



**UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA - CETP**

Montevideo, Uruguay

**Sistema Internacional de
Catástrofes**

Febrero de 2015

Autores:

**Gastón Baldenegro
Horacio García
Ignacio Castro
Sandino Fratti
Yennifer Herrera**

Tutor:

Martín Rodríguez

Agradecimientos

A nuestras familias y amigos por acompañarnos a lo largo de este camino.

Al Ing. Martín Rodríguez por su gran apoyo, tiempo dedicado al proyecto, el cual hizo posible la realización del mismo.

Índice

Agradecimientos.....	2
Introducción.....	5
1. Objetivos.....	6
2. Estructura del Documento.....	7
3. Resumen.....	8
4. Palabras Claves.....	9
5. Estado del arte.....	10
6. Problema planteado.....	24
7. Solución Propuesta.....	26
8. Gestión de Proyecto.....	49
9. Problemas encontrados.....	50
10. Conclusiones.....	51
11. Trabajos a Futuro.....	52
12. Referencias.....	53
13. Glosario.....	55

Introducción

Los desastres tanto naturales como provocados por el hombre se están haciendo cada vez más frecuentes, estos van desde inundaciones que dejan sin hogar a cientos o miles de personas, a terremotos que dejan miles de pérdidas de vidas y cientos de familias destrozadas. Los desastres van más allá de las pérdidas tangibles ya sean físicas o materiales, estos quedan inmersos en la sociedad, en el medio ambiente causando pérdidas y daños a veces irreversibles.

Inicialmente, un desastre es un evento adverso que azota a una sociedad o medio ambiente afectando el desarrollo normal del mismo generando grandes pérdidas y a su vez el término desastre natural hace referencia a las enormes pérdidas en vidas humanas y económicas, ocasionadas por eventos o fenómenos naturales como los terremotos, inundaciones, tsunamis, deslizamientos de tierra, deforestación, contaminación ambiental.

Los fenómenos naturales, como la lluvia, terremotos, huracanes o el viento, se convierten en desastre cuando superan un límite de normalidad, medido generalmente a través de un parámetro.

Los efectos de un desastre pueden amplificarse debido a una mala planificación de los asentamientos humanos, falta de medidas de seguridad, planes de emergencia y sistemas de alerta.

A fin de la capacidad institucional para reducir el riesgo colectivo de desastres, éstos pueden desencadenar otros eventos que reducirán la posibilidad de sobrevivir a éste debido a carencias en la planificación y en las medidas de seguridad.

En resumen, los desastres no son naturales, en cambio los fenómenos si son naturales. Los desastres siempre se presentan por la acción del hombre en su entorno y siempre que provoquen daños en vidas humanas o materiales.

1. Objetivos

El objetivo del proyecto consiste en construir una plataforma de soporte ante catástrofes naturales que brindará ayuda tanto a ciudadanos como a rescatistas, ya sean pertenecientes a organismos estatales u organizaciones no gubernamentales (ONG), para mejorar la gestión de las mismas, y un proceso más rápido y efectivo en la recuperación de la población en general ante estos desastres. Facilitando la organización y localización de los puntos de riesgo tomando en cuenta diversos factores según la situación que se presenta.

El fin de la aplicación a desarrollar será una herramienta capaz de alertar sobre riesgos provocados por catástrofes, y organizar todas las tareas que se van a llevar a cabo en la etapa de gestión del desastre.

Para lograr el mismo se hará un análisis de las herramientas disponibles para desarrollar aplicaciones *web* y para dispositivos móviles. A su vez se investigará sobre aplicaciones *web* y para dispositivos móviles a fin de obtener conocimientos sobre las mismas y poder aplicarlos en el desarrollo del proyecto.

Se pretende evaluar las tecnologías disponibles, examinando ventajas y desventajas de las mismas principalmente en comparación con tecnologías para desarrollar aplicaciones *web* y para dispositivos móviles. Todo esto teniendo en cuenta las características de la solución a desarrollar.

Para lograr el desarrollo del producto se utilizarán los conocimientos adquiridos en la carrera cursada, se realizará el desarrollo de la aplicación siguiendo metodologías aprendidas en la misma.

Se va a llevar a cabo el proceso de desarrollo de las aplicaciones con su correspondiente documentación, incluyendo en ésta el relevamiento de los requerimientos, análisis, diseño y la implementación de la solución.

2. Estructura del Documento

En el capítulo 3 se presenta un resumen del documento y en el capítulo 4 se registran palabras claves.

En el capítulo 5 se documentan los conocimientos logrados en la investigación del estado del arte. En el capítulo siguiente se detalla la propuesta planteada en el proyecto.

En el capítulo 7 se documenta el proceso de desarrollo de la solución a desarrollar.

A continuación se detalla información referente a la gestión del proyecto (capítulo 8). El capítulo 9 se detalla los problemas encontrados y en los dos siguientes capítulos se desarrollan las conclusiones y los trabajos a futuro respectivamente.

Finalmente, en los capítulos 12, 13 y 14 se encuentran las referencias, el glosario de términos y los anexos del informe.

3. Resumen

Actualmente los desastres naturales se están haciendo cada vez más frecuentes.

El problema que se plantea es el desarrollo de una plataforma para el soporte ante catástrofes, para mejorar la gestión y hacer que el proceso de recuperación de las mismas sea más rápido y efectivo ante estos desastres.

Facilitando la organización y gestión de las situaciones que se presenten. En la misma los administradores serán quienes realicen el mantenimiento de las mismas, los rescatistas serán los encargados de cubrir los planes de gestión y emergencia para cada evento. Los usuarios registrados podrán estar informados sobre la situación del mismo, realizar donaciones, reportar personas desaparecidas y realizar pedidos de ayuda, el mismo tipo de solicitud que los visitantes, que son usuarios no registrados al sistema. También se dispondrá de una aplicación para dispositivos móviles donde se podrán llevar a cabo varias de las funcionalidades antes descritas.

El objetivo final es el desarrollo de la solución que cumpla con las necesidades planteadas, todo esto siguiendo un proceso formal de desarrollo de *software*, donde se documente cada una de las etapas y se realice la documentación necesaria para que se pueda seguir desarrollando distintas funcionalidades adicionales en el futuro, partiendo de todo el desarrollo y documentación realizada en este proyecto.

Para realizar el desarrollo del proyecto se realizará una investigación del estado actual de las herramientas existentes enfocadas en catástrofes y tecnologías existentes al momento en que se desarrolla el proyecto, evaluando de manera crítica las características de las aplicaciones similares y las tecnologías disponibles seleccionando las adecuadas y útiles para el desarrollo del proyecto.

4. Palabras Claves

Catástrofe, desastre, evento, Java EE, Javascript, Android, Phonegap, MySQL, JSON, JQuery Mobile, Bootstrap, Eclipse, Android studio.

5. Estado del arte

5.1. Catástrofes.

Se pretende relevar información sobre catástrofes, tanto naturales como provocadas por el hombre, organismos enfocados a la gestión de desastres y tecnologías informáticas, las cuales van a ser utilizadas en el desarrollo de una aplicación elaborada por el equipo de trabajo.

En principio, en la gestión de catástrofes existen varias etapas en que se encuentra clasificado el comportamiento y manejo de los desastres, según el organismo que clasifique, las mismas tienen distintos nombres, en el caso del Sistema Nacional de Emergencias (SINAE [1]), la etapa en que se basará el enfoque se llama:

Atención de desastres: "Etapa en la que se realizan un conjunto de acciones preventivas y de respuesta dirigidas a la adecuada protección de la población, de los bienes y del medio ambiente ante la ocurrencia de una catástrofe"

En esta fase, las aplicaciones informáticas deben tener un papel muy importante, inmediatamente de la ocurrencia del evento o dentro del período de emergencia, en el cual se deben cumplir tareas tales como evacuación, búsqueda, salvamento, rescate, asistencia sanitaria, entre otras.

Para ello se investigaron tecnologías disponibles tanto para *web* como para dispositivos móviles, las cuales pueden ser aplicadas en el desarrollo de un *software* que atienda esta problemática.

El fin de la aplicación a desarrollar será una herramienta capaz de alertar, como así de organizar todas las tareas que se van a llevar a cabo en la etapa de gestión del desastre.

5.1.1. Catástrofes: Un riesgo Inevitable.

Los desastres tanto naturales como provocados por el hombre se están haciendo cada vez más frecuentes. Los desastres van más allá de las pérdidas tangibles ya sean físicas o materiales, estos quedan inmersos en la sociedad, en el medio ambiente causando pérdidas y daños a veces irreversibles.

Inicialmente, un desastre es un evento adverso que azota a una sociedad o medio ambiente afectando el desarrollo normal del mismo generando grandes pérdidas y a su vez el término desastre natural hace referencia a las enormes pérdidas en vidas humanas y económicas, ocasionadas por eventos o fenómenos naturales como los terremotos, inundaciones, tsunamis, deslizamientos de tierra, deforestación, contaminación ambiental.

Los efectos de un desastre pueden amplificarse debido a una mala planificación de los asentamientos humanos, falta de medidas de seguridad, planes de emergencia y sistemas de alerta.

En resumen, los desastres no son naturales, en cambio los fenómenos si son naturales. Los desastres siempre se presentan por la acción del hombre en su entorno y siempre que provoquen daños en vidas humanas o materiales.

“No cabe duda de que ningún país es inmune a las catástrofes. Cuando sobreviene una catástrofe, el funcionamiento de las líneas de comunicaciones es esencial.” Dr. Hamadoun I. Touré, Secretario General de la Unión Internacional de Telecomunicaciones (UIT).

El SINAE es el organismo estatal que se encarga de la gestión de desastres en Uruguay.

Esta institución se desarrolla ya sea por sus propios medios o con la ayuda de otros organismos estatales u organismos no gubernamentales (ONG) tanto nacionales o a nivel mundial.

A nivel mundial, una de las organizaciones que se entiende como referente al pertenecer a una de las principales potencias es la *Federal Emergency Management Agency* (FEMA [2]) cuyo cometido principal es apoyar a todos los ciudadanos y a las agencias de primera respuesta y garantizar que EE.UU. trabaje en conjunto para desarrollar, mantener y mejorar la capacidad de preparación, protección, respuesta, recuperación y mitigación ante los peligros.

5.2. SINAE, referente en Uruguay.

Bajo la Ley del Poder Ejecutivo N° 18.621 (25/10/2009) se crea el Sistema Nacional de Emergencias como un sistema público de carácter permanente.

Los fines que persigue son:

- protección de las personas
- protección de los bienes de significación
- protección del medio ambiente

5.2.1. Estructura organizacional.

Se compone de tres arterias fundamentales:

Instancias de coordinación y ejecución descentralizada y primaria de actividades de prevención, mitigación, atención, rehabilitación y recuperación, ante el acaecimiento eventual o real de situaciones de desastre con impacto local, y en el marco de las políticas públicas de descentralización consagradas en la legislación nacional del Uruguay.

Comités Departamentales de Emergencias, los cuales son los órganos responsables de la formulación de políticas y estrategias a nivel departamental.

Centros Coordinadores de Emergencias Departamentales en cada departamento. Son coordinados por un funcionario de la máxima jerarquía, designado por el Intendente respectivo, con amplios conocimientos en el tema de la gestión de riesgos.

5.2.2. Sistemas de Información.

Tipos de amenazas

El SINAE tiene elaborado un inventario de amenazas a nivel Nacional

Para cada amenaza tiene una breve introducción con alguna reseña sobre máximos históricos.

Además, posee una biblioteca multimedia con: imágenes, videos, mapoteca y otras presentaciones.

Otras informaciones que brinda son: marco legal, planes e informes, publicaciones, recomendaciones, entre otros.

5.2.3. Catálogo histórico de Amenazas.

El SINAIE realizó una búsqueda de información a través de diversas fuentes (prensa escrita, informes técnicos, bases de datos y registros de instituciones públicas y de técnicos independientes, datos provenientes de sensores satelitales y sitios web) como así también la publicación "Eventos adversos y gestión del riesgo. Hacia un Sistema de Información de Riesgos" del Sistema Nacional de Emergencias realizado en el marco del Proyecto C de Unidos en la Acción Proyecto (2011).

Gracias a todo esto confeccionó un catálogo histórico.

Tipos de históricos registrados:

- Fenómenos Meteorológicos
- Incendios
- Inundaciones

5.2.4. Geoservicios *Web Map Service (WMS)* y *Web Feature Service (WFS)*.

Los geoservicios son servicios web de información geográfica, regidos por estándares [Open Geospatial Consortium \(OGC\)](#) e *International Organization for Standardization (ISO)*, permiten que las aplicaciones puedan consumir información geográfica de forma automática a través de internet desde diferentes fuentes o servidores. Los tipos de geoservicios más comunes son: los *Web Map Service (WMS)* y los *Web Feature Service (WFS)* que brindan información geográfica en formato imagen y vectorial respectivamente.

5.2.5. Geoservicio *WFS* del SINAIE.

La *WFS* es un servicio estándar del *OGC* y permite realizar consultas básicas, la recuperación y edición de los elementos geográficos.

Las capas ofrecidas por el SINAIE:

- Incendios - (Forestales, Urbanos, de Campo)
- Inundaciones - (Gradual y Repentina)

- Fenómenos Meteorológicos - (Granizo, Ola de Frío, Tornados, Tromba Marina, Tormentas y Vientos Fuertes)

5.2.6. Geoservicio WMS del SINAE.

El servicio *WMS* también definido por la *OGC*, es un estándar de representación de información geográfica en forma de un archivo de imagen digital.

Capas ofrecidas por el SINAE:

- Incendios (forestales, de interface, de campo, urbanos)
- Inundaciones (gradual y repentina)
- Fenómenos Meteorológicos (granizo, ola de frío, tornados, tromba marina, tormentas y vientos fuertes)

Mapa de Combustibles Vegetales (categorías de combustión de coberturas vegetales).

5.2.6.1. Visualizador de mapas.

De forma similar a los Geoservicios ofrecidos por el SINAE, se brinda un visualizador de Mapas propios en los cuales también se pueden observar las capas existentes a través del servicio *WMS*.

5.2.6.2. Utilización del Geoservicio WMS.

Superposición de capa *WMS* sobre una capa de Google Maps [3].

Las capas de Google Maps utilizan la proyección de *Mercator* que parte de un modelo esférico de la tierra.

Si se quiere superponer capas *WMS*, que se tienen en otra proyección, sobre las de Google Maps, son necesarios ajustes de código fuente al momento de crear la instancia del mapa.

Los cambios en el código van a permitir que la capa *WMS* herede la proyección base del mapa que en este caso es la *Mercator*.

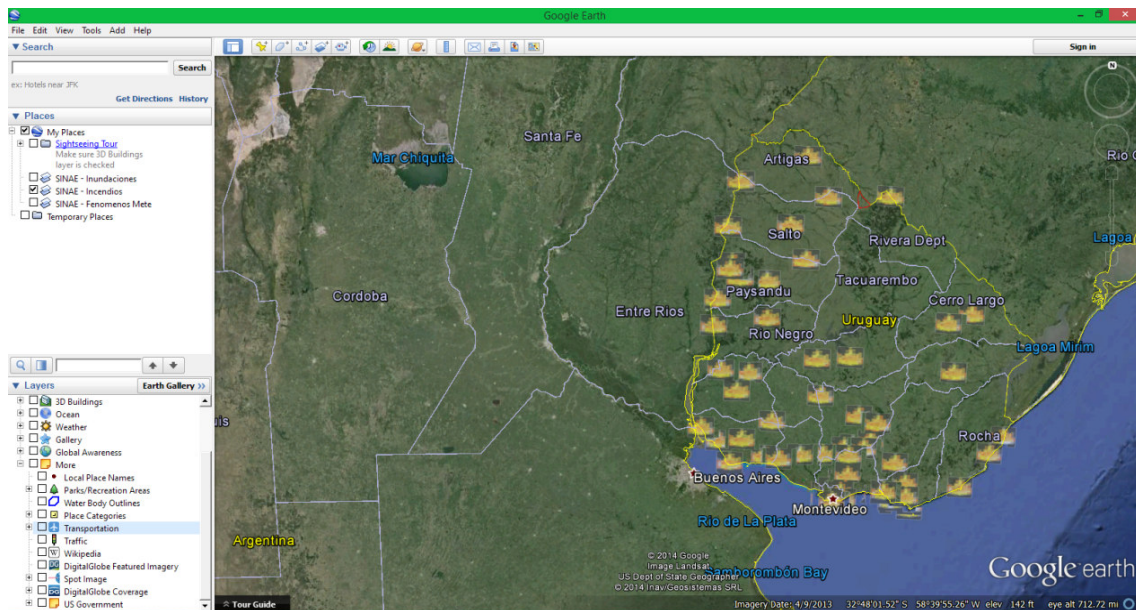


Imagen 1 - Capa Incendios en Google Maps - WMS SINAE

5.3. Aplicaciones enfocadas en Catástrofes

5.3.1. PDC Disaster Alert [4]



Imagen 2 – Disaster Alert

La aplicación despliega eventos activos en una forma interactiva en un mapa global, y en una lista a medida que se están produciendo los mismos en todo el mundo.

Información adicional sobre el evento puede ser visualizada y compartida.

“Eventos activos” hace referencia a una recopilación de incidentes actuales y en tiempo real que son un compilado de fuentes autorizadas, que han sido designados “potencialmente peligrosos para las personas, los bienes o activos” por la aplicación PDC Disaster AWARE [5].

La aplicación se puede descargar gratuitamente. La misma actualmente incluye: huracanes, ciclones tropicales, terremotos, inundaciones, eventos provocados por el hombre, fenómenos marinos, erupciones volcánicas, tormentas, tsunamis.

Entre las características más destacadas se incluyen:

- visualizar eventos activos en un mapa interactivo o en lista de alerta
- búsqueda de ubicaciones
- seleccionar distintos tipos de mapas de fondo
- visualizar más información de los eventos activos
- actualizaciones automáticas cada cinco minutos
- selección de zona horaria predeterminada

5.3.2. ubAlert [6]

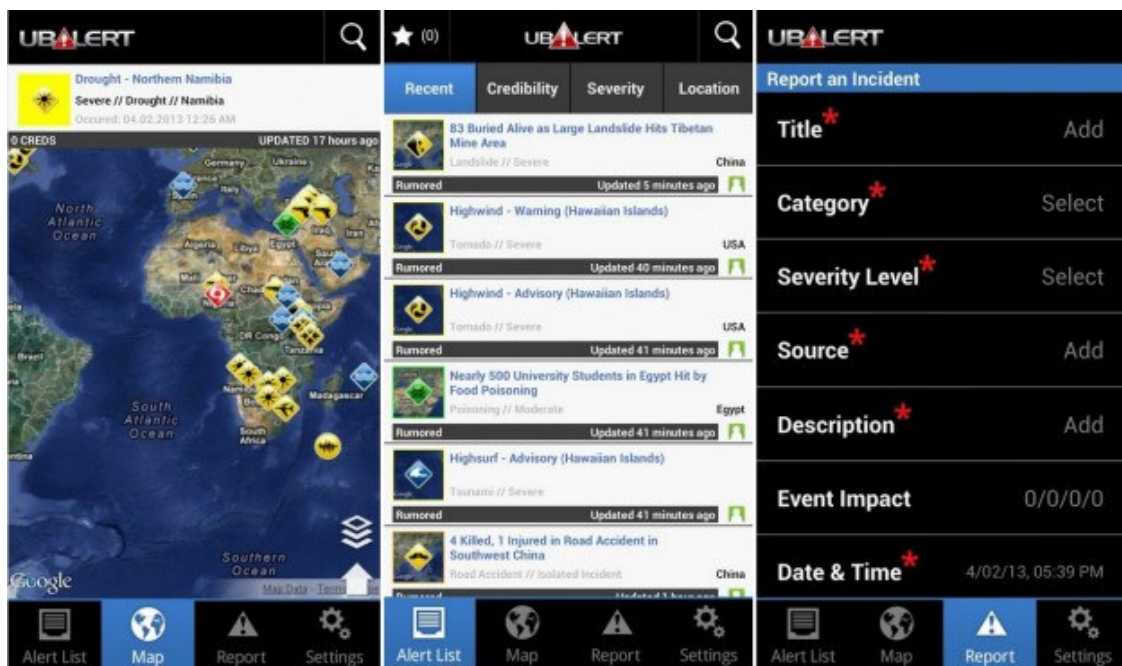


Imagen 3 – ubAlert aplicación móvil Android

La aplicación ubAlert brinda toda la información sobre desastres que suceden en todo el mundo, las personas tendrán acceso en tiempo real a todo lo que necesiten saber acerca de un evento. Las alertas contienen detalles básicos de eventos, estadísticas de impacto, mapas, imágenes, videos. El usuario puede compartir al instante alertas con amigos o familiares para sacarlos de peligros a través de correo electrónico, Facebook [7] o Twitter [8].

La aplicación es una red de alertas de desastres en el mundo, mediante la combinación de datos de las instituciones globales. La aplicación alerta de las emergencias, validando la confiabilidad del contenido reportado y alerta inmediatamente a los que puedan verse afectados, dependiendo de la severidad y ubicación.

Características básicas incluyen:

- Mapa interactivo
- Seleccionar acontecimientos por gravedad /renombre /credibilidad
- Informar de un evento y presentar una foto o un vídeo
- Compartir eventos por correo electrónico, publicar enlaces a Facebook o Twitter
- Guardar alertas a su lista de “vigilancia” para la visualización rápida
- Proporcionar retroalimentación sobre la confiabilidad del mensaje para evitar mensajes fraudulentos
- Ajustar la configuración para mostrar la distancia en kilómetros o millas

5.3.3. RSOE EDIS Notifier Lite [9]

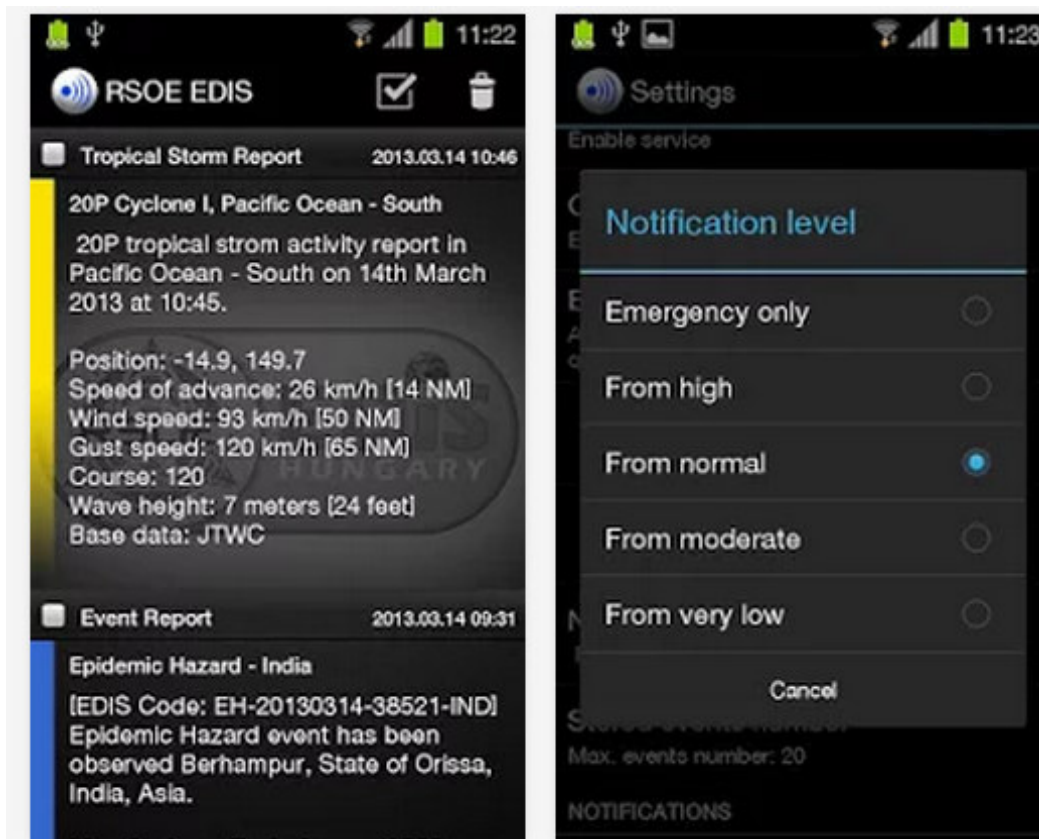


Imagen 4 - RSOE aplicación móvil Android

Opera en el marco de su propio sitio *web*, tiene como objetivo monitorear y documentar todos los eventos a nivel mundial que puedan causar desastres o emergencias. La idea es la de gestionar la información sobre los eventos que ponen en peligro la seguridad de la población. Utiliza la velocidad y el espectro de datos de internet para obtener información. Monitorea y obtiene datos de varias organizaciones extranjeras para hacerse de la información rápidamente y certificada. Los datos se publican de forma continua en su propia página web y también en la aplicación para dispositivos móviles RSOE EDIS *Notifier Lite*.

La aplicación proporciona información en tiempo real sobre los eventos de catástrofe sucedidos en el mundo. Todo lo que se necesita es un teléfono con Android y una conexión a Internet. Hay cinco niveles de notificación (5 - emergencia, 4 - alerta, 3 - advertencia, 2 - notificación, 1 - de información), cada uno de los que puede asignar un sonido de notificación diferente para cada uno. También puede configurar las llamadas horas de silencio durante el cual sólo recibe nivel 5 mensajes. Si se prefiere no tener ningún sonido de notificación, puede activar el encendido del LED o el modo de vibración en el menú de ajustes.

Pinchando sobre los iconos del evento aparece información detallada sobre la situación: descripción, peligros, riesgos cercanos y fotos o videos en caso de que estén disponibles.

5.3.4. FEMA, aplicación para móviles.

1.1.1.



Imagen 5 – FEMA para móviles

Con la aplicación de FEMA para dispositivos móviles el usuario podrá obtener consejos de seguridad ante desastres, listas interactivas para ayudarlo a preparar un equipo de suministros de emergencia, y un lugar dónde almacenar información sobre su lugar de encuentro en caso de emergencias. También encontrará un mapa con información sobre los refugios y Centros de Recuperación por Desastre (DRC) de FEMA abiertos en su área.

La aplicación también tiene una función para reportar desastres, que permite a los usuarios tomar una fotografía en una zona de desastre y la enviará, junto con una breve descripción de texto, más tarde los mismos son aprobados y se pueden desplegar en un mapa público para que otros usuarios lo vean.

5.3.5. Comparación de aplicaciones móviles.

	Disaster Alert	ubAlert	RSOE EDIS	FEMA
Mapa Interactivo para desplegar todos los desastres	SI	SI	NO	NO
Visualizar detalles del evento al hacer <i>click</i>	SI	SI	NO	NO
Lista de eventos	SI	SI	SI	
Compartir eventos por correo o redes	NO	SI	NO	SI
Reportar eventos	SI	SI	NO	NO
Noticias	NO	SI	NO	SI
Agregar comentarios	NO	SI	NO	NO
Adicionar datos, fotos, video o enlace	SI	SI	NO	NO
Realizar Donaciones	NO	NO	NO	SI
Cambiar fondo mapa	SI	SI	SI	NO
Postularse como voluntario	NO	NO	NO	SI
Ubicación refugios, lugares de encuentro.	NO	NO	NO	SI

Imagen 6 – Tabla comparativa de aplicaciones

5.4. Actualidad de los dispositivos móviles.

No cabe duda de que la movilidad ha sido un cambio de paradigma en el consumo de contenidos digitales casi tan importante como lo ha sido la popularidad que tomó internet.

Usando la tecnología existente se ha creado un mundo que prácticamente transcurre a través de internet, llamado "mundo *online*".

Se compra en las tiendas *online*, buscamos en las redes sociales a las personas que nos interesan, por ejemplo gente que conocemos personalmente o por el nombre, e inmediatamente la buscamos en las redes sociales para saber mucho más sobre ellas.

En la actualidad hay mucha gente que ve sus series o películas favoritas, utilizando sus dispositivos móviles y por ejemplo opinan en tiempo real sobre lo que están viendo.

Las nuevas generaciones se sienten muy cómodas usando dispositivos móviles. Las personas adultas e inclusive muchos niños tienen sus propios dispositivos móviles y lo consideran algo muy importante en sus vidas.

Los dispositivos móviles no se utilizan únicamente para hablar por teléfono se utilizan en gran tipo de aplicaciones que facilitan la experiencia.

Los dispositivos móviles se han convertido en una herramienta informativa muy importante y en algunos casos ha superado los medios de comunicación tradicionales como son la prensa escrita, la radio y la televisión, motivo por el cual se piensa que esta tecnología supondrá el final de ediciones impresas, por ejemplo.

No hay duda de que las nuevas tecnologías están cambiando nuestros hábitos de conducta.

Las empresas y los estados no pueden ignorar estos cambios y tendrán que redefinir sus estrategias para adaptarse a esta nueva situación.

Ante este escenario, hoy ya no basta con diseñar y ejecutar una buena estrategia de gestión o planificación ante las emergencias, es necesario utilizar las nuevas tecnologías que nos permitan unir gestión, planes e internet, y crear sinergias entre ellos.

La tecnología puede ayudarnos, nos permite ofrecer a los usuarios la posibilidad de registrarse, pedir ayuda, reportar personas desaparecidas y visualizar o acceder a contenidos amenos, bien estructurados, fáciles de digerir, acompañados de videos, mapas, infografías, opiniones, información. y sobre todo adaptados perfectamente a los dispositivos móviles.

Un ejemplo podría ser crear en Twitter un *hashtag* específico para el evento que generó la emergencia, y tener a la población informada, de este modo se generarán comentarios y podría la gente usar los canales que ofrecen las nuevas tecnologías.

Hay varias organizaciones o asociaciones que ya están utilizando la tecnología para reportar dichos eventos y utilizando el poder de internet, pero las restantes se deben plantear una evolución en los modelos de gestión y planificación de las emergencias para conectar personas y en general todo aquello que se gestione o controle a través de internet.

Además, hay que tener en cuenta que los *smartphones* ofrecen infinidad de herramientas que facilitan las interacciones, vamos a mencionar algunas de ellas: códigos *quick response* (QR), geo localización, notificaciones, *Short Message Service* (SMS), social *log-in*, aplicaciones móviles.

Teniendo las aplicaciones móviles para gestionar las emergencias los organismos públicos o asociaciones podrían conseguir una participación más activa por parte de la ciudadanía luego de sucedida la misma.

En las administraciones públicas o las asociaciones se manejan y gestionan una gran cantidad de datos de diferentes tipos sobre un evento que provocó una catástrofe, éstos pueden ser datos relacionados a personas desaparecidas, solicitudes de ayuda, daños en rutas o inundaciones por ejemplo. Si se publican a través de portales *online* o aplicaciones móviles éste tipo de información podría ser utilizada por los ciudadanos comunes, rescatistas u organizaciones para evitar daños mayores o poder mejorar la respuesta ante las emergencias surgidas por una catástrofe de cualquier tipo. Este tipo de entornos contribuyen además a que la gestión de las catástrofes haga que los ciudadanos se sientan más seguros y parte de la solución de la misma al poder contribuir o ayudar.

También la tecnología puede ayudar a crear una sociedad verdaderamente más solidaria. Concretamente los dispositivos móviles, como los *smartphones*, *tablets* o incluso teléfonos básicos pueden proporcionar a los ciudadanos un medio para comunicarse con otros aparatos integrados a las organizaciones que estén liderando o manejando la gestión de la catástrofe, y al mismo tiempo pueden ser un canal para notificar a los gestores una retroalimentación adecuada, que les facilite la toma de decisiones con la información que se genera.

Pero para pasar a gestionar las catástrofes a través de aplicaciones *web* o móviles no podemos centrarnos solo en la tecnología, hay que poner foco también en otras cosas. No tiene ningún sentido implementar grandes soluciones tecnológicas si no están enmarcadas en un plan en el que encajen y con unos claros objetivos a los que realmente contribuyan. Por tanto es necesario un plan, una visión integral que implique todos los ámbitos de actuación ante la gestión de los eventos.

5.4.1. Android [10]

En el mundo de dispositivos móviles, la mayoría tiene bastante claro que Android es uno de los principales referentes en sistemas operativos para dispositivos móviles.

Hace algunos años, Google [11] decidió expandir su negocio hacia los dispositivos móviles y decidió crear un sistema operativo móvil propio, gratis y con varios de los más grandes fabricantes de celulares como respaldo.

Android es un sistema operativo basado en el núcleo Linux [12] que fue desarrollado inicialmente por *Android Incorporation*, una firma comprada por Google en 2005.

Desde su aparición, Android revolucionó el mercado de los *smartphones* e inició una carrera tecnológica que continúa al día de hoy.

Android tiene la ventaja que es un sistema operativo que puede ser adoptado por cualquier fabricante de celulares, aunque existe un consorcio de los fabricantes más importantes (Samsung [13], Sony [49], LG [15], HTC [16] entre otros) y permite realizar tareas que se asemejan a una computadora personal, como navegar la *web*, leer *e-mails*, descargar aplicaciones.

Además de todo esto, Android cuenta con las aplicaciones Google, que permiten acceso a Google *Maps*, YouTube [17], Gmail [18] y muchas más aplicaciones oficiales con solo ingresar una cuenta Google. También, cuenta con Google Play [19], una tienda con cientos de miles de aplicaciones pagas gratis para descargar, con categorías que van desde juegos a productividad, pasado por estilo de vida, utilidades y mucho más.

Android está basado en Linux, un sistema operativo libre, gratuito y multiplataforma. Dalvik es la actual máquina virtual de Android, la cual será sustituida por Android runtime (ART), a partir de la versión 4.5 del sistema operativo.

