

UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE INGENIERÍA

Departamento de Ingeniería Informática



**Desarrollo de prototipo de sistema informático para la organización y
coordinación de organismos competentes y voluntariado frente a
catástrofes**

Diego Enrique Olavarría Úbeda

Profesor Guía: Carolina Alejandra Bonacic Castro

Trabajo de titulación presentado en conformidad con los
requisitos para obtener el título de Ingeniero en Ejecución
en Computación e Informática.

Santiago, Chile

2016

RESUMEN

El presente documento corresponde a una propuesta de aplicación, la cual tomará forma bajo el nombre de **VoluntariS**, y pretende ser una solución efectiva a un problema encontrado en la gestión de voluntariado en situaciones de catástrofe: la desorganización de los grupos voluntarios. Este proyecto se gestó a base de uno de los problemas a resolver presentados en las propuestas de desarrollo de tecnología para ciudades inteligentes, así como de las experiencias de los voluntarios presentes en el incendio de Valparaíso del 2014.

VoluntariS está diseñado como una aplicación móvil, que permite a múltiples coordinadores contactar a múltiples grupos de personas. El prototipo fue programado en **HTML** para la capa del cliente, **Javascript** para la capa de funciones, y **NoSQL** para manejar la base de datos. Se optó por utilizar la metodología de desarrollo evolutivo con un proceso exploratorio basado en prototipos. Esto para reducir la repetición del trabajo en el proceso de desarrollo.

VoluntariS demuestra las posibilidades que posee el uso de los dispositivos móviles como herramientas de distribución de masas en distintos puntos geográficos. Busca sostenerse por sí mismo como aplicación móvil de gran utilidad en la organización y distribución de esfuerzos de voluntariado, y también ser incorporado junto a los avances realizados por el equipo de **Faltan Manos**¹ para la conformación de una aplicación de mayor alcance.

Palabras clave: Aplicación Móvil, Voluntariado, Modelo de Desarrollo Evolutivo, Prototipo de apoyo para la gestión de voluntariado.

Este trabajo es parte del Proyecto Fondef [ID15110560](#), "Plataforma de apoyo a la gestión de emergencia y aplicaciones."

¹ <http://faltanmanos.org/>

DEDICATORIA

Dedicado a la actual ciencia, que persigue intereses económicos y no humanitarios.

AGRADECIMIENTOS

En primer lugar, como corresponde, agradezco a mis padres Raúl Olavarría Mac-Taggart y Lorena Úbeda Zurita, guías incuestionables y los mejores padres que alguien podría merecer. Cada palabra fue un apoyo y una vuelta al camino del éxito personal.

A José Farías, cuya amistad de años ha significado un apoyo irreemplazable y una sonrisa eterna.

A Herman Barraza, hermano de armas con quien vencimos a la universidad, y quien ha sido un gran apoyo dentro y fuera de ésta.

A María Teresa Flores, quien fue una luz en muchos periodos de oscuridad. Que este enorme trabajo signifique para ella que si yo pude llegar a este punto, ella puede 10 veces.

A Javiera y Tomás Olavarría, mis primos, quienes siempre supieron cómo hacerme sentir bienvenido y acompañado.

A mi profesor guía Carolina Bonacic, que siempre me recibió con una sonrisa y buena voluntad para resolver mis dudas, y corrigió y ayudó a redactar este documento.

Al equipo de **Faltan Manos**, Federico Jensen, Cristian Estrada y Nicolás Aranda por su excelente voluntad de asistencia a la ciudadanía. Son una inspiración y un ejemplo para la juventud.

A todas aquellas personas que buscan el bien para el prójimo, y especialmente para aquellos que me apoyaron todo este tiempo.

TABLA DE CONTENIDO

CAPÍTULO 1. INTRODUCCIÓN	1
1.1 ANTECEDENTES Y MOTIVACIÓN.....	1
1.2 DESCRIPCIÓN DEL PROBLEMA	2
1.3 SOLUCIÓN PROPUESTA	3
1.4 OBJETIVOS Y ALCANCES DEL PROYECTO	4
1.4.1 Objetivo general	4
1.4.2 Objetivos específicos	4
1.4.3 Alcances	4
1.5 ORGANIZACIÓN DEL DOCUMENTO	5
CAPÍTULO 2. CONCEPTOS.....	7
2.1 MARCO TEÓRICO	7
2.1.1 Sobre las emergencias en Chile y el mundo	7
2.1.2 Participación de la comunidad	9
2.1.3 Voluntariado en Chile	10
2.1.4 Herramientas existentes	11
2.2 ESTUDIO PREVIO.....	17
2.2.1 Respuesta ciudadana y gubernamental	17
2.2.2 Participación de voluntariados en Chile	19
CAPÍTULO 3. DESARROLLO DEL SOFTWARE	21
3.1 METODOLOGÍA.....	21
3.1.1 Desarrollo basado en prototipado	21
3.2 HERRAMIENTAS UTILIZADAS.....	23
3.3 MODELO DE NEGOCIOS	23
3.4 ELABORACIÓN	24
3.4.1 Análisis y diseño de la elaboración	24
3.5 REQUERIMIENTOS DEL SISTEMA.....	26

3.5.1	Requerimientos funcionales.....	26
3.5.2	Requerimientos no funcionales.....	31
3.6	DIAGRAMA DE PAQUETES	33
3.7	DIAGRAMAS DE CASOS DE USO	33
3.7.1	Gestión de Perfil.....	33
3.7.2	Gestión de puntos de trabajo	34
3.8	ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS.....	35
3.8.1	Primera Iteración	35
	Prototipo Usuario, 1ra Iteración	36
	Prototipo Coordinador, 1ra Iteración	38
3.8.2	Segunda Iteración	40
	Prototipo Usuario, 2da Iteración.....	40
	Prototipo Coordinador, 2da Iteración	41
3.8.3	Tercera Iteración	43
	Prototipo Usuario, 3ra Iteración	43
3.8.4	Cuarta Iteración.....	44
	Prototipo Usuario, 4ta Iteración.....	45
	Prototipo Coordinador, 4ta Iteración	45
3.8.5	Quinta Iteración	46
	Prototipo Usuario, 5ta Iteración.....	47
	Prototipo Coordinador, 5ta Iteración	49
3.9	DISEÑO DEL SOFTWARE	51
3.9.1	Base de datos	51
3.9.2	Modelo de datos.....	53
3.10	DIAGRAMAS DE ARQUITECTURA	58
3.10.1	Diagrama de clases	58
3.10.2	Diagrama de componentes	60

3.10.3	Diagramas de secuencia.....	61
CAPÍTULO 4.	IMPLEMENTACIÓN.....	62
4.1	ALGORITMO.....	62
4.2	CONSTRUCCIÓN DEL SOFTWARE	68
4.2.1	Interfaz	68
4.2.2	Prototipo Voluntario.....	69
4.2.3	Prototipo Coordinador	77
4.2.4	Proceso de selección	82
4.2.5	Proceso de finalización de una actividad	90
CAPÍTULO 5.	EVALUACIÓN Y RESULTADOS	93
5.1	Evaluación.....	93
5.1.1	Evaluación pruebas unitarias	93
5.1.2	Evaluación pruebas de integración	96
5.1.3	Evaluación pruebas de validación.....	98
5.1.4	Evaluación pruebas de sistema	99
5.1.5	Evaluación pruebas de aceptación	100
5.2	Resultados	101
CAPÍTULO 6.	DISCUSIÓN Y CONCLUSIONES.....	106
CAPÍTULO 7.	BIBLIOGRAFÍA	109
CAPÍTULO 8.	ANEXOS	113
8.1	Diagramas de Secuencia	113
8.2	Algoritmo	131
8.3	Pruebas	135
8.3.1	Pruebas unitarias	135
8.3.2	Pruebas de integración	140
8.3.3	Pruebas de validación	143
8.4	Resultados	147

ÍNDICE DE CUADROS Y/O TABLAS

Tabla 2-1: Tipos de desastres (EM-DAT, 2009)	7
Tabla 3-1: Requerimientos Funcionales del prototipo Voluntario (RFV)	27
Tabla 3-2: Requerimientos Funcionales del prototipo Coordinador (RFC)	28
Tabla 3-3: Requerimientos Funcionales para la Primera Iteración.....	29
Tabla 3-4: Requerimientos Funcionales Segunda Iteración.....	29
Tabla 3-5: Requerimientos Funcionales para la Tercera Iteración.....	30
Tabla 3-6: Requerimientos Funcionales para la Cuarta Iteración	30
Tabla 3-7: Requerimientos Funcionales para la Quinta Generación.....	31
Tabla 3-8: Requerimientos No Funcionales.....	31
Tabla 3-9: Comparación de bases de datos (Amazon, 2016)	51
Tabla 3-10: Atributos tabla Voluntario.....	55
Tabla 3-11: Atributos tabla Coordinador	55
Tabla 3-12: Atributos tabla Grupo.....	56
Tabla 3-13: Atributos tabla Punto	56
Tabla 3-14: Atributos tabla Postulación	57
Tabla 3-15: Atributos tabla Mensaje	57
Tabla 4-1: Itinerario modelo	63
Tabla 4-2: Ejemplo Itinerario 1	64
Tabla 4-3: Ejemplo Itinerario 2, Parte 1	66
Tabla 4-4: Ejemplo Itinerario 2, Parte 2	67
Tabla 5-1: Pruebas unitarias: Prototipo Voluntario	93
Tabla 5-2: Pruebas unitarias: Prototipo Coordinador	94
Tabla 5-3: Resumen resultados de pruebas unitarias	95
Tabla 5-4: Resultado pruebas unitarias generales: prototipo Voluntario.....	95
Tabla 5-5: Plan de prueba: Perfil Voluntario	97
Tabla 5-6: Plan de prueba: Perfil Coordinador	98
Tabla 5-7: Caso de prueba para la creación de un punto de trabajo	99
Tabla 5-8: Prueba Beta 01	100
Tabla 5-9: Prueba Beta 02.....	100
Tabla 5-10: Prueba Beta 03.....	101
Tabla 5-11: Datos de punto generado	101
Tabla 5-12: Datos de usuarios registrados	102
Tabla 5-13: Datos de postulaciones creadas por voluntarios.....	103

Tabla 5-14: Datos de mensajes resultantes de postulaciones	104
Tabla 5-15: Itinerario solución	105
Tabla 8-1: Pruebas unitarias, tabla Voluntario	135
Tabla 8-2: Pruebas unitarias, tabla Coordinador	135
Tabla 8-3: Pruebas unitarias, tabla Punto	135
Tabla 8-4: Pruebas unitarias, tabla Postulación	136
Tabla 8-5: Pruebas unitarias, tabla Grupo	136
Tabla 8-6: Pruebas unitarias, tabla Mensaje	136
Tabla 8-7: Resultado pruebas unitarias generales: prototipo Coordinador	137
Tabla 8-8: Resultado pruebas generales: tabla Voluntario	138
Tabla 8-9: Resultado pruebas generales: tabla Coordinador	138
Tabla 8-10: Resultado pruebas generales: tabla Punto	138
Tabla 8-11: Resultado pruebas generales: tabla Postulación	139
Tabla 8-12: Resultado pruebas generales: tabla Grupo	139
Tabla 8-13: Resultado pruebas generales: tabla Mensaje	140
Tabla 8-14: Plan de prueba: Crear Punto	140
Tabla 8-15: Plan de prueba: Crear Postulación	141
Tabla 8-16: Plan de prueba: Iniciar trabajos en un Punto	141
Tabla 8-17: Plan de prueba: Terminar trabajos en un Punto	142
Tabla 8-18: Plan de prueba: Crear Grupo	142
Tabla 8-19: Caso de prueba para la creación de una postulación	143
Tabla 8-20: Caso de prueba para iniciar el llamado de un punto de trabajo	143
Tabla 8-21: Caso de prueba para finalizar la actividad de un punto de trabajo	144
Tabla 8-22: Caso de prueba para la creación de un grupo	145
Tabla 8-23: Caso de prueba para el manejo del perfil del voluntario	145
Tabla 8-24: Caso de prueba para el manejo del perfil del coordinador	146
Tabla 8-25: Plan de pruebas prototipo Voluntario	147
Tabla 8-26: Plan de pruebas prototipo Coordinador	149

ÍNDICE DE ILUSTRACIONES

Ilustración 2-1: MicroMappers (iRevolutions, 2014)	12
Ilustración 2-2: Ushahidi (Ushaguzi, 2013)	12
Ilustración 2-3: Eden (Demo) (Sahana Software Foundation, 2004)	13
Ilustración 2-4: VolunteerLocal (VolunteerLocal, 2009)	14
Ilustración 2-5: Google Maps API (Google, 2016)	16
Ilustración 2-6: NUKEMAP (Wellerstein, 2014)	16
Ilustración 2-7: Gráfico de participación de voluntariado	20
Ilustración 2-8: Gráfico de razones de no-participación de voluntariado	20
Ilustración 3-1: Etapas de una iteración	22
Ilustración 3-2: Diagrama de paquetes	33
Ilustración 3-3: Diagrama de casos de uso: Gestión de perfil	34
Ilustración 3-4: Diagrama de casos de uso: Gestión de puntos de trabajo	35
Ilustración 3-5: Iteración Voluntario 1: Vista de Login y Vista de Registro	36
Ilustración 3-6: Iteración Voluntario 1: Vista Inicio	37
Ilustración 3-7: Iteración Voluntario 1: Vista Menú, Vista Grupo y Vista Perfil	38
Ilustración 3-8: Iteración Coordinador 1: Vista de Login y Vista de Registro	39
Ilustración 3-9: Iteración Coordinador 1: Vista Menú, Vista Inicio y Vista Perfil	40
Ilustración 3-10: Iteración Voluntario 2: Vista de Registro	41
Ilustración 3-11: Iteración Coordinador 2: Vista Inicio	42
Ilustración 3-12: Iteración Coordinador 2: Vista Menú y Vista Actividades	43
Ilustración 3-13: Iteración Voluntario 3: Vista Inicio	44
Ilustración 3-14: Iteración Voluntario 4: Vista Inicio, pestaña Misiones	45
Ilustración 3-15: Iteración Coordinador 4: Vista Inicio y Vista Punto	46
Ilustración 3-16: Iteración Voluntario 5: Vista Inicio/Llamado	47
Ilustración 3-17: Iteración Voluntario 5: Vista Grupo	48
Ilustración 3-18: Iteración Voluntario 5: Vista Perfil	49
Ilustración 3-19: Iteración Coordinador 4: Vista Punto	49
Ilustración 3-20: Iteración Coordinador 5: Vista Grupos	50
Ilustración 3-21: Iteración Coordinador 5: Vista Actividades	50
Ilustración 3-22: Modelo de datos	54
Ilustración 3-23: Diagrama de clases Voluntario	59
Ilustración 3-24: Diagrama de clases Coordinador	60
Ilustración 3-25: Diagrama de componentes	61

Ilustración 4-1: Vista Inicio de sesión Voluntario	70
Ilustración 4-2: Vista Registro nuevo Voluntario	71
Ilustración 4-3: Ejemplo Registro nuevo Voluntario	71
Ilustración 4-4: Registro exitoso de Voluntario	72
Ilustración 4-5: Ejemplo Inicio de sesión/Inicio de sesión exitoso de Voluntario	73
Ilustración 4-6: Vista Inicio Voluntario, pestaña Misiones	73
Ilustración 4-7: Vista Inicio Voluntario, pestaña Mensajes	74
Ilustración 4-8: Vista Menú Voluntario	74
Ilustración 4-9: Vista Grupo de trabajo Voluntario	75
Ilustración 4-10: Escoger nuevo grupo Voluntario	76
Ilustración 4-11: Vista Perfil Voluntario	76
Ilustración 4-12: Vista Inicio Coordinador	78
Ilustración 4-13: Vista Menú Coordinador	78
Ilustración 4-14: Vista Gestión de Actividades Coordinador	79
Ilustración 4-15: Vista Gestión de Grupos Coordinador	80
Ilustración 4-16: Vista nuevo Grupo creado Coordinador	80
Ilustración 4-17: Grupos actualizados/Grupo ejemplo eliminado Coordinador	81
Ilustración 4-18: Vista Perfil Coordinador	81
Ilustración 4-19: Coordinador - creando punto de trabajo	82
Ilustración 4-20: Coordinador - Nuevo punto creado	83
Ilustración 4-21: Coordinador: Listado de actividades	83
Ilustración 4-22: Voluntario - Escogiendo punto de trabajo	84
Ilustración 4-23: Voluntario - Escogiendo fechas de trabajo	85
Ilustración 4-24: Voluntario - Éxito en postulación	86
Ilustración 4-25: Coordinador - Revisión de punto de trabajo/Vista punto de trabajo	86
Ilustración 4-26: Coordinador - Notificar postulantes	87
Ilustración 4-27: Coordinador - Resultados en correo electrónico	87
Ilustración 4-28: Voluntario – Notificaciones éxito/no éxito	88
Ilustración 4-29: Voluntario – Llamado a trabajar/Mensaje de llamado	89
Ilustración 4-30: Voluntario - Detalles y cálculo de ruta	89
Ilustración 4-31: Coordinador - Evaluación de voluntarios	90
Ilustración 4-32: Voluntario - Ranking actualizado	91
Ilustración 4-33: Coordinador - Actividad terminada	91
Ilustración 4-34: Coordinador - Eliminando punto	92
Ilustración 4-35: Coordinador - Punto eliminado	92

Ilustración 5-1: Diagrama Down-Top, módulos para las pruebas de integración	97
Ilustración 8-1: Diagrama de secuencia Visualizando Perfil Voluntario	113
Ilustración 8-2: Diagrama de secuencia Visualizando Perfil Coordinador	114
Ilustración 8-3: Diagrama de secuencia Agregando información a perfil Voluntario	115
Ilustración 8-4: Diagrama de secuencia Agregando información a perfil Coordinador	116
Ilustración 8-5: Diagrama de secuencia Editando grupo de trabajo	117
Ilustración 8-6: Diagrama de secuencia Visualizando grupo de trabajo.....	118
Ilustración 8-7: Diagrama de secuencia Visualizando grupos de trabajo/Eliminando grupo de trabajo	119
Ilustración 8-8: Diagrama de secuencia Creando grupo de trabajo	120
Ilustración 8-9: Diagrama de secuencia Creando punto de trabajo	121
Ilustración 8-10: Diagrama de secuencia Eliminando punto de trabajo	123
Ilustración 8-11: Diagrama de secuencia Visualizando actividades activas.....	124
Ilustración 8-12: Diagrama de secuencia Postulando a actividad	125
Ilustración 8-13: Diagrama de secuencia Llamando a postulantes	126
Ilustración 8-14: Diagrama de secuencia Visualizando resultado de postulación	128
Ilustración 8-15: Diagrama de secuencia Finalizando actividad	130

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

En la búsqueda de problemáticas existentes a nivel país, hay una que, pese al gran movimiento ciudadano que evoca, no posee un gran apoyo tecnológico ni utiliza herramientas de gestión: La coordinación eficaz de las personas que desean ayudar en eventos de catástrofes. Es sabido como este país enfrenta continuas emergencias ambientales (Biblioteca Nacional de Chile, 2014). En las páginas del Instituto Nacional de Estadísticas podemos encontrar algunos datos:

1960-2016	31 Terremotos (sobre 7° Richter)
2010-2014	71 Tsunamis
2010-2014	190 Temporales
2010-2014	25 Actividad volcánica (algunos duran todo el año)
2010-2014	22446 Incendios forestales (INE, 2015)

Sismos, incendios, inundaciones o aludes son un ejemplo de los acontecimientos que afectan al país, y obliga a que existan distintos organismos de respuesta para asegurar el bienestar de la ciudadanía. Un ejemplo de éstos es **AMSCA**, o **Agrupación Médica Social Chile Ayuda**, la que nació en Rancagua como un colectivo de voluntariado de profesionales en respuesta al incendio de Valparaíso del año 2014, prestando atención y socorro a cientos de afectados (AMSCA, 2015).

Además de los organismos gubernamentales pertinentes, es el mismo pueblo chileno quien forma gran parte de la recuperación y las labores de reconstrucción en zonas de catástrofe, al destacar en su capacidad de solidarizar con el prójimo en casos de necesidad.

La posibilidad de diseñar aplicaciones informáticas capaces de facilitar, canalizar y organizar labores de ayuda social permitiría ampliar la participación de la ciudadanía cuando muchas manos sean necesarias, y así, colaborar también en la formación de una sociedad consciente y preparada.

Permitir que la ciudadanía participe en tales labores implica poder ser capaces de distribuir y jerarquizar sus esfuerzos y habilidades, a través de un flujo de información simple,

oportuno y claro para las personas que desean ayudar, evitando el desorden y el caos, y privilegiando los esfuerzos por colaborar en resultados positivos, efectivos y concretos.

1.2 DESCRIPCIÓN DEL PROBLEMA

En nuestro país existen medidas que ayudan en la prevención y en la mitigación de amenazas naturales. Pero debe tomarse especial atención a que Chile no posee una herramienta a nivel país que permita la coordinación de las organizaciones voluntarias existentes, sino que sólo las congrega. Actualmente, la **INJUV**, o **Instituto Nacional de la Juventud**, toma ese papel (América Solidaria, s.f.).

La **INJUV** posee un programa llamado **Red Nacional de Voluntariado Juvenil**, donde los grupos de voluntariado pueden registrarse y participar activamente. Cabe decir que, dada una emergencia, estos grupos siguen trabajando independientemente. Por ejemplo, durante el incendio de Valparaíso, aproximadamente 15.000 voluntarios llegaron al lugar del siniestro a realizar labores de reconstrucción, pero se presentaron problemas de descoordinación como voluntarios ociosos al no existir una central que asigne las tareas, entrega de suministros a falsos damnificados, y una excesiva cantidad de personas que decidieron subir por su propio criterio a los cerros a ayudar debido a la falta de información por parte de las autoridades en el lugar del desastre, pero que ocasionó estancamientos en las labores de asistencia inclusive por parte de organizaciones como la **Policía De Investigaciones**, **Carabineros de Chile**, **Bomberos**, ambulancias, **Cruz Roja**, etc. (Diario Digital ElAconcagua, 2015). Esto da cuenta de que, si bien existen canales de comunicación para convocar a cantidades más que suficientes de asistentes, no significa que se logre que éstos vayan a realizar automáticamente un trabajo pleno, debido a las consecuencias de intentar manejar grandes cantidades de personas sin un plan de trabajo que las englobe tanto en lo general como a cada una dentro de sus capacidades. La **Red Nacional de Voluntariado Juvenil** cumple en parte este propósito, pero no realiza las tareas de gestionar las tareas que deban realizarse, ni asignar estas tareas a los grupos voluntarios apropiados.

Sin tomar en cuenta los problemas directos de una situación de desastre, los cuales son previstos por las organizaciones existentes en este país y en el resto del mundo, queda ayudar a evitar y solucionar los problemas ocasionados por la participación humana en las distintas etapas de un desastre. Pese a los muchos beneficios de que existan personas y organizaciones que estén dispuestas a prestar ayuda, aparecen también inconvenientes con respecto a su participación. Dado que no se tiene una comunicación efectiva entre los distintos

equipos de voluntariado una vez que ocurrió una catástrofe, todas estas organizaciones se presentan a ayudar sin coordinar sus esfuerzos entre sí, formándose un desorden que implica utilizar tiempo muchas veces valioso en sólo organizar las tareas que cada voluntario realizará. Al mismo tiempo, es posible también que no se logre reunir una cantidad suficiente de gente dispuesta a realizar trabajos. También debe considerarse que no todas las zonas de desastre son apropiadas para el trabajo de voluntarios, necesitándose solo expertos en las áreas requeridas de auxilio y asistencia.

Como se ha mencionado, existen medidas que ayudan a la creación de planes para el actuar de la comunidad durante y después de un desastre. Los gobiernos y las organizaciones de asistencia cumplen su propósito de mitigar los daños humanos e infraestructurales de la población, y esto se logra porque se debió crear redes de comunicación entre estas entidades para asegurar completa cooperación sin estorbar sus esfuerzos entre sí, pero no existe un sistema de comunicación directa entre los equipos de voluntariado y organizaciones que proporcionen la coordinación necesaria para un trabajo óptimo.

Los problemas mencionados ocurren por la escasa coordinación y comunicación entre los actores y entidades claves, que puede ser solucionado en parte por una herramienta informática que ayude a gestionar el proceso.

1.3 SOLUCIÓN PROPUESTA

El proyecto de título consiste en desarrollar un producto tecnológico presentado a una entidad organizadora, **Faltan Manos** (Faltan Manos, 2015), con el fin de generar un beneficio a nivel ciudadano, implantándose en los planes de asistencia de los grupos de voluntariado participantes.

A nivel de software, el proyecto consiste en diseñar un prototipo para una aplicación móvil para teléfonos inteligentes (*Smartphones*). Se desarrolló para teléfonos inteligentes con sistema operativo **Android**. Su fin principal es organizar, ordenar y asignar tareas a múltiples grupos de voluntariado dentro de rangos de tiempo y objetivos a cumplir dentro de las necesidades existentes en la comunidad afectada por una catástrofe.

Los mayores desafíos del proyecto son, desde el punto de vista de programación, aprender el uso de bases de datos en entornos móviles, gestión de cuentas de usuario y manejo de funciones en la capa del servidor, diseño del software y de la interfaz gráfica para su

implementación en equipos móviles, aprendizaje de usabilidad y experiencia del usuario en la plataforma.

La temática de la aplicación es poder gestionar una gran cantidad de usuarios desde un sistema central, donde un coordinador pueda fijar los requisitos de participación en un marcador fijado en un mapa geográfico, para que el sistema organice a los usuarios postulantes según su índice de participación en tareas anteriores y su distancia al punto de trabajo, y les informe cuándo y dónde son necesarios.

1.4 OBJETIVOS Y ALCANCES DEL PROYECTO

1.4.1 Objetivo general

El objetivo general es desarrollar un prototipo funcional de una aplicación móvil que pueda ser utilizada como apoyo para la gestión de postulantes a voluntariado en situaciones de catástrofe.

1.4.2 Objetivos específicos

Dentro de los objetivos específicos, se tiene:

- Definir los requisitos requeridos.
- Definir el modelo de datos apropiado.
- Diseñar el sistema.
- Construir el sistema.
- Realizar pruebas del sistema.

1.4.3 Alcances

Dentro de los alcances, se tiene:

- Sobre el modelo de datos escogidos para trabajar: Debido a la gran cantidad de posibilidades de tareas y necesidades que puedan requerirse de los voluntarios, se decidió trabajar con una cantidad reducida de tareas, elegidas de manera más general. Esto quiere decir que, por ejemplo, todas las posibles tareas que sean parte del rubro de salud se concentran en una sola opción llamada "Salud".

- Se plantea que esta solución será incorporada en una siguiente etapa en el sistema construido por la entidad **Faltan Manos**. Es por esto que, debe considerarse que el prototipo pueda ser adaptable de forma modular a otros sistemas al momento de ser construido.
- El prototipo considera un único nivel de *ranking*, valor utilizado para puntuar a cada voluntario según su nivel de participación en tareas anteriores y que se explicará en mayor profundidad más adelante, para realizar el ordenamiento de voluntarios, en vez de múltiples valores por cada tipo de tarea. Esto es para facilitar el desarrollo y las pruebas realizadas en las etapas de construcción del prototipo.

1.5 ORGANIZACIÓN DEL DOCUMENTO

El presente trabajo de título está organizado en 7 capítulos:

Capítulo 1: Se expone la introducción al tema, con los antecedentes y motivaciones, la descripción del problema, la solución propuesta, los objetivos a cumplir y la metodología utilizada.

Capítulo 2: Se describen los conceptos fundamentales para entender este trabajo, separando el capítulo en dos puntos, el marco teórico, donde se explican las catástrofes y su influencia en la ciudadanía, y el estudio previo, que hace un análisis sobre la respuesta de la ciudadanía y del gobierno frente a las situaciones de emergencia causadas por catástrofes.

Capítulo 3: Se expone el desarrollo del prototipo, donde se explica el tratamiento de la metodología y los diagramas que conformarán al prototipo terminado. También se detallan los requisitos del prototipo, los casos de uso implicados, y se presentan los *mockups* desarrollados para la aplicación. Finalmente se detallan los diagramas de clases y de componentes.

Capítulo 4: Se explica la implementación del algoritmo utilizado para la solución del problema planteado, y se presenta el prototipo final creado y su funcionamiento

Capítulo 5: Se lleva a cabo la validación y evaluación de sistema a través de una serie de pruebas: pruebas unitarias, de integración, de validación, de sistema y de aceptación.

Capítulo 6: Se desarrollan en detalle las conclusiones obtenidas.

Capítulo 7: Bibliografía.

CAPÍTULO 2. CONCEPTOS

2.1 MARCO TEÓRICO

2.1.1 Sobre las emergencias en Chile y el mundo

Una catástrofe es todo suceso grave, repentino y considerable ocurrido dentro de una comunidad. Pueden ser naturales, como los terremotos erupciones volcánicas o huracanes, o provocados por el hombre, como los incendios o accidentes industriales. Toda situación de catástrofe provoca situaciones de riesgo en los que existe la posibilidad de que se produzca un daño, ya sea desde afectar a un grupo de personas, hasta alcanzar a todo un sector ciudadano (Páez, Arroyo, & Fernandez, 1995).

Existen numerosos tipos de desastres y emergencias que pueden aparecer, como se muestra en la Tabla 2-1, y la frecuencia de su ocurrencia es variable dependiendo de la zona geográfica, del clima y la demografía del lugar:

Tabla 2-1: Tipos de desastres (EM-DAT, 2009)

Tipo de desastre	Sub-grupo	Definición	Ejemplos
Natural	Geofísico	Peligro originado por movimientos terrestres.	Terremotos, movimiento terrestre, actividad volcánica.
	Meteorológico	Peligro causado por climas extremos de vida corta y por condiciones atmosféricas que duran desde minutos a días.	Temperaturas extremas, niebla, tormentas.
	Hidrológico	Peligro causado por movimientos de agua, superficial o subterránea, dulce o salada.	Inundaciones, deslizamientos de tierra, mareas.
	Climatológico	Peligro causado por procesos atmosféricos de vida larga y clima de duración estacional a décadas.	Sequía, explosión de lagos glaciares, fuegos fatuos.
	Biológico	Peligro causado por la exposición a organismos, sustancias tóxicas, o enfermedades.	Epidemias, infestación de insectos, accidentes de animales.

Tipo de desastre	Sub-grupo	Definición	Ejemplos
	Extraterrestre	Peligro causado por asteroides, meteoritos y cometas que entran a la atmósfera, y por cambios en las condiciones interplanetarias de la Tierra.	Impactos, clima especial.
Tecnológico	Accidente industrial		Derrame químico, colapsos, explosiones, incendios, escapes de gas, envenenamientos, radiación.
	Accidente de transporte		Accidentes aéreos, en caminos, en rieles, marítimos
	Accidente misceláneo		Colapsos, explosiones, incendios, otros

Por ejemplo, en el caso de Chile, es particular la cantidad de sismos que ocurren, dado que este país está ubicado en el Cinturón de Fuego del Pacífico, lo que lo hace el segundo país sísmico más activo del mundo después de Japón. Este es un ejemplo de las situaciones de emergencia para las que Chile debe estar preparado constantemente.

Cuando ocurre una catástrofe, como los siempre esperados terremotos en Chile, se da a lugar el llamado ciclo de desastres (Gavidia), el cual especifica las distintas etapas por las que la ciudadanía pasa al enfrentar una situación como tal:

- a) **Mitigación:** Para disponer a la población frente a las situaciones de desastre, deben aplicarse medidas de mitigación, para aminorar el daño directo del desastre a ocurrir, y así reducir el impacto de éste.
- b) **Preparación:** Son las actividades que preparan a la comunidad y a todas las organizaciones pertinentes para responder de manera rápida y eficiente ante un desastre. Pueden ser actividades de prevención, para educar a la población a estar dispuesta para enfrentar una situación de catástrofe, y preparación, para crear planes de organización y respuesta frente a posibles contingencias, y disponer medidas de alerta para notificar de manera pronta y formal a la población y a las entidades pertinentes.
- c) **Respuesta:** La ocurrencia de una catástrofe natural trae consigo múltiples problemas en la zona en que ocurra. Las medidas de respuesta se aplican inmediatamente después de ocurrido el incidente, e incluso poco después de haber comenzado. Estas

medidas incluyen evacuación, búsqueda, rescate de personas y la rápida disposición de alimentos, agua, medicinas y refugio.

- d) **Recuperación:** Después de ocurrido y evaluado el desastre, comienzan los procesos de recuperación. Primeramente, deben restablecerse los servicios y necesidades básicas a la población afectada, como comida, agua y techo, y otros servicios como comunicación y asistencia médica y psicológica. A esta etapa se le llama también rehabilitación. Posteriormente, comienza la reconstrucción, donde empiezan las labores de restauración del sistema de vida de la población afectada por el desastre, reparando las infraestructuras y alrededores y revitalizando el flujo social y económico (National & Community Service, 2006).

2.1.2 Participación de la comunidad

Generalmente los problemas de una comunidad son tratadas y resueltas por gobiernos locales y equipos de respuesta. Cuando la emergencia supera las capacidades de tales entidades, entran en juego el gobierno nacional, organizaciones de ayuda frente a emergencias, y organizaciones de voluntariado, y todas estas entidades acuden a la zona de desastre a prestar servicios y asistencia. Los **equipos de voluntariado** son la entidad a la que se refiere este informe.

Cuando hay que crear medidas de prevención y alerta frente a catástrofes, la cantidad de gente participante es limitada, contable y controlable, permitiendo que las tareas se organicen mejor. Después de una catástrofe, sabiendo que se requiere toda la ayuda posible para poder volver a levantar a la comunidad, la cantidad de gente que pueda llegar a ayudar y asistir es impredecible. Dentro de las personas que convergen en desastres, se han catalogado 6 grupos:

- **Voluntarios**, que ayudan a las víctimas o a otras organizaciones.
- **Los que regresan**, que son las personas afectadas por el desastre que fueron evacuadas.
- **Ansiosos**, que provienen de otras zonas que no fueron afectadas e intentan obtener información sobre familiares y amigos.
- **Curiosos**.
- **Fans**, que expresan gratitud o muestran apoyo a trabajadores y auxiliares
- **Explotadores**, que aprovechan las circunstancias para obtener un beneficio personal.

De los seis grupos, el primero es el que conforma las organizaciones de voluntariado (National & Community Service, 2006).

2.1.3 Voluntariado en Chile

Según John Wilson, del Departamento de Sociología de la Universidad de Duke, el voluntariado se define por toda acción que:

- Va en beneficio de otros
- Es asumido de forma libre y sin coerción
- No busca remuneración financiera alguna
- Opera de forma organizada, en el marco de una planificación (Wilson, 2000).

El voluntariado en este país se congrega de dos formas: mediante organizaciones de voluntariado, en el que los ciudadanos pueden inscribirse y participar en nombre de tal organización, y mediante grupos autónomos de personas que se ponen de acuerdo para trabajar de manera independiente. En casos de catástrofe, todas estas personas asistirán a ayudar en lo que puedan. Aunque esto puede traer inconvenientes, como se explicará más adelante.

La diferencia de la participación de estas personas en las distintas etapas del ciclo de desastres es que en las tres primeras etapas del ciclo los voluntarios participantes se encuentran en la localidad del siniestro, mientras que es en la última etapa en la que se acercan personas y equipos de trabajo de otras regiones para participar, y en quienes se enfocan los análisis posteriores.

La **INJUV** comprende la principal red de voluntariado del país, a través de su programa **Red Nacional de Voluntariado Juvenil**. Su principal función es agrupar organizaciones civiles de voluntariado que integren voluntarios de entre 15 y 29 años de edad, de distinta nacionalidad y que trabajen dentro del territorio de Chile, apoyando su sostenibilidad. Cuando ocurre una catástrofe, la **INJUV** realiza un llamado a nivel país para que los equipos de trabajo congregados participen llamando a los voluntarios registrados en cada equipo, especificando el lugar donde deben participar

La **Red de Organizaciones de Voluntariado de Chile** (Chile Voluntario, 2015) también congrega muchos programas de voluntariado, que permite que tales organizaciones participen en actividades de distinta índole, como promover la resiliencia de niños y niñas a

través del arte, tarea que cumple **CreArte** (CreArte, 2016), Contribuir al desarrollo físico, ético y social de los jóvenes como lo hace la **Asociación Guías y Scouts de Chile** (Guías y scouts de Chile, 2016).

2.1.4 Herramientas existentes

Dentro del ámbito de la mitigación de desastres, la mayoría de los países poseen organismos creados para prevenir o solucionar los problemas derivados de catástrofes. Éstos a su vez desarrollan programas y sistemas de apoyo para cada etapa e mitigación de desastres:

Google Person Finder: Aplicación *Web* de código abierto, desarrollado por voluntarios de la compañía **Google** en respuesta al terremoto en Haití del año 2010, utilizando sus servidores. Ayudó a la reubicación de personas desaparecidas en ese año y en muchos otros casos de desastre permitiendo a los usuarios subir imágenes de familiares y amigos desaparecidos, y funcionando como una central de datos compatibilizados desde múltiples registros como la prensa o servicios de búsqueda de personas desaparecidas. Sigue funcionando para las emergencias globales más sobresalientes (en este último periodo para los dos terremotos ocurrido en Nepal en abril y mayo del año 2015), pero con la condición de que, una vez finalizada las etapas de búsqueda, los repositorios se eliminan para proteger la privacidad de las personas registradas para su búsqueda (Google, 2010).

MicroMappers: Plataforma desarrollada por **QCRI** en conjunto con las **Naciones Unidas** y la **Standby Task Force**. Conecta el uso de mapas con el despliegue y filtrado de información como texto, imágenes, videos e imágenes satelitales a través de una serie de marcadores llamados “*Clickers*”, correspondientes para cada tipo de información, como lo que se muestra en la Ilustración 2-1. Además, combina computación humana y de máquina al utilizar el motor de inteligencia artificial **AIDR** (*Artificial Intelligence for Disaster Response*, o Inteligencia Artificial para Respuesta a Desastres), que implica clasificar *tweets* y mensajes de texto al aplicar un filtrado de manera colaborativa, abierta y distribuida (iRevolutions, 2014).

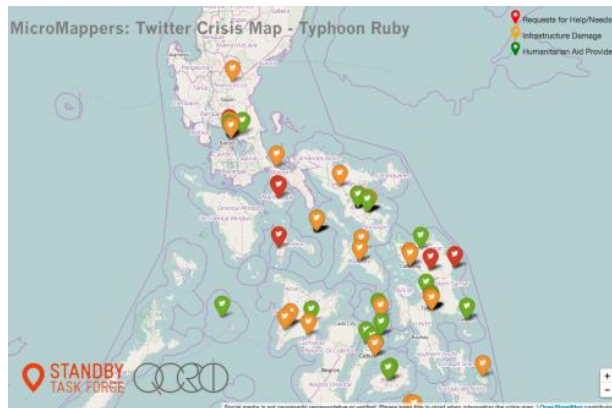


Ilustración 2-1: MicroMappers (iRevolutions, 2014)

Ushahidi (Testimonio, en Swahili): Creado por **Ushaguzi**, una iniciativa en conjunto formado para monitorear las elecciones en Kenia en el año 2013 debido a la violencia postelectoral del 2007, es un sistema de código abierto cuyo propósito es contar lo que está ocurriendo durante una situación de emergencia, y generar un mapa interactivo en tiempo real, visible desde cualquier ordenador o *smartphone*, y así, realizar un seguimiento durante las situaciones de crisis (Ushaguzi, 2013). Puede decirse que posee un enfoque más informativo que otros sistemas de asistencia. Aún así, posee dos cualidades que lo destacan de otras aplicaciones: es compatible con cualquier teléfono móvil, haciéndolo más accesible que las aplicaciones dirigidas a *smartphone*, recogiendo tanto imágenes y videos desde un teléfono inteligente como mensajes de texto desde un teléfono tradicional. Por otra parte, al trabajar con código abierto, permite que cualquier organización o gobierno lo utilice para sus propias necesidades (Moreno, 2012). Puede verse un ejemplo de la interfaz de esta aplicación en la Ilustración 2-2.

