

Proyecto Fin de Carrera

Aplicación móvil para alertas de emergencia CAP



Universidad Carlos III de Madrid
Departamento de Informática
Junio 2011

Omar Siddiqi Agudo

Tutor: Pablo Acuña

Título: Aplicación móvil para alertas de emergencia CAP
Autor: Omar Siddiqi Agudo
Director: Pablo Acuña

EL TRIBUNAL

Presidente: David Díez Cebollero

Vocal: Jorge Blasco Alis

Secretario: Teresa Onorati

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 5 de Julio de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de MATRÍCULA DE HONOR.

VOCAL

PRESIDENTE

SECRETARIO

Agradecimientos

Me gustaría agradecer en primer lugar a mi tutor del proyecto Pablo Acuña por su apoyo y gran esfuerzo a lo largo del proyecto. He aprendido mucho y el proyecto no sería lo que es sin sus aportaciones.

A mi madre Ana por su apoyo incondicional y por dejarme siempre seguir mi pasión y darme todo lo necesario para hacerlo.

A mi novia Laura por su apoyo, por creer en mí siempre y por animarme cuando lo necesitaba.

A mi amigo Jorge por todos esos años acompañándome en la Universidad.

A mi amigo Oriol por su interés y ayuda.

A mi amigo y compañero Javier por muchas prácticas y clases compartidas, me has enseñado mucho.

Y finalmente a todos mis amigos más cercanos, compañeros y familia por estar siempre ahí y animarme.

Gracias a todos.

Resumen

Este proyecto plantea un problema en el ámbito de la difusión de alertas de emergencia, concretamente la falta de soluciones completas de entrega final de dichas alertas. Se plantean cuatro desventajas en la forma actual de recibir alertas de emergencia:

- **Fragmentación de fuentes:** un usuario interesado en información de varias fuentes debe acceder a cada una de manera independiente. No existen soluciones completamente integradas de agregación.
- **Presentación heterogénea:** cada organización presenta la información de una alerta de una manera diferente y la experiencia no es homogénea para el usuario.
- **Falta de personalización:** las alertas presentadas por cada organización son estáticas y fijas de tal manera que un usuario no puede filtrar, organizar o visualizar las alertas de la manera más conveniente para él.
- **Comunicación unidireccional:** con el sistema actual el flujo de información es unidireccional, de las organizaciones al usuario. La solución propuesta no elimina esta desventaja aunque provee una base para extender la funcionalidad en este sentido en el futuro.

Se analiza el estado del arte de la difusión de alertas de emergencia desde tres dimensiones diferentes: el estándar CAP para el intercambio de alertas de emergencia, aplicaciones existentes para la recepción de alertas y las tecnologías móviles avanzadas. En base al estudio se propone una solución que se basa en la creación de una aplicación móvil sobre una plataforma de última generación: iOS.

Se documenta el proceso de creación de un prototipo como prueba de concepto de la solución, incluyendo un estudio previo de la metodología a seguir durante el desarrollo. Se incluyen los tres apartados fundamentales de un proyecto de ingeniería del software: Análisis y Diseño, Implementación y Evaluación. Se presenta además un presupuesto orientativo que puede servir como guía para plantear la viabilidad económica de un producto similar.

Al final del proyecto se presenta la solución desarrollada, contrastándola con los objetivos planteados y se detallan posibles extensiones futuras. La contribución final del proyecto está formada tanto por el estudio y análisis realizados como por una aplicación funcional que se incluye en el mercado público a disposición de cualquier usuario.

Palabras clave: alertas, gestión de emergencias, CAP, aplicaciones móviles, iOS, iPhone.

Abstract

This project presents an existing problem in the realm of emergency alert dissemination, specifically the lack of complete solutions for delivery of said alerts. There are four disadvantages with the current method for receiving emergency alerts:

- **Fragmented sources:** a user who is interested in information from various sources must access each one individually. There are no completely integrated aggregation solutions.
- **Differing presentation:** each organization or source delivers the information contained in an alert in a different manner and the experience is not consistent for the user.
- **Lack of personalization:** the alerts that are delivered are generally static and fixed in such a way that a user cannot filter, order or visualize them in the most convenient manner.
- **One-way communication:** with the current method the flow of information is one-way, from organizations to users. The proposed solution does not eliminate this disadvantage although it provides the framework to extend functionality in this direction in the future.

A state of the art analysis was performed on emergency alerts dissemination based on three key dimensions: the CAP standard for alert exchange, existing applications and advanced mobile technologies. Based on this study we propose a solution centred on creating a mobile application for a current mobile platform: iOS.

The process of creating the prototype as a proof of concept solution is completely documented, including a study of the chosen development methodology. The three fundamental aspects of any software engineering project are included: Analysis & Design, Implementation and Evaluation. Furthermore, there is a budget proposal, which serves as a guide to the economic viability of the project.

At the end of the document the developed solution is presented and measured against the presented objectives, along with possible future upgrades. The contribution of the project is made up of both the analysis and the functional application that is included in the public domain available to any user.

Keywords: alerts, emergency management, CAP, mobile applications, iOS, iPhone

Índice General

Agradecimientos	5
Resumen	7
Abstract	8
Índice General	9
Índice de Figuras	12
Índice de Tablas	15
1 Introducción y objetivos	16
1.1 Definición del problema	16
1.2 Objetivos	19
1.3 Motivaciones	19
1.4 Descripción del documento	20
2 Estado del arte	21
2.1 Alertas de emergencia	21
2.1.1 Estándar CAP	21
2.1.1.1 Especificación CAP 1.2	22
2.1.2 Organizaciones y fuentes	25
2.1.3 Otras iniciativas	25
2.1.3.1 Alert Hub de Google	25
2.1.3.2 FEMA: adopción de CAP	26
2.2 Aplicaciones existentes	27
2.2.1 Aplicaciones de escritorio	27
2.2.1.1 PDC Active Hazards Widget	27
2.2.2 Aplicaciones web	28
2.2.2.1 RSOE-EDIS	28
2.2.2.2 PDC Natural Hazards and Vulnerabilities Atlas	31
2.2.3 Aplicaciones Móviles	32
2.2.3.1 Disaster Alert	32
2.2.3.2 Quake Alert	32
2.2.4 Tabla comparativa de aplicaciones existentes	34
2.3 Tecnologías móviles	35
2.3.1 Introducción	35
2.3.2 Plataforma iOS	36

2.3.3 Plataforma Android	37
2.3.4 Otras plataformas	38
2.3.5 Conclusiones	38
3 Desarrollo de la solución	39
<hr/>	
3.1 Análisis y Diseño	40
3.1.1 Análisis inicial	40
3.1.2 Diseño mediante prototipado rápido	41
3.2 Implementación del prototipo	45
3.2.1 Metodología de desarrollo	45
3.2.1.1 Metodologías ágiles	45
3.2.1.2 SCRUM	49
3.2.1.3 Metodología definitiva	50
3.2.2 Implementación	52
3.2.2.1 Arquitectura iOS: Modelo-Vista-Controlador	52
3.2.2.2 Interfaces y Vistas	56
3.3 Evaluación	70
3.3.1 Evaluación de requisitos	70
3.3.2 Evaluación del diseño con un experto	70
4 Presupuesto	77
<hr/>	
4.1 Presupuesto resumido	77
4.2 Presupuesto detallado	78
5 Solución y Conclusiones	79
<hr/>	
5.1 Solución final	79
5.1.1 Presentación de la aplicación	79
5.1.1.1 Manual de uso	80
5.1.2 Cumplimiento de objetivos del proyecto	85
5.1.3 Mejoras futuras	85
5.2 Contribución	86
5.2.1 World Alerts App	86
5.3 Conclusiones personales	87
6 Referencias	88
<hr/>	
6.1 Referencias y enlaces	88
Anexo A: Estructura de mensaje CAP	89
<hr/>	
Modelo de objeto de documento	89

Diccionario de datos	90
Anexo B: Documentación del Desarrollo	104
Historias de Usuario (Product Backlog)	104
Priorización de Historias	105
Tareas individuales (Sprint Backlog Unificado)	106
Definición detallada de Historias de Usuario	108