Existen múltiples herramientas gratuitas para empezar a programar aplicaciones para Android (Android Studio [20], Eclipse [21]), las cuales han hecho posible el crecimiento de la comunidad de programadores que han optado por desarrollar para esta plataforma, generando una extensa y variada lista de aplicaciones.

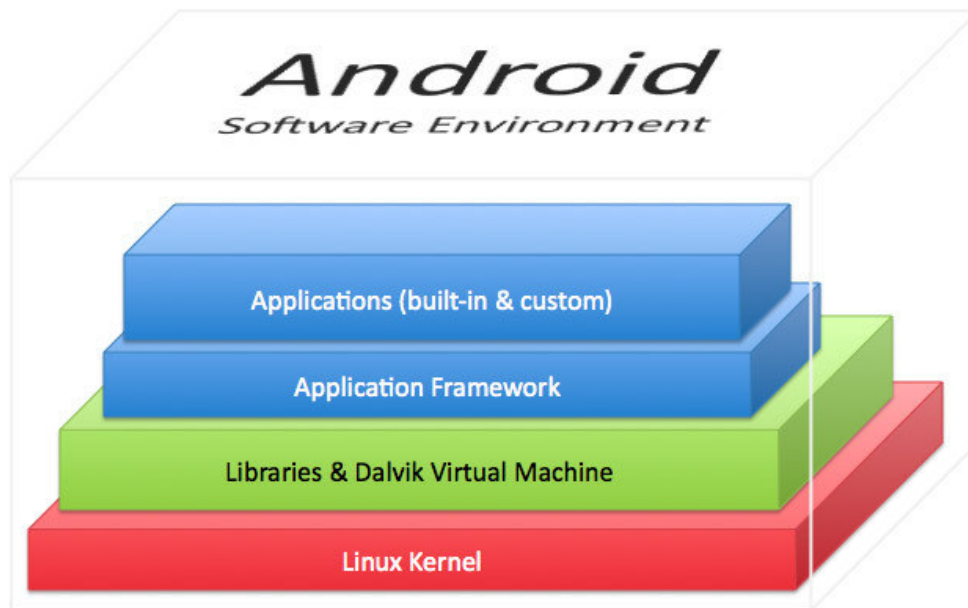


Imagen 7 - Estructura del Sistema Operativo *Android*

En la tienda Google Play se encuentran diversas aplicaciones relacionadas con la temática del proyecto, las cuales son presentadas a continuación.

5.4.2. PhoneGap [22]

Al desarrollar un proyecto móvil es complejo decidir para que tipo de plataformas se realizará. Generalmente se opta por algún sistema operativo, pero sucede muchas veces que el desarrollo se ve castigado por el costo y el tiempo hasta la puesta en producción.

En este punto entra en juego PhoneGap, ya que su principal ventaja es que se puede utilizar el mismo código para las plataformas soportadas por el *framework*, que son: Android, Blackberry 10 [23], Firefox OS [24], IOS [25], Ubuntu [26], Windows Phone 7/8 [27], Windows 8 [28], Tizen [29].

Respecto al listado de sistemas operativos móviles soportados, cabe que acotar que no es una verdad absoluta el soporte al 100%. Para algunas plataformas es necesario realizar pequeñas modificaciones en el código JavaScript [30] y HTML [31] de PhoneGap, según las características definidas por el fabricante de la plataforma nativa.

Las aplicaciones resultantes son híbridas, o sea no son realmente aplicaciones nativas al dispositivo, pero tampoco se trata de aplicaciones *web*.

Phonegap se puede utilizar localmente y luego empaquetar con la plataforma que se tiene instalada. Además, brinda la opción de poder empaquetar en la nube para cualquier plataforma, sin necesidad de tener la misma instalada y esto es posible gracias a Phonegap *Build* [32].

Las APIs que maneja PhoneGap ofrecen acceso a varias funcionalidades del dispositivo móvil como ser: el acelerómetro, la cámara, aplicaciones de capturas de audio y video; permite trabajar con las conexiones de red que cuenta, contactos almacenados, obtener datos relevantes del mismo, maneja eventos, se pueden trabajar con archivos binarios, se puede obtener la posición geográfica, permite utilizar notificaciones, acceso al vibrador y facilita el uso de base de datos. Estas API se conectan al sistema operativo usando el código nativo del sistema huésped a través de una interfaz de funciones foráneas en Javascript. Con todas estas APIs se pueden desarrollar aplicaciones realmente atractivas basadas en tecnologías *web* y que tengan acceso a *hardware* del dispositivo, con el fin de hacerlas más nativas.

6. Problema planteado

Según la letra que se presentó en el trabajo a realizar, la misma hace referencia a una plataforma de soporte ante catástrofes naturales, el propósito de la misma es centralizar toda la información relevante y de socorro en un portal *web*, que permitirá acceder a las últimas novedades de cada catástrofe sucedida, se podrá reportar personas desaparecidas, se deberá poder ingresar solicitudes de ayuda, también se podrán llevar a cabo de varios tipos de donaciones a las catástrofes a través de las ONG relacionadas a las mismas. El sitio *web* será la arteria principal para filtrar toda la información relevante existente en la *web* relacionada al incidente o evento, la cual puede provenir de orígenes tan diversos como ser YouTube, CNN [33], BBC [34].

En la letra se plantea estudiar y aplicar en la práctica las tecnologías Java Enterprise Edition [35], Android SDK [36], Phonegap.

Además, evaluar los servicios que brinda el SINAÉ.

La propuesta plantea la existencia de cuatro tipos de usuarios, ellos son:

- Administrador se encarga principalmente de realizar tareas de mantenimiento y administración permitiendo la instanciación del sistema ante nuevos eventos.
- Rescatistas su cometido será ejecutar tareas de atención de catástrofes y recuperación de éstas.
- Visitante es cualquier usuario que se encuentre en situación de acceder a la *web*.
- Usuario registrado puede solicitar ayuda, reporta y busca personas desaparecidas y por ejemplo puede realizar donaciones de varios tipos entre ellas las económicas podrán ser realizadas a través de tarjeta o PayPal [37] .

En último lugar se plantea desarrollar una aplicación Android nativa que brinde a los usuarios la posibilidad de pedir ayuda en caso de emergencia, y una utilizando Phonegap que permita a los rescatistas ejecutar los procedimientos en caso de emergencia.

En resumen, luego de la investigación de distintas tecnologías (lenguajes, *frameworks*), el listado final es el siguiente:

- Java como lenguaje base para la plataforma *web*
- JBoss [38] como servidor de aplicaciones Java *Enterprise Edition* (Java EE o J2EE)
- Spring [39] como *framework* Java para desarrollo web
- Eclipse como *Integrated Development Environment* (IDE) para Java
- Android como sistema operativo móvil
- Android Studio como IDE para Android
- Phonegap para la creación de aplicaciones móviles híbridas

7. Solución Propuesta

2.

7.1. Introducción.

Para elaborar la solución a la problemática planteada, se dividió el proyecto en las siguientes etapas:

1. Análisis
2. Diseño
3. Implementación
4. Pruebas
5. Mantenimiento

7.1.1. Misión.

Ayudar a las personas a nivel mundial a mitigar las consecuencias de las catástrofes, evitando las pérdidas de vidas humanas y económicas.

7.1.2. Visión.

Ser la aplicación de *software* referente en el mundo, para gestionar y mejorar el tratamiento de las catástrofes.

7.2. Análisis

Los objetivos que se plantean al realizar un análisis del problema es poder modelar el dominio del mismo para poder comprender mejor el contexto del problema y contar con una primera aproximación de la estructura de la solución.

En esta etapa se logra una descripción más precisa de lo que se espera del sistema.

7.2.1. Relevamiento de requerimientos

Para el relevamiento de requerimientos se partió de la problemática planteada y del conocimiento adquirido en la investigación inicial.

Para más referencias, ver anexo [A] Requerimientos de Software.

7.3. Alcance

Este sistema brindará diversas funcionalidades tanto para los rescatistas y administradores del mismo que les permiten llevar un control de los incidentes y un flujo de trabajo previamente planificado y pautado.

Para el usuario general las posibles víctimas, SIC brinda un tenant al cual acceden, este es específico referente a cada catástrofe, en el cual ellos podrán hacer un seguimiento del estado de la misma, mantenerse informados, pedir ayuda y realizar donaciones.

Para este tipo de usuario parte del sistema está replicado en una versión móvil en la cual podrán pedir ayuda rápidamente y llevar un seguimiento de las diversas catástrofes en la región en la cual se encuentran situados.

El rescatista además de poder gestionar sus acciones dentro de una catástrofe mediante un BackOffice (desde ahora llamaremos BO). Podrá hacerlo a través de su dispositivo móvil ya que SIC brinda una versión phonegap del flujo de trabajo dentro de las catástrofes.

7.3.1. Planificación para lograr el Alcance

A continuación se listan las funcionalidades con sus casos de usos y el detalle de la iteración en la que se van a implementar.

Iteración 1

- ABM Catástrofe
 - Alta Catástrofe
 - Baja Catástrofe
 - Modificación Catástrofe
 - Listado Catástrofe
- Pedido de Ayuda
 - Ayuda Web
 - Ayuda Mobile
- Realizar Donación
 - Donación Monetaria
 - Donación Material
 - Donación de trabajo

Iteración 2

- ABM Usuario
 - Alta Usuario (Administrador y Rescatista)
 - Baja Usuario
 - Modificación Usuario
 - Listado Usuario
 -
- ABM Clientes

- Alta Cliente
- Baja Cliente
- Modificación Cliente
- Listado Cliente
- Login/Logout
 - Login Facebook
 - Login twitter
 - Login Propio
- ABM ONGs
 - Alta ONG
 - Baja ONG
 - Modificación ONG
 - Listado ONG

Iteración 3

- Gestión de Desaparecidos
 - Administración Desaparecidos
 - Reporte Desaparecidos
 - Búsqueda Desaparecidos
- Gestión de Riesgo
 - Alta Gestión de Riesgo
 - Modificación Gestión de Riesgo
 - Listado Gestión de Riesgo
- Plan de Emergencia
 - Alta Plan de Emergencia
 - Modificación Plan de emergencia
 - Listado Plan de Emergencia
 - Asignación de Planes de Emergencia

Iteración 4

- Vistas Reportes
 - Donaciones en el tiempo
 - Solicitudes de ayuda en el tiempo
 - Navegabilidad del sitio *Web*
- Noticias e Información
 - Gestión de información de interfaces externas varias

Fase Elaboración

Iteración 1

- Alta Catástrofe
- Acceso Catástrofe (sitio/móvil)
- Pedido de Ayuda
 - *Mobile*
 - *Web*
 - Envío de mail de notificación
- ABM Ongs (BO)
 - Alta Ong
 - Baja Ong
 - Listado Ong
 - Asociación con las catástrofes.
- Listado de Ongs (sitio)
- Realizar Donación (sitio)
 - Monetaria
 - Material

Iteración 2

- Desaparecidos (sitio)
 - Alta
 - Listado (también en el móvil)
 - Envío de mail de notificación
- Catástrofe (BO)
 - Modificación
 - Listado
 - Baja
- ABM Usuarios Administradores/Rescatistas (BO)
 - Alta
 - Baja
 - Modificación
 - Listado

Iteración 3

- Registro Usuario
 - Registro Redes sociales (sitio, móvil)
 - Registro Propio (sitio, móvil)
 - Envío de notificación.
- Login/Logout (sitio, móvil)

- Login propio y con redes sociales
- Logout del sitio
- Listado de Noticias (sitio)
 - Integración con Google Feed API [40]
 - Despliegue embebido en el sitio.

Iteración 4

- Gestión de Desaparecidos (BO)
 - Administración Desaparecidos
 - Reporte Desaparecidos
 - Búsqueda Desaparecidos
- Gestión de Riesgo (BO, Phonegap)
 - Alta Gestión de Riesgo
 - Modificación Gestión de Riesgo
 - Listado Gestión de Riesgo
- Plan de Emergencia (BO, Phonegap)
 - Alta Plan de Emergencia
 - Modificación Plan de emergencia
 - Listado Plan de Emergencia
 - Asignación de Planes de Emergencia

Iteración 5

- Vistas Reportes
 - Donaciones en el tiempo
 - Solicitudes de ayuda en el tiempo
 - Navegabilidad del sitio *Web*

Fase Construcción

Iteración 1

- Alta Catástrofe (BO)
- Listado Catástrofe (móvil)
- Pedido de Ayuda
 - Mobile
 - *Web*
 - Envío de mail de notificación
- Listado de Ongs (sitio)
- Realizar Donación (sitio)
 - Monetaria
 - Material

Iteración 2

- Desaparecidos (sitio)
 - Alta
 - Listado (también en el móvil)
 - Envío de mail de notificación
- Catástrofe (BO)
 - Modificación
 - Listado
 - Baja
- ABM Ongs (BO)
 - Alta Ong
 - Baja Ong
 - Listado Ong
 - Asociación con las catástrofes.

Iteración 3

- Registro Usuario
 - Registro Redes sociales (sitio, móvil)
 - Registro Propio (sitio, móvil)
- Login/Logout (sitio, móvil)
 - Login propio y con redes sociales
 - Logout del sitio
- ABM Usuarios Administradores/Rescatistas (BO)
 - Alta
 - Baja
 - Modificación

- Listado
- Listado de Noticias (sitio)
 - Integración con Google Feed API
 - Despliegue embebido en el sitio.

Iteración 4

- Gestión de Desaparecidos (BO)
 - Administración Desaparecidos
 - Reporte Desaparecidos
 - Búsqueda Desaparecidos
- Gestión de Riesgo (BO, Phonegap)
 - Alta Gestión de Riesgo
 - Modificación Gestión de Riesgo
 - Listado Gestión de Riesgo
- Plan de Emergencia (BO, Phonegap)
 - Alta Plan de Emergencia
 - Modificación Plan de emergencia
 - Listado Plan de Emergencia
 - Asignación de Planes de Emergencia

Iteración 5

- Vistas Reportes
 - Donaciones en el tiempo
 - Solicitudes de ayuda en el tiempo
 - Navegabilidad del sitio *Web*

7.4. Funcionalidades en general.

SIC es un sistema web compuesto por un BO y un cliente multi-tenant, el BO es utilizado exclusivamente por los administradores y/o rescatistas en el cual ellos podrán crear instancias de sitios clientes para cada catástrofe, además de administrar y gestionar datos como usuarios, rescatistas, organizaciones no gubernamentales (ONG), riesgos y planes de emergencia con los cuales maneja un workflow de acciones a tomar para cada catástrofe basado en los riesgos y planes generados.

A partir de estas catástrofes el BO accede a reportes varios como son la navegabilidad de cada cliente *tenant*, personas desaparecidas, donaciones realizadas y pedidos de ayuda.

El BO brinda una lista de catástrofes activas, las cuales se pueden visualizar e identificar en un mapa en su sitio *tenant* donde se distingue la región afectada.

Cada sitio *tenant* permite el clásico registro completando los datos de usuario y contraseña en el sitio. El usuario tiene la posibilidad de acceder a diversas fuentes de información embebidas en SIC, reportar y buscar desaparecidos, pedir ayuda y realizar donaciones a las diferentes ONG, cada cliente SIC permite visitantes (usuarios no registrados) los cuales acceder a todas las funcionalidades del mismo a excepción de reportar desaparecidos y realizar donaciones.

En cada cliente se muestra el mapa con la región afectada y los pedidos de ayuda en gestión.

SIC Mobile es una versión genérica y reducida del SIC cliente *web*, éste indica al usuario que sus datos serán obtenidos por la aplicación para el registro en el sistema al momento de la instalación, si no los consigue del dispositivo solicita los mismos para que sean ingresados manualmente.

SIC Mobile identifica la ubicación del usuario y si está dentro de la región de alguna de las catástrofes se centra en ella pero no lo limita a acceder a las otras si así lo desea. Por lo tanto brinda un listado de catástrofes activas, a partir del cual el usuario podrá acceder a cada una de ellas.

SIC Mobile permite solicitar ayuda a través de un formulario o del botón de pánico, ambos toman la ubicación del usuario para realizar la solicitud de ayuda. La ubicación del mismo solo será accesible si se encuentra en una región de catástrofe, además de esto mantiene informado a los usuarios de los desaparecidos reportados y las nuevas noticias para las catástrofes.

7.4.1. Descripción general

7.4.1.1. Comunicación entre las aplicaciones

SIC y SICMobile se comunican a una capa de servicios mediante protocolo http, esta capa de servicio es la encargada de comunicarse con la lógica la cual accede directamente al *Data Access Object* (DAO, Objeto de Acceso a Datos) capa de

datos que van contra la base de datos MySQL [41] mediante *mysql-connector-java*.

7.4.1.2. Interface con software

Consume diversos *Really Simple Syndication* (RSS) como por ejemplo los de CNN o BBC con lo cual mantener los sitios bien informados para esto se utilizará Google Feed Api.

Para la gestión de mapas y regiones se utilizó Google Developers *Google Maps API*, *google analytics* para manejar el acceso y uso del cliente en la aplicación.

7.4.2. Funciones específicas del sistema

7.4.2.1. BackOffice

Alta de Usuarios: al crear un usuario se deben ingresar los datos requeridos nombre, apellido, nombre de usuario (*nickname*), correo electrónico (*email*), contraseña (*password*), celular y tipo de usuario.

El nombre de usuario se controla que sea único en el sistema y la contraseña se oculta utilizando el algoritmo criptográfico *Message-Digest Algorithm 5* (MD5).

Se pueden ingresar también fecha de nacimiento, sexo y dirección.

Cuando se selecciona tipo de usuario rescatista, el sistema permite ingresar los datos: tipo de rescatista, domicilio y situar la ubicación geográfica de esa residencia para persistir la latitud y longitud de la misma.

Modificación de Usuario: se pueden modificar todos los datos de éstos a excepción del nombre de usuario y el tipo de usuario.

Baja y Listado de Usuarios: se presenta una vista donde se listan los usuarios del *backend* del sistema y se visualiza una grilla con datos pertenecientes a los usuarios, la baja de estos usuarios es lógica.

Inicio de sesión: Dependiendo del tipo de usuario que inicie sesión serán las funcionalidades que éste podrá realizar. El sistema solicita el *nickname* y la contraseña del usuario, y controla que estos coincidan entre sí con un usuario activo.

Alta ONG: al dar de alta una ONG se solicitan el nombre de la ONG, dirección, teléfono, *email*, página *web*, origen, datos de la cuenta *PayPal*.

Modificación ONG: todos los datos de la misma se pueden modificar.

Baja ONG: la baja de éstas es física dado que la relación con las catástrofes no se pierde ya que es una tabla no relacionada que se copia en cada catástrofe.

Listado ONG: despliega un listado con todas las ONG registradas.

Reportes: el sistema permite a los usuarios administradores listar las donaciones en el tiempo, las solicitudes de ayuda en el tiempo y uso del sitio *web*

.

7.4.2.2. Sitios Catástrofes.

Pedido de ayuda en dispositivos móviles: posee un botón de pánico en el cual el usuario con solo pulsarlo envía una alerta con sus coordenadas. La aplicación reconoce a qué catástrofe corresponde el pedido de ayuda en base a las coordenadas de la ubicación de la víctima.

Pedido de ayuda web: en la pantalla de pedido de ayuda *web* el usuario anónimo o registrado ingresará teléfono, mensaje y ubicación.

Realizar Donación: la funcionalidad de realizar donación es exclusiva de la *web*, las donaciones pueden ser de tres tipos: económicas, de bienes o de servicios. El usuario debe seleccionar la ONG a la cual realizará la donación, además del tipo de donación que desea realizar.

Reportar desaparecido: el usuario que reporta un desaparecido debe ingresar nombre, apellido, edad, sexo, fecha en que desapareció y una descripción detallada de la persona, también debe ingresar nombre, teléfono y relación del contacto. Podrá ingresar si así lo desea la fotografía del desaparecido, e indicar último paradero conocido del mismo.

Listado de Desaparecidos: se despliega un listado de los desaparecidos.

Registro de Usuario: Si el usuario elige registrarse en el sitio de la catástrofe deberá ingresar nombre, apellido, *nickname*, *email*, contraseña, confirmación de la misma, fecha de nacimiento, sexo, número de celular, dirección.

Log-in: se despliega una ventana con los campos de *email* y contraseña.

Logout: botón que borra la sesión del usuario en el sistema.

Home: despliega un mapa que indica la zona afectada por la catástrofe. También se puede ver una galería de imágenes de personas desaparecidas y las noticias configuradas desde el BO para mostrar en la catástrofe.

7.4.2.3. SIC Mobile

Si el usuario no está dentro de una región de catástrofe se despliega un listado con las catástrofes activas para que el usuario acceda a alguna de ellas si así lo desea. De estar dentro de una región de catástrofe, la aplicación mostrará directamente el panel principal de la catástrofe y el botón de pánico. Las funciones que el usuario puede realizar en la aplicación son: registro de usuario, pedir ayuda mediante el botón de pánico, puede pedir ayuda para sí.

Puede visualizar las noticias seleccionadas para la catástrofe.

7.4.3. Características de los usuarios

Usuarios del BackOffice:

- Administrador: este usuario debe de ser una persona capaz de gestionar catástrofes.
- Rescatista: debe de tener un nivel alto de conocimiento del sistema y del negocio, es un perfil técnico de campo, es decir usuarios de consulta el cual utiliza la herramienta como fuente de información y guía para realizar su trabajo.

Usuarios Sitios Catástrofes:

- Visitantes: usuarios sin ningún perfil específico.
- Usuarios registrados: al igual que los visitantes no tienen un perfil específico.

Usuarios de dispositivos móviles: al igual que los usuarios de los sitios *web* d e cada catástrofe no tienen un perfil específico.

7.4.4. Modelo de dominio

En el modelo de dominio se representan de forma visual las clases conceptuales o los objetos del mundo real en el contexto del problema planteado.

Utilizando la notación *Unified Modeling Language (UML)*, el modelo de dominio se representa con un conjunto de entidades en los que no se define ninguna operación.

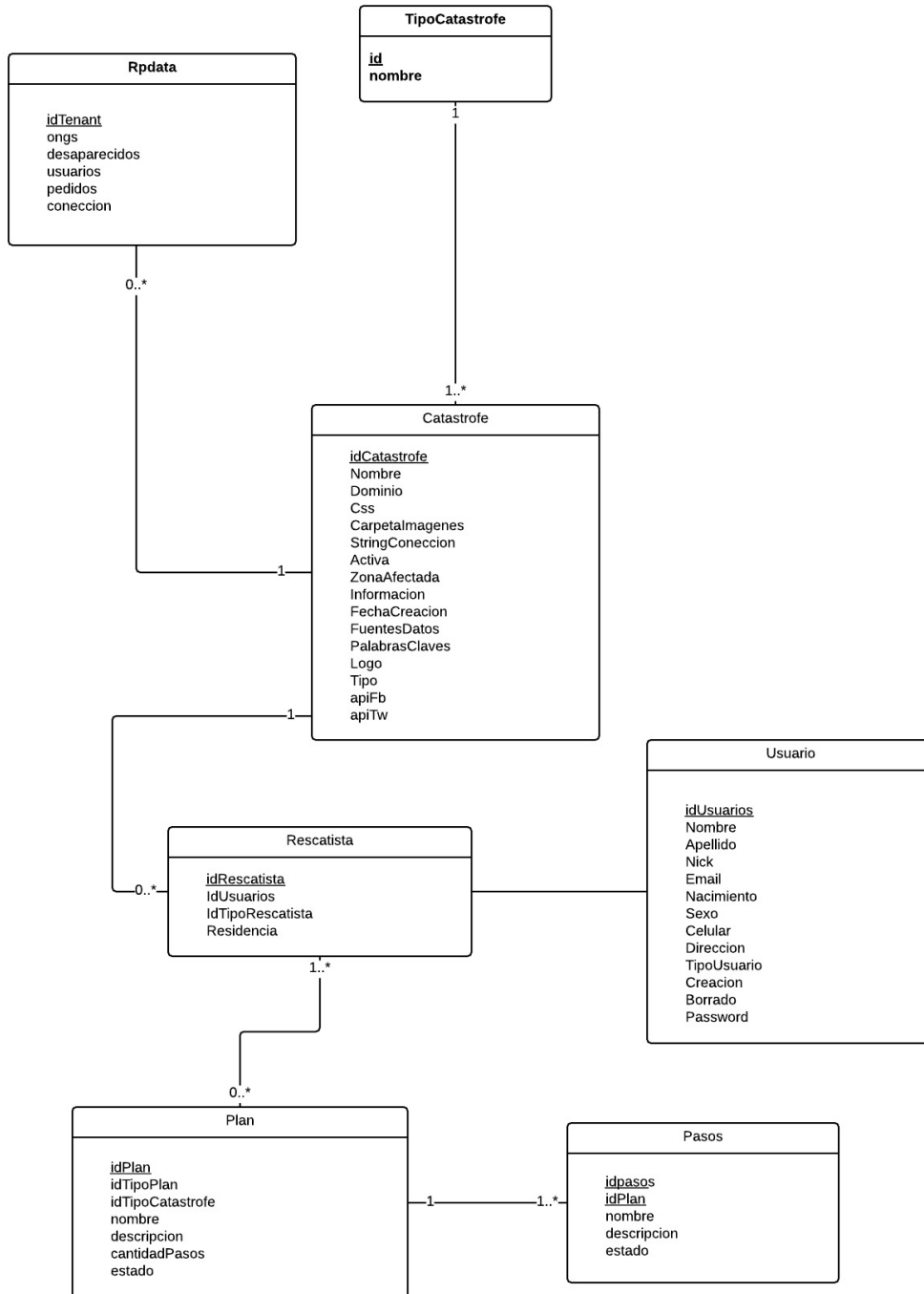


Imagen 8 - Modelo de Dominio BO

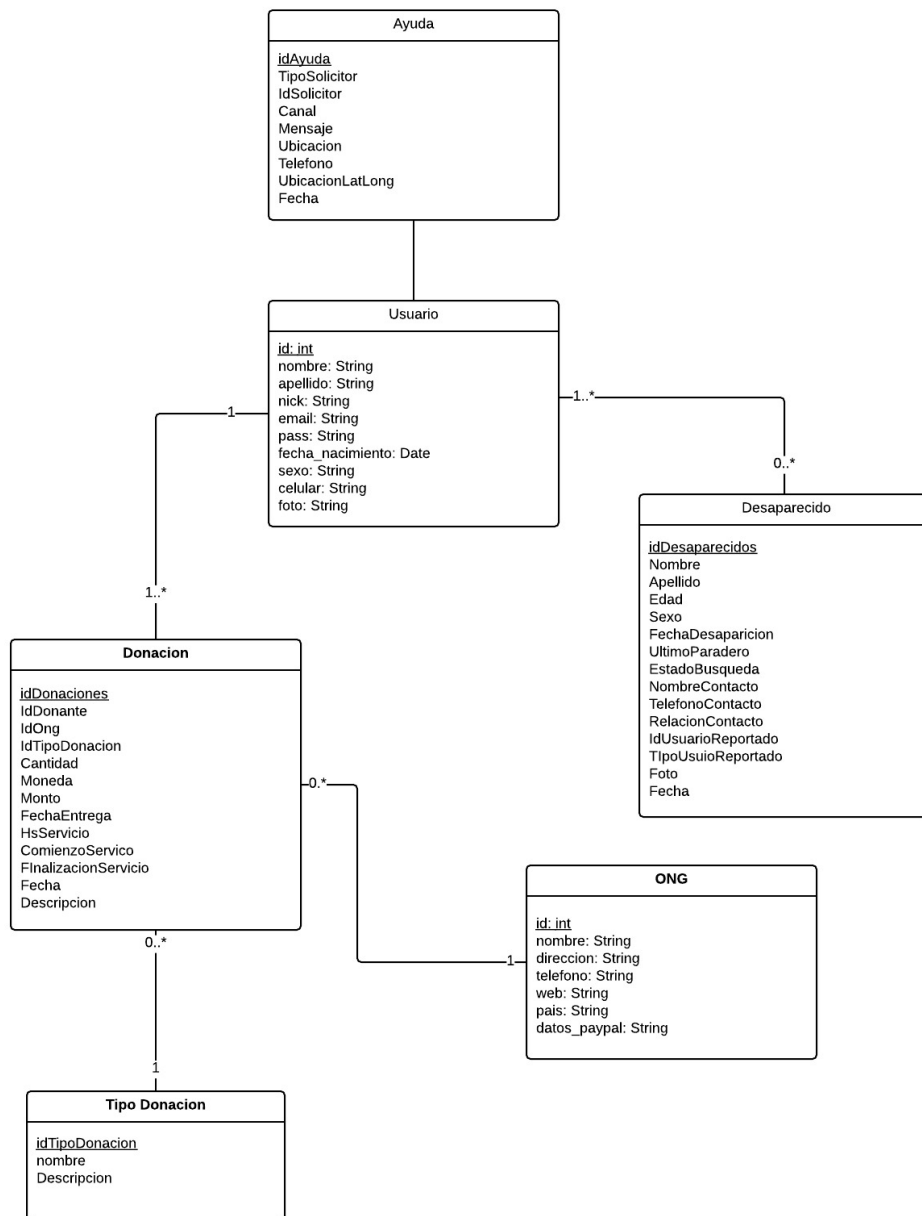


Imagen 9 - Modelo de Dominio Tenant

Para más información acerca de entidades y relaciones, referirse al anexo [C]: Modelo de Dominio.

A continuación se presentan algunos de los los diagramas que se construyeron para el diseño de la plataforma web. Para más detalles referirse al anexo [G] - Modelo de Diseño.

