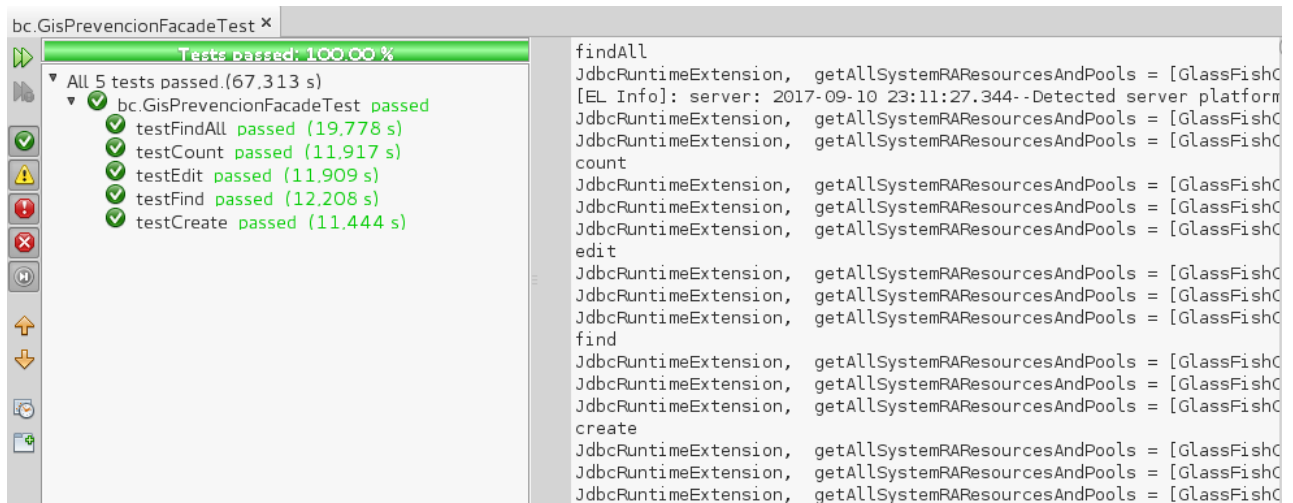
    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPrevencionFacadeLocal instance = (GisPrevencionFacadeLocal) container.getContext().lookup("java:global/classes/GisPrevencionFacade");
        List<GisPrevencion> expResult = new LinkedList<>();
        expResult.add(new GisPrevencion(1));
        expResult.add(new GisPrevencion(2));
        expResult.add(new GisPrevencion(3));
        expResult.add(new GisPrevencion(4));
        expResult.add(new GisPrevencion(5));

        List<GisPrevencion> result = instance.findAll();
        assertEquals(result, result);

        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisPrevencionFacadeLocal instance = (GisPrevencionFacadeLocal) container.getContext().lookup("java:global/classes/GisPrevencionFacade");
        int expResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expResult, result);
        container.close();
    }
}

```



```

bc.GisPrevencionFacadeTest x
Tests passed: 100.00 %
All 5 tests passed.(67.313 s)
  bc.GisPrevencionFacadeTest passed
    testFindAll passed (19.778 s)
    testCount passed (11.917 s)
    testEdit passed (11.909 s)
    testFind passed (12.208 s)
    testCreate passed (11.444 s)

findAll
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
[EL Info]: server: 2017-09-10 23:11:27.344--Detected server platform
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
count
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
edit
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
find
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
create
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC

```

Ilustración No. 62: Prueba unitaria a la clase *GisPrevencion*.

Clase ServicioWeb

```

package bc;

import bc.GisServicioWeb;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisServicioWebFacadeTest {

    public GisServicioWebFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisServicioWeb entity = new GisServicioWeb();
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisServicioWebFacadeLocal instance = (GisServicioWebFacadeLocal)container.getContext().lookup("java:global/classes/GisServicioWebFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisServicioWeb entity = new GisServicioWeb(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisServicioWebFacadeLocal instance = (GisServicioWebFacadeLocal)container.getContext().lookup("java:global/classes/GisServicioWebFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisServicioWebFacadeLocal instance = (GisServicioWebFacadeLocal)container.getContext().lookup("java:global/classes/GisServicioWebFacade");
        GisServicioWeb expResult = new GisServicioWeb(1);
        GisServicioWeb result = instance.find(id);
        assertEquals(expResult, result);
        container.close();
    }
}

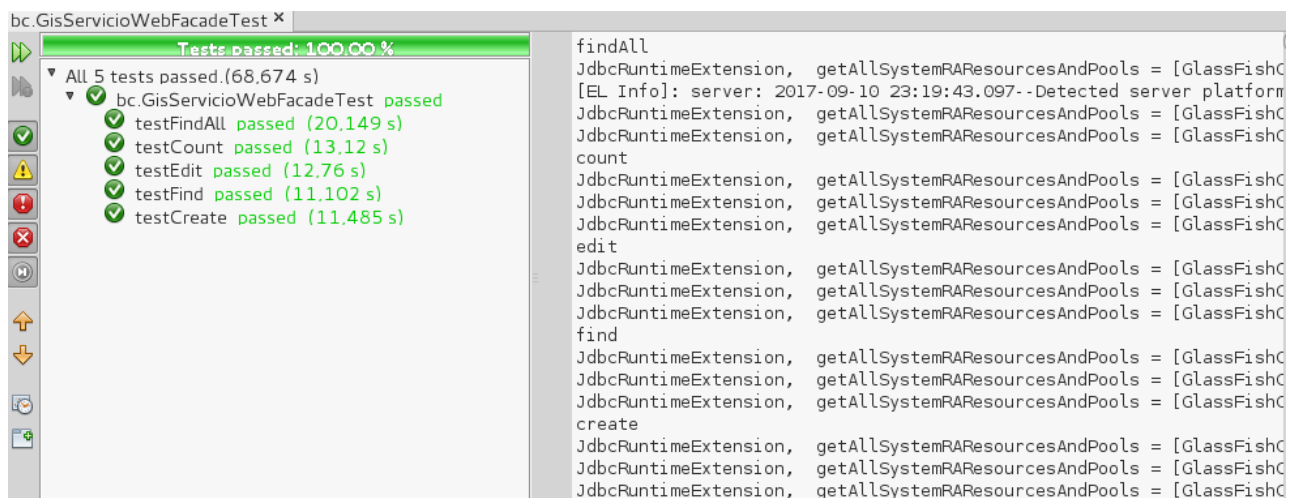
```

```

@Test
public void testFindAll() throws Exception {
    System.out.println("findAll");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisServicioWebFacadeLocal instance = (GisServicioWebFacadeLocal) container.getContext().lookup("java:global/classes/GisServicioWebFacade");
    List<GisServicioWeb> expectedResult = new LinkedList<>();
    expectedResult.add(new GisServicioWeb(1));
    expectedResult.add(new GisServicioWeb(2));
    expectedResult.add(new GisServicioWeb(3));
    List<GisServicioWeb> result = instance.findAll();
    assertEquals(result, expectedResult);
    container.close();
}

@Test
public void testCount() throws Exception {
    System.out.println("count");
    EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
    GisServicioWebFacadeLocal instance = (GisServicioWebFacadeLocal) container.getContext().lookup("java:global/classes/GisServicioWebFacade");
    int expectedResult = instance.findAll().size();
    int result = instance.count();
    assertEquals(result, expectedResult);
    container.close();
}
}

```



bc.GisServicioWebFacadeTest x

Tests passed: 100.00 %

All 5 tests passed. (68.674 s)

- bc.GisServicioWebFacadeTest passed
 - testFindAll passed (20.149 s)
 - testCount passed (13.12 s)
 - testEdit passed (12.76 s)
 - testFind passed (11.102 s)
 - testCreate passed (11.485 s)

findAll

JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
[EL Info]: server: 2017-09-10 23:19:43.097--Detected server platform
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
count
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
edit
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
find
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
create
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC
getAllSystemRAResourcesAndPools = [GlassFishC
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFishC

Ilustración No. 63: Prueba unitaria a la clase GisServicioWeb.

Clase UsuarioMail