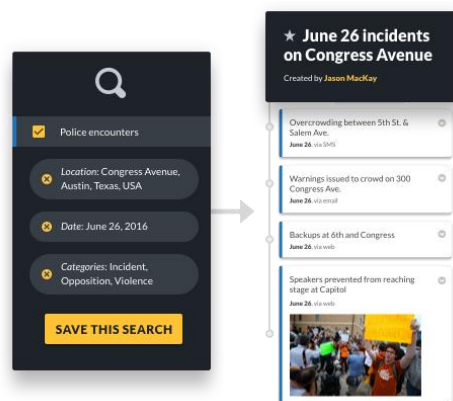


Ilustración 2-2: Ushahidi (Ushaguzi, 2013)

Eden: Es una plataforma de *software* de código abierto que provee soluciones para el manejo de desastres, posibilitando a la comunidad reducir el impacto de los desastres al permitir especificar las necesidades de la comunidad afectada, entregando herramientas especializadas para cada rubro especificado, y coordinando a las entidades correspondientes y sus recursos (Sahana Software Foundation, 2004). Una demostración de estas soluciones se muestra en la Ilustración 2-3. Fue creada por **Sahana Software Foundation**, una organización sin fines de lucro originada en Sri Lanka (“Sahana” significa “alivio” en singalés), que promueve el uso de software libre y de código abierto para el manejo de información y recursos en situaciones de emergencia y de catástrofes. Esta aplicación puede incluir operaciones de rescate, organización de voluntariado, concientización de situaciones, manejo de refugios, y logística, y se ha utilizado, por ejemplo, en Pakistán después del terremoto ocurrido el 2005 para administrar los reportes de personas desaparecidas, En China después del terremoto en Chengdu-Situan para encontrar y reunir con su familia a 42 sobrevivientes, y se ha incorporado por la Cruz Roja en Centro Inteligente de Comandos de IBM luego de los incendios forestales del año 2012 en Chile (University of Maryland, 2013).

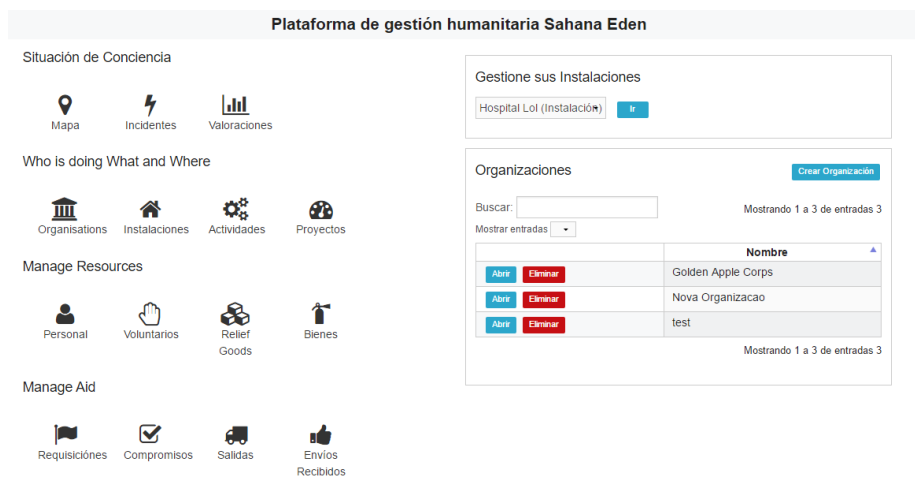
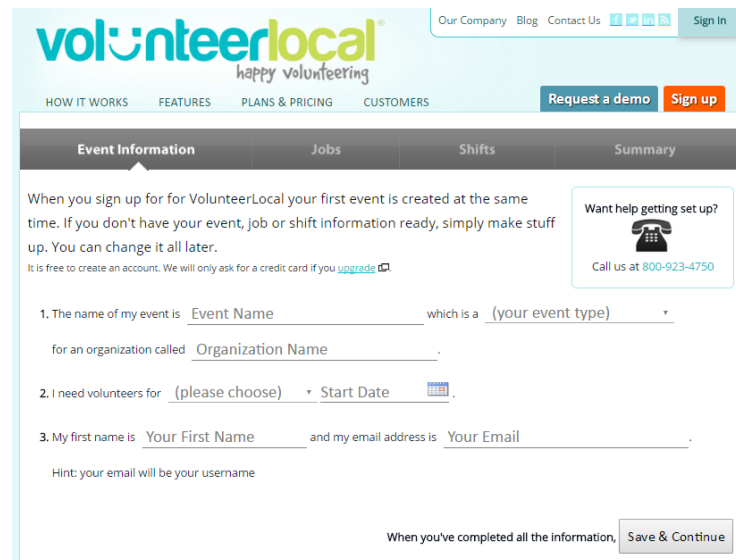


Ilustración 2-3: Eden (Demo) (Sahana Software Foundation, 2004)

VolunteerLocal: Es un sistema *Web* creado el 2003 para manejar las grandes cantidades de voluntarios que aparecieron para el festival *Des Moines Arts*. Permite la gestión de eventos de distinta índole de manera autónoma, permitiendo a los organizadores publicar sus eventos para que las personas que deseen participar como voluntarios ingresen sus datos y la fecha en la que pueden participar en un formulario como en la Ilustración 2-4, manteniendo total comunicación entre organizadores y voluntarios (VolunteerLocal, 2015).



The screenshot shows the VolunteerLocal website interface. At the top, the logo 'volunteerlocal' is displayed with the tagline 'happy volunteering'. Navigation links include 'Our Company', 'Blog', 'Contact Us', and 'Sign In'. A secondary navigation bar contains 'HOW IT WORKS', 'FEATURES', 'PLANS & PRICING', 'CUSTOMERS', 'Request a demo', and 'Sign up'. The main content area is titled 'Event Information' and includes tabs for 'Jobs', 'Shifts', and 'Summary'. A text block explains that a first event is created upon sign-up and can be modified later. A 'Want help getting set up?' sidebar offers a phone icon and the number 800-923-4750. The form consists of three numbered steps: 1. Event Name and Organization Name; 2. Number of volunteers and Start Date; 3. User's first name and email address. A hint states that the email will be the username. A 'Save & Continue' button is at the bottom right.

Ilustración 2-4: VolunteerLocal (VolunteerLocal, 2009)

VolunteerMatch: Sistema Web lanzado en 1998 para los usuarios que estén en búsqueda de trabajos voluntarios en los que puedan participar, mostrando un catálogo de eventos y causas publicados por organizaciones sin fines de lucro, y permitiendo a los voluntarios buscar la categoría de trabajos a la que quieren asistir, dependiendo de la zona donde se encuentren. Ha formado parte de los esfuerzos creados a raíz de eventos como los ataques del 11 de Septiembre o el Huracán Katrina (VolunteerMatch, 2015).

De las dos aplicaciones mencionadas anteriormente, **VolunteerLocal** es la que más se acerca a la función de gestionar a las grandes cantidades de voluntarios que pueden aparecer en casos de catástrofes. La diferencia radica en que el gestor de evento debe especificar las fechas en las que se requieren los voluntarios, siendo que esta operación podría ser automática por la urgencia que puede implicar conseguir y organizar voluntarios al ocurrir una catástrofe. Por su parte, **VolunteerMatch** permite que sean los usuarios quienes busquen los trabajos que quieran realizar, lo que implica que, si bien les da más autonomía a los usuarios, no garantiza que un gran número de ellos participe cuando ocurra una emergencia.

Por su parte, la **Cruz Roja** presenta múltiples aplicaciones para el usuario común de teléfonos inteligentes, distribuyendo información sobre cómo prevenir y actuar en situaciones de desastre. Para los miembros de la **Cruz Roja**, existen sistemas como **DMIS (Disaster Management Information System)**, o **Sistema de Información de Manejo de Desastres**, que permite el intercambio de información sobre desastres, monitoreando amenazas potenciales que podrían afectar las acciones frente a emergencias, reportando emergencias a nivel de

campo, y accediendo a información operacional y técnica de todo el mundo, funcionando, al igual que algunos de los sistemas anteriores, como un medio preventivo (Red Cross).

DMIS comenzó en Febrero del 2001 en respuesta a la necesidad de decisiones informadas, rapidez y preparación operacional eficiente, asistiendo al intercambio de información en la comunidad y en el soporte de una disposición y respuesta eficiente contra desastres por parte de **Cruz Roja** a nivel global. **DMIS** dará paso a **Strategy 2020**, con similares funcionalidades, integrando a la **Cruz Roja** y **Red Crescent** en un trabajo en conjunto con múltiples entidades (International Federation of Red Cross and Red Crescent Societies, 2010).

Sobre la Geolocalización:

La Geolocalización se refiere el uso de **GPS** (*Global Positioning System*, o Sistema de Posicionamiento Global) y de la conexión a Internet de un dispositivo móvil para identificar la posición geográfica de un dispositivo (Google, 2016). El sistema más conocido que utiliza esta función es **Google** a través de su servicio **Google Maps**, el cual puede ser implementado en otro sistema a través de su **API** (*Application Programming Interface*, o Interfaz de Programación de Aplicaciones). Una **API** es un conjunto de comandos, funciones, protocolos y objetos que pueden ser proveídos por un sistema externo para ser invocados en la creación de software, y así disminuir la cantidad de código que el programador debe escribir (Techterms, 2016). **Google**, a través de sus muchas **API**, permite a diseñadores y programadores utilizar el servicio de **Google Maps** para implementar un mapa con un gran abanico de funcionalidades y utilidades, como la búsqueda de puntos de interés, cálculo de rutas y direcciones, posicionamiento de marcadores, etcétera (Google, 2016). La Ilustración 2-5 muestra la página *Web* desde donde se puede acceder a los servicios ofrecidos por **Google** para utilizar esta **API**.

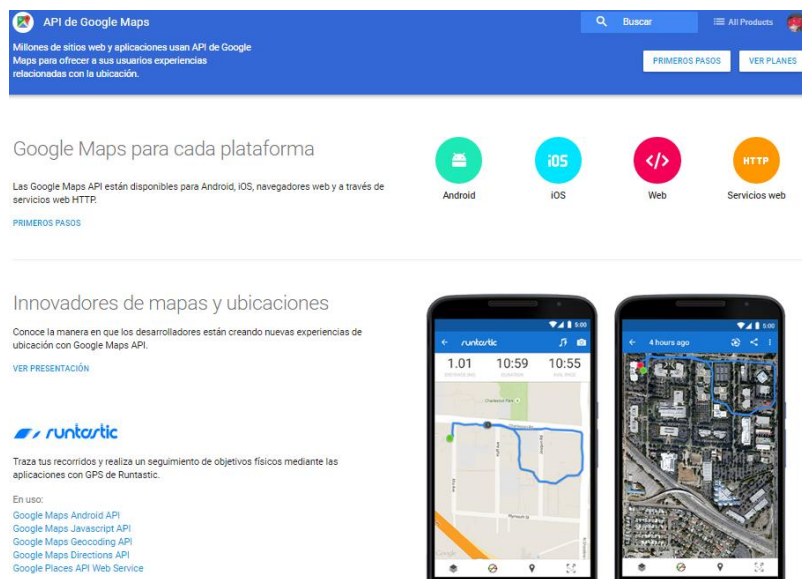


Ilustración 2-5: Google Maps API (Google, 2016)

Algunas de las aplicaciones principales en usar **Google Maps** para temas relacionados con los desastres y su preparación, respuesta o mitigación son:

NUKEMAP: Simulador *Web* creado por Alex Wellerstein entre los años 2012 y 2014, que permite simular los efectos de bombas nucleares en una ciudad o punto geográfico. Posee valores preestablecidos de estudio como la bomba *Fat Man* que impactó en Kagasaki, o las pruebas realizadas por Corea del Norte. Además, permite conocer efectos como la cantidad de bajas a distintas distancias de la explosión o el radio de efecto de las lluvias radioactivas utilizando una interfaz como en la Ilustración 2-6 (Wellerstein, 2014).



Ilustración 2-6: NUKEMAP (Wellerstein, 2014)

FloodMap: Simulador *Web* creado por Sameer Burleque el año 2012, que aprovecha la información que **Google Maps** entrega sobre la elevación de los puntos terrestres para que, cuando el usuario especifica la altura en que el océano se eleve, muestre las zonas que podrían verse inundadas (Burle, 2012).

2.2 ESTUDIO PREVIO

2.2.1 Respuesta ciudadana y gubernamental

Según un análisis realizado por Walter Alejandro Imilan, Antropólogo y Doctor en Planificación Urbana y Regional (Universidad de Chile), a los múltiples reportes realizados sobre el terremoto del 27 de Febrero del año 2010 en nuestro país, destaca tres puntos coincidentes respecto a la capacidad de aprendizaje de la sociedad en su participación frente a las catástrofes:

- Debe cambiar el enfoque de los procesos de recuperación del flujo social y económico, ya que se posee una visión centrada en la recuperación de las viviendas, y que debe evolucionar a una perspectiva multidimensional y más sistémica que integre también el desarrollo productivo, equipamiento, salud física y mental, entre otros aspectos que en su conjunto conforman el progreso de la ciudadanía.
- Las decisiones deben realizarse en conjunto entre las autoridades y la población afectada, obteniendo un punto de vista más completo respecto a la situación de la población y las medidas que se tomen para la solución de sus problemas.
- Descentralizar la gestión de la **ONEMI**, o la **Oficina Nacional de Emergencia del Ministerio del Interior y Seguridad Pública** (ONEMI, 1974), de manera que las acciones a tomar en el caso de una catástrofe involucren respuesta y comunicación fluida entre el gobierno y los municipios en vez de darle la responsabilidad a una división, agencia o subsecretaría. Los municipios de las localidades afectadas deben recibir un apoyo directo con todos los estamentos gubernamentales para que sus capacidades no se vean superadas. Destaca en este punto lo ocurrido el 27 de Febrero del 2010 la disminución de la calidad de los procesos de reconstrucción debido justamente al poco apoyo recibido por los municipios (Imilan, 2015).

En resumen, la meta principal que deben imponerse las autoridades es crear una red de comunicación que involucre al Estado directamente, a los gobiernos locales, y a la población. Aún hoy en día, más de 5 años después del terremoto del 2010, la **ONEMI** y el **SHOA, Servicio Hidrográfico y Oceanográfico de la Armada** (SHOA, 1990), siguen vigentes, pese a la promesa realizada de crear un nuevo sistema nacional de emergencia y protección civil, y pese también a la pérdida de credibilidad de estas instituciones debido a errores como el retraso en la alerta de tsunami y la pérdida de vidas en la costa debido a esto (La Tercera, 2016).

Actualmente, la **ONEMI** trabaja en conjunto con la Organización de las Naciones Unidas, u **ONU**, en la creación del proyecto **Adecuación de Estándares Internacionales para la Respuesta en Emergencias en Chile**, que busca aumentar la eficiencia en el uso de recursos destinados a la respuesta a una catástrofe, incorporar estándares internacionales en la entrega de ayuda humanitaria, y fortalecer el **Sistema Nacional de Protección Civil** (ONEMI, 2016).

El **Sistema Nacional de Protección Civil** es un sistema asumido por las autoridades y servicios públicos y privados, el mundo de la ciencia y la tecnología, y la comunidad organizada (ONEMI, 2013). Organiza los esfuerzos a través de un uso escalonado de recursos, lo que implica que su accionar es ejecutado jerárquicamente en subsistemas desde un nivel nacional, regional, provincial y comunal, coordinando estos esfuerzos por medio de mandos:

- Autoridad: Autoridad administrativa respectiva al nivel: Ministro del Interior, Intendente Regional, Gobernador Provincial, Alcalde.
- Coordinación: Director Nacional **ONEMI**, Director Regional **ONEMI** y Directores de Protección Civil y Emergencia de Gobernaciones Provinciales y Municipios.
- Técnico: De acuerdo al tipo de evento.

El accionar del Sistema Nacional de Protección Civil, aunque inculca la comunicación entre distintos niveles gubernamentales, no asegura la organización conjunta entre las autoridades de una localidad y su población. Volviendo al caso del terremoto del 27 de febrero, según un proyecto de investigación posterior al suceso dirigido por **LatinLab**, el **GSAPP (Graduate School of Architecture, Planning and Preservation)** de la **Universidad de Columbia**, y el **Santiago Research Cell**, Se reconoció como uno de los actores públicos en el proceso de reconstrucción a la Junta de Vecinos de Curicó con principal interés en el bien

comunitario. Los demás actores públicos, como la Intendencia del Maule, la Municipalidad, o el **EGIS** (Entidad de Gestión Inmobiliaria Social) Municipal, tenían como principal interés la reconstrucción urbana, pero no iban más allá de la recuperación de la infraestructura (LatinLab, GSAPP, Santiago Research Cell, 2015).

2.2.2 Participación de voluntariados en Chile

La participación de la ciudadanía cambia dependiendo de la situación económica por la que esté pasando el país. Por ejemplo, al analizar las actividades de voluntariado de principios de los años '90, se debe tomar en cuenta el periodo por el que Chile estaba pasando, al transcurrir el fin del Régimen Militar y el inicio de la democracia en este país. Durante dicha época, el trabajo de voluntariado con mayor visibilidad fue el de las Damas de Colores, a cargo de las esposas de jefes de las Fuerzas Armadas y que asistía a niños, enfermos y discapacitados, lo que llevó a la monopolización del concepto de “voluntariado” por la politización de estas prácticas en la opinión pública. Así mismo, nacieron muchas iniciativas como Ollas Comunes, los Comités Solidarios, y los Trabajos de Verano universitarios, que buscaban enfrentar la situación de pobreza causada por la crisis económica de principios de los años '80, y en donde la Iglesia Católica jugó un papel principal, y otras iniciativas que ayudaban a personas que eran perseguidas o que trabajaban por la restitución de la democracia (Zulueta Azócar, 2003).

Actualmente, el desarrollo económico ha permitido que haya más personas capaces de ayudar a otras, junto con el avance tecnológico respecto a otras épocas, lo que implica, por ejemplo, mayor cobertura de la prensa frente a estas iniciativas, y el acceso y la organización de iniciativas a través de Internet. Así mismo, existe un mayor respaldo desde el sector empresarial, se han implementado programas y departamentos de voluntariado, y se ha formado una mayor toma de conciencia de la importancia del voluntariado (Zulueta Azócar, 2003). Hoy en día, para la organización de las personas que realizan trabajo de voluntariado lo hacen generalmente a través de su Iglesia, de fundaciones u ONGs (Organización No Gubernamental), de su institución educacional o junto a organizaciones familiares o vecinales, según estadísticas obtenidas de un estudio realizado por Fundación Trascender y GFK Collect, a través de encuestas con respuestas múltiples a 2.600 personas, y representadas en la Ilustración 2-7, donde se indica, por ejemplo, que es más común que el voluntariado participe en iniciativas organizadas por colectivos religiosos y a través de fundaciones u ONGs. (Fundación Trascender & GFK Collect, 2013).

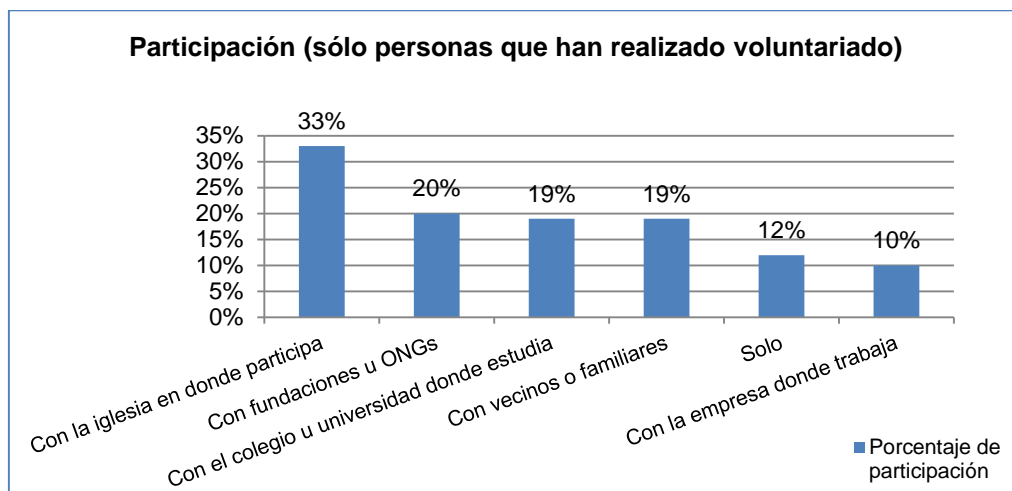


Ilustración 2-7: Gráfico de participación de voluntariado

El mismo estudio también evoca a las razones de no realizar voluntariado, también a través de encuestas con respuestas múltiples, como se representa en la Ilustración 2-8:



Ilustración 2-8: Gráfico de razones de no-participación de voluntariado

Puede verse que la razón más común para no participar en tareas de voluntariado es la falta de tiempo del encuestado al momento de realizarse llamados, con un 53% de personas indicando esta alternativa, seguida por la falta de interés en realizar este tipo de actividades.

CAPÍTULO 3. DESARROLLO DEL SOFTWARE

En este capítulo se presentan las características del proyecto y la metodología utilizada para su desarrollo. El objetivo de esta sección es comprender los requisitos del sistema y justificar las razones de sus métodos de construcción.

3.1 METODOLOGÍA

Se ha optado por utilizar un modelo de desarrollo evolutivo con un proceso exploratorio basado en prototipos. Un prototipo es una versión del software utilizada como prueba del software implementado. El modelo de prototipos empleado es una adaptación del modelo de desarrollo en espiral creado por Barry Boehm (Boehm, 1988). En dicho modelo, el ciclo de desarrollo se representa como una espiral, en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra como ocurre en el modelo cascada.

El uso de un modelo de desarrollo evolutivo basado en prototipo se fundamenta en comenzar el desarrollo de un *software* a partir de sus funcionalidades mínimas, e ir desarrollándolo progresivamente.

Se escogió esta metodología debido a que el prototipo, en su desarrollo, se encuentra sujeto a cambios, por lo que se requiere retroalimentación por parte de los *stakeholders* para validar las funcionalidades producidas en cada iteración.

3.1.1 Desarrollo basado en prototipado

Un prototipo es una versión del *Software* utilizado para demostrar los conceptos aprendidos, probar las opciones de diseño y alcanzar una mejor comprensión del problema y sus posibles soluciones. Por estas razones, un prototipo posee un funcionamiento limitado en cuanto a su capacidad, confiabilidad o eficiencia.

A continuación, se explican las etapas en la construcción de las distintas iteraciones de desarrollo, representadas en la Ilustración 3-1:

1. Definición de requisitos del sistema, para poder identificar los requisitos que debe cumplir cada prototipo.

2. Diseño del prototipo. Se crean diagramas que documenten distintas perspectivas del prototipo a desarrollar.
3. Implementación del prototipo (codificación).
4. Pruebas. Utilizar el prototipo con el fin de probar que cumple los requisitos para los que fue diseñado.
5. Revisar y mejorar el prototipo. Iterar regresando al paso 1.

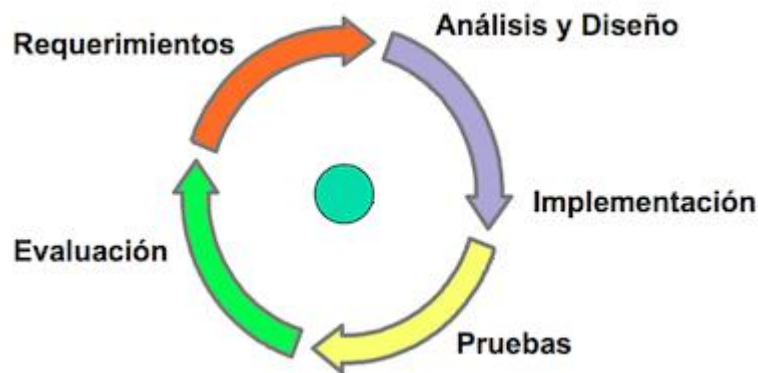


Ilustración 3-1: Etapas de una iteración

Algunas de las principales ventajas de trabajar con una metodología basada en prototipos son:

1. Reducción de riesgo e incertidumbre en el caso de no tener claros los requerimientos.
2. Progreso desde las primeras etapas.
3. Adquisición de conocimientos en las primeras etapas para su aplicación en futuras iteraciones.
4. Incremento en la aceptación del producto por parte del cliente y/o usuario.
5. Mejora de la comunicación entre desarrollador y cliente.

Sin embargo, algunas de las desventajas más relevantes son (Sommerville, 2005):

1. El prototipo se desarrolla a través de múltiples entregas, por lo que no es recomendable desarrollar documentación por cada prototipo realizable.
2. Los cambios continuos del prototipo pueden comprometer la estructura del *Software* final.
3. Posible baja calidad en el desarrollo del prototipo.

3.2 HERRAMIENTAS UTILIZADAS

Para la interfaz para el usuario y el coordinador del sistema, se utilizaron las siguientes herramientas:

- Editor **SUBLIME Text** v.2.0.2
- Lenguaje de programación HTML
- Framework **ionic** v.1.7.15

Para la comunicación de la interfaz con la base de datos, se utilizaron las siguientes herramientas:

- **AngularJS** v.1.3.5

Para la programación de la base de datos, se utilizaron las siguientes herramientas:

- Base de datos no relacional **MongoDB** v.2.4.9

Para la programación del servidor, se utilizaron las siguientes herramientas:

- **NodeJS** v.0.12.2
- **Express** v.4.9.8

Sistema operativo:

- Linux **Ubuntu** V.14.04
- Microsoft **Windows** 10 Home Single Language

3.3 MODELO DE NEGOCIOS

Siendo una aplicación de propósito social, no posee fines de lucro más allá del reconocimiento de su gestación. Al mismo tiempo, siendo su fin ser una herramienta ciudadana, busca alcanzar la mayor cantidad de usuarios que idealmente pertenezcan a grupos de voluntariado. De implementarse a gran escala, la mantención del sistema sería asumida por la entidad a cargo.

Para su difusión, se plantea realizar una campaña de información respecto al uso de la aplicación como una nueva herramienta de contacto y gestión, indicando a los gestores de cada equipo de voluntariado, o incluso a la INJUV, que notifiquen a los miembros sobre la

aplicación, y, una vez que ésta haya tenido un alcance considerable, incentivar a los usuarios a que informen a amigos o cercanos a también informar sobre la aplicación.

Se plantea también la implementación de este prototipo con el sistema construido por el equipo de **Faltan Manos** (Faltan Manos, 2015). La estrategia discutida comprende el uso de la aplicación final, la que integra a **VoluntariS** (Nombre que se le dio al prototipo creado) y a **Faltan Manos**, como un sistema multi-capas, donde la primera capa muestre puntos de trabajo que requieran personas o equipos voluntarios, la segunda capa se activaría cuando ocurra una catástrofe y realizaría las acciones del prototipo planteado en este informe, y una tercera capa que entidades como empresas y eventos pueden arrendar para fijar sus propios puntos de trabajo, a modo de servicio *Premium*. El arriendo de la tercera capa se determinó como solución viable para la mantención del sistema final.

Sobre el grupo objetivo al que va dirigida esta aplicación, se toma en cuenta que los asistentes a trabajos de voluntariado tienden a ser jóvenes entre 17 y 30 años, que poseen conocimiento y experiencia en el manejo de tecnología digital reciente e Internet. (Faltan Manos, 2015) Esto no significa que el grupo objetivo se limitará al recién especificado ya que, en última instancia, no se conoce quienes serán los usuarios de la aplicación. Se toman estos rasgos como referencia para el enfoque que se le dará al desarrollo de la aplicación.

3.4 ELABORACIÓN

El prototipo fue desarrollado según la metodología, produciendo múltiples iteraciones que originaron 2 prototipos, uno para el uso de los coordinadores del sistema, y otro para los demás usuarios.

Por otro lado, las tareas más destacadas de esta fase de la metodología fueron el análisis y el diseño.

3.4.1 Análisis y diseño de la elaboración

Sobre la primera iteración, se destacan los siguientes puntos:

- Construir la base de las vistas del prototipo.
- Hacer ingreso a una base de datos los datos de los usuarios.
- Hacer ingreso a una base de datos los datos del coordinador del sistema.

Con este prototipo se conforma una de las bases de la aplicación, asegurando la posibilidad de registrar y almacenar en la base de datos a los usuarios que deseen ser parte del sistema, mostrando también la primera versión de cómo se distribuirán las vistas. Se decidió empezar siguiendo tales metas para asegurar la participación del voluntariado, además de que la posibilidad de integrar los demás requerimientos en las siguientes iteraciones se puede realizar modularmente.

Sobre la segunda iteración, se destacan los siguientes puntos:

- Permitir la especificación y visualización de los puntos donde se espera que asistan los voluntarios.
- Permitir la especificación de la posición en el mapa de donde el usuario desea que se considere al ser llamado.

Esta iteración contempló integrar la API de **Google Maps** para que la visualización de los puntos registrados en la base de datos sea más gráfica. Esto también requirió añadir más atributos a los datos de los usuarios y de los puntos para registrar las coordenadas de latitud y longitud de éstos en el mapa. El coordinador del sistema sería capaz de añadir nuevos puntos al mapa, y el sistema permitiría que estos puntos sean indicados en el mapa desplegado a los voluntarios, concretando la comunicación coordinador-usuario.

Sobre la tercera iteración, se destacan los siguientes puntos:

- Dar información sobre cada punto desplegado en el mapa a los usuarios.
- Permitir la elección de la tarea a realizar para cada usuario, especificando su disponibilidad de trabajo.

Se consideró la creación de una nueva colección de datos, la cual recogerá la elección de cada usuario y la almacenará junto con la información más relevante de éste y el trabajo escogido, como su posición o la fecha escogida de asistencia.

Sobre la cuarta iteración, se destacan los siguientes puntos:

- Permitir a los voluntarios postular su disponibilidad de tiempo para ser llamados a trabajar.
- Distribuir a los voluntarios postulantes en cada punto de trabajo de forma que haya un flujo constante de voluntarios durante el periodo especificado de trabajo, manteniendo una cantidad constante de voluntarios trabajando en el punto especificado.

Se implementó en el servidor una función prototipo de asignación de voluntarios postulantes, el cual recoge los datos de la postulación y determina si cada voluntario es llamado o no tomando en cuenta la distancia entre los puntos geográficos entre la actividad y el usuario.

Sobre la quinta iteración, se destacan los siguientes puntos:

- Informar y notificar sobre el resultado de la postulación de cada usuario.
- Permitir al usuario encontrar el punto geográfico donde fue llamado, y calcular la ruta que debe tomar.
- Permitir al usuario modificar el equipo para el que trabajará.
- Implementar un sistema de *ranking* como parámetro de elección de voluntariado. Permitir al usuario saber su puntuación en *forma* de ranking dentro del sistema mediante un medidor gráfico.
- Implementar un medio de notificaciones para el aviso a los usuarios sobre su proceso de postulación.
- Mejorar la interfaz de la aplicación.

La aplicación mostraría una vista principal distinta dependiendo del estado del usuario dentro del sistema. Si fue llamado a trabajar, el sistema notificará a los postulantes a su dispositivo, y el mapa de la aplicación lo redirigirá automáticamente al punto donde fue llamado, mostrando también un conjunto de botones distinto a lo que se vería en la vista normal de la aplicación. El puntaje de *ranking* aumentará según lo estipule el coordinador al terminar la actividad mediante una retroalimentación de la actividad realizada.

3.5 REQUERIMIENTOS DEL SISTEMA

3.5.1 Requerimientos funcionales

Los requerimientos para los prototipos finales creados poseen los requerimientos de las fases anteriores:

Prototipo Voluntario:

En la Tabla 3-1 se presentan los requerimientos funcionales para el prototipo del usuario (RFV):

Tabla 3-1: Requerimientos Funcionales del prototipo Voluntario (RFV)

Identificador	Descripción
RFV-01	El sistema permite al usuario crear una nueva cuenta de usuario.
RFV-02	El sistema permite al usuario ver la información personal ingresada al registrarse a la base de datos.
RFV-03	El sistema permite el reingreso de un usuario al sistema.
RFV-04	El sistema permite al usuario explorar las vistas.
RFV-05	El sistema permite a los usuarios ver y explorar un mapa.
RFV-06	El sistema permite a los usuarios buscar su posición actual en el mapa.
RFV-07	El sistema permite a cada usuario especificar un punto en el mapa como ubicación de referencia para el momento en que postule a una actividad.
RFV-08	El sistema permite a los usuarios visualizar los puntos de trabajo disponibles para asistir, distribuidos geográficamente.
RFV-09	El sistema permite a los usuarios ver la información respectiva a cada punto de trabajo mostrado en el mapa.
RFV-10	El sistema permite a cada usuario registrar su postulación para el punto de trabajo para el que desea trabajar.
RFV-11	El sistema permite a cada usuario especificar el tiempo en que desea asistir al punto de trabajo.
RFV-12	El sistema permite a cada usuario determinar si fue o no escogido para trabajar en la actividad en la que postuló.
RFV-13	El sistema permite al usuario revisar la ubicación del punto para el que fue llamado
RFV-14	El sistema permite a cada usuario calcular la ruta a tomar desde la ubicación especificada al registrarse en el sistema, hasta el punto de trabajo asignado.
RFV-15	El sistema permite a cada usuario revisar su puntaje de <i>ranking</i> dentro del sistema.
RFV-16	El sistema permite a cada usuario modificar el equipo para el que trabaja.

Prototipo Coordinador:

En la Tabla 3-2 se presentan los requerimientos funcionales para el prototipo del coordinador (RFC):

Tabla 3-2: Requerimientos Funcionales del prototipo Coordinador (RFC)

Identificador	Descripción
RFC-01	El sistema permite el ingreso de un usuario coordinador.
RFC-02	El sistema permite al coordinador crear una nueva cuenta de coordinador.
RFC-03	El sistema permite al coordinador ver la información personal ingresada al registrarse a la base de datos.
RFC-04	El sistema permite el reingreso de un usuario al sistema.
RFC-05	El sistema permite al coordinador explorar las vistas.
RFC-06	El sistema permite al coordinador ver y explorar un mapa.
RFC-07	El sistema permite al coordinador buscar su posición actual en el mapa.
RFC-08	El sistema permite al coordinador crear nuevos marcadores en el mapa para representar los puntos de trabajo.
RFC-09	El sistema permite a los usuarios visualizar los puntos de trabajo disponibles para asistir, distribuidos geográficamente.
RFC-10	El sistema permite al coordinador eliminar marcadores previamente creados.
RFC-11	El sistema permite al coordinador ver la información respectiva a cada punto de trabajo mostrado en el mapa.
RFC-12	El sistema permite al coordinador revisar la cantidad de postulantes en cada punto de trabajo.
RFC-13	El sistema permite al coordinador escoger un marcador y ejecutar el algoritmo para determinar los voluntarios que asistirán a éste.
RFC-14	El sistema permite al coordinador realizar un <i>feedback</i> del trabajo realizado.
RFC-15	El sistema permite al coordinador actualizar el puntaje de <i>ranking</i> de cada usuario participante en cada punto.
RFC-16	El sistema permite que solo el coordinador creador de un punto pueda iniciar la actividad de ese punto.
RFC-17	El sistema permite al coordinador crear y eliminar grupos y equipos de trabajo del sistema.

En la Tabla 3-3 se presentan los requisitos funcionales implementados en la primera iteración:

Tabla 3-3: Requerimientos Funcionales para la Primera Iteración

Identificador	Descripción
RFV-01	El sistema permite al usuario crear una nueva cuenta de usuario.
RFV-02	El sistema permite al usuario ver la información personal ingresada al registrarse a la base de datos.
RFV-03	El sistema permite el reingreso de un usuario al sistema.
RFV-04	El sistema permite al usuario explorar las vistas.
RFC-01	El sistema permite el ingreso de un usuario coordinador.
RFC-02	El sistema permite al coordinador pueda crear una nueva cuenta de coordinador.
RFC-03	El sistema permite al coordinador ver la información personal ingresada al registrarse a la base de datos.
RFC-04	El sistema permite el reingreso del coordinador al sistema.
RFC-05	El sistema permite al coordinador explorar las vistas.

En la Tabla 3-4 se presentan los requisitos funcionales implementados en la segunda iteración:

Tabla 3-4: Requerimientos Funcionales Segunda Iteración

Identificador	Descripción
RFV-05	El sistema permite a los usuarios ver y explorar un mapa.
RFV-06	El sistema permite a los usuarios buscar su posición actual en el mapa.
RFV-07	El sistema permite a cada usuario especificar un punto en el mapa como ubicación de referencia para el momento en que postule a una actividad.
RFV-08	El sistema permite a los usuarios visualizar los puntos de trabajo disponibles para asistir, distribuidos geográficamente.
RFV-06	El sistema permite al coordinador ver y explorar un mapa.
RFC-07	El sistema permite al coordinador buscar su posición actual en el mapa.
RFC-08	El sistema permite al coordinador crear nuevos marcadores en el mapa para representar los puntos de trabajo.
RFC-09	El sistema permite al coordinador visualizar los puntos de trabajo disponibles para asistir, distribuidos geográficamente.

Identificador	Descripción
RFC-10	El sistema permite al coordinador eliminar marcadores previamente creados.

En la Tabla 3-5 se presentan los requisitos funcionales implementados en la tercera iteración:

Tabla 3-5: Requerimientos Funcionales para la Tercera Iteración

Identificador	Descripción
RFV-09	El sistema permite a los usuarios ver la información respectiva a cada punto de trabajo mostrado en el mapa.
RFV-10	El sistema permite a cada usuario registrar su postulación para el punto de trabajo para el que desea trabajar.
RFC-11	El sistema permite al coordinador ver la información respectiva a cada punto de trabajo mostrado en el mapa.

En la Tabla 3-6 se presentan los requisitos funcionales implementados en la cuarta iteración:

Tabla 3-6: Requerimientos Funcionales para la Cuarta Iteración

Identificador	Descripción
RFV-11	El sistema permite a cada usuario especificar el tiempo en que desea asistir al punto de trabajo.
RFC-12	El sistema permite al coordinador revisar la cantidad de postulantes en cada punto de trabajo.
RFC-13	El sistema permite al coordinador escoger un marcador y correr el algoritmo para determinar los voluntarios que asistirán a éste.

En la Tabla 3-7 se presentan los requisitos funcionales implementados en la quinta iteración:

Tabla 3-7: Requerimientos Funcionales para la Quinta Generación

Identificador	Descripción
RFV-12	El sistema permite a cada usuario determinar si fue o no escogido para trabajar en la actividad en la que postuló.
RFV-13	El sistema permite al usuario revisar la ubicación del punto para el que fue llamado
RFV-14	El sistema permite a cada usuario calcular la ruta a tomar desde la ubicación especificada al registrarse en el sistema, hasta el punto de trabajo asignado.
RFV-15	El sistema permite a cada usuario revisar su puntaje de <i>ranking</i> dentro del sistema.
RFV-16	El sistema permite a cada usuario modificar el equipo para el que trabaja.
RFC-14	El sistema permite al coordinador realizar un <i>feedback</i> del trabajo realizado.
RFC-15	El sistema permite al coordinador actualizar el puntaje de <i>ranking</i> de cada usuario participante en cada punto.
RFC-16	El sistema permite que solo el coordinador creador de un punto pueda iniciar la actividad de ese punto.
RFC-17	El sistema permite al coordinador crear y eliminar grupos y equipos de trabajo del sistema.

3.5.2 Requerimientos no funcionales

En la Tabla 3-8 se presentan los requerimientos no funcionales (RNF) del sistema:

Tabla 3-8: Requerimientos No Funcionales

Identificador	Descripción
RNF-01	El sistema debe soportar el acceso desde distintos sistemas operativos de dispositivos móviles.
RNF-02	El sistema debe seguir un estándar de diseño en toda la aplicación.
RNF-03	El sistema debe ser intuitivo, en menos de un minuto el usuario debe poder acceder a cualquier función.
RNF-04	El sistema debe ofrecer contenidos categorizados y ordenados en el menú, de manera que no sea necesario que el usuario recuerde información para deducir la ubicación del contenido.
RNF-05	El sistema debe considerar privilegios de acceso a las operaciones del servidor.

Identificador	Descripción
RNF-06	El sistema debe mostrar solo la información requerida de acuerdo a los alcances de cada usuario para la privacidad de los datos.
RNF-07	El sistema debe notificar al usuario en caso de producirse errores.
RNF-08	El sistema deberá soportar consultas simultáneas de usuarios, por lo que debe estar operacional y disponible las 24 horas del día, los siete días de la semana.
RNF-09	El sistema debe permitir el ingreso de usuarios registrados a través de un <i>login</i> de cuenta y contraseña.
RNF-10	El sistema debe ser capaz de almacenar la información especificada por los usuarios.
RNF-11	El sistema debe controlar que cada voluntario solo pueda ingresar una postulación a la vez

Requerimientos técnicos:

Se debe tener en consideración los requerimientos mínimos para un óptimo funcionamiento de la aplicación, tanto como los usuarios de la herramienta como el servidor que debe alojar al sistema.