Índice de Figuras

Figura 1: Página de USGS sobre un terremoto (www.usgs.gov).....	17
Figura 2: Fuente RSS de USGS visualizado con un navegador (www.usgs.gov).....	18
Figura 3: Estructura de una mensaje CAP (oasis-open.org).....	23
Figura 4: una alerta en la página principal de Google (google.org)	26
Figura 5: PDC Active Hazards Widget (pdg.org/widget).....	28
Figura 6: Mapa con alertas del sistema EDIS (hisz.rseo.hu).....	29
Figura 7: Listado de alertas de EDIS (hisz.rseo.hu)	29
Figura 8: Listado de alertas de terremoto de EDIS (hisz.rseo.hu).....	29
Figura 9: Listado de alertas de Tsunami de EDIS (hisz.rseo.hu).....	30
Figura 10: Listado de alertas de volcanes de EDIS (hisz.rseo.hu)	30
Figura 11: PDC Natural Hazards & Vulnerabilities Atlas (pdc.org/atlas)	31
Figura 12: Pantallas de la aplicación Disaster Alert del PDC (itunes.apple.com)	32
Figura 13: Listado y mapa de terremotos de Quake Alert (itunes.apple.com).....	33
Figura 14: Pantallas de configuración y aviso de Quake Alert (itunes.apple.com) ...	33
Figura 15: Adopción de smartphones en EE.UU. (blog.nielsen.com).....	35
Figura 16: Cuota de mercado de plataformas móviles en EE.UU. (comScore, 2011).36	
Figura 17: Escala de colores de alerta de seguridad nacional en EE.UU.....	42
Figura 18: Maqueta de listado de alertas y selección de categorías.....	43
Figura 19: Maqueta de vista resumen y vista detalle de una alerta.....	44
Figura 20: Maqueta de vista en mapa de una alerta.....	44
Figura 21: Arquitectura MVC.....	52
Figura 22: Diagrama de clases principales de modelo	53
Figura 23: Diagrama de clases principales de controlador.....	54
Figura 24: Listado de alertas vacío	56
Figura 25: Barra superior inicial del listado de alertas	56
Figura 26: Listado cargando alertas	57
Figura 27: Vista inicial de cargado de alertas	57
Figura 28: Celda de alerta individual en el listado	58
Figura 29: Vista de categorías.....	59
Figura 30: Vista resumen de una alerta.....	60
Figura 31: Icono de mapa disponible y no disponible	60

Figura 32: Icono de recursos disponibles y no disponibles.....	61
Figura 33: Vista de la página web contenida en una alerta	61
Figura 34: Vista de envío de una alerta por correo electrónico	62
Figura 35: Vista en mapa y vista de recursos de una alerta	62
Figura 36: Vistas de todos los campos de una alerta I.....	63
Figura 37: Vistas de todos los campos de una alerta II.....	63
Figura 38: Vista inicial de resumen de alerta	64
Figura 40: Vista en mapa general.....	66
Figura 41: Tres tipos de mapa	67
Figura 42: Marcador de una alerta en la vista en mapa.....	67
Figura 43: Ajustes de selección de fuentes	67
Figura 44: Selección de fuentes individuales de NWS	68
Figura 45: Ajustes de filtrado y ocultación de alertas caducadas	68
Figura 46: Ajustes de filtrado de Urgencia, Severidad y Certeza	68
Figura 47: Ajuste de límite de descarga.....	69
Figura 48: Propuesta de colores para severidad y urgencia	73
Figura 49: Opciones para mostrar color en el listado de alertas	74
Figura 50: Opciones para mostrar color en la vista resumen de alerta	74
Figura 51: Opciones para mostrar valor de certeza.....	75
Figura 52: Propuesta de textura como muestra de certeza	75
Figura 53: Propuesta de gráfico de niveles de Urgencia, Severidad y Certeza.....	76
Figura 54: Barra inferior de la aplicación	80
Figura 55: Selección de fuentes en la aplicación.....	80
Figura 56: Acción de descarga de alertas y parada	80
Figura 57: Descripción de alerta en listado.....	80
Figura 58: Descripción de vista resumen de alerta.....	81
Figura 59: Vista de compartir alerta por correo	82
Figura 60: Botón de ordenación de listado de alertas.....	82
Figura 61: Selección de categorías de alertas.....	82
Figura 62: Selección de opciones de filtrado de alertas.....	83
Figura 63: Vista en mapa de alertas.....	83
Figura 64: Selección de una alerta en mapa.....	83
Figura 65: Sección de ayuda de la aplicación	84
Figura 66: Opción de limite de descarga de la aplicación.....	84

Figura 67: Icono de World Alerts86

Figura 68: Zona de publicidad en la vista resumen de alerta87

Índice de Tablas

Tabla 1: Organizaciones especializadas en emergencias.....	25
Tabla 2: Tabla comparativa de aplicaciones existentes	34
Tabla 3: Comparativa de metodologías ágiles	48
Tabla 4: Clases más importantes de la aplicación.....	55
Tabla 5: Colores de nivel de importancia.....	58
Tabla 6: Presupuesto resumido	77
Tabla 7: Presupuesto detallado	78

1 Introducción y objetivos

Muchos de los problemas que se presentan en la vida cotidiana tienen como solución la obtención, manipulación, comunicación y presentación de información. Es ésta la esencia de las TIC (Tecnologías de la Información y Comunicación) y de la Ingeniería Informática: la resolución de los problemas relacionados con la información y su tratamiento, considerando siempre las restricciones impuestas por el contexto. Mediante la aplicación de la tecnología existente y las técnicas acumuladas tras años de aprendizaje, el Informático crea soluciones robustas, efectivas y eficientes que facilitan y enriquecen la vida de las personas. Para un estudiante de la Informática son evidentes las innumerables fuentes de información que existen en la sociedad y sus posibles usos para resolver problemas grandes y pequeños.

El problema que se estudia en este proyecto es la deficiencia de un eslabón crítico en la cadena de información en el ámbito de las emergencias. Existen muchas fuentes fiables que recogen información sobre emergencias de manera constante y rigurosa, pero cuando llega el momento de hacer llegar esta información a manos de los que más podrían beneficiarse, los mecanismos existentes no proporcionan una solución completamente integrada y configurable. Una de las causas de este problema es la naturaleza distribuida de la información, provocada por la existencia de múltiples organizaciones (fuentes) separadas por especialidades o ámbitos geográficos. Otra causa importante es la falta de contextualización de la información en base a las necesidades de las personas. Cuando se trata de una información tan crítica, cada instante de tiempo es importante y no se puede desaprovechar.

En este proyecto se estudian los mecanismos actuales de difusión de información sobre emergencias en forma de alertas, se analizan sus deficiencias y se propone una mejor solución basada en tecnologías de última generación. Se pretende contribuir una plantilla de solución que puede desplegarse de manera inmediata o servir como guía para la creación de otras soluciones similares. Como consecuencia de esta solución se obtiene un estudio sobre los aspectos más importantes a contemplar en un escenario de difusión de alertas contextualizadas, considerando tanto las necesidades del usuario final como los recursos disponibles (tanto tecnológicos como de información).

1.1 Definición del problema

Nunca ha sido más evidente que en la primera década del siglo XXI el gran impacto de los desastres en la humanidad. Desde los ataques terroristas del 11 de Septiembre, hasta los múltiples terremotos pasando por pandemias mundiales, es innegable que las situaciones de emergencia afectan a todos por igual y cambian el curso de la historia, causando pérdidas humanas y económicas de gran escala.

Hasta relativamente poco la única respuesta que existía ante tales catástrofes era siempre a posteriori. Con la llegada de la era de la información esto empezó a cambiar. Las tecnologías de previsión o detección y las redes de comunicación, cada vez más rápidas y extensas, permiten reducir drásticamente el tiempo de respuesta ante situaciones de emergencia.