```

package bc;

import bc.GisUsuarioMail;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisUsuarioMailFacadeTest {

    public GisUsuarioMailFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisUsuarioMail entity = new GisUsuarioMail();
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisUsuarioMailFacadeLocal instance = (GisUsuarioMailFacadeLocal) container.getContext().lookup("java:global/classes/GisUsuarioMailFacade");
        instance.create(entity);
        container.close();
    }
}

```

```

    }

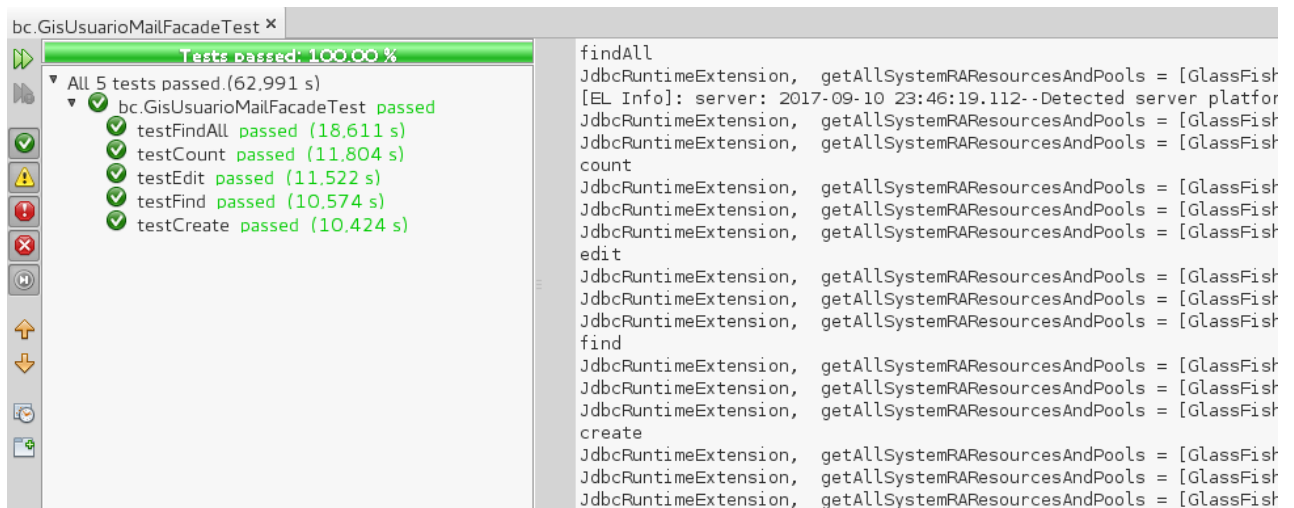
    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisUsuarioMail entity = new GisUsuarioMail(1);
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisUsuarioMailFacadeLocal instance = (GisUsuarioMailFacadeLocal) container.getContext().lookup("java:global/classes/GisUsuarioMailFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisUsuarioMailFacadeLocal instance = (GisUsuarioMailFacadeLocal) container.getContext().lookup("java:global/classes/GisUsuarioMailFacade");
        GisUsuarioMail expectedResult = new GisUsuarioMail(1);
        GisUsuarioMail result = instance.find(id);
        assertEquals(expectedResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisUsuarioMailFacadeLocal instance = (GisUsuarioMailFacadeLocal) container.getContext().lookup("java:global/classes/GisUsuarioMailFacade");
        List<GisUsuarioMail> expectedResult = new LinkedList<>();
        expectedResult.add(new GisUsuarioMail(1));
        expectedResult.add(new GisUsuarioMail(2));
        List<GisUsuarioMail> result = instance.findAll();
        assertEquals(result, expectedResult);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisUsuarioMailFacadeLocal instance = (GisUsuarioMailFacadeLocal) container.getContext().lookup("java:global/classes/GisUsuarioMailFacade");
        int expectedResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expectedResult, result);
        container.close();
    }
}

```



bc.GisUsuarioMailFacadeTest x

Tests passed: 100.00 %

- ▼ All 5 tests passed. (62.991 s)
 - bc.GisUsuarioMailFacadeTest passed
 - testFindAll passed (18.611 s)
 - testCount passed (11.804 s)
 - testEdit passed (11.522 s)
 - testFind passed (10.574 s)
 - testCreate passed (10.424 s)

findAll

JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
[EL Info]: server: 2017-09-10 23:46:19.112-- Detected server platfor
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
count
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
edit
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
find
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
create
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRResourcesAndPools = [GlassFish

Ilustración No. 64: Prueba unitaria a la clase GisUsuarioMail.

Clase Vehiculo

```

package bc;

import be.GisHechoPublico;
import be.GisVehiculo;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisVehiculoFacadeTest {

    public GisVehiculoFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisVehiculo entity = new GisVehiculo();
        entity.setGisHechoPublico(new GisHechoPublico(1));
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVehiculoFacadeLocal instance = (GisVehiculoFacadeLocal) container.getContext().lookup("java:global/classes/GisVehiculoFacade");
        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisVehiculo entity = new GisVehiculo(1);
        entity.setGisHechoPublico(new GisHechoPublico(1));
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVehiculoFacadeLocal instance = (GisVehiculoFacadeLocal) container.getContext().lookup("java:global/classes/GisVehiculoFacade");
        instance.edit(entity);
        container.close();
    }

    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVehiculoFacadeLocal instance = (GisVehiculoFacadeLocal) container.getContext().lookup("java:global/classes/GisVehiculoFacade");
        GisVehiculo expectedResult = new GisVehiculo(1);
        GisVehiculo result = instance.find(id);
        assertEquals(expectedResult, result);

        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVehiculoFacadeLocal instance = (GisVehiculoFacadeLocal) container.getContext().lookup("java:global/classes/GisVehiculoFacade");
        List<GisVehiculo> expectedResult = new LinkedList<>();
        expectedResult.add(new GisVehiculo(1));
        expectedResult.add(new GisVehiculo(2));
        expectedResult.add(new GisVehiculo(3));
        expectedResult.add(new GisVehiculo(4));
        List<GisVehiculo> result = instance.findAll();
        assertEquals(result, expectedResult);
        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVehiculoFacadeLocal instance = (GisVehiculoFacadeLocal) container.getContext().lookup("java:global/classes/GisVehiculoFacade");
        int expectedResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expectedResult, result);
        container.close();
    }
}

```

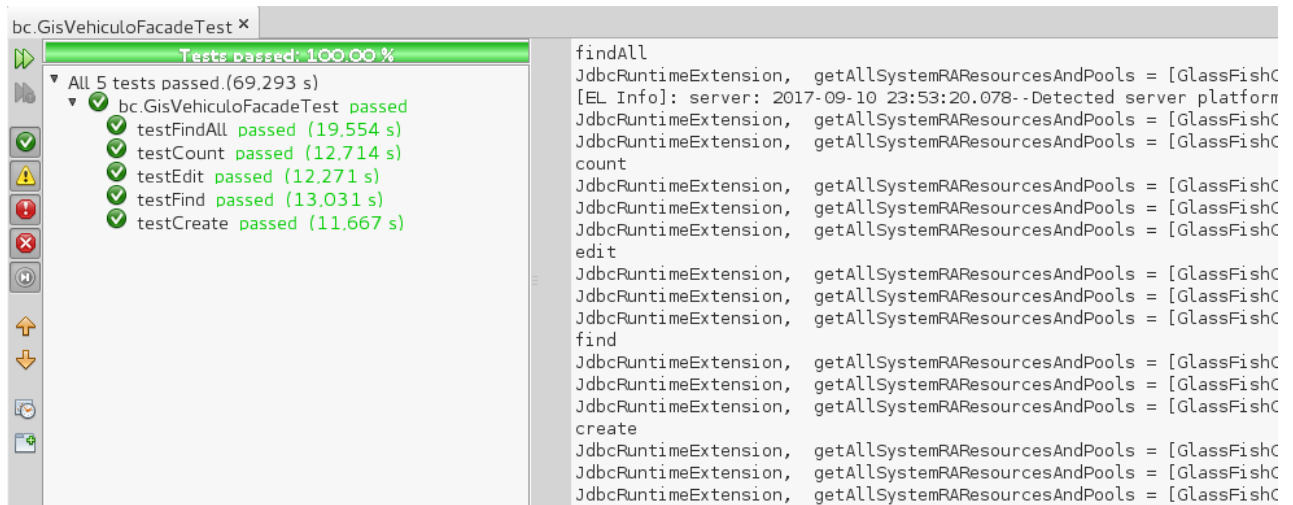


Ilustración No. 65: Prueba unitaria a la clase GisVehiculo.

Clase VerificacionHecho