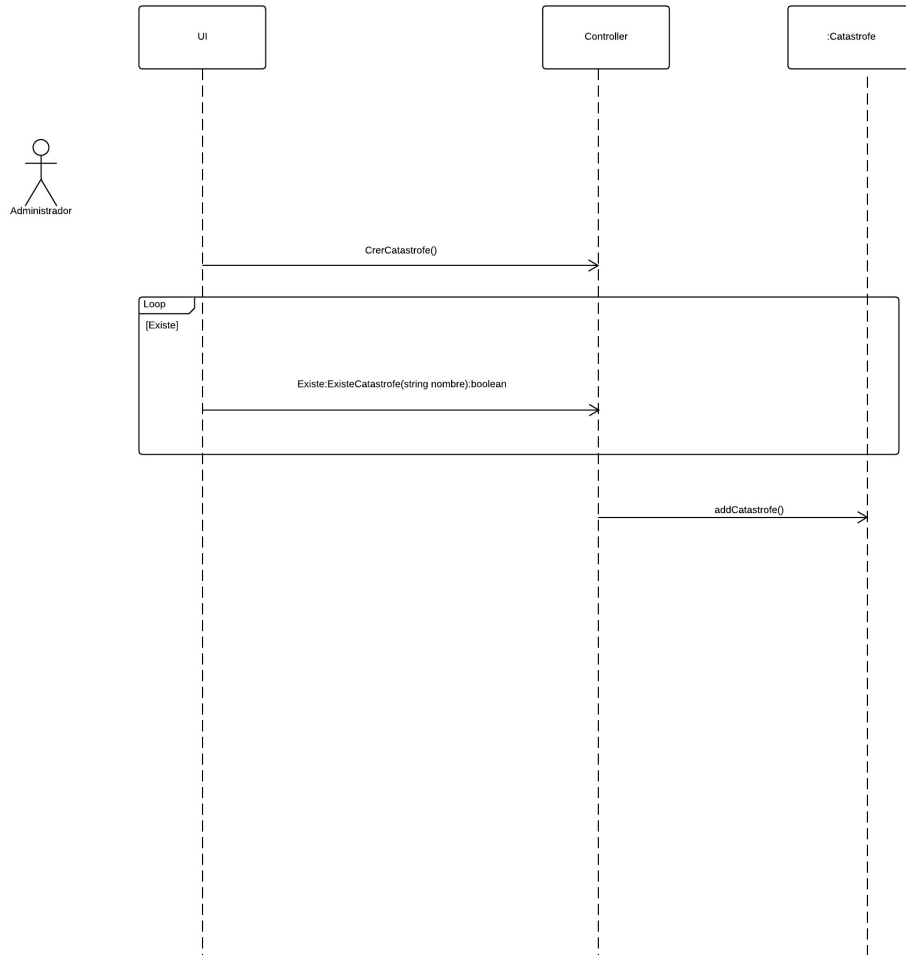


Imagen 11 – DSS – Alta de Catástrofe

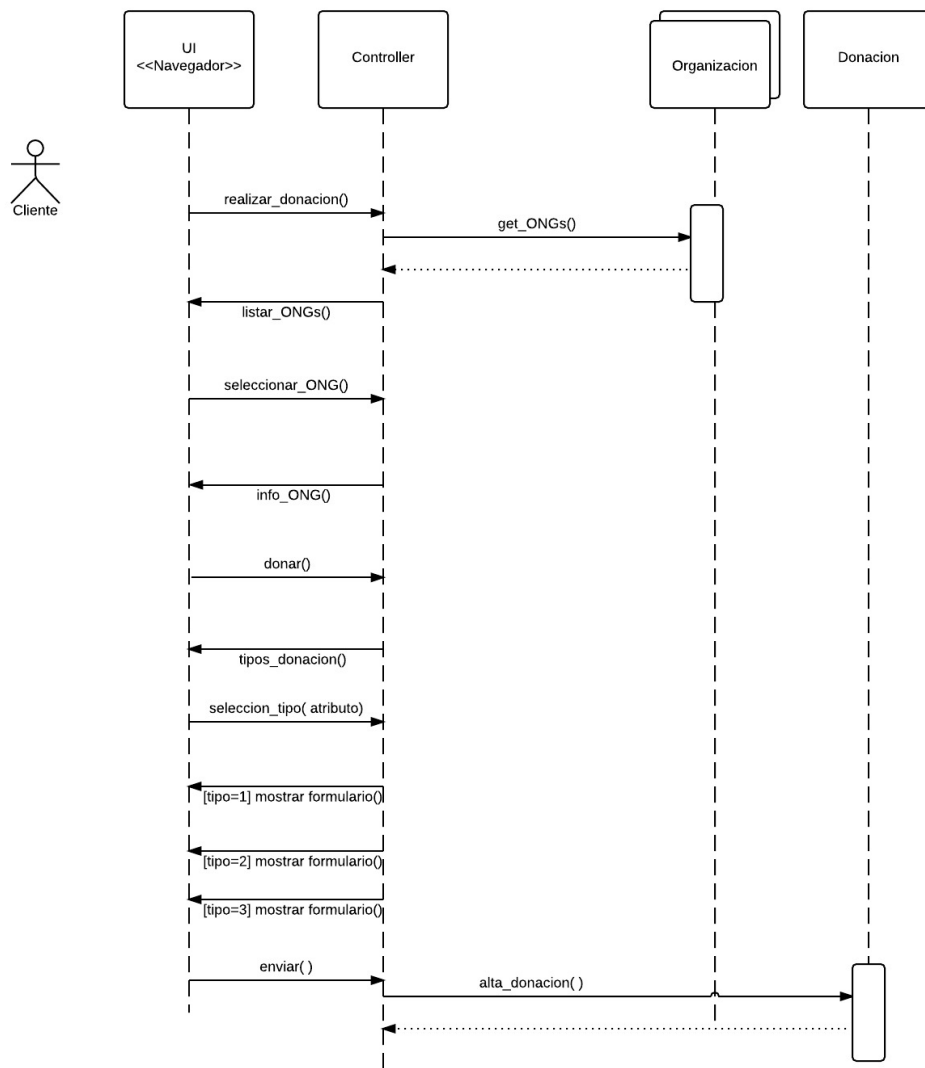


Imagen 12 – DSS – Realizar Donación

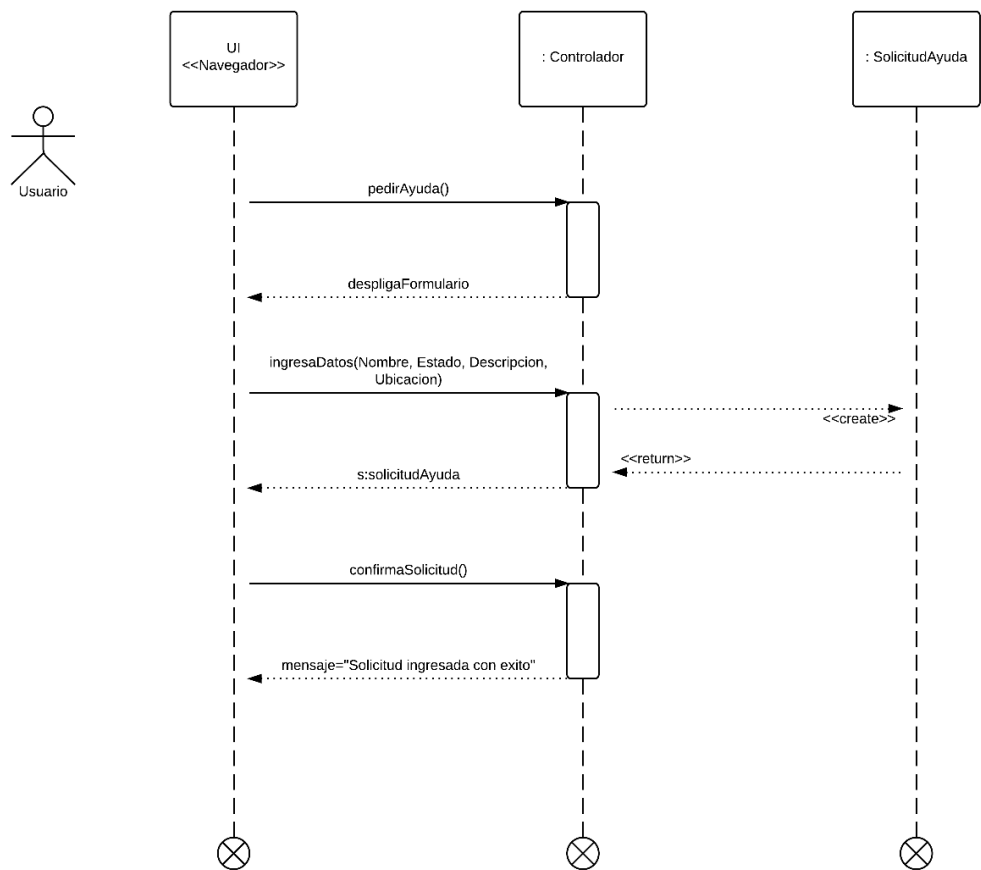


Imagen 13 – DSS – Pedir Ayuda

7.4.6. Arquitectura

Se proporciona una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

Proporciona una descripción de la distribución e interacción entre componentes del sistema. El mismo ha sido generado directamente del Análisis de los casos de uso y del Modelo de Diseño.

7.4.7. Visión general

En esta iteración el documento de arquitectura contiene información acerca de los casos de uso más relevantes a la hora de diseñar el sistema a construir, seguido de los subsistemas de la arquitectura y su distribución física al momento de su despliegue.

7.4.8. Descomposición en Subsistemas

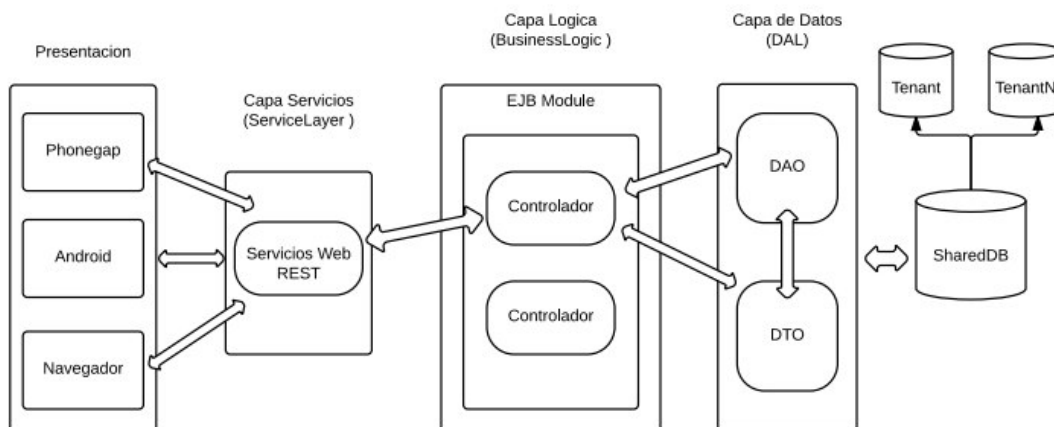


Imagen 14 - Descomposición en subsistemas

2.1.1. Capa Presentación

Tiene la interfaz de usuario. Se encarga de mostrar los componentes gráficos y de la interacción con el usuario pero no implementar ninguna parte de la lógica del negocio, la misma se consume a través de los servicios *web* expuestos por esta capa.

2.1.2. Capa Servicios

Contiene los servicios REST expuestos por la lógica de negocios. Permite la comunicación con los dispositivos móviles y el navegador, los cuales consumen de estos, permitiendo que el usuario pueda realizar los casos de uso correspondientes.

2.1.3. Capa Lógica

Este subsistema se encarga principalmente de proveer todos los datos necesarios para la capa de presentación. Tiene toda la lógica asociada a la aplicación. La cual es expuesta como servicios *web* para ser consumidos desde cualquier componente de la Capa Presentación.

2.1.4. Capa de Datos

El subsistema de base de datos tiene como objetivo guardar de forma persistente todos aquellos datos que sean necesarios y tener los mismos disponibles. Se utiliza JPA como interfaz para el acceso a datos que quiebra la dependencia de los componentes que este utiliza. Los componentes Objeto de Transferencia de Datos (DTO), se alojan en la capa de datos pero son compartidos por la capa de negocios cumpliendo la tarea de transportar información.

7.4.9. Diseño de Clases

Las clases principales son: `UsuarioController`, `CatastrofeController`, `OngController`.

UsuarioController: agrupa todos los métodos referentes al Usuario y las acciones que éste mismo puede realizar en el sistema, interactuando con otros componentes en el caso que sea necesario. Expone la interfaz `IUsuario`.

CatastrofeController: agrupa todos los métodos referentes a la Catástrofe y todas las funcionalidades que son requeridas por los diferentes procesos que ocurren en el sitio de la Catástrofe. Expone la interfaz `ICatastrofe`.

OngController: encapsula todos los métodos referentes a esta entidad y su interacción con otros componentes. Expone la interfaz `IOng`.

7.4.10. Vista del Modelo de Implementación

Subsistemas Componentes

Controlador

Pertenece a la capa de negocio, está encargado de manejar la lógica y delegar funcionalidades a otros componentes del sistema.

DAO.

Gestiona el acceso a datos en la aplicación. Tiene funciones que permiten obtener objetos para el lenguaje java a partir de la información almacenada en la base de datos.

Interfaces.

Controladores - Servidor de aplicaciones.

Se expone una interfaz para los servicios *web*. De esta manera es posible la independencia de la lógica de negocio y los consumidores de la misma.

Acceso a datos - DAO.

La base de datos tiene una interfaz de acceso a datos. La misma abstrae el componente de bases de datos y la implementación física de este.

7.4.11. Vista del Modelo de Distribución

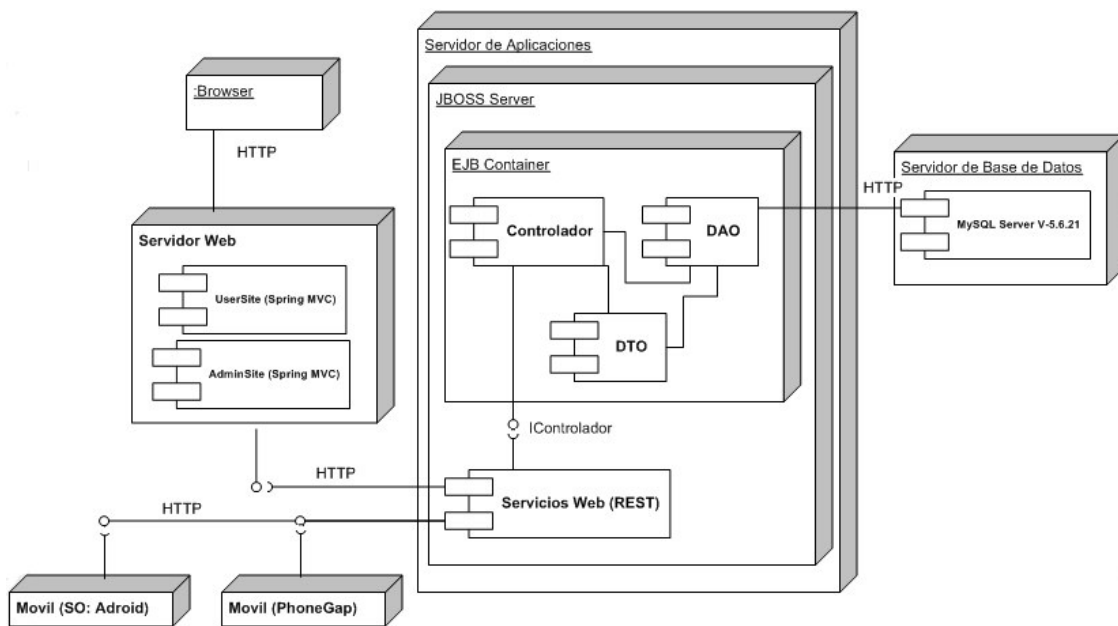


Imagen 15 - Diagrama de distribución

Nodos.

Servidor de Base de Datos (MySQL Server).

Este servidor es el encargado de alojar los datos del sistema y prestar los servicios que actúan o consultan sobre los mismos. Se accederá al mismo por medio del servidor de aplicaciones.

Servidor de aplicaciones (JBoss Server-WildFly).

El servidor de aplicaciones es el principal intermediario entre la interfaz de usuario y los datos del sistema. El mismo contiene la mayor parte de la lógica de negocio del mismo. Se utilizará JBoss como servidor de aplicaciones (contenedor de EJB).

Servidor Web.

Almacena los documentos necesarios para la interacción con el usuario cuando este lo solicita. El nodo Servidor web y Servidor de aplicaciones pueden ser alojados ambos en el mismo servidor, en este caso en el servidor de aplicaciones JBoss.

Base de Datos.

Base de Datos Compartida.

Se describe el modelo que relaciona las entidades del sistema entre sí para la Base de Datos Compartida. Los atributos no serán mostrados en el siguiente modelo para lograr una mejor visualización en el documento.

7.4.12. Modelo entidad relacional

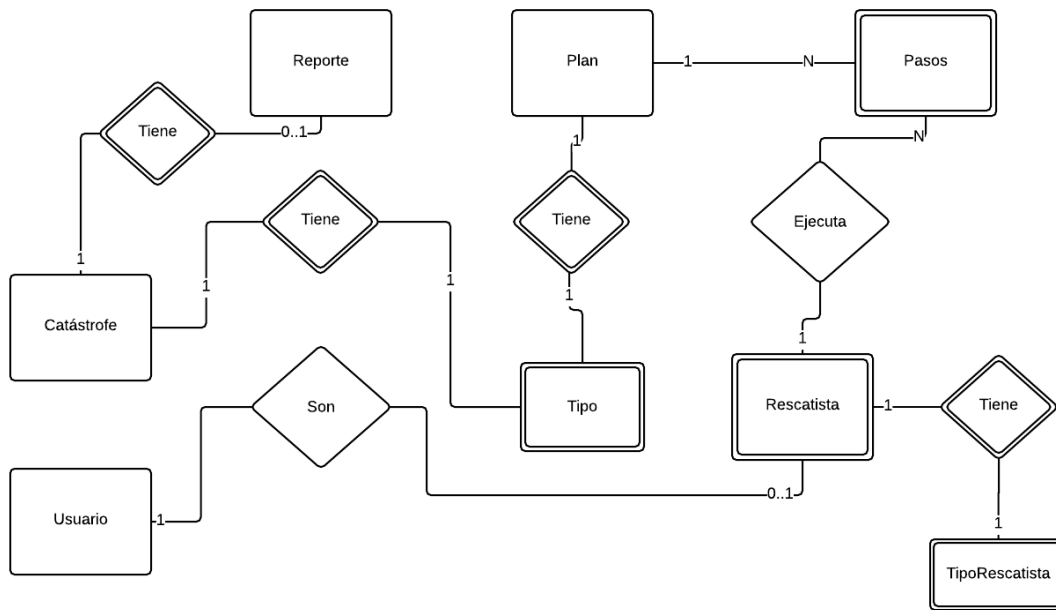


Imagen 16 - Modelo entidad-relación BO

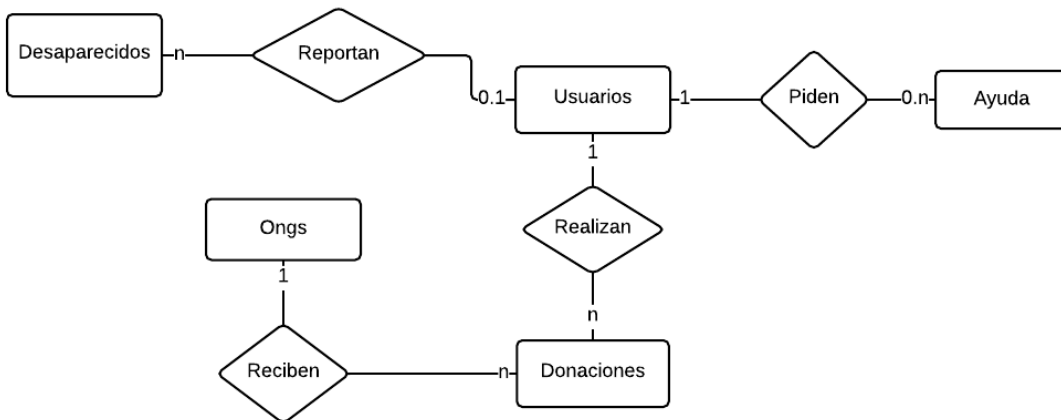


Imagen 17 - Modelo entidad-relación Tenant

7.4.13. Modelo de base de datos – Mapeo de clases a tablas

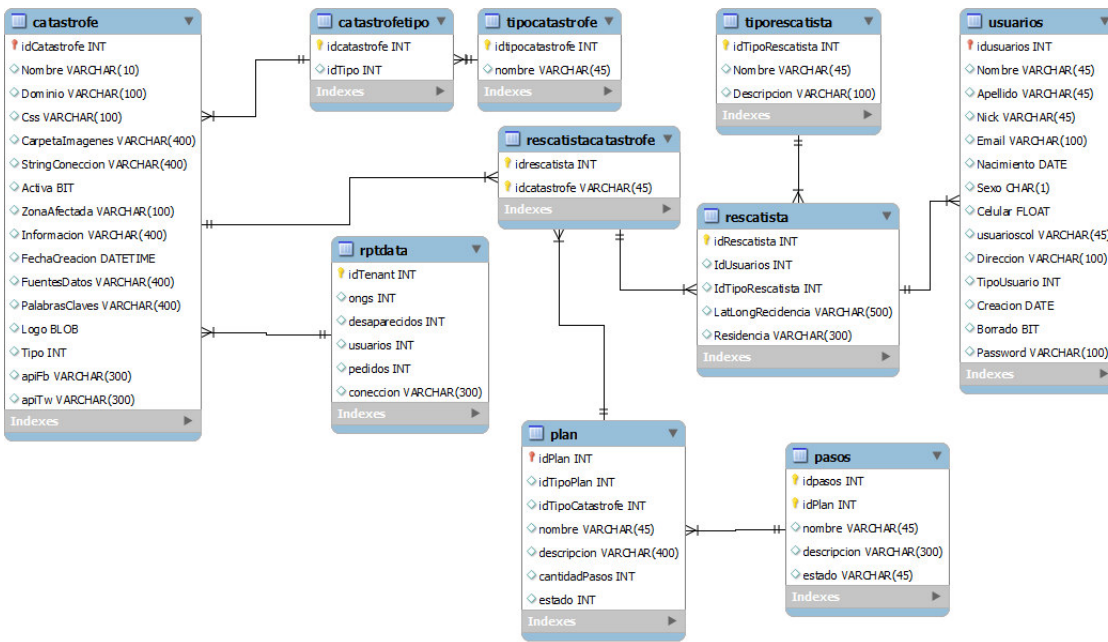


Imagen 18

- Modelo de base de datos BO

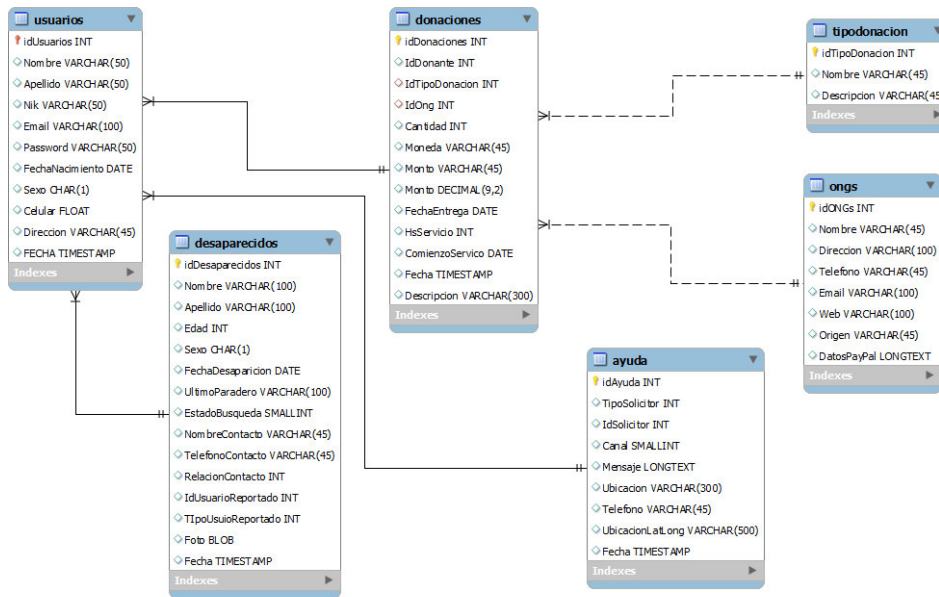


Imagen 19- Modelo de base de datos Tenant (mapeo de clases a tablas)

7.4.14. Características de almacenamiento

Dado que la aplicación es *multi-tenant* con bases de datos distribuidas una por cada *tenant*, será fácil extender la aplicación del modelo de datos para satisfacer las necesidades de los clientes individuales, y la restauración de datos de un cliente de copias de seguridad en caso de falla, es un procedimiento relativamente simple. Este enfoque proporciona aislamiento máximo de datos de clientes.

7.4.15. TECNOLOGÍAS UTILIZADAS

Para la elección de las tecnologías utilizadas, se realizó un estudio de las tecnologías disponibles y adecuadas para el desarrollo de este proyecto. Dicho estudio se ve reflejado en el Anexo [E]: Análisis de Tecnologías.

8. Gestión de Proyecto

La planificación es un proceso que comienza con una misión, metas y objetivos que deben lograrse. Con la planificación de proyectos de desarrollo de software se busca ordenar el qué hacer durante el proyecto y asignar adecuadamente los recursos y tareas para cumplir los objetivos propuestos.

8.1.1. Gestión de horas

La administración de la carga horaria es fundamental en la gestión de un proyecto.

Se dividieron las tareas y la carga horaria de acuerdo a la disponibilidad de tiempo de los integrantes.

Para la asignación de trabajos, se aplicó el criterio de dividir las tareas por sus características.

Además de la aplicación de división de tareas, se realizaron reuniones semanales para hacer una puesta a punto y transmitir los conocimientos, avances y novedades.

9. Problemas encontrados

Dificultades que experimentamos durante el desarrollo proyecto:

Generación del ambiente estable de desarrollo. Se dificultó generar un ambiente estable para el grupo desarrollara en paralelo. En un principio se decidió utilizar máquinas virtuales con sistema operativo Ubuntu. El momento de cambio se generó a partir de la lentitud con la cual funcionaban los emuladores y las eventuales caídas del sistema operativo nativo. Debido a esas dificultades, se decidió migrar el proyecto para trabajar con Sistema Operativo nativo Windows 7/8.

Migrar y configurar un ambiente estable en Windows llevó entre 8 y 12 días.

Ninguno de los integrantes del grupo es desarrollador con experiencia en java, por ende la curva de aprendizaje fue mayor. Eso generó que la etapa de desarrollo aumentara entre un 15% y 20% respecto a su carga horaria.

El agravamiento de la enfermedad de uno de nuestros integrantes provoco una mayor sobrecarga horaria (mayor a un 10%) en los demás integrantes. Debido a eso se aplazó la finalización del proyecto respecto a los plazos pautados originalmente.

Decisiones en torno a las tecnologías generaron estiramiento de los plazos pactados. Utilizar Spring como Framework aportó potencialidad al proyecto, pero impactó negativamente en la cantidad de horas de desarrollo.

10. Conclusiones

Se propuso la necesidad de desarrollar una plataforma de soporte ante catástrofes naturales, la cual tiene como finalidad centralizar toda la información relevante y de socorro en un único portal *web*, a su vez se debía desarrollar una aplicación *android* nativa brindando a los usuarios la posibilidad de pedir ayuda en caso de emergencia y una aplicación utilizando *phonegap* que permita a los rescatistas ejecutar los procedimientos en caso de emergencia.

El *software* desarrollado cumple con varias de las funcionalidades requeridas según el problema planteado.

Para llevar a cabo el mismo se investigaron y analizaron las tecnologías disponibles en el mercado, y se decidió utilizar herramientas tales como *Android Studio*, *Phonegap*, *Spring* teniendo en cuenta los posibles riesgos que esto suponía al proyecto ya que no se contaba con desarrolladores prácticos en las mismas. Sin embargo, las metas se lograron cumplir.

Al realizar el estado del arte se vio la importancia que significaba para los ciudadanos el uso de las aplicaciones para salvaguardar sus vidas, y mejorar la respuesta de los rescatistas para la gestión y recuperación ante catástrofes.

Esto también ha dejado saldo positivo ya que permitió aprender el manejo de nuevas herramientas a los integrantes del equipo.

El tiempo disponible originalmente para la realización del proyecto no ha sido suficiente en cuanto a las funcionalidades que se solicitaban desarrollar y aplicar, el producto logrado es satisfactorio y una buena base para continuar desarrollando la herramienta, luego de haber obtenido una prórroga en el tiempo de entrega.

Referente a lo académico se logró obtener gran experiencia para el desarrollo de proyectos para terceros, y cumpliendo con los requerimientos solicitados. Y ha sido muy beneficioso para los integrantes del equipo, logrando un ambiente agradable de trabajo donde se respetaron las ideas de cada integrante del equipo, y donde se explotaron las fortalezas de cada uno. Y las dificultades encontradas se enfrentaron en grupo.

Durante el desarrollo se planteó la posibilidad de aplicar la solución creada a la realidad mundial para enfrentar catástrofes, y se notó que en varias aplicaciones están involucradas varias instituciones gubernamentales. Por lo

tanto se ha llegado a la conclusión que son proyectos en general encarados por gobiernos de países con economías muy saneadas.

11. Trabajos a Futuro

Mejorar en varios ítems de la solución respecto a la seguridad de la información del sitio. Debido a la gran cantidad de información recabada de los usuarios que se registran en el sistema y a su vez realizan donaciones.

También se planteó la mejora de poder integrar facebook y twitter para publicar información a través de ellas y poder registrarse en el sitio con las mismas.

Algo importante sería poder utilizar url's seguras, *Hypertext Transfer Protocol Secure* (https).

Una gran mejora para la aplicación desarrollada creemos debería ser que la vista móvil soporte más S.O.

Otra mejora importante podría ser la integración con la herramienta Paypal para las donaciones económicas.

También se podría desplegar la aplicación en una plataforma de *Cloud Computing*.

12. Referencias

1. SINAIE [En línea] [Citado el: xx de febrero de 2015] <http://www.sinae.gub.uy/>
2. FEMA [En línea] [Citado el: xx de febrero de 2015] <http://www.fema.gov/es>
3. Google Maps [En línea] [Citado el: xx de febrero de 2015] <https://www.google.com.uy/maps/>
4. PDC Disaster Alert [En línea] [Citado el: xx de febrero de 2015] <http://www.pdc.org/solutions/tools/disaster-alert-app/>
5. PDC Disaster Aware [En línea] [Citado el: xx de febrero de 2015] <http://www.pdc.org/solutions/products/disasteraware/>
6. ubAlert [En línea] [Citado el: xx de febrero de 2015] <http://www.ubalert.com/>
7. Facebook [En línea] [Citado el: xx de febrero de 2015] <https://www.facebook.com/>
8. Twitter [En línea] [Citado el: xx de febrero de 2015] <https://twitter.com/>
9. RSOE EDIS Notifier Lite [En línea] [Citado el: xx de febrero de 2015] <http://hisz.rsoe.hu/>
10. Android [En línea] [Citado el: xx de febrero de 2015] <http://www.android.com/>
11. Google [En línea] [Citado el: xx de febrero de 2015] <https://www.google.com>
12. Linux [En línea] [Citado el: xx de febrero de 2015] <http://www.linux.org/>
13. Samsung [En línea] [Citado el: xx de febrero de 2015] <http://www.samsung.com/latin/home>
14. Sony [En línea] [Citado el: xx de febrero de 2015] <http://www.sony.com/>
15. LG [En línea] [Citado el: xx de febrero de 2015] <http://www.lg.com/es/moviles>
16. HTC [En línea] [Citado el: xx de febrero de 2015] <http://www.htc.com/latam/>
17. YouTube [En línea] [Citado el: xx de febrero de 2015] <https://www.youtube.com/>
18. Gmail [En línea] [Citado el: xx de febrero de 2015] <https://www.gmail.com>
19. Google Play [En línea] [Citado el: xx de febrero de 2015] <https://play.google.com/>
20. Android Studio [En línea] [Citado el: xx de febrero de 2015] <https://developer.android.com/sdk/installing/studio.html>
21. Eclipse [En línea] [Citado el: xx de febrero de 2015] <https://www.eclipse.org/>
22. Phonegap [En línea] [Citado el: xx de febrero de 2015] <http://www.codalot.com/blog/programacion-de-apps-moviles-con-html5-css-y-javascript-apache-cordova-y-phonegap/>
23. Blackberry 10 [En línea] [Citado el: xx de febrero de 2015] <http://us.blackberry.com/software/smartphones/blackberry-10-os.html#tab-1>

24. Firefox OS [En línea] [Citado el: xx de febrero de 2015] <https://www.mozilla.org/es-ES/firefox/os/>
25. IOS [En línea] [Citado el: xx de febrero de 2015] <https://www.apple.com/es/ios/>
26. Ubuntu [En línea] [Citado el: xx de febrero de 2015] <http://www.ubuntu.com/>
27. Windows Phone 7/8 [En línea] [Citado el: xx de febrero de 2015] <http://www.windowsphone.com/es-es>
28. Windows 8 [En línea] [Citado el: xx de febrero de 2015] <http://windows.microsoft.com/es-es/windows-8/meet>
29. Tizen [En línea] [Citado el: xx de febrero de 2015] <https://www.tizen.org/es>
30. Javascript [En línea] [Citado el: xx de febrero de 2015] <http://es.wikipedia.org/wiki/JavaScript>
31. HTML [En línea] [Citado el: xx de febrero de 2015] <http://www.w3.org/html/wg/drafts/html/master/>
32. Phonegap Build [En línea] [Citado el: xx de febrero de 2015] <https://build.phonegap.com/>
33. CNN [En línea] [Citado el: xx de febrero de 2015] <http://edition.cnn.com/>
34. BBC [En línea] [Citado el: xx de febrero de 2015] <http://www.bbc.com/>
35. Java [En línea] [Citado el: xx de febrero de 2015] <https://www.java.com/es/>
36. Android Studio [En línea] [Citado el: xx de febrero de 2015] <https://developer.android.com/sdk/installing/studio.html>
37. PayPal [En línea] [Citado el: xx de febrero de 2015] <https://www.paypal.com/uy/webapps/mpp/home>
38. JBoss [En línea] [Citado el: xx de febrero de 2015] <http://www.jboss.org/>
39. Spring [En línea] [Citado el: xx de febrero de 2015] <http://spring.io/>
40. Google Feed Api [En línea] [Citado el: xx de febrero de 2015]
41. MySQL [En línea] [Citado el: xx de febrero de 2015] <http://www.mysql.com/>

13. Glosario

Framework

La palabra "framework" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

IDE

Sigla en inglés de *integrated development environment*, un entorno de desarrollo integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación. Es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

API

API (del [inglés](#) *Application Programming Interface*) es el conjunto de [funciones y procedimientos](#) (o [métodos](#), en la [programación orientada a objetos](#)) que ofrece cierta [biblioteca](#) para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

API Java

Como el lenguaje Java es un Lenguaje Orientado a Objetos, la API de Java provee de un conjunto de clases utilitarias para efectuar toda clase de tareas necesarias dentro de un programa. La API Java está organizada en paquetes lógicos, donde cada paquete contiene un conjunto de clases relacionadas semánticamente.

API Phonegap

Las APIs que maneja PhoneGap ofrecen: acceso a varias funcionalidades del dispositivo móvil como ser: el acelerómetro, a la cámara, a aplicaciones de capturas de audio y video; permite trabajar con las conexiones de red que cuenta, contactos almacenados, obtener datos relevantes del mismo, maneja eventos, se pueden trabajar con archivos binarios, se puede obtener la posición geográfica, permite utilizar notificaciones, acceso al vibrador y facilita el uso de base de datos.

API REST

Java API for RESTful *Web Services* es una API del lenguaje de programación Java que proporciona soporte en la creación de servicios web de acuerdo con el estilo arquitectónico *Representational State Transfer* (REST)

API JPA

Java *Persistence* API, más conocida por sus siglas JPA, es la API de persistencia desarrollada para la plataforma Java EE.

SIC

Sigla de Sistema Internacional de Catástrofes. Esta sigla fue la que adoptó el grupo para el nombre del proyecto.

Web Service

Un servicio *web* (en inglés, *Web Service*) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

ABM

Cuando utilizamos esta sigla, nos referimos al conjunto de 3 casos de uso básicos que se van a implementar: altas, bajas y modificación.

SINAE

El SINAE (Sistema Nacional de Emergencia), es el organismo estatal que se encarga de la gestión de desastres en Uruguay.

FEMA

FEMA (*Federal Emergency Management Agency*) cuyo cometido principal es apoyar a todos los ciudadanos y a las agencias de primera respuesta y garantizar que EEUU trabaje en conjunto para desarrollar, mantener y mejorar la capacidad de preparación, protección, respuesta, recuperación y mitigación ante los peligros.

UIT

La Unión Internacional de Telecomunicaciones es una agencia especializada del sistema de Naciones Unidas que tiene su sede en Ginebra.

ONG

Sigla de organización no gubernamental, institución sin ánimo de lucro que no depende del gobierno y realiza actividades de interés social.