Organizaciones como "*United States Geological Survey*" (usgs.gov) y "*National Oceanographic and Atmospheric Administration*" (noaa.gov) existen como fuentes de información fiable, actual y especializada en cuanto a alertas de actividad sísmica o meteorológica respectivamente. De la misma manera existen otras organizaciones en diversos ámbitos que toman un rol similar como fuente de información:

- **CDC:** *Centers For Disease Control & Prevention*, especializados en enfermedades y salud pública (EE.UU.) (cdc.gov)

- **FEMA:** *Federal Emergency Management Agency*, agencia gubernamental encargada en coordinar la respuesta ante desastres naturales (EE.UU.) (fema.gov)
- **CSEM:** *Centre Sismologique Euro-Méditerranéen*, centro europeo-mediterráneo de sismología (UE) (emsc-csem.org)
- **ECMWF:** *European Centre for Medium-Range Weather Forecasts*, centro europeo especializado en predicciones meteorológicas a medio plazo (EU) (ecmwf.int)
- **JMA:** *Japan Meteorological Agency*, agencia encargada de la predicción y mitigación de desastres naturales (Japón) (jma.go.jp)

Es evidente por tanto que la obtención y el análisis de la información relativa a desastres naturales y emergencias es una tarea de la cual se encargan múltiples organizaciones separadas tanto por especialidad como por zonas geográficas. La información existe, el siguiente paso es hacerla llegar a las manos de los afectados.

Para obtener la información que proveen estas organizaciones el medio más habitual es el acceso directo a través de Internet, sea mediante la página oficial de las organizaciones o mediante protocolos de sindicación y difusión como RSS¹ (*Really Simple Syndication*). Un ejemplo de ambos se puede observar a continuación en **Figura 1** y **Figura 2**.

The screenshot shows the USGS Earthquake Hazards Program website. The main heading is "Magnitude 4.9 - SOUTHERN SUMATRA, INDONESIA" with the date and time "2011 April 05 13:57:24 UTC". Below this, there are tabs for "Details", "Summary", "Maps", and "Scientific & Technical". The "Details" tab is selected, showing "Earthquake Details". A note states "This event has been reviewed by a seismologist." Below this is a table with the following data:

Magnitude	4.9
Date-Time	Tuesday, April 05, 2011 at 13:57:24 UTC Tuesday, April 05, 2011 at 08:57:24 PM at epicenter Time of Earthquake in other Time Zones
Location	4.572°S, 101.190°E
Depth	5.3 km (3.3 miles) (poorly constrained)
Region	SOUTHERN SUMATRA, INDONESIA
Distances	146 km (90 miles) WSW of Bengkulu, Sumatra, Indonesia 232 km (144 miles) SW of Lubuklinggau, Sumatra, Indonesia 361 km (224 miles) WSW of Perabumulih, Sumatra, Indonesia 653 km (405 miles) WNW of JAKARTA, Java, Indonesia
Location Uncertainty	horizontal +/- 19.9 km (12.4 miles); depth +/- 6.3 km (3.9 miles)
Parameters	NST= 57, Nph= 60, Dmin=197 km, Rmss=0.97 sec, Gp=108°, M-type=body wave magnitude (Mb), Version=9
Source	USGS NEIC (WDCS-D)
Event ID	usc0002jak

Below the table, there is a link "Did you feel it?" with a description: "Report shaking and damage at your location. You can also view a map displaying accumulated data from your report and others." At the bottom, there is a "Preliminary Earthquake Report" link and the text "U.S. Geological Survey, National Earthquake Information Center, World Data Center for Seismology - Denver".

Figura 1: Página de USGS sobre un terremoto (www.usgs.gov)

¹ RSS: Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos.

Feed for USGS M 5+ Earthquakes
 Subscribe to this feed using:
 Always use this reader to subscribe to feeds.

Feed preview

M 6.0, Kepulauan Talaud, Indonesia

 Tuesday, April 5, 2011 11:14:15 UTC
 Tuesday, April 5, 2011 07:14:15 PM at epicenter
Depth: 17.60 km (10.94 mi)

M 5.0, Solomon Islands

 Tuesday, April 5, 2011 09:57:15 UTC
 Tuesday, April 5, 2011 08:57:15 PM at epicenter
Depth: 28.00 km (17.40 mi)

M 5.4, Kepulauan Talaud, Indonesia

 Tuesday, April 5, 2011 07:46:18 UTC
 Tuesday, April 5, 2011 03:46:18 PM at epicenter
Depth: 12.40 km (7.71 mi)

M 5.2, Java, Indonesia

 Tuesday, April 5, 2011 04:34:41 UTC
 Tuesday, April 5, 2011 11:34:41 AM at epicenter
Depth: 42.70 km (26.53 mi)

Figura 2: Fuente RSS de USGS visualizado con un navegador (www.usgs.gov)

Aunque es un comienzo, esta manera de difundir alertas de emergencia sufre las siguientes desventajas:

1. **Fragmentación de fuentes:** un usuario interesado en información de varias fuentes debe acceder a cada una de manera independiente. Como aproximación a la agregación en un lugar de las alertas se puede utilizar un lector de las fuentes de sindicación RSS, pero la presentación queda muy simplificada y se reduce la usabilidad de la información.
2. **Presentación heterogénea:** cada organización presenta la información de una alerta de una manera diferente y la experiencia no es homogénea para el usuario. Cuando la interpretación rápida de la alerta es un aspecto clave, la existencia de múltiples interfaces es muy perjudicial.
3. **Falta de personalización:** las alertas presentadas por cada organización son estáticas y fijas de tal manera que un usuario no puede filtrar, organizar o visualizar las alertas de la manera más conveniente para él.
4. **Comunicación unidireccional:** con el sistema actual el flujo de información es unidireccional, de las organizaciones al usuario. No se consideran las posibles aportaciones de los propios afectados por una emergencia como: estimación de número de afectados por localización GPS (por las organizaciones en base a las respuestas), confirmación o negación de una alerta por observadores presentes en el lugar afectado, aportación de medios como grabaciones de vídeo, etc.

En este proyecto se estudian en detalle los aspectos, tanto positivos como negativos, de la tecnología actual de difusión de alertas. Se propone una solución que reduce o elimina los problemas existentes, y se detalla el proceso seguido para servir como base para el desarrollo continuado de la solución.

1.2 Objetivos

Una vez expuesto el problema pasamos a definir claramente los objetivos del proyecto, contra los que se evaluará la solución final obtenida. Los objetivos son:

1. Estudiar el estado actual de la tecnología de difusión de alertas de emergencia. Esto incluye aplicaciones e iniciativas existentes, además del uso de estándares para permitir interoperabilidad de sistemas.
2. Diseñar una plantilla de solución que reduzca o elimine las deficiencias existentes en la difusión de alertas. Este diseño contemplará tanto la naturaleza de la información tratada (alertas) como aspectos de diseño de interfaces y usabilidad.
3. Crear un prototipo como prueba de concepto de la solución, proporcionando una muestra no sólo del producto final sino también del proceso seguido hasta obtenerlo. Este prototipo puede servir de plantilla para futuras soluciones ampliadas o incluso podría llegar a ser desplegado en el mercado de aplicaciones para la plataforma escogida.

En definitiva, se estudiará el elemento básico de información del problema (la alerta de emergencia), cómo diseñar una solución que se adapte lo mejor posible a su difusión, y se proporcionará un prototipo como prueba de concepto.

1.3 Motivaciones

La elección de este proyecto final de carrera estuvo motivada por varios factores tanto personales como profesionales. Como trabajo representativo de una carrera de Ingeniería Informática engloba muchas de las dimensiones de la misma: Ingeniería del Software, Programación, Diseño, Documentación, Evaluación, etc. Se ha desarrollado el trabajo teniendo en cuenta la aplicación del conocimiento variado que se ha acumulado a lo largo de los cinco cursos académicos.