```

package bc;

import bc.GisHechoPublico;
import bc.GisVerificacionHecho;
import java.util.LinkedList;
import java.util.List;
import javax.ejb.embeddable.EJBContainer;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class GisVerificacionHechoFacadeTest {

    public GisVerificacionHechoFacadeTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    @Test
    public void testCreate() throws Exception {
        System.out.println("create");
        GisVerificacionHecho entity = new GisVerificacionHecho();
        entity.setGisHechoPublico(new GisHechoPublico(1));
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVerificacionHechoFacadeLocal instance = (GisVerificacionHechoFacadeLocal) container.getContext().lookup("java:global/classes/GisVerificacionHechoFacade");
    }
}

```

```

        instance.create(entity);
        container.close();
    }

    @Test
    public void testEdit() throws Exception {
        System.out.println("edit");
        GisVerificacionHecho entity = new GisVerificacionHecho(1);
        entity.setGisHechoPublico(new GisHechoPublico(1));
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVerificacionHechoFacadeLocal instance = (GisVerificacionHechoFacadeLocal) container.getContext().lookup("java:global/classes/GisVerificacionHechoFacade");
        instance.edit(entity);
        container.close();
    }

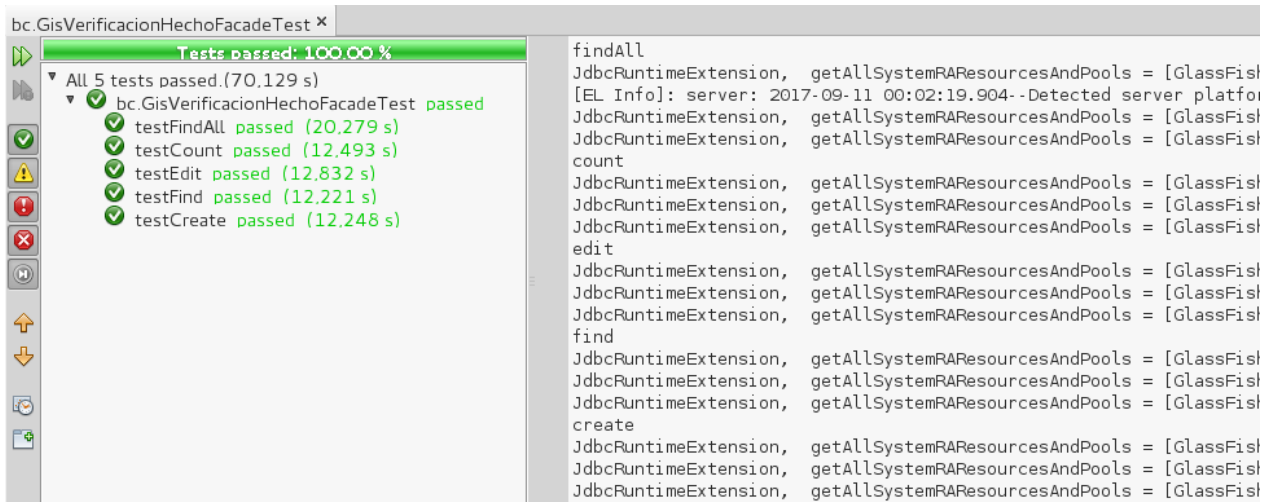
    @Test
    public void testFind() throws Exception {
        System.out.println("find");
        Object id = 1;
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVerificacionHechoFacadeLocal instance = (GisVerificacionHechoFacadeLocal) container.getContext().lookup("java:global/classes/GisVerificacionHechoFacade");
        GisVerificacionHecho expectedResult = new GisVerificacionHecho(1);
        GisVerificacionHecho result = instance.find(id);
        assertEquals(expectedResult, result);
        container.close();
    }

    @Test
    public void testFindAll() throws Exception {
        System.out.println("findAll");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVerificacionHechoFacadeLocal instance = (GisVerificacionHechoFacadeLocal) container.getContext().lookup("java:global/classes/GisVerificacionHechoFacade");
        List<GisVerificacionHecho> expectedResult = new LinkedList<>();
        expectedResult.add(new GisVerificacionHecho(1));
        expectedResult.add(new GisVerificacionHecho(2));
        expectedResult.add(new GisVerificacionHecho(3));
        List<GisVerificacionHecho> result = instance.findAll();
        assertEquals(result, expectedResult);