1. Usuarios: Los dispositivos a utilizar deben trabajar con los sistemas operativos **Android**.
2. La aplicación necesita un mínimo de 6 megabytes de almacenamiento disponible para que pueda ser instalada y funcionar.
3. Servidor: Para poder utilizar la aplicación será necesaria la contratación de un Host dedicado que provea el almacenamiento de la base de datos.

El hardware del Host deberá tener:

1. Procesador 2,2 GHz, 75 W, 2MB de caché o superior.
2. Memoria de 1 GB o superior.
3. Espacio de almacenamiento de 50 GB o superior.
4. Acceso a la red.

3.6 DIAGRAMA DE PAQUETES

El diagrama de paquetes de la Ilustración 3.2 muestra las agrupaciones lógicas en las que se divide el sistema. La gestión de puntos de trabajo abarca depende de la gestión de perfil al requerir que haya usuarios voluntarios y coordinadores para el uso efectivo de la aplicación.

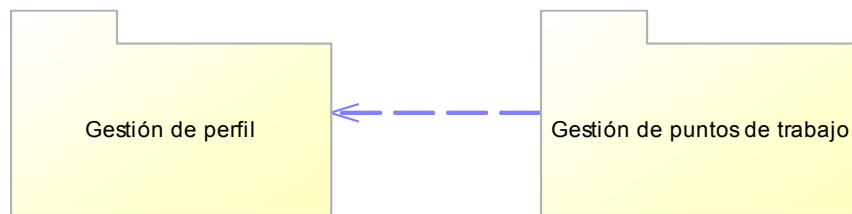


Ilustración 3-2: Diagrama de paquetes

3.7 DIAGRAMAS DE CASOS DE USO

A continuación, se describen los casos de uso, agrupados en las principales funcionalidades del sistema: Gestión de perfil, y gestión de los puntos de trabajo.

3.7.1 Gestión de Perfil

Este caso de uso hace referencia a los requisitos funcionales relacionados con la creación y manejo de cuentas de usuario y su información pertinente. Ambos, Voluntario y Coordinador, pueden crear cuentas de usuario en el sistema ingresando su información, y visualizar esta información en el futuro. El Coordinador, por su parte, puede crear y eliminar grupos de trabajo del sistema. Estos grupos de trabajo son visualizados como alternativas para cada Voluntario para escoger grupo o equipo con el cual estar afiliado, o cambiar de afiliación en algún momento posterior al registro. Se detalla el caso de uso en la Ilustración 3-3.

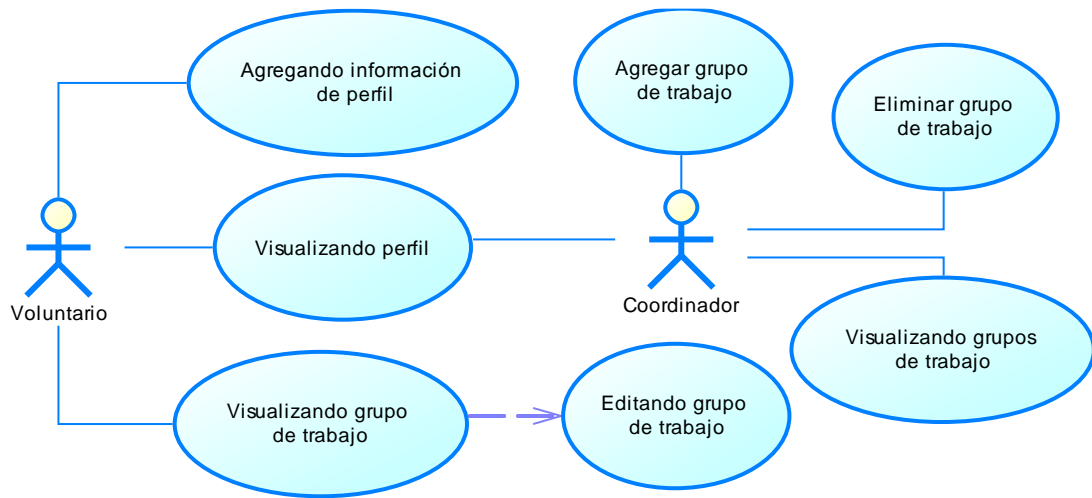


Ilustración 3-3: Diagrama de casos de uso: Gestión de perfil

3.7.2 Gestión de puntos de trabajo

Este caso de uso hace referencia a los requisitos funcionales relacionados con la creación y manejo de puntos de trabajo y de postulaciones a estos puntos. Un Coordinador puede crear y eliminar puntos de trabajo, que podrán ser visualizados por los demás Coordinadores y Voluntarios, y en donde cada Voluntario puede crear una postulación a uno de estos puntos. El coordinador también puede iniciar un llamado a la actividad de uno de estos puntos, momento en que cada Voluntario revisa el resultado de la postulación, y luego finalizar la actividad de este punto. Se detalla el caso de uso en la Ilustración 3-4.

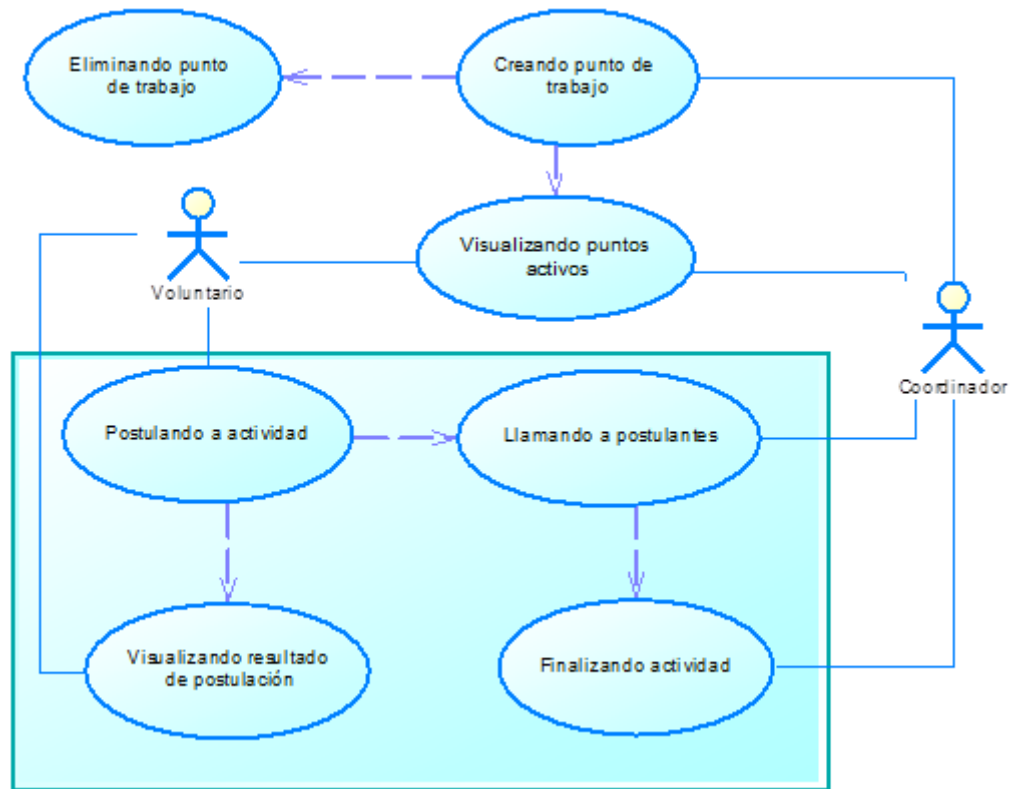


Ilustración 3-4: Diagrama de casos de uso: Gestión de puntos de trabajo

3.8 ANÁLISIS Y ESPECIFICACIÓN DE REQUISITOS

En esta sección se presenta la evolución de las vistas de la aplicación a través de *mockups*, mostrando los cambios de funcionalidades y diseño realizados de las 5 iteraciones realizadas, llegando a la versión final presentada en el capítulo de IMPLEMENTACIÓN. Desde la segunda iteración en adelante, se mostrarán solo los cambios realizados desde la iteración anterior. Además, se modela el sistema a través de un diagrama de clases conceptual en UML.

3.8.1 Primera Iteración

La primera iteración comprende los siguientes puntos:

- Construir la base de las vistas del prototipo.
- Hacer ingreso a una base de datos los datos de los usuarios.
- Hacer ingreso a una base de datos los datos del coordinador del sistema.

Siguiendo estos puntos, se crearon las primeras iteraciones para los prototipos de voluntario y de coordinador, en donde se implementó la base de las vistas del prototipo, con un menú lateral y pestañas para una de las vistas del prototipo del voluntario. Además, se permite manejar cuentas de usuario diferenciadas para ambos tipos de usuario:

Prototipo Usuario, 1ra Iteración

Vista de Login: Desde esta vista, representada en la Ilustración 3-5, el usuario puede iniciar sesión en el sistema, o crear una nueva cuenta de usuario voluntario. Desde esta Vista también puede dirigirse a la Vista de Registro.

Vista de Registro: Desde esta vista, representada también por la Ilustración 3-5, el usuario puede crear una nueva cuenta de usuario voluntario, ingresando datos como su correo electrónico, una contraseña, ambos nombres (segundo nombre opcional) y ambos apellidos, la cédula de identidad, y escoger un grupo o equipo de trabajo dentro de los que estén registrados en la base de datos. Presionando el botón *Registrarse*, se termina el registro, revisando si todos los campos fueron llenados, y se crea una nueva cuenta, devolviendo al usuario a la vista de *Login*.



Ilustración 3-5: Iteración Voluntario 1: Vista de Login y Vista de Registro

Vista Inicio: En esta vista, representada en la Ilustración 3-6, se muestra la vista principal de la aplicación para el voluntario. Presenta también las pestañas Misiones y Mensajes que integrarán funciones implementadas en las siguientes iteraciones.

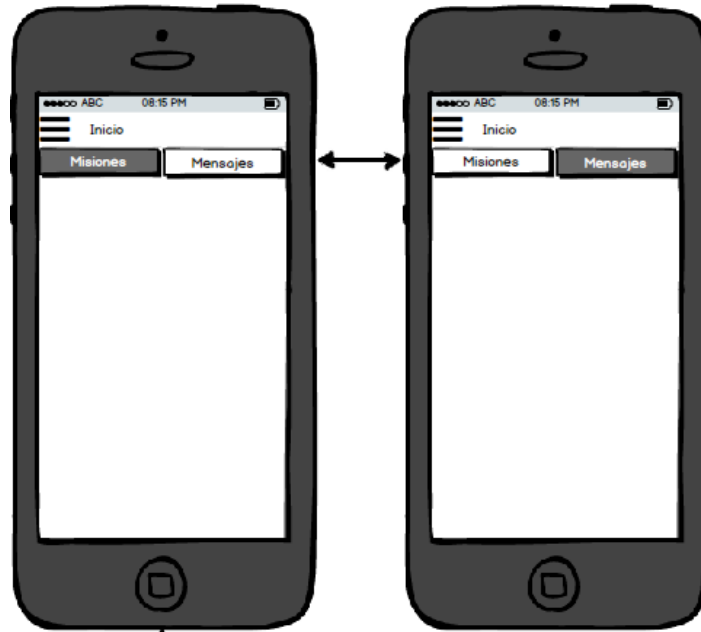


Ilustración 3-6: Iteración Voluntario 1: Vista Inicio

Vista Menú: Esta vista, representada en la Ilustración 3-7, muestra el menú que permite moverse por la aplicación a las demás vistas existentes en la aplicación, Inicio, Grupo y Perfil.

Vista Grupo: Esta vista, representada en la Ilustración 3-7, muestra la primera versión de la vista de Grupo. Permite al usuario visualizar el grupo o equipo con el que se identifica en el sistema.

Vista Perfil: Esta vista, representada en la Ilustración 3-7, muestra la primera versión de la vista de perfil, que permite visualizar los datos registrados por el usuario al crear una nueva cuenta. Muestra en pantalla información que puede ser vista públicamente como el nombre completo del usuario, y su correo electrónico.



Ilustración 3-7: Iteración Voluntario 1: Vista Menú, Vista Grupo y Vista Perfil

Prototipo Coordinador, 1ra Iteración

Vista de Login: Desde esta vista, representada en la Ilustración 3-8, el usuario puede iniciar sesión en el sistema, o crear una nueva cuenta de usuario coordinador.

Vista de Registro: Desde esta vista, representada por la Ilustración 3-8, el usuario puede crear una nueva cuenta de usuario coordinador, ingresando datos como su correo electrónico, una contraseña y un nombre completo. Presionando el botón *Registrarse*, se termina el registro, revisando si todos los campos fueron llenados, y se crea una nueva cuenta, devolviendo al usuario a la vista de *Login*.



Ilustración 3-8: Iteración Coordinador 1: Vista de Login y Vista de Registro

Vista Menú: Esta vista, representada en la Ilustración 3-9, muestra el menú que permite moverse por la aplicación a las demás vistas existentes en la aplicación, Inicio, Grupos (sin Vista aún), y Perfil.

Vista Inicio: En esta vista, representada en la Ilustración 3-9, se muestra la vista principal de la aplicación para el voluntario, que integrará funciones implementadas en iteraciones futuras.

Vista Perfil: Esta vista, representada en la Ilustración 3-9, muestra la primera versión de la vista de perfil, que permite visualizar los datos registrados por el usuario coordinador al crear una nueva cuenta. Muestra en pantalla información que puede ser vista públicamente como el nombre completo del usuario, y su correo electrónico.

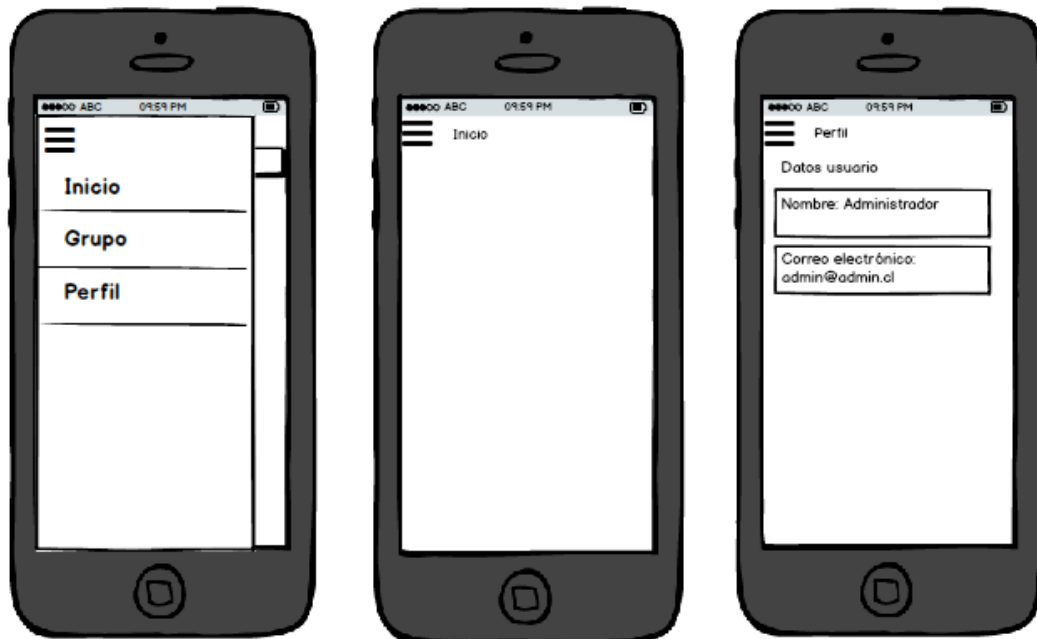


Ilustración 3-9: Iteración Coordinador 1: Vista Menú, Vista Inicio y Vista Perfil

3.8.2 Segunda Iteración

La segunda iteración comprende los siguientes puntos:

- Permitir la especificación y visualización de los puntos donde se espera que asistan los voluntarios.
- Permitir la especificación de la posición en el mapa de donde el usuario desea que se considere al ser llamado.

Siguiendo estos puntos, se crearon las segundas iteraciones para los prototipos de voluntario y de coordinador, en donde se implementó la primera versión del mapa de **Google Maps**, que permite que los usuarios voluntarios especifiquen una posición geográfica desde donde quieran ser considerados para los llamados. Además, permite también a los coordinadores crear y gestionar puntos de trabajo.

Prototipo Usuario, 2da Iteración

Vista de Registro: Desde esta vista, representada por la Ilustración 3-10, el usuario puede crear una nueva cuenta de usuario voluntario, ingresando datos como su correo electrónico, una contraseña, ambos nombres y ambos apellidos, la cédula de identidad, y escoger un grupo o equipo de trabajo dentro de los que estén registrados en la base de datos.

Respecto a la iteración anterior, ahora puede especificar una posición geográfica de donde será considerado cuando postule a una actividad, y permitir a la aplicación buscar la posición actual del dispositivo con el botón *Buscar mi posición*. Presionando el botón *Registrarse*, se termina el registro, revisando si todos los campos fueron llenados, y se crea una nueva cuenta, devolviendo al usuario a la vista de *Login*.



Ilustración 3-10: Iteración Voluntario 2: Vista de Registro

Prototipo Coordinador, 2da Iteración

Vista Inicio: En esta vista, representada en la Ilustración 3-11, se muestra la vista principal de la aplicación para el coordinador. Permite visualizar un mapa, donde el coordinador podrá crear un marcador para representar un punto de trabajo, donde especifica información como el nombre del punto de trabajo, los detalles de la actividad, la fecha de inicio, los días de duración de la actividad y la cantidad de voluntarios necesarios por día, y un botón para terminar de crear el marcador.



Ilustración 3-11: Iteración Coordinador 2: Vista Inicio

Vista Menú: Esta vista, representada en la Ilustración 3-12, muestra el menú que permite moverse por la aplicación a las demás vistas existentes en la aplicación. Ahora muestra también la posibilidad de acceder a la vista Actividades.

Vista Actividades: Esta vista, representada por la Ilustración 3-12, mostrará en futura iteraciones las actividades los puntos de trabajo registrados en el sistema por los coordinadores.



Ilustración 3-12: Iteración Coordinador 2: Vista Menú y Vista Actividades

3.8.3 Tercera Iteración

La tercera iteración comprende los siguientes puntos:

- Dar información sobre cada punto desplegado en el mapa a los usuarios.
- Permitir la elección de la tarea a realizar para cada usuario, especificando su disponibilidad de trabajo.

Siguiendo estos puntos, se creó la tercera iteración para el prototipo de voluntario. Se permite que los voluntarios vean la información sobre cada punto de trabajo, e identifiquen y escojan un punto de trabajo para trabajar. El prototipo para los coordinadores se mantuvo igual, por lo que no se especificará su iteración en esta sección.

Prototipo Usuario, 3ra Iteración

Vista Inicio: En esta vista, representada en la Ilustración 3-13, se muestra la vista principal de la aplicación para el voluntario. Presenta también las pestañas Misiones y Mensajes que integrarán funciones implementadas en las siguientes iteraciones. Respecto a la iteración anterior, en la pestaña de Misiones ahora se visualiza un mapa que muestra los marcadores de los puntos creados por los coordinadores, junto con la información referente a cada marcador. Además, permite al voluntario postular su participación a la actividad referida a ese marcador.



Ilustración 3-13: Iteración Voluntario 3: Vista Inicio

3.8.4 Cuarta Iteración

La cuarta iteración comprende los siguientes puntos:

- Permitir a los voluntarios postular su disponibilidad de tiempo para ser llamados a trabajar.
- Distribuir a los voluntarios postulantes en cada punto de trabajo de forma que haya un flujo constante de voluntarios durante el periodo especificado de trabajo, manteniendo una cantidad constante de voluntarios trabajando en el punto especificado.

Siguiendo estos puntos, se creó la cuarta iteración para el prototipo de voluntario y para el prototipo del coordinador. Se permite que los voluntarios vean la información sobre cada punto de trabajo, e identifiquen y escojan un punto de trabajo para trabajar, con la diferencia de que, en esta iteración, el voluntario especifica las fechas en que desea asistir y retirarse de la actividad. En la iteración del prototipo para los coordinadores, se implementa la función de llamar al algoritmo que recibe los datos de las postulaciones, y los ordena para determinar cuáles voluntarios asistirán a la actividad.

Prototipo Usuario, 4ta Iteración

Vista Inicio: En esta vista, representada en la Ilustración 3-14, se muestra la vista principal de la aplicación para el voluntario. Presenta también las pestañas Misiones y Mensajes que integrarán funciones implementadas en las siguientes iteraciones. En la pestaña de Misiones se visualiza un mapa que muestra los marcadores de los puntos creados por los coordinadores, junto con la información referente a cada marcador. Además, permite al voluntario postular su participación a la actividad referida a ese marcador, especificando las fechas de inicio y fin de sus trabajos.



Ilustración 3-14: Iteración Voluntario 4: Vista Inicio, pestaña Misiones

Prototipo Coordinador, 4ta Iteración

Vista Inicio: En esta vista, representada en la Ilustración 3-15, se muestra la vista principal de la aplicación para el coordinador. Permite visualizar un mapa, donde el coordinador podrá crear un marcador para representar un punto de trabajo, donde especifica información como el nombre del punto de trabajo, los detalles de la actividad, la fecha de inicio, los días de duración de la actividad y la cantidad de voluntarios necesarios por día, y un botón para terminar de crear el marcador. Respecto a versiones anteriores, ahora cada marcador creado muestra al ser presionado un enlace llamado “Ver”, que lleva al coordinador a una vista con información relativa a ese punto específico.

Vista Punto: Esta vista, representada por la ilustración 3-15, muestra al coordinador ver la cantidad de postulantes a la actividad del punto de trabajo escogido, y permite correr al algoritmo que realiza la elección de los postulantes que serán llamados a trabajar al punto de trabajo.



Ilustración 3-15: Iteración Coordinador 4: Vista Inicio y Vista Punto

3.8.5 Quinta Iteración

La quinta iteración comprende los siguientes puntos:

- Informar y notificar sobre el resultado de la postulación de cada usuario.
- Permitir al usuario encontrar el punto geográfico donde fue llamado, y calcular la ruta que debe tomar.
- Permitir al usuario modificar el equipo para el que trabajará.
- Implementar un sistema de *ranking* como parámetro de elección de voluntariado. Permitir al usuario saber su puntuación en *forma* de ranking dentro de la aplicación mediante un medidor gráfico.
- Mejorar la interfaz de la aplicación.

Siguiendo estos puntos, se creó la quinta iteración para el prototipo de voluntario y para el prototipo del coordinador. Se permite que los coordinadores hagan los llamados para los puntos creados por ellos, haciendo que el sistema escoja a los postulantes que vayan a

participar. Cuando una actividad finaliza, un valor Ranking irá subiendo para cada voluntario, según si participaron satisfactoriamente en la actividad. Para el prototipo del voluntario, éste podrá ver la ruta a tomar desde el punto geográfico actual, hasta el punto de trabajo al que debe asistir. Además, se agregaron y corrigieron algunos detalles de las interfaces.

Prototipo Usuario, 5ta Iteración

Vista Inicio: En esta vista, representada en la Ilustración 3-16, se muestra la vista principal de la aplicación para el voluntario. Para la representación de esta iteración, se obvió la interfaz original mostrada en Ilustraciones anteriores, ya que es la misma mostrada en la iteración anterior, y se muestra la interfaz en el momento en que el voluntario es escogido en una actividad. En la pestaña Misiones, se le muestra en un mapa el punto donde se realiza la tarea, y se le permite calcular la ruta desde el punto geográfico escogido al registrarse al sistema, hasta la posición en que se realizará la tarea del punto de trabajo. En la pestaña Mensajes, se visualiza la fecha en que el usuario fue llamado a trabajar, y un botón que vuelve a mostrar al usuario la posición del punto geográfico donde se encuentran las tareas a realizar.

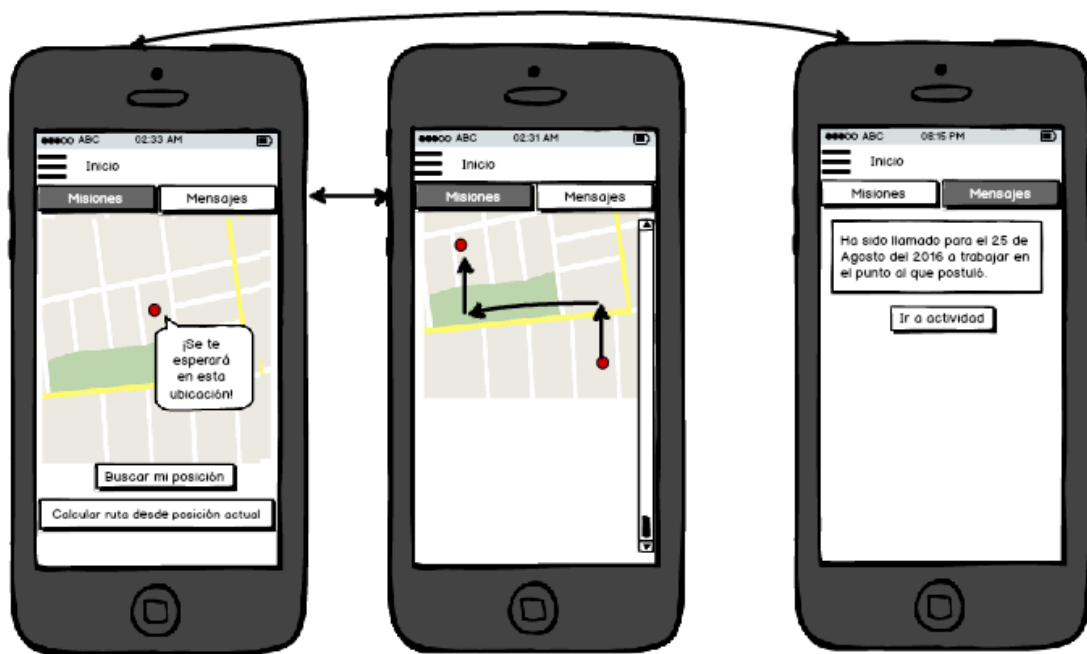


Ilustración 3-16: Iteración Voluntario 5: Vista Inicio/Llamado

Vista Grupo: Esta vista, representada en la Ilustración 3-17, muestra la vista de Grupo. Permite al usuario visualizar el grupo o equipo con el que se identifica en el sistema, y,

en esta iteración, modificar el equipo de trabajo escogiendo uno del menú desplegable, y presionando el botón *Aceptar cambio de equipo*.

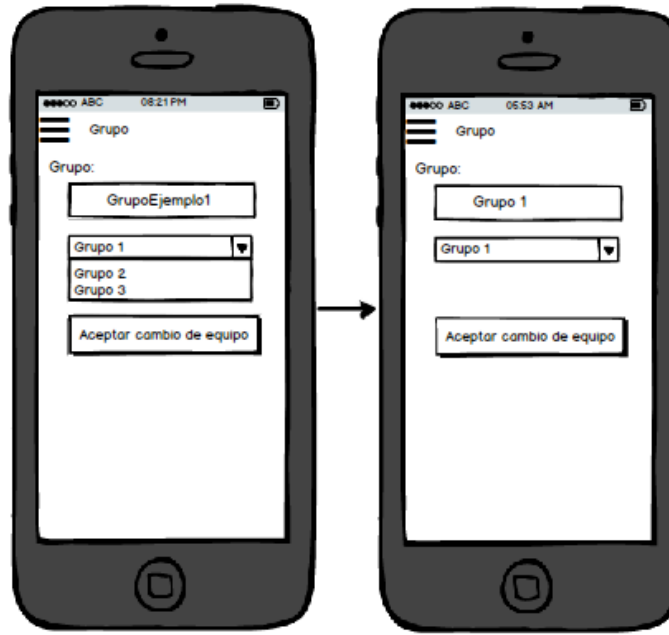


Ilustración 3-17: Iteración Voluntario 5: Vista Grupo

Vista Perfil: Esta vista, representada en la Ilustración 3-18, muestra la vista de perfil, que permite visualizar los datos registrados por el usuario al crear una nueva cuenta. Muestra en pantalla información que puede ser vista públicamente como el nombre completo del usuario, su correo electrónico, y la del punto geográfico escogido por el voluntario al registrarse al sistema. En esta iteración se muestra también el valor de *Ranking* y un gráfico visual que va mostrando un progreso según el valor de *Ranking*.

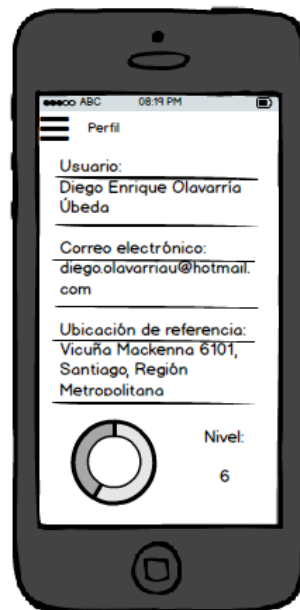


Ilustración 3-18: Iteración Voluntario 5: Vista Perfil

Prototipo Coordinador, 5ta Iteración

Vista Punto: Esta vista, representada por la ilustración 3-19, muestra al coordinador ver la cantidad de postulantes a la actividad del punto de trabajo escogido, y permite correr al algoritmo que realiza la elección de los postulantes que serán llamados a trabajar al punto de trabajo.

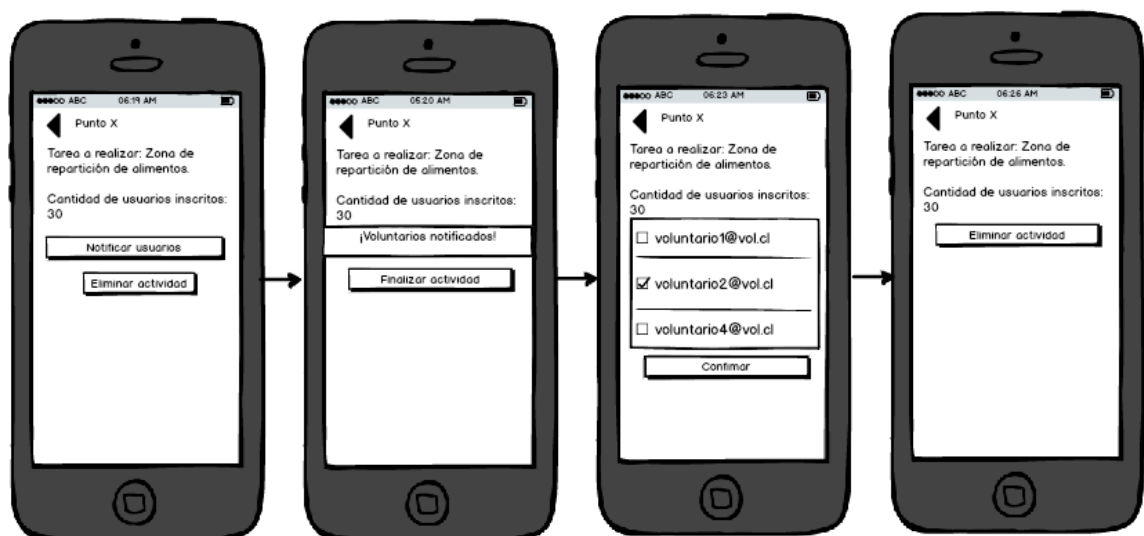


Ilustración 3-19: Iteración Coordinador 4: Vista Punto

Vista Grupos: Esta vista, representada por la Ilustración 3-20, ahora muestra los grupos y equipos de trabajo registrados, y el proceso de creación de éstos.

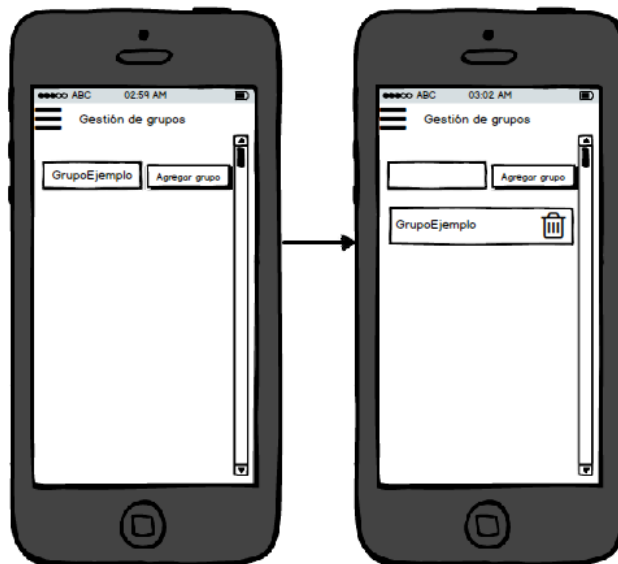


Ilustración 3-20: Iteración Coordinador 5: Vista Grupos

Vista Actividades: Esta vista, representada por la Ilustración 3-21, muestra las actividades los puntos de trabajo registrados en el sistema por los coordinadores.

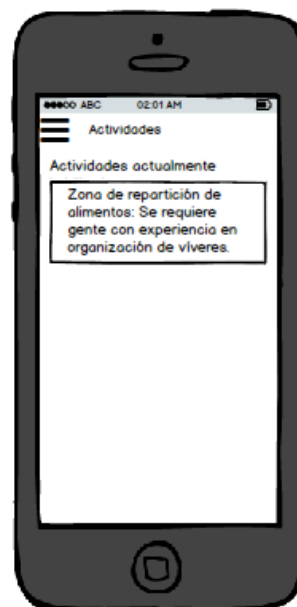


Ilustración 3-21: Iteración Coordinador 5: Vista Actividades

3.9 DISEÑO DEL SOFTWARE

3.9.1 Base de datos

Para el almacenamiento de datos se utilizó una base de datos no relacional, o NoSQL. El uso de una base de datos no relacional fue una decisión basada en las características que ofrece respecto a las bases de datos relacionales, que se aprecia en la Tabla 3-9:

Tabla 3-9: Comparación de bases de datos (Amazon, 2016)

Característica	Base de datos relacional	Base de datos NoSQL
Modelo de datos	Ordenan los datos en estructuras llamadas tablas, formadas por filas y columnas. Poseen un esquema que define tablas, columnas, índices, relaciones entre tablas, entre otros.	Suelen no contener un esquema, ya que utilizan claves para recuperar valores.
Propiedades ACID (Atomicidad: una transacción, en las bases de datos relacionales, se ejecutan completamente o no se ejecutan. Consistencia: Una vez que se ha ejecutado una transacción los datos se acoplan al esquema de la base de datos. Aislamiento: Las transacciones simultáneas se ejecutan por separado. Durabilidad: Capacidad de recuperarse de un error inesperado del sistema)	Las bases de datos relacionales se rigen por las propiedades ACID para su estructura.	Intercambian algunas de las propiedades ACID de las bases de datos relacionales por un modelado de datos más flexible y que se escala de forma horizontal (si se agregan más nodos, mejora el rendimiento). Mejor respuesta frente a una combinación de cuellos de botella de desempeño, escalabilidad, complejidad operativa y aumento de costos de administración y soporte.
Desempeño	Depende del subsistema de disco. Es necesario optimizar consultas, índices y estructura para obtener máximo desempeño.	Depende del tamaño del clúster de hardware subyacente, la latencia (retardos) de red y la aplicación que realiza la llamada.

Característica	Base de datos relacional	Base de datos NoSQL
Escalado	Para ampliar la escalabilidad se debe utilizar hardware más rápido. No poseen un sistema inherente distribuido, deben realizarse inversiones adicionales.	Utiliza clústeres distribuidos de hardware de bajo costo, aumentando el desempeño sin aumentar la latencia.
API (<i>Application Programming Interface</i>)	Utiliza un lenguaje de consultas estructurado (SQL , <i>Structured Query Language</i>) para almacenar y recuperar datos.	Utilizan API basadas en objetos, que permiten almacenar y recuperar estructuras de datos, mediante pares clave-valor.
Herramientas	Las herramientas que ofrecen simplifican el desarrollo de aplicaciones de bases de datos.	Ofrecen herramientas de administración de clústeres y escalado. Las aplicaciones representan la interfaz primaria de los datos subyacentes.

Las características mencionadas explican la decisión de escoger un modelo de base de datos no relacional. El uso de un sistema no esquemático permite acceder de manera más simple y rápida a la información para cada usuario, permitiendo a múltiples usuarios acceder e interactuar con los datos almacenados, ofreciendo una escalabilidad más estable y ágil.

El motor de base de datos escogido es **MongoDB**. Es una base de datos de código abierto que utiliza un modelo de datos orientado a documentos de pares clave-valor, con un formato de almacenamiento de documentos e intercambio de datos llamado **BSON**. Un único documento es capaz de almacenar toda la información necesaria sobre una colección. Soporta un diseño de esquema dinámico, que permite que los documentos en una colección tengan diferentes campos y estructuras (TechTarget, 2014).

Algunas de las principales características de **MongoDB** son:

1. **Ausencia de transacciones:** utiliza índices secundarios para las consultas. Le permite a **MongoDB** ser más rápido, y le da su propiedad de ser escalable a nivel horizontal.
2. **Escalabilidad horizontal.**

3. **BSON: Binary-JSON.** Formato de JSON que permite manejar datos binarios. **BSON** es un almacenamiento orientado a documentos. No es necesario especificar una estructura para los documentos. Usando BSON, **MongoDB** es más rápido en la búsqueda, indexación, almacenamiento y recuperación de la información.
4. **Soporta *queries* dinámicas:** A diferencia de otras bases de datos no-relacionales como **CouchDB**.
5. **Indexación de documentos:** Segura la unicidad de cada documento a través de una clave **_id**.
6. **GridFS:** Sistema proveído por **MongoDB** en caso de necesitarse espacio adicional.
7. **Replicación de datos:** Provee un mecanismo de replicación del tipo maestro-esclavo, lo que permite que si la base de datos principal (maestra) falla, la réplica (esclava) ocupa su lugar (Gracia, 2013).

3.9.2 Modelo de datos

En la Ilustración 3-22 se presenta el modelo conceptual de la base de datos. Se utiliza para determinar el dominio de la aplicación. Cada zona se identifica con 4 colores, y cada tabla se corresponde con los modelos de clases mostrados más adelante:

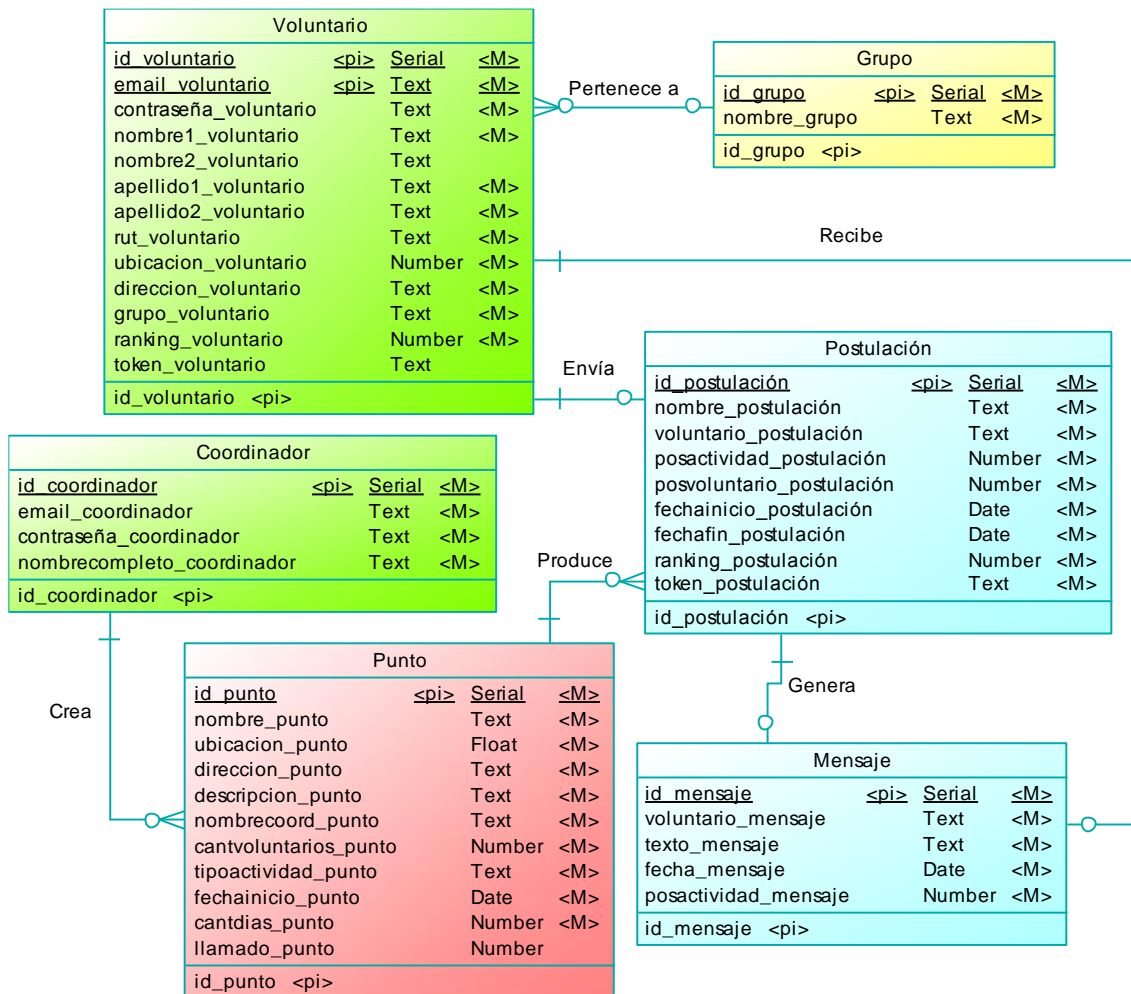


Ilustración 3-22: Modelo de datos

1. **Zona color verde:** Esta zona contiene a las tablas de datos que representan a los usuarios del sistema, Voluntarios y Coordinadores. Ambas tablas de datos contienen la información de perfil de sus respectivos usuarios, y están representadas por las Tablas 3-10 y 3-11.