A nivel personal se ha elegido este trabajo ya que conlleva el aprendizaje de una nueva plataforma de desarrollo: iOS¹. Es una plataforma en auge en estos momentos, y contar con experiencia de desarrollo es un punto importante a incluir en el currículum de un Ingeniero recién titulado. Además, al contemplar la posibilidad de lanzar al mercado la aplicación que surgiera como fruto del trabajo se abre paso a la experiencia más completa de llevar un producto desde concepto hasta la realidad. Desde el marketing y la difusión de la aplicación hasta el mantenimiento y la expansión, tiene el potencial de convertirse en una experiencia muy enriquecedora y de gran valor formativo.

¹ iOS: el sistema operativo de dispositivos post-PC desarrollados por Apple Inc., utilizado en iPhone, iPad y AppleTV.

1.4 Descripción del documento

Después de realizar esta introducción al problema y presentación del proyecto, a continuación se detallan los diferentes apartados presentes en la memoria y un resumen de su contenido.

- **Estado del arte:** en este apartado se presenta un estudio previo del estado de varios aspectos relacionados con el proyecto y su contexto. Se comienza el estudio con una vista detallada de las alertas de emergencia, incluyendo una descripción del estándar CAP, una muestra de organizaciones relacionadas con las emergencias y un resumen de otras iniciativas relacionadas y complementarias al proyecto. A continuación se incluye un análisis de aplicaciones similares existentes con una comparativa de ventajas y desventajas, y para finalizar se presenta un estudio de las tecnologías móviles como plataforma y su viabilidad para desarrollar aplicaciones útiles y exitosas.
- **Desarrollo de la solución:** este apartado incluye todo aquello relacionado con la creación de la prueba de concepto propuesta como solución al problema. Contiene el análisis y diseño realizado, una descripción del proceso de implementación (en cuanto a metodología y programación) y finalmente los métodos de evaluación utilizados y sus resultados.
- **Presupuesto:** se presenta en este apartado un presupuesto basado en el proceso de desarrollo de la prueba de concepto. Este presupuesto se ajusta lo máximo a la realidad para confirmar de nuevo la viabilidad económica del proyecto en caso de tomarse como solución en el futuro.
- **Solución y conclusiones:** aquí se presenta la solución final propuesta en base al problema planteado en la introducción, detallando sus aspectos más importantes y contrastándolos con los objetivos definidos. Se presenta la contribución del proyecto y se cierra la memoria con unas conclusiones personales.

2 Estado del arte

En los siguientes apartados se presenta un estudio del estado actual del campo de alertas de emergencia, con un énfasis en los estándares existentes y la aplicación de tecnologías móviles.

2.1 Alertas de emergencia

La RAE define una emergencia como una "*Situación de peligro o desastre que requiere una acción inmediata*". Es en esta definición en la que se basa el concepto de alerta de emergencia, "*situación declarada con el fin de tomar precauciones específicas debido a la probable y cercana ocurrencia de un suceso o accidente*". Esta definición está tomada del Real Decreto 393/2007, de 23 de marzo, por el que se aprobó la Norma Básica de Autoprotección de los centros, establecimientos y dependencias dedicados a actividades que puedan dar origen a situaciones de emergencia¹.

2.1.1 Estándar CAP

El estándar CAP (Common Alerting Protocol) es un formato sencillo pero general para el intercambio de alertas de todo tipo de emergencias y avisos públicos, sobre todo tipo de redes de comunicación. CAP permite difundir un mensaje de manera sistemática a través de una multitud de sistemas de alerta, aumentando su efectividad a la vez que simplifica la tarea de alertar. CAP proporciona una plantilla para mensajes de alerta efectivos basada en las mejores prácticas identificadas en investigaciones académicas y experiencia en el mundo real (OASIS, 2011).

El estándar lo impulsa principalmente la organización OASIS (Organization for the Advancement of Structured Information Standards), que es un consorcio sin ánimo de lucro que potencia el desarrollo y la adopción de estándares abiertos para la sociedad informática global. La organización promueve el consenso en el sector y produce estándares globales para seguridad, *cloud computing*, SOA, servicios web, gestión de emergencias y otras áreas. Los estándares de OASIS ofrecen la posibilidad de reducir costes, estimular la innovación y proteger el derecho a la elección libre de la tecnología. Los miembros del consorcio representan los líderes del mercado, tanto público como privado, con más de 5000 participantes de más de 600 organizaciones distribuidos en 100 países

El estándar CAP se encuentra ya en su versión 1.2 (segunda versión desde 1.0), actualizado por última vez en Julio del 2010. Cada vez es mayor el número de organizaciones que han optado por adoptar el estándar, siendo una de las más destacadas recientemente la Agencia Federal para el Manejo de Emergencias, o FEMA por sus siglas en inglés (Federal Emergency Management Agency, EE.UU.) para su sistema de alertado público (IPAWS).

Una de las ventajas de adoptar este estándar es la interoperabilidad que proporciona. El hecho de que múltiples sistemas de organizaciones diferentes se basen en un mismo estándar de intercambio de mensajes de alerta facilita la interoperación entre los mismos. Se puede considerar como una especie de integración vertical que favorece al sector completo de notificaciones de alertas, desde las organizaciones que las generan hasta las personas a las que van destinadas.

Debido a la tendencia que lleva el estándar CAP y las ventajas descritas, se considera la mejor opción a seguir en cuanto a la difusión de alertas de emergencia se refiere, y encaja perfectamente con los objetivos y requisitos de este proyecto.

¹ BOE: http://www.boe.es/aeboe/consultas/bases_datos/doc.php?id=BOE-A-2007-6237

2.1.1.1 Especificación CAP 1.2

El estándar CAP proporciona un formato de mensaje digital abierto y no-exclusivo para todo tipo de alertas y notificaciones. No contempla ninguna aplicación o método de telecomunicaciones en concreto. El formato es compatible con tecnologías emergentes como los servicios web, además de algunos formatos existentes como Specific Area Messaging Encoding (SAME), de la NOAA, a la vez que proporciona mejoras que incluyen:

- Enfoque geográfico flexible usando coordenadas de longitud/latitud, formas geométricas y otras representaciones en tres dimensiones
- Mensajes multilingües y multi-audiencia
- Tiempos efectivos y de caducidad en fases o con retardo
- Opciones de actualización o cancelación de mensajes
- Compatibilidad con firmas digitales
- Facilidad para imágenes y audio digitales

Los beneficios clave de CAP incluyen la reducción de costes y complejidad operacional mediante la eliminación de múltiples interfaces de software con los múltiples sistemas involucrados en la detección y difusión de alertas de emergencia. La estructura de un mensaje CAP consta de los siguientes segmentos:

- **<alert>**: proporciona información básica de la alerta: su objetivo, la fuente y su estado, además de un identificador unívoco que permitirá relacionar la alerta con futuras actualizaciones o cancelaciones.
- **<info>**: un segmento de este tipo sirve para ampliar los detalles de la alerta. Los campos más destacados de este segmento son los que describen la urgencia, severidad y certeza de la alerta; además puede contener una descripción detallada e instrucciones sobre cómo actuar. Si se desea proporcionar la información en múltiples idiomas se utilizarían tantos segmentos <info> como sean necesarios.
- **<resource>**: este segmento proporciona un enlace a información complementaria, relacionada con el segmento <info> que lo contiene, típicamente como un recurso digital (imagen, audio, video, etc.).
- **<area>**: un segmento que describe el área geográfica a la que aplica el segmento <info> que lo contiene. Se aceptan descripciones textuales y codificadas pero se prefieren representaciones geoespaciales mediante formas geométricas y medidas de altitud/longitud/latitud.

El siguiente diagrama muestra una representación de la relación de los segmentos descritos con sus respectivos campos. Aquellos campos marcados en **negrita** son obligatorios, y los marcados en *cursiva* tienen valores por defecto especificados. Un asterisco (*) denota la posible multiplicidad de un campo o segmento.