Servidor de aplicaciones (JBOSS)

Es un [servidor](#) en una [red de computadores](#) que ejecuta ciertas [aplicaciones](#). Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

PDC

Pacific Disaster Center (PDC) es un centro de ciencia, la información y la tecnología aplicada, que trabaja para reducir los riesgos de desastres y los impactos sobre la vida, la propiedad y las economías de todo el mundo.

RSOE EDIS

Asociación Nacional Húngara, *Radio Distress-Signalling and Infocommunications* (RSOE), opera las emergencias a través de *Emergency and Disaster Information Service* (EDIS).

OGC

Open Geospatial Consortium. Su fin es la definición de estándares abiertos e interoperables dentro de los [Sistemas de Información Geográfica](#) y de la [World Wide Web](#).

WMS

El servicio *Web Map Service* (WMS) definido por el OGC ([Open Geospatial Consortium](#)) produce [mapas](#) de datos referenciados espacialmente, de forma dinámica a partir de [información geográfica](#).

WFS

Web Feature Service del Consorcio [Open Geospatial Consortium](#) es un servicio estándar, que ofrece una [interfaz](#) de comunicación que permite interactuar con los mapas servidos por el estándar [WMS](#)

UIT

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas para las tecnologías de la información y la comunicación – TIC

MVC

El modelo–vista–controlador (MVC) es un patrón de [arquitectura de software](#) que separa los [datos](#) y la [lógica de negocio](#) de una aplicación de la [interfaz de usuario](#) y el módulo encargado de gestionar los eventos y las comunicaciones.

JSF

Java Server Faces (JSF) es una tecnología y framework para aplicaciones [Java](#) basadas en web que simplifica el desarrollo de [interfaces de usuario](#) en aplicaciones [Java EE](#).

Vaadin

Vaadin es un *framework* para el desarrollo de aplicaciones *web Open Source* modernas o también llamadas [RIA](#) (*Rich Internet Applications*) que posee una librería

con numerosos componentes de usuario con los que construir aplicaciones *web* profesionales similares a los interfaces de una aplicación de escritorio

J2EE (JAVA EE)

Java Platform, Enterprise Edition es una plataforma de programación (parte de la [Plataforma Java](#)) para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación [Java](#).

Tecnologías J2EE

JSP's, Servlets, Javabeans, EJB, JTS, JTA, RMI, CORBA, Java Mail, JMS, JNDI

LGPL

[Licencia Pública General de GNU](#) (GPL de GNU)

GOOGLE

Google [Inc.](#) es una [empresa multinacional estadounidense](#) especializada en productos y servicios relacionados con [Internet](#), software, dispositivos electrónicos y otras tecnologías.

EMAIL / YOUTUBE / GOOGLE TALK / GOOGLE MAPS / GOOGLE WEB TOOLKIT

Productos de Google

LINUX

Linux es un sistema operativo libre, basado en Unix.

DALVIK

Dalvik es la [máquina virtual](#) que utiliza la plataforma para dispositivos móviles [Android](#). Dalvik ha sido diseñada por [Dan Bornstein](#) con contribuciones de otros ingenieros de [Google](#).

HTML

HyperText Markup Language (Lenguaje de Marcas de Hipertexto). Es el lenguaje de marcado predominante para la construcción de páginas *web*. Este lenguaje es utilizado para describir la estructura y el contenido en forma de texto, así para complementar dicho texto con objetos tipo imágenes.

Tiles

Tiles o plantillas que podemos usar para crear Look and Feel comunes entre varias vistas dentro de una aplicación web.

Velocity

Velocity es un proyecto de Apache para el desarrollo de Templates para la capa de vista dentro del modelo MVC.

Hashtag

Es una etiqueta de metadatos precedida de un carácter especial (#) con el fin de que tanto el sistema como el usuario la identifiquen de forma rápida.

Núcleo Linux.

El núcleo de linux se puede definir como el corazón de este sistema operativo. Es el encargado de que el software y el hardware del computador puedan trabajar juntos.

LED

Un led (light-emitting diode: 'diodo emisor de luz') es un componente opto electrónico pasivo y, más concretamente, un diodo que emite luz.

Mercator

La proyección de Mercator es un tipo de proyección cartográfica cilíndrica, ideada por Gerardus Mercator en 1569, para elaborar mapas de la superficie terrestre.

IE 10

Windows Internet Explorer 10 (abreviado como IE10)

Mozilla Firefox

Mozilla Firefox es un navegador web libre y de código abierto desarrollado para Microsoft Windows, Mac OS X y GNU/Linux coordinado por la Corporación Mozilla y la Fundación Mozilla.

Google Chrome

Google Chrome es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto.

XML

Por extensión (X) HTML. El W3C es el encargado de formular la especificación de las Hojas de Estilo en Cascada.

Javascript

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos basado en prototipos, imperativo, débilmente tipado y dinámico.

Nube

La computación en la nube conocida también como servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, es un paradigma que permite ofrecer servicios de computación a través de Internet.

Hardware

El término hardware se refiere a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

Software

Se conoce como software al soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

JDBC

Java Database Connectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java

RMI

RMI (Java Remote Method Invocation) es un mecanismo ofrecido por Java para invocar un método de manera remota.

JMS

La API Java Message Service (en español servicio de mensajes Java), también conocida por sus siglas JMS, es la solución creada por Sun Microsystems para el uso de colas de mensajes.

Enterprise JavaBeans

Los Enterprise JavaBeans (también conocidos por sus siglas EJB) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE.

Servlets

El servlet es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.

Portlets

Los portlets son componentes modulares de las interfaces de usuario gestionadas y visualizadas en un portal web.

JavaServer Pages

Java *Server Pages* (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas *web* dinámicas basadas en HTML, XML, entre otros tipos de documentos.

HTML5

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la *World Wide Web*, HTML.

JSON

JSON, acrónimo de JavaScript *Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre un único socket TCP.

HOST

El término host ("anfitrión", en español) es usado en informática para referirse a las computadoras conectadas a una red, que proveen y utilizan servicios de ella.

Anexos

1. Requerimientos de Software
2. Modelo de Dominio
3. Modelo de Casos de Uso
4. Análisis de Tecnología
5. Documento de Visión
6. Descripción de la Arquitectura
7. Modelo de Diseño
8. Modelo de Datos
9. Entrevista SINAE

ANEXO [A]

Proyecto: SIC

Especificación de Requerimientos de Software para SIC, SICMobile

Versión 2.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
-------	---------	-------------	-------

17/10/2014	1.1	Versión inicial de los requerimientos.	Yennifer Herrera
22/10/2014	1.2	Ajustes de los requerimientos funcionales.	Yennifer Herrera
27/10/2014	1.3	Adición de interfaces de usuarios	Yennifer Herrera
07/02/2015	2.0	Actualización	Yennifer Herrera
08/02/2015	3.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

1.	Introducción.....	3
1.1.	Propósito.....	3
1.2.	Alcance.....	3
1.3.	Definiciones, siglas y abreviaturas.....	4
1.4.	Referencias.....	4
2.	Descripción general.....	4
2.1.	Perspectiva del producto.....	4
2.1.1.	Interfases de usuario.....	4
2.1.2.	Interfaces con hardware.....	8
2.1.3.	Interfaces con software.....	8
2.2.	Funciones del producto.....	9
2.2.1	BackOffice.....	9
2.2.2	Sitio Catástrofe.....	12
2.2.3	SICMobile.....	14
2.2.4	SICMobile Phongap.....	14
2.3.	Características de los usuarios.....	15
2.4.	Restricciones de diseño.....	16
2.5.	Supuestos y dependencias.....	16

- **Introducción**

Se define la especificación de requerimientos funcionales y arquitectónicos y de ambiente para el funcionamiento de SIC y SICMobile.

13.1. Propósito

Este documento refleja el funcionamiento y el alcance detallado de SIC y SICMobile. El cual sirve como herramienta para los desarrolladores a la hora de implementar el mismo, acotándose así a lo descrito en este documento.

13.2. Alcance

SIC es un sistema web compuesto por un BackOffice y un cliente multitenant el BackOffice será utilizado exclusivamente por los administradores y o rescatistas en el cual ellos podrán crear instancias de sitios clientes para cada catástrofe identificada, además de administrar y gestionar datos como Usuarios, ONGs, Rescatistas, Riesgos y Planes de emergencia, con los cuales maneja un workflow de acciones a tomar para cada catástrofe basado en los riesgos y planes generados, este workflow no solo se actualiza desde el BO sino que también desde el dispositivo móvil en PhoneGap para los rescatistas.

A partir de estas catástrofes el BO debe poder acceder a reportes varios como serian la navegabilidad de cada cliente tenant, personas desaparecidas, Donaciones realizadas y pedidos de ayuda.

El BO debe brindar una vista de catástrofes activas, en la cual se puedan visualizar e identificar en un mapa las regiones afectadas. A partir de este mapa se debe brindar la opción de visualizar exclusivamente una zona seleccionada y apreciar en ella los diferentes pedidos de ayuda discriminados por estados, accediendo mediante este a los datos específicos del pedido.

La versión publica de SIC el cliente tenant que es generado para cada catástrofe debe brindar la posibilidad de registrarse mediante Facebook, y twitter además de permitir el clásico registro completando los datos en el sitio, este usuario debe tener la posibilidad de acceder a diversas fuentes de información embebidas en SIC, reportar y buscar desaparecidos, Pedir ayuda y realizar donaciones a las diferentes ONGs, cada cliente SIC debe permitir visitantes (usuarios no registrados) los cuales pueden acceder a todas las funcionalidades del mismo a excepción de reportar desaparecidos y realizar donaciones.

En cada cliente se debe mostrarse el mapa con la región afectada.

SICMobile Es una versión genérica y reducida del SIC Web, este debe indicarle al usuario que sus datos serán obtenidos por la aplicación para el registro en el sistema al momento de instalación si no los consigue del teléfono debe pedírselo manualmente.

SICMobile debe identificar la ubicación del usuario y si está dentro de la región de alguna de las catástrofes debe centrarse en ella pero no limitarlo a acceder a las otras si así lo desea

Por lo tanto debe brindar un listado de catástrofes activas, a partir del cual el usuario podrá acceder a cada una de ellas.

SICMobile debe permitir solicitar ayuda mediante un formulario y un botón de pánico ambos toman la ubicación del usuario para realizar el llamado. Este último solo será accesible si el usuario se encuentra en una región de catástrofe, además de esto debe mantener informado al usuario de los desaparecidos reportados y las nuevas noticias referentes a la o las catástrofes.

13.3. Definiciones, siglas y abreviaturas.

BO - BackOffice

SIC - Sistema Internacional de Catástrofes

API - Application Programming Interface

MySQL - motor de base de datos

DAO - Data Access Object

IE10 - Internet Explorer 10 (navegador)

LG Nexus 4 E960 - Dispositivo Mobile

LAN - Local Area Network

13.4. Referencias

Arquitectura del Sistema

Modelo de Diseño

Modelo de Datos

Glosario

- **Descripción general**

SIC debe estar optimizado para los navegadores IE10 en adelante, Mozilla Firefox y Chrome.

SICMobile debe estar optimizado para los dispositivos LG Nexus4 E960.

13.5. Perspectiva del producto

- **Interfaces de usuario**

A modo ilustrativo se presentan algunas interfaces críticas para el sistema. Para tomar de referencia al momento de desarrollar las funcionalidades.

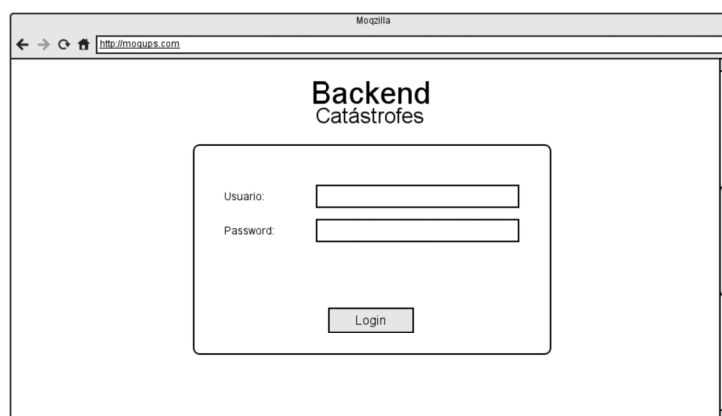


Ilustración 1 - Login BO



Ilustración 2 - Panel de Control BackOffice

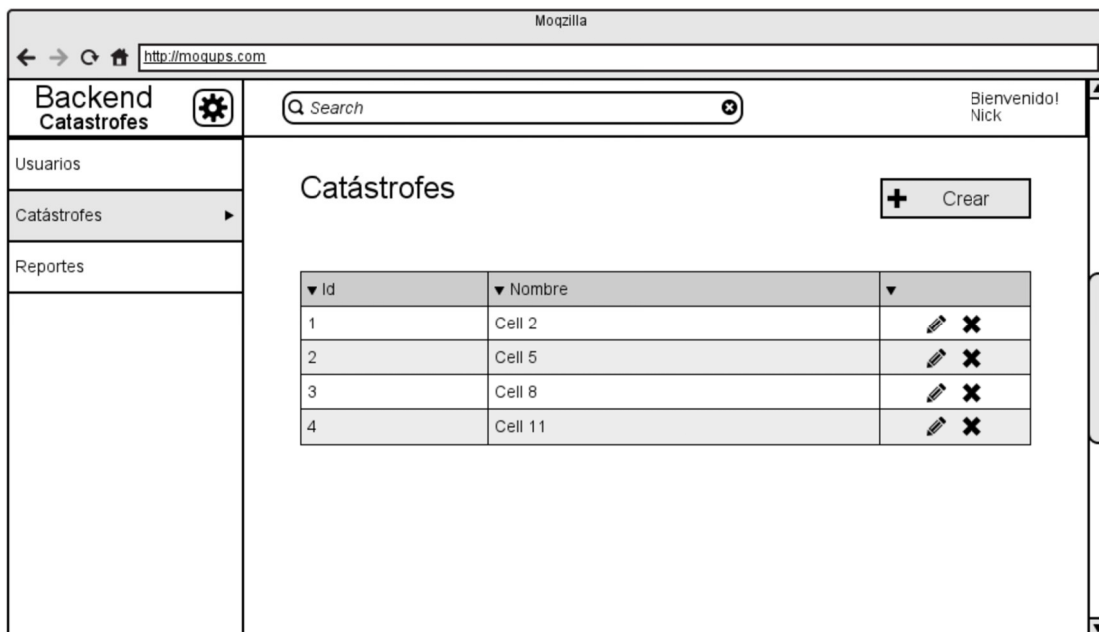


Ilustración 3 - Panel de Catástrofes

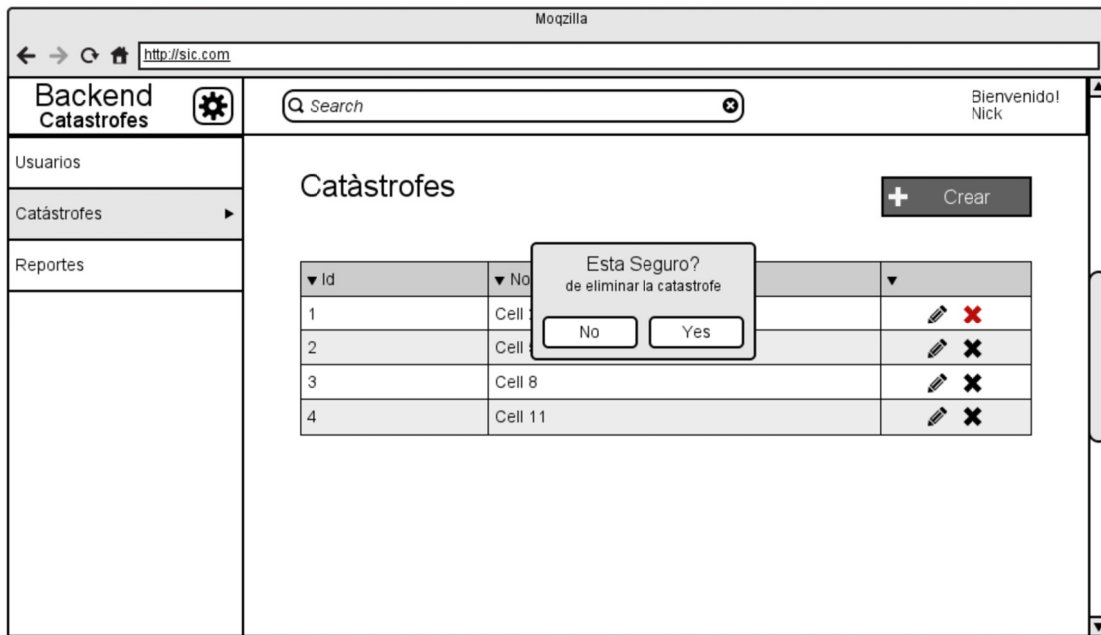


Ilustración 4 - Eliminar Catástrofe

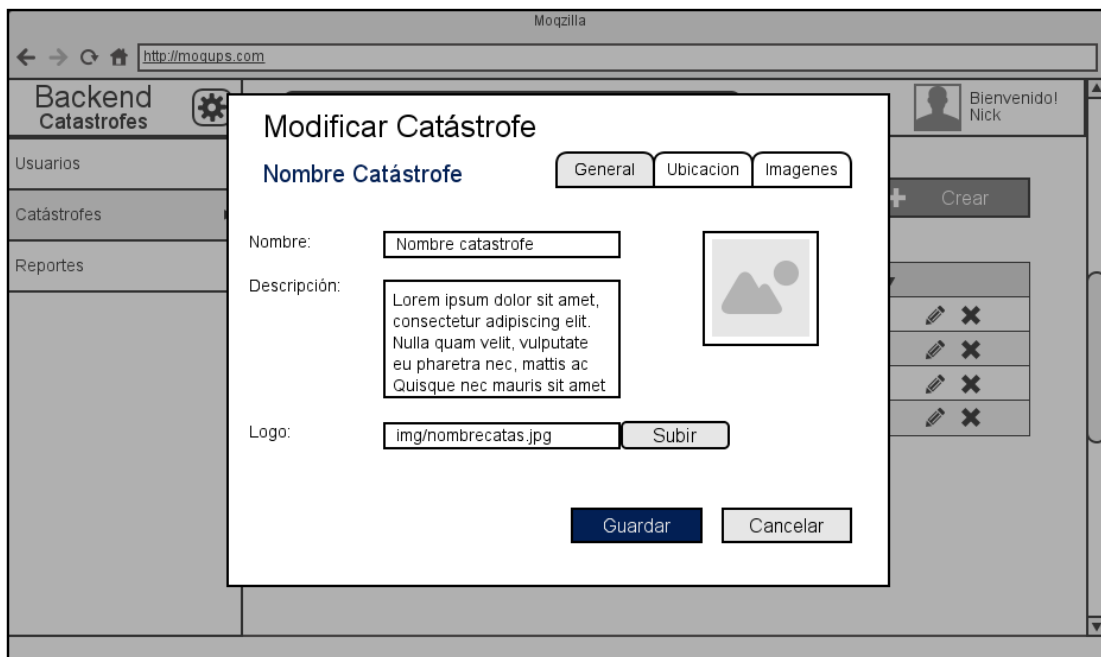


Ilustración 5 - Modificar Catástrofe

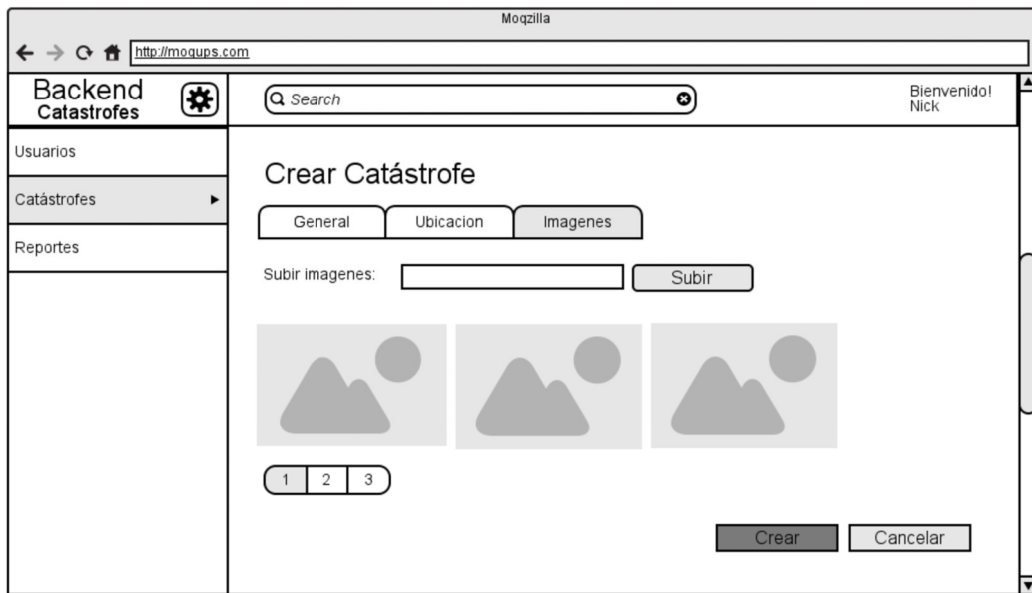
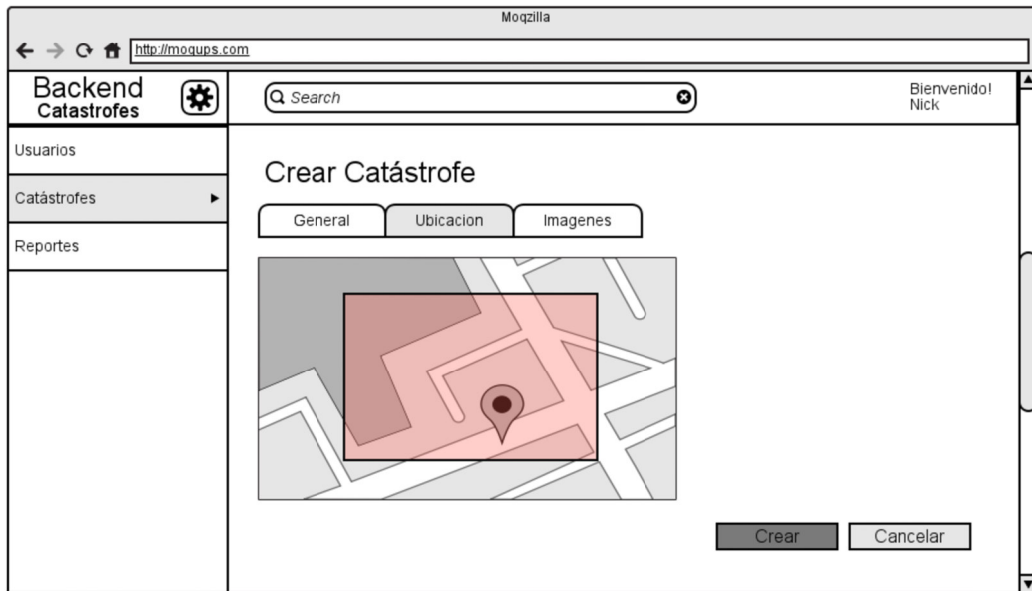
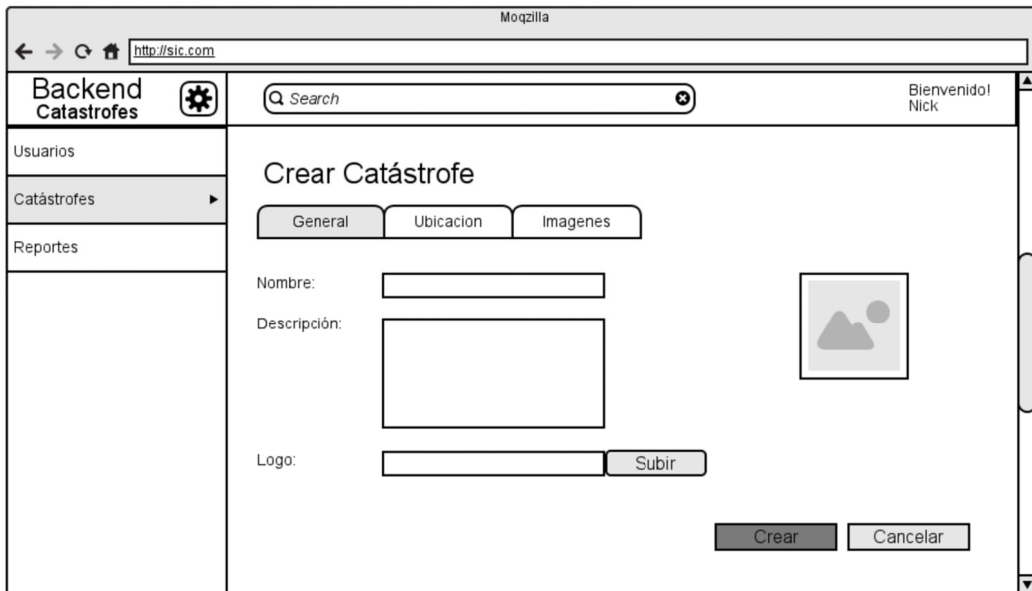


Ilustración 6 - Crear Catástrofe



Ilustración 7 - Botón de pánico Pedido de ayuda Mobile

▪ Interfaces con hardware

SIC y SICMobile se comunican a una capa de servicios mediante protocolo http, esta capa de servicio es la encargada de comunicarse con la lógica la cual accede directamente a los DAOS (capa de datos) que van contra la base de datos en MySQL mediante mysql-connector-java.

Las bases de datos no tienen por qué estar dentro del mismo servidor que la aplicación pero es deseable que estén dentro de la misma red LAN.

El Back Office del Aplicativo se comunica directamente a la capa lógica evitando así el pasaje por los servicios.

▪ Interfaces con software

Tanto SIC Web como SICMobile interactúan con ciertas APIs de redes sociales para el registro de usuario.

SIC Web implementa las APIs de, Twitter y Facebook. Y SICMobile se limita a implementar la API de Facebook.

Además de consumir diversos RSS como por ejemplo los de CNN o BBC con lo cual mantener los sitios bien informados para esto se utilizara Google Ajax Feed Api.

Para la gestión de mapas y regiones se utilizara Geo Developers at Google API, google analytics para manejar el acceso y uso del cliente en la aplicación, REST API de Paypal para el pago de donaciones económicas mediante tarjeta de crédito o cuenta PayPal

13.6. Funciones del producto

2.2.1 BackOffice

Alta de Usuario

Al crear un usuario se debe poder ingresar los siguientes datos como requeridos nombre, apellido, email, password y celular y tipo de usuario.

El email debe de validar su correcto formato y debe ser único en el sistema, la password debe de mostrarse codificada además de viajar al servidor de manera encriptada

El tipo de usuario debe de desplegarse en un combo Box con las opciones de seleccione cuando no se ha elegido nada, Administrador y Rescatista.

Además de estos campos requeridos la pantalla debe permitir ingresar Fecha de nacimiento, sexo (radiobuttons con las opciones de Femenino y Masculino) y dirección.

Cuando se selecciona tipo de usuario Rescatista, el sistema debe permitir ingresar los datos pertinentes al rescatista como lo son

El tipo de rescatista representado con un combo con las opciones correspondientes,

La residencia del mismo y desplegar un mapa donde se pueda geo localizar esa residencia para poder persistir la latitud y longitud de la misma.

Esta alta impacta sobre las tablas Usuarios y rescatista si así corresponde.

Modificación de Usuario

Se debe poder visualizar todos los datos, de estos se podrán modificar todos a excepción del email y el tipo de usuario

Si el tipo de usuario es rescatista, los datos pertinentes del rescatista se deben de poder visualizarse pero solo se debe poder editar el lugar de residencia.

Baja y Listado de Usuario

Se debe de presentar una vista donde se listen los usuarios del backend del sistema mediante esta vista se debe de permitir la opción de crear uno nuevo y visualizar una grilla con los datos más pertinentes de los usuarios como lo serían por ejemplo Nombre, Apellido, Email, celular y Tipo de usuario esta grilla debe de contener la opción de editar los usuarios listados y eliminarlos, cuando se seleccione la opción de eliminar alguno de esos usuarios el sistema debe pedir confirmación de la acción la baja de estos usuarios debe ser lógica.

Inicio de sesión

Se debe controlar el acceso al sistema. Dependiendo del tipo de usuario que inicie sesión serán las funcionalidades que este podrá realizar.

El sistema debe de solicitar el mail y la contraseña del usuario, y controlar que estos coincidan entre sí con un usuario activo.

Si es necesario que la contraseña del usuario viaje entre capas esta debe viajar de manera encriptada.

Alta ONG

En el formulario de alta de ONG se deben desplegar los siguientes datos a completar, Nombre de la ONG, dirección, teléfono, mail, web, Origen, Datos PayPal, todos los campos a excepción de los datos Paypal, web y origen son requeridos.

Modificación ONG

Todos los datos de la misa se deben de poder modificar.

Baja ONG

La baja de esta es física dado que la relación con las catástrofes no se pierde ya que es una tabla no relacionada que se copia en cada catástrofe.

Listado ONG

Se debe presentar un listado con todas las Ongs registradas en el Back Office además de la posibilidad de poder modificar o borrar una ya existente así como acceder desde este listado al formulario de alta.

Alta Gestión de Riesgo

Se debe presentar un formulario en el cual el usuario pueda ingresar los siguientes campos requeridos

Nombre del posible riesgo, descripción, debe poder seleccionar de un combo box el tipo de riesgo que está definiendo, subir un documento del tipo word, o pdf en el cual tenga el detalle de la gestión del riesgo, y desplegar un mapa donde el usuario pueda marcar la zona afectada por el posible incidente y seleccionar de una lista el plan de contingencia relacionado.

Además opcionalmente el usuario debe poder ingresar, la ciudad, el departamento, la fecha de ocurrencia (periodo día/mes), número de ocurrencias, número de afectados, pérdidas monetarias y costos de la contención del riesgo

Listado Gestión de riesgo

Se debe presentar un listado con todos los riesgos creados, además de esta misma vista se debe brindar la posibilidad modificar y acceder al formulario de alta

Datos a mostrar en el listado, nombre, tipo, fecha de ocurrencia departamento o ciudad.

Alta Plan de Emergencia

En la pantalla de alta del plan se espera que el administrador pueda ingresar

El nombre del plan, los costos que este conlleva (monetarios) dar una descripción general y subir el archivo detallado del plan.

A su vez este debe poder ingresar los pasos requeridos para ejecutar el plan

Se debe de presentar en la misma pantalla una grilla donde se vayan cargando los pasos, en la cual se debe poder ingresar por cada paso el ID del mismo, el nombre, la descripción, el tipo de ejecutor (tipo de rescatista) y cantidad de ejecutores de este tipo

ID	Nombre	Descripción	Tipo	cantidad
1	Evacuación	Despejar el área afectada	Bomberos	5

Listado Plan de Emergencia

Se debe presentar un listado con todos los Planes de emergencia, además de esta misma vista se debe brindar la posibilidad modificar y acceder al formulario de alta

Datos a mostrar en el listado, nombre, costos, fecha de creación

NOTA: Todos los datos de la gestión de riesgos y del plan de emergencia son editables. Pero no se debe permitir borrar ningún riesgo ni ningún plan.

Configuración de Workflows

Para cada catástrofe se puede configurar uno o varios flujos de trabajo

El formulario para generar estos flujos de trabajo debe de contar simplemente con tres campos requeridos donde se indique el Plan al que hace referencia (idPlan), La catástrofe (IdCatastrofe) el responsable del workflow (idUsuarioBO). Además de un campo de comentarios opcional.

Simplemente indicando estos datos el sistema por detrás debe crear los pasos del workflow en base a los pasos del plan de emergencia al que se vinculó.

Inbox Workflows

Esta pantalla debe presentarse a modo de wizard en la cual primero nos presente los Workflows activos donde podemos filtrar por catástrofe.

En esa pantalla podemos seleccionar el WF deseado y como siguiente paso nos debe presentar el estado de los pasos del mismo, que se ejecutó, quien y cuando se ejecutó, en que paso esta y además podemos accionar sobre el paso actual es decir podemos cambiar el estado del paso actual asignando un ejecutor y pasando al paso siguiente. Si se desea el usuario puede dar el WF como terminado sin haberse culminado el flujo.

Reportes varios

El sistema debe brindar tres tipos de reportes por Catástrofe

Para esto debe generar una pantalla estilo wizard, donde el usuario pueda seleccionar de un listado la catástrofe de la cual quiera consultar información

Al dar siguiente se debe presentar tres pestañas con los tipos de reportes.

Donaciones en el tiempo, Solicitud de ayuda en el tiempo y uso del sitio web.

Para estos tres tipos se debe desplegar un listado con la información pertinente y la opción de exportar la información en xml, csv o pdf (a definir.)

Además de una gráfica en el tiempo que represente la data desplegada.

2.2.2 Sitio Catástrofe

Pedido de ayuda Mobile

El dispositivo móvil debe de tener una especie de botón de pánico en el cual el usuario con solo pulsarlo envíe una alerta con sus coordenadas al sitio de la catástrofe. La aplicación deberá saber a qué catástrofe corresponde el pedido de ayuda en base a las coordenadas de la ubicación de la víctima.

Además si así lo desea podrá acceder a un formulario de ayuda que se comportara de la misma manera que el botón de pánico pero con la simple diferencia que permite al usuario ingresar la descripción que crea pertinente.

El usuario al momento de instalar la aplicación está informado que sus datos como numero de teléfono ubicación y demás son retenidos y utilizados por la aplicación.

Pedido de ayuda web

En la pantalla de pedido de ayuda web se deben desplegar los campos de nombre, teléfono, mensaje un combo donde indica si la ayuda es para sí mismo o para otra persona, si el usuario esta deslogueado este combo queda automáticamente seleccionado en anónimo y no se puede editar.