Tabla 3-10: Atributos tabla Voluntario

Atributos VOLUNTARIO	Descripción
id_voluntario	Identificador único para cada voluntario en la base de datos.
email_voluntario	Correo electrónico del voluntario. Funciona también como identificador en la base de datos, y para ingresar a la cuenta del usuario.
contraseña_voluntario	Contraseña para la cuenta de usuario del voluntario.
nombre1_voluntario	Primer nombre del voluntario.
nombre2_voluntario	Segundo nombre del voluntario. Es opcional.
apellido1_voluntario	Primer apellido del voluntario.
apellido2_voluntario	Segundo apellido del voluntario.
rut_voluntario	R.U.T. del voluntario.
ubicacion_voluntario	Coordenadas que identifican la posición escogida por el voluntario para ser escogido por el sistema en el proceso de postulaciones.
direccion_voluntario	Ubicación en texto determinada por el punto escogido con <i>ubicación_voluntario</i> .
grupo_voluntario	Grupo o equipo de trabajo para el que el voluntario desea trabajar. Se copia de <i>nombre_grupo</i> .
ranking_voluntario	Valor que indica la cantidad de tareas realizada con éxito por el voluntario. Se utiliza también en el proceso de postulaciones.
token_voluntario	Valor identificador que identifica al dispositivo utilizado por el voluntario. Se utiliza para las notificaciones.

Tabla 3-11: Atributos tabla Coordinador

Atributos COORDINADOR	Descripción
id_coordinador	Identificador único para cada coordinador en la base de datos.
email_coordinador	Correo electrónico del coordinador. Funciona también como identificador en la base de datos, y para ingresar a la cuenta del usuario.
contraseña_coordinador	Contraseña para la cuenta de usuario del coordinador.
nombrecompleto_coordinador	Nombre completo del coordinador

2. **Zona color amarillo:** Esta zona contiene a la tabla de datos que representa a los grupos y equipos de trabajo registrados. Esta tabla de datos es manejada por los coordinadores, que pueden registrar y eliminar grupos y equipos de la base de datos, y está representada por la Tabla 3-12. Los voluntarios pueden escoger al

grupo con el cual desean representarse, a la vez de que sirve para tener mejor información de los registros de los voluntarios participantes.

Tabla 3-12: Atributos tabla Grupo

Atributos GRUPO	Descripción
id_grupo	Identificador único para cada grupo en la base de datos.
nombre_grupo	Nombre del grupo o equipo de trabajo.

3. **Zona color rojo:** Esta zona contiene a la tabla de datos que representa a los puntos de trabajo que se mostrarán en el mapa. Esta tabla de datos contiene información sobre cada punto, y está representada por la Tabla 3-13. Los coordinadores pueden crear y eliminar puntos, aunque cada coordinador solo puede eliminar los puntos creados por él.

Tabla 3-13: Atributos tabla Punto

Atributos PUNTO	Descripción
id_punto	Identificador único para cada punto en la base de datos.
nombre_punto	Título del punto. Indica a grandes rasgos el propósito del marcador.
ubicacion_punto	Coordenadas que identifican la posición escogida por el coordinador para posicionar el marcador del punto en el mapa.
direccion_punto	Ubicación en texto determinada por el punto escogido con <i>ubicación_punto</i> .
descripcion_punto	Descripción que permite especificar requisitos y características particulares del punto de trabajo, y así dar a conocer a los voluntarios que tipo de trabajo en específico se realizará
nombrecoord_punto	Correo electrónico del coordinador que creó el marcador. Se utiliza para que solo el coordinador que creó el marcador pueda trabajar con él.
cantvoluntarios_punto	Valor que indica la cantidad de voluntarios necesarios en cada día de trabajo.
tipoactividad_punto	Opción que permite saber al sistema qué ícono desplegar en el marcador para mostrar gráficamente que tipo de actividad se pide realizar.
fechainicio_punto	Valor que indica la fecha en que se iniciarán los trabajos.
cantdias_punto	Valor que indica cuántos días durarán los trabajos.
llamado_punto	Valor que indica si el coordinador ya realizó el llamado de los postulantes a trabajar en el punto.

4. **Zona color azul:** Esta zona contiene a las tablas que representan a las postulaciones de los voluntarios y los mensajes de aceptación. las postulaciones son creadas por cada voluntario cuando desea participar en un punto de trabajo. Los mensajes se crean al terminar el algoritmo de llamado de voluntarios. Ambas tablas conforman el proceso de llamado de voluntarios, y están representadas por las Tablas 3-14 y 3-15.

Tabla 3-14: Atributos tabla Postulación

Atributos POSTULACIÓN	Descripción
id_postulacion	Identificador único para cada postulación en la base de datos.
nombre_postulacion	Título de la postulación. Se copia de <i>nombre_punto</i> .
voluntario_postulacion	Correo electrónico del voluntario postulante. Se copia de <i>email_voluntario</i> .
posactividad_postulacion	Coordenadas que identifican la posición del punto de trabajo. Se copia de <i>ubicación_punto</i> .
posvoluntario_postulacion	Coordenadas que identifican la posición escogida por el voluntario al registrarse. Se copia de <i>ubicación_voluntario</i> .
fechainicio_postulacion	Valor que indica la fecha en que el postulante desea empezar a trabajar en el punto.
fechafin_postulacion	Valor que indica la fecha en que el postulante desea terminar su participación de los trabajos en el punto.
ranking_postulacion	Valor que indica la cantidad de tareas realizada con éxito por el voluntario. Se copia de <i>ranking_voluntario</i> .
token_postulacion	Valor identificador que identifica al dispositivo utilizado por el voluntario. Se copia de <i>token_voluntario</i> .

Tabla 3-15: Atributos tabla Mensaje

Atributos POSTULACIÓN	Descripción
id_mensaje	Identificador único para cada mensaje en la base de datos.
voluntario_mensaje	Correo electrónico del voluntario postulante. Se copia de <i>voluntario_postulación</i> .
texto_mensaje	Texto que indica al voluntario que su postulación fue aceptada, indicando también la fecha en que debe presentarse utilizando <i>fecha_mensaje</i> .
fecha_mensaje	Valor que indica la fecha en que el postulante debe presentarse a trabajar. Se utiliza <i>fechainicio_postulación</i> , pero puede verse aplazada según se determinó con el algoritmo.
posactividad_mensaje	Coordenadas que identifican la posición del punto de trabajo. Se copia de <i>posactividad_postulación</i> . Se utiliza para que el voluntario pueda también calcular la ruta desde su posición actual.

3.10 DIAGRAMAS DE ARQUITECTURA

3.10.1 Diagrama de clases

Las ilustraciones 3-23 y 3-24 presentan los diagramas de clases para los prototipos de Voluntario y Coordinador. Estos diagramas se utilizan para modelar las clases a implementar. Debido al tamaño de los diagramas, se mostrarán primero los diagramas completos para cada prototipo, y por secciones para su explicación.

En ambos diagramas, las zonas de color amarillo representan las tablas de la base de datos, explicadas en el capítulo anterior. Las zonas de color verde representan a los controladores de los prototipos, que manejan el comportamiento de los datos leídos y escritos en la base de datos, y mostrados y recibidos por las vistas de los prototipos, que conforman la comunicación entre aplicación y usuario, y que son representadas por las zonas de color rojo.

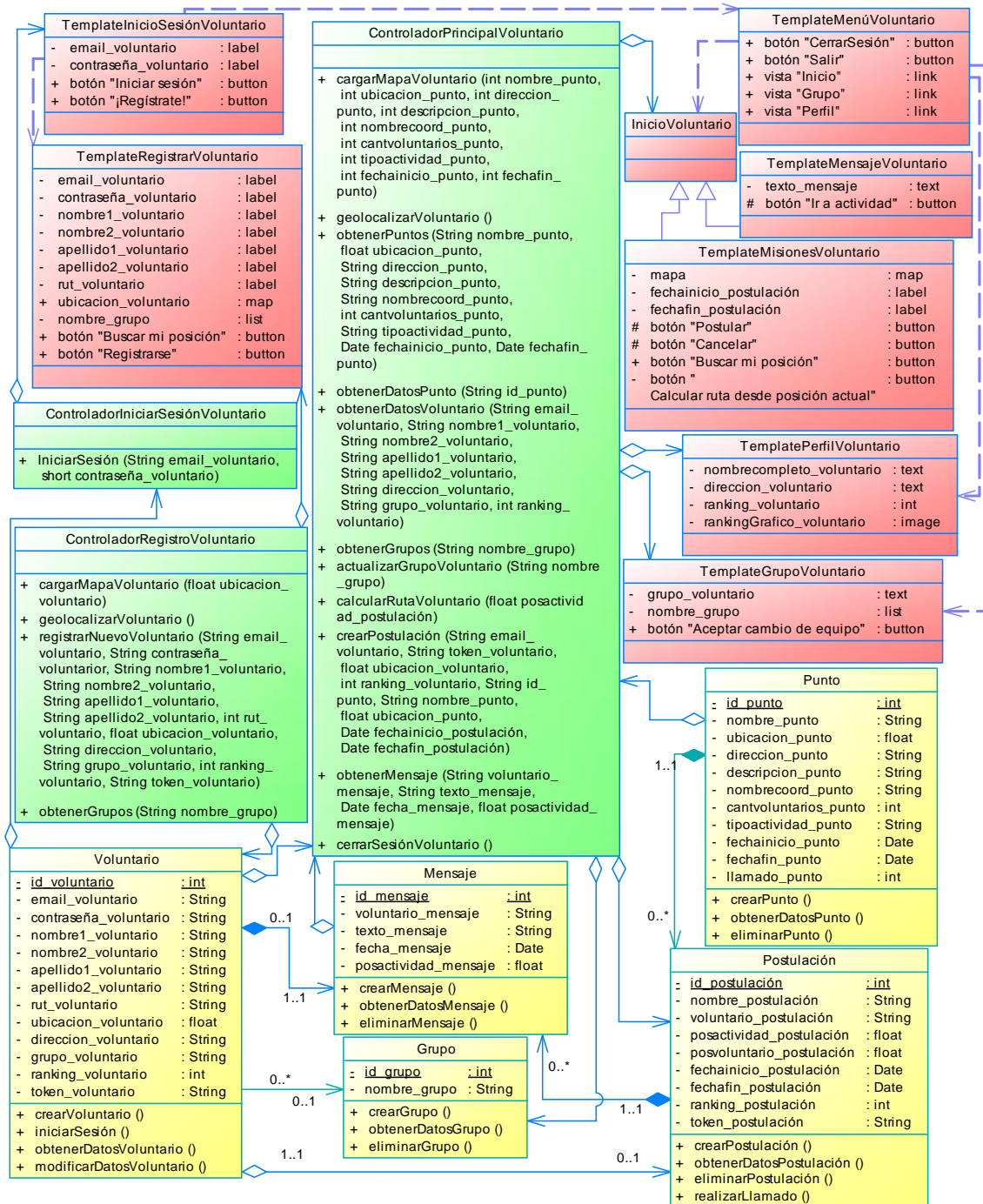


Ilustración 3-23: Diagrama de clases Voluntario

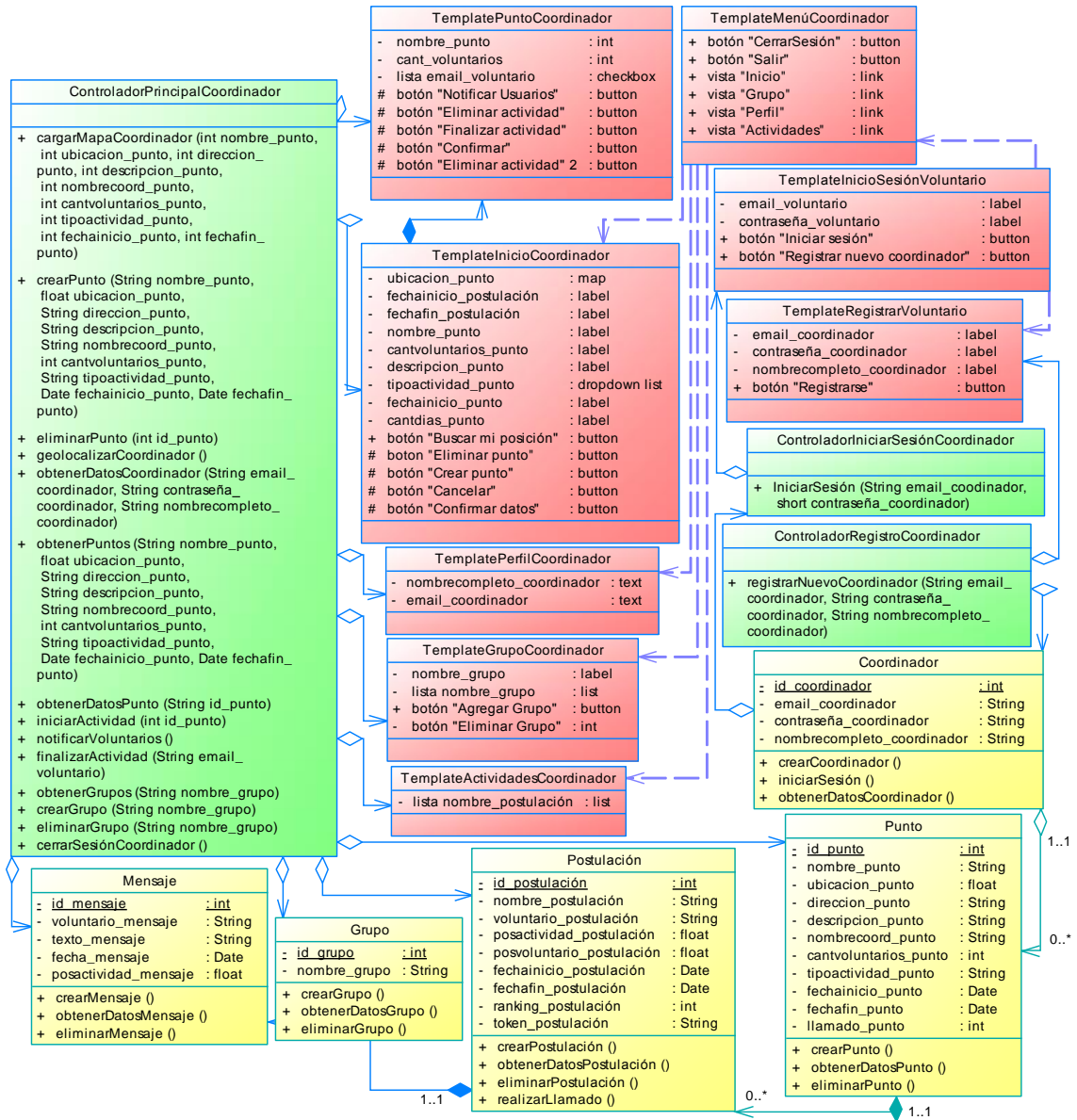


Ilustración 3-24: Diagrama de clases Coordinador

3.10.2 Diagrama de componentes

En la Ilustración 3-25 se muestra el diagrama de componentes del sistema, que muestra el sistema dividido en componentes, donde cada componente representa una parte física del sistema, y cómo estos componentes interactúan entre sí. Las líneas punteadas entre componentes muestran la dependencia que existe entre estos, siendo el lado de la flecha el componente dependiente.

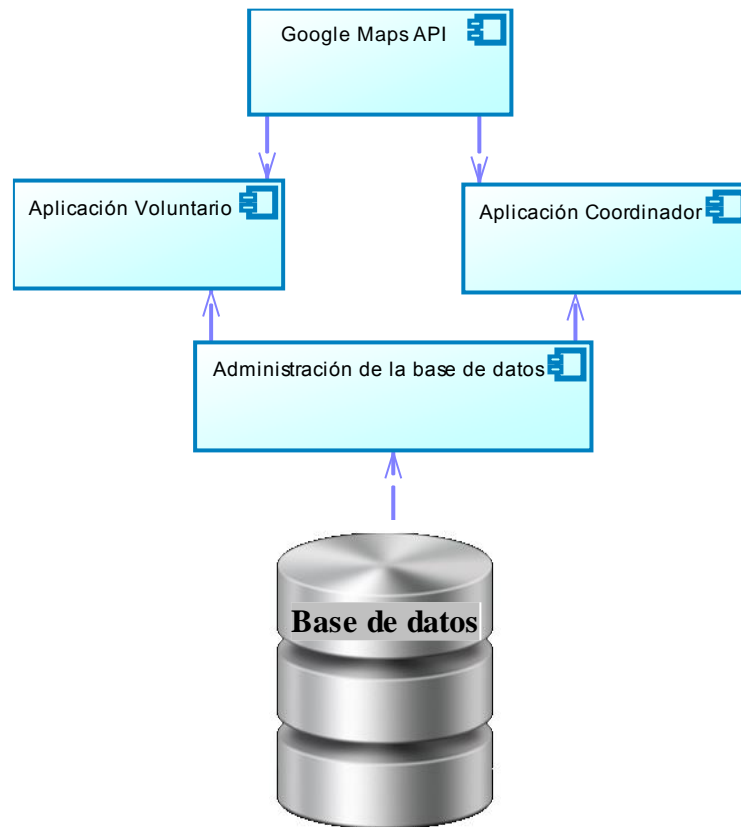


Ilustración 3-25: Diagrama de componentes

Se destaca la **API** de **Google Maps**, debido a la importación de sus rutinas y servicios para la creación de las funciones de ambos sistemas, y para el mapa mostrado en algunos de los *templates* de ambas aplicaciones.

3.10.3 Diagramas de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo, en base a los casos de uso especificados, mostrando los mensajes enviados y recibidos en el transcurso de una operación. En los diagramas de secuencia “*Llamando a postulantes*” y “*Finalizando actividad*” se asume que se cargaron los puntos y sus marcadores en el mapa y se comienza desde que se accede a la vista de un punto, para evitar crear un diagrama ilegible por la cantidad de información resultante en el espacio reducido del documento. Ver Anexo 8.1.

CAPÍTULO 4. IMPLEMENTACIÓN

En este capítulo se presenta el algoritmo a utilizar en la asignación de voluntarios para cada punto, y el resultado de la construcción del primer prototipo final. El objetivo de esta sección es explicar el funcionamiento en conjunto de este sistema.

4.1 ALGORITMO

El análisis del algoritmo de asignación de voluntarios se realiza a partir de las postulaciones realizadas por los voluntarios para cada actividad. Cada postulación registra consigo, como datos más relevantes, el correo electrónico de cada voluntario (debido a que son únicos dentro del sistema), las fechas de inicio y fin escogidas por el voluntario, la distancia entre el punto de trabajo y el punto especificado por el voluntario al registrarse, y su *ranking*.

El sistema toma en cuenta los dos últimos datos para el orden en que se debe analizar cada postulación, privilegiando el *ranking* sobre la distancia relativa, de manera que los voluntarios mejor valorados y más experimentados con el sistema sean los más factibles de ser llamados, tomando la distancia como valor secundario. Por ejemplo, dentro de tres postulaciones realizadas, si una corresponde a la de un voluntario que se encuentra a 300 kilómetros de distancia del punto de trabajo y posee un valor ranking de 7, otra postulación corresponde a un voluntario con un valor ranking de 3 y que se encuentre a 50 kilómetros de distancia, y la última postulación a la de un voluntario con valor ranking 7 y que se encuentre a 700 kilómetros de distancia, se le reservará primero un cupo al primer voluntario mencionado, luego al tercero, y finalmente al que posee valor ranking 3. Esto es porque se privilegia la búsqueda de voluntarios mejor valorizados respecto a voluntarios que se encuentren cerca de la localidad, que podrían no resultar en un mejor desempeño.

Sobre la asignación de cupos, se utilizan en primer lugar dos valores: La cantidad de días que durará la actividad, y la cantidad de voluntarios necesarios por día. Ambos valores son decididos por el coordinador al crear el punto de trabajo en el sistema, por lo que también requieren el estudio posterior a una catástrofe para ser especificados. Utilizando estos valores se genera un itinerario, representado en la Tabla 4-1:

Tabla 4-1: Itinerario modelo

Día Cupo	Día 1		Día 2		...		Día N	
Cupo 1	E (Fecha día 1)	S (Fecha día 1)	E (Fecha día 2)	S (Fecha día 2)	E (Fecha día 3)	S (Fecha día 3)	E (Fecha día 4)	S (Fecha día 4)
Cupo 2	E (Fecha día 1)	S (Fecha día 1)	E (Fecha día 2)	S (Fecha día 2)	E (Fecha día 3)	S (Fecha día 3)	E (Fecha día 4)	S (Fecha día 4)
...	E (Fecha día 1)	S (Fecha día 1)	E (Fecha día 2)	S (Fecha día 2)	E (Fecha día 3)	S (Fecha día 3)	E (Fecha día 4)	S (Fecha día 4)
Cupo M	E (Fecha día 1)	S (Fecha día 1)	E (Fecha día 2)	S (Fecha día 2)	E (Fecha día 3)	S (Fecha día 3)	E (Fecha día 4)	S (Fecha día 4)

M = Cantidad de voluntarios requeridos por día

N = Duración de la actividad en días

E = Entrada de turno
S = Salida de turno

Bajo cada “E” y “S” dice “*Fecha día X*”. Este valor se utiliza para comparar la posición en el itinerario con las fechas especificadas por los voluntarios en sus postulaciones, y también para determinar si tal día se encuentra libre o no. De estar ocupado, tendrá almacenado en esa posición el nombre del voluntario asignado en vez de una fecha.

En el Anexo 8.2 se presenta el algoritmo para escoger a postulantes y ordenarlos en el itinerario anterior. Este proceso se realiza con cada punto de trabajo, en el momento que lo indique el coordinador correspondiente.

En la primera parte del algoritmo, se declara el itinerario, y se procede a analizar una por una las postulaciones realizadas. Cuando se encuentra un cupo libre, se le reservan los días correspondientes al voluntario que envió la correspondiente postulación. En la segunda parte, se busca registrar en el Itinerario las postulaciones que no se asignaron en la primera parte debido a que no se encontraron periodos de tiempo que coincidan con las fechas postuladas, por lo que se resuelve ir descontando días de la postulación, significando en que tal voluntario llegue días más tarde a trabajar, para que su postulación puede ser registrada. Debido al orden en que se asignan las reservas, se asegura que a los postulantes con un mejor ranking, o sea, que tengan mayor prioridad para ser llamados, se les reserve un cupo.

El itinerario se va llenando según el orden explicado recién, y según las fechas especificadas por cada voluntario en sus postulaciones. Se proyecta que en un día algunos voluntarios se irán, por lo que debería llegar otro que lo suplante y así mantener la misma cantidad de voluntarios por día. Es por esto que se representa en el itinerario con las letras “E” y “S” en cada día, y que, en cada espacio de la matriz de 2x2 se crea un nuevo *array* de 2 celdas.

La tabla 4-2 muestra un ejemplo resultante de un cálculo realizado con dos postulantes en una actividad que requería 4 voluntarios por día, por 8 días comenzando desde el 14 de Junio del 2016. Se utilizó un ejemplo simple para poder apreciar de manera sencilla el procedimiento utilizado:

Voluntario 1: ejemplo@ejemplo.cl

- Fecha de inicio: 15 de Junio de 2016.
- Fecha de finalización: 19 de Junio de 2016.
- *Ranking*: 4.
- Distancia relativa aproximada: 14 kilómetros.

Voluntario 2: diego.olavarriau@hotmail.com

- Fecha de inicio: 16 de Junio de 2016.
- Fecha de finalización: 18 de Junio de 2016.
- *Ranking*: 2.
- Distancia relativa aproximada: 34 kilómetros.

Ambas postulaciones ya están ordenadas según lo realiza el algoritmo. Primero se analizará *ejemplo@ejemplo.cl* debido a poseer un nivel *ranking* mayor al de *diego.olavarriau@hotmail.com*. El itinerario resultante es representado en la Tabla 4-2:

Tabla 4-2: Ejemplo Itinerario 1

	14 de Junio		15 de Junio		16 de Junio		17 de Junio		18 de Junio		19 de Junio		20 de Junio		21 de Junio	
1	E	S	ejemplo	S	ejemplo	S	ejemplo	S	ejemplo	S	E	ejemplo	E	S	E	S
2	E	S	E	S	diego.olavarriau	S	diego.olavarriau	S	E	diego.olavarriau	E	S	E	S	E	S
3	E	S	E	S	E	S	E	S	E	S	E	S	E	S	E	S
4	E	S	E	S	E	S	E	S	E	S	E	S	E	S	E	S

Se aprecia cómo el primer voluntario en ser asignado a la tarea fue *ejemplo@ejemplo.cl* en la primera fila, asignándole las fechas concertadas en la postulación en el primer cupo. El 19 de Junio, día en que *ejemplo@ejemplo.cl* se retira de las labores, se posicionó al voluntario en el cuadro que antes tenía una “S” en vez de hacerlo en el cuadro con una “E” como en los demás días. Esto significa que, como lo estipuló el voluntario en su postulación, ese día terminará su turno, permitiendo que ese mismo día, si otro voluntario se hubiese postulado así, se cubra la entrada al turno.

En el caso del segundo voluntario, como su postulación comparte algunos días con su antecesor, se le asignó al segundo cupo, o sea, a la segunda fila.

La tabla 4-3 muestra un ejemplo resultante de un cálculo realizado con tres postulantes en una actividad que requería 3 voluntarios por día, por 7 días comenzando desde el 8 de Julio del 2016:

Voluntario 1: *ejemplo@ejemplo.cl*

- Fecha de inicio: 9 de Julio de 2016.
- Fecha de finalización: 14 de Julio de 2016.
- *Ranking*: 1.
- Distancia relativa aproximada: 34 kilómetros.

Voluntario 2: *diego.olavarriau@hotmail.com*

- Fecha de inicio: 9 de Julio de 2016.
- Fecha de finalización: 14 de Julio de 2016.
- *Ranking*: 4.
- Distancia relativa aproximada: 78 kilómetros.

Voluntario 3: *asist@trabajo.cl*

- Fecha de inicio: 8 de Julio de 2016.
- Fecha de finalización: 10 de Julio de 2016.
- *Ranking*: 4.
- Distancia relativa aproximada: 14 kilómetros.

Primero, se deben ordenar las postulaciones. Esto se realiza tomando los valores de *ranking* y la distancia, con el *ranking* como prioridad. En este caso, hay dos postulaciones de voluntarios con un nivel *ranking* igual a 4, mayor al de *ejemplo@ejemplo.cl* de nivel *ranking*

CAPÍTULO 4. IMPLEMENTACIÓN

igual a 2, por lo que se ordenarán desde la menor distancia al punto de trabajo. Las postulaciones serán analizadas en el siguiente orden:

Voluntario 1: `asist@trabajo.cl`.

- Fecha de inicio: 8 de Julio de 2016.
- Fecha de finalización: 10 de Julio de 2016.
- Ranking: 4.
- Distancia relativa aproximada: 14 kilómetros.

Voluntario 2: `diego.olavarriau@hotmail.com`

- Fecha de inicio: 9 de Julio de 2016.
- Fecha de finalización: 14 de Julio de 2016.
- *Ranking*: 4.
- Distancia relativa aproximada: 78 kilómetros.

Voluntario 3: `ejemplo@ejemplo.cl`

- Fecha de inicio: 9 de Julio de 2016.
- Fecha de finalización: 14 de Julio de 2016.
- *Ranking*: 1.
- Distancia relativa aproximada: 34 kilómetros.

El itinerario resultante es representado en la Tabla 4-3:

Tabla 4-3: Ejemplo Itinerario 2, Parte 1

	8 de Julio		9 de Julio		10 de Julio		11 de Julio		12 de Julio		13 de Julio		14 de Julio	
1	asist	S	asist	S	E	asist	E	S	E	S	E	S	E	S
2	E	S	diego.ola varriau	S	diego.ola varriau	S	diego.ola varriau	S	diego.ola varriau	S	diego.ola varriau	S	E	diego.ol avarriau

Se aprecia cómo el primer voluntario en ser asignado a la tarea fue `asist@voluntario.cl` en la primera fila, asignándole las fechas concertadas en la postulación en el primer cupo. En el caso del segundo voluntario, como su postulación comparte algunos días con su antecesor, se le asignó al segundo cupo, o sea, en la segunda fila. El tercer voluntario

CAPÍTULO 4. IMPLEMENTACIÓN

hace uso de la segunda parte del algoritmo al no encontrarse un cupo que coincida a primera vista con las fechas especificadas en la postulación. Para esto, el sistema toma en cuenta:

- La postulación de *ejemplo@ejemplo.cl* comprende 6 días de trabajo. Su distancia al punto es de 34 kilómetros, cantidad menor a 500 kilómetros, por lo que el mínimo de días que el voluntario podrá asistir es de 4 días (En el algoritmo se especifica que cada 500 kilómetros de distancia el mínimo de días al que se podrá reducir la postulación de un voluntario aumenta en uno)
- Existe un intervalo libre desde el 10 de Julio hasta el 14 de Julio, o sea, 5 días, mayor a 4 días.

En el caso de *ejemplo@ejemplo.cl*, como su intervalo de tiempo no coincide con algún tiempo libre, pero cumple las condiciones mencionadas anteriormente, recibirá una reducción de su tiempo, lo que significa que en vez de analizar su postulación como si empezara el 9 de Julio, será analizada como si empezara el 10 de Julio, para el que existe un intervalo disponible en la primera fila, con el resultado mostrado en el Itinerario de la Tabla 4-4:

Tabla 4-4: Ejemplo Itinerario 2, Parte 2

8 de Julio			9 de Julio		10 de Julio		11 de Julio		12 de Julio		13 de Julio		14 de Julio	
1	asist	S	asist	S	ejemplo	asist	ejemplo	S	ejemplo	S	ejemplo	S	E	ejemplo
2	E	S	diego.ola varriau	S	diego.ola varriau	S	diego.ola varriau	S	diego.ola varriau	S	diego.ola varriau	S	E	diego.ol avarriau

Como puede verse, solo fue necesario reducir un solo día a la última postulación para darle cabida a la postulación al itinerario. Se implementó que la reducción de días se hiciese solo desde el día de llegada y no en los días finales ya que de esta manera se le permite al voluntario un(os) día(s) más para planificar el trayecto al punto de trabajo.

Una vez escogidos los voluntarios postulantes que serán llamados, se envían notificaciones a éstos utilizando la información de los voluntarios almacenada en el *array* *Seleccionados*, utilizando el *token* o identificador del dispositivo para enviar una notificación a los usuarios correspondientes en caso de que haya sido escogido o no.

4.2 CONSTRUCCIÓN DEL SOFTWARE

El desarrollo del prototipo siguió la progresión mostrada en los *mockups* de la sección 3.7. Implicó construir dos versiones: una para el usuario común, que permita al voluntario registrar una cuenta de usuario, escoger el punto de trabajo al cual asistir, y ser llamado a un punto geográfico para trabajar; y otra versión para el coordinador, que le permita crear y gestionar puntos de trabajo, y llamar a los voluntarios elegidos por el sistema dentro de las postulaciones realizadas.

La revisión de las vistas de cada prototipo se hará mostrando las vistas principales de cada prototipo por separado, para luego complementar la exposición siguiendo un ejemplo de los casos de uso principales implementados en el sistema, especificando cuales vistas corresponden al prototipo creado para los usuarios, y cuales para el prototipo de los coordinadores. Cabe decir que las ilustraciones mostradas corresponden a imágenes tomadas con el prototipo ejecutándose desde un emulador. Esto es por la mayor facilidad que existe al tomar las imágenes desde un computador respecto a hacerlo desde un dispositivo móvil.

4.2.1 Interfaz

Para el desarrollo de la interfaz de la aplicación se siguió como pauta las Heurísticas de Nielsen (Nielsen, 1995):

- a) **Visibilidad del estado del sistema:** La visibilidad del estado del sistema se realiza a través de retroalimentación visual y auditiva. La realización exitosa de las funciones principales muestra como resultado final un mensaje en la pantalla.
- b) **Utilizar el lenguaje de los usuarios:** El sistema emplea palabras y frases del nivel de los usuarios. No utiliza mensajes del nivel de desarrollo.
- c) **Control y libertad para el usuario:** El usuario puede navegar con libertad entre las vistas de la aplicación, incluso cuando accede erróneamente a una vista puede volver fácilmente al estado anterior.
- d) **Consistencia y estándares:** El prototipo mantiene el mismo estándar de diseño en todas las vistas. De esta forma el usuario puede conocer el significado de los elementos de la interfaz en todas las vistas.
- e) **Prevención de errores:** La aplicación ha sido sometida a un plan de QA, para evitar la aparición de errores durante su uso.

- f) **Minimizar la carga de la memoria del usuario:** El prototipo hace uso de elementos distintivos para minimizar la información que debe recordar el usuario. Algunos ejemplos de esto son los marcadores mostrados en el mapa.
- g) **Flexibilidad y eficiencia de uso:** Dado que el resultado final de este trabajo sigue siendo un prototipo, el prototipo no permite la personalización de acciones por parte del usuario, por lo que esta heurística no se ha abordado, sin embargo, una vez terminado el proyecto en su totalidad el prototipo podría actualizarse para integrar nuevos módulos y aceleradores, y mejorar la velocidad de interacción entre los usuarios y el sistema.
- h) **Diálogos estéticos y diseños minimalistas:** En las vistas de la aplicación se muestra solo la información relevante para el usuario, como el mapa, y los botones se van mostrando según el usuario vaya realizando acciones en la aplicación.
- i) **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** Se han creado algunos mensajes de errores para responder frente a problemas de comunicación con el servidor. Sin embargo, solo se utilizan para notificar de errores, no ayudan al usuario a prevenirlos.
- j) **Ayuda y documentación:** Como guía para el usuario, se muestran pequeños textos en algunas vistas para asistir al usuario en las distintas funcionalidades de la aplicación.

4.2.2 Prototipo Voluntario

Las principales interfaces del prototipo construido para los usuarios son:

- Inicio de sesión
- Registro
- Inicio
 - Mapa
 - Mensaje
- Grupo de trabajo
- Perfil



Inicio de sesión

Voluntaris

Correo electrónico

Contraseña

Iniciar sesión

¡Regístrate!

Ilustración 4-1: Vista Inicio de sesión Voluntario

La Ilustración 4-1 muestra la primera vista presentada al usuario. Como fue especificado en los requerimientos, permite la opción de crear una nueva cuenta de voluntario, o iniciar sesión si el voluntario ya posee una cuenta. Iniciar sesión en el sistema requiere ingresar un correo electrónico que esté registrado en el sistema, así como de la contraseña asociada correspondiente, y presionar el botón *Iniciar sesión*. Se evitó la implementación de servicios como **Twitter** o **Facebook**, ya que hay veces en que, en estas plataformas, los usuarios no ingresan un nombre válido, mientras que este sistema debe funcionar con nombres completos para obtener registros correctos en el caso de cualquier dificultad en terreno. Si el voluntario no posee una cuenta de usuario y desea crear una cuenta propia, debe presionar el botón *Registrarse*, lo que dirigirá al voluntario a la vista de Registro de nuevo voluntario, como en la Ilustración 4-2.

Crear una nueva cuenta

Correo electrónico

Contraseña

Son tambien necesarios los siguientes datos!

1er nombre

2do nombre

Apellido paterno

Apellido materno

La cédula de identidad debe ir sin puntos ni guión, pero debe contener su dígito verificador.

11111111K

La dirección que escogas en el mapa se utilizará como punto de referencia para la búsqueda de voluntarios. Puedes escoger la ubicación de tu hogar o una ubicación en común con tus amigos o equipo deseado de trabajo. ESTE PUNTO NO PODRÁ SER MODIFICADO POSTERIORMENTE.

Buscar mi posición

Map

Satellite

Bombero Salas

Mo

Crear una nueva cuenta

La dirección que escogas en el mapa se utilizará como punto de referencia para la búsqueda de voluntarios. Puedes escoger la ubicación de tu hogar o una ubicación en común con tus amigos o equipo deseado de trabajo. ESTE PUNTO NO PODRÁ SER MODIFICADO POSTERIORMENTE.

Buscar mi posición

Map

Satellite

Bombero Salas

Mo

Seleccione Equipo: Equipo

Registrarse

Ilustración 4-2: Vista Registro nuevo Voluntario

La Ilustración 4-2 corresponde a la vista de registro de nuevo voluntario. Como se mencionó en los requerimientos, permite a los voluntarios ingresar su información personal para poder ser identificado dentro del sistema, y muestra un mapa que en el que el voluntario debe escoger un punto geográfico como punto de referencia para el momento en que se analice su postulación a un trabajo voluntario. La Ilustración 4-3 muestra un ejemplo del uso de la vista presentada en la Ilustración 4-2:

Crear una nueva cuenta

diego.olavarriau@hotmail.com

.....

Son tambien necesarios los siguientes datos!

Diego

Enrique

Olavarria

Úbeda

La cédula de identidad debe ir sin puntos ni guión, pero debe contener su dígito verificador.

179209411

La dirección que escogas en el mapa se utilizará como punto de referencia para la búsqueda de voluntarios. Puedes escoger la ubicación de tu hogar o una ubicación en común con tus amigos o equipo deseado de trabajo. ESTE PUNTO NO PODRÁ SER MODIFICADO POSTERIORMENTE.

Buscar mi posición

Map

Satellite

PE157 - Parada / Mall Florida Center, La Florida, La Florida, Región Metropolitana, Chile

Crear una nueva cuenta

La dirección que escogas en el mapa se utilizará como punto de referencia para la búsqueda de voluntarios. Puedes escoger la ubicación de tu hogar o una ubicación en común con tus amigos o equipo deseado de trabajo. ESTE PUNTO NO PODRÁ SER MODIFICADO POSTERIORMENTE.

Buscar mi posición

Map

Satellite

PE157 - Parada / Mall Florida Center, La Florida, La Florida, Región Metropolitana, Chile

Seleccione Equipo: Grupo 1

Registrarse

Ilustración 4-3: Ejemplo Registro nuevo Voluntario

La Ilustración 4-3 muestra un ejemplo de cómo se llena el formulario presente en las vistas de Registro con información. Todos los campos de texto requieren información, excepto el del segundo nombre, que es opcional. El punto geográfico especificado por el voluntario se utilizará como parámetro para la selección de voluntarios a trabajar en un punto de trabajo. El equipo de trabajo que escoja el voluntario se utilizará como información adicional al momento de ordenar la información existente de los asistentes en una zona de reconstrucción.