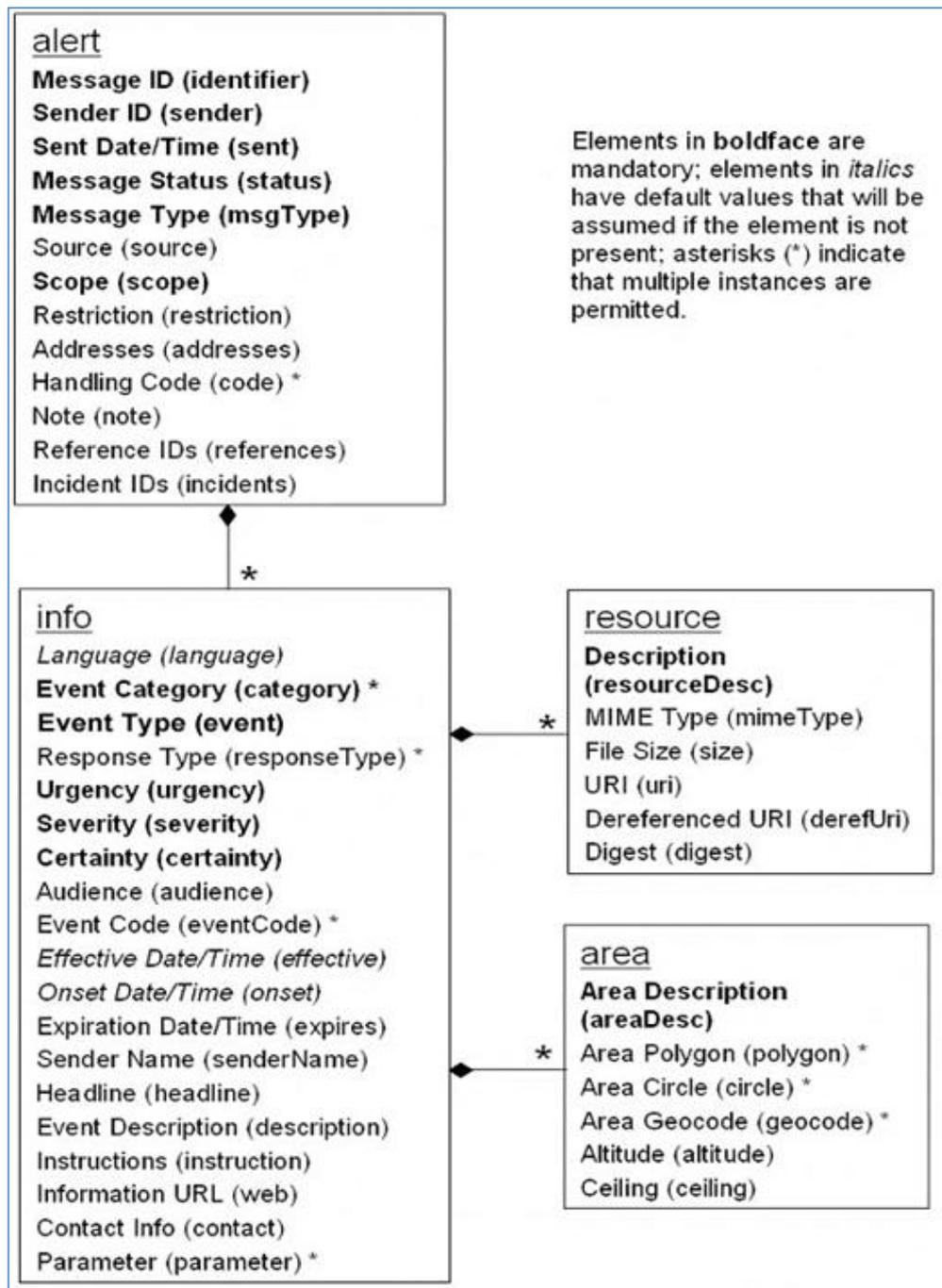


Figura 3: Estructura de una mensaje CAP (oasis-open.org)

Se puede observar claramente la estructura de un mensaje de alerta CAP. Una alerta contiene cero o más elementos de información, que a su vez contiene cero o más recursos o descripciones de área. Los campos marcados en negrita son aquellos obligatorios de rellenar si se incluye un elemento del mensaje. Esto es un aspecto importante a considerar, ya que si se pretenden recibir mensajes de múltiples fuentes es probable que cada una utilice diferentes campos según la información que tenga disponible. La existencia de campos obligatorios asegura una cantidad mínima de información de la cual se dispondrá de cada alerta. Algunos de los campos de mensaje a destacar son:

- **Message ID:** un identificador único del mensaje que permite referenciarlo en el futuro para actualizaciones o cancelaciones.

- **Sender ID:** un identificador único del emisor, típicamente la URL de la organización.
- **Sent Date/Time:** hora y fecha de envío, importantes para conocer la relevancia de la alerta.

Los siguientes campos se corresponden al elemento <info> que es opcional pero deseable incluir:

- **Event Category:** una de las siguientes categorías definidas por el estándar:
 - **Geo:** geofísico (terremoto, alud)
 - **Met:** meteorológico (tormentas, inundaciones)
 - **Safety:** emergencias generales y seguridad pública
 - **Security:** seguridad policial, militar y nacional (disturbios)
 - **Rescue:** rescate y recuperación
 - **Fire:** incendios o fuego
 - **Health:** emergencias médicas y de salud pública (epidemias)
 - **Env:** medio ambiente (contaminación)
 - **Transport:** transporte público y privado (tráfico, accidentes)
 - **Infra:** otras infraestructuras (telecomunicaciones)
 - **CBRNE:** peligro químico, biológico, radiológico, nuclear o explosivo.
 - **Other:** otros eventos
- **Urgency:** la urgencia denota la rapidez con la que se deben tomar acciones frente a la alerta (desde inmediato hasta pasado)
- **Severity:** la severidad indica la extensión del efecto de la alerta (desde extrema hasta mínima)
- **Certainty:** la certeza indica la probabilidad de que ocurra la alerta (desde observado hasta improbable).
- **Headline:** un resumen abreviado de la alerta, el titular de la alerta (la información más crítica).
- **Event Description:** contiene una descripción más detallada de la alerta.
- **Instructions:** instrucciones sobre cómo actuar o responder ante la alerta, útil si no es obvio como reaccionar.

Es importante conocer la estructura que impone el estándar CAP sobre los mensajes de alerta, ya que es ésta la que define tanto el contenido como la asociación de los diferentes campos. El estudio del estándar es el punto de partida para comprender lo que es esencialmente una alerta, cómo la pueden usar las organizaciones que desean informar de emergencias y qué se puede extraer de utilidad para el destinatario final.

Para una descripción detallada del estándar CAP consultar con el Anexo A al final del documento.

2.1.2 Organizaciones y fuentes

Una vez estudiado el estándar CAP que define la estructura de los mensajes de alerta se ha procedido al estudio de su uso en la actualidad. Es importante que un estándar sea adoptado por las organizaciones encargadas de emitir alertas, ya que su mayor adopción facilita la integración de un sistema completo de difusión de alertas y la creación de soluciones como la propuesta en este proyecto.

En la siguiente tabla se recogen algunas de las organizaciones mundiales que se especializan en situaciones de emergencia de alguna clase. Para cada organización se indica: nombre, ámbito geográfico, misión u objetivo, y medios de difusión de alertas.

Organización	Ámbito Geográfico	Misión y objetivo	Medios de difusión
CDC: Center for disease control & prevention	EE.UU.	Prevención, detección y estudio de epidemias y enfermedades	web
FEMA: Federal Emergency Management Agency	EE.UU.	Respuesta ante emergencias federales	web,CAP*
CSEM: Centre Sismologique Euro-Méditerranéen	UE	Estudio de eventos sísmológicos	web
ECMWF: European Centre for Medium-Range Weather Forecasts	UE	Estudio y predicción de la meteorología a nivel Europeo	web
JMA: Japan Meteorological Agency	Japón	Estudio y predicción de eventos de meteorología y geología en Japón y Asia	web
USGS: US Geological Survey	EE.UU. (Mundial)	Estudio de eventos geológicos mundiales, especialmente terremotos	CAP, web
NOAA: National Oceanographic and Atmospheric Administration	EE.UU.	Estudio y aviso de eventos de meteorología.	CAP, web
RSOE-EDIS (Hungría)	Mundial	Alertas de emergencia mundiales de toda categoría	CAP**, web

Tabla 1: Organizaciones especializadas en emergencias

*Previsión de implementar CAP en un sistema futuro

**Fuentes CAP mediante suscripción pagada

2.1.3 Otras iniciativas

2.1.3.1 Alert Hub de Google

Alert Hub es una iniciativa reciente de la rama filantrópica de Google (google.org) cuyo principal objetivo es proporcionar un sistema centralizado para agregar alertas de emergencia de todo el mundo para luego poder difundirlas de manera rápida y efectiva. Google ya tiene experiencia con emergencias, y ha usado su alcance y escala para alertar de situaciones de emergencia a nivel mundial. En la siguiente imagen se puede observar una de estas alertas durante el terremoto de Japón del 2011.