        container.close();
    }

    @Test
    public void testCount() throws Exception {
        System.out.println("count");
        EJBContainer container = javax.ejb.embeddable.EJBContainer.createEJBContainer();
        GisVerificacionHechoFacadeLocal instance = (GisVerificacionHechoFacadeLocal) container.getContext().lookup("java:global/classes/GisVerificacionHechoFacade");
        int expectedResult = instance.findAll().size();
        int result = instance.count();
        assertEquals(expectedResult, result);
        container.close();
    }
}

```



bc.GisVerificacionHechoFacadeTest ×

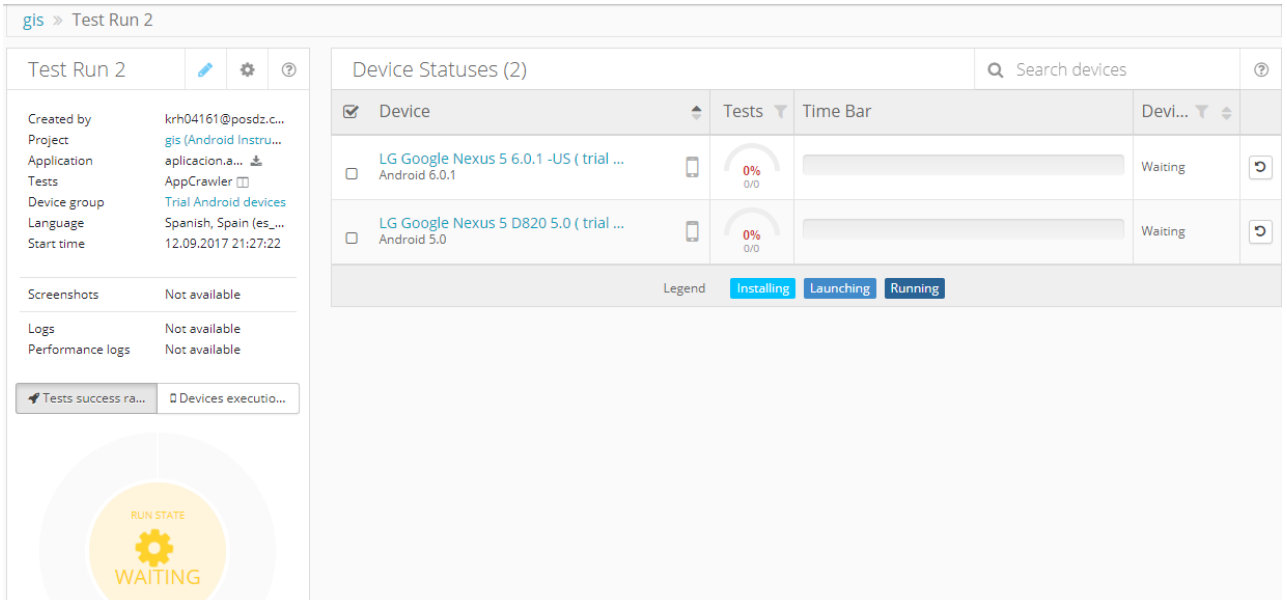
Tests passed: 100.00 %

- ▼ All 5 tests passed. (70.129 s)
 - bc.GisVerificacionHechoFacadeTest passed
 - testFindAll passed (20.279 s)
 - testCount passed (12.493 s)
 - testEdit passed (12.832 s)
 - testFind passed (12.221 s)
 - testCreate passed (12.248 s)

findAll
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
[EL Info]: server: 2017-09-11 00:02:19.904 - Detected server platfor
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
count
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
edit
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
find
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
create
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish
JdbcRuntimeExtension, getAllSystemRAResourcesAndPools = [GlassFish

Ilustración No. 66: Prueba unitaria a la clase GisVerificacionHecho.

Prueba de rendimiento al aplicativo móvil usando Testdroid, una aplicación ágil de prueba de aplicaciones.



gjs » Test Run 2

Test Run 2

Created by: krh04161@posdz.c...
Project: gjs (Android Instru...
Application: aplicacion.a...
Tests: AppCrawler
Device group: Trial Android devices
Language: Spanish, Spain (es_...
Start time: 12.09.2017 21:27:22

Screenshots: Not available
Logs: Not available
Performance logs: Not available

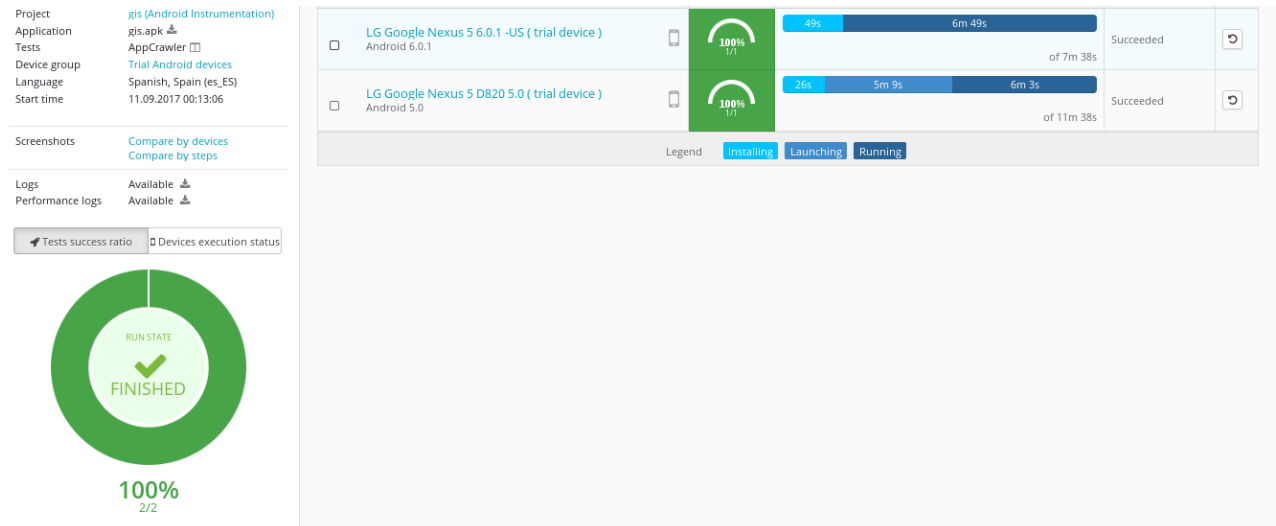
Tests success ra... | Devices executio...

RUN STATE
WAITING

Device	Tests	Time Bar	Devi...
<input type="checkbox"/> LG Google Nexus 5 6.0.1 -US (trial device) Android 6.0.1	0% 0/0		Waiting
<input type="checkbox"/> LG Google Nexus 5 D820 5.0 (trial device) Android 5.0	0% 0/0		Waiting

Legend: Installing | Launching | Running

Ilustración No. 67: Prueba en proceso del aplicativo móvil.



Project: gjs (Android Instrumentation)
Application: gjs.apk
Tests: AppCrawler
Device group: Trial Android devices
Language: Spanish, Spain (es_ES)
Start time: 11.09.2017 00:13:06

Screenshots: Compare by devices
Compare by steps

Logs: Available
Performance logs: Available

Tests success ratio | Devices execution status

RUN STATE
FINISHED
100%
2/2

Device	Tests	Time Bar	Devi...
<input type="checkbox"/> LG Google Nexus 5 6.0.1 -US (trial device) Android 6.0.1	100% 1/1	49s 6m 49s of 7m 38s	Succeeded
<input type="checkbox"/> LG Google Nexus 5 D820 5.0 (trial device) Android 5.0	100% 1/1	26s 5m 9s 6m 3s of 11m 38s	Succeeded

Legend: Installing | Launching | Running

Ilustración No. 68: Prueba terminada del aplicativo móvil.

4.16. Gestión de Riesgos

1. Introducción

1.1. Propósito

El presente documento presenta el análisis de los riesgos identificados durante la fase de Inicio del proyecto. Para cada riesgo observado se valorarán sus efectos y el contexto de aparición, para el caso en que se convierta en un hecho. Además, definiremos estrategias para mitigar el impacto de los posibles efectos de los riesgos.

1.2. Ámbito

El ámbito del análisis de riesgos cubre toda la extensión del proyecto observado desde su fase inicial. Será necesario durante el desarrollo del proyecto revisar y actualizar los contenidos del análisis de riesgos en caso de que se detecten nuevos riesgos no visibles en la Fase inicial.

2. Identificación de Riesgos

2.1. Clasificación de los Riesgos

Se clasifican en:

- Personal - Equipo de Trabajo.- Cuando miembros del equipo de trabajo impiden el Desarrollo adecuado del Proyecto, ya sea por el ingreso de alguno o el funcionamiento no adecuado de alguno entre otros.
- Organización y gestión.- Cuando existen problemas en la gestión y administración del proyecto.
- Requerimientos.- Cuando algún requerimiento impide el desarrollo del proyecto, ya sea por falta de especificación, mal entendimiento o alguna otra problemática que impida el desarrollo.
- Desarrollo e implementación.- Cuando se presentan problemas en el diseño, modelado y desarrollo del proyecto como sistema.
- Planificación y Tiempo.- Cuando existen problemas o errores en la planificación o cuando alguna actividad afectara el tiempo planeado para el proyecto, Aumentando el tiempo para finalizar este.
- Externo.- Cualquier actividad, suceso o elemento externo que amenace el cumplimiento del proyecto.

2.2. Lista de Riesgos

A continuación, se describe los posibles riesgos de nuestro proyecto detallando su descripción y las causas.

Identificación	Riesgo Identificado	Descripción del Riesgo	Causas del Riesgo
R1	No cumplir a tiempo con las actividades programadas.	Falta de tiempo y cruce de horarios por parte del equipo de trabajo, para generar los entregables.	Los integrantes del equipo no cumplen por motivos de permisos o casusas extra.
R2	Pasarnos del tiempo indicado para la elaboración del producto.	Complejidad de las actividades en el plan de trabajo que ocasionan retrasos y actividades extra temporales.	Deficiente coordinación en el desarrollo establecido de las actividades.
R3	La calidad del producto no es la esperada.	Falta de tiempo en la comprobación, diseño e implementación superior al esperado.	Deficiente abstracción de requerimientos.
R4	Poco conocimiento de implementación GPS.	El conocimiento adquirido por el equipo no es lo suficiente para concluir con este tema.	Falta de cursos de android.
R5	El desarrollo de funciones erróneas o una interfaz inadecuada.	Falta de tiempo en la comprobación, diseño e implementación superior al esperado.	No realizar pruebas unitarias y funcionales al software.

R6	Carecer de un plan de contingencia ante cualquier ataque de hackers.	Inexperiencia por parte del grupo de proyecto en temas de seguridad informática.	No definir este tipo de riesgo anticipadamente.
R7	Desistimiento de algún miembro del equipo de trabajo.	Retiro del proyecto de un miembro del equipo por problemas de fuerza mayor.	Enfermedad, problemas personales, problemas familiares, etc.