Esta función manda mail a la organización de SIC y además al rescatista asignado en el momento a la catástrofe (ver gestión de workflow y pasos.)

Realizar Donación

La funcionalidad de realizar una donación es exclusiva de la web, las donaciones pueden ser de tres tipos económica, de bienes o de servicios, en el formulario el usuario debe indicar su nombre y número de teléfono, además del tipo de donación que desea realizar, si la donación es económica se despliegan los datos para que el usuario ingrese el monto y la moneda en la que realizará la donación y a continuación haga login en el servicio Paypal para efectuar el pago de la misma.

Si la donación es de bienes se despliegan los siguiente campos para que el usuario complete, descripción del bien que desea donar, cantidad y fecha de entrega.

Si la donación es de servicio se debe indicar de que tipo, la cantidad de horas del mismo y la fecha de inicio y finalización del mismo. Además de un campo de descripción general que será obligatorio completar.

En todos estos casos se envía un mail de notificación a la ONG a la cual se realizó la donación con la información pertinente para que se pongan en contacto con el donante si es necesario.

Reportar Desaparecido

El formulario para reportar desaparecidos debe de contar con los siguientes datos requeridos, Nombre, Apellido, Edad, Sexo, Fecha en la que desapareció,

descripción detallada, Nombre y teléfono de contacto para la comunicación en caso de obtener noticias, relación con el contacto.

Además opcionalmente debe poder ingresar la foto del desaparecido
Indicar último paradero del mismo donde debe escribir zona o dirección en donde se lo vio.

Por detrás el sistema debe registrar el estado de la búsqueda como 1-Buscando, indicar el tipo de usuario que dio de alta el desaparecido en este caso es 2-FO y como esta funcionalidad solo la pueden realizar usuarios registrados se debe mantener la relación con el usuario logueado (IdUsuarioReportado).

Listado de Desaparecidos

El sitio debe de brindar un acceso al listado de los desaparecidos, es deseable que se muestren las miniaturas de las imágenes de los mismos con una breve reseña de los datos como puede ser Nombre, Apellido, Edad y tel. de contacto. Donde al hacer click sobre uno de estos abra un modal con la información detallada que se ingresó al darle de alta a este desaparecido.

En esta pantalla debe haber filtros por nombre ya que se deben de listar todos ellos tal vez el usuario busque a uno específico.

Registro de Cliente

En el formulario de registro de usuario se deben desplegar en primera instancia las opciones de registrarse mediante las cuentas de Facebook, Twitter, o la propia del sistema.

Si el usuario elige alguna de las opciones externas con solo confirmar el registro se toman todos los datos pertinentes y se establece el login en el sistema automáticamente.

En cambio si el usuario elige registrarse mediante el sitio de la catástrofe deberá completar el formulario de registro de usuario, el cual contiene los siguientes campos, nombre, apellido, Nick email, contraseña, confirmación de contraseña, fecha de nacimiento, sexo, celular y dirección.

En esta funcionalidad todos los campos a excepción de la fecha de nacimiento son requeridos.

Login

Se debe desplegar un popup con los campos de email y contraseña y un botón de login además de los botones con la opción de loguearse mediante Facebook o twitter, si el usuario desea completar los datos de mail y contraseña es porque el usuario se registró mediante el registro propio de la aplicación, de lo contrario al hacer click en alguna de las redes sociales se debe desplegar el popup de inicio de sesión de la api de la red seleccionada.

Logout

Simplemente es un botón que borra la sesión del usuario en el sistema.

Home

El home debe de ser una fuente de información para el cliente
Este debe contar con un mapa que indique la zona afectada de la catástrofe.

También se debe visualizar un slider de imágenes de personas reportadas como desaparecidas, que aún no hayan sido encontradas.

Además de toda información configurada desde el BackOffice para mostrar en la catástrofe, videos de YouTube y datos consumidos de la CNN, o cualquier otra fuente de noticia.

2.2.3 SICMobile

Como indicamos anteriormente la versión Mobile debe de contar con dos paneles dependiendo de si el dispositivo se encuentra dentro de una región de catástrofe o no, si el usuario no está dentro de una región de catástrofe se debe desplegar un listado con las catástrofes activas para que el usuario si así lo desea acceda a alguna de ellas.

Si el usuario si se encuentra dentro de una región de catástrofe, la aplicación debe mostrarse directamente el panel principal de la catástrofe que contiene información reducida del sitio y el botón de pánico.

El usuario siempre tiene la posibilidad de re dirigirse al panel general.

Si el usuario se encuentra en una zona de catástrofe el botón de pánico siempre debe de estar visible ya sea que el usuario este navegando dentro de la página de la catástrofe o en el panel principal.

Las funciones que el usuario puede realizar desde Mobile son, registrarse al igual que en el sitio manualmente o a partir de alguna red social, pedir ayuda rápida mediante el botón de pánico, o pedir ayuda detallada donde se presenta un formulario en el cual el usuario puede pedir ayuda para sí mismo o para otra persona. Consultar información de ONGs, Desaparecidos, o de alguna fuente de información configurada.

2.2.4 SICMobile PhoneGap

Este sistema brinda al rescatista, acceso a parte de la gestión del workflow. Mediante el dispositivo el rescatista puede acceder a los pasos pendientes en el flujo de trabajo al cual fue asignado para una catástrofe y dar como terminado uno o varios de los pasos en él.

Además puede acceder a la listado de desaparecidos de esa catástrofe e indicar alguno de estos como encontrado.

13.7. Características de los usuarios

Usuarios del BackOffice

- Administrador: este usuario debe de ser una persona capaz de gestionar las catástrofes y los rescatistas, debe de tener un nivel alto de conocimiento del sistema y del negocio.
- Rescatista: son perfiles técnicos de campo, es decir usuarios de consulta el cual utiliza la herramienta como fuente de información y guía para realizar su trabajo.

Usuarios Sitios Catástrofe

- Visitantes: usuarios sin ningún perfil específico, este usuario puede tener cualquier nivel educativo, social y cultural.
- Usuarios registrados: al igual que los visitantes no tienen un perfil específico.

Usuarios Mobile

- Al igual que los usuarios de los sitios web de cada catástrofe no tienen un perfil específico.

Usuarios PhoneGap

- Son rescatistas (perfil nombrado en la descripción de usuarios del BackOffice.)

13.8. Restricciones de diseño

Dado a los requerimientos solicitados, se determina el desarrollo del sistema con las siguientes tecnologías, la presentación de SIC ya sea el cliente tenant como e BackOffice debe estar implementada con MVC Spring, SICMobile debe implementar PhoneGap y Android nativo.

El servidor de aplicaciones debe implementar JBOSS y exponer servicios REST, y las bases de datos todas ellas corren sobre el motor de MySQL

13.9. Supuestos y dependencias

Cualquier inconveniente con las APIs externas respecto a la implementación o integración con la tecnología puede llegar a generar un cambio de restricción de diseño y o talvez de alcance.

ANEXO [B]

Proyecto: SIC

Modelo de Dominio

Versión 3.0

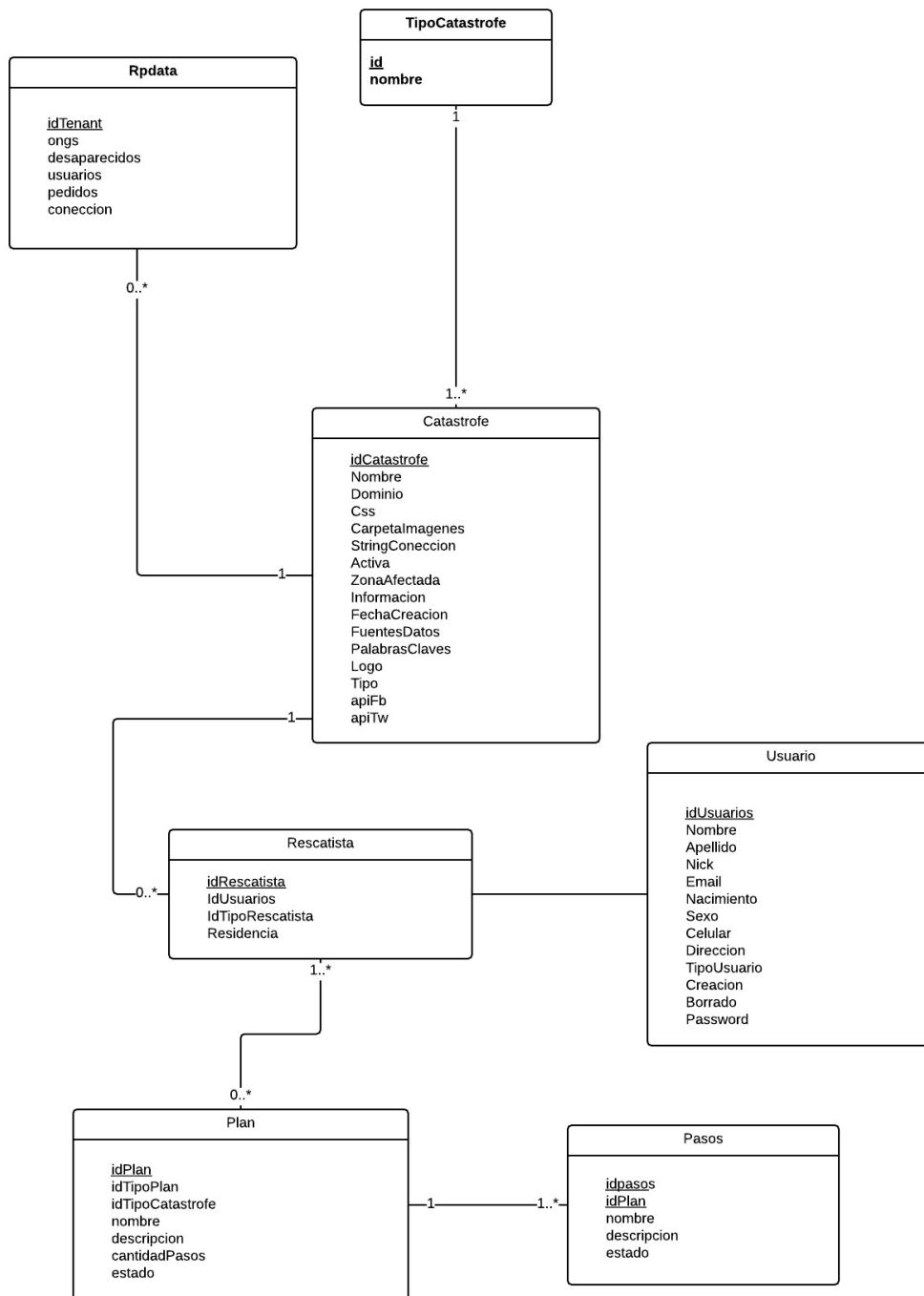
Historia de revisiones

Fecha	Versión	Descripción	Autor
25/10/2014	1.0	Se agrega diagrama y observaciones	Sandino Fratti
05/02/2015	2.0	Actualización del Documento	Juan Ignacio Castro
08/02/2015	3.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

1. Modelo de Dominio – Back Office.....	3
1.1. Observaciones.....	3
1.2. Conceptos.....	3
2. Modelo de Dominio - Tenant.....	5
2.1. Observaciones.....	5

• **Modelo de Dominio – Back Office**



13.10. Observaciones

Unicidad

- Un usuario se identifica por su correo electrónico, a pesar de tener un id de sistema que será único también.
- Una catástrofe se identifica por su dominio, a pesar de tener un id de sistema que será único también.
- Los demás conceptos modelados que sean únicos se identificarán por su id correspondiente.

13.11. Conceptos

Nombre	Catástrofe
Descripción	Sitio que se creara bajo un dominio único, para una catástrofe particular donde se encontrará toda la información sobre dicha catástrofe.

Nombre	Tipo de Catástrofe
Descripción	Detalla un tipo de catástrofe específica

Nombre	Usuario
Descripción	Todos las personas que acceden a la aplicación clientes, personas que acceden a un sitio específico, administradores, personas que manejan la aplicación y rescatistas, personas que acceden a sección específica de la aplicación diseñada especialmente para ellos.

Nombre	Rpdata
Descripción	Representa datos necesarios para la generación de los reportes.

Nombre	Rescatista
Descripción	Representa un usuario de la catástrofe que ejecuta planes.

Nombre	Plan
Descripción	Ligado a una gestión de riesgo, representa un conjunto de pasos a seguir y un orden específico.

Nombre	Pasos
Descripción	Representa una etapa específica de

	un plan (una tarea)
--	---------------------

- **Modelo de Dominio - Tenant**

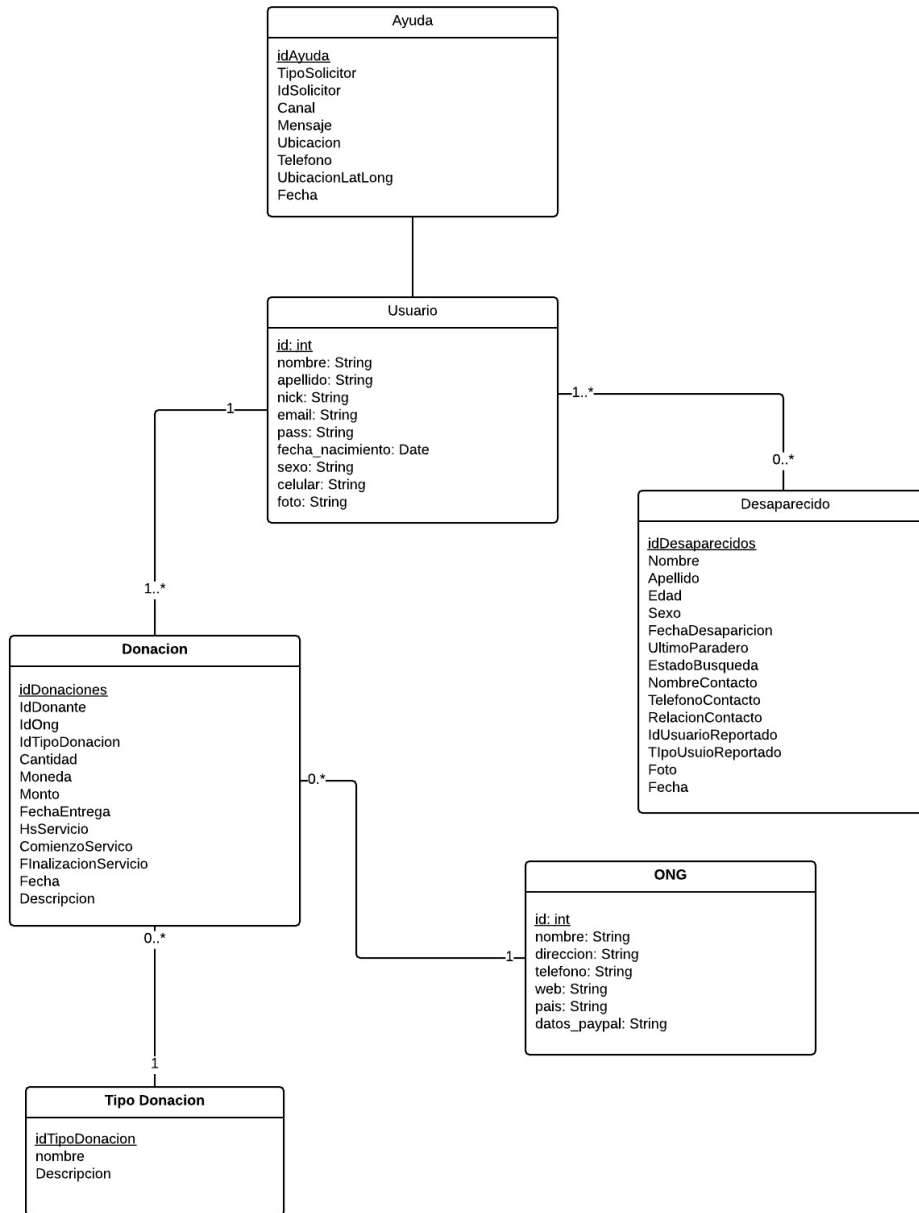


Imagen 1 – Modelo de Dominio - Tenant

13.12. Observaciones

Unicidad

- Un usuario se identifica por su correo electrónico, a pesar de tener un id de sistema que será único también.
- Los demás conceptos modelados que sean únicos se identificarán por su id correspondiente.

13.13. Conceptos

Nombre	Usuario
Descripción	Todos las personas que acceden a la aplicación clientes, personas que acceden a sección específica de la aplicación diseñada especialmente para ellos.

Nombre	ONG
Descripción	Entidad ligada a la catástrofe que recibe donaciones de parte de los clientes.

Nombre	Donación
Descripción	Representa la transferencia de servicios, dinero, etc. de un usuario a una ONG.

Nombre	Ayuda
Descripción	Representa una operación que realiza un cliente sobre la catástrofe, bajo ciertas circunstancias.

Nombre	Tipo de Donación
Descripción	Detalla un tipo de donación específica

Nombre	Desaparecido
Descripción	Persona que ha sido reportada como desaparecida en la catástrofe.

ANEXO [C]

Proyecto: S.I.C

Modelo de Casos de Uso

Versión 2.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
18/10/2014	1.0	Versión original	Horacio Garcia
21/10/2014	1.1	Modificación	Gastón Baldenegro
09/02/2015	2.0	Ajustes generales	Yennifer Herrera
09/02/2015	3.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

Historia de revisiones.....	1
Contenido.....	2
1. Actores.....	4
1.1. Usuario.....	4
1.2. Administrador.....	4
1.3. Visitante.....	4
1.4. Rescatista.....	4
1.5. Sistemas de Tarjetas.....	4
1.6. Sistema PayPal.....	4
2. Casos de Uso.....	5
2.1. Diagramas de Casos De Uso.....	5
2.2. Crear catástrofe.....	7
2.2.1. Descripción.....	7
2.2.2. Pre-condiciones.....	7
2.2.3. Flujo de eventos principal.....	7
2.2.4. Flujos de eventos alternativos.....	7
2.2.5. Post-condiciones.....	7
2.3. Realizar Donación.....	8
2.3.1. Descripción.....	8
2.3.2. Pre-condiciones.....	8
2.3.3. Flujo de eventos principal.....	8
2.3.4. Flujos de eventos alternativos.....	8
2.3.5. Post-condiciones.....	8
2.4. Solicitud de ayuda.....	9
2.4.1. Descripción.....	9
2.4.2. Pre-condiciones.....	9
2.4.3. Flujo de eventos principal.....	9
2.4.4. Flujos de eventos alternativos.....	9
2.4.5. Post-condiciones.....	9
2.5. Botón de pánico.....	10
2.5.1. Descripción.....	10
2.5.2. Pre-condiciones.....	10
2.5.3. Flujo de eventos principal.....	10
2.5.4. Flujos de eventos alternativos.....	10
2.5.5. Post-condiciones.....	10
3. Diagramas de Interaccion.....	11
3.1. Crear catástrofe.....	11
3.2. Realizar Donación.....	12
3.3. Solicitud de ayuda.....	13
3.4. Botón de pánico.....	14

- **Actores**

- **Usuario.**

Persona que ingresa al sitio (Frontoffice) para interactuar con el sitio, realizar una donación, buscar o registrar personas desaparecidas, etc.

- **Administrador.**

Persona que ingresa al sitio (BackOffice) para administrar los sitios de las catástrofes.

- **Visitante.**

Persona que se encuentra en la web, y no inició sesión, este usuario es quien va a realizar una solicitud de ayuda en caso de emergencia.

- **Rescatista.**

Es el usuario de la aplicación que ejecutará las tareas pertinentes a la atención de las catástrofes y recuperación de las mismas.

- **Sistemas de Tarjetas.**

Sistema externo de pago a través de tarjetas de crédito y débito.

- **Sistema PayPal.**

Sistema externo de pago a través de PayPal.

- **Casos de Uso**

- **Diagramas de Casos De Uso**

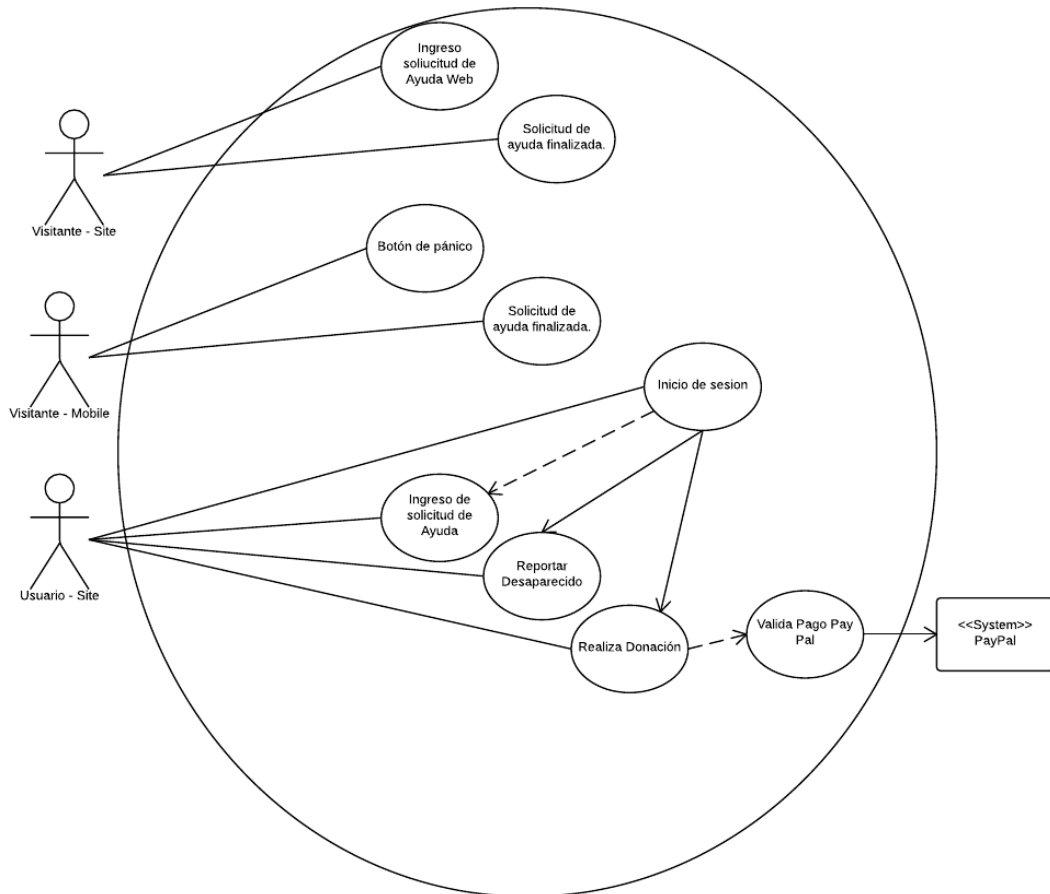


Ilustración 8 - Sitio Catástrofe

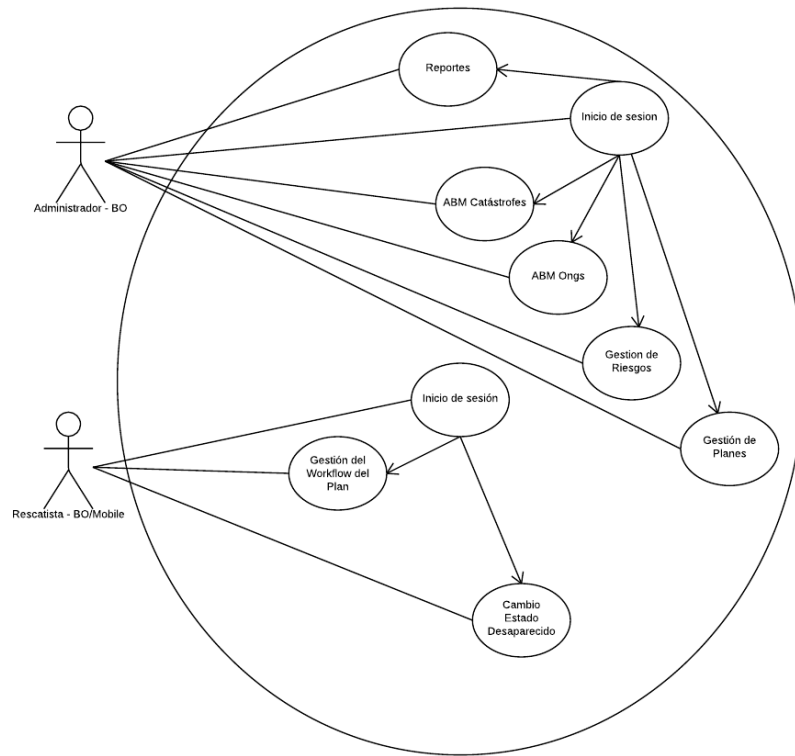


Ilustración 9 -BackOffice

- **Crear Catástrofe**

- **Descripción**

Crea una nueva catástrofe y guarda en la base de datos para que quede accesible en el sistema. El administrador del sistema ingresa los datos de la nueva catástrofe a crear, nombre de la catástrofe, ubicación, logo, descripción, imágenes, etc.

- **Pre-condiciones**

10. La catástrofe no debe existir en el sistema.

- **Flujo de eventos principal**

- El administrador selecciona crear nueva catástrofe (click)
- El sistema despliega formulario sobre nueva catástrofe
- El administrador ingresa el nombre de la catástrofe
- El sistema verifica que no exista ya la catástrofe y habilita el ingreso de los datos básicos de la catástrofe (nombre not null)
- El administrador ingresa datos básicos (ubicación, descripción)
- El sistema valida datos (not null: Descripción, Ubicación)
- El administrador confirma creación de la nueva catástrofe
- El sistema crea la nueva catástrofe, mensaje: "catástrofe creada con éxito"
- Fin CU

- **Flujos de eventos alternativos**

Flujo alternativo 1

- 1A. El nombre de la catástrofe ya existe en la base de datos
El sistema despliega un mensaje de error.
Vuelve al paso 1 del flujo normal.

- **Post-condiciones**

- 14. Queda registrada la nueva catástrofe.

- **Realizar Donación**

- **Descripción**

- Permite a los usuarios realizar donaciones a una ONG registrada en el sitio de la catástrofe.

- **Pre-condiciones**

- 11. Debe existir la catástrofe.
- 12. La catástrofe debe tener asociadas al menos una ONG.
- 13. En el caso del usuario, debe haber iniciado sesión previamente en el sitio de la catástrofe.

- **Flujo de eventos principal**

- El usuario Accede a la sección de Ongs.
- El sistema lista las ONGs asociada a la catástrofe.
- El usuario selecciona donar a una ONG
- El sistema despliega un modal con el formulario de la donación.
- El usuario selecciona el tipo de donación (Económica, Bienes , Servicios)
 - Caso 1 Económica
 - El sistema despliega el formulario de donaciones
 - El usuario puede optar por hacer el pago mediante Paypal o completar los datos para ser contactado.
 - El cliente selecciona enviar donación.
 - Caso 2 Bienes
 - El sistema despliega un formulario
 - El cliente completa el formulario
 - El cliente selecciona enviar donación
 - Caso 3 Servicios
 - El sistema despliega un formulario
 - El cliente completa el formulario
 - El cliente selecciona enviar donación
- El sistema guarda un registro de la donación, y envía un mail de notificación tanto al donante como a la ONG.

- **Flujos de eventos alternativos**

- **Flujo alternativo 1**

- 1A. El sistema no puede conectar con el sitio de Paypal
 - El sistema despliega un mensaje de error.
 - Vuelve al paso 7 del flujo normal.
- 1B. El sistema encuentra un error en el formulario
 - El sistema despliega un msj de error
 - Vuelve al paso 7 del flujo normal

- **Post-condiciones**

- 15. Se da alta en un registro en la tabla de donaciones, para la ONG, usuario y tipo de donación utilizada.

- **Solicitud de Ayuda**

- **Descripción.**

- Quando un visitante a través de la web solicita ayuda al sitio.

- **Pre-condiciones**

- 14. La catástrofe debe existir en el sistema.
 - 15. El visitante debe tener una conexión a internet.

- **Flujo de eventos principal**

- El visitante solicita ayuda (Selecciona botón ayuda)
 - El sistema despliega formulario de solicitud
 - El visitante ingresa datos (Nombre, teléfono, Descripción, Ubicación)
 - El sistema valida ingreso de solicitud (descripción not null)
 - El visitante confirma ingreso de solicitud
 - El sistema registra solicitud de ayuda, mensaje "solicitud ingresada con éxito"
 - Fin CU

- **Flujos de eventos alternativos**

- 42. **Flujo alternativo 1**

- 4A – Datos inválidos (descripción null)

- 42.1.1.** Mensaje "ingrese descripción"

- 42.1.2.** Vuelve al paso 2 del flujo normal

- **Post-condiciones**

- 16. Queda registrada la solicitud de ayuda.
 - 17. Envía mail de notificación a la organización de SIC y a los rescatistas involucrados en el momento en la catástrofe.

- **Botón de Pánico**

- **Descripción.**

Un visitante a través de un dispositivo móvil solicita ayuda.