Figura 4: una alerta en la página principal de Google (google.org)

Su objetivo es ir más allá y proporcionar información precisa y relevante de todas las emergencias, lo más rápido posible y en todos los lugares posibles. Para ello requieren fuentes fiables de alertas de emergencia que se comuniquen mediante un estándar. Google propone usar CAP para emitir alertas de emergencia, y que cada organización las agregue en fuentes de difusión (*feeds*). Google ha proporcionado acceso a múltiples librerías y herramientas para facilitar la generación de alertas CAP y fuentes, algo crítico para acelerar la adopción del estándar. El siguiente paso es agregar todas las fuentes en el llamado *alert-hub* creado y gestionado por Google. Se trata de un repositorio centralizado que sirve como punto de agregación de alertas de múltiples fuentes, y que permite a un cliente final interesado en recibir alertas simplificar su proceso de suscripción a un sólo lugar. El repositorio se basa en el protocolo pubsubhubbub¹ para facilitar las suscripciones a las fuentes de alertas.

En resumen Google quiere mejorar el acceso a la información de alertas de emergencia de fuentes fiables, quieren incrementar la adopción global del estándar CAP y quieren mejorar la agregación de alertas y su difusión. Es importante destacar que podría servir en el futuro como una fuente fiable de alertas para la solución propuesta en este proyecto. Esta iniciativa refuerza la solución propuesta y su éxito es claramente beneficioso para el futuro de soluciones similares y el sector en general.

2.1.3.2 FEMA: adopción de CAP

En Septiembre del 2010 la organización FEMA (fema.gov) anunció su intención de adoptar el estándar CAP para su sistema integrado de alertas públicas (IPAWS). El objetivo del sistema es ampliar el método tradicional de alertado de emergencias permitiendo que los oficiales de gestión de emergencias puedan alcanzar el mayor número de personas posibles por medio de la mayor variedad posible de dispositivos y medios (como radio, televisión, móviles, ordenadores y otros). La adopción del estándar facilitará la integración e interoperabilidad de los diferentes sistemas, y simplificará la generación de mensajes bajo un mismo formato. FEMA también pretende crear un sistema de agregación de alertas dentro de IPAWS, permitiendo el acceso mediante servicios web a desarrolladores de todas las plataformas.

Se pueden observar similitudes entre la iniciativa de FEMA y la de Google con su AlertHub, aunque ambos se encuentran en ámbitos diferentes (estatal y privado respectivamente). Aun así, ambas iniciativas impulsan la adopción del estándar CAP para las alertas de emergencia y ambas están de acuerdo en la necesidad de proporcionar medios efectivos de difusión y acceso a las alertas para múltiples

¹ <http://code.google.com/p/pubsubhubbub/>

plataformas. De nuevo esta iniciativa complementa la solución propuesta en este proyecto.

2.2 Aplicaciones existentes

Durante la realización del proyecto se han buscado soluciones existentes al problema propuesto, en el ámbito de las aplicaciones de escritorio, web y móviles.

2.2.1 Aplicaciones de escritorio

2.2.1.1 PDC Active Hazards Widget

La organización PDC (Pacific Disaster Center) tiene como misión proporcionar apoyo a la investigación y al análisis aplicados para mejorar la gestión de situaciones de emergencia y asistencia humanitaria para las comunidades de la zona del Pacífico y el mundo. Tiene su sede en Hawaii, EE.UU., y una de sus actividades principales consiste en la recopilación de información sobre situaciones de emergencia o desastres. Esta información proviene de organizaciones o instituciones especializadas en distintos ámbitos como meteorología o actividad sísmica, y antes de llegar al usuario de la aplicación pasa por las manos del PDC. De alguna manera la propia organización sirve como punto de agregación de las alertas, actuando en gran parte como editor tanto del contenido de las propias alertas como de su formato y presentación. Además es la propia organización la que filtra las alertas que llegarán a manos de los usuarios, alertas que definen como "*active hazards*" o peligros activos de la siguiente forma:

"Los peligros activos son una colección de incidentes actuales agregados de fuentes autoritativas de eventos globales. Los eventos en el sistema han sido designados como potencialmente peligrosos a personas, propiedades u otros bienes mediante un proceso automatizado del sistema DisasterAWARE del PDC en conjunto con profesionales de gestión de emergencias." -(traducido de pdc.org/projects/mobile/disaster_alert_app.htm)

Ente otros servicios y aplicaciones se proporciona un *widget*¹ creado sobre la plataforma Adobe Flex que está a disposición de cualquier desarrollador de aplicaciones relacionadas con las emergencias. El widget proporciona una interfaz de mapa sobre la que se pueden mostrar emergencias de una multitud de categorías, además de realizar algunos otros ajustes menores de personalización. Una muestra del widget se puede observar en la **Figura 5**.

¹ Widget: un elemento reutilizable de interfaz gráfica que muestra un conjunto de elementos gráficos contenedores de información.



Figura 5: PDC Active Hazards Widget (pdg.org/widget)

Este widget se puede incorporar tanto en una aplicación de escritorio como una aplicación web, ya que la plataforma Adobe Flex permite ambas opciones. Tiene la ventaja de proporcionar una interfaz unificada y parcialmente personalizable, además de permitir acceso a las alertas. Su principal desventaja es que la fuente de alertas es única: la propia PDC. Las alertas son seleccionadas por la organización, y las fuentes son seleccionadas a priori sin intervención del usuario final.

2.2.2 Aplicaciones web

2.2.2.1 RSOE-EDIS

Esta aplicación web llamada *Emergency and Disaster Information Service* (EDIS) pertenece a la organización Húngara *Hungarian National Association of Radio Distress-Signalling and Infocommunications* (RSOE) y permite visualizar alertas de emergencia de todo el mundo recopiladas por la organización. Se proporcionan dos modos de visualización: sobre un mapa y en listado por categorías. Se puede ver una muestra de la interfaz en las siguientes imágenes:



Figura 6: Mapa con alertas del sistema EDIS (hisz.rsoe.hu)

Current Emergencies						
pd.	Date (UTC)	Event	Country	Location	Level	Details
4	02.05.2011	Volcano Eruption	Ecuador	Cordillera Oriental, [Tungurahua volcano]	██████████	i
	24.04.2011	Complex Emergency	Colombia	Statewide, [32 provinces]	██████████	i
19	26.04.2011	Volcano Activity	Philippines	Province of Batangas (Luzon), [Taal Volcano]	██████████	i

Short Time Event(s)						
pd.	Date (UTC)	Event	Country	Location	Level	Details
	Today	Biological Hazard	USA	State of Colorado, Durango	██████████	i
	Today	Vehicle Accident	Nepal	Sindhuli District, [BP Highway]	██████████	i
	Today	Volcano Eruption	Russia [Asia]	Kamchatka Peninsula (Far-East), [Shiveluch Volcano]	██████████	i
	Today	Technological Disaster	Turkey	Province of Malatya, Malatya	██████████	i
	Today	Volcano Eruption	Russia [Asia]	Kamchatka Peninsula (Far-East), [Kizimen Volcano]	██████████	i
1	Today	Tornado	New Zealand	North Island, Albanv [Albanv Meoa Centre]	██████████	i

Figura 7: Listado de alertas de EDIS (hisz.rsoe.hu)