Tabla No. 19: *Lista de riesgos del proyecto. (Elaboración equipo de trabajo).*

3. Planificación de Respuesta a los Riesgos

A continuación, se describe las consideraciones a tomar para evitar o minimizar el impacto de los riesgos del proyecto:

Riesgo (Identificador)	Consideraciones a Tomar
R1	<ul style="list-style-type: none"> - Coordinar actividades de manera que todos los integrantes del equipo puedan cumplir con sus responsabilidades - Mayor comunicación entre los integrantes del equipo.
R2	<ul style="list-style-type: none"> - Establecer reuniones entre los integrantes del equipo. - Una mejor distribución de tiempos para la elaboración de los entregables que se solicitan según las actividades.
R3	<ul style="list-style-type: none"> - Tener mayor énfasis en las pruebas para otorgar un producto con la calidad necesaria. - La corroboración de requerimientos, interfaces y revisar prototipos con el cliente debe ser constante.
R4	<ul style="list-style-type: none"> - Asesoramiento externo sobre el tema e ir leyendo manuales para el buen desarrollo de la implementación del GPS.
R5	<ul style="list-style-type: none"> - Comunicación frecuente con el cliente sobre la aceptación de los requerimientos en las distintas fases de desarrollo del proyecto - Mayor análisis de los requerimientos durante el desarrollo del producto.
R6	<ul style="list-style-type: none"> - Elaboración de un plan de proyecto, junto con identificación de riesgos y consideraciones a tomar para evitar que se presenten.
R7	<ul style="list-style-type: none"> - Tener una persona alterna o sentirse uno mismo con la capacidad suficiente para seguir para seguir sólo con el apoyo.

Tabla No. 20: *Respuestas a los riesgos del proyecto (Elaboración equipo de trabajo)*

4.17. Material de Apoyo al Usuario Final

Ver Anexo 04.

CAPÍTULO 5. MATERIALES Y MÉTODOS

5.1. Tipo de diseño de investigación.

El tipo de diseño a aplicarse en el presente trabajo será de diseño experimental pura, tomando en cuenta que se manipulará de manera discreta la variable independiente (sistema informático) sobre la variable dependiente (gestión de incidencias sociales) a la cantidad determinada de incidencias ocurridas en un determinado mes, analizando la información durante la pre y post prueba para verificar la hipótesis planteada.

5.2. Material de estudio.

5.2.1. Unidad de estudio.

Vecino del distrito de Florencia de Mora.

5.2.2. Población.

La población a tomar en cuenta para el presente proyecto serán las incidencias sociales registradas en el mes de junio del 2017. El total de incidencias son 90 y fueron tomadas de las oficinas de seguridad ciudadana y de la oficina de limpieza pública que es a dónde llega nuestro alcance.

5.2.3. Muestra.

Como la población es menor a 100, la muestra se tomará sobre toda la población. $N = n = 90$. Donde:

N = población.

n = muestra

Para la determinación de la muestra se empleará la técnica del muestreo aleatorio simple, dada por la siguiente ecuación:

$$n = \frac{NZ^2 pq}{(N-1)e^2 + Z^2 pq}$$

Donde:

- n: Tamaño de la muestra
- N: Tamaño de la población
- Z: Valor de Z, 1.96 para $\alpha = 0.05$
- p: Proporción de éxito. En caso de desconocerse, se aplica la opción más desfavorable ($p=0.5$)
- q: $(1 - p)$
- e: Error de muestreo ($e=0.05$).

$$n = 73.07 = 73$$

La muestra a estudiar son 73 incidencias sociales observadas en el mes de junio del 2017.

5.3. Técnicas, procedimientos e instrumentos.

5.3.1. Para recolectar datos.

INDICADOR	TÉCNICA	INSTRUMENTO	PROCEDIMIENTO
<u>Tiempo promedio requerido para registrar la incidencia</u> Tiempo total actual de registro de incidencia	Observación	Aplicación móvil.	Seleccionar un conjunto de 73 incidencias reportados por el usuario, con el fin de monitorear el tiempo que requiere para registrar una incidencia a través del dispositivo móvil. Dicha aplicación registrará el tiempo desde el inicio y término del registro de la incidencia, durante un mes.
<u>Tiempo promedio requerido para derivar la incidencia.</u> Tiempo total actual de derivación de incidencia	Observación	Aplicación web.	Seleccionar un conjunto de 73 incidencias registradas en el aplicativo web, con el fin de monitorear el tiempo que requiere para derivar una incidencia a través del sistema web. Dicha aplicación registrará el tiempo desde el inicio y término de derivación de la incidencia, durante un mes.

$\frac{\text{Tiempo promedio de procesamiento de información.}}{\text{Tiempo total actual de procesamiento de información}}$	Observación	Aplicación web.	Seleccionar un conjunto de 73 incidencias registradas en el sistema web, con el fin de monitorear de las 73 incidencias registradas, cuántas han sido procesadas (leídas, descartadas, derivadas y atendidas) durante un mes.
--	-------------	-----------------	---

Tabla No. 21: *Técnicas, procedimientos e instrumentos de recolección de información.*
(Elaboración propia)

5.3.2. Para procesar datos.

INDICADOR	TÉCNICA	INSTRUMENTO	PROCEDIMIENTO
$\frac{\text{Tiempo promedio requerido para registrar la incidencia}}{\text{Tiempo total actual de registro de incidencia}}$	Operaciones de agregación	Hoja de cálculo Excel	Se realizará una consulta a la base de datos para obtener el tiempo promedio requerido de registro entre todas las incidencias registradas durante un rango de tiempo.
$\frac{\text{Tiempo promedio requerido para derivar la incidencia.}}{\text{Tiempo total actual de derivación de incidencia}}$	Operaciones de agregación	Hoja de cálculo Excel	Se realizará una consulta a la base de datos para obtener el tiempo promedio requerido de derivación entre todas las incidencias derivadas durante un rango de tiempo.
$\frac{\text{Tiempo promedio de procesamiento de información.}}{\text{Tiempo total actual de procesamiento de información}}$	Operaciones de agregación	Hoja de cálculo Excel	Se realizará una consulta a la base de datos para obtener el promedio de incidencias procesadas (atendidas) durante un rango de tiempo.

Tabla No. 22: *Técnicas, procedimientos e instrumentos de procesamiento de información.*
(Elaboración propia)

CAPÍTULO 6. RESULTADOS

6.1. Medición de indicadores.

A continuación, se presentan las mediciones realizadas por cada indicador según las técnicas y los instrumentos definidos.

1. Tiempo promedio requerido para registrar una incidencia / Tiempo total actual de registro de una incidencia.