- **Pre-condiciones**

16. La catástrofe debe existir en el sistema.

17. El visitante debe tener un dispositivo móvil conectado a internet.

- **Flujo de eventos principal**

- El visitante solicita ayuda (Selecciona botón de pánico)
- El sistema registra solicitud de ayuda, mensaje "solicitud ingresada con éxito"
- Fin CU

- **Flujos de eventos alternativos**

43. **Flujo Alternativo 1**

43.1. 3A - El visitante cancela

43.2. Fin CU

44. **Flujo Alternativo 2**

44.1. 4A - El sistema detecta posición incorrecta, mensaje "No hay catástrofe en su posición, ingrese descripción de su situación"

44.2. El visitante ingresa descripción

44.3. El sistema registra solicitud de ayuda, mensaje "solicitud ingresada con éxito"

44.4. Vuelve al paso 5 del flujo normal

- **Post-condiciones**

18. Queda registrada la solicitud de ayuda.

19. Envía mail de notificación a la organización de SIC y a los rescatistas involucrados en el momento en la catástrofe.

• Diagramas de Interacción

○ Crear Catástrofe

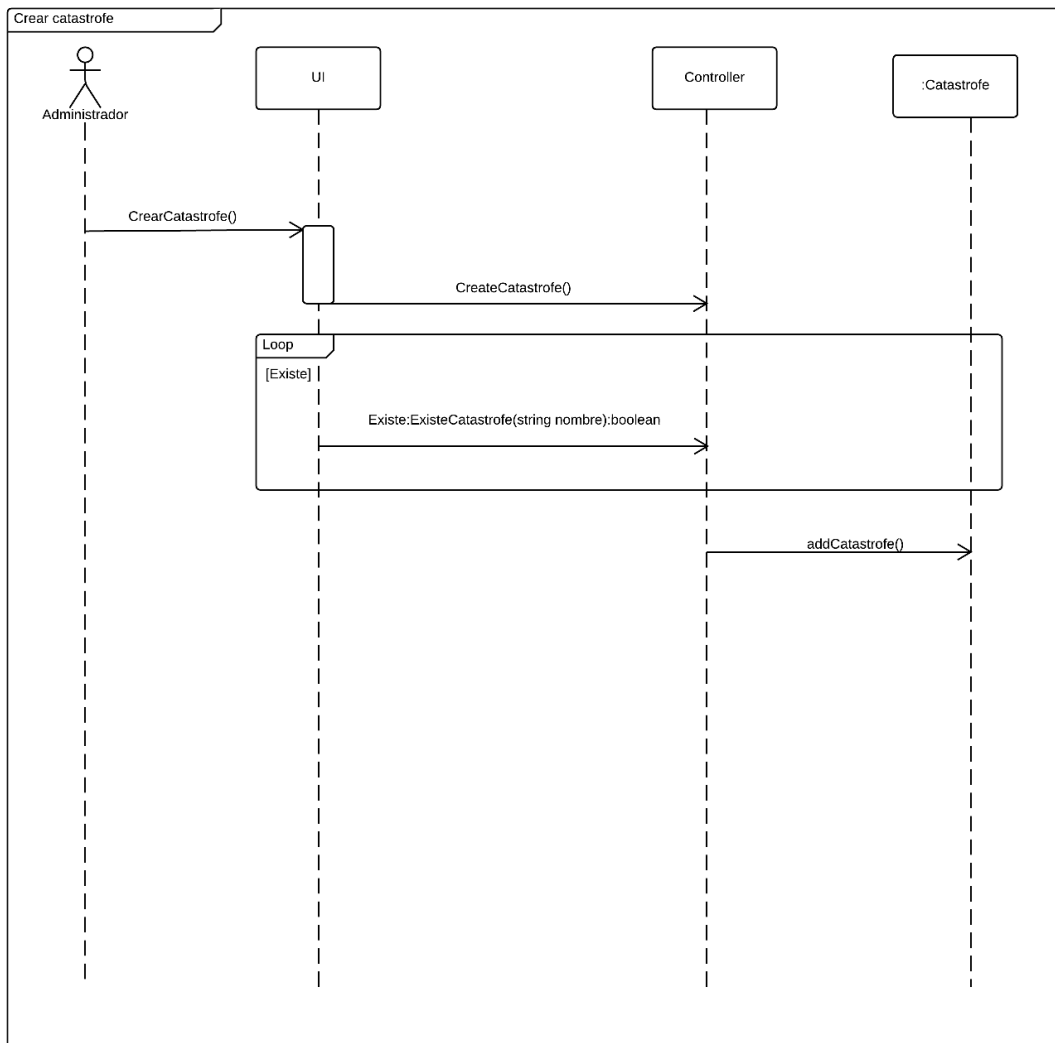


Ilustración 10 - Crear Catástrofe

○ **Realizar Donación**

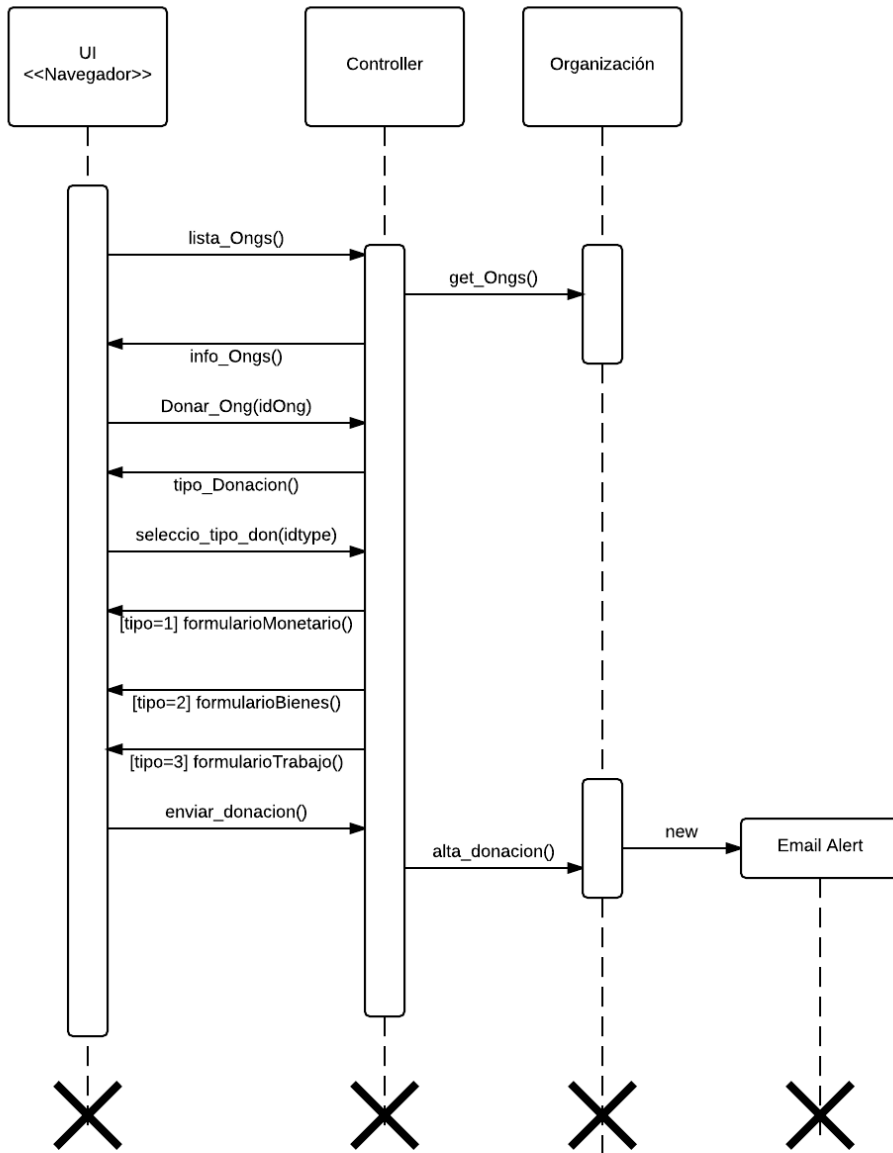


Ilustración 11 - Realizar Donación

○ Solicitud de ayuda

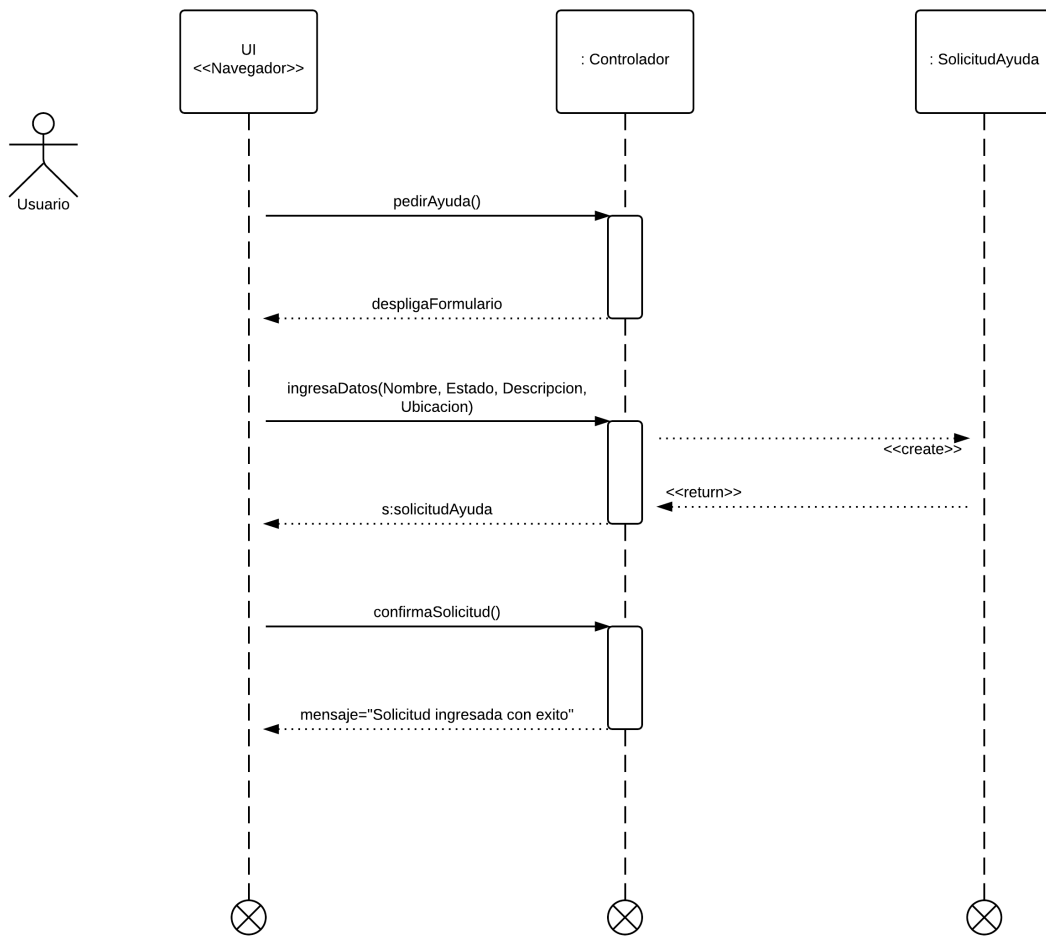


Ilustración 12 - Solicitud de Ayuda

- **Botón de pánico**

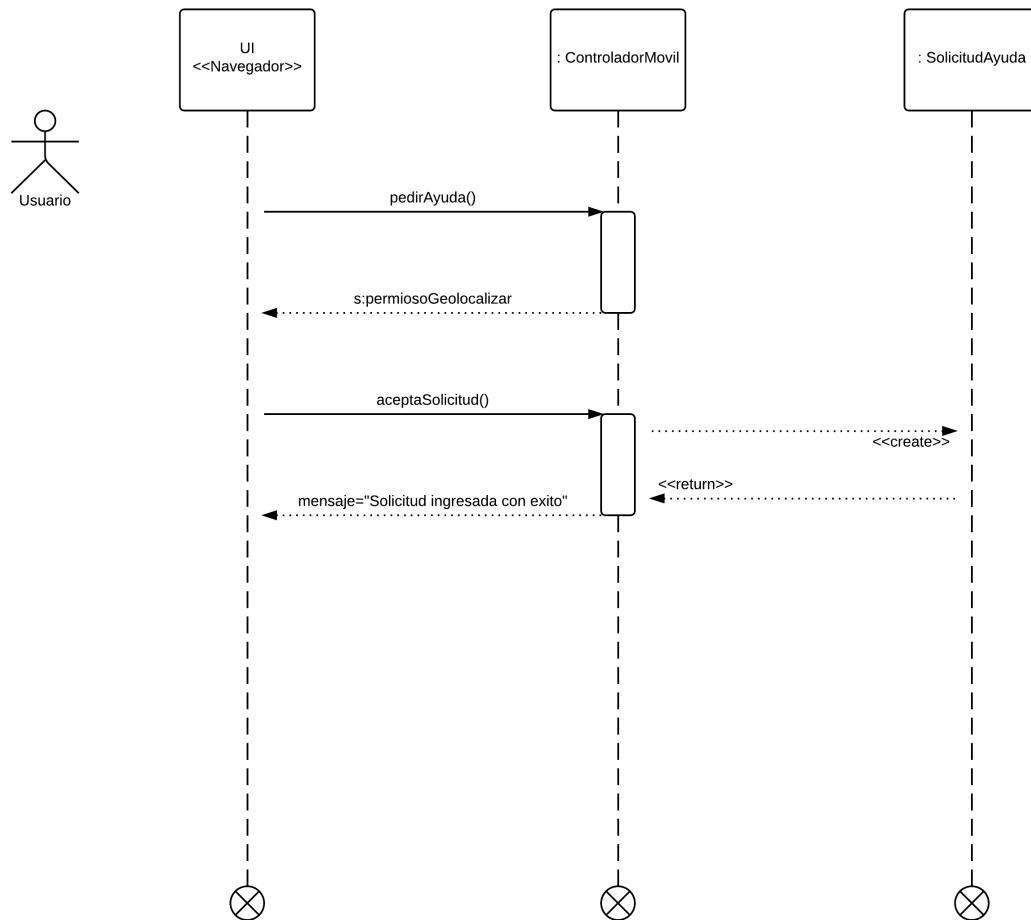


Ilustración 13 - Solicitud Ayuda Botón de Pánico

ANEXO [D]

Proyecto: SIC

Análisis de Tecnologías

Versión 2.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
05/02/2015	1.0	Primera Versión del Documento	Juan Ignacio Castro

08/02/2015	2.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera
------------	-----	--------------------------------	---

Contenido

Contenido.....	2
1. Análisis de tecnologías.....	3
2. Plataformas de desarrollo web.....	3
2.1. Java.....	3
2.2. Java EE - Aplicaciones empresariales en Java.....	3
2.3. Frameworks para aplicación web.....	4
2.4. Spring Framework.....	4
3. Patrón de Diseño.....	5
3.1. MVC.....	5
4. Servidor de Aplicaciones.....	6
4.1. JBoss Application Server.....	6
4.2. Características de JBoss.....	6
5. Lenguajes de Programación.....	6
5.1. Java.....	6
5.2. JavaScript.....	6
5.3. JQuery Mobile.....	7
5.4. JSON.....	7
5.5. Ajax.....	7
5.6. Bases de Datos Mysql.....	7
5.7. HTML5.....	7
5.8. CSS.....	7
6. Herramientas de Desarrollo.....	8
6.1. Eclipse.....	8
6.2. Android Studio.....	8
7. Versionado.....	9
7.1. GIT.....	9
8. Servicios Web.....	9
8.1. Servicios Web REST.....	10

- **Análisis de tecnologías:**

Se detalle el estudio de la actualidad de las tecnologías disponibles con el fin de evaluar las mismas y así poder obtener los resultados esperados, evitando caer en los errores comunes.

- **Plataformas de desarrollo web**

En los tiempos que corren las principales y más populares plataformas del lado del servidor son:

- Java
- PHP
- Microsoft .NET
- Python
- Ruby On Rails

- **Java**

Java es desde hace buen tiempo la plataforma más utilizada en el mundo empresarial. Con varios años en el mercado es una tecnología madura y popular con miles de herramientas de todo tipo y además de cientos de comunidades de programadores Java.

La principal ventaja de Java para el desarrollo de aplicaciones web es que la plataforma no se creó originalmente para ese fin, sino que fueron apareciendo proyectos como es el de [Tomcat](#) en 1999.

Las extensiones que se agregan a la plataforma se acuerdan mediante el Java Community Process (JCP), el cual está compuesto por más de mil miembros que trabajan sobre más de 300 Java Specification Requests (JSR)

La curva de aprendizaje de Java no es fácil. Esto no se debe a que el lenguaje en sí mismo sea muy complejo, sino porque además de conocer el propio lenguaje Java, se debe tener en cuenta que hay miles de librerías y herramientas de terceros.

En resumen, Java es una de las plataformas más rápida y escalable del mercado, y además permite un mayor grado de manejo a bajo nivel para optimizar el rendimiento.

- **Java EE - Aplicaciones empresariales en Java.**

El concepto de aplicación empresarial está fuertemente ligado con el concepto de un sistema distribuido. La aplicación tendrá la necesidad de contemplar aspectos como un aumento del número de visitas, aumento de aplicaciones, entre otros.

Un sistema distribuido permitirá cubrir estas necesidades ya que mejora tres aspectos fundamentales en una aplicación: disponibilidad, escalabilidad y mantenimiento.

Alta disponibilidad: El sistema debe estar funcionando las 24 horas del día los 365 días al año

Escalabilidad: El sistema debe tener la capacidad de crecer cuando se incrementa la carga de trabajo (el número de peticiones). Cada máquina tiene una capacidad

finita de recursos y por lo tanto sólo puede servir un número limitado de peticiones.

Mantenimiento: El sistema debe tener versatilidad a la hora de actualizar, depurar fallos y mantener un sistema. La solución al mantenimiento es la construcción de la lógica de negocio en unidades reusables y modulares.

Si se utilizara un sistema centralizado, un cambio en las necesidades del sistema provoca un colapso y la adaptación a dicho cambio puede resultar fatal.

Para realizar este cometido, se debe utilizar un servidor de aplicaciones para Java EE (cumpliendo con el estándar J2EE). Se utilizará Jboss como primera opción.

- **Frameworks para aplicación web**

El termino Framework se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

En pocas palabras, un framework se puede considerar como una aplicación genérica incompleta y totalmente configurable a la que podemos añadirle componentes para desarrollar una aplicación en concreto.

Los frameworks web más conocidos y utilizados para Java son: Spring, Java Server Faces (JSF), Vaadin y Google Web Toolkit.

Para el desarrollo de la aplicación se utilizará el framework Spring.

- **Spring Framework**

El beneficio de usar Spring sobre otros frameworks es que es de código abierto. Otra ventaja sustancial de Spring es su estructura en capas que le permite elegir sólo aquellos componentes que necesite mientras que nos ofrece un framework de desarrollo de aplicaciones Java EE muy completa y sin fracturas. Spring nos proporciona una arquitectura en capas lo que nos permite usar lo que vamos a utilizar y prescindiendo de lo innecesario. Spring permite a los desarrolladores centrarse en la programación de Objetos Planos Java (POJO), esto permite realizar pruebas e integración continuas. La Inyección de Dependencias y la inversión del control hacen la conectividad a base de datos más simple.

Spring gestiona y contiene la configuración y ciclo de vida de aplicación de los objetos.

Es bastante liviano o ligero en tamaño, la versión más ligera del framework de Spring sólo ocupa 1 MB, o sea que para la cantidad de servicios que ofrece es insignificante su tamaño. Además, la sobrecarga en términos de procesador es similarmente minúscula.

Spring utiliza el patrón web Modelo Vista Controlador (MVC) que se construye como el núcleo de funcionalidad Spring. Este framework puede dar cabida a muchas tecnologías de presentación como Java Server Pages (JSP), Tiles, Velocity, etc. Además, es altamente configurable. Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar de que no impone ningún modelo de [programación](#) en particular, este framework se ha vuelto popular en la comunidad al ser considerado una alternativa o incluso un complemento al modelo EJB (Enterprise JavaBean).

- **Patrón de Diseño**

- **MVC**

MVC (Modelo Vista Controlador), es de los patrones de diseño más utilizados en la actualidad, ya que se utiliza en una gran variedad de aplicaciones y brinda beneficios a los desarrolladores.

Si se utiliza en frameworks como Spring o Microsoft ASP .NET, pero no es exclusivo de una tecnología, debido a que es adaptable.

El patrón MVC (Modelo Vista Controlador) define, separa e independiza la capa del Modelo (Modelo del Negocio), la Vista (interfaz de usuario y la capa del Controlador).

Esta estructura de bajo acoplamiento, facilita su mantenimiento

Las tres capas que componen el patrón MVC son:

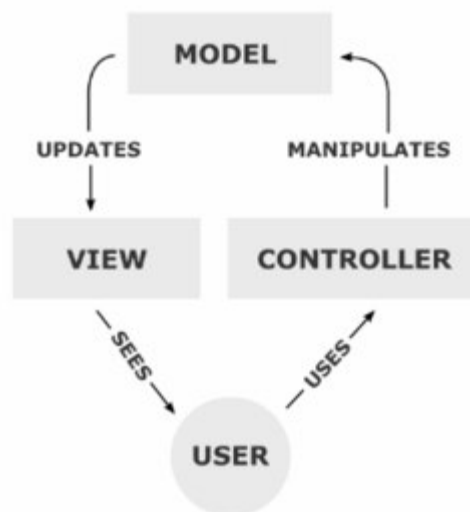


Imagen 1 – Modelo - Vista - Controlador

Modelo: Representa el modelo de datos de la aplicación. El modelo no tiene conocimiento directo tanto de la capa Controlador como de la Vista, no tiene referencia a ellos, es independiente a ellos.

El sistema se encargará relacionarlos y avisar a las vistas cuando el modelo cambie.

Vista: La Vista es la representación visual de los datos representados en la capa Modelo. Tiene una referencia al modelo Se encarga de interactuar con el usuario.

Controlador: Esta capa se encarga de responder a las acciones del usuario. Se enfoca en tratar con los datos de la capa Modelo. Tiene contacto con la capa Vista. En las arquitecturas de 3 capas como MVC, la capa Controlador se encarga del relacionamiento con las capas más bajas, como la de Negocio.

- **Servidor de Aplicaciones**

- **JBoss Application Server**

JBoss es un proyecto de código abierto, un servidor de aplicaciones basado en J2EE, implementado en un 100% en Java. Como está basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte.

JBoss implementa todo el paquete de servicios del estándar J2EE y además ofrece características como: clustering, Java Management Extensions (JMX), Web Services y la integración Internet Inter-ORB Protocol (IIOP), los objetos hacen peticiones y reciben respuestas, definidas como Object Request Broker (ORB).

La principal característica desde que JBoss está bajo la Licencia Pública General Reducida de GNU (LGPL) es que se puede usar libremente sin costo alguno en cualquier aplicación comercial redistribuida.

- **Características de JBoss**

- Código abierto
- Escalable
- Alto desempeño
- Arquitectura Modular
- Producto de licencia de código abierto sin coste adicional
- Cumple los estándares
- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios
- Flexibilidad consistente
- Servicios del middleware para cualquier objeto de Java
- Ayuda profesional 24x7 de la fuente
- Soporte completo para JMX

- **Lenguajes de Programación**

Como [lenguaje](#) se entiende a un [sistema](#) de [comunicación](#) que posee una determinada estructura, contenido y uso. La [programación](#) es, en el vocabulario propio de la [informática](#), el procedimiento de escritura del código fuente de un [software](#). De esta manera, puede decirse que la programación le indica al programa informático qué acción tiene que llevar a cabo y cuál es el modo de concretarla.

Lenguajes utilizados en el proyecto:

- **Java**

Java es un [lenguaje de programación](#) de [propósito general](#), [concurrente](#), [orientado a objetos](#) que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los [desarrolladores](#) de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo.

- **JavaScript**

JavaScript es un lenguaje de scripting basado en objetos. La alianza hizo que Javascript se diseñó como un hermano pequeño de Java, solamente útil dentro de las páginas web y mucho más fácil de utilizar, de modo que cualquier persona, sin conocimientos de programación pudiese adentrarse en el lenguaje y utilizarlo a sus anchas

- **JQuery Mobile**

JQuery Mobile es un framework Javascript para desarrollo de aplicaciones web para móviles, basado en jQuery. En resumen, JQuery es una forma de convertir el desarrollo de la parte de cliente de una aplicación web en algo mucho más divertido, rápido y sencillo, facilitando la interacción con los elementos del árbol de documento, el manejo de eventos, el uso de animaciones, etc.

- **JSON**

JSON es un acrónimo de *JavaScript Object Notation*, un formato ligero originalmente concebido para el intercambio de datos en Internet. Se considera un subconjunto de la notación literal para representar objetos, arrays, cadenas, booleanos y números en Javascript.

- **Ajax**

AJAX es un acrónimo para *Asynchronous JavaScript And XML* (JavaScript Asíncrono y XML). Permite la creación de aplicaciones interactivas en el desarrollo web.

Es decir, con AJAX seremos capaces de crear sitios web que se ejecuten directamente en el navegador del usuario manteniendo una comunicación con el servidor siempre que sea necesario pero sin recargar la página que estamos visualizando, simplemente se realizarán cambios sobre ella.

- **Bases de Datos Mysql**

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha

- **HTML5**

HTML5 es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella. HTML5 está siendo desarrollado por Ian Hickson de Google Inc. y David Hyatt de Apple Inc. junto con todas las personas que participan en *Web Hypertext Application Technology Working Group*.

- **CSS**

CSS es un lenguaje de estilo que define la presentación de los documentos HTML. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas

- **Herramientas de Desarrollo**

- **Eclipse**

Con sus perspectivas, editores y vistas Eclipse permite trabajar en un determinado entorno de trabajo de forma óptima.

El desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí.

El IDE proporciona asistentes y ayudas para la creación de proyectos, por ejemplo cuando se crea uno, se abre la perspectiva adecuada al tipo de proyecto que se esté creando, con la colección de vistas, editores y ventanas pre-configuradas por defecto.

Incluye un potente depurador de código, que visualmente ayuda a mejorar el código. Para ello sólo se debe ejecutar el programa en modo depuración.

Están disponibles una gran cantidad de plug-in, unos publicados por Eclipse, otros por terceros. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que sea necesaria, existe el plug-in adecuado.

Una cuestión muy interesante es que cuando se tiene una vista activa, se visualizan en la barra de herramientas iconos extra, que permiten el acceso rápido a las funciones más usadas de dicha vista.

El coloreado de código en el editor es una característica muy interesante, realizando para ello el reconocimiento sintáctico de todas aquellas palabras que son reservadas en el lenguaje Java.

Así mismo permite completar el código con sugerencias dependientes del contexto, lo cual permitirá escribir código más rápidamente.

- **Android Studio**

Android Studio es un entorno de desarrollo de Android desarrollado por Google. Es una alternativa a Eclipse + ADT Plugin, proporciona herramientas de desarrollo integradas para de Android, tanto para desarrollar como depurar.

Las principales características son:

- Soporte para Gradle.
- Refactorización específica para Android y soluciones rápidas.
- Herramientas para medir el rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- ProGuard y capacidad para firmar apps
- Asistentes basados en plantillas para crear diseños y componentes en Android.
- Un editor rico en cuanto a diseño, que permite realizar drag and drop a los componentes UI. Obtener vista previa de los layouts en múltiples configuraciones de pantalla en simultáneo.

Tiene actualizaciones permanentes y está disponible para varias plataformas, como Windows y Linux.

- **Versionado**

- **GIT**

Es un software de [control de versiones](#) diseñado por [Linus Torvalds](#), pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de [código fuente](#).

Principales características de Git:

- Auditoría del código: saber quién ha tocado qué y cuándo
- Control sobre cómo ha cambiado nuestro proyecto con el paso del tiempo
- Volver hacia atrás de una forma rápida
- Control de versiones a través de etiquetas: versión 1.0, versión 1.0.1, versión 1.1, etc. Sabremos exactamente que había en cada una de ellas y las diferencias entre cualquiera de ellas dos
- Seguridad: todas las estructuras internas de datos están firmadas con SHA1. No se puede cambiar el código sin que nos enteremos
- Mejora nuestra capacidad de trabajar en equipo
- Merging y branching extremadamente eficientes

- **Servicios Web**

Los servicios web o web services son componentes que permiten a la comunicación entre componentes de una aplicación o incluso entre dos aplicaciones independientes. La interface de los servicios web está definida en términos de los mensajes que él mismo acepta y retorna, por lo cual, la aplicación "consumidora" del mismo puede ser implementado en cualquier plataforma y lenguaje de programación, sólo tiene que poder crear y consumir los mensajes definidos por su interface

Un servicio web es descripto, publicado e invocado a través de la red. Pueden ser accedidos mediante la utilización de protocolos web como ser HTTP / XML.

Ventajas

- Interoperabilidad entre componentes.
- Ubicuidad
- Encapsulamiento
- Facilidad de utilización
- Soporte de la industria

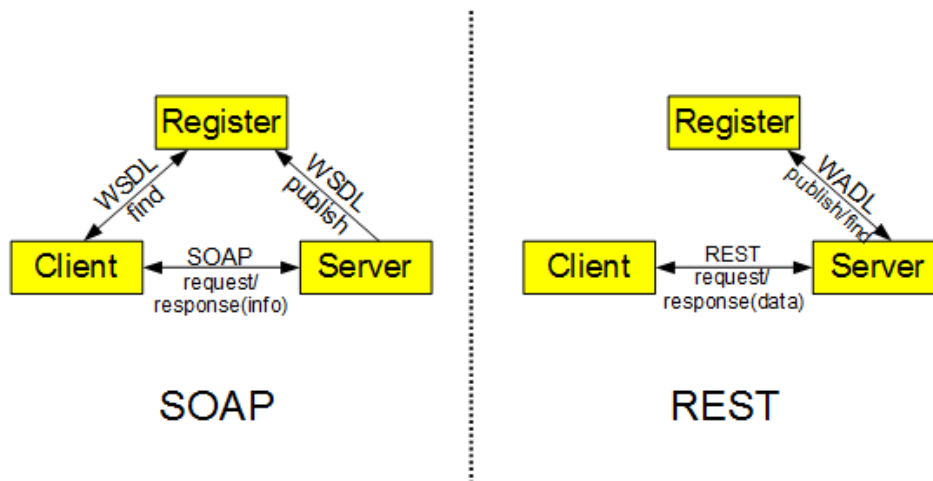


Imagen 2– Servicios Web SOAP y REST

Desventajas

- Se debe conocer donde está alojado el servicio (WSDL).
- Confiabilidad. Se desconoce cuándo el servicio no está disponible.
- Responsabilidad

○ **Servicios Web : REST (*Representational state transfer*)**

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podríamos considerar REST como un framework para construir aplicaciones web respetando HTTP.

Por lo tanto REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet.

ANEXO [E]

Proyecto: SIC

Documento de Visión

Versión 2.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
01/02/2015	1.0	Primera Versión del Documento	Juan Ignacio Castro
08/02/2015	2.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

Proyecto: SIC.....	1
Documento de Visión.....	1
Versión 2.0.....	1
Historia de revisiones.....	1
Contenido.....	2
1. Introducción.....	3
1.1. Propósito.....	3
1.2. Alcance.....	3
2. Misión y Visión.....	3
2.1. Misión.....	3
2.2. Visión.....	3

- **Introducción**

- **Propósito**

El propósito de este documento es definir el marco para la visión y misión, de forma que los participantes en el proyecto realicen sus tareas y objetivos basados en los mismos.

- **Alcance**

El alcance del documento abarca toda la extensión del proyecto.

- **Misión y Visión**

La misión es lo que busca ser este proyecto a partir del momento de su puesta en producción.

Por otro lado, la visión es lo que pretende ser el proyecto a largo plazo.

- **Misión**

Proyecto SIC pretende ser una solución integral de información para gestionar catástrofes. Pretende abarcar tanto el punto de vista del desaparecido, como así también la de los actores principales en la gestión de un plan de emergencia, o cuando ocurre un hecho desastroso.

En resumen, el proyecto SIC pretende ayudar a las personas a nivel mundial a mitigar las consecuencias de las catástrofes, evitando las pérdidas de vidas humanas y económicas.

- **Visión**

El producto desarrollado por este proyecto pretende convertirse en una valiosa herramienta de ayuda en el momento que ocurren una catástrofe y su posterior aparte para la gestión del mismo. Por otro lado, pretende ser una aplicación respaldo para ayudar a los individuos que posean un teléfono móvil y se encuentran en una zona de catástrofe.

En conclusión, proyecto S.I.C. apunta a ser la aplicación de software referente en el mundo, para gestionar y mejorar el tratamiento de las catástrofes.

ANEXO [F]

Proyecto: S.I.C.

Descripción de la Arquitectura

Versión [1.3]

Historia de revisiones

Fecha	Versión	Descripción	Autor
10/22/2014	1.0	Vista Modelo de Diseño	Sandino Fratti
10/26/2014	1.1	Se agregan vista C.U	Sandino Fratti
10/27/2014	1.2	Se modifican puntos en todo el doc.	Sandino Fratti
08/02/2015	1.3	Se modifican diagramas C.U	Sandino Fratti
09/02/2015	2.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

1. Introducción.....	3
1.1. Propósito.....	3
1.2. Alcance.....	3
1.3. Definiciones, siglas y abreviaturas.....	3
1.4. Referencias.....	3
1.5. Visión general.....	3
2. Vista del Modelo de Casos de Uso.....	4
2.1. Diagrama de Casos de Uso relevantes a la Arquitectura.....	4
2.2. Casos de Uso relevantes a la Arquitectura.....	5
2.2.1. Crear Catástrofe.....	5
2.2.2. Realizar Donación.....	5
2.2.3. Solicitud de Ayuda.....	5
2.2.4. Botón de Pánico.....	5
3. Vista del Modelo de Diseño.....	5
3.1. Descomposición en Subsistemas.....	5
3.1.1. Presentación.....	5
3.1.2. Capa Servicios.....	6
3.1.3. Capa Lógica.....	6
3.2. Diseño de Clases.....	6
3.2.1. UsuarioController.....	6
3.2.2. CatastrofeController.....	6
3.2.3. OngController.....	6
4. Vista del Modelo de Implementación.....	7
4.1. Subsistemas.....	7
4.2. Componentes.....	7
4.2.1. Controlador.....	7
4.2.2. DAO.....	7
4.3. Interfaces.....	7
4.3.1. Controladores - Servidor de aplicaciones.....	7
4.3.2. Acceso a datos - DAO.....	7
5. Vista del Modelo de Distribución.....	7
5.1. Diagrama de Distribución.....	7
5.2. Nodos.....	8
5.2.1. Servidor de Base de Datos (MySQL Server).....	8
5.2.2. Servidor de aplicaciones (JBoss Server-WildFly).....	8
5.2.3. Servidor de Web.....	8

- **Introducción**

19.1. Propósito

Este documento proporciona una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

19.2. Alcance

Este Documento de Arquitectura de Software proporciona una descripción de la distribución e interacción entre componentes del sistema. El mismo ha sido generado directamente del Análisis de los casos de uso y del Modelo de Diseño

19.3. Definiciones, siglas y abreviaturas.

Para esta información refiérase al glosario.

19.4. Referencias

Modelo Casos de Usos

Modelo de Diseño

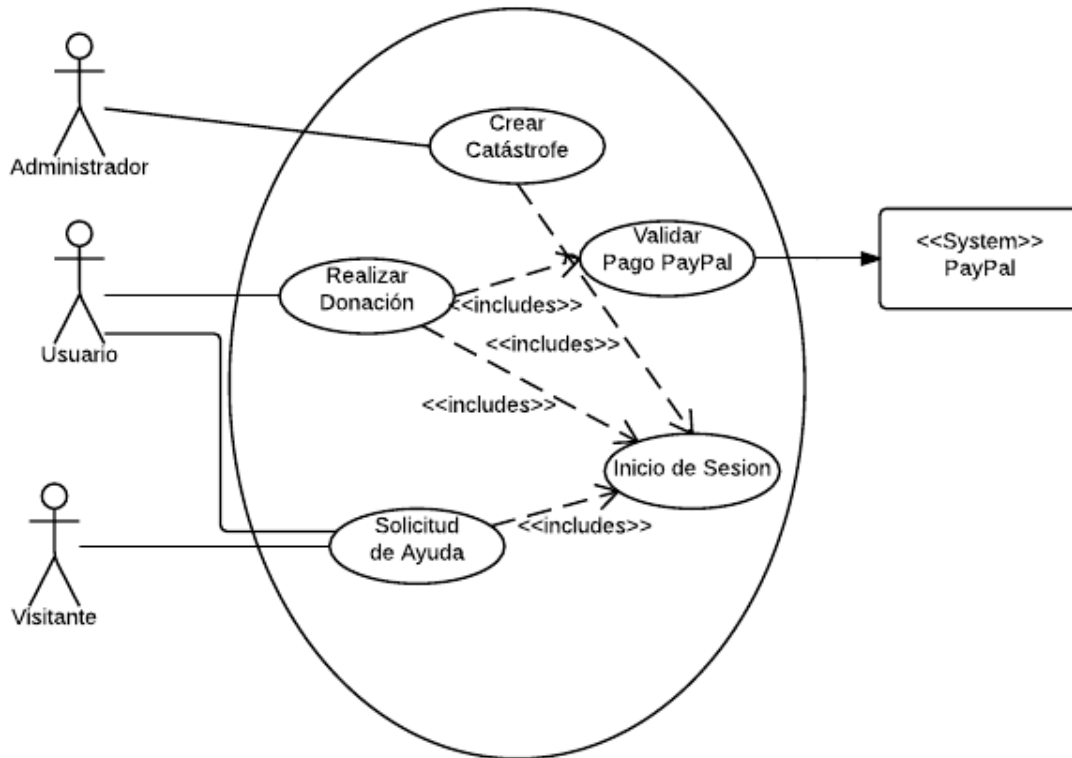
19.5. Visión general

En esta iteración el documento de arquitectura contiene información acerca de los casos de uso más relevantes a la hora de diseñar el sistema a construir, seguido de los subsistemas de la arquitectura y su distribución física al momento de su despliegue.