Earthquake(s)								
Date/Time (UTC)	Magnitude	Area	Country	State/Prov./Gov.	Location	Risk	Source	Details
3.05.2011 14:30:16	2.2	North America	United States	California	Mesquite Oasis	⬆️	USGS-RSOE	i
3.05.2011 14:56:22	2.3	North America	United States	Alaska	Chickaloon	⬆️	USGS-RSOE	i
3.05.2011 14:15:14	3.9	Asia	Kazakhstan		Kazatkom	⬆️	EMSC	i
3.05.2011 14:40:26	3.0	Pacific Ocean	New Zealand	Woodville County	Coutts Island	⬆️	GEONET	i
3.05.2011 14:40:13	3.0	Caribbean	Dominican Republic	Provincia de La Altagracia	Palo Bonito	⬆️	USGS-RSOE	i
3.05.2011 13:15:14	3.5	North America	United States	Alaska	Skwentna	⬆️	USGS-RSOE	i
3.05.2011 14:35:14	2.7	Caribbean	U.S. Virgin Islands		Sorgenfri	⬆️	USGS-RSOE	i
3.05.2011 12:10:12	2.1	Europe	Czech Republic		Dolni Jiretin	⬆️	EMSC	i
3.05.2011 13:10:20	3.5	Africa	Djibouti		Gaherre	⬆️	EMSC	i
3.05.2011 11:55:14	2.1	Caribbean	Puerto Rico		Matilde	⬆️	USGS-RSOE	i
3.05.2011 11:55:22	3.2	North America	United States	Alaska	Eska	⬆️	USGS-RSOE	i
3.05.2011 11:30:12	2.1	North America	United States	California	San Juan Bautista	⬆️	USGS-RSOE	i
3.05.2011 12:10:21	2.4	Europe	Spain		Barcena de la Abadia	⬆️	EMSC	i
3.05.2011 10:30:13	3.1	North America	United States	California	Covina	⬆️	USGS-RSOE	i
3.05.2011 14:15:22	2.3	Europe	Albania		Bradvice	⬆️	EMSC	i
3.05.2011 09:50:13	2.2	Caribbean	Puerto Rico		El Combate	⬆️	USGS-RSOE	i
3.05.2011 10:05:11	4.4	Europe	Russia		Slavnoye	⬆️	EMSC	i

Figura 8: Listado de alertas de terremoto de EDIS (hisz.rsoe.hu)

Tsunami Information						
Pacific Ocean Region						
Date/Time (UTC)	Message	Location	Magnitude	Depth	Status	Details
3.04.2011 04:25 AM	Tsunami Information Bulletin	Solomon Islands	6.9	98 km		
Indian Ocean Region						
Date/Time (UTC)	Message	Location	Magnitude	Depth	Status	Details
3.04.2011 20:15 PM	Tsunami Information Bulletin	South Of Java Indonesia	6.7	0 km		
Caribbean Sea Region						
Date/Time (UTC)	Message	Location	Magnitude	Depth	Status	Details
5.03.2011 14:46 PM	Tsunami Information Statement	South Sandwich Islands Region	6.6	113 km		
Hawaii Region						
Date/Time (UTC)	Message	Location	Magnitude	Depth	Status	Details
3.04.2011 04:26 AM	Tsunami Information Statement	Solomon Islands	6.9	0 km		

Figura 9: Listado de alertas de Tsunami de EDIS (hisz.rsos.hu)

Supervolcanoes Monitoring System								
Name of Volcano	VEI	Continent	Country	Location	Last eruption	Last activity	Status	Details
ake Taupo	8	Australia & New-Zealand	New Zealand	North Island	~26,500 years ago	..		
ake Toba	8	Indonesian Archipelago	Indonesia	Northern Sumatra	~74,000 years ago	..		
Whakamaru	8	Australia & New-Zealand	New Zealand	North Island	~254,000 years ago	..		
Yellowstone Caldera	8	North-America	USA (Pacific Region)	State of Wyoming	640,000 years ago	..		
Crater Lake Caldera	8	North-America	USA (Pacific Region)	State of Idaho	2.1 million years ago	..		
San Juan Caldera	8	North-America	USA (Pacific Region)	State of Colorado	~27.8 million years ago	No data		
Mount Tambora	7	Indonesian Archipelago	Indonesia	Sumbawa Island	1815	20.03.2011		
Mount Fuji	7	Asia	China	The border North Korea and China	~969 CE	No data		
Mount Fuji	7	Asia	Japan	Osumi Islands, Kagoshima Prefecture	~6,300 years ago	29.04.2011		
Crater Lake	7	Europe	Germany	Rhineland-Palatinate	~12,900 years ago	..		

Figura 10: Listado de alertas de volcanes de EDIS (hisz.rsos.hu)

En definitiva es una aplicación útil, tiene opción de visualizar las alertas sobre una interfaz de mapa o en forma de listado y muestra mucha información sobre cada alerta. Las desventajas principales son la falta de personalización para el usuario, la selección cerrada de fuentes y la selección previa de alertas por la organización. La RSOE proporciona acceso directo a las fuentes de alertas CAP para ser utilizadas en cualquier aplicación que se desee, pero este acceso trae consigo un coste mensual que ronda los 300€ (consultado Marzo del 2011).

2.2.2.2 PDC Natural Hazards and Vulnerabilities Atlas

La siguiente aplicación es muy parecida a la proporcionada por RSOE, aunque con una interfaz más limitada. Se trata de una aplicación web de PDC (Pacific Disaster Center) que permite visualizar las alertas de la organización sobre un mapa con diferentes niveles de personalización por capas. (Para más información sobre el PDC consultar el apartado 2.2.1.1). Se puede observar una muestra de la aplicación en la siguiente imagen:



Figura 11: PDC Natural Hazards & Vulnerabilities Atlas (pdc.org/atlas)

Esta aplicación, aunque útil, tiene como desventaja principal la falta de selección de fuentes por parte del usuario. Las alertas provienen única y exclusivamente de la organización PDC y son seleccionadas por la misma.

2.2.3 Aplicaciones Móviles

2.2.3.1 Disaster Alert



Figura 12: Pantallas de la aplicación Disaster Alert del PDC (itunes.apple.com)

Esta aplicación existe para la plataforma iOS y Android, y proviene de la organización PDC (Pacific Disaster Center). (Para más información sobre el PDC consultar el apartado 2.2.1.1). La aplicación proporciona una solución al problema propuesto que conlleva la inserción de una capa adicional entre las organizaciones que emiten las alertas y el usuario. En esta capa la propia organización PDC es la que decide tanto las fuentes de alertas, como su formato final y filtrado. El usuario obtiene por tanto un simple visor de alertas seleccionadas por el PDC, con personalización limitada y sin la habilidad de seleccionar fuentes cualesquiera. Tiene la ventaja de proporcionar visualización tanto en listado como en mapa, además de permitir acceso a los detalles de cada alerta. Como desventaja principal cabe destacar una interfaz sobrecargada (los elementos de la interfaz están muy cercanos y a veces poseen un tamaño demasiado reducido provocando selecciones no intencionadas) y la falta de personalización (tanto selección de fuentes como filtrado de alertas).