Al terminar de llenar el formulario y presionar el botón *Registrarse*, se notifica al voluntario del éxito en el registro, y se regresa automáticamente a la vista de Inicio de sesión, como en la Ilustración 4-4.



Ilustración 4-4: Registro exitoso de Voluntario

Una vez de vuelta en la vista de Inicio de sesión, el voluntario puede ingresar los datos ingresados al registrar la nueva cuenta para iniciar sesión en el sistema.

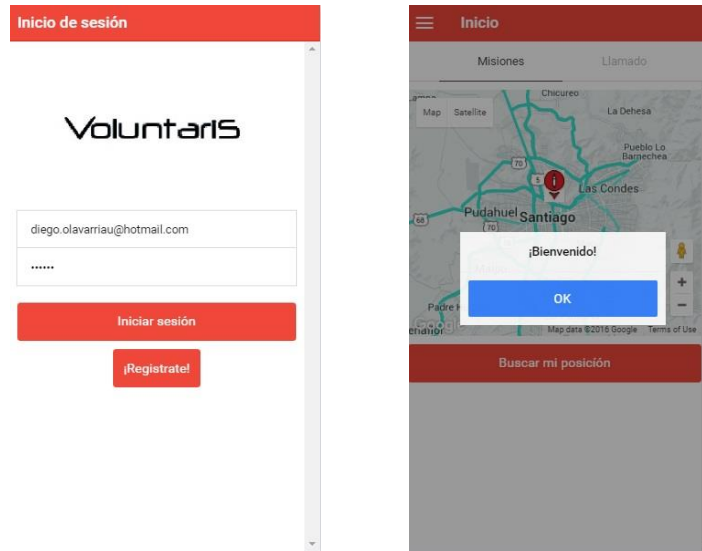


Ilustración 4-5: Ejemplo Inicio de sesión/Inicio de sesión exitoso de Voluntario

La Ilustración 4-5 muestra un ejemplo de cómo funciona el inicio de sesión con la cuenta de usuario creada como ejemplo. Al ingresar el correo electrónico y la contraseña y presionar el botón *Iniciar sesión*, la aplicación dirige automáticamente al voluntario a la vista Inicio, notificando del éxito en el inicio de sesión.

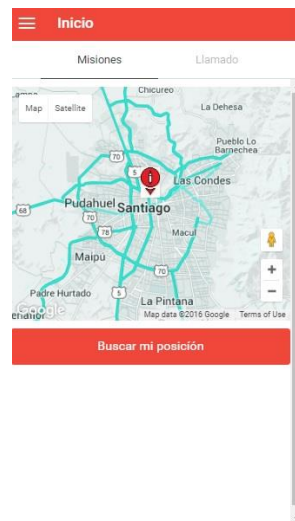


Ilustración 4-6: Vista Inicio Voluntario, pestaña Misiones

La Ilustración 4-6 muestra la vista principal del sistema del voluntario. Esta vista está dividida en dos pestañas: Misiones y Mensajes, siendo la pestaña de Misiones la que se muestra en la Ilustración 4-6. Esta pestaña muestra los puntos de trabajo creados por los

coordinadores en un mapa interactivo. El botón *Buscar mi posición* geolocaliza al voluntario en el mapa, mostrando su posición global. El papel de esta vista en el proceso de postulación para trabajos voluntarios se explicará más adelante.



Ilustración 4-7: Vista Inicio Voluntario, pestaña Mensajes

La Ilustración 4-7 ilustra la pestaña Mensajes, perteneciente a la vista Inicio. En esta vista, y en las demás vistas del prototipo, en la esquina superior izquierda, se encuentra un botón que sirve para desplegar un menú lateral. Este menú permite explorar las demás vistas del prototipo. Esta vista se utiliza para indicar al voluntario la fecha para la que fue llamado a trabajar en la actividad a la que haya postulado. Se explicará de esto más adelante.

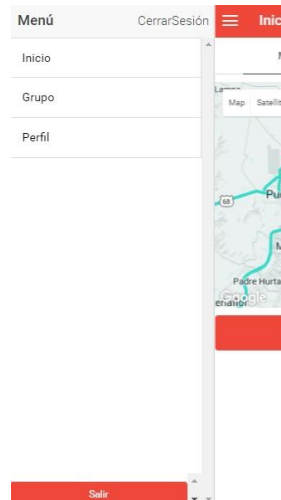


Ilustración 4-8: Vista Menú Voluntario

La Ilustración 4-8 muestra el menú que permite al voluntario explorar las vistas presentes en la aplicación. Cada opción, Inicio, Grupo y Perfil, lleva a una vista distinta. También tiene presentes los botones *Cerrar sesión*, en la parte superior del menú, que elimina la información local del usuario de la aplicación cuando es presionado, por consiguiente, cerrando la sesión de éste del sistema. El botón situado en la parte inferior, *Salir*, se utiliza para cerrar la aplicación, pero manteniendo la sesión abierta, permitiendo al usuario no tener que iniciar sesión cada vez que desee abrir la aplicación. Presionar las opciones Inicio, Grupo o Perfil llevará al voluntario a las vistas Inicio, Grupo de trabajo, o Perfil respectivamente. Este menú puede también ser desplegado desde las demás vistas, así como de ambas pestañas de la vista Inicio.



Ilustración 4-9: Vista Grupo de trabajo Voluntario

La Ilustración 4-9 muestra la vista de Grupo de trabajo del voluntario. En esta vista se muestra el grupo o equipo de trabajo en el cual el voluntario está registrado actualmente. Se le permite modificar el grupo en el cual quiere estar registrado seleccionando una de las opciones presentes en el menú desplegable (Estas opciones son gestionadas por los coordinadores del sistema), y presionando en el botón *Aceptar cambio de equipo*.

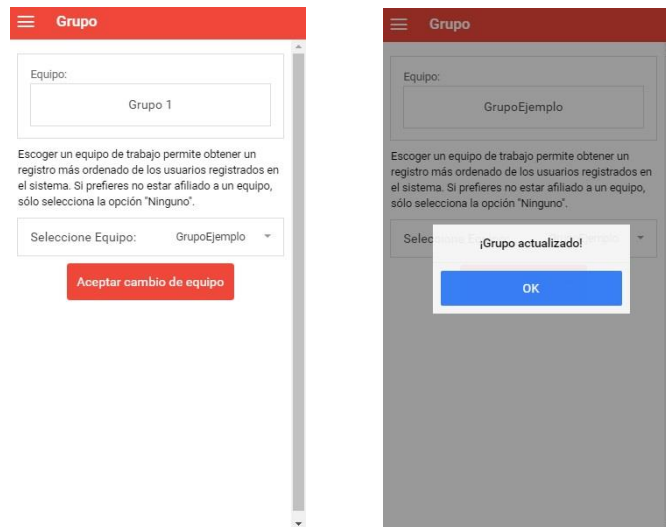


Ilustración 4-10: Escoger nuevo grupo Voluntario

En la Ilustración 4-10 se ve el proceso de cambio de grupo o equipo de trabajo: El usuario escoge uno de los grupos disponibles, siendo en este caso **GrupoEjemplo**, aunque también se le presenta al usuario la opción **Ninguno**, y presiona el botón *Aceptar cambio de equipo*, notificando al usuario del cambio exitoso de equipo.

Escoger un grupo o equipo permite al sistema mantener un registro más metódico de los usuarios registrados.



Ilustración 4-11: Vista Perfil Voluntario

La Ilustración 4-11 muestra la vista de Perfil de voluntario, donde se despliegan los datos ingresados por el usuario al registrarse en el sistema, su nombre completo, el correo electrónico utilizado como identificador de la cuenta del usuario, y la ubicación escogida por el usuario como punto de referencia (Es posible que esta ubicación no indique la dirección exacta del punto escogido).

Más abajo, se muestra gráficamente el ranking del voluntario en el sistema. Este valor simboliza el prestigio que ha alcanzado el voluntario, aumentando al participar exitosamente en las actividades presentes en el mapa. Al ir aumentando el ranking del voluntario, el disco irá mostrando un progreso decimal, es decir, irá completando la circunferencia con un color específico hasta alcanzar un ranking múltiplo de diez, cambiando de color siguiendo un patrón de arcoíris.

4.2.3 Prototipo Coordinador

Las principales interfaces del prototipo construido para el(los) coordinadores(es) son:

- Inicio de sesión
- Registro
- Inicio
- Gestión de actividades
- Gestión de grupos de trabajo
- Perfil

Las vistas de Inicio de sesión y de Registro poseen las mismas funciones que las del prototipo de voluntarios, pero el registro de nuevos coordinadores solo involucra el nombre, un correo electrónico y una contraseña.

Cuando un coordinador inicia sesión, la aplicación dirige automáticamente al coordinador a la vista Inicio, notificando del éxito en el inicio de sesión.



Ilustración 4-12: Vista Inicio Coordinador

La Ilustración 4-12 muestra la vista de Inicio del prototipo del coordinador. Esta pestaña muestra un mapa donde los coordinadores pueden crear, gestionar y eliminar puntos de trabajo. El botón *Buscar mi posición* geo-localiza al coordinador en el mapa, mostrando su posición global. El papel de esta vista en el proceso de postulación para trabajos voluntarios se explicará más adelante.

En esta vista, en la esquina superior izquierda, se encuentra un botón que sirve para desplegar un menú lateral. Este menú permite explorar las demás vistas del prototipo:

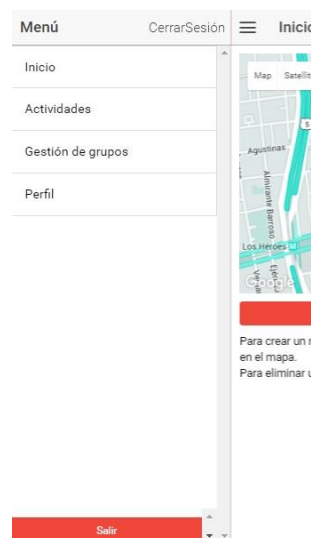


Ilustración 4-13: Vista Menú Coordinador

La Ilustración 4-13 muestra el menú que permite al Coordinador explorar las vistas presentes en la aplicación. Cada opción, Inicio, Actividades, Gestión de grupos y Perfil, lleva a una vista distinta. También tiene presentes los botones *Cerrar sesión*, en la parte superior del menú, que elimina la información almacenada en el dispositivo del coordinador de la aplicación cuando es presionado, cerrando la sesión de éste del sistema. El botón situado en la parte inferior, *Salir*, se utiliza para cerrar la aplicación, pero manteniendo la sesión abierta, permitiendo al usuario no tener que iniciar sesión cada vez que desee abrir la aplicación.

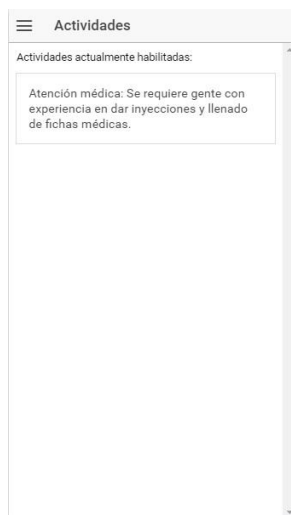


Ilustración 4-14: Vista Gestión de Actividades Coordinador

La Ilustración 4-14 muestra la vista de gestión de actividades actualmente habilitadas. Lista los puntos de trabajo creados en el sistema por los coordinadores, mostrando el título y la descripción de cada punto.



Ilustración 4-15: Vista Gestión de Grupos Coordinador

La Ilustración 4-15 muestra la vista de Gestión de grupos. Se listan los grupos y equipos registrados en el sistema, cada uno con la opción de ser eliminado del registro. En la parte superior de la vista, se encuentra una sección de entrada de texto que permite ingresar nuevos grupos al sistema, presionando el botón *Agregar Grupo*. Al agregar un nuevo grupo, se notifica del éxito al coordinador, como en la Ilustración 4-16.



Ilustración 4-16: Vista nuevo Grupo creado Coordinador

Una vez que un nuevo grupo fue creado, la lista se actualiza mostrando también el nuevo grupo registrado, como en la Ilustración 4-17., que también muestra el grupo recién creado siendo eliminado del registro.

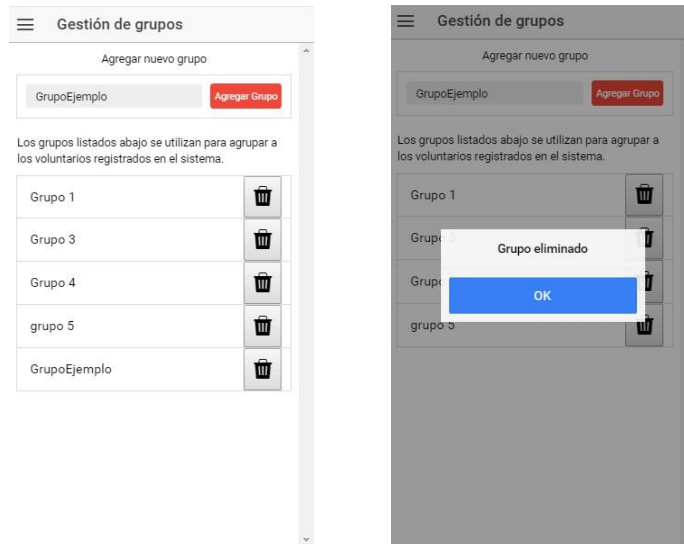


Ilustración 4-17: Grupos actualizados/Grupo ejemplo eliminado Coordinador

Para eliminar uno de los grupos, como en la Ilustración 4-17, se busca el grupo a eliminar, y se presiona el ícono ubicado a su lado. Si se ha eliminado correctamente, la aplicación notifica al coordinador de su éxito, actualizando la lista de grupos.

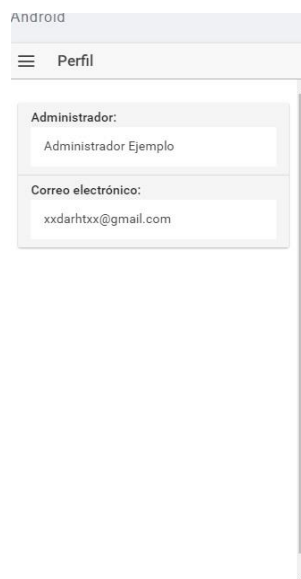


Ilustración 4-18: Vista Perfil Coordinador

La Ilustración 4-18 muestra la vista del Perfil de los coordinadores, que despliega los datos ingresados por el coordinador al registrarse en el sistema, su nombre completo y su correo electrónico.

4.2.4 Proceso de selección

La asignación de trabajo comienza con un coordinador creando un punto de trabajo en una ubicación a elección, como en la Ilustración 4-19:

Ilustración 4-19: Coordinador - creando punto de trabajo

Para la creación de un nuevo punto de trabajo, un coordinador debe especificar una ubicación en el mapa presionando el mapa de la vista de Inicio, y especificando los detalles del punto de trabajo en el formulario que se mostrará en consecuencia. Este formulario requiere los siguientes datos:

- Tarea requerida: Un nombre que sintetice de qué se trata la tarea, funcionará como el título de la actividad.
- Voluntarios requeridos por día: Un número que indique la cantidad de personas necesarias por día para trabajar en la actividad.
- Descripción: Escribir los detalles y requerimientos de la actividad que no puedan ser resumidos en el título de la actividad. También sirve para informar a los voluntarios cuales son las aptitudes específicas que se les requiere de ellos.
- Rubro: Especificar el rubro de la actividad permite categorizar el tipo de trabajo a realizar. El rubro se utiliza también para indicar al sistema qué marcador utilizar para mostrar el punto en el mapa.
- Fecha de inicio de la actividad: Fecha en que se espera que inicien los trabajos. Esta fecha debe ser posterior al día en que se cree el punto.

- Días de trabajo: Un número que indique cuantos días durarán los trabajos.

Una vez que se llenen todos los campos y se presione el botón *Confirmar datos*, el punto será registrado y desplegado en el mapa junto con los puntos creados anteriormente, y se notificará del éxito de esta operación al coordinador como en la Ilustración 4-20.

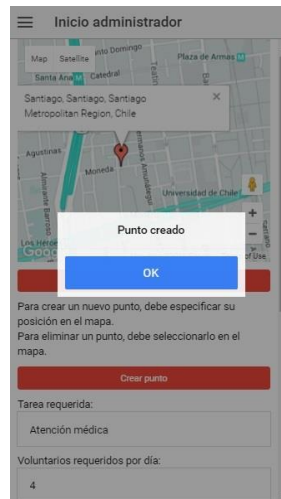


Ilustración 4-20: Coordinador - Nuevo punto creado

Al ser creado y registrado el nuevo punto, además de verlo en el mapa, se verá listado en la vista de Gestión de actividades, como en la Ilustración 4-21.



Ilustración 4-21: Coordinador: Listado de actividades

Los puntos creados se mostrarán también en la vista de Inicio de los usuarios, como en la Ilustración 4-22. Cada punto está a cargo del coordinador que creó el punto, lo que quiere decir que solo dicho coordinador puede realizar el llamado a trabajar a ese punto, y solo dicho coordinador puede eliminar ese punto.



Ilustración 4-22: Voluntario - Escogiendo punto de trabajo

En la Ilustración 4-22, se muestra como un usuario puede visualizar en el mapa de su interfaz el punto recién creado por el coordinador. Presionando el marcador, se abre una pestaña explicativa con los datos del punto de trabajo, permitiendo al usuario informarse sobre los puntos a los que puede asistir. Cuando el voluntario decida asistir a uno de los puntos de trabajo y presione en el marcador correspondiente a este punto, se mostrará el botón *Participar*, y al presionar este botón, aparecerá un formulario como en la Ilustración 4-23.

The image shows two side-by-side screenshots of a mobile application interface. Both screens display a map with a location pin and a calendar for June 2016. The calendar is a grid with days of the week (lun, mar, mié, jue, vie, sáb, dom) and dates. The date 14 is highlighted in blue. Below the calendar, there are input fields for 'dd-mm-aaaa' and buttons for 'Postular' and 'Cancelar'.

Atención médica

Se requiere gente con experiencia en dar inyecciones y llenado de fichas médicas.

Cantidad de voluntarios necesarios por día: 4

Inicio: 14 de Junio de 2016 (7 días)

Junio-2016

lun.	mar.	mié.	jue.	vie.	sáb.	dom.
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

dd-06-2016

Fin:

dd-mm-aaaa

Postular **Cancelar**

Ilustración 4-23: Voluntario - Escogiendo fechas de trabajo

En la Ilustración 4-23 se muestra un pequeño formulario donde el usuario puede escoger las fechas en que puede comenzar y terminar su trabajo voluntario. Las fechas disponibles para asistir dependerán de lo especificado por el coordinador al crear el punto. El voluntario postulante no puede escoger una fecha de inicio anterior a la fecha de inicio de la actividad, ni una fecha de término posterior a la cantidad de días que durarán los trabajos.

Al terminar de escoger las fechas en que el voluntario desea asistir al punto de trabajo, se le presentan los botones *Postular* y *Cancelar*. El botón *Cancelar* devolverá la vista a su estado original, permitiendo al usuario escoger otro punto. Si el usuario está seguro de su postulación, presionará el botón *Postular* y será notificado de su acción, como en la Ilustración 4-24.

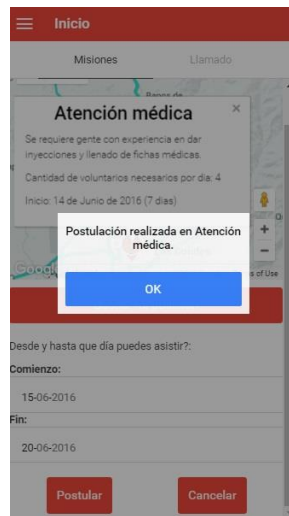


Ilustración 4-24: Voluntario - Éxito en postulación

Una vez realizada la postulación, se registrará en el sistema, junto con las postulaciones realizadas por los demás usuarios.

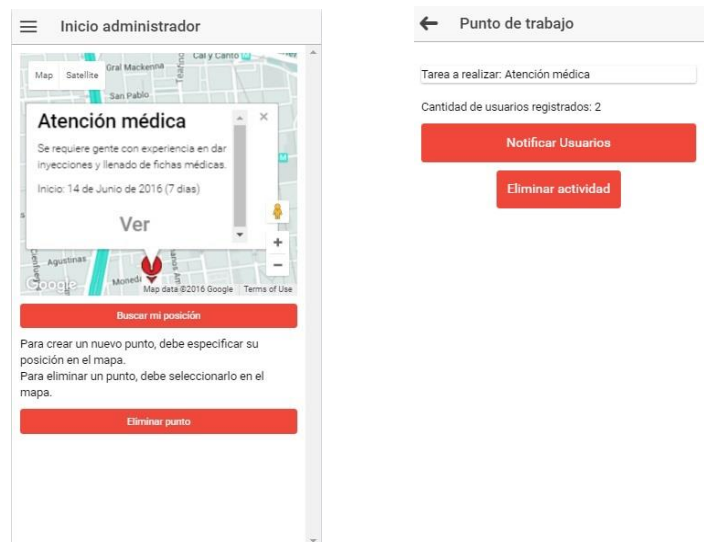


Ilustración 4-25: Coordinador - Revisión de punto de trabajo/Vista punto de trabajo

Quando el coordinador decida que es momento de hacer el llamado de voluntarios postulantes a trabajar, debe presionar el marcador, haciendo que se despliegue una pestaña con un resumen de la información sobre el punto de trabajo respectivo, y un hipervínculo en *Ver*. Presionando en *Ver*, se lleva al coordinador a una vista personalizada para cada punto, que muestra el nombre de la actividad, la cantidad de voluntarios que han postulado a la

actividad, y los botones *Notificar Usuarios* y *Eliminar actividad*. Si otro coordinador intentase acceder a esta vista de punto de trabajo, no podría hacerlo, ya que el hipervínculo *Ver* solo aparecerá para el creador de cada punto.

Al presionar en *Notificar Usuarios*, el sistema analiza las postulaciones de los voluntarios a ese punto en específico utilizando el algoritmo explicado anteriormente, y envía un mensaje a los voluntarios escogidos, como el que se ve en la Ilustración 4-28, además de notificar al coordinador que el análisis se completó, como se ve en la Ilustración 4-26.



Ilustración 4-26: Coordinador - Notificar postulantes

Además, se le envía un correo electrónico al coordinador a la dirección que él utilizó cuando se registró en el sistema, como en la Ilustración 4-27.



Ilustración 4-27: Coordinador - Resultados en correo electrónico

En la Ilustración 4-27, se ve el correo electrónico recibido por el coordinador del sistema, donde puede ver los correos de los voluntarios que fueron escogidos para trabajar, la fecha a la que deben asistir a las labores, y el orden en que fueron escogidos. Esto es para ser utilizado al avisarle al sistema sobre el término de una actividad, permitiendo al coordinador comparar el registro de voluntarios escogidos respecto a los que asistieron a trabajar, y puntuando su participación en la actividad. Esto se explicará con mayor claridad al llegar a la Ilustración 4-31.

Habiéndose realizado el llamado de voluntarios, los voluntarios recibirán una notificación a su dispositivo avisando si su postulación fue aceptada o no, como se muestra en la Ilustración 4-28.

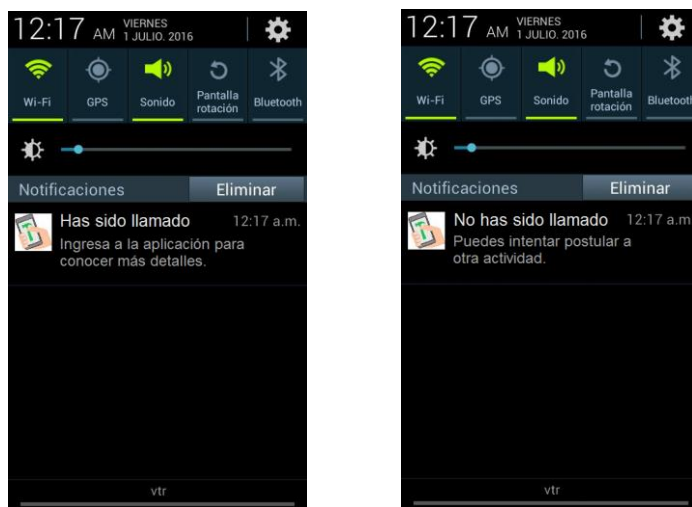


Ilustración 4-28: Voluntario – Notificaciones éxito/no éxito

Si fueron escogidos para trabajar, verán en el mapa de su aplicación el punto al cual deberán asistir, y al moverse a la pestaña Llamado en la vista Inicio verán el mensaje recibido que indica la fecha en que deben presentarse, y un botón que permite encontrar nuevamente el punto de trabajo si es que el usuario lo desea, como se ilustra en la Ilustración 4-29. Si no fueron escogidos, se notifica de esta situación, y se les permite escoger otra actividad.

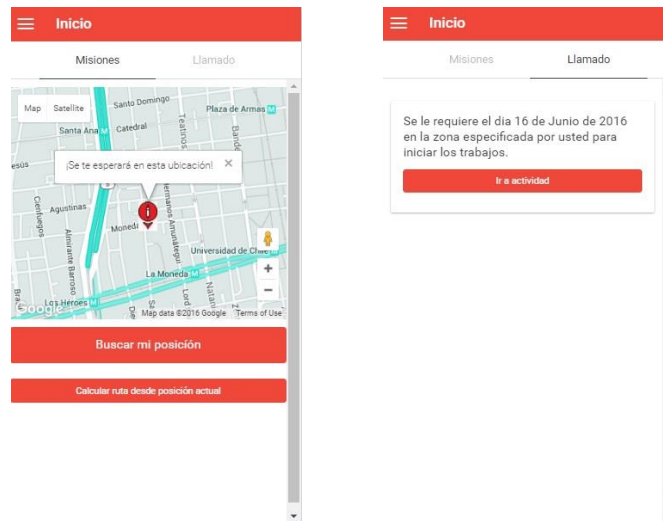


Ilustración 4-29: Voluntario – Llamado a trabajar/Mensaje de llamado

Se facilita también la función de calcular una ruta para llegar desde el punto geográfico actual donde se encuentre el voluntario, hasta el punto de trabajo. Esta ruta se calcula presionando el botón *Calcular ruta desde posición actual*. Un ejemplo se ve en la Ilustración 4-30.



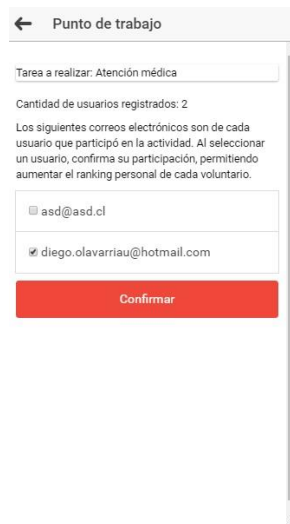
Ilustración 4-30: Voluntario - Detalles y cálculo de ruta

Así como, en la Ilustración 4-30, se ve la ruta a tomar y el transporte público disponible para recorrer esa ruta, el usuario también puede aumentar el zoom y analizar la ruta, viendo exactamente los pasos a seguir para llegar al punto de trabajo.

Hasta ahora, se ha explicado el proceso para iniciar los trabajos en un punto especificado por el coordinador en el mapa, por lo que queda esperar a terminar los trabajos para pasar a los pasos siguientes.

4.2.5 Proceso de finalización de una actividad

Finalizar una actividad implica hacer una retroalimentación sobre la participación del voluntariado en la actividad realizada. Dentro de esta aplicación, esto significa aumentar el ranking de los voluntarios participantes, aumentando su ranking en el sistema, haciéndolos más valiosos para llamados futuros. Rankear a los voluntarios se realiza siguiendo los mismos pasos vistos en la Ilustración 4-25, solo que en este caso se muestra algo parecido a la Ilustración 4-26, con el botón *Finalizar actividad* en la pantalla, el cual al presionarse lleva al coordinador a una vista similar a la de la Ilustración 4-31.



The screenshot shows a mobile application interface titled "Punto de trabajo". At the top, there is a back arrow and the title. Below the title, there is a text input field containing "Tarea a realizar: Atención médica". Underneath, it says "Cantidad de usuarios registrados: 2". A paragraph of text explains the process: "Los siguientes correos electrónicos son de cada usuario que participó en la actividad. Al seleccionar un usuario, confirma su participación, permitiendo aumentar el ranking personal de cada voluntario." Below this text is a list of two email addresses: "asd@asd.cl" and "diego.olavarriau@hotmail.com". Each email has a checkbox next to it. The checkbox for "diego.olavarriau@hotmail.com" is checked. At the bottom of the list is a red button labeled "Confirmar".

Ilustración 4-31: Coordinador - Evaluación de voluntarios

La Ilustración 4-31 lista los voluntarios que participaron en la actividad, permitiendo al coordinador escoger quienes cumplieron con las expectativas de trabajo y quienes no a través de *checkboxes* para cada voluntario. Una vez que el coordinador haya escogido a los voluntarios que cumplieron con su trabajo, debe presionar el botón *Confirmar* para que la aplicación actualice la puntuación de ranking de dichos voluntarios, como se ve en la Ilustración 4-32.



Ilustración 4-32: Voluntario - Ranking actualizado

Como se ve en la Ilustración 4-32, el ranking del voluntario aumentó en un valor respecto a lo que se vio en la Ilustración 4-11. Este cambio también se ilustra gráficamente en el disco ubicado al lado izquierdo del valor ranking.

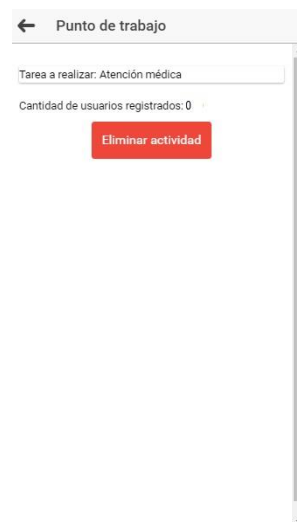


Ilustración 4-33: Coordinador - Actividad terminada

Una vez enviada la orden de actualización de *ranking*, solo queda cerrar y eliminar la actividad. Para esto, se muestra un botón en la vista llamado *Eliminar actividad*, que al ser presionado elimina el punto de trabajo del mapa y del sistema, finalizando complementemente el proceso de trabajo. Esto también se puede hacer desde la vista Inicio, aunque esta opción no asegura que se haya realizado un llamado o que la actividad haya finalizado.

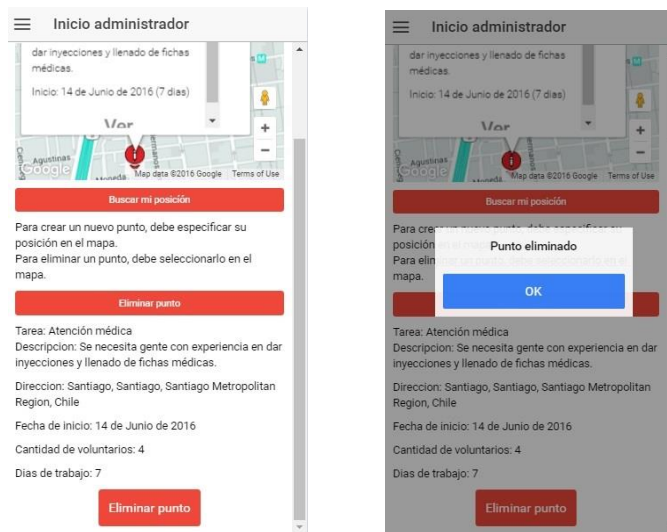


Ilustración 4-34: Coordinador - Eliminando punto

Como se mencionó anteriormente, es posible para el coordinador creador del marcador eliminarlo desde la vista Inicio, presionando en el marcador a eliminar y después presionando en el botón *Eliminar punto*, el cual muestra los datos del punto de trabajo para que el coordinador decida si realmente debe ser eliminado o no. De estar seguro, el coordinador debe presionar el segundo botón *Eliminar punto* ubicado en la parte inferior de la vista, lo que eliminará definitivamente el marcador. Una vez hecho esto, se notifica de la acción al coordinador, y mostrará el mapa sin el punto recién eliminado, como en la Ilustración 4-34.

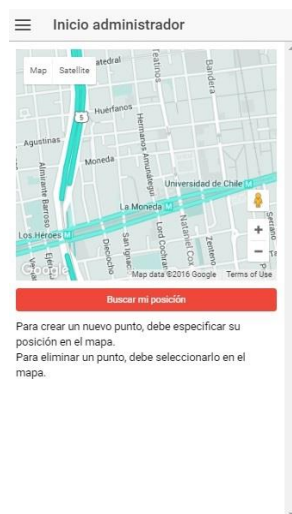


Ilustración 4-35: Coordinador - Punto eliminado

CAPÍTULO 5. EVALUACIÓN Y RESULTADOS

En este capítulo se presentan los resultados obtenidos en la evaluación del *Software* abordando 5 niveles de pruebas: pruebas unitarias, pruebas de integración, pruebas de validación, pruebas de sistema y pruebas de aceptación (Leon Martinez, Aguilar Garcia, Vega Morales, & Gomez Flores, 2013).

5.1 Evaluación

5.1.1 Evaluación pruebas unitarias

A continuación, se muestran las pruebas unitarias realizadas ordenadas por nombre y por el paquete al que pertenecen utilizando 2 letras mayúsculas para distinguir a qué paquete corresponden. En las Tablas 5-1 y 5-2 se muestran las pruebas unitarias para los prototipos creados, y en las Tablas 8-1 hasta la Tabla 8-6, presentes en el Anexo 8.3.1, se muestran las pruebas unitarias relacionadas a cada tabla de la base de datos:

Tabla 5-1: Pruebas unitarias: Prototipo Voluntario

Paquete	Nombre prueba unitaria
A_aplicaciónVoluntario	AA_CargarMapa
A_aplicaciónVoluntario	AB_CentrarPosición
A_aplicaciónVoluntario	AC_CentrarPosiciónInicial
A_aplicaciónVoluntario	AD_CargarMarcador
A_aplicaciónVoluntario	AE_CargarInfowindow
A_aplicaciónVoluntario	AF_ObtenerLatYLong
A_aplicaciónVoluntario	AG_ObtenerToken
A_aplicaciónVoluntario	AH_ValidarRegistroCuenta
A_aplicaciónVoluntario	AI_ValidarInicioSesión
A_aplicaciónVoluntario	AJ_ValidarActualizarGrupo
A_aplicaciónVoluntario	AK_ObtenerFechaPostulación
A_aplicaciónVoluntario	AL_ValidarPostulación
A_aplicaciónVoluntario	AM_RecibirNotificación
A_aplicaciónVoluntario	AN_EncontrarTarea

A_aplicaciónVoluntario	AO_CalcularRutaATarea
A_aplicaciónVoluntario	AP_RankingComolmagen

Tabla 5-2: Pruebas unitarias: Prototipo Coordinador

Paquete	Nombre prueba unitaria
B_aplicaciónCoordinador	BA_CargarMapa
B_aplicaciónCoordinador	BB_CentrarPosición
B_aplicaciónCoordinador	BC_CentrarPosiciónInicial
B_aplicaciónCoordinador	BD_CargarMarcador
B_aplicaciónCoordinador	BE_CargarInfowindow
B_aplicaciónCoordinador	BF_ObtenerLatYLong
B_aplicaciónCoordinador	BG_ValidarRegistroCuenta
B_aplicaciónCoordinador	BH_ValidarInicioSesión
B_aplicaciónCoordinador	BI_ObtenerFechaPunto
B_aplicaciónCoordinador	BJ_ValidarCrearPunto
B_aplicaciónCoordinador	BK_ValidarEliminarPunto
B_aplicaciónCoordinador	BL_IniciarLlamado
B_aplicaciónCoordinador	BM_RecibirEmail
B_aplicaciónCoordinador	BN_ConfirmarParticipaciones
B_aplicaciónCoordinador	BO_FinalizarActividad
B_aplicaciónCoordinador	BP_ValidarCrearGrupo
B_aplicaciónCoordinador	BQ_ValidarEliminarGrupo

Las Tablas 5-3 y 5-4, y desde la Tabla 8-7 hasta la Tabla 8-13 presentes en el Anexo 8.3.1, se muestran los resultados de las pruebas unitarias realizadas. Se realizaron un total de 55 pruebas unitarias, entre las que corresponden a ambos prototipos y las que gestionan la base de datos:

Tabla 5-3: Resumen resultados de pruebas unitarias

Pruebas

Cantidad de pruebas: 55 Fallas: 0 ÉXITO:100%

Paquetes	Cantidad de pruebas	Fallas	Estado
<u>A AplicaciónVoluntario</u>	16	0	✓
<u>B AplicaciónCoordinador</u>	17	0	✓
<u>C ServidorVoluntario</u>	4	0	✓
<u>D ServidorCoordinador</u>	2	0	✓
<u>E ServidorPunto</u>	5	0	✓
<u>F ServidorPostulación</u>	5	0	✓
<u>G ServidorGrupo</u>	3	0	✓
<u>H ServidorMensaje</u>	3	0	✓

Tabla 5-4: Resultado pruebas unitarias generales: prototipo Voluntario

Pruebas A_aplicaciónVoluntario

Cantidad de pruebas: 16 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
AA_CargarMapa	Cargar y visualizar mapa de Google Maps.	Se muestra el mapa en la vista.	✓
AB_CentrarPosición	Utilizar botón para que el mapa muestre la posición global del usuario.	Se muestra posición del usuario en el mapa.	✓
AC_CentrarPosiciónInicial	Al cargar el mapa, mostrar una posición arbitraria.	Se muestra posición predefinida en el mapa.	✓
AD_CargarMarcador	Mostrar los puntos de trabajo gráficamente en el mapa con marcadores.	Se muestra marcador en el mapa.	✓
AE_CargarInfowindow	Desplegar una ventana informativa al presionar un marcador.	Se muestra ventana con información sobre marcador.	✓
AF_ObtenerLatYLong	Obtener valores de latitud y longitud al presionar una posición en el mapa al crear una cuenta de usuario.	Se obtiene latitud y longitud.	✓
AG_ObtenerToken	Obtener identificador del dispositivo para el envío de notificaciones.	Se obtiene identificador <i>token</i> del dispositivo.	✓
AH_ValidarRegistroCuenta	Asegurar que la creación de una nueva cuenta de usuario fue exitosa.	Vista Login se muestra en la aplicación con aviso de éxito.	✓

CAPÍTULO 5. EVALUACIÓN Y RESULTADOS

AI_ValidarInicioSesión	Asegurar que el inicio de sesión del usuario en el sistema fue exitoso.	Vista Inicio se muestra en la aplicación con aviso de éxito.	✓
AJ_ValidarActualizarGrupo	Asegurar que el grupo fue actualizado.	Se muestra aviso de éxito.	✓
AK_ObtenerFechaPostulación	Obtener las fechas especificadas por el voluntario en su postulación.	Se recibe fecha en formato <i>DateTime</i> .	✓
AL_ValidarPostulación	Asegurar que la postulación al punto de trabajo fue recibida.	Se muestra aviso de éxito.	✓
AM_RecibirNotificación	Se envía un aviso al dispositivo del usuario informando sobre el estado de la postulación	Dispositivo recibe notificación.	✓
AN_EncontrarTarea	Utilizar botón para mostrar posición y marcador de la actividad a la que se debe asistir.	Se muestra posición de la actividad en el mapa.	✓
AO_CalcularRutaATarea	Utilizar botón para que el mapa muestre ruta entre posición del usuario y el punto de trabajo.	Se muestra propuesta de ruta a tomar para llegar al punto.	✓
AP_RankingComolmagen	Representar el valor de <i>ranking</i> gráficamente.	Se muestra un gráfico de disco en vista Perfil	✓

De manera independiente, las pruebas unitarias funcionan con un 100% de éxito, que se ha mantenido durante todo el transcurso del desarrollo del sistema al irse probando constantemente.

5.1.2 Evaluación pruebas de integración

A continuación, se presentan las pruebas de integración, cuyo objetivo es identificar errores que puedan surgir por la combinación de los métodos probados en el capítulo 5.2. Para esto se utilizó la técnica *bottom-up*, en donde se comienza el análisis con los módulos de nivel inferior y se verifica que estos módulos llamen correctamente a los módulos de nivel superior, como se muestra en la Ilustración 5-1, donde se toman en el mismo conjunto los módulos relacionados con operaciones de Voluntario y Coordinador, debido a la manera en que se comunican los procesos entre sí para conformar la tarea principal del sistema. Los planes de prueba escogidos son los más significativos y donde se podrían encontrar errores de interacción entre módulos.