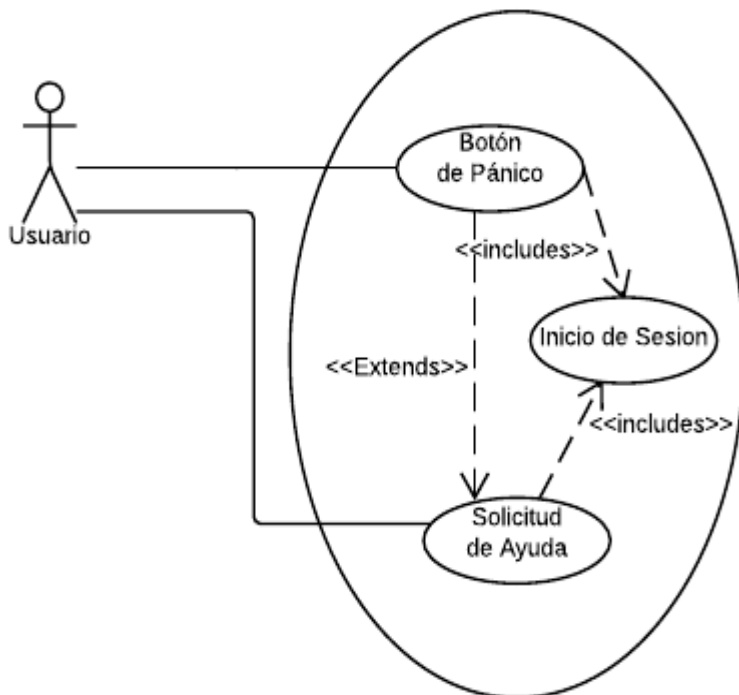
- **Vista del Modelo de Casos de Uso**

19.6. Diagrama de Casos de Uso relevantes a la Arquitectura

Web



Android



19.7. Casos de Uso relevantes a la Arquitectura

Se detallan los Casos de Uso y actores más significantes para definir la Arquitectura del sistema. Estos casos de uso son:

Crear Catástrofe.

Realizar Donación.

Solicitud de Ayuda.

Botón de Pánico.

▪ **Crear Catástrofe.**

Crea una nueva catástrofe y guarda en la base de datos para que quede accesible en el sistema. El administrador del sistema ingresa los datos de la nueva catástrofe a crear, nombre de la catástrofe, ubicación, logo, descripción, imágenes, etc.

▪ **Realizar Donación.**

Permite a los usuarios realizar donaciones a una ONG registrada en el sitio de la catástrofe.

▪ **Solicitud de Ayuda.**

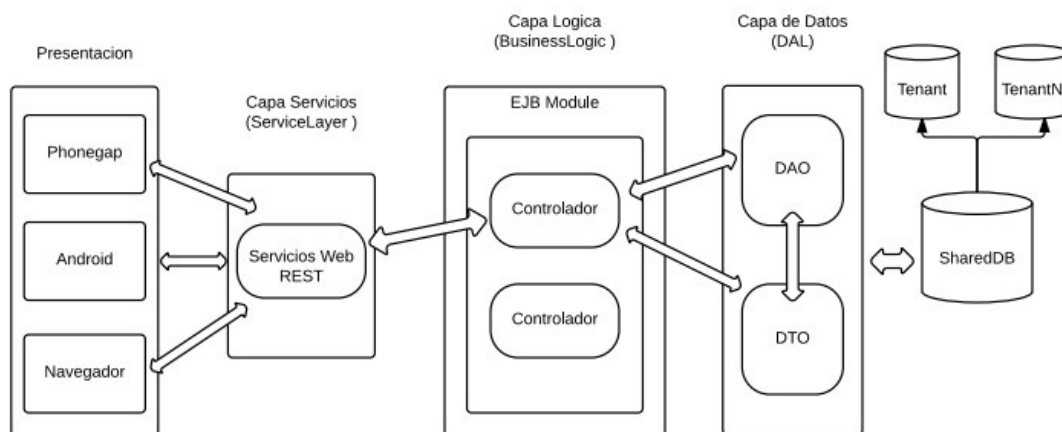
Cuando un visitante a través de la web solicita ayuda al sitio.

▪ **Botón de Pánico.**

Cuando un visitante a través de la web solicita ayuda al sitio.

• **Vista del Modelo de Diseño**

19.8. Descomposición en Subsistemas



▪ **Presentación**

Tiene la interfaz de usuario. Se encarga de mostrar los componentes gráficos y de la interacción con el usuario pero no implementar ninguna parte de la lógica del negocio, la misma se consume a través de los servicios web expuestos por esta capa.

▪ **Capa Servicios**

Contiene los servicios (REST) expuestos por la lógica de negocios. Permite la comunicación con los dispositivos móviles y el navegador, los cuales consumen de

estos, permitiendo que el usuario pueda realizar los casos de uso correspondientes.

- **Capa Lógica**

Este subsistema se encarga principalmente de proveer todos los datos necesarios para la capa de presentación. Tiene toda la lógica asociada a la aplicación. La cual es expuesta como servicios web para ser consumidos desde cualquier componente de la Capa Presentación.

4.1.4 Capa de Datos

El subsistema de base de datos tiene como objetivo guardar de forma persistente todos aquellos datos que sean necesarios tener disponible. Se utiliza JPA como interfaz para el acceso a datos que quiebra la dependencia de los componentes que este utiliza. Los componentes DTO's, se alojan en la capa de datos pero son compartidos por la capa de negocios cumpliendo la tarea de transportar información.

19.9. Diseño de Clases

Las clases principales son:

- UsuarioController
- CatastrofeController
- OngController

- **UsuarioController**

Agrupar todos los métodos referentes al Usuario y las acciones que éste mismo puede realizar en el sistema, interactuando con otros componentes en el caso que sea necesario. Expone la interfaz IUserario.

- **CatastrofeController**

Agrupar todos los métodos referentes a la Catástrofe y todas las funcionalidades que son requeridas por los diferentes procesos que ocurren en el sitio de la Catástrofe. Expone la interfaz ICatastrofe.

- **OngController**

Encapsula todos los métodos referentes a esta entidad y su interacción con otros componentes. Expone la interfaz IOng.

- **Vista del Modelo de Implementación**

19.10. Subsistemas

19.11. Componentes

- **Controlador**

Pertenece a la capa de negocio, es encargado de manejar la lógica y delegar funcionalidades a otros componentes del sistema.

- **DAO**

Gestiona el acceso a datos en la aplicación. Tiene funciones que permiten obtener objetos para el lenguaje java a partir de la información almacenada en la base de datos.

19.12. Interfaces

- **Controladores - Servidor de aplicaciones**

Se expone una interfaz para los servicios web. De esta manera es posible la independencia de la lógica de negocio y los consumidores de la misma.

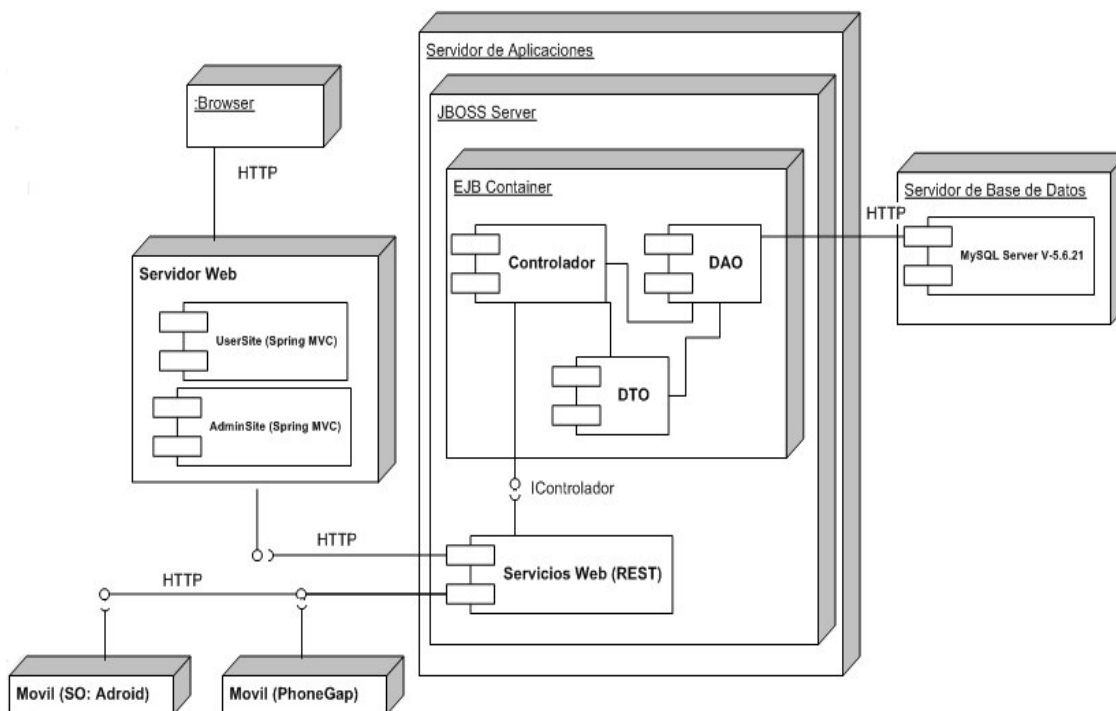
- **Acceso a datos - DAO**

-

La base de datos tiene una interfaz de acceso a datos. La misma abstrae el componente de bases de datos y la implementación física de este.

- **Vista del Modelo de Distribución**

19.13. Diagrama de Distribución



19.14. Nodos

- **Servidor de Base de Datos (MySQL Server)**

Este servidor es el encargado de alojar los datos del sistema y prestar los servicios que actúan o consultan sobre los mismos. Se accederá al mismo por medio del servidor de aplicaciones.

- **Servidor de aplicaciones (JBoss Server-WildFly)**

El servidor de aplicaciones es el principal intermediario entre la interfaz de usuario y los datos del sistema. El mismo contiene la mayor parte de la lógica de negocio del mismo. Se utilizará JBoss como servidor de aplicaciones (contenedor EJB).

- **Servidor de Web**

Almacena los documentos necesarios para la interacción con el usuario cuando este lo solicita. El nodo Servidor web y Servidor de aplicaciones pueden ser alojados ambos en el mismo servidor, en este caso en el servidor de aplicaciones JBoss.

ANEXO [G]

Proyecto: S.I.C.

Modelo de Diseño

Versión 3.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
26/10/2014	1.0	Versión inicial del documento	Gastón
27/10/2014	2.0	Actualización del documento	Ignacio
08/02/2015	3.0	Revisión General	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

1. Introducción.....	3
1.1. Propósito.....	3
El propósito es hacer una descripción específica de los componentes a incluir en el proyecto..	3
1.2. Visión general.....	3
2. Diseño de Casos de Uso.....	3
2.1. Crear Catástrofe.....	3
2.1.1. Diagrama de secuencia.....	3
2.2. Realizar Donación.....	4
2.2.1. Diagrama de secuencia.....	4
2.3. Solicitud de Ayuda.....	5
2.3.1. Diagrama de secuencia.....	5
2.4. Botón de Pánico.....	6
2.4.1. Diagrama de secuencia.....	6
3. Diagrama de Clases de Diseño.....	7

- **Introducción**

Este documento corresponde al Modelo de Diseño del proyecto S.I.C., en el mismo se brinda una visión general del Diseño del proyecto Web y del proyecto para dispositivos móviles.

19.15. Propósito

El propósito es hacer una descripción específica de los componentes a incluir en el proyecto.

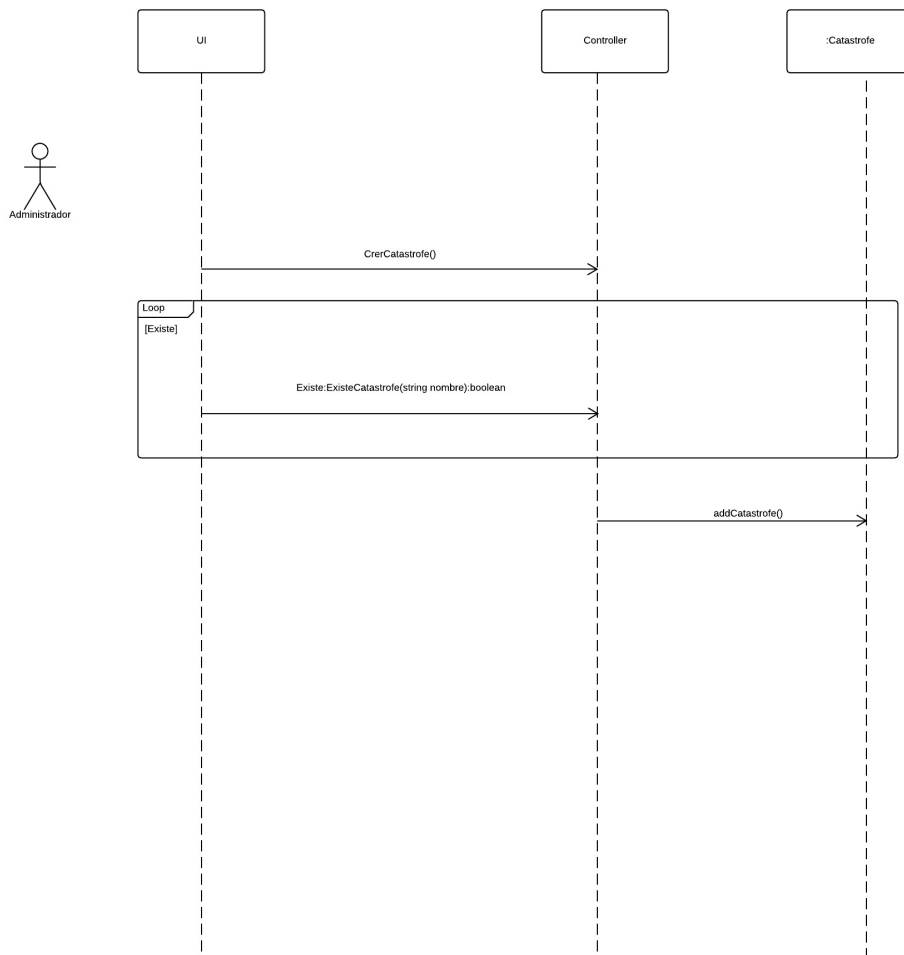
19.16. Visión general

El diseño será realizado teniendo en cuenta la arquitectura.

- **Diseño de Casos de Uso**

19.17. Crear Catástrofe

- **Diagrama de secuencia**

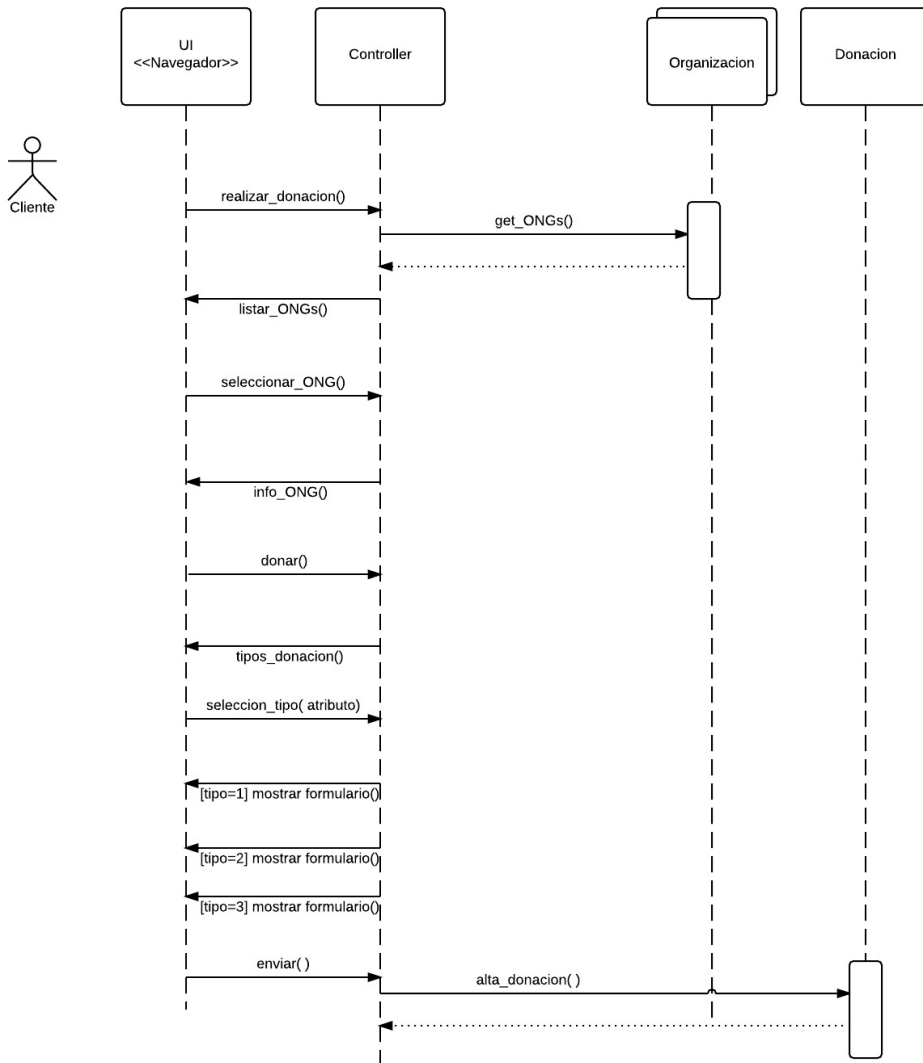


19.18.
Realiz
a
r

Donación

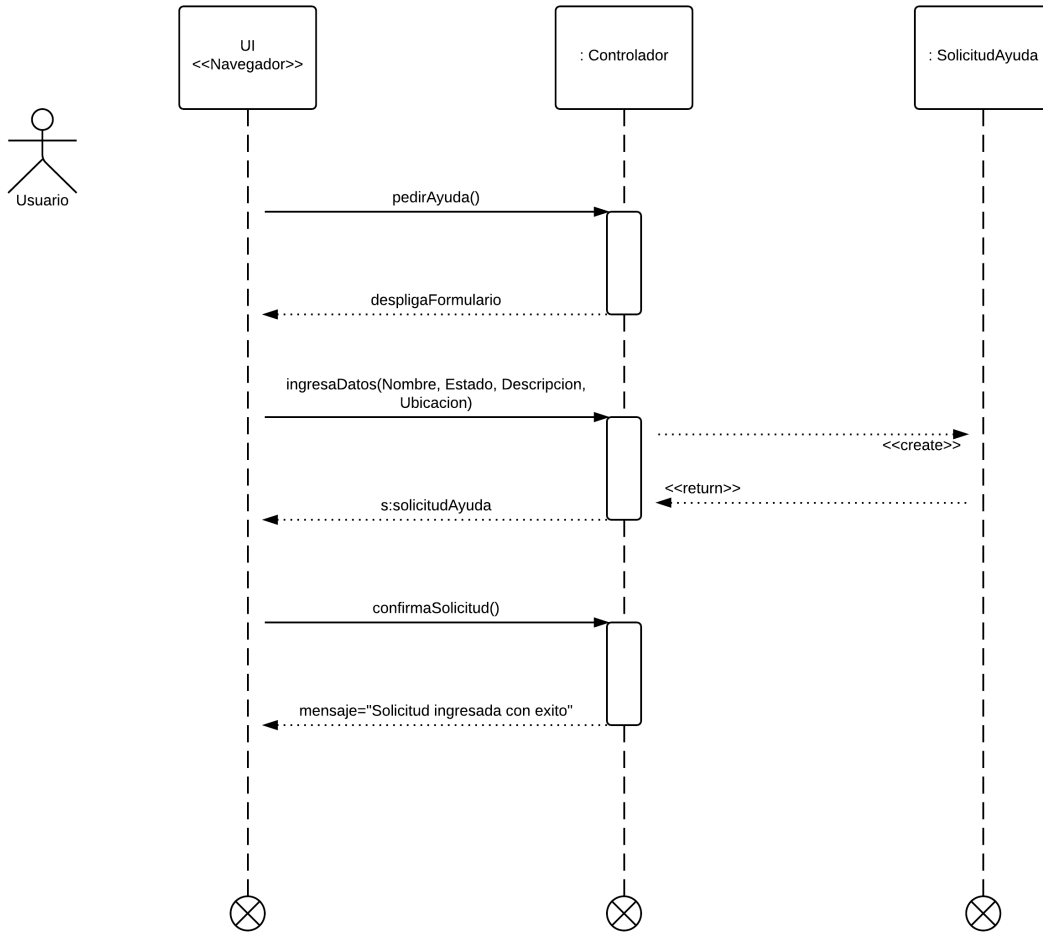
▪ Diagrama de secuencia

19.19.

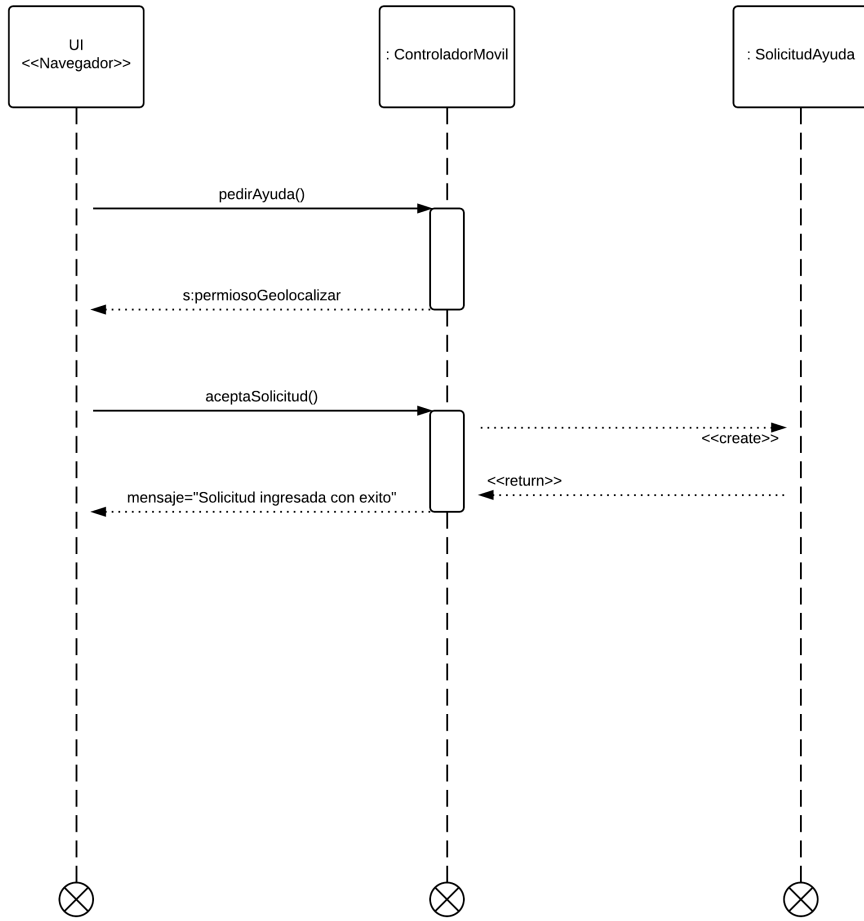
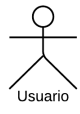


Solicitud de Ayuda

▪ Diagrama de secuencia



19.20. Botón de Pánico
 ▪ **Diagrama de secuencia**



Proyecto: S.I.C.
Modelo de Diseño
Versión 3.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
26/10/2014	1.0	Versión inicial del documento	Gastón
27/10/2014	2.0	Actualización del documento	Ignacio
08/02/2015	3.0	Revisión General	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

1. Introducción.....	3
1.1. Propósito.....	3
El propósito es hacer una descripción específica de los componentes a incluir en el proyecto..	3
1.2. Visión general.....	3
2. Diseño de Casos de Uso.....	3
2.1. Crear Catástrofe.....	3
2.1.1. Diagrama de secuencia.....	3
2.2. Realizar Donación.....	4
2.2.1. Diagrama de secuencia.....	4
2.3. Solicitud de Ayuda.....	5
2.3.1. Diagrama de secuencia.....	5
2.4. Botón de Pánico.....	6
2.4.1. Diagrama de secuencia.....	6
3. Diagrama de Clases de Diseño.....	7

- **Introducción**

Este documento corresponde al Modelo de Diseño del proyecto S.I.C., en el mismo se brinda una visión general del Diseño del proyecto Web y del proyecto para dispositivos móviles.

19.21. Propósito

El propósito es hacer una descripción específica de los componentes a incluir en el proyecto.

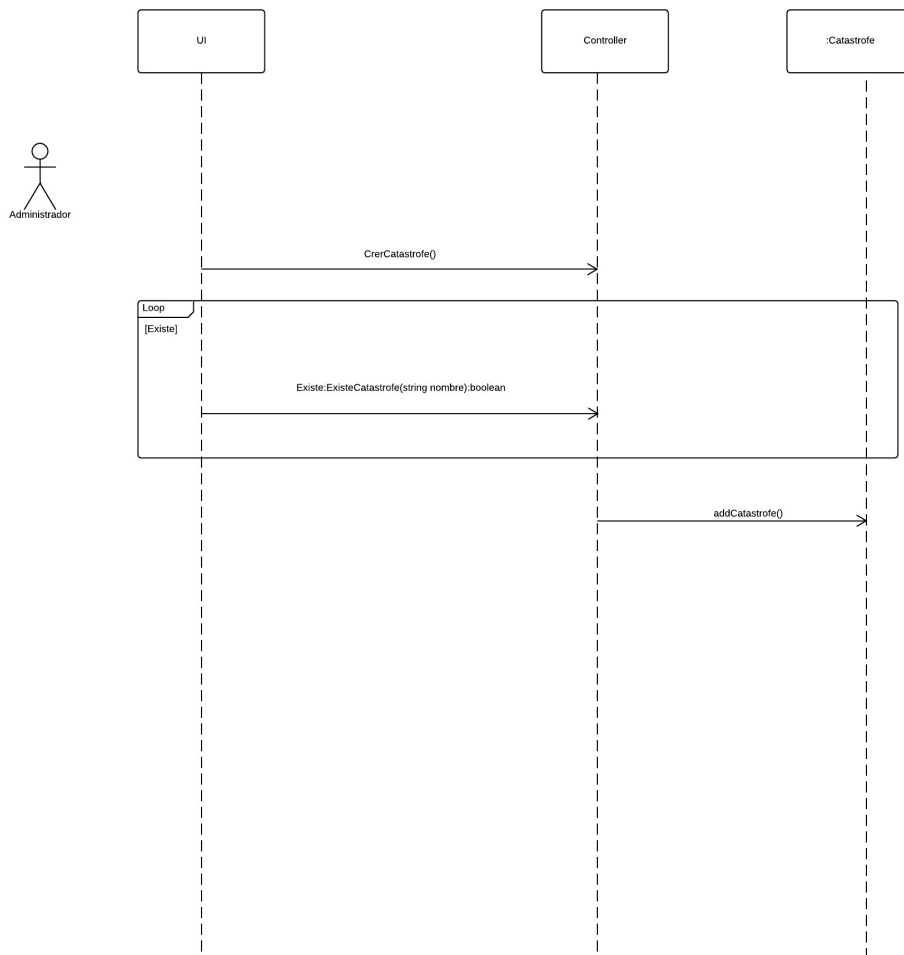
19.22. Visión general

El diseño será realizado teniendo en cuenta la arquitectura.

- **Diseño de Casos de Uso**

19.23. Crear Catástrofe

- **Diagrama de secuencia**

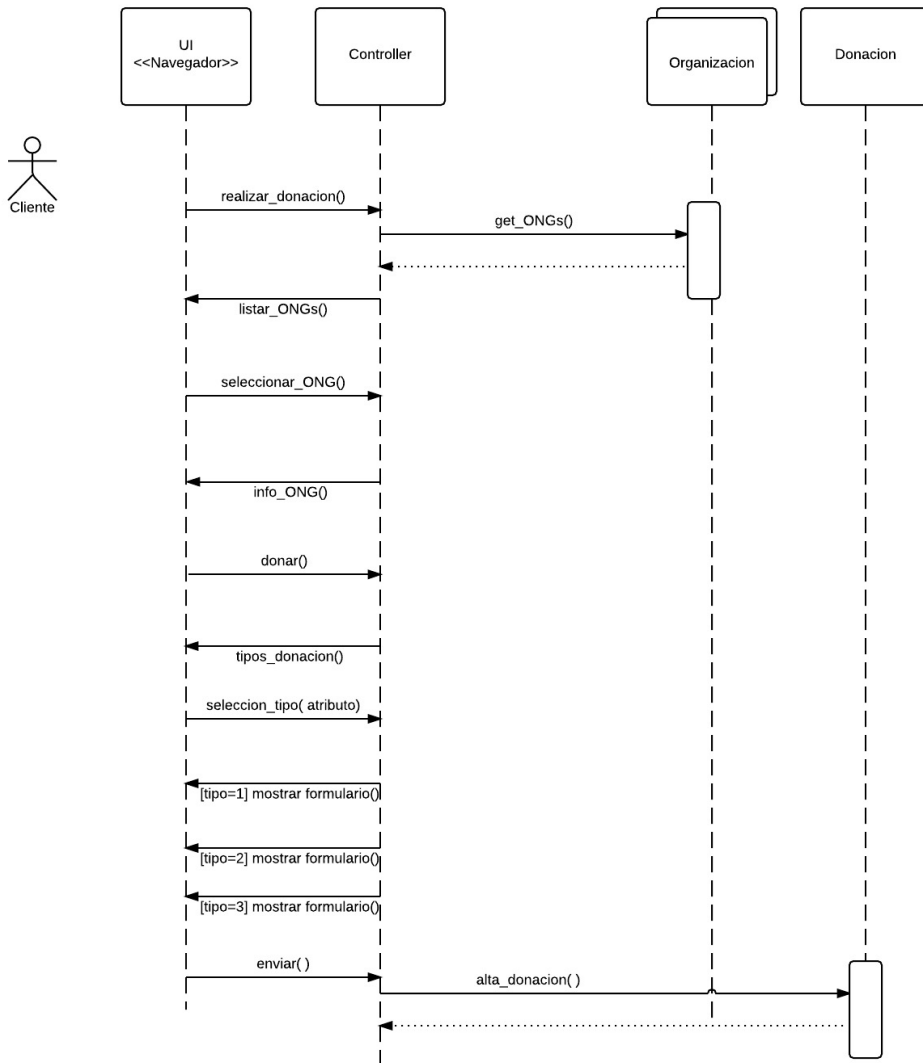


19.24. Realizar

Donación

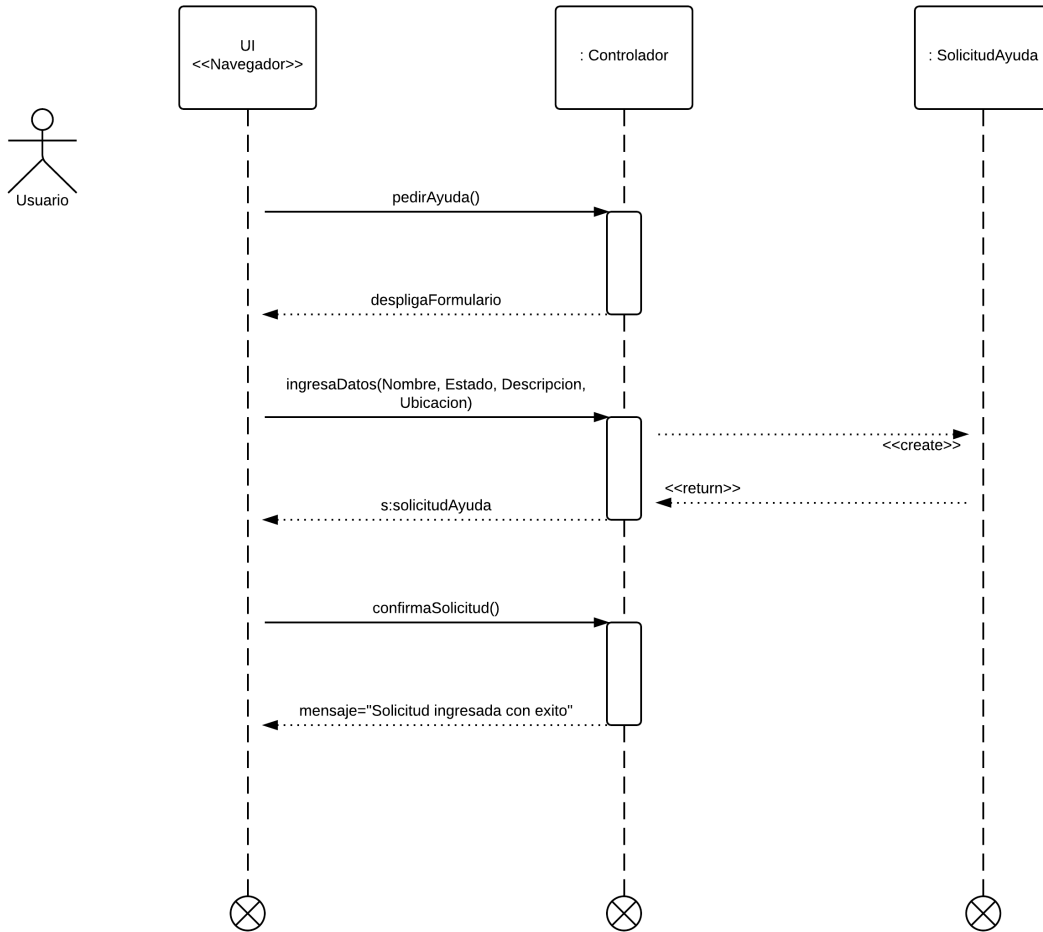
▪ Diagrama de secuencia

19.25.