Prueba z para la diferencia de promedios en el tiempo durante el registro de una incidencia.

Hipótesis estadísticas:

$$H_0 : \bar{X} \geq \mu$$

$$H_a : \bar{X} < \mu$$

Donde:

H_0 : Hipótesis nula.

H_a : Hipótesis a probar.

X : Promedio de tiempo que el usuario se tarda en registrar una incidencia desde su dispositivo móvil.

μ : Promedio de tiempo que el usuario se tarda en registrar una incidencia de manera manual como lo hace actualmente.

Nivel de significancia:

$\alpha = 0,05$: Nivel de seguridad del 95%.

Distribución de la prueba:

Se aplica la distribución normal estandarizada.

Función para la prueba:

$$Z = \frac{\bar{X}_D - \bar{X}_A}{\sqrt{\frac{\sigma_A^2}{n_A} + \frac{\sigma_D^2}{n_D}}}$$

Donde:

\bar{X}_D : Promedio de tiempo que los usuarios se tardan en registrar una incidencia utilizando la aplicación móvil = 1.50 min

\bar{X}_A : Promedio de tiempo que los usuarios se tardan en registrar una incidencia de forma manual como actualmente lo hacen = 20.00 min

σ_A : Desviación Estándar antes = 1.70

σ_D : Desviación Estándar antes utilizando la aplicación móvil = 1.49

n_A : Número de encuestados antes = 73

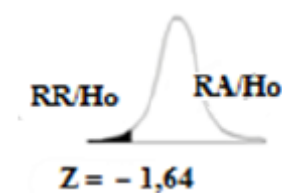
n_D : Número de usuarios que usaron la aplicación móvil = 73

Según la información recolectada, en un Pre test se obtiene un tiempo promedio de 20.00 minutos en el que los encuestados tardan en registrar una incidencia de forma manual. Después de usada la aplicación móvil, en un Post test que la base de información arroja, se obtiene un promedio de 1.50 minutos para el mismo propósito utilizando la aplicación móvil. **(Ver anexo N° 01).**

Valor experimental:

$$Z = \frac{1.50 - 20}{\sqrt{\frac{1.70^2}{73} + \frac{1.49^2}{73}}} = \frac{-18.50}{0.26} = -71.15$$

Región de Aceptación y Rechazo de la Hipótesis Nula:



Como $Z_0 = -71.15$ pertenece a la región de rechazo de la hipótesis nula, entonces se acepta la hipótesis alternativa.

2. Tiempo promedio requerido para derivar una incidencia / Tiempo total actual de derivación de una incidencia.

Prueba z para la diferencia de promedios en el tiempo durante el registro de una incidencia.

Hipótesis estadísticas:

$$H_0 : \bar{X} \geq \mu$$

$$H_a : \bar{X} < \mu$$

Donde:

H_0 : Hipótesis nula.

H_a : Hipótesis a probar.

X : Promedio de tiempo que el administrador web se tarda en derivar una incidencia desde la aplicación web.

μ : Promedio de tiempo que el Jefe de Área se tarda en derivar una incidencia de manera manual como lo hace actualmente.

Nivel de significancia:

$\alpha=0,05$: Nivel de seguridad del 95%.

Distribución de la prueba:

Se aplica la distribución normal estandarizada.

Función para la prueba:

$$Z = \frac{\bar{X}_D - \bar{X}_A}{\sqrt{\frac{\sigma_A^2}{n_A} + \frac{\sigma_D^2}{n_D}}}$$

Donde:

\bar{X}_D : Promedio de tiempo que el administrador web se tarda en derivar una incidencia utilizando la aplicación móvil = 1.00 min

\bar{X}_A : Promedio de tiempo que el Jefe de Área se tarda en derivar una incidencia de forma manual como actualmente lo hacen = 15.00 min

σ_A : Desviación Estándar antes = 1.70

σ_D : Desviación Estándar antes utilizando la aplicación móvil = 1.49

n_A : Número de encuestados antes = 73

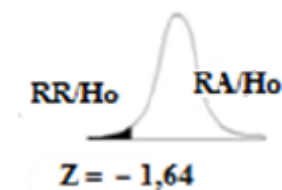
n_D : Número de usuarios que usaron la aplicación móvil = 73

Según la información recolectada, en un Pre test se obtiene un tiempo promedio de 15.00 minutos en el que los encuestados tardan en derivar una incidencia de forma manual. Después de usada la aplicación móvil, en un Post test que la base de información arroja, se obtiene un promedio de 1.00 minuto para el mismo propósito utilizando la aplicación móvil. **(Ver anexo N° 02)**

Valor experimental:

$$Z = \frac{1.00 - 15.00}{\sqrt{\frac{1.70^2}{73} + \frac{1.49^2}{73}}} = \frac{-14.00}{0.26} = -53.85$$

Región de Aceptación y Rechazo de la Hipótesis Nula:



Como $Z_0 = -53.85$ pertenece a la región de rechazo de la hipótesis nula, entonces se acepta la hipótesis alternativa.

3. Tiempo promedio de procesamiento de información / Tiempo total actual de procesamiento de información.

Tiempo promedio de procesamiento de información = tiempo de incidencias registradas + tiempo de incidencias derivadas + tiempo extra. El tiempo de incidencias registradas y tiempo de incidencias derivadas se explicó en el punto 1 y 2 respectivamente del presente capítulo. El tiempo extra en el modo habitual se refiere a que la incidencia que es registrada en mesa de partes puede suceder tres casos: 1.- Que la incidencia sea procesada el mismo día, quiere decir derivada y su pronta solución y el reclamante satisfecho, 2.- Que la incidencia, luego de ser registrada sea extraviada y no llegue a su destino y por lo tanto no hay solución a dicha incidencia y 3.- Que la incidencia llegue a su destino tardíamente, quiere decir que terceros (vecinos) ya han solucionado el problema o simplemente el hecho pasó desapercibido por varios días.

En el sistema informático propuesto, el tiempo extra se refiere a que una vez reportado la incidencia vía web o móvil por parte del reclamante el tiempo que demora el sistema en dar una respuesta depende del tipo de conexión a la red de internet en la que estamos en el momento de reportar el hecho.

Se debe mencionar que toda incidencia registrada por parte del usuario del distrito, automáticamente recibe un mensaje en su dispositivo móvil del cual reportó la incidencia; indicándole el código del registro de la incidencia reportada como indica la siguiente ilustración para que posteriormente el usuario con ese código recibido haga su consulta vía telefónica a la municipalidad sobre el estado de su registro de incidencia. **(Ver anexo 3).**

Ilustración No. 69: Se genera un código de incidencia.

CAPÍTULO 7. DISCUSIÓN

Después de ver los resultados de la implementación del sistema informático para la gestión de incidencias sociales en el distrito de Florencia de Mora – Trujillo, se encontraron aspectos importantes y oportunidades de mejora basados en la cuantificación de los indicadores propuestos para la validación de la hipótesis defendida.

De forma general, la propuesta tecnológica presentada ayuda considerablemente en el registro de incidencias sociales en el distrito de Florencia. Esta ayuda es tanto para el usuario que registra la incidencia y para la municipalidad que es la encargada de procesar dicha incidencia registrada. Además de incentivar y promover el uso de herramientas tecnológicas móviles.