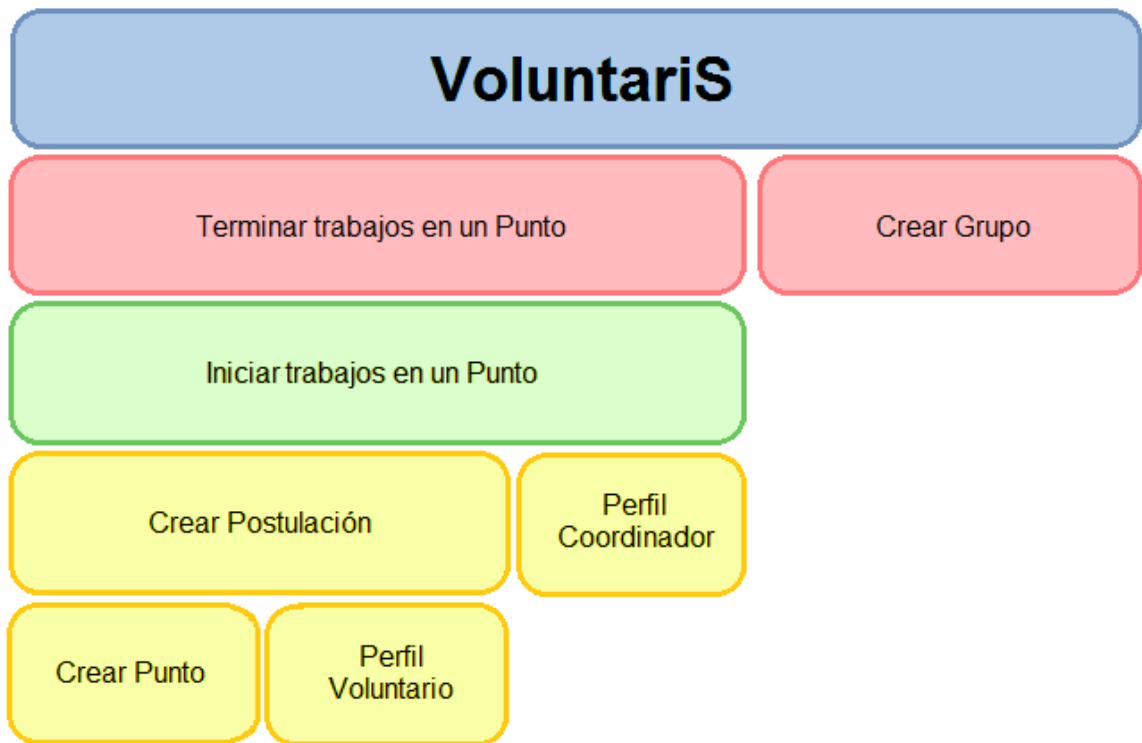


Ilustración 5-1: Diagrama Down-Top, módulos para las pruebas de integración

Las Tablas 5-5 y 5-6, y desde la Tabla 8-14 hasta la Tabla 8-18 presentes en el Anexo 8.3.2, se detallan los planes de prueba, en donde se evaluaron las pruebas unitarias trabajando en distintos módulos.

Tabla 5-5: Plan de prueba: Perfil Voluntario

Ítem	Descripción
Identificación	Plan de prueba 01 – Perfil Voluntario
Elementos a probar	Crear tabla Voluntario, cargar mapa, centrar posición, cargar marcador, cargar <i>infowindow</i> , obtener latitud y longitud, obtener <i>token</i> , validar registro, interfaz del perfil del Voluntario.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Queda la información del Voluntario guardada en el sistema.
Tareas para satisfacer el proceso	Se ingresa la información del voluntario. Validar que la información queda correctamente guardada.

Ítem	Descripción
Descripción	Se inicia el registro de la información del perfil, ingresando el correo electrónico, la contraseña, el primer nombre, el segundo nombre (opcional), el primer apellido, el segundo apellido, la cédula de identidad, una posición en el mapa, y el grupo de trabajo. Finalmente, se guarda la información en el sistema.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.
Riesgos	Sin riesgos.

Tabla 5-6: Plan de prueba: Perfil Coordinador

Ítem	Descripción
Identificación	Plan de prueba 02 – Perfil Coordinador
Elementos a probar	Crear tabla Coordinador, validar registro, interfaz del perfil del Coordinador.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Queda la información del Coordinador guardada en el sistema.
Tareas para satisfacer el proceso	Se ingresa la información del coordinador. Validar que la información queda correctamente guardada.
Descripción	Se inicia el registro de la información del perfil, ingresando el correo electrónico, la contraseña, y el nombre completo. Finalmente, se guarda la información en el sistema.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.
Riesgos	Sin riesgos.

5.1.3 Evaluación pruebas de validación

A continuación, se presentan las pruebas de validación, cuyo objetivo es verificar si el *Software* cumple con las especificaciones definidas. La técnica empleada es la de caja negra, donde se define una entrada y se compara la salida obtenida con la salida esperada. La plantilla utilizada para definir los casos de prueba consideran un identificador del caso de prueba, los casos de uso sobre los cuales se construye la prueba, los módulos o funcionalidades sobre los que se realiza la prueba, una descripción del caso de prueba, el estado previo de la aplicación antes de comenzar con el caso de prueba, la secuencia de pasos a realizar, si este se encuentra automatizado o no, el resultado obtenido una vez aplicado los pasos, el responsable

de aplicar el caso de prueba y finalmente la fecha de creación del caso de prueba. Los casos de pruebas se detallan en la tabla 5-7, y desde la Tabla 8-19 hasta la Tabla 8-24 presentes en el Anexo 8.3.3.

Tabla 5-7: Caso de prueba para la creación de un punto de trabajo

Elemento plantilla	Descripción
Identificador	Caso de prueba-01
Casos de uso asociados	CU-08
Módulo/funcionalidad	Crear punto.
Descripción	El caso de prueba consiste en la creación de un punto de trabajo, con detalles y características, y su visualización en el mapa de la aplicación.
Pre-condiciones	Se debe estar en la vista principal de la aplicación Coordinador. Deben existir categorías de actividades.
Secuencia de pasos	1.- Se presiona en el mapa. 2.- Se ingresa la información solicitada por el sistema para la creación del punto. 3.- Se confirma la creación del punto. 4.- Se verifica en el mapa que el punto fue creado.
Resultado esperado	Se espera que la aplicación notifique visualmente la creación exitosa del nuevo punto de trabajo. Se espera que la información del punto quede registrada en el sistema. Se espera que la información visualizada en el mapa sea la misma que la información ingresada.
Automatizado	No.
Resultado efectivo	Aceptado.
Responsable	Desarrollador
Fecha de creación	02/07/2016

5.1.4 Evaluación pruebas de sistema

Para esta evaluación se consideraron 3 tipos de pruebas: Pruebas de funcionalidad, pruebas de contenido y pruebas de usabilidad.

1. Pruebas de funcionalidad: Las pruebas de funcionalidad fueron realizadas y validadas en el capítulo 5.2, asegurando que el sistema cubre las necesidades de funcionamiento acordadas en las especificaciones de diseño. El sistema lleva a cabo correctamente las funciones requeridas.

Las pruebas de funcionalidades del capítulo 5.2 fueron realizadas en el escenario de múltiples voluntarios postulando a una actividad gestionada en un nuevo punto de trabajo

creado por el coordinador. Se evaluaron los casos de prueba con una base de datos previamente poblada, de manera que se puede probar que se pueda trabajar con la información previamente ingresa y que no exista problema al ingresar nueva información.

2. Pruebas de contenido: En esta prueba se verificó que el contenido presentado por la aplicación sea coherente con lo que espera ver el usuario. Para esto no se hace uso de palabras cercanas al lenguaje del desarrollador y del sistema, con lo que se busca que todo el contenido presentado en la aplicación pueda transmitir una idea adecuada al usuario.

3. Prueba de usabilidad: Se realizaron evaluaciones de usabilidad del prototipo Voluntario final con dos personas durante la realización de las pruebas de aceptación realizadas en el punto 5.1.5.

5.1.5 Evaluación pruebas de aceptación

La evaluación fue realizada por una persona, en compañía del desarrollador (prueba Beta). A continuación, desde la Tabla 5.8 hasta la Tabla 5.10 se presentan las pruebas realizadas por el cliente y las observaciones realizadas en cada prueba.

Tabla 5-8: Prueba Beta 01

Elemento plantilla	Descripción
Identificador	Prueba Beta 01
Fecha	04/07/2016
Prueba realizada	La prueba realizada consistió en la creación de una nueva cuenta de usuario Voluntario.
Observaciones	Al momento de especificar la posición que debe escoger el usuario para ser llamado, se debe destacar mejor el botón de geo-localización y la funcionalidad del mapa. Por lo demás, resulta intuitivo.

Tabla 5-9: Prueba Beta 02

Elemento plantilla	Descripción
Identificador	Prueba Beta 02
Fecha	04/07/2016
Prueba realizada	La prueba realizada consistió en la postulación del Voluntario a una actividad.
Observaciones	Se debe dar más información sobre la función de los cuadros para ingresar las fechas de inicio y fin.

Tabla 5-10: Prueba Beta 03

Elemento plantilla	Descripción
Identificador	Prueba Beta 03
Fecha	04/07/2016
Prueba realizada	La prueba realizada consistió en revisar el resultado de la postulación
Observaciones	Se destaca la ayuda de la aplicación al cargar automáticamente el punto de destino. Búsqueda de ruta es intuitiva.

5.2 Resultados

En esta sección se presentan los resultados obtenidos de una prueba realizada con un dispositivo con la versión de Administrador instalada para crear un punto de trabajo, y 5 usuarios, cada uno con su propio dispositivo y cuentas de usuario, postulando al punto de trabajo, ingresando fechas de inicio y fin distintas. Se mostrarán solo los datos que competen a la prueba realizada. El punto de trabajo utilizado como prueba comprende la siguiente información, mostrada en la Tabla 5-11:

Tabla 5-11: Datos de punto generado

Elemento PUNTO	Información
Id	575d4a3ecd39251100a138ed
Nombre	Atención médica
Ubicación	[-33.44286575790805, -70.65741062164307]
Dirección	Santiago, Santiago, Santiago Metropolitan Region, Chile
Descripción	Se necesita gente con experiencia en dar inyecciones y llenado de fichas médicas.
Coordinador	admin@admin.cl
Cantidad de voluntarios por día	3
Tipo de actividad	Salud
Fecha de inicio	2016-06-14T04:00:00.000Z
Cantidad de días	7

Cada usuario participante comprende su propio conjunto de datos, mostrados en la Tabla 5-12:

Tabla 5-12: Datos de usuarios registrados

Elemento VOLUNTARIO	Información	Elemento VOLUNTARIO	Información
Correo electrónico	Voluntario1 @ejemplo.com	Correo electrónico	Voluntario2 @ejemplo.com
Ubicación	[-33.40452642154241, -70.68359101452412]	Ubicación	[-33.51146894349046, -70.76987779841166]
Ranking	2	Ranking	5

Elemento VOLUNTARIO	Información	Elemento VOLUNTARIO	Información
Correo electrónico	Voluntario3 @ejemplo.com	Correo electrónico	Voluntario4 @ejemplo.com
Ubicación	[-29.91039397845104, -71.24989306948752]	Ubicación	[-34.17221697048133, -70.73504497824164]
Ranking	2	Ranking	2

Elemento VOLUNTARIO	Información
Correo electrónico	Voluntario5 @ejemplo.com
Ubicación	[-33.44155651941707, -70.64640011635224]
Ranking	0

Cuando los usuarios realizan su postulación al punto de trabajo especificado, se generan los siguientes datos, representados en la Tabla 5-13, ordenados por ranking y distancia a la ubicación del punto de trabajo como se trabajó en el algoritmo:

Tabla 5-13: Datos de postulaciones creadas por voluntarios

Elemento POSTULACIÓN	Información
Nombre	Atención Médica
Id punto	575d4a3ecd39251100a138ed
Voluntario	Voluntario2@ejemplo.com
Ubicación actividad	[-33.44286575790805, -70.65741062164307]
Ubicación voluntario	[-33.51146894349046, -70.76987779841166]
Fecha de inicio	2016-06-14T04:00:00.000Z
Fecha de finalización	2016-06-19T04:00:00.000Z
Ranking	5

Elemento POSTULACIÓN	Información
Nombre	Atención Médica
Id punto	575d4a3ecd39251100a138ed
Voluntario	Voluntario1@ejemplo.com
Ubicación actividad	[-33.44286575790805, -70.65741062164307]
Ubicación voluntario	[-33.40452642154241, -70.68359101452412]
Fecha de inicio	2016-06-14T04:00:00.000Z
Fecha de finalización	2016-06-16T04:00:00.000Z
Ranking	2

Elemento POSTULACIÓN	Información
Nombre	Atención Médica
Id punto	575d4a3ecd39251100a138ed
Voluntario	Voluntario4@ejemplo.com
Ubicación actividad	[-33.44286575790805, -70.65741062164307]
Ubicación voluntario	[-34.17221697048133, -70.73504497824164]
Fecha de inicio	2016-06-15T04:00:00.000Z
Fecha de finalización	2016-06-20T04:00:00.000Z
Ranking	2

Elemento POSTULACIÓN	Información
Nombre	Atención Médica
Id punto	575d4a3ecd39251100a138ed
Voluntario	Voluntario3@ejemplo.com
Ubicación actividad	[-33.44286575790805, -70.65741062164307]
Ubicación voluntario	[-29.91039397845104, -71.24989306948752]
Fecha de inicio	2016-06-14T04:00:00.000Z
Fecha de finalización	2016-06-20T04:00:00.000Z
Ranking	2

Elemento POSTULACIÓN	Información
Nombre	Atención Médica
Id punto	575d4a3ecd39251100a138ed
Voluntario	Voluntario5@ejemplo.com
Ubicación actividad	[-33.44286575790805, -70.65741062164307]
Ubicación voluntario	[-33.44155651941707, -70.64640011635224]
Fecha de inicio	2016-06-15T04:00:00.000Z
Fecha de finalización	2016-06-19T04:00:00.000Z
Ranking	0

A continuación, en la Tabla 5-14, se muestran los mensajes generados al finalizar el algoritmo. Se puede notar que sólo 4 voluntarios fueron llamados, excluyendo a Voluntario5. Además, el día para el que Voluntario4 fue llamado fue aplazado para el 16 de Junio respecto a la postulación que especificaba su asistencia propuesta para el 15 de Junio:

Tabla 5-14: Datos de mensajes resultantes de postulaciones

Elemento MENSAJE	Información	Elemento MENSAJE	Información
Voluntario	Voluntario1@ejemplo.com	Voluntario	Voluntario2@ejemplo.com
Mensaje	Se le requiere el día 14 de Junio de 2016 en el punto escogido por usted para iniciar los trabajos.	Mensaje	Se le requiere el día 14 de Junio de 2016 en el punto escogido por usted para iniciar los trabajos.
Fecha	2016-07-05T04:00:00.000Z	Fecha	2016-07-05T04:00:00.000Z
Ubicación actividad	[-33.44286575790805, -70.65741062164307]	Ubicación actividad	[-33.44286575790805, -70.65741062164307]

Elemento MENSAJE	Información	Elemento MENSAJE	Información
Voluntario	Voluntario3@ejemplo.com	Voluntario	Voluntario4@ejemplo.com
Mensaje	Se le requiere el día 15 de Junio de 2016 en el punto escogido por usted para iniciar los trabajos.	Mensaje	Se le requiere el día 16 de Junio de 2016 en el punto escogido por usted para iniciar los trabajos.
Fecha	2016-07-05T04:00:00.000Z	Fecha	2016-07-05T04:00:00.000Z
Ubicación actividad	[-33.44286575790805, -70.65741062164307]	Ubicación actividad	[-33.44286575790805, -70.65741062164307]

CAPÍTULO 5. EVALUACIÓN Y RESULTADOS

El itinerario resultante se representa en la Tabla 5-15, en un formato similar al utilizado en el capítulo 4, resaltando en color amarillo los cupos asignados, y con color verde los días en que hay recambio de voluntario:

Tabla 5-15: Itinerario solución

	14 de junio		15 de junio		16 de junio		17 de junio		18 de junio		19 de junio		20 de junio	
1	V2		V2		V2		V2		V2			V2		
2	V1		V1		V4	V1	V4		V4		V4			V4
3			V3		V3		V3		V3		V3			V3

Puede verse que, en las pruebas realizadas a pequeña escala, el algoritmo cumple con distribuir un grupo de voluntarios a través del prototipo creado, creando un itinerario que cubre la mayoría de los días de trabajo especificados, de manera que es posible contar con una cantidad razonable de voluntarios planificados para trabajar en la tarea a la que postularon.

Continuando con la metodología, se presentan en las Tablas 8-25 y 8-26, presentes en el Anexo 8.4, las matrices de prueba en cada prototipo para cada requerimiento especificado en el sub-capítulo 3.5.1, que validan si los requerimientos se cumplieron o no.

CAPÍTULO 6. DISCUSIÓN Y CONCLUSIONES

Después del desarrollo del proyecto de titulación, que incluyó el estudio de la situación del voluntariado en Chile frente a las catástrofes, se puede concluir lo siguiente:

Primero, se debe concluir que se cumplió el objetivo general del trabajo de titulación, puesto que se logró desarrollar una aplicación prototipo que permita el manejo de múltiples usuarios, y que los ordene para obtener un plan de trabajo óptimo para cada actividad, sirviendo como apoyo para la gestión de postulantes a voluntariado en situaciones de catástrofe. Respecto a los objetivos específicos planteados, se puede decir que se cumplieron, ya que se acordaron las variables necesarias para formar un algoritmo capaz de lograr la meta principal: Coordinar y distribuir un conjunto de postulaciones de voluntarios para trabajar en un punto específico en el mapa. Se fijaron los requerimientos y se analizaron para desarrollar prototipos funcionales del sistema tanto para los usuarios como para los coordinadores. Las pruebas del sistema fueron limitadas, se hicieron a nivel individuo, no con una gran cantidad de personas, debido a su condición de ser una aplicación móvil aún en desarrollo. Aun así, el prototipo funciona correctamente al ser instalado en un dispositivo móvil, y puede conectarse con el servidor del sistema e intercambiar datos, y el algoritmo de asignación de voluntarios funciona y convoca correctamente a múltiples usuarios postulantes, por lo que el sistema cumple a pequeña y mediana escala.

Se estimó que el modelo de datos utilizado fuera simple, ya que el objetivo principal contempla el uso de una aplicación móvil que involucra a usuarios que pueden ser inexpertos en el uso de dicha tecnología, por lo que lo ideal es que la aplicación utilice la menor cantidad de variables posible para que la carga de trabajo del usuario al usar la aplicación no lo abrume. Además, se buscó que las consultas a la base de datos fueran sencillas, llevando a un sistema de baja complejidad y adaptable para cambios y futuras versiones.

Sobre el capítulo de Implementación y los ejemplos mostrados del actuar del algoritmo, cabe decir que estudiar un par de casos particulares hace que esta propuesta necesite la validación de otros casos para poder ser aplicable en situaciones reales. De todos modos, durante la construcción y las pruebas del sistema se plantearon múltiples situaciones que llevaron a cubrir muchos casos específicos, como el caso de que un voluntario desee trabajar sólo un día, o que no haya un plazo que alcance para la postulación de un voluntario

con un *ranking* alto, por lo que el planteamiento teórico y el prototipo desarrollado muestran una idea de cómo debería funcionar un sistema completo cuando se lleve a producción.

Por último, cabe mencionar las mejoras al sistema, para que ésta se pueda implementar y el proyecto pase a una etapa de producción:

- Potenciar el desempeño del software: Como el proyecto fue desarrollado para ser un prototipo, carece de validaciones de desempeño como las que una aplicación móvil final debería tener, ya que el desarrollo se enfocó en las secciones funcionales del sistema, dejando de lado rasgos indispensables para la solución esperada.
- Implementar un sistema *Ranking* para cada tipo de tarea requerida en los voluntarios. El proceso de desarrollo buscó principalmente implementar un algoritmo funcional para la distribución de los voluntarios postulantes en una actividad, y se obviaron aspectos más específicos, resultando en un proceso de selección funcional pero abierto a mejoras.
- Perfeccionar el algoritmo para aumentar la cantidad de atributos utilizados para ordenar las postulaciones para ser analizadas. Actualmente, el sistema ordena las postulaciones por valor de *ranking*, y por la distancia entre voluntario postulante y punto de trabajo. Otros factores que pudiesen ser tomados en cuenta para escoger voluntarios son, por ejemplo, la cantidad de días propuestos, para privilegiar a los que trabajen por más días, o las fechas de inicio o fin, para asegurar que para los primeros o últimos días de trabajo todos los cupos diarios estén ocupados, entre otras características.
- Permitir almacenar información enviada desde el servidor como información local en el dispositivo. Ya se realiza esto con los datos de la sesión iniciada en la aplicación, pero poder guardar localmente información como la ruta a tomar hasta el punto de trabajo podría tener mucha utilidad en locaciones sin conexión a Internet.

El estudio realizado sobre herramientas existentes que involucren la asistencia y apoyo de autoridades y población en casos de desastre permitió dar cuenta de la magnitud que alcanza la necesidad de saber responder a tales emergencias, y esto responde a la conciencia existente en todo el mundo sobre la urgencia que algunas de estas emergencias pueden alcanzar. De todos modos, es merecedor hacer una comparación del prototipo creado con las herramientas ya existentes:

- **Google Person Finder** tiene como meta facilitar la reubicación de personas desaparecidas, por lo que comparten con el prototipo desarrollado la ventaja de que la gente aportará también con esfuerzos para el funcionamiento de ambos sistemas.
- **VoluntariS** y **MicroMappers** comparten el uso de mapas para mostrar información, aunque **MicroMappers** posee un enfoque mucho más ilustrativo respecto a la meta del prototipo **VoluntariS**.
- Al hablar de **Ushahidi** se mencionó su capacidad de mostrar en tiempo real lo que esté ocurriendo en una localidad en casos de emergencia, funcionando como una fuente de noticias más rápido y amplio que **VoluntariS**, aunque sin implementar la coordinación de voluntariado. **VoluntariS** combina el uso de mapas para especificar puntos destacados en una situación de emergencia, el manejo y coordinación de entidades presente en **Eden** y en **VolunteerLocal**, y la capacidad de mostrar puntos de trabajo, como una especie de catálogo para que la población decida donde participar, que posee **VolunteerMatch**.
- El funcionamiento del prototipo creado depende fundamentalmente del uso de **Google Maps**, gracias a las herramientas que provee para desarrolladores.

El estudio realizado sobre herramientas de asistencia dio a conocer que no existe una herramienta que tenga como objetivo específico la coordinación de voluntarios. Así mismo, este estudio crea también una mejor apreciación de las metas a lograr en el desarrollo de **VoluntariS**.

CAPÍTULO 7. BIBLIOGRAFÍA

Amazon. (2016). *Amazon Web Services*. Recuperado el 24 de Junio de 2016, de <https://aws.amazon.com/es/nosql/>

América Solidaria. (s.f.). *Americasolidaria.org*. Recuperado el 08 de Mayo de 2015, de <http://www.americasolidaria.org/informate/el-instituto-nacional-de-la-juventud-asumio-el-desafio-de-apoyar-en-la-coordinacion-de-los-voluntariosas-del-pais/>

AMSCA. (2015). *INJUV*. Recuperado el 15 de Julio de 2015, de <http://www.unvoluntariosuma.cl/organizacion/agrupacion-medica-social-chile-ayuda-amsca/>

Biblioteca Nacional de Chile. (2014). *Memoria Chilena*. Recuperado el 31 de Mayo de 2015, de <http://www.memoriachilena.cl/602/w3-article-3576.html#cronologia>

Boehm, B. W. (1988). *A Spiral Model of Software Development and Enhancement*, *IEEE Computer*.

Burle, S. (2012). *FloodMap*. Recuperado el 20 de Junio de 2016

Chile Voluntario. (2015). *Red de Organizaciones de Voluntariado de Chile*. Recuperado el 21 de Junio de 2016, de <http://www.chilevoluntario.cl/>

CreArte. (2016). *CreArte*. Recuperado el 21 de Junio de 2016, de <http://www.createarte.cl/>

Diario Digital ElAconcagua. (15 de Marzo de 2015). *elaconcagua.cl*. Recuperado el 8 de Mayo de 2015, de <http://elaconcagua.cl/2011/comandante-de-bomberos-de-san-felipe-acusa-enorme-descoordinacion-en-combate-de-incendio-en-valparaiso/>

EM-DAT. (2009). *Emergency Events Database*. Recuperado el 21 de Junio de 2016, de <http://www.emdat.be/classification>

Faltan Manos. (2015). *Faltan Manos*. Recuperado el 7 de Mayo de 2016, de <http://faltanmanos.org/la-historia/>

Fundación Trascender & GFK Collect. (2013). *Estudio Nacional de Voluntariado 2013*.

CAPÍTULO 7. BIBLIOGRAFÍA

Gavidia, S. (s.f.). *Recursos Educativos en Español para la Medicina de Emergencia*. Recuperado el 8 de Mayo de 2015, de <http://www.reeme.arizona.edu/materials/Ciclos%20de%20Desastres.pdf>

Google. (1 de Junio de 2016). *Google*. Recuperado el 3 de Mayo de 2016, de <https://developers.google.com/maps/articles/geolocation>

Google. (1 de Junio de 2016). *Google*. Recuperado el 3 de Mayo de 2016, de <https://developers.google.com/maps/?hl=es>

Google. (2010). *Google Person Finder*. Recuperado el 8 de Mayo de 2015, de <https://google.org/personfinder/global/home.html>

Gracia, L. M. (30 de Enero de 2013). *Un poco de Java*. Recuperado el 26 de Junio de 2016, de <https://unpocodejava.wordpress.com/2013/01/30/un-poco-de-mongodb-que-es-que-ofrece/>

Guías y scouts de Chile. (2016). *Guías y scouts de Chile*. Recuperado el 21 de Junio de 2016, de <http://www.guiasy scoutsdechile.cl/>

Imilan, W. A. (26 de Marzo de 2015). Opinión: ¿Aprendemos algo de las catástrofes?

INE. (Febrero de 2015). Recuperado el 7 de Mayo de 2016, de http://www.ine.cl/canales/chile_estadistico/estadisticas_medio_ambiente/2014/informe-medio-ambiente2014.pdf

International Federation of Red Cross and Red Crescent Societies. (2010). *Saving Lives, Changing Minds*.

iRevolutions. (12 de Diciembre de 2014). *iRevolutions*. Recuperado el 19 de Junio de 2016, de <https://irevolutions.org/2014/12/12/micromappers-humanitarian-technology/>

La Tercera. (22 de Febrero de 2016). La nueva Onemi: promesa incumplida del 27/F.

LatinLab, GSAPP, Santiago Research Cell. (25 de Febrero de 2015). *LEARNING FROM 27F: A Comparative Assessment Of Urban Reconstruction Processes After The 2010 Earthquake In Chile*. Recuperado el Mayo de 2016, de http://globalcenters.columbia.edu/santiago/santiago/files/globalcenters_chile/L27F_Final%20Report%20-%20ESP%20-26-2-2015.pdf

Leon Martinez, N. E., Aguilar Garcia, J. L., Vega Morales, E. F., & Gomez Flores, L. C. (2013 йил 4-Diciembre). *Computer tool for software testing documentation under research activities*. Retrieved 2015 йил 24-Noviembre

Moreno, A. (16 de Noviembre de 2012). Ushahidi, una aplicación para el cambio social, se presenta en Madrid. *Applicantes*.

National & Community Service. (2006). *Volunteer Centers of Michigan*. Recuperado el 5 de Mayo de 2015, de http://www.mivolunteers.org/CMDocs/VCM/Resources/Disaster%20Response/Managing_Spontaneous_Volunteers_in_Times_of_Disaster_2009_UPS_HON%20%281%29.pdf

Nielsen, J. (1 de Enero de 1995). *Nielsen Normal Group*. Recuperado el 24 de Noviembre de 2015, de <https://www.nngroup.com/articles/ten-usability-heuristics/>

ONEMI. (Marzo de 1974). *Oficina Nacional de Emergencia del Ministerio del Interior y Seguridad Pública*. Recuperado el 21 de Junio de 2016, de <http://www.onemi.cl/>

ONEMI. (04 de Mayo de 2016). *ONEMI y la ONU inician proyecto para fortalecer la respuesta ante emergencias*. Recuperado el 09 de Junio de 2016, de <http://www.onemi.cl/noticia/onemi-y-la-onu-inician-proyecto-para-fortalecer-la-respuesta-ante-emergencias/>

ONEMI. (29 de Abril de 2013). *Sistema Nacional de Protección Civil*. Recuperado el 4 de Mayo de 2016, de http://www.sicurlatinoamerica.cl/varios/SICUR_presentaciones/jornada4/Sistema_Nacional_Proteccion_Civil.pdf

Páez, D., Arroyo, E., & Fernandez, I. (1995). *Catástrofes, Situaciones de Riesgo y Factores Psicosociales*.

Red Cross. (s.f.). *redcross.eu*. Recuperado el 7 de Noviembre de 2015, de http://www.redcross.eu/en/upload/documents/pdf/437_dm_tools.pdf

Sahana Software Foundation. (Mayo de 2004). *Plataforma de gestión humanitaria Sahana Eden*. Recuperado el 22 de Junio de 2016, de <http://demo.eden.sahanafoundation.org/>

Sahana Software Foundation. (Mayo de 2004). *Sahana Foundation - Open Source Disaster Management Software*. Recuperado el Junio de 2016, de <http://eden.sahanafoundation.org/>

SHOA. (1990). *Servicio Hidrográfico y Oceanográfico de la Armada*. Recuperado el 21 de Junio de 2016, de <http://www.shoa.cl/>

Sommerville, I. (2005). *Ingeniería de Software*. Pearson Addison-Wesley.

TechTarget. (Marzo de 2014). *SearchData Management*. Recuperado el 26 de Junio de 2016, de <http://searchdatamanagement.techtarget.com/definition/MongoDB>

Techterms. (20 de Junio de 2016). *Techterms*. Recuperado el 4 de Mayo de 2016, de <http://techterms.com/definition/api>

Universidad de Chile. (s.f.). *Red de Académicos FAU*. Recuperado el Junio de 2016, de <http://academicos.uchilefau.cl/imilan-walter-alejandro/>

University of Maryland. (29 de Marzo de 2013). Recuperado el 7 de Mayo de 2016, de <http://www.rhsmith.umd.edu/news/disaster-management-software-meets-sandy-challenge-ready-next-crisis>

Ushaguzi. (2013). *Ushahidi.com*. Recuperado el 7 de Noviembre de 2015, de <http://www.ushahidi.com/>

VolunteerLocal. (2009). *VolunteerLocal*. Recuperado el 21 de Junio de 2016, de https://app.volunteerlocal.com/start/event_information.php

VolunteerLocal. (2015). *VolunteerLocal*. Recuperado el 7 de Noviembre de 2015, de <https://www.volunteerlocal.com/>

VolunteerMatch. (2015). *VolunteerMatch*. Recuperado el 7 de Noviembre de 2015, de <http://www.volunteermatch.org/>

Wellerstein, A. (2014). *NUKEMAP*. Recuperado el 20 de Junio de 2016, de <http://nuclearsecrecy.com/nukemap/>

Wilson, J. (2000). *Volunteering*. Durham, North Carolina: Annual Review of Sociology.

Zulueta Azócar, S. (2003). *La Evolución del Voluntariado en Chile entre los años 1990 y 2002*. Santiago, Chile: Universidad Católica de Chile.

CAPÍTULO 8. ANEXOS

8.1 Diagramas de Secuencia

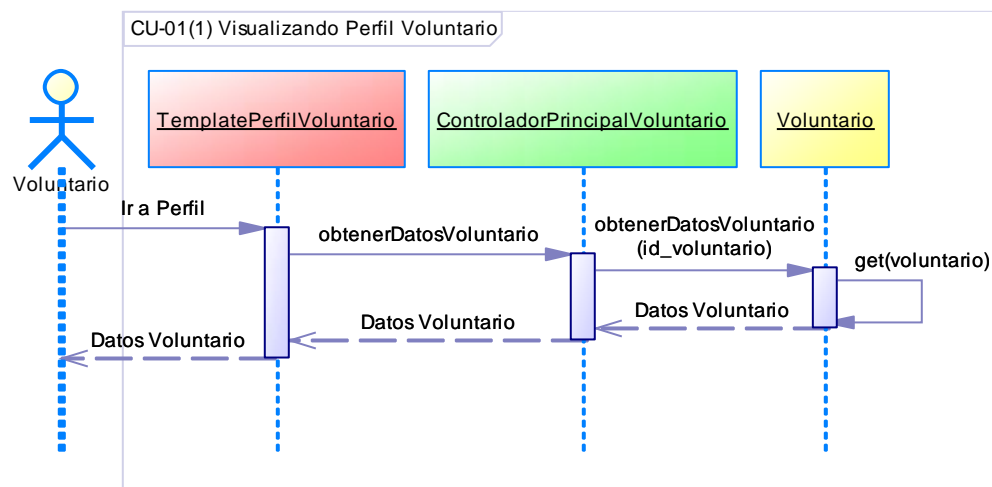


Ilustración 8-1: Diagrama de secuencia Visualizando Perfil Voluntario

- obtenerDatosVoluntario(): El sistema realiza la petición a la base de datos para mostrar la información de perfil del usuario.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos del usuario Voluntario.

Operación: Visualizando Perfil Voluntario (Ilustración 8-1).

Actor: Voluntario.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: Sin excepciones.

Post-condiciones: La información del voluntario queda visible en la aplicación.

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Perfil para ingresar template de Perfil. En el template de Perfil se muestra la información obtenida de la base de datos sobre el usuario (nombre1_voluntario, nombre2_voluntario, apellido1_voluntario, y apellido 2_voluntario, agrupados para formar el nombre completo del usuario; email_voluntario,

que muestra el correo electrónico del usuario; dirección_voluntario donde el usuario situó el punto de referencia en el mapa, y el valor de ranking_voluntario que muestra el valor *ranking* del usuario, con una imagen que representa al valor *ranking* gráficamente).

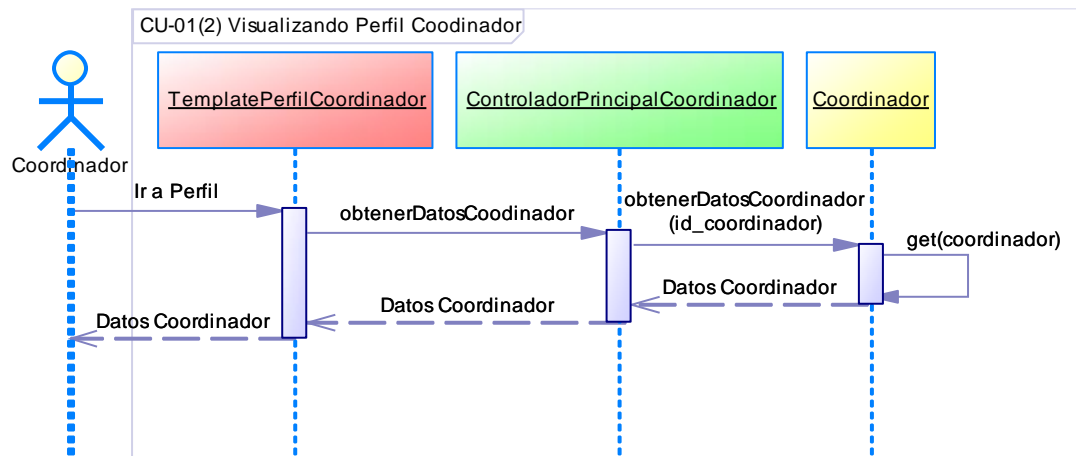


Ilustración 8-2: Diagrama de secuencia Visualizando Perfil Coordinador

- obtenerDatosCoordinador(): El sistema realiza la petición a la base de datos para mostrar la información de perfil del usuario.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos del usuario Coordinador.

Operación: Visualizando Perfil Coordinador (Ilustración 8-2).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: Sin excepciones.

Post-condiciones: La información del coordinador queda visible en la aplicación.

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Perfil para ingresar template de Perfil. En el template de Perfil se muestra la información obtenida de la base de datos sobre el usuario (nombrecompleto_coordinador, que muestra el nombre del coordinador ingresado al registrarse; y email_coordinador, que muestra el correo electrónico del coordinador).

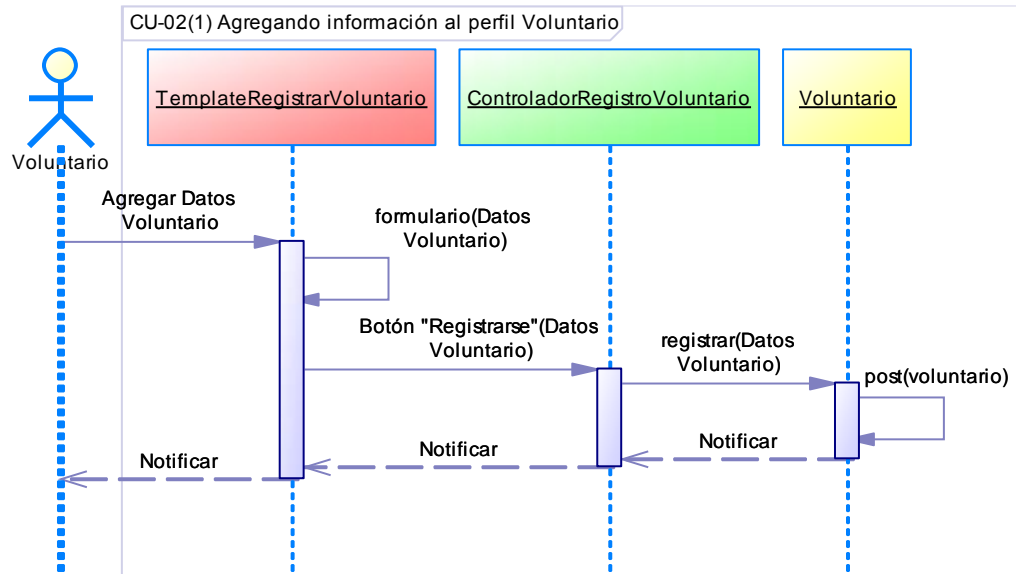


Ilustración 8-3: Diagrama de secuencia Agregando información a perfil Voluntario

- Botón “Registrarse”(): Invoca la función registrar().
- registrar(): Envía la orden de recibir los datos ingresados por el usuario en el formulario del template TemplateRegistrarVoluntario, y los envía al sistema para ser almacenados en la base de datos.
- post(): Método invocado por **Express** para almacenar información a la base de datos. En este caso, almacena los datos ingresados por el usuario en la tabla Voluntario.

Operación: Agregando información al perfil Voluntario (Ilustración 8-3).

Actor: Voluntario.

Pre-condiciones: Sin pre-condiciones.

Excepciones: Sin excepciones.

Post-condiciones: El usuario queda con su información registrada en el sistema.

Descripción: El usuario inicia el registro de su información en la aplicación. En el template TemplateRegistrarVoluntario se despliega un formulario en donde se le solicita al usuario los datos para su perfil (email_voluntario, contraseña_voluntario, nombre1_voluntario, nombre2_voluntario, apellido1_voluntario, apellido2_voluntario, rut_voluntario, ubicación_voluntario, que es el punto de referencia para las postulaciones, y grupo_voluntario,

que es opcional). Cuando termina de ingresar estos datos, el usuario presiona el botón “Registrarse”, llamando a la función registrar() que hace que la base de datos reciba los datos del formulario y los almacene en la tabla Voluntario, creando un nuevo usuario voluntario en el sistema.