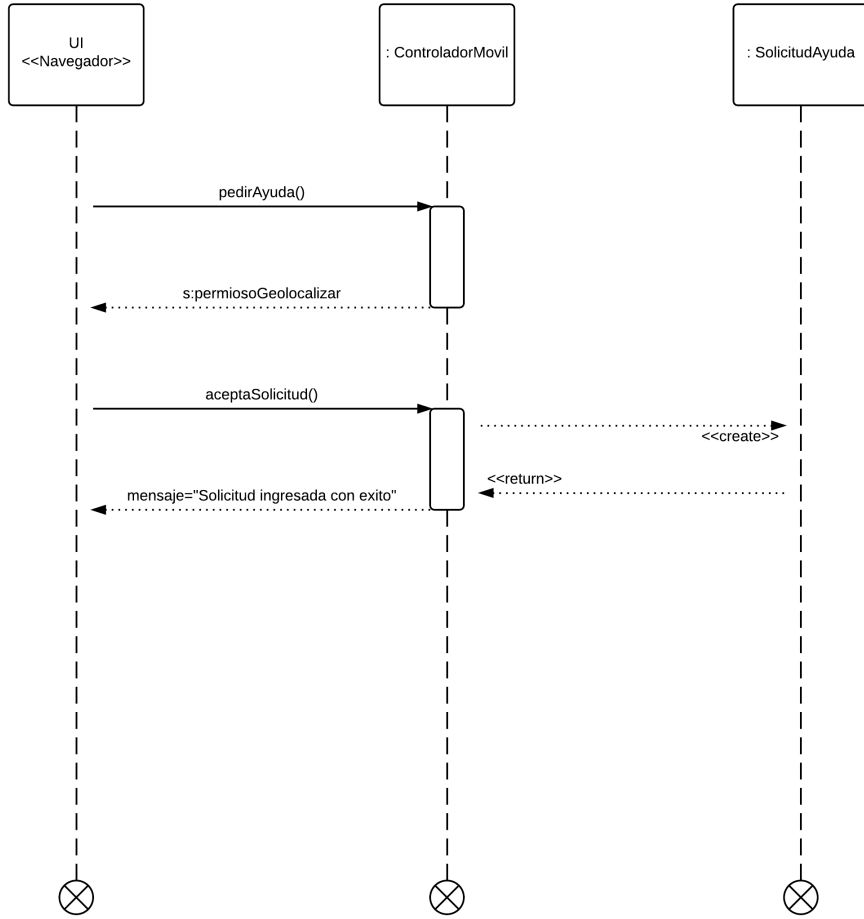
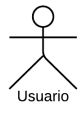


Solicitud de Ayuda

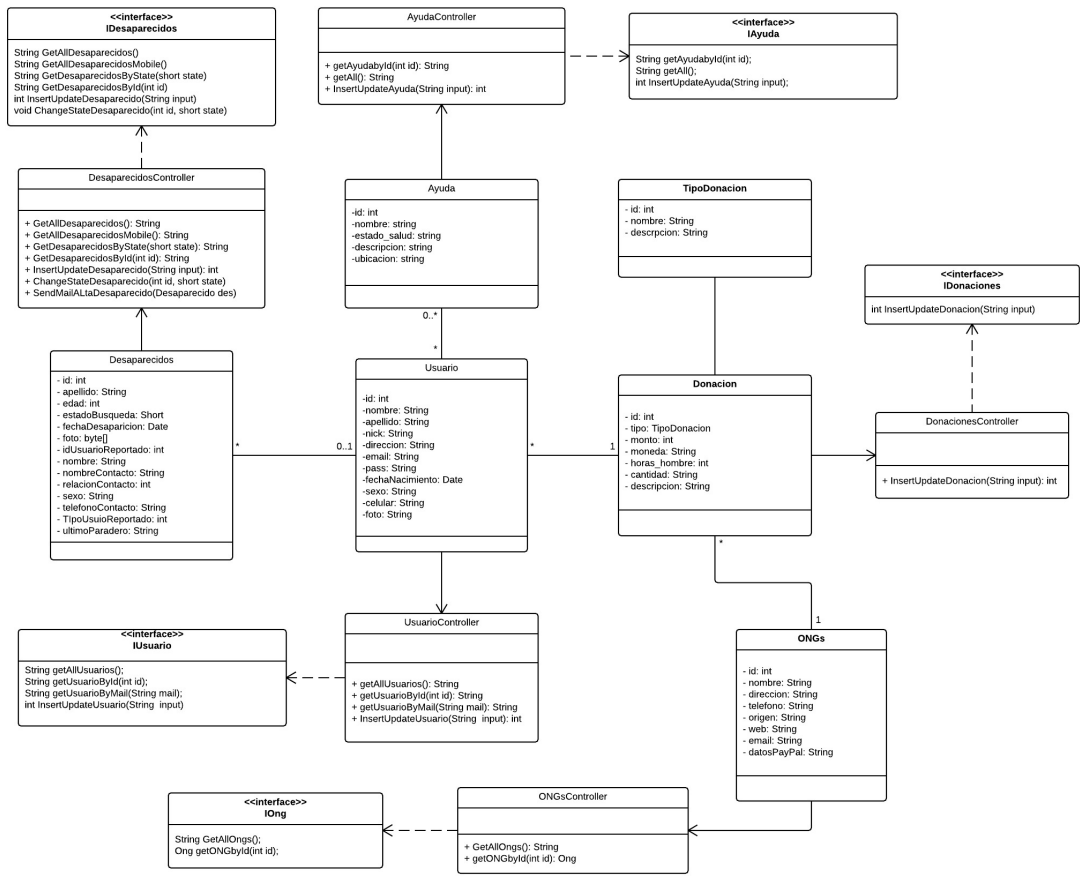
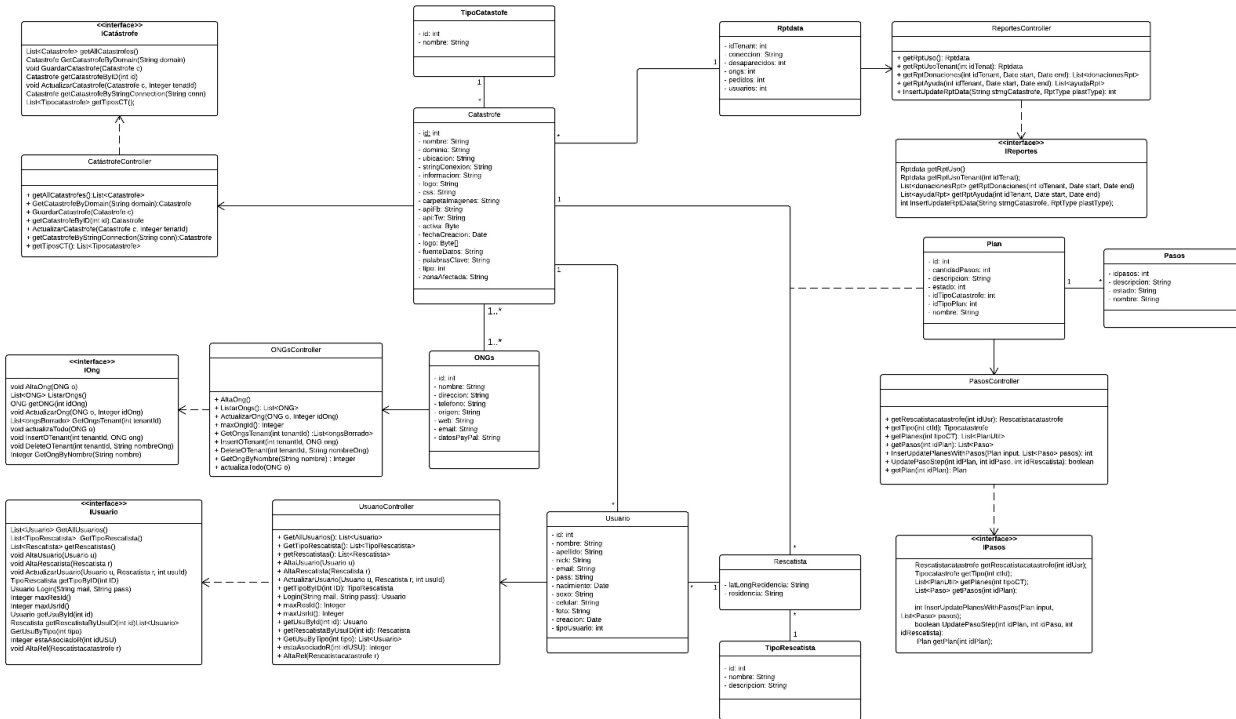
▪ Diagrama de secuencia



19.26. Botón de Pánico
 ▪ **Diagrama de secuencia**



• Diagrama de Clases de Diseño



Proyecto: S.I.C.
Modelo de Datos
Versión 3.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
21/10/2014	1.0	Versión inicial del documento	Gastón Baldenegro
23/10/2014	2.0	Nueva versión y revisión general	Ignacio Castro
27/10/2014	2.1	Ajustes de datos de desaparecidos	Yennifer Herrera
08/02/2014	2.2	Ajustes generales	Yennifer Herrera.
08/02/2014	2.3	Se actualizan diagramas.	Sandino Fratti
08/02/2015	3.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Contenido

1.	Base de Datos Compartida.....	3
1.1.	Introducción.....	3
1.2.	Modelo de entidad relacional.....	3
1.3.	Modelo de base de datos – Mapeo de clases a tablas.....	4
1.3.1.	Tabla: Usuarios.....	4
1.3.2.	Tabla: ONGs.....	5
1.3.3.	Tabla: Rescatista.....	5
1.3.4.	Tabla: TipoRescatista.....	5
1.3.5.	Tabla: Catastrofe.....	6
1.3.6.	Tabla: TipoRiesgo.....	6
1.3.7.	Tabla: GestionRiesgo.....	6
1.3.8.	Tabla: PlanEmergencia.....	7
1.3.9.	Tabla: PasosPlanEmergencia.....	7
1.3.10.	Tabla: EjecutoresPasosPlanEmergencia.....	7
1.3.11.	Tabla: Workflow.....	8
1.3.12.	Tabla: WorkFlowPasos.....	8
2.	Base de Datos Sitios Tenant.....	9
2.1.	Introducción.....	9
2.2.	Modelo de entidad relacional.....	9
2.3.	Modelo de base de datos – Mapeo de clases a tablas.....	10
2.3.1.	Tabla: Usuarios.....	11
2.3.2.	Tabla: Ayuda.....	11
2.3.3.	Tabla: Desaparecidos.....	11
2.3.4.	Tabla: ONGS.....	12
2.3.5.	Tabla: Tipo.....	12
2.3.6.	Tabla: Donaciones.....	12
3.	Características de almacenamiento.....	13

- **Base de Datos Compartida**

19.27. Introducción

Se describe el modelo que relaciona las entidades del sistema entre sí para la Base de Datos Compartida. Los atributos no serán mostrados en el siguiente modelo para lograr una mejor visualización en el documento.

19.28. Modelo de entidad relacional

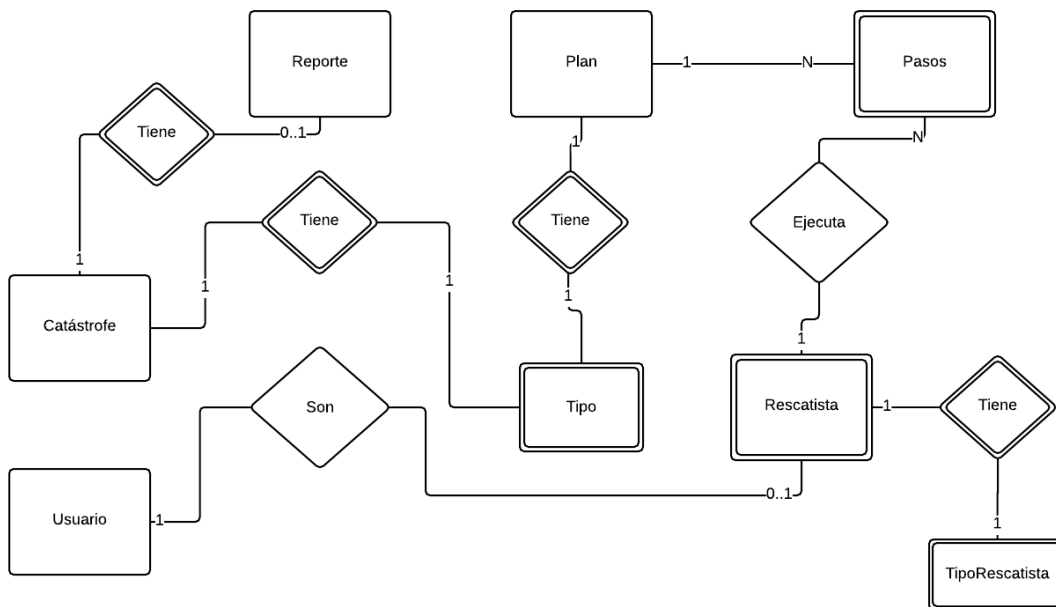


Ilustración 14 - Base de Datos Compartida

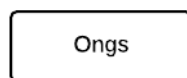


Ilustración 15 - BD Ongs BackOffice

NOTA: La tabla ONG se encuentra bajo un esquema diferente al resto de la base de datos compartida dado que no tiene relación alguna con el resto de las tablas. Ya que no es más que el almacenamiento de las *Ongs* que se podrán configurar para cada catástrofe, data que se copia parcialmente en cada base de dato específica de cada catástrofe.

19.29. Modelo de base de datos – Mapeo de clases a tablas

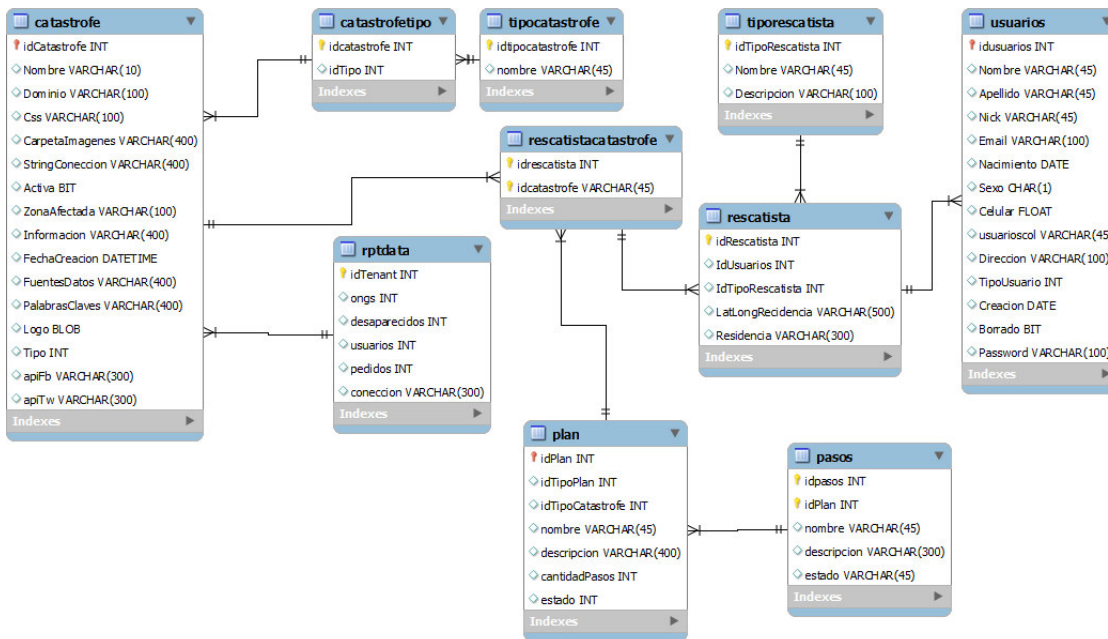


Ilustración 4 - Mapeo Clases y Tablas Base de Datos Central

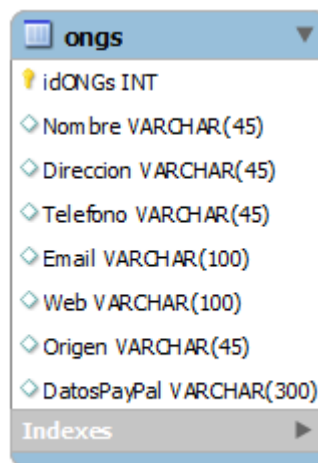


Ilustración 5 - Mapeo Clases y Tablas ONG

1.3.1 Tabla: Usuarios

1. Clase/s de Diseño Asociada/s

Mapea la información de la clase Usuarios. La misma cuenta con los siguientes campos:

- IdUsuario: identificador de la tabla.
- Nombre: nombre del usuario.
- Apellido: apellido del usuario.
- Nick: nick del usuario.
- Email: indica mail del usuario.
- Fecha Nacimiento: fecha de nacimiento del usuario.
- Sexo: indica el sexo del usuario.
- Password: contraseña para el acceso al sistema del usuario.
- Celular: teléfono móvil del usuario.
- Dirección: dirección del usuario.
- TipoUsuario: indica si el usuario es administrador o rescatista.
- Creación: fecha de alta del usuario.
- Borrado: indica si el usuario fue dado de baja del sistema.

▪ **Tabla: ONGs**

2. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase ONGs. La misma cuenta con los siguientes campos:

- IdONG: identificador de la ONG.
- Nombre: nombre de la ONG.
- Dirección: dirección de la ONG.
- Teléfono: teléfono de la ONG.
- Email: indica mail de la ONG.
- Web: indica el sitio web de la ONG.
- Origen: indica el origen de datos de la ONG.
- DatosPayPal: indica el medio de pago de la ONG.

▪ **Tabla: Rescatista**

3. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Rescatista. La misma cuenta con los siguientes campos:

- IdRescatista :Identificador del Rescatista
- IdUsuario : Identificador del Usuario
- IdTipoRescatista : Identificador del Tipo de Rescatista
- Residencia : Residencia del Rescatista
- LatLongResidencia : Ubicación del Rescatista

▪ **Tabla: TipoRescatista**

4. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase TipoRescatista. La misma cuenta con los siguientes campos:

- IdTipoRescatista: Identificador del Tipo de Rescatista
- Nombre : Nombre del Tipo de Rescatista
- Descripción : Descripción del Tipo de Rescatista

▪ **Tabla: Catastrofe**

5. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Catastrofe. La misma cuenta con los siguientes campos:

- IdCatástrofe: identificador de la catástrofe.
- Nombre: nombre de la catástrofe.
- Dominio: dominio de la catástrofe.
- Logo: logo de la catástrofe.
- CarpetaImágenes: carpeta Imágenes de la catástrofe.
- StringConexion: conexión a la BD catástrofe.
- Activa: indica si esta activa o no la catástrofe.
- FechaCreacion: fecha de creación de la catástrofe.
- ZonaAfectada: ubicación de la zona de catástrofe.
- Información: información sobre la catástrofe.
- FuentesDatos: orígenes de noticias.
- PalabrasClaves: parámetros para realizar búsquedas dentro de los orígenes de noticias.
- Tipo: tipo de catástrofe.
- apiFb: clave de la api de Facebook.
- apiTw: clave de la api de Twitter.
- Css: nombre de la hoja de estilo.

▪ **Tabla: Plan**

6. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Plan. La misma cuenta con los siguientes campos:

- idPlan: identificador del plan.
- idTipoPlan: nombre del tipo de riesgo.
- idTipoCatastrofe: identificador del tipo de catástrofe a la que corresponde el plan.

- Nombre: nombre del plan.
- Descripción: descripción del tipo de riesgo.
- cantidadPasos: cantidad de pasos que conforman el plan.
- Estado: si está completo o pendiente.

▪ **Tabla: Pasos**

7. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Pasos. La misma cuenta con los siguientes campos:

- idpasos: identificador del paso.
- idPlan: identificador del plan asociado.
- Nombre: nombre del paso.
- Descripción: descripción del paso.
- estado: indica si el paso está completo o pendiente.

▪ **Tabla: Tipocatastrofe**

8. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Tipocatastrofe. La misma cuenta con los siguientes campos:

- idtipocatastrofe: identificador del tipo de catástrofe.
- nombre: nombre del tipo de catástrofe.

1.3.9. Tabla: rescatistacatastrofe

1.3.9.1. Clase/s de Diseño Asociada/s

Mapea la información de la clase rescatistacatastrofe. La misma cuenta con los siguientes campos:

- idrescatista: identificador del rescatista asociado.
- idcatastrofe: idcatastrofe de la catástrofe asociada.

1.3.10. Tabla: raptada

1.3.10.1. Clase/s de Diseño Asociada/s

Mapea la información de la clase raptada. La misma cuenta con los siguientes campos:

- idTenant: identificador de la catástrofe asociada.
- ongs: cantidad de ongs asociadas a la catástrofe.
- desaparecidos: cantidad de desaparecidos reportados en la catástrofe.

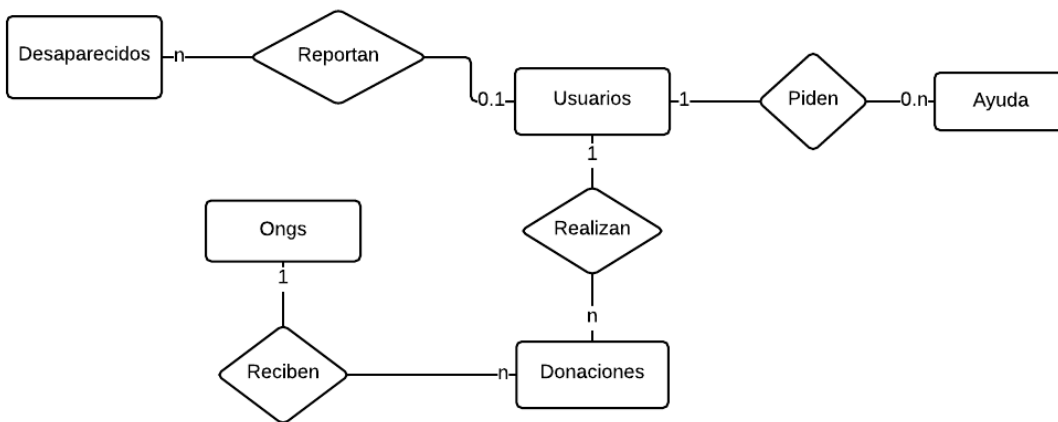
- usuarios: cantidad de usuarios registrados en el sitio correspondiente a la catástrofe.
- pedidos: cantidad de pedidos de ayuda realizados en el sitio de la catástrofe.
- coneccion: cadena de conexión de la catástrofe (Connection string).

Base de Datos Sitios Tenant

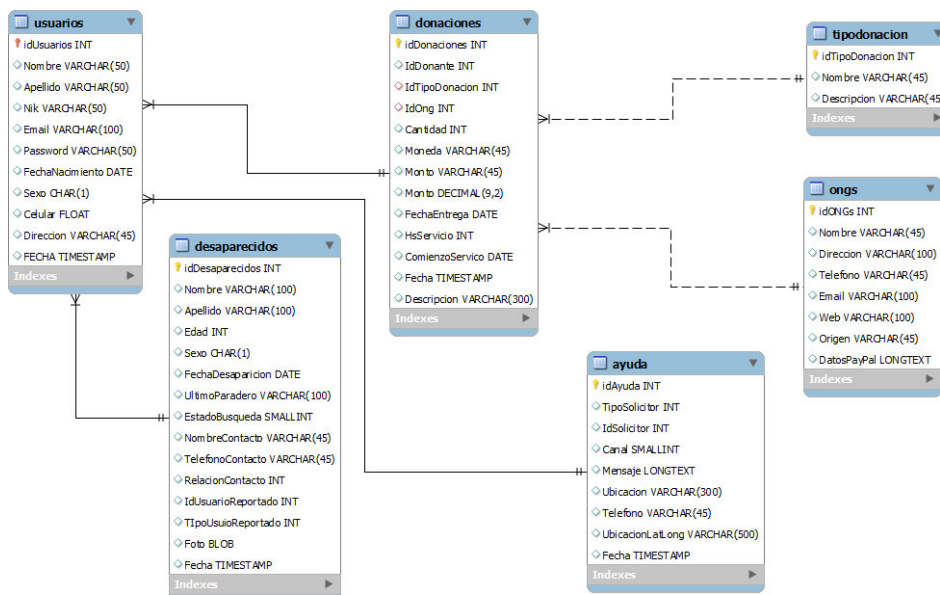
19.30. Introducción

Se describe el modelo que relaciona las entidades del sistema entre sí para la Base de Datos Sitios Tenant. Los atributos no serán mostrados en el siguiente modelo para lograr una mejor visualización en el documento.

19.31. Modelo de entidad relacional



19.32. Modelo de base de datos – Mapeo de clases a tablas



▪ **Tabla: Usuarios**

9. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Usuario. La misma cuenta con los siguientes campos:

- IdUsuario: identificador del usuario.
- Nombre: nombre del usuario.
- Apellido: apellido del usuario.
- Nick: apodo del usuario.
- Email: correo electrónico del usuario.
- Password: clave del usuario.
- FechaNacimiento: fecha de Nacimiento.
- Sexo: sexo (Masculino /Femenino).
- Celular: celular del usuario.
- Dirección: dirección del usuario.
- Fecha: fecha de alta del usuario.

▪ **Tabla: Ayuda**

10. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Ayuda. La misma cuenta con los siguientes campos:

- IdAyuda: identificador del Usuario.
- TipoSolicitor: Tipo de Solicitor.
- IdSolicitor: identificador del Solicitor.
- Canal: Tipo de canal.
- Mensaje: Mensaje.
- UbicacionLatLong: coordenadas de la ubicación.
- Ubicación: Datos de la Ubicación.
- Telefono: teléfono especificado.
- Fecha: fecha de realización.

▪ **Tabla: Desaparecidos**

11. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Desaparecidos. La misma cuenta con los siguientes campos:

- IdDesaparecido: Identificador del Desaparecido.
- Nombre: Nombre del Desaparecido.
- Apellido: Apellido del Desaparecido.
- Edad: Edad del Desaparecido.
- Sexo: Sexo del Desaparecido.
- Imagen: foto más reciente del desaparecido.
- Descripción: Descripción física del desaparecido.
- FechaDesaparicion: Fecha de la Desaparición.
- UltimoParadero: Ultima ubicación del Desparecido.

- UltimoParaderoLatLong: Datos Ultima Ubicación del Desparecido.
- EstadoBusqueda : Estado de la Búsqueda
- NombreContacto : Nombre de Contacto
- TelefonoContacto : Teléfono de Contacto
- RelacionContacto : Tipo de Relación del Contacto
- IdUsuarioReportado : Identificador del Usuario Reportado
- TIpoUsuioReportado: Tipo de Usuario Reportado
- Foto: foto adjunta.
- Fecha: fecha reportado.

▪ **Tabla: ONGS**

12. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase ONGs. La misma cuenta con los siguientes campos:

- IdONG : Identificador de la ONG
- Nombre : Nombre de la ONG
- Direccion : Dirección de la ONG
- Telefono : Teléfono de la ONG
- Email : email de la ONG
- Web : Sitio web de la ONG
- Origen : Origen ONG
- DatosPayPal : Datos PayPal de la ONG

▪ **Tabla: Tipo**

13. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Tipo. La misma cuenta con los siguientes campos:

- IdTipoDonacion : Identificador del Tipo de Donación
- Nombre : Nombre del Tipo de Donación
- Descripción : Descripción del Tipo de Donación

▪ **Tabla: Donaciones**

14. *Clase/s de Diseño Asociada/s*

Mapea la información de la clase Donaciones. La misma cuenta con los siguientes campos:

- IdDonacion :Identificador de la Donación
- IdDonante : Identificador del Usuario Donante
- IdOng : Identificador de la ONG
- IdTipoDonacion : Identificador del tipo de ONG
- Cantidad : Cantidad de la Donación
- Moneda : Moneda de la Donación
- Monto : Monto de la Donación
- Fecha Entrega : Fecha de la Donación

- HsServicio : Horas de Servicio
- ComienzoServicio : Comienzo del Servicio
- FInalizacionServicio : Fin del Servicio

- **Características de almacenamiento**

Dado que nuestra aplicación es multi-tenant con bases de datos distribuidas una por cada tenant, será fácil extender la aplicación del modelo de datos para satisfacer las necesidades de los clientes individuales, y la restauración de datos de un cliente de copias de seguridad en caso de un fracaso, es un procedimiento relativamente simple. Este enfoque proporciona aislamiento máximo de datos de clientes.

ANEXO [I]

Proyecto: SIC

Entrevista SINAE

Versión 2.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
05/02/2015	1.0	Primera Versión del Documento	Juan Ignacio Castro
08/02/2015	2.0	Revisión General del Documento	Gastón Baldenegro, Horacio García, Ignacio Castro, Sandino Fratti, Yennifer Herrera

Entrevista realizada a personal de SINAE

En resumen: El personal del SINAE está más interesado en un sistema de gestión de información que en un sistema de gestión de riesgos.

Puntos importantes relevados a partir de la entrevista:

1. Para ellos les es importante el manejo de información. Por ejemplo cual es la catástrofe, zonas dañadas y las recurrencias de éstas, cuántas personas se vieron afectadas por la misma. De estos datos no solo les importa el presente sino que también es muy importante los registros pasados, porque de esta forma ellos pueden tomar medidas y prevenciones a futuro.
2. El SINAE no gestiona pedidos de ayuda o cualquier solicitud directa de la población.
3. Ante un incidente, primero se toman medidas locales hay una organización departamental que gestiona las primeras acciones, siempre informando al SINAE de los acontecimientos. Constantemente se informa el número de personas dañadas, zonas afectadas, etc. El SINAE evalúa si es necesario tomar a su cargo la situación, y también lo hace en el caso de que la zona departamental se vea desbordada, la cual va a pedir apoyo al SINAE. Las población debe recurrir a los medios de ayuda conocidos como ser bomberos, policía, policlínica, teléfono de emergencias 911.
4. Por ejemplo ante llamados a las emergencias mencionadas en el punto 3, estos medios son los que deciden si es un acontecimiento desmedido y tienen que solicitar asistencia a la junta departamental y la misma resuelve si debe pedir ayuda al SINAE.
5. El SINAE tiene varios cuerpos con los que trabaja en conjunto y toman decisiones uno de ellos por ejemplo está formado por los militares que generalmente son los que se encargan del trabajo más de campo, designar personal, maquinaria, etc.
6. El sector técnico es quien se encarga de la organización en casos de emergencias y del flujo de incidentes, de lo que es la gestión del riesgo y el plan de emergencia o contingencia.
7. El SINAE en si es el organismo que tiene el poder legal de dar una alerta oficial, y de gestionar otros recursos u organizaciones estatales para la contingencia de las emergencias.
8. A su vez también son el organismo nacional que puede decidir cortar suministros en determinada zona para abastecer a otra, por ejemplo de faltar energía en un barrio el SINAE tiene la potestad para decidir cortar la energía en otro barrio y abastecer al primero de energía.
9. El SINAE puede y debe actuar ante cualquier caso masivo de alerta de cualquier tipo, como puede ser emergencias climáticas, inundaciones, incendios, siniestros sanitarios y de servicios esenciales como lo son el agua, la luz o las comunicaciones.
10. Actualmente los funcionarios se manejan en papel y la comunicación se realiza a través de los aparatos telefónicos, no utilizan ningún tipo de la tecnología móvil. Lo único que poseen informatizado es el sitio web, un twitter, y poseen un sistema de geo localización que no utilizan porque no les sirve para tomar decisiones. En este momento se está desarrollando un sistema con geo localización que es una base de datos histórica desde los años 30 en el que se ven reflejado todos los incidentes que hubo por zonas, período y tipo. Lo cual ellos ven como una gran herramienta para prevenir y saber cómo actuar antes una emergencia.

11. El SINAE no lleva control alguno sobre las personas desaparecidas porque según ellos no es una realidad para Uruguay, si manejan las personas evacuadas y auto evacuadas. Y les resulta interesante el tener algún tipo de herramienta para poder contabilizar a las personas de forma más cabal.
12. Con respecto a las donaciones ellos no las manejan, las mismas son recaudadas por las ONG, y en general se dona lo que la zona departamental solicite o indique que es lo que necesita. No hay donaciones de cosas que no se precisen en esa situación o que no ha sido solicitada, dado que genera caos, problemas políticos y administrativos. Al día de hoy que existe una inundación en el departamento de Durazno, si la zona departamental dice que necesita ropa y alimentos, únicamente se va a poder donar ropa y alimentos, y tampoco más cantidad de lo solicitado por la zona departamental.
13. En definitiva la zona departamental le indica a las ONG que necesitan y las ONG se encargan de las campañas para recolectar o recaudar esos objetos solicitados.
14. Nos dio a entender que no les interesa una aplicación en la que participe la población en forma masiva, sino que ellos prefieren una herramienta de gestión y comunicación interna entre el SINAE, la junta departamental y cada uno de los entes de ayuda en la situación (bomberos, policía o médicos, por ejemplo).
15. Para que la preservación o almacenamiento de datos sea más completa y detallada, él ve de gran utilidad una herramienta en la cual la gente de campo pueda registrar mediante dispositivos móviles las zonas afectadas, es decir determinar el lugar y el radio afectado. La probabilidad de personas afectadas, y utilizando el dispositivo capturar fotos por ejemplo. Para comenzar una primera etapa de informe del incidente.
16. Ellos tienen unos flujos de acciones a tomar, y varios protocolos definidos.