Con los resultados obtenidos y mostrados en el capítulo anterior producto de la investigación determinamos:

- El uso de aplicación móvil ha disminuido en 18.50 minutos el promedio de tiempo utilizado que es de 20.00 min para el registro de una incidencia en forma manual. Este resultado se obtiene gracias a la facilidad de manejo de la aplicación móvil para registrar una incidencia en contraste con lo habitual que para reportar una incidencia se tiene que ir al municipio –le toma 20 min en promedio ir-. Esta mejora en la reducción de tiempo de registrar una incidencia mejora considerablemente el tratamiento de la incidencia social de parte del usuario y del municipio.
- El uso de aplicación web ha disminuido en 14min el promedio de tiempo utilizado (15 min) para derivar una incidencia en forma manual. Este resultado se obtiene gracias a la facilidad de la aplicación web para derivar una incidencia en contraste a lo que habitualmente hace el Jefe de Área que para derivar una incidencia registra en su cuaderno de cargos y luego deriva dicho documento (incidencia) al área u entidad externa para su posible solución. Esta mejora en la reducción de tiempo de derivar una incidencia mejora considerablemente el tiempo de la posible solución de la incidencia social presentada.
- Con el sistema informático web/móvil toda incidencia procesada (tiempo de incidencias registradas + tiempo de incidencias derivadas + tiempo extra) ha disminuido considerablemente en 55 min el promedio de tiempo utilizado (60 min) para procesar una incidencia en forma manual. Se ve claramente la gran ventaja de la aplicación web/móvil frente al modo habitual en el proceso de la incidencia.

CONCLUSIONES

De acuerdo a los objetivos establecidos en la presente investigación, se obtienen las siguientes conclusiones:

- Se ha logrado implementar un sistema informático orientado a optimizar el registro de incidencias sociales en el distrito de Florencia de Mora. Software orientado al cumplimiento de métricas de calidad que han garantizado el correcto desempeño y procesamiento de la información útil para la medición de los resultados.
- De los resultados obtenidos, se puede deducir que el registro de incidencias utilizando el aplicativo móvil, el tiempo, se reduce considerablemente (7,5% o 18.5min) por tanto es una gran ayuda a la gestión municipal y vecino del distrito.
- Se identificaron resultados favorables al momento de derivar una incidencia tanto a la oficina interna del municipio como a la entidad externa competente para la posible solución de las incidencias registradas. Este resultado favorable se traduce en la reducción del 7% o 14min el tiempo de derivar una incidencia.
- Toda incidencia procesada por el aplicativo web/móvil significa una reducción del 7% o 55min frente al procesamiento de una incidencia en el modo habitual (Manual).

RECOMENDACIONES

Se recomienda:

- Capacitar a todo el personal que labora en la municipalidad distrital de Florencia de Mora, para que ellos sean en la medida de lo posible los capacitadores de la población florenciana.
- Que, el Alcalde en sus distintas reuniones con la población de a conocer la existencia del aplicativo y repartir entre ellos la guía del usuario.
- Implementar otras opciones dentro del aplicativo que faciliten al usuario-vecino a agilizar otros tipos de trámites que desee realizar en la municipalidad como por ejemplo seguimiento de una compra-venta de un predio, etc.
- Dar a conocer de la existencia y bondades del aplicativo a otras municipalidades distritales, para que puedan ser implementadas en ellas y así contribuir de una u otra manera la pronta solución de sus problemas sociales.
- También es preciso indicar que se debe consultar con otros profesionales más experimentados en el desarrollo de este tipo de aplicaciones para poder mejorar el presente trabajo en beneficio de la población y de las autoridades del distrito.

REFERENCIAS

- Alayo, J. B. (2017). *Solución software para el proceso de pedidos en servicios gastronómicos que aporte a la fidelización de los consumidores de Trujillo*. (Tesis de Grado). Universidad Privada del Norte, Trujillo.
- Álvarez, A; de las Heras, R; y Lasa, C. (2012). *Métodos Ágiles y Scrum*. s.l. : Anaya Multimedia-Anaya Interactiva, 2012. 9788441531048.
- Abreu, J. (Noviembre 2012). Daena: International Journal of Good Conscience. 7(3) 123-130.
- DeConceptos.com. (párr. 1). Recuperado de <http://deconceptos.com/general/incidencia>.
- Fernández, J. (2014). *Implantación de un Sistema de Gestión de Incidencias*. (Proyecto final de carrera). Universidad Politécnica de Valencia, Valencia.
- Fombona, J; Rodríguez, C; Barriada, C. (2012). «Gestión de incidencias informáticas: el caso de la Universidad de Oviedo y la Facultad de Formación del Profesorado». En: «Innovación y buenas prácticas en el gobierno y la gestión de universidades» [monográfico en línea]. *Revista de Universidad y Sociedad del Conocimiento (RUSC)*. Vol. 9, n.º 2, págs. 100-114 UOC
- IPS Agencia de Noticias (2014). Aplicaciones móviles buscan la seguridad de las mujeres en India. Recuperado de <http://www.ipsnoticias.net/2014/11/aplicaciones-moviles-buscan-la-seguridad-de-las-mujeres-en-india/>
- ITIL®-Gestión de Servicios. Gestión de Incidentes. *Clasificación del Incidente*. (párr. 1-6). Recuperado de http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_incidentes/introduccion_objetivos_gestion_de_incidentes/clasificacion_y_registro_de_incidentes.php
- Lance, T. (20 de Febrero de 2016). Los 3 tipos de aplicaciones móviles: ventajas e inconvenientes [Mensaje de Blog]. Recuperado de <https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes/>
- Letelier, P. (2011). *Metodologías ágiles para el desarrollo de software: Extreme Programming*. 2011.
- Martin, R. (2004). *UML para Programadores Java*. Madrid: Pearson Educación S.A.

- Nieto, A. (08 de Febrero de 2011). ¿Qué es Android?. Recuperado de <http://www.xatakandroid.com/sistema-operativo/que-es-android>
- NOSOLOSIG (2013). Geolocalización y aplicaciones móviles para la gestión de incidencias en la vía pública. Recuperado de: <http://www.nosolosig.com/articulos/216-geolocalizacion-y-aplicaciones-moviles-para-la-gestion-de-incidencias-en-la-via-publica>