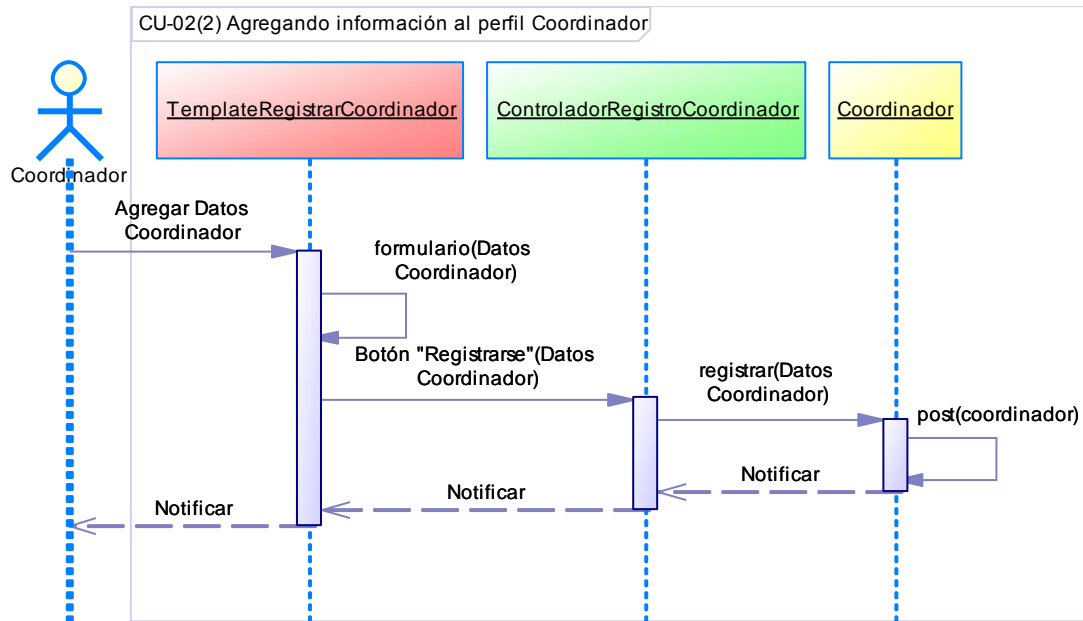


Ilustración 8-4: Diagrama de secuencia Agregando información a perfil Coordinador

- Botón “Registrarse”(): Invoca la función registrar().
- registrar(): Envía la orden de recibir los datos ingresados por el usuario en el formulario del template TemplateRegistrarCoordinador, y los envía al sistema para ser almacenados en la base de datos.
- post(): Método invocado por **Express** para almacenar información a la base de datos. En este caso, almacena los datos ingresados por el usuario en la tabla Coordinador.

Operación: Agregando información al perfil Coordinador (Ilustración 8-4).

Actor: Coordinador.

Pre-condiciones: Sin pre-condiciones.

Excepciones: Sin excepciones.

Post-condiciones: El usuario queda con su información registrada en el sistema.

Descripción: El usuario inicia el registro de su información en la aplicación. En el template TemplateRegistrarCoordinador se despliega un formulario en donde se le solicita al usuario los datos para su perfil (email_coordinador, contraseña_voluntario, y nombrecompleto_voluntario). Cuando termina de ingresar estos datos, el usuario presiona el botón “Registrarse”, llamando a la función registrar() que hace que la base de datos reciba los datos del formulario y los almacene en la tabla Coordinador, creando un nuevo usuario coordinador en el sistema.

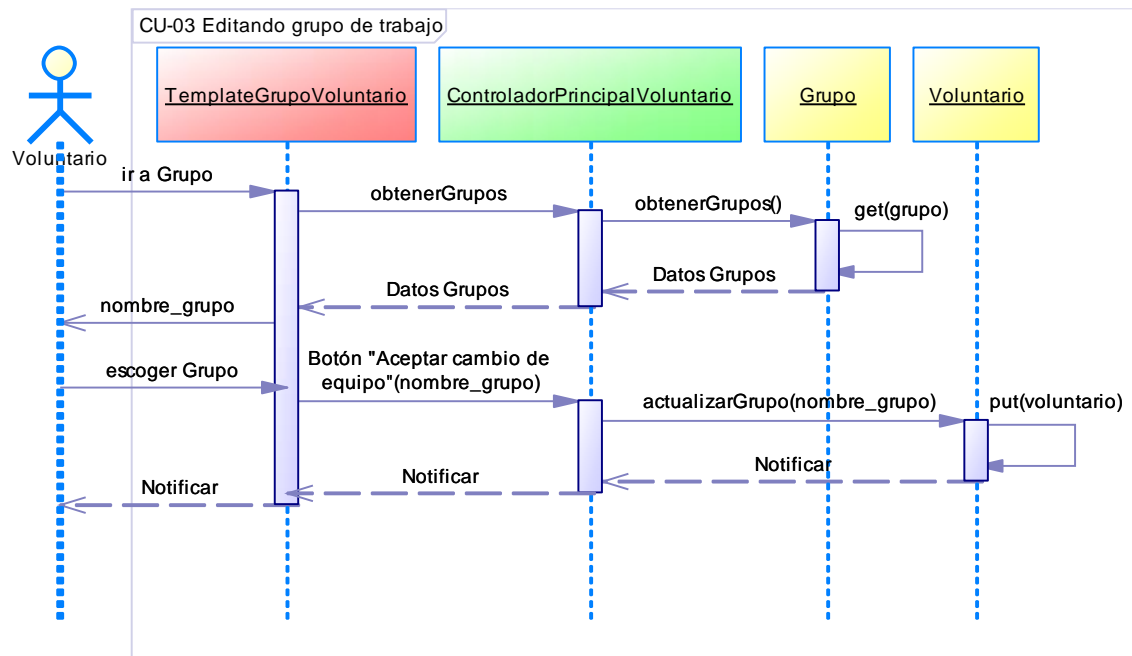


Ilustración 8-5: Diagrama de secuencia Editando grupo de trabajo

- obtenerGrupos(): El sistema realiza la petición a la base de datos para mostrar los grupos registrados a modo de opciones.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla Grupo.
- Botón “Aceptar cambio de equipo”(): Invoca la función actualizarGrupo().
- actualizarGrupo (): Envía la orden de recibir la opción escogida por el usuario en el template TemplateGrupoVoluntario, y lo envía al sistema para que reemplace la opción que el usuario haya escogido anteriormente.
- put(): Método invocado por **Express** para actualizar información registrada en la base de datos. En este caso, reemplaza el valor grupo_voluntario en la tabla Voluntario.

Operación: Editando grupo de trabajo (Ilustración 8-5).

Actor: Voluntario.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: Sin excepciones.

Post-condiciones: La información del grupo de trabajo del usuario queda actualizada.

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Grupo para ingresar template de Grupo. En el template de Grupo se muestra un menú de despliegue donde se muestran los grupos registrados en el sistema a través de la función obtenerGrupos() (nombre_grupo). Si el usuario desea estar afiliado a otro equipo, escoge una de las opciones del menú de despliegue, y presiona el botón “Aceptar cambio de equipo”, llamando a la función actualizarGrupo() que recibe el nombre del grupo escogido en el menú de despliegue, y lo envía a la base de datos, al atributo grupo_voluntario en la tabla Voluntario para reemplazar el valor registrado anteriormente.

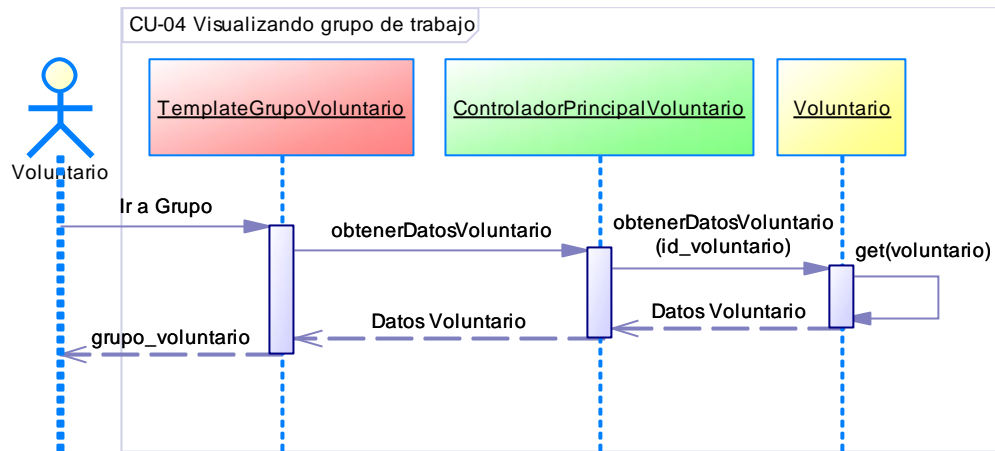


Ilustración 8-6: Diagrama de secuencia Visualizando grupo de trabajo

- obtenerDatosVoluntario(): El sistema realiza la petición a la base de datos para mostrar el grupo en el que el usuario está afiliado.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, En este caso, recupera datos de la tabla Voluntario.

Operación: Visualizando grupo de trabajo (Ilustración 8-6).

Actor: Voluntario.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: Sin excepciones.

Post-condiciones: El grupo o equipo de trabajo del usuario queda visible en la aplicación.

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Grupo para ingresar template de Grupo. En el template de Grupo se muestra el grupo o equipo con el que el usuario se ha afiliado a través de la función obtenerDatosVoluntario(), que recibe el nombre del grupo del usuario desde la tabla Voluntario (grupo_voluntario).

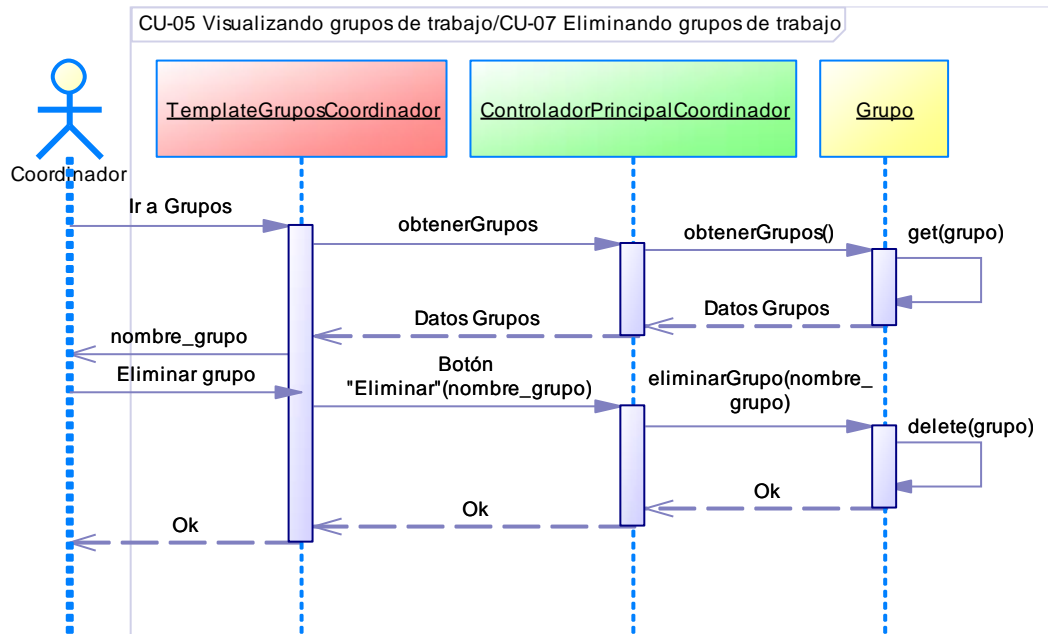


Ilustración 8-7: Diagrama de secuencia Visualizando grupos de trabajo/Eliminando grupo de trabajo

- obtenerGrupos(): El sistema realiza la petición a la base de datos para mostrar los grupos registrados.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla Grupo.
- Botón "Eliminar"(): Invoca la función eliminarGrupo().
- eliminarGrupo(): El sistema realiza la petición a la base de datos para eliminar un grupo o equipo de la base de datos.
- delete(): Método invocado por **Express** para eliminar información de la base de datos. En este caso, elimina datos de la tabla Grupo.

Operación: Visualizando grupos de trabajo/Eliminando grupo de trabajo (Ilustración 8-7).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: No hay grupos o equipos de trabajo registrados.

Post-condiciones: Los grupos o equipos de trabajos quedan visibles en la aplicación. Grupo o equipo es eliminado del sistema

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Grupo para ingresar template de Grupo. En el template de Grupo se muestran los grupos registrados en el sistema a través de la función obtenerGrupos() (nombre_grupo). Al lado de cada grupo mostrado, hay un botón con un ícono que permite eliminar dicho grupo del sistema. Al presionar este botón, se llama a la función eliminarGrupo(), que recibe el nombre del grupo escogido, y lo busca en la base de datos para eliminar su registro.

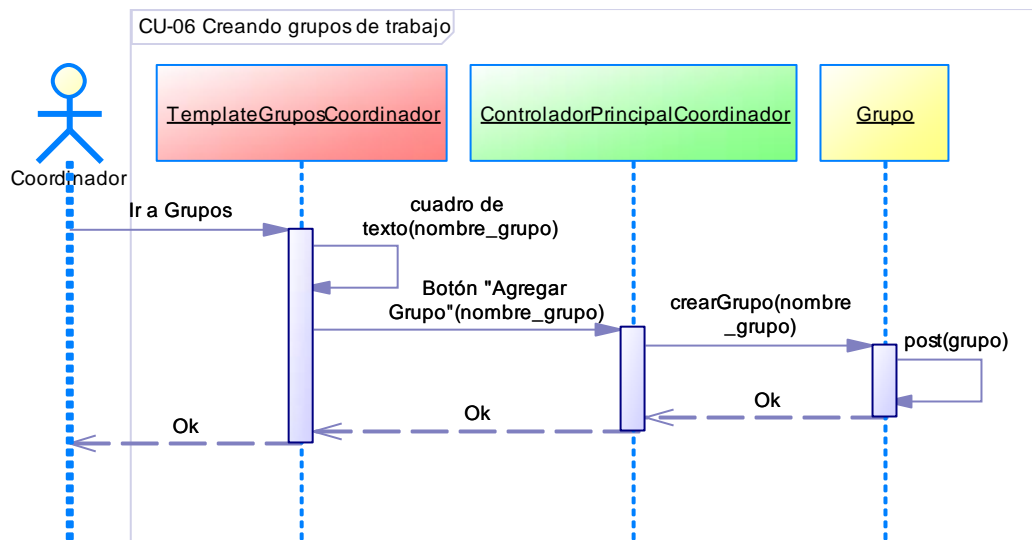


Ilustración 8-8: Diagrama de secuencia Creando grupo de trabajo

- Botón “Agregar Grupo”(): Invoca la función crearGrupo().
- crearGrupo(): Envía la orden de recibir los datos ingresados por el usuario en el cuadro de texto del template TemplateGruposCoordinador, y los envía al sistema para ser almacenados en la base de datos.
- post(): Método invocado por **Express** para almacenar información a la base de datos. En este caso, almacena los datos ingresados por el usuario en la tabla Grupo.

Operación: Creando grupos de trabajo (Ilustración 8-8).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: Sin excepciones.

Post-condiciones: Nuevo grupo o equipo de trabajo es registrado en el sistema.

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Grupo para ingresar al template de Grupo, TemplateGruposCoordinador. En el template de Grupo se muestra un cuadro de texto donde el usuario puede ingresar el nombre del nuevo grupo o equipo a registrar (nombre_grupo). Para registrarlo, presiona el botón “Agregar Grupo”, que llama a la función crearGrupo() para enviar los datos del cuadro de texto a la tabla Grupo en la base de datos, registrándolo en el sistema.

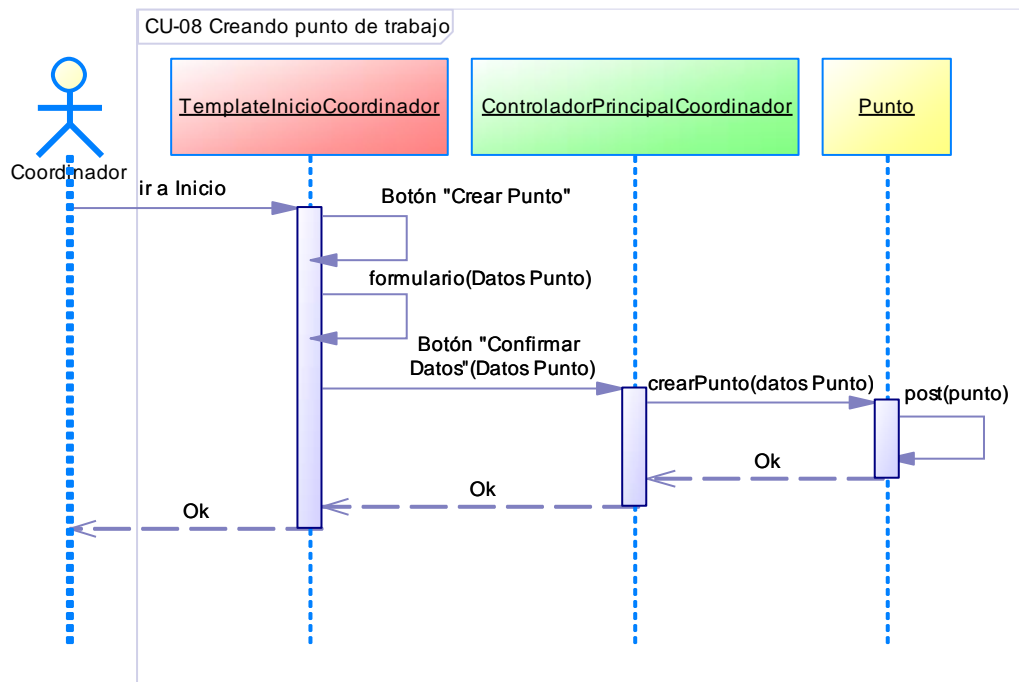


Ilustración 8-9: Diagrama de secuencia Creando punto de trabajo

- Botón “Crear Punto”(): Despliega formulario para ingresar datos de nuevo punto.
- Botón “Confirmar Datos”(): Invoca la función crearPunto().
- crearPunto(): Envía la orden de recibir los datos ingresados por el usuario en el mapa y en el formulario del template TemplateInicioCoordinador, y los envía al sistema para ser almacenados en la base de datos.

- post(): Método invocado por **Express** para almacenar información a la base de datos. En este caso, almacena los datos ingresados por el usuario en la tabla Punto.

Operación: Creando punto de trabajo (Ilustración 8-9).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación.

Excepciones: Sin excepciones.

Post-condiciones: Nuevo punto es registrado en el sistema.

Descripción: El usuario ingresa al template principal de la aplicación, TemplateInicioCoordinador. Aquí se muestra un mapa. El usuario presiona en el mapa para obtener la posición donde se encontrará el marcador (ubicación_punto, dirección_punto), y se muestra el botón “Crear Punto”, que al presionarlo despliega un formulario para que el usuario ingrese los datos sobre el punto a crear (nombre_punto, descripción_punto, cantvoluntarios_punto, tipoactividad_punto, fechainicio_punto, fechafin_punto). Al terminar, presiona el botón “Confirmar Datos”, que llama a la función crearPunto() para enviar los datos del mapa y del formulario a la tabla Punto en la base de datos, registrando el nuevo punto y la actividad de este punto en el sistema.

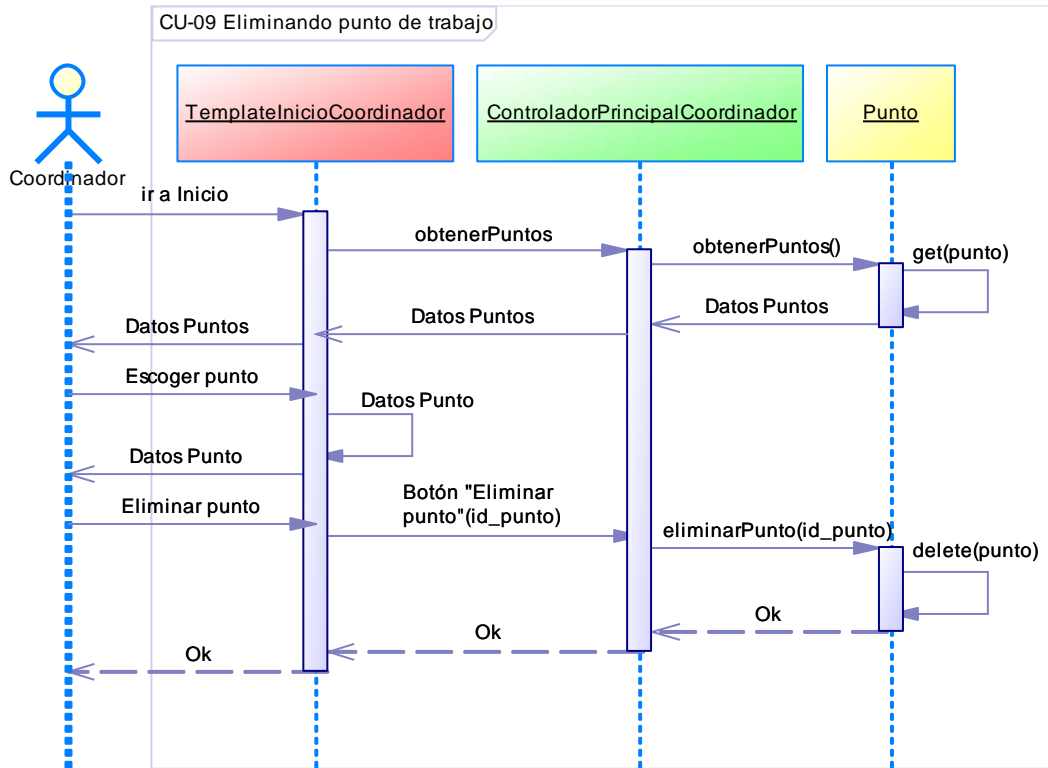


Ilustración 8-10: Diagrama de secuencia Eliminando punto de trabajo

- obtenerPuntos(): El sistema realiza la petición a la base de datos para mostrar los puntos registrados en el mapa.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla Punto.
- Botón "Eliminar Punto": Invoca la función registrar().
- eliminarGrupo(): El sistema realiza la petición a la base de datos para eliminar un punto de la base de datos.
- delete(): Método invocado por **Express** para eliminar información de la base de datos. En este caso, elimina datos de la tabla Grupo.

Operación: Eliminando punto de trabajo (Ilustración 8-10).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación. El usuario debe haber creado por lo menos un punto.

Excepciones: No hay puntos creados por el coordinador. Otro coordinador fue el que creó el punto. Ya hay postulantes al punto de trabajo.

Post-condiciones: Punto es eliminado.

Descripción: El usuario ingresa al template principal de la aplicación, `TemplateInicioCoordinador`. Aquí se muestra un mapa con los puntos existentes representados por marcadores. El usuario presiona uno de los puntos en el mapa, y se muestra la información respectiva a dicho punto, y se despliega el botón “Eliminar punto”. Al presionarlo, se llama a la función `eliminarPunto()`, que recibe un identificador del punto escogido (`id_punto`), y lo busca en la base de datos para eliminar su registro.

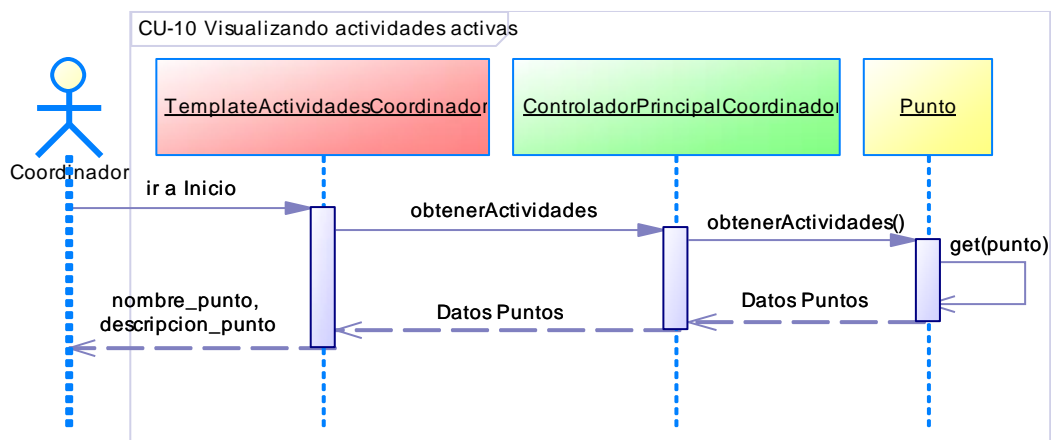


Ilustración 8-11: Diagrama de secuencia Visualizando actividades activas

- `obtenerActividades()`: El sistema realiza la petición a la base de datos para mostrar los puntos registrados.
- `get()`: Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla `Punto`.

Operación: Visualizando actividades activas (Ilustración 8-11).

Actor: Coordinador.

Pre-condiciones: Debe haber por lo menos un punto de trabajo creado.

Excepciones: No hay puntos de trabajo creados.

Post-condiciones: La información de los puntos activos queda visible en la aplicación

Descripción: El usuario ingresa al menú de la aplicación, luego presiona en Actividades para ingresar template de Actividades, `TemplateActividadesCoordinador`. En el template de

Actividades se muestra información sobre los puntos registrados activos en el sistema a través de la función obtenerActividades() (nombre_punto, descripción_punto).

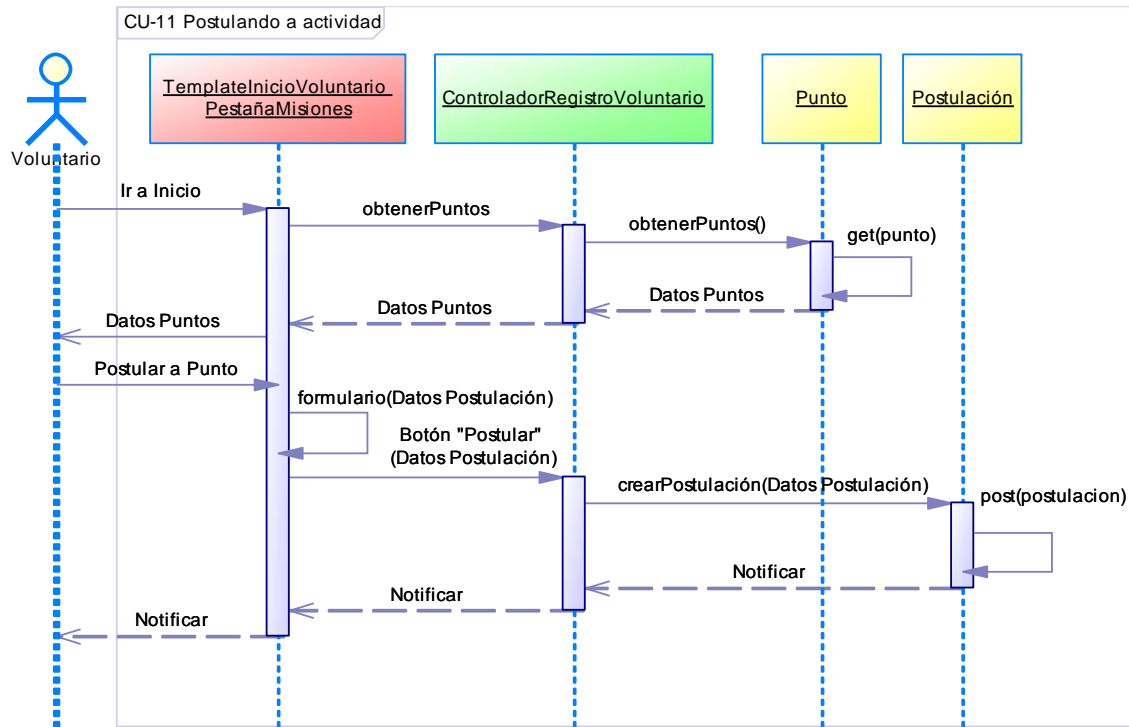


Ilustración 8-12: Diagrama de secuencia Postulando a actividad

- obtenerPuntos(): El sistema realiza la petición a la base de datos para mostrar los puntos registrados en el mapa.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla Punto.
- Botón "Participar()": Despliega el formulario para ingresar los datos sobre la postulación
- Botón "Postular()": Invoca la función crearPostulación().
- crearPostulación(): Envía la orden de recibir los datos ingresados por el usuario en el mapa y en el formulario del template TemplateInicioVoluntario, y los envía al sistema para ser almacenados en la base de datos.
- post(): Método invocado por **Express** para almacenar información a la base de datos. En este caso, almacena los datos ingresados por el usuario en la tabla Postulación.

Operación: Postulando a actividad (Ilustración 8-12).

Actor: Voluntario.

Pre-condiciones: El usuario inició sesión en la aplicación. Debe haber por lo menos un punto de trabajo creado.

Excepciones: No hay puntos de trabajo creados. El usuario ya postuló en otra actividad.

Post-condiciones: Nueva postulación es registrada en el sistema.

Descripción: El usuario ingresa al template principal de la aplicación, TemplateInicioVoluntario, a la pestaña Misiones. Aquí se muestra un mapa. El usuario presiona uno de los puntos en el mapa, y se muestra el botón “Participar”. Al presionarlo, se despliega un formulario para que el usuario ingrese los datos de su postulación (fechainicio_postulación, fechafin_postulación). Al terminar, presiona el botón “Postular”, que llama a la función crearPostulación() para enviar los datos del punto (nombre_punto, id_punto, ubicación_punto), del usuario (email_voluntario, ubicación_voluntario, y ranking_voluntario) y del formulario a la tabla Postulación en la base de datos, registrando la nueva postulación en el sistema.

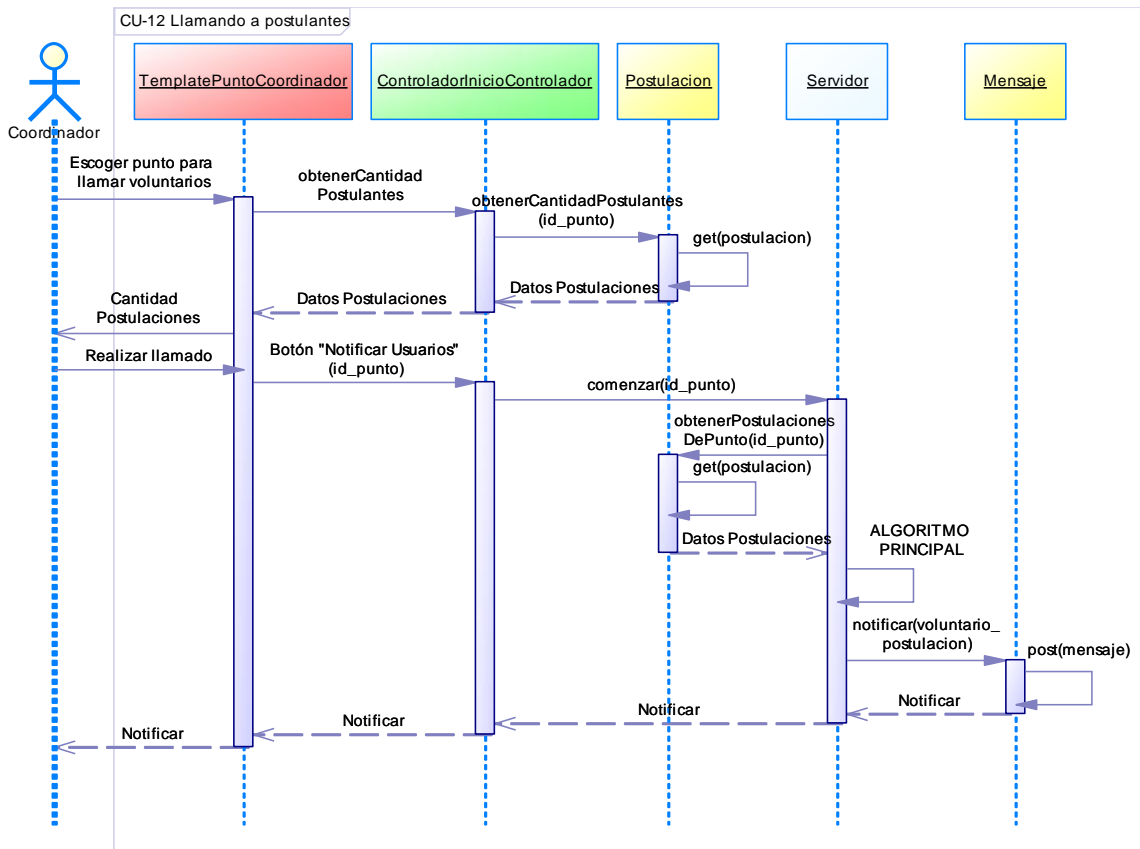


Ilustración 8-13: Diagrama de secuencia Llamando a postulantes

- obtenerCantidadPostulantes(): El sistema realiza la petición a la base de datos para mostrar la cantidad de postulaciones realizadas en un punto específico.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla Postulación.
- Botón “Notificar Usuarios”(): Invoca la función comenzar().
- comenzar(): Inicia la función principal del servidor y del sistema.
- obtenerPostulacionesDePunto(): Función del servidor que realiza la petición a la base de datos para obtener las postulaciones del punto trabajado.
- ALGORITMO PRINCIPAL: Ver Capítulo 4.
- notificar(): Función del servidor que envía la orden de recibir los datos de las postulaciones escogidas, y almacenarlas en la base de datos.
- post(): Método invocado por **Express** para almacenar información a la base de datos. En este caso, almacena los datos de las postulaciones escogidas por el algoritmo en la tabla Mensaje.

Operación: Llamando a postulantes (Ilustración 8-13).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación. Debe haber por lo menos un punto de trabajo creado. Debe haber por lo menos un postulante a trabajar en la actividad correspondiente al punto de trabajo.

Excepciones: No hay puntos creados por el coordinador. Otro coordinador fue el que creó el punto. Ningún voluntario ha postulado a la actividad.

Post-condiciones: Postulantes son avisados de que fueron escogidos. Se visualiza en el template TemplatePuntoCoordinador que los voluntarios postulantes fueron escogidos y avisados. Coordinador recibe en su correo electrónico un listado de los usuarios escogidos.

Descripción: El usuario ingresa al template principal de la aplicación, TemplateInicioVoluntario. Presiona uno de los puntos creados por él en el mapa, y es dirigido al template TemplatePuntoCoordinador. Existe un TemplatePuntoCoordinador por cada punto activo. Aquí se muestra la cantidad de postulaciones a trabajar realizadas por los voluntarios, a través de la función obtenerCantidadPostulantes(). Más abajo, se muestra el botón “Notificar Usuarios” que, al ser presionado, llama a la función comenzar(), que inicia el algoritmo de asignación de voluntarios, pidiendo a la base de datos las postulaciones del punto trabajado a través de la función obtenerPostulacionesDePunto(), analiza las postulaciones, y notifica a los voluntarios cuyas postulaciones fueron aceptadas llamando a la función notificar(), para enviar los datos de

las postulaciones aceptadas a la tabla Mensajes en la base de datos (voluntario_postulación, fechainicio_postulación, ubicación_postulación), registrando las postulaciones aceptadas.

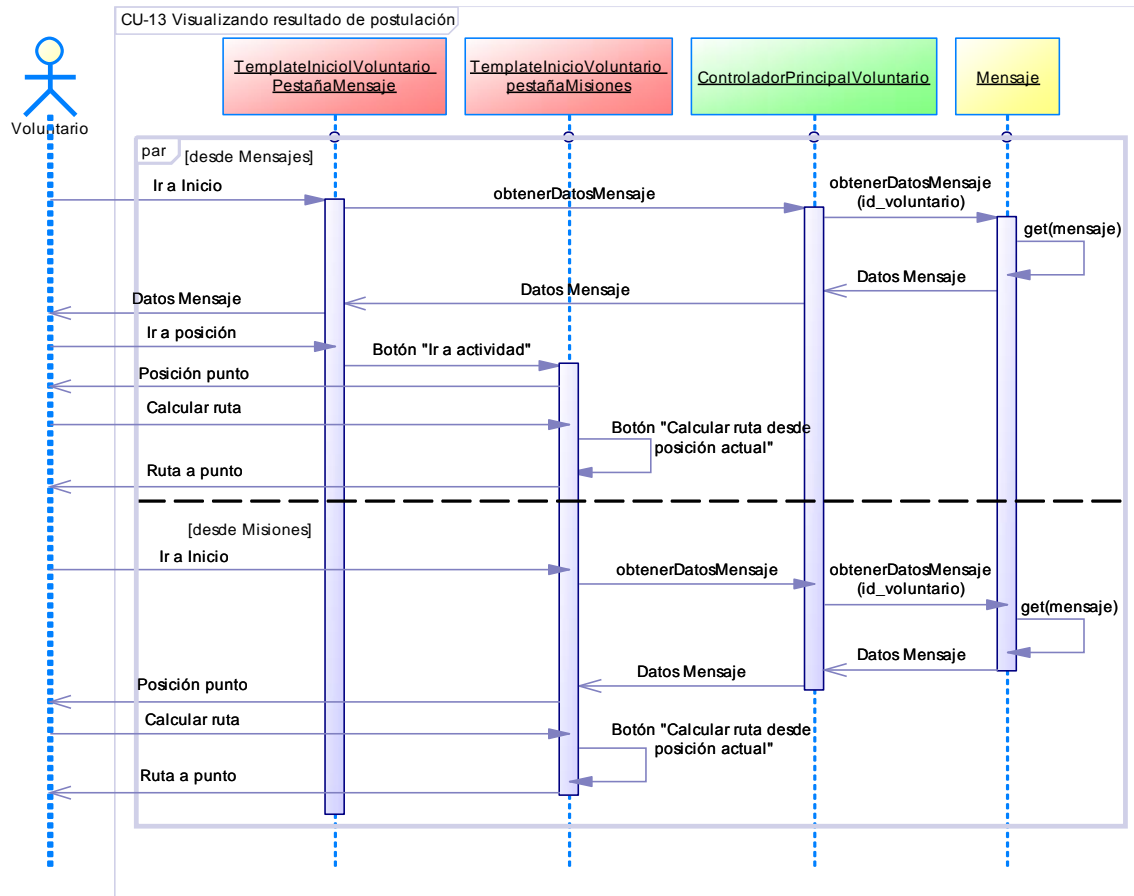


Ilustración 8-14: Diagrama de secuencia Visualizando resultado de postulación

- obtenerDatosMensaje(): El sistema realiza la petición a la base de datos para mostrar información sobre el mensaje y sobre la actividad a la que deberá asistir el voluntario.
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera datos de la tabla Mensaje.
- Botón "Ir a actividad": Lleva al usuario desde la pestaña Mensajes a la pestaña Misiones.
- Botón "Calcular ruta desde posición actual": Cambia lo que muestra el mapa, desde mostrar el punto al que el voluntario debe asistir, hasta mostrar el recorrido desde la posición actual del usuario al punto donde se encuentra la actividad.

Operación: Visualizando resultado de postulación (Ilustración 8-14).

Actor: Voluntario.

Pre-condiciones: El usuario inició sesión en la aplicación. Usuario debe haber postulado a una actividad. Postulación fue aceptada.

Excepciones: Postulación no fue aceptada.

Post-condiciones: Se visualiza en el mapa la posición donde se encuentra el punto de trabajo al que debe asistir el voluntario.

Descripción: (Desde Mensajes): Luego de ser notificado, el usuario ingresa al template principal de la aplicación, TemplateInicioVoluntario, a la pestaña Mensaje. Aquí, se muestra la información obtenida desde la base de datos del mensaje recibido a través de la función obtenerDatosMensaje() (texto_mensaje, fecha_mensaje), y un botón llamado “Ir a actividad”. Al presionar este botón, se lleva al usuario a la pestaña Misiones, donde se carga el punto a partir de los datos del mensaje en el mapa (posactividad_mensaje). Bajo el mapa, se muestra el botón “Calcular ruta desde posición actual”. Al presionar este botón, el mapa cambia para mostrar el recorrido que el usuario debe tomar desde su posición actual hasta el punto al que debe asistir. **(Desde Misiones):** Luego de ser notificado, el usuario ingresa al template principal de la aplicación, TemplateInicioVoluntario, a la pestaña Misiones. Aquí, se muestra el mapa con el punto al que el usuario debe asistir desde los datos obtenidos a través de la función obtenerDatosMensaje() (posactividad_mensaje) de la base de datos. Bajo el mapa, se muestra el botón “Calcular ruta desde posición actual”. Al presionar este botón, el mapa cambia para mostrar el recorrido que el usuario debe tomar desde su posición actual hasta el punto al que debe asistir.

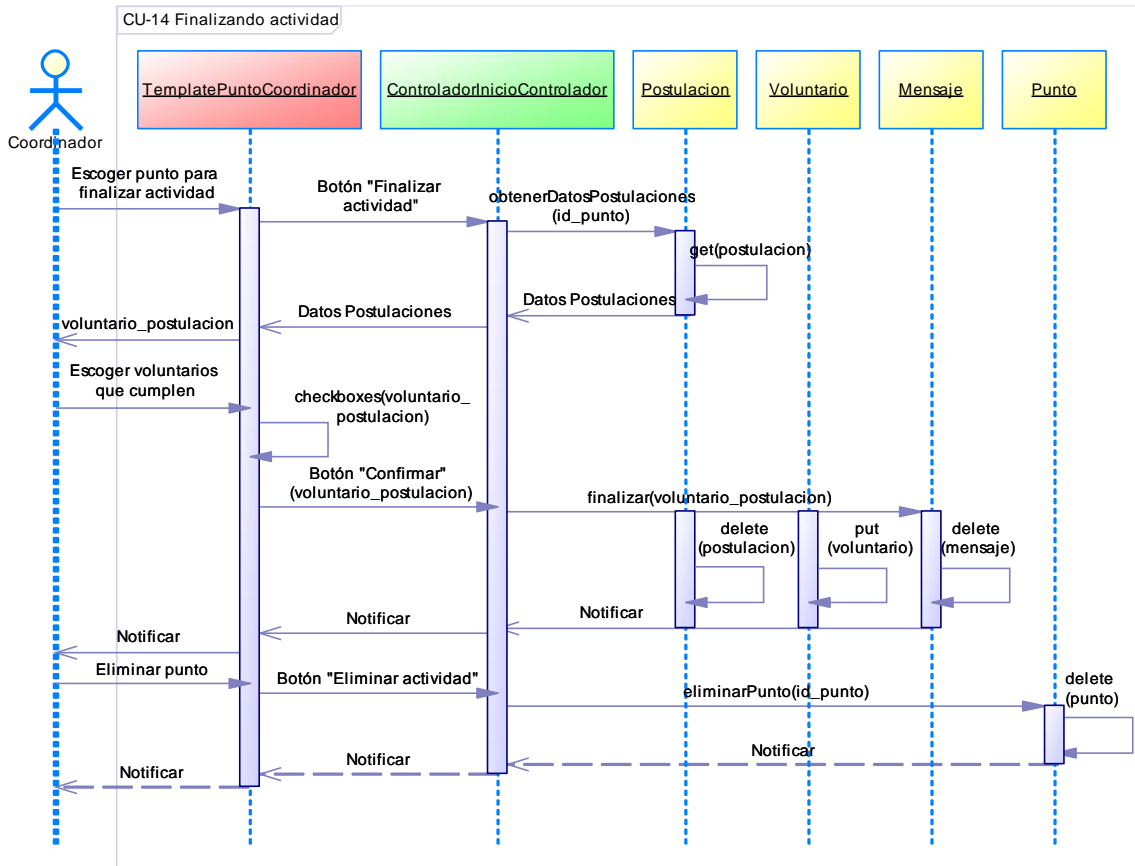


Ilustración 8-15: Diagrama de secuencia Finalizando actividad

- obtenerDatosPostulantes(): El sistema realiza la petición a la base de datos para mostrar los correos electrónicos de los postulantes que fueron llamados a la actividad (participantes).
- get(): Método invocado por **Express** para recuperar información de la base de datos. En este caso, recupera los datos de la tabla Postulación.
- checkboxes(): Listado de los voluntarios participantes para seleccionar a los que cumplieron con el trabajo, con un *checkbox* asociado a cada uno.
- Botón "Confirmar"(): Invoca la función finalizar().
- finalizar(): recibe los correos electrónicos de los participantes que fueron aprobados por el usuario, y envía la orden de eliminar la postulación, actualizar el valor *ranking* del voluntario participante, y eliminar el mensaje recibido por el voluntario participante.
- delete(): Método invocado por **Express** para eliminar información de la base de datos. En este caso, elimina datos de la tabla Postulación, Mensaje y Punto.
- put(): Método invocado por **Express** para actualizar información registrada en la base de datos. En este caso, reemplaza el valor *ranking_voluntario* en la tabla Voluntario.

Operación: Finalizando actividad (Ilustración 8-15).

Actor: Coordinador.

Pre-condiciones: El usuario inició sesión en la aplicación. La actividad correspondiente al punto de trabajo en el mapa debe haber finalizado. El punto debe haber sido creado por el usuario.

Excepciones: Otro coordinador fue el que creó el punto.

Post-condiciones: Punto y marcador es eliminado. Postulaciones y Mensajes son eliminadas. Valor *Ranking* de voluntarios escogidos aumenta en 1.

Descripción: El usuario ingresa al template principal de la aplicación, TemplateInicioVoluntario. Presiona uno de los puntos creados por él en el mapa, y es dirigido al template TemplatePuntoCoordinador. Existe un TemplatePuntoCoordinador por cada punto activo. Aquí, se muestra el botón “Finalizar actividad” que, al ser presionado, llama a la función obtenerDatosPostulaciones(), que recibe los datos de la tabla Postulación para mostrar los correos electrónicos de los voluntarios participantes en una lista de *checkboxes* (voluntario_postulación). Escoge a los voluntarios que cumplieron de manera óptima con el trabajo pedido marcando el *checkbox* correspondiente, y presiona el botón “Confirmar” para llamar a la función finalizar(). La función finalizar() recibe a los participantes que escogió el usuario, y aumenta en una unidad el valor ranking_voluntario de cada voluntario participante en la tabla Voluntario. Además, elimina las postulaciones y los mensajes relacionados al punto de las tablas Postulación (voluntario_postulación) y Mensaje (voluntario_mensaje), respectivamente. Luego, presiona el botón “Eliminar actividad”, que termina de eliminar la información del punto de la tabla Punto, y su marcador en el mapa.

8.2 Algoritmo

Especificar tamaño e información de Itinerario:

```
//La matriz Itinerario será de 3 dimensiones: cantidadDías x
//cantidadVoluntariosRequeridosPorDía x 2.
Itinerario = [ ];
//Columnas: Cantidad de días que durarán los trabajos.
Por i, desde i = 0 hasta i ≤ cantidadDías {
    Itinerario = [ ];
    //Filas: Cantidad de voluntarios requeridos por día = cupos.
    Por j, desde j = 0 hasta j ≤ cantidadVoluntariosRequeridosPorDía {
```



```

//Se reserva un array de 2 espacios para indicar entrada y salida de voluntario por
día.
Itinerario[ i ][ j ] = [ ];
//Se guardan las fechas en cada espacio.
Itinerario[ i ][ j ][ 0 ] = new Fecha(díaInicioActividad);
Itinerario[ i ][ j ][ 1 ] = new Fecha(díaInicioActividad);
}
díaInicioActividad++;
}

```

Ordenar voluntarios postulantes:

//Función que ordena el *array* de voluntarios postulantes. 1ro por ranking, luego por distancia al punto de trabajo.
 Ordenar(voluntario ranking || voluntario.distanciaAPunto);

Algoritmo principal:

```

Seleccionados = [ ];
Contador = 0;
Contador2 = 0;
Contador3 = 0;
//Por cada voluntario postulante, recorrer array de voluntarios postulantes.
Por i, desde i = 0; hasta i ≤ cantidadVoluntariosPostulantes {
    //INICIO PRIMERA PARTE
    //Recorrer filas de matriz Itinerario.
    Por k, desde k = 0, hasta k ≤ cantidadVoluntariosRequeridosPorDía {
        // Recorrer columnas de matriz Itinerario.
        Por j, desde j = 0, hasta j ≤ cantidadDías {
            //Ver si 1. El voluntario aún no es analizado por el algoritmo, 2. Si la fecha
            analizada en el Itinerario esta libre, y 3. Si la fecha de inicio de actividad que escogió el
            voluntario coincide con la fecha del itinerario analizada.
            Si ((voluntario no ha sido analizado) && (itinerario[ j ][ k ][ 0 ] es fecha libre)
            && (itinerario[ j ][ k ][ 0 ] = voluntario[ i ].fechaInicio)) {
                //Contar desde la fecha escogida del itinerario para ver si hay un
                periodo libre desde esa fecha según la cantidad de días que el voluntario especificó que
                asistirá.
                Contador = 0;
                Por l, desde l = 0 hasta l ≤ díasPropuestosPorVoluntario {
                    Si ((l < cantidadDíasPropuestosPorVoluntario) && (itinerario[ j+1
                    ][ k ][ 0 ] es fecha libre)){
                        Contador++;
                    }
                    Si ((l == cantidadDíasPropuestosPorVoluntario) && (itinerario[
                    j+1 ][ k ][ 1 ] es fecha libre)){
                        Contador++;
                    }
                }
            }
        }
    }
}

```

```

        //Si se encuentra un periodo de tiempo libre que coincida con los días
        especificadas en la postulación, y si el voluntario aún no es asignado.
        Si ((Contador == cantidadDíasPropuestosPorVoluntario) &&
        (voluntario no ha sido asignado)) {
            //Postulación es marcada.
            Voluntario asignado;
            //Se reserva el periodo de tiempo para el voluntario analizado.
            Por l, desde l = 0 hasta l ≤
            cantidadDíasPropuestosPorVoluntario {
                //Si solo trabaja un día, se marca para el mismo día del
                itinerario la entrada y la salida de los trabajos.
                Si solo trabaja un día {
                    ltinerario[ j+l ][ k ][ 0 ] = voluntario;
                    ltinerario[ j+l ][ k ][ 1 ] = voluntario;
                }
                En otro caso {
                    //En otro caso, se asignan como entrada todos los
                    días reservados, excepto el último día, que es el día en que el voluntario se retira.
                    Si (l < cantidadDíasPropuestosPorVoluntario) {
                        ltinerario[ j+l ][ k ][ 0 ] = voluntario;
                        //voluntario asignado;
                    }
                    En otro caso {
                        ltinerario[ j+l ][ k ][ 1 ] = voluntario;
                        //voluntario asignado
                    }
                }
            }
            //Marcar voluntario. Almacenar token del voluntario en array.
            Seleccionados[ Contador3 ] = voluntario;
            Contador3++;
        }
        //Se reinicia contador.
        Contador = 0;
    }
}
//FIN DE PRIMERA PARTE
//INICIO DE SEGUNDA PARTE.
//Recorrer filas de matriz ltinerario.
Por k, desde k = 0, hasta k ≤ cantidadVoluntariosRequeridosPorDía {
    //Recorrer columnas de matriz ltinerario.
    Por j, desde j = 0, hasta j ≤ cantidadDías {
        //Si el voluntario no fue registrado en la primera parte del algoritmo
        Si ((voluntario aún no es asignado) && (voluntario aún no es analizado)) {
            //Mientras la cantidad de días en que el voluntario desee asistir sea
            mayor a 4 días, más un día cada 500 kilómetros de distancia al punto de trabajo.
            Mientras (cantidadDíasPropuestosPorVoluntario ≥ (4 + voluntario[ i
            ].distanciaAPunto / 500)) {
                //Aplazar la llegada del voluntario un día.
                voluntario[ i ].fechaInicio++;
                Contador2++;
            }
        }
    }
}

```

//Ver si 1. El voluntario aún no es analizado por el algoritmo, 2. Si la fecha analizada en el Itinerario esta libre, y 3. Si la fecha de inicio de actividad que escogió el voluntario coincide con la fecha del itinerario analizada.

Si ((voluntario no ha sido analizado) && (itinerario[j][k][0] es fecha libre) && (itinerario[j][k][0] = voluntario[i].fechaInicio)) {

//Contar desde la fecha escogida del itinerario para ver si hay un periodo libre desde esa fecha según la cantidad de días que el voluntario especificó que asistirá.

Contador = 0;

Por l, desde l = 0 hasta l ≤ díasPropuestosPorVoluntario {

Si ((l < cantidadDíasPropuestosPorVoluntario) &&

(itinerario[j+1][k][0] es fecha libre)){

Contador++;

}

Si ((l == cantidadDíasPropuestosPorVoluntario) &&

(itinerario[j+1][k][1] es fecha libre)){

Contador++;

}

}

//Si se encuentra un periodo de tiempo libre que coincida con los días especificadas en la postulación, y si el voluntario aún no es asignado.

Si ((Contador == cantidadDíasPropuestosPorVoluntario) && (voluntario no ha sido asignado)) {

//Postulación es marcada.

Voluntario asignado;

//Se reserva el periodo de tiempo para el voluntario

analizado.

Por l, desde l = 0 hasta l ≤

cantidadDíasPropuestosPorVoluntario {

//Si solo trabaja un día, se marca para el mismo día del itinerario la entrada y la salida de los trabajos.

Si solo trabaja un día {

Itinerario[j+l][k][0] = voluntario;

Itinerario[j+l][k][1] = voluntario;

}

//En otro caso, se asignan como entrada todos los días reservados, excepto el último día, que es el día en que el voluntario se retira.

En otro caso {

Si (l <

cantidadDíasPropuestosPorVoluntario) {

Itinerario[j+l][k][0] =

voluntario;

//voluntario asignado;

}

En otro caso {

Itinerario[j+l][k][1] =

voluntario;

//voluntario asignado

}

}

}

```

en array.
//Marcar voluntario. Almacenar token del voluntario
Seleccionados[ Contador3 ] = voluntario;
Contador3++;
}
}
}
}
//Se devuelve la fecha de inicio de actividad de la postulación a su fecha
original para buscar otra fecha en el Itinerario.
voluntario[ i ].fechaInicio = voluntario[ i ].fechaInicio - Contador2;
Contador2 = 0;
}
}
}
}

```

8.3 Pruebas

8.3.1 Pruebas unitarias

Tabla 8-1: Pruebas unitarias, tabla Voluntario

Paquete	Nombre prueba unitaria
C_servidorVoluntario	CA_PostVoluntario
C_servidorVoluntario	CB_UpdateRankingVol
C_servidorVoluntario	CC_UpdateGrupoVol
C_servidorVoluntario	CD_GetVoluntarioPorEmail

Tabla 8-2: Pruebas unitarias, tabla Coordinador

Paquete	Nombre prueba unitaria
D_servidorCoordinador	DA_PostCoordinador
D_servidorCoordinador	DB_GetCoordinadorPorEmail

Tabla 8-3: Pruebas unitarias, tabla Punto

Paquete	Nombre prueba unitaria
E_servidorPunto	EA_PostPunto
E_servidorPunto	EB_GetPuntos
E_servidorPunto	EC_GetPuntoPorId

E_servidorPunto	ED_UpdateEstadoPunto
E_servidorPunto	EE_DeletePuntoPorId

Tabla 8-4: Pruebas unitarias, tabla Postulación

Paquete	Nombre prueba unitaria
F_servidorPostulación	FA_PostPostulación
F_servidorPostulación	FB_GetPostulaciones
F_servidorPostulación	FC_GetPuntoPorId
F_servidorPostulación	FD_DeletePostulacionPorVol
F_servidorPostulación	FD_DeletePostulaciónPorId

Tabla 8-5: Pruebas unitarias, tabla Grupo

Paquete	Nombre prueba unitaria
G_servidorGrupo	GA_PostGrupo
G_servidorGrupo	GB_GetGrupos
G_servidorGrupo	GC_DeleteGrupoPorNombre

Tabla 8-6: Pruebas unitarias, tabla Mensaje

Paquete	Nombre prueba unitaria
H_servidorMensaje	HA_PostMensaje
H_servidorMensaje	HB_GetMensajePorVol
H_servidorMensaje	HC_DeleteMensajePorVol

Resultados Pruebas Unitarias:

Tabla 8-7: Resultado pruebas unitarias generales: prototipo Coordinador

Pruebas B_aplicaciónCoordinador

Cantidad de pruebas: 17 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
BA_CargarMapa	Cargar y visualizar mapa de Google Maps.	Se muestra el mapa en la vista.	✓
BB_CentrarPosición	Utilizar botón para que el mapa muestre la posición global del usuario.	Se muestra posición del usuario en el mapa.	✓
BC_CentrarPosiciónInicial	Al cargar el mapa, mostrar una posición arbitraria.	Se muestra posición predefinida en el mapa.	✓
BD_CargarMarcador	Mostrar los puntos de trabajo gráficamente en el mapa con marcadores.	Se muestra marcador en el mapa.	✓
BE_CargarInfowindow	Desplegar una ventana informativa al presionar un marcador.	Se muestra ventana con información sobre marcador.	✓
BF_ObtenerLatYLong	Obtener valores de latitud y longitud al presionar una posición en el mapa al crear una cuenta de usuario.	Se obtiene latitud y longitud.	✓
BG_ValidarRegistroCuenta	Asegurar que la creación de una nueva cuenta de coordinador fue exitosa.	Vista Login se muestra en la aplicación con aviso de éxito.	✓
BH_ValidarInicioSesión	Asegurar que el inicio de sesión del usuario en el sistema fue exitoso.	Vista Inicio se muestra en la aplicación con aviso de éxito.	✓
BI_ObtenerFechaPunto	Obtener la fecha especificada por el coordinador al crear el punto.	Se recibe fecha en formato <i>DateTime</i> .	✓
BJ_ValidarCrearPunto	Asegurar que el punto fue creado.	Se muestra aviso de éxito.	✓
BK_ValidarEliminarPunto	Asegurar que la eliminación de un punto fue exitosa.	Se muestra aviso de éxito.	✓
BL_IniciarLlamado	Utilizar un botón para iniciar algoritmo y llamar a voluntarios postulantes a un punto.	Se muestra aviso de éxito.	✓
BM_RecibirEmail	Generar un correo electrónico con listado de participantes.	Coordinador recibe correo electrónico.	✓
BN_ConfirmarParticipaciones	Listar participantes y usar <i>checkboxes</i> para especificar voluntarios que cumplieron con su trabajo.	Se muestra aviso de éxito.	✓
BO_FinalizarActividad	Utilizar un botón para finalizar la actividad.	Se muestra aviso de éxito.	✓
BP_ValidarCrearGrupo	Asegurar que creación de grupo de trabajo fue exitosa.	Se muestra aviso de éxito.	✓
BQ_ValidarEliminarGrupo	Asegurar que la eliminación de un grupo fue exitosa.	Se muestra aviso de éxito.	✓

Tabla 8-8: Resultado pruebas generales: tabla Voluntario

Pruebas C_servidorVoluntario

Cantidad de pruebas: 4 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
CA_PostVoluntario	Registrar nueva cuenta de usuario Voluntario.	Nuevo Voluntario es creado.	✓
CB_UpdateRankingVol	Cuando un voluntario es evaluado positivamente al terminar una actividad, aumenta su valor <i>Ranking</i> .	Valor ranking_voluntario cambia a 1.	✓
CC_UpdateGrupoVol	Cada voluntario puede modificar el grupo o equipo de trabajo para el que está afiliado.	Valor grupo_voluntario es actualizado.	✓
CD_GetVoluntarioPorEmail	Obtener datos sobre el voluntario, por ejemplo, para mostrarlos en la vista Perfil.	Datos de tabla Voluntario son visibles.	✓

Tabla 8-9: Resultado pruebas generales: tabla Coordinador

Pruebas D_servidorCoordinador

Cantidad de pruebas: 2 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
DA_PostCoordinador	Registrar nueva cuenta de usuario Coordinador.	Nuevo Coordinador es creado.	✓
DB_GetCoordinadorPorEmail	Obtener datos sobre el coordinador, por ejemplo, para mostrarlos en la vista Perfil o para filtrar puntos que puede gestionar.	Datos de Coordinador son visibles.	✓

Tabla 8-10: Resultado pruebas generales: tabla Punto

Pruebas E_servidorPunto

Cantidad de pruebas: 5 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
EA_PostPunto	Registrar nuevo punto de trabajo	Nuevo Punto es creado.	✓
EB_GetPuntos	Obtener todos los puntos almacenados para mostrarlos en el mapa en forma de marcadores.	Marcadores mostrados en el mapa.	✓

EC_GetPuntoPorId	Obtener información sobre un punto en específico, por ejemplo, para mostrar <i>Infowindow</i> al presionar en marcador.	Datos de Punto quedan visibles.	✓
ED_UpdateEstadoPunto	Cuando el coordinador realiza el llamado a un punto, se debe marcar el punto para que voluntarios no puedan postular a él.	Valor llamado_punto pasa de 0 a 1.	✓
EE_DeletePuntoPorId	Eliminar un punto en específico.	Punto es eliminado.	✓

Tabla 8-11: Resultado pruebas generales: tabla Postulación

Pruebas F_servidorPostulación

Cantidad de pruebas: 5 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
FA_PostPostulación	Registrar nuevo punto de trabajo	Nuevo Punto es creado.	✓
FB_GetPostulaciones	Obtener todos los puntos almacenados para mostrarlos en el mapa en forma de marcadores.	Marcadores mostrados en el mapa.	✓
FC_GetPuntoPorId	Obtener información sobre un punto en específico, por ejemplo, para mostrar <i>Infowindow</i> al presionar en marcador.	Datos de Punto quedan visibles.	✓
FD_DeletePostulacionPorVol	Eliminar las postulaciones a un punto al finalizar la actividad.	Postulaciones son eliminadas.	✓
FD_DeletePostulaciónPorId	Eliminar las postulaciones a un punto, al eliminar el punto antes de iniciar el llamado.	Postulaciones son eliminadas.	✓

Tabla 8-12: Resultado pruebas generales: tabla Grupo

Pruebas G_servidorGrupo

Cantidad de pruebas: 3 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
GA_PostGrupo	Registrar nuevo grupo o equipo de trabajo.	Nuevo Grupo es creado.	✓
GB_GetGrupos	Obtener todos los grupos.	Grupos mostrados en la aplicación.	✓
GC_DeleteGrupoPorNombre	Eliminar un grupo en específico.	Grupo es eliminado.	✓

Tabla 8-13: Resultado pruebas generales: tabla Mensaje

Pruebas H_servidorMensaje

Cantidad de pruebas: 3 Fallas: 0

Tests	Descripción	Resultado esperado	Estado
HA_PostMensaje	Registrar mensaje.	Nuevo Mensaje es creado.	✓
HB_GetMensajePorVol	Obtener el mensaje para cada voluntario al ser llamado a una actividad.	Mensaje mostrado en la aplicación.	✓
HC_DeleteMensajePorVol	Eliminar los mensajes de los voluntarios participantes en una actividad finalizada..	Mensaje es eliminado.	✓

8.3.2 Pruebas de integración

Tabla 8-14: Plan de prueba: Crear Punto

Ítem	Descripción
Identificación	Plan de prueba 03 – Crear Punto
Elementos a probar	Crear tabla Punto, obtener puntos, cargar mapa, centrar posición, cargar marcador, cargar <i>infowindow</i> , obtener latitud y longitud, obtener fecha punto, validar punto, interfaz del punto.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Queda la información del Punto guardada en el sistema.
Tareas para satisfacer el proceso	Se ingresa la información del punto. Validar que la información queda correctamente guardada.
Descripción	Se inicia la creación de un punto de trabajo, presionando un punto en el mapa e ingresando el nombre de la actividad, la descripción, la cantidad de voluntarios necesarios, el tipo de actividad, la fecha de inicio, y la duración en días. Finalmente, se guarda la información en el sistema.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.
Riesgos	No hay conexión a la API de Google Maps. De no haber conexión, no se podrá especificar la posición del punto.

Tabla 8-15: Plan de prueba: Crear Postulación

Ítem	Descripción
Identificación	Plan de prueba 04 – Crear Postulación
Elementos a probar	Crear tabla Postulación, obtener puntos, cargar mapa, centrar posición, cargar marcador, cargar <i>infowindow</i> , obtener latitud y longitud, obtener punto por id, obtener fecha postulación, obtener <i>token</i> , validar postulación.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Queda la información de la Postulación guardada en el sistema.
Tareas para satisfacer el proceso	Se ingresa la información de la postulación. Validar que la información queda correctamente guardada. Voluntario no puede crear una nueva postulación.
Descripción	Se inicia la creación de una postulación a un punto, presionando un marcador en el mapa e ingresando la fecha de inicio y la fecha de fin en los que quiere asistir el voluntario. Finalmente, se guarda la información en el sistema.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.
Riesgos	No hay conexión a la API de Google Maps. De no haber conexión, no se podrá especificar el punto de trabajo.

Tabla 8-16: Plan de prueba: Iniciar trabajos en un Punto

Ítem	Descripción
Identificación	Plan de prueba 05 – Iniciar trabajos en un Punto
Elementos a probar	Cargar mapa, centrar posición, cargar marcador, cargar <i>infowindow</i> , obtener latitud y longitud, obtener postulaciones, iniciar llamado, actualizar estado punto, recibir email, confirmar participaciones, crear mensaje, recibir notificación, encontrar tarea, calcular ruta a tarea.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Queda la información del Mensaje guardado en el sistema. Valor de Llamado de Punto actualizado al iniciar el algoritmo.
Tareas para satisfacer el proceso	Se inicia el llamado de voluntarios. Validar que el llamado fue realizado. Validar que Voluntario recibe Notificación. Ver actividad para la que Voluntario fue llamado en el mapa.
Descripción	Se inicia llamado de voluntarios presionando un marcador en el mapa y luego un botón. Se muestra que el llamado fue realizado. Se verifica que Voluntario fue llamado.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.

Ítem	Descripción
Riesgos	No hay conexión a la API de Google Maps. De no haber conexión, no se podrá especificar la posición del punto.

Tabla 8-17: Plan de prueba: Terminar trabajos en un Punto

Ítem	Descripción
Identificación	Plan de prueba 06 –Terminar trabajos en un Punto
Elementos a probar	Cargar mapa, centrar posición, cargar marcador, cargar infowindow, obtener latitud y longitud, obtener postulaciones, finalizar actividad, actualizar ranking por voluntario, eliminar mensaje por voluntario, eliminar postulación por id.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Valor de ranking de Voluntarios aumenta una unidad. Eliminar Mensajes, Postulaciones, y Punto al terminar la actividad.
Tareas para satisfacer el proceso	Confirmar voluntarios participantes. Finalizar actividad.
Descripción	Se presiona botón para finalizar la actividad. Se escogen los participantes que cumplieron su trabajo. Se finaliza actividad. Se actualiza el valor de ranking de los participantes, se eliminan los mensajes de los participantes, se eliminan las postulaciones de los participantes. Se elimina el punto.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.
Riesgos	No hay conexión a la API de Google Maps. De no haber conexión, no se podrá especificar la posición del punto.

Tabla 8-18: Plan de prueba: Crear Grupo

Ítem	Descripción
Identificación	Plan de prueba 07 – Crear Grupo
Elementos a probar	Crear tabla Grupo, validar grupo.
Enfoque	<i>Down-Top</i>
Criterio de aceptación	Queda la información del Grupo guardada en el sistema.
Tareas para satisfacer el proceso	Se ingresa la información del grupo. Validar que la información queda correctamente guardada.
Descripción	Se inicia la creación de un grupo, ingresando el nombre del grupo. Finalmente, se guarda la información en el sistema.
Responsabilidades	Desarrollador: Crear plan de prueba. Desarrollador: Ejecutar plan de prueba. Desarrollador: Validar plan de prueba.

Ítem	Descripción
Riesgos	Sin riesgos.

8.3.3 Pruebas de validación

Tabla 8-19: Caso de prueba para la creación de una postulación

Elemento plantilla	Descripción
Identificador	Caso de prueba-02
Casos de uso asociados	CU-11
Módulo/funcionalidad	Crear postulación.
Descripción	El caso de prueba consiste en la postulación de un voluntario para su participación en un punto de trabajo, detallando las condiciones para su aceptación.
Pre-condiciones	Se debe estar en la vista principal de la aplicación Voluntario. Deben existir puntos de trabajo mostrados en el mapa.
Secuencia de pasos	1.- Se presiona un punto en el mapa. 2.- Se ingresa la información solicitada por el sistema para la creación de la postulación. 3.- Se confirma la creación de la postulación. 4.- Se verifica que no puede crearse una nueva postulación.
Resultado esperado	Se espera que la aplicación notifique visualmente la creación exitosa de la nueva postulación.
Automatizado	No.
Resultado efectivo	Aceptado.
Responsable	Desarrollador
Fecha de creación	02/07/2016

Tabla 8-20: Caso de prueba para iniciar el llamado de un punto de trabajo

Elemento plantilla	Descripción
Identificador	Caso de prueba-03
Casos de uso asociados	CU-12
Módulo/funcionalidad	Iniciar trabajos en un punto, Crear Punto, Crear Postulación.
Descripción	El caso de prueba consiste en iniciar el llamado de voluntarios postulantes a trabajar a un punto.
Pre-condiciones	Se debe estar en la vista principal de la aplicación Coordinador. Deben existir puntos de trabajo mostrados en el mapa.

Elemento plantilla	Descripción
Secuencia de pasos	1.- Se presiona el punto en el mapa donde se quiere llamar a los voluntarios. 2.- Se dirige a la vista del punto de trabajo. 3.- Se presiona el botón que inicia el llamado. 4.- Se verifica que el llamado fue realizado.
Resultado esperado	Se espera que la aplicación notifique visualmente que el llamado fue realizado.
Automatizado	No.
Resultado efectivo	Aceptado.
Responsable	Desarrollador
Fecha de creación	02/07/2016

Tabla 8-21: Caso de prueba para finalizar la actividad de un punto de trabajo

Elemento plantilla	Descripción
Identificador	Caso de prueba-04
Casos de uso asociados	CU-14
Módulo/funcionalidad	Terminar trabajos en un punto.
Descripción	El caso de prueba consiste en finalizar los trabajos en un punto y evaluar a los participantes.
Pre-condiciones	Se debe estar en la vista principal de la aplicación Coordinador. Los trabajos en un punto deben haber concluido.
Secuencia de pasos	1.- Se presiona el punto en el mapa donde se quieren finalizar los trabajos. 2.- Se dirige a la vista del punto de trabajo. 3.- Se presiona el botón que finaliza la actividad. 4.- Se escogen los voluntarios que cumplieron su trabajo. 5.- Se presiona el botón que confirma la selección 6.- Se elimina el punto.
Resultado esperado	Se espera que la aplicación notifique visualmente que el punto fue eliminado. El valor <i>ranking</i> de los voluntarios participantes aumentó.
Automatizado	No.
Resultado efectivo	Aceptado.
Responsable	Desarrollador
Fecha de creación	02/07/2016

Tabla 8-22: Caso de prueba para la creación de un grupo

Elemento plantilla	Descripción
Identificador	Caso de prueba-05
Casos de uso asociados	CU-06
Módulo/funcionalidad	Crear grupo.
Descripción	El caso de prueba consiste en la creación de un grupo o equipo de trabajo, y su visualización en el listado de grupos.
Pre-condiciones	Se debe estar en la vista principal de la aplicación Coordinador.
Secuencia de pasos	1.- Se dirige a la vista de Grupos a través del menú de navegación de la aplicación. 3.- Se ingresa el nombre del grupo. 3.- Se confirma la creación del grupo. 4.- Se verifica que el grupo fue creado.
Resultado esperado	Se espera que la aplicación notifique visualmente la creación exitosa del nuevo grupo.
Automatizado	No.
Resultado efectivo	Aceptado.
Responsable	Desarrollador
Fecha de creación	02/07/2016

Tabla 8-23: Caso de prueba para el manejo del perfil del voluntario

Elemento plantilla	Descripción
Identificador	Caso de prueba-06
Casos de uso asociados	CU-01, CU-02
Módulo/funcionalidad	Perfil Voluntario.
Descripción	El caso de prueba consiste en crear y visualizar la información de perfil de un voluntario.
Pre-condiciones	Sin pre-condiciones.
Secuencia de pasos	1.- Se inicia la aplicación. 2.- Se dirige a la vista de registro. 3.- Se ingresa la información solicitada por el sistema para la creación del voluntario. 3.- Se confirma la creación del voluntario. 4.- Se verifica que el voluntario fue creado.
Resultado esperado	Se espera que la aplicación notifique visualmente la creación exitosa del nuevo voluntario. Se espera que la información de perfil del voluntario quede registrada en el sistema. Se espera que la información visualizada en la vista de Perfil sea la misma que la información ingresada.
Automatizado	No.
Resultado efectivo	Aceptado.

Elemento plantilla	Descripción
Responsable	Desarrollador
Fecha de creación	02/07/2016

Tabla 8-24: Caso de prueba para el manejo del perfil del coordinador

Elemento plantilla	Descripción
Identificador	Caso de prueba-07
Casos de uso asociados	CU-01, CU-02
Módulo/funcionalidad	Perfil Coordinador.
Descripción	El caso de prueba consiste en crear y visualizar la información de perfil de un coordinador.
Pre-condiciones	Sin pre-condiciones.
Secuencia de pasos	1.- Se inicia la aplicación. 2.- Se dirige a la vista de registro. 3.- Se ingresa la información solicitada por el sistema para la creación del coordinador. 3.- Se confirma la creación del coordinador. 4.- Se verifica que el coordinador fue creado.
Resultado esperado	Se espera que la aplicación notifique visualmente la creación exitosa del nuevo coordinador. Se espera que la información de perfil del coordinador quede registrada en el sistema. Se espera que la información visualizada en la vista de Perfil sea la misma que la información ingresada.
Automatizado	No.
Resultado efectivo	Aceptado.
Responsable	Desarrollador
Fecha de creación	02/07/2016

8.4 Resultados

Tabla 8-25: Plan de pruebas prototipo Voluntario

N° Requisito	Descripción del requerimiento	Entrada o acción	Resultado esperado del sistema	Confirmación
RFV-01	Acceso Usuario	El sistema permite al usuario crear una nueva cuenta de usuario.	Se muestra una vista con un formulario para que el usuario ingrese su información.	OK
RFV-02	Datos Usuario	El sistema permite al usuario ver la información personal ingresada al registrarse a la base de datos.	El sistema posee la vista Perfil donde el usuario puede revisar los datos personales ingresados.	OK
RFV-03	Acceso Usuario	El sistema permite el reingreso de un usuario al sistema.	El sistema posee la vista Iniciar sesión que permite el inicio de sesión y reingreso de un usuario existente al sistema.	OK
RFV-04	Sistema	El sistema permite al usuario explorar las vistas.	El sistema posee un menú lateral que permite explorar las vistas.	OK
RFV-05	Sistema	El sistema permite a los usuarios ver y explorar un mapa.	El sistema posee en la vista Inicio una interfaz que carga un mapa interactivo.	OK
RFV-06	Sistema	El sistema permite a los usuarios buscar su posición actual en el mapa.	El sistema posee un botón que calcula la posición actual del dispositivo y lo muestra en el mapa.	OK
RFV-07	Sistema	El sistema permite a cada usuario especificar un punto en el mapa como ubicación de referencia para el momento en que postule a una actividad.	Al momento de registrarse, se le presenta al usuario un mapa donde puede escoger un punto de referencia.	OK

N° Requisito	Descripción del requerimiento	Entrada o acción	Resultado esperado del sistema	Confirmación
RFV-08	Sistema	El sistema permite a los usuarios visualizar los puntos de trabajo disponibles para asistir, distribuidos geográficamente.	El sistema muestra en la vista Inicio los marcadores que representan los puntos de trabajo en el mapa.	OK
RFV-09	Sistema	El sistema permite a los usuarios ver la información respectiva a cada punto de trabajo mostrado en el mapa.	El sistema muestra una pestaña con la información del marcador al ser presionado.	OK
RFV-10	Sistema	El sistema permite a cada usuario registrar su postulación para el punto de trabajo para el que desea trabajar.	El sistema muestra un formulario cuando el usuario decide postular a una actividad al presionar el marcador correspondiente.	OK
RFV-11	Sistema	El sistema permite a cada usuario especificar el tiempo en que desea asistir al punto de trabajo.	El usuario debe especificar fechas de llegada y salida de la actividad, y su postulación será registrada.	OK
RFV-12	Sistema	El sistema permite a cada usuario determinar si fue o no escogido para trabajar en la actividad en la que postuló.	El sistema cambia su comportamiento de interfaz cuando el usuario posee un llamado a trabajar.	OK
RFV-13	Sistema	El sistema permite al usuario revisar la ubicación del punto para el que fue llamado.	El sistema provee un botón que muestra en el mapa la posición del punto de trabajo.	OK

N° Requisito	Descripción del requerimiento	Entrada o acción	Resultado esperado del sistema	Confirmación
RFV-14	Sistema	El sistema permite a cada usuario calcular la ruta a tomar desde la ubicación especificada al registrarse en el sistema, hasta el punto de trabajo asignado.	El sistema provee un botón que calcula la ruta y la muestra en el mapa.	OK
RFV-15	Datos Usuario	El sistema permite a cada usuario revisar su puntaje de <i>ranking</i> dentro del sistema.	El sistema muestra en la vista Perfil el nivel de ranking y una representación gráfica de éste.	OK
RFV-16	Datos Usuario	El sistema permite a cada usuario modificar el equipo para el que trabaja.	El sistema provee en la vista Grupo la opción de modifica el grupo o equipo de trabajo.	OK

Tabla 8-26: Plan de pruebas prototipo Coordinador

N° Requisito	Descripción del requerimiento	Entrada o acción	Resultado esperado del sistema	Confirmación
RFC-01	Acceso Coordinador	El sistema permite el ingreso de un usuario coordinador.	El sistema provee dos prototipos distintos con distintas funcionalidades para usuarios y coordinadores	OK
RFC-02	Acceso Coordinador	El sistema permite al coordinador pueda crear una nueva cuenta de coordinador.	Se muestra una vista con un formulario para que el coordinador ingrese su información.	OK
RFC-03	Datos Coordinador	El sistema permite al coordinador ver la información personal ingresada al registrarse a la base de datos.	El sistema posee la vista Perfil donde el coordinador puede revisar los datos personales ingresados.	OK

N° Requisito	Descripción del requerimiento	Entrada o acción	Resultado esperado del sistema	Confirmación
RFC-04	Acceso Coordinador	El sistema permite el reingreso del coordinador al sistema.	El sistema posee la vista Iniciar sesión que permite el inicio de sesión y reingreso del coordinador al sistema.	OK
RFC-05	Sistema	El sistema permite al coordinador explorar las vistas.	El sistema posee un menú lateral que permite explorar las vistas.	OK
RFC-06	Sistema	El sistema permite al coordinador ver y explorar un mapa.	El sistema posee en la vista Inicio una interfaz que carga un mapa interactivo.	OK
RFC-07	Sistema	El sistema permite al coordinador buscar su posición actual en el mapa.	El sistema posee un botón que calcula la posición actual del dispositivo y lo muestra en el mapa.	OK
RFC-08	Sistema	El sistema permite al coordinador crear nuevos marcadores en el mapa para representar los puntos de trabajo.	El sistema provee un formulario para ingresar la información relativa al marcador creado.	OK
RFC-09	Sistema	El sistema permite a los usuarios visualizar los puntos de trabajo disponibles para asistir, distribuidos geográficamente.	El sistema muestra en la vista Inicio los marcadores que representan los puntos de trabajo en el mapa.	OK
RFC-10	Sistema	El sistema permite al coordinador eliminar marcadores previamente creados.	El sistema muestra una pestaña con la información del marcador al ser presionado.	OK
RFC-11	Sistema	El sistema permite al coordinador ver la información respectiva a cada punto de trabajo mostrado en el mapa.	El sistema muestra una pestaña con la información del marcador al ser presionado.	OK

N° Requisito	Descripción del requerimiento	Entrada o acción	Resultado esperado del sistema	Confirmación
RFC-12	Sistema	El sistema permite al coordinador revisar la cantidad de postulantes en cada punto de trabajo.	El sistema muestra una vista personalizada por cada marcador, que muestra información relativa a éste.	OK
RFC-13	Sistema	El sistema permite al coordinador escoger un punto y correr el algoritmo para determinar los voluntarios que asistirán a éste.	El sistema incorpora un algoritmo de servidor que realiza los cálculos.	OK
RFC-14	Sistema	El sistema permite al coordinador realizar un <i>feedback</i> del trabajo realizado.	El sistema muestra un listado de los voluntarios participantes con <i>checkboxes</i> al finalizar una tarea.	OK
RFC-15	Datos Usuario	El sistema permite al coordinador actualizar el puntaje de <i>ranking</i> de cada usuario participante en cada punto.	El sistema recibe el <i>feedback</i> realizado por el coordinador y actualiza el <i>ranking</i> de los voluntarios que éste escogió.	OK
RFC-16	Sistema	El sistema permite que solo el coordinador creador de un punto pueda iniciar la actividad de ese punto.	El sistema reconoce quien es el coordinador que intenta acceder al punto, por lo que solo el creador del punto puede acceder o eliminar el punto.	Ok
RFC-17	Sistema	El sistema permite al coordinador crear y eliminar grupos y equipos de trabajo del sistema.	El sistema provee un formulario para ingresar la información relativa al grupo creado.	Ok