- Padilla, P. A. (2017). *Desarrollo de un sistema domótico con tecnología móvil y arquitectura ARM para reducir el consumo de energía eléctrica en los departamentos de la ciudad de Trujillo*. (Tesis de Grado). Universidad Privada del Norte. Trujillo.
- PNUD. (Noviembre 2013). RESUMEN: Informe Regional de Desarrollo Humano 2013-2014. *Seguridad Ciudadana con Rostro Humano: Diagnóstico y Propuestas para América Latina*.
- Pérez, D; Sepúlveda, J y Oliveros, Y. (2011). *Extreme Programming (XP)*. s.l. : EAE, 2011. 9783846576274.
- Rigoberto. (13 de Setiembre de 2012). Ingeniería de Software: UML y RUP [Mensaje de Blog]. Recuperado de <http://ingenieriadesoftware.rigo.blogspot.pe/2012/09/uml-y-rup.html>
- Remon, M. (2013). *Desarrollo de Aplicaciones con JAVA*. (1a. ed.). Lima: Editora Macro E.I.R.L.
- Sidnei, R. (2014). *Object-Oriented Analysis and Design for Information Systems: Modeling with UML*. s.l. : Elsevier, 2014. 9780124172937.
- Stanek, W. (2013). *Microsoft SQL Server*. Madrid: Anaya Multimedia.
- Solis, C. (2014). *Android Studio*. Recuperado de <http://manuales-para-todos.blogspot.pe/2014/11/curso-android-studio-mega.html>.
- Ye, N y Wu, T. (2014). *Developing Windows-Based and Web-Enabled Information Systems*. Reino Unido : : Illustrated, 2014. 9781482253870.

ANEXOS

Anexo No.01: Cuadro de pre-test (modo habitual) vs post-test (app móvil) de registro de incidencias.

Anexo N° 02: Cuadro de pre-test (modo habitual) vs post-test (app web) de derivación de incidencias.

Anexo No. 03: Cuadro de tiempo de procesamiento de información en modo habitual y modo web/móvil.

ANEXO No. 01

Cuadro de pre-test (modo habitual) vs post-test (app móvil) de registro de incidencias

Usuario	Pre-test (Habitual)	Post-test (Móvil)	Usuario	Pre-test (Habitual)	Post-test (Móvil)
1	22.00	2.00	38	19.00	2.00
2	20.00	2.00	39	22.00	2.00
3	24.00	2.00	40	24.00	1.00
4	22.00	2.00	41	22.00	1.00
5	16.00	1.00	42	16.00	1.00
6	21.00	2.00	43	25.00	1.00
7	20.00	2.00	44	20.00	2.00
8	19.00	1.00	45	23.00	1.00
9	22.00	1.00	46	17.00	2.00
10	23.00	2.00	47	17.00	2.00
11	25.00	2.00	48	24.00	1.00
12	18.00	1.00	49	25.00	1.00
13	22.00	1.00	50	21.00	2.00
14	19.00	1.00	51	18.00	2.00
15	16.00	2.00	52	20.00	1.00
16	19.00	2.00	53	24.00	2.00
17	25.00	1.00	54	20.00	2.00
18	24.00	1.00	55	17.00	1.00
19	15.00	2.00	56	17.00	1.00
20	24.00	2.00	57	20.00	1.00
21	16.00	2.00	58	23.00	2.00
22	24.00	2.00	59	20.00	2.00
23	17.00	2.00	60	25.00	1.00
24	19.00	1.00	61	15.00	1.00
25	19.00	1.00	62	20.00	1.00
26	16.00	2.00	63	23.00	2.00
27	20.00	1.00	64	16.00	1.00
28	21.00	2.00	65	15.00	2.00
29	19.00	2.00	66	16.00	2.00
30	20.00	1.00	67	24.00	1.00
31	24.00	1.00	68	17.00	2.00
32	22.00	1.00	69	19.00	1.00
33	19.00	2.00	70	15.00	2.00
34	24.00	1.00	71	23.00	2.00
35	25.00	1.00	72	22.00	1.00
36	20.00	2.00	73	21.00	1.00
37	25.00	2.00			

ANEXO No. 02

Cuadro de pre-test (modo habitual) vs post-test (app web) de derivación de incidencias

Usuario	Pre-test (Habitual)	Post-test (Web)	Usuario	Pre-test (Habitual)	Post-test (Web)
1	15.00	1.00	38	18.00	2.00
2	15.00	2.00	39	13.00	1.00
3	16.00	1.00	40	18.00	1.00
4	18.00	1.00	41	17.00	1.00
5	18.00	2.00	42	15.00	2.00
6	17.00	1.00	43	17.00	1.00
7	14.00	1.00	44	18.00	1.00
8	17.00	2.00	45	15.00	2.00
9	13.00	1.00	46	15.00	2.00
10	18.00	1.00	47	18.00	1.00
11	17.00	1.00	48	15.00	1.00
12	17.00	2.00	49	13.00	1.00
13	16.00	2.00	50	18.00	2.00
14	13.00	2.00	51	18.00	1.00
15	17.00	1.00	52	15.00	2.00
16	15.00	1.00	53	18.00	2.00
17	13.00	1.00	54	16.00	1.00
18	15.00	2.00	55	18.00	2.00
19	13.00	1.00	56	15.00	2.00
20	16.00	2.00	57	15.00	1.00
21	15.00	1.00	58	13.00	1.00
22	15.00	1.00	59	18.00	1.00
23	16.00	2.00	60	17.00	2.00
24	13.00	1.00	61	16.00	2.00
25	18.00	2.00	62	15.00	1.00
26	16.00	2.00	63	17.00	1.00
27	15.00	2.00	64	13.00	2.00
28	13.00	1.00	65	14.00	2.00
29	15.00	1.00	66	17.00	1.00
30	14.00	2.00	67	16.00	1.00
31	18.00	1.00	68	15.00	1.00
32	16.00	1.00	69	15.00	2.00
33	18.00	1.00	70	15.00	1.00
34	14.00	1.00	71	16.00	2.00
35	18.00	2.00	72	15.00	2.00
36	18.00	1.00	73	18.00	2.00
37	18.00	1.00			

ANEXO No. 03

Cuadro de tiempo de procesamiento de información en modo habitual y modo web/móvil.

Usuario	Modo habitual				Modo Web/Móvil			
	Tiempo de registro	Tiempo de derivación	Tiempo extra	Tiempo total de proceso	Tiempo de registro	Tiempo de derivación	Tiempo extra	Tiempo total de proceso
1	22	16	29	67	1	2	2.1	5.1
2	19	14	24	57	2	1	2.3	5.3
3	20	17	22	59	1	2	2.7	5.7
4	20	15	20	55	2	2	1.5	5.5
5	20	14	25	59	2	1	1.2	4.2
6	19	15	21	55	2	1	1.6	4.6
7	22	17	21	60	1	2	2.9	5.9
8	22	13	30	65	1	1	1.6	3.6
9	23	17	28	68	2	2	2.2	6.2
10	18	16	29	63	2	1	1.0	4.0
11	20	14	20	54	1	2	2.0	5.0
12	18	17	29	64	2	1	2.3	5.3
13	23	15	30	68	1	2	2.9	5.9
14	20	12	29	61	1	2	1.2	4.2
15	23	12	26	61	2	1	1.0	4.0
16	23	14	23	60	2	1	1.3	4.3
17	20	14	20	54	2	1	2.0	5.0
18	20	13	26	59	1	1	2.1	4.1
19	21	13	30	64	2	1	1.0	4.0
20	19	14	30	63	1	1	2.1	4.1
21	18	17	29	64	2	1	1.3	4.3
22	22	13	30	65	1	2	2.7	5.7
23	22	12	20	54	2	2	2.5	6.5
24	23	12	25	60	2	2	2.2	6.2
25	21	12	30	63	1	1	1.6	3.6
26	21	15	21	57	2	2	2.9	6.9
27	20	14	24	58	2	2	2.6	6.6
28	19	16	24	59	1	2	2.2	5.2
29	20	17	21	58	1	2	1.0	4.0
30	19	12	28	59	2	2	1.0	5.0
31	19	12	22	53	2	1	2.3	5.3
32	21	12	29	62	1	2	2.9	5.9
33	20	15	20	55	2	1	2.2	5.2
34	21	15	27	63	1	1	1.0	3.0
35	22	13	25	60	1	2	2.3	5.3

36	21	14	26	61	2	1	2.0	5.0
37	23	12	21	56	2	1	1.1	4.1
38	19	15	23	57	2	2	1.0	5.0
39	20	12	26	58	1	2	2.0	5.0
40	19	17	30	66	1	2	1.0	4.0
41	19	14	23	56	1	2	2.3	5.3
42	23	17	23	63	2	1	2.9	5.9
43	22	12	30	64	1	2	1.2	4.2
44	21	14	30	65	2	2	1.0	5.0
45	22	17	23	62	1	2	1.3	4.3
46	21	12	22	55	2	1	2.0	5.0
47	23	14	21	58	2	2	1.1	5.1
48	21	15	21	57	1	1	1.5	3.5
49	20	15	29	64	1	2	1.2	4.2
50	23	14	28	65	2	1	2.6	5.6
51	20	16	30	66	2	2	1.9	5.9
52	19	15	25	59	2	2	2.6	6.6
53	18	13	30	61	2	1	1.2	4.2
54	19	14	27	60	1	2	1.0	4.0
55	18	14	28	60	2	1	1.0	4.0
56	21	17	27	65	1	2	1.3	4.3
57	22	16	30	68	1	1	2.9	4.9
58	23	15	26	64	1	1	1.2	3.2
59	22	14	25	61	1	2	1.0	4.0
60	19	16	24	59	2	1	2.3	5.3
61	18	13	26	57	2	1	1.0	4.0
62	18	17	28	63	2	2	1.1	5.1
63	19	13	29	61	2	1	1.0	4.0
64	21	13	22	56	2	1	1.2	4.2
65	23	15	21	59	2	2	2.6	6.6
66	21	15	27	63	1	2	2.9	5.9
67	23	15	24	62	2	2	2.6	6.6
68	19	17	25	61	2	2	2.2	6.2
69	20	15	23	58	1	1	1.0	3.0
70	21	15	27	63	2	1	1.0	4.0
71	22	14	30	66	2	1	1.3	4.3
72	20	16	28	64	1	2	2.9	5.9
73	22	15	20	57	1	1	1.2	3.2