2.2.3.2 Quake Alert

Esta aplicación para iPhone tiene una funcionalidad más sencilla, ya que se centra simplemente en mostrar terremotos recientes en forma de lista y sobre un mapa. La única información proporcionada sobre cada terremoto es su lugar y su intensidad. La configuración de la aplicación permite establecer los criterios de aviso de terremoto en base a cercanía a un código postal e intensidad mínima en la escala Richter. Por último, la aplicación da acceso a un formulario proporcionado por la USGS que permite avisar de un terremoto que se haya sentido personalmente. En las siguientes capturas de pantalla se muestra la aplicación:

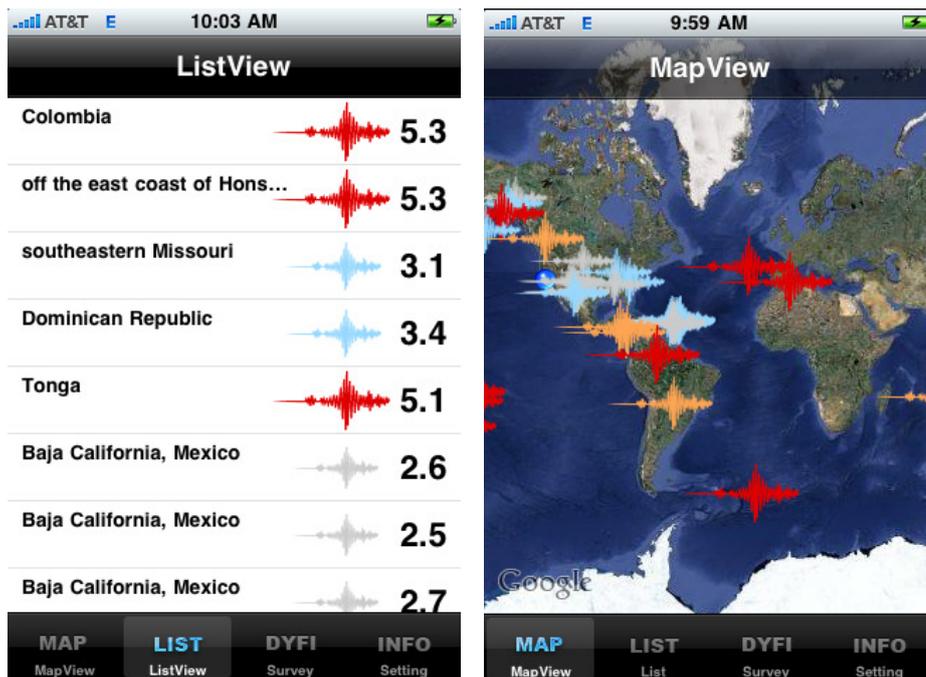


Figura 13: Listado y mapa de terremotos de Quake Alert (itunes.apple.com)

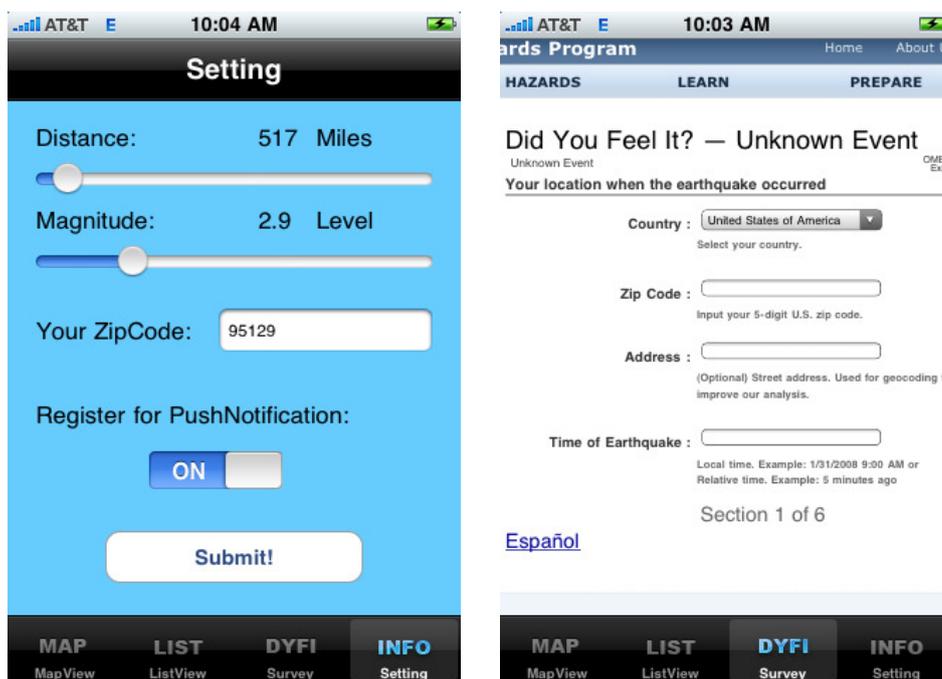


Figura 14: Pantallas de configuración y aviso de Quake Alert (itunes.apple.com)

La aplicación es sencilla y de funcionalidad limitada, aunque contempla algunos de los aspectos definidos para la solución propuesta en este proyecto como la personalización y la interfaz unificada. La personalización es limitada y la fuente de alertas es única y fija (USGS). La aplicación cuenta con notificaciones en tiempo real, además de una visualización en listado y en mapa. La visualización en mapa no es tan clara como podría ser ya que los iconos usados para representar los terremotos se solapan y no son precisos (ver Figura 13).

2.2.4 Tabla comparativa de aplicaciones existentes

En la siguiente tabla se resumen las diferentes aplicaciones existentes que se han analizado, destacando los aspectos clave según la definición del problema y sus respectivas ventajas y desventajas. La última entrada de la tabla se refiere a la solución propuesta en este proyecto.

Nombre	Plataforma	Descripción	Fuentes	Visualización	Personalización	Ventajas	Desventajas
PDC Widget	Escritorio	Widget compuesto por un mapa con personalización de qué alertas se ven sobre la misma. Diseñado para ser insertado en cualquier aplicación.	PDC	Mapa	Categorías de alertas a mostrar	Desplegable como parte de otras aplicaciones	Fuente fija, poca personalización
RSOE-EDIS	Web	Aplicación web con visualización de emergencias mundiales sobre mapa y en listados detallados y separados por categorías.	Múltiples	Mapa y tablas, fijos	Sólo geográfica	Información variada y detallada	Mínima personalización, coste mensual de acceso a datos
PDC Hazards Atlas	Web	Aplicación web con visualización de alertas del PDC sobre mapa.	PDC	Mapa	Categorías de alertas a mostrar	Múltiples categorías y buena visualización en mapa	Falta de detalle, fuente fija
Disaster Alert	Móvil	Aplicación móvil para visualizar las alertas de PDC en listado y sobre mapa.	PDC	Listado y mapa	Ninguna (de alertas)	Visualización en listado y mapa	Fuente fija
Quake Alert	Móvil	Aplicación móvil para visualizar información básica de terremotos en listado o sobre un mapa.	USGS	Listado sencillo, mapa	Filtrado por proximidad e intensidad de terremoto	Filtrado por proximidad, comunicación bidireccional, notificaciones	Fuente única y fija, interfaz sencilla con poco detalle
Solución propuesta	Móvil	Aplicación móvil para mostrar alertas de múltiples fuentes, con filtrado y personalización.	Múltiples definidas por usuario	Listado, resumen, detalle y mapa	Ordenación de listado. Filtrado por categoría, severidad, urgencia, certeza y caducidad	Selección de fuentes, filtrado por múltiples campos, varias visualizaciones (desde listado hasta detalle máximo)	Falta de comunicación bidireccional y notificaciones en tiempo real.

Tabla 2: Tabla comparativa de aplicaciones existentes

2.3 Tecnologías móviles

2.3.1 Introducción

En la última década las tecnologías móviles han avanzado a un paso acelerado. La adopción del teléfono móvil como medio de comunicación ha sido una de las grandes tendencias mundiales en el sector tecnológico. Desde países desarrollados donde el número de móviles supera a la población (Alemania - 130% de penetración, (Machine 2 Machine), España - 112% (ITU, 2009)) hasta países Africanos en desarrollo donde la adopción es cada vez mayor - el impacto es innegable.

A la vez que la telefonía móvil se hace más accesible, aumentan también sus capacidades. Las redes 3G, y pronto 4G, proporcionan velocidades de conexión de datos de banda ancha, permitiendo aplicaciones nunca antes vistas en plataformas móviles. Si además se añade la capacidad de computación del propio dispositivo, las plataformas móviles se presentan como un pilar indispensable de la tecnología actual.

La llegada de los llamados *smartphones*, móviles con capacidades próximas a las de un ordenador, supone la siguiente evolución del sector. Según estudios recientes la adopción de smartphones ronda el 27% entre la población Estadounidense (The Nielsen Company, 2011), y el 31% en Europa (comScore, 2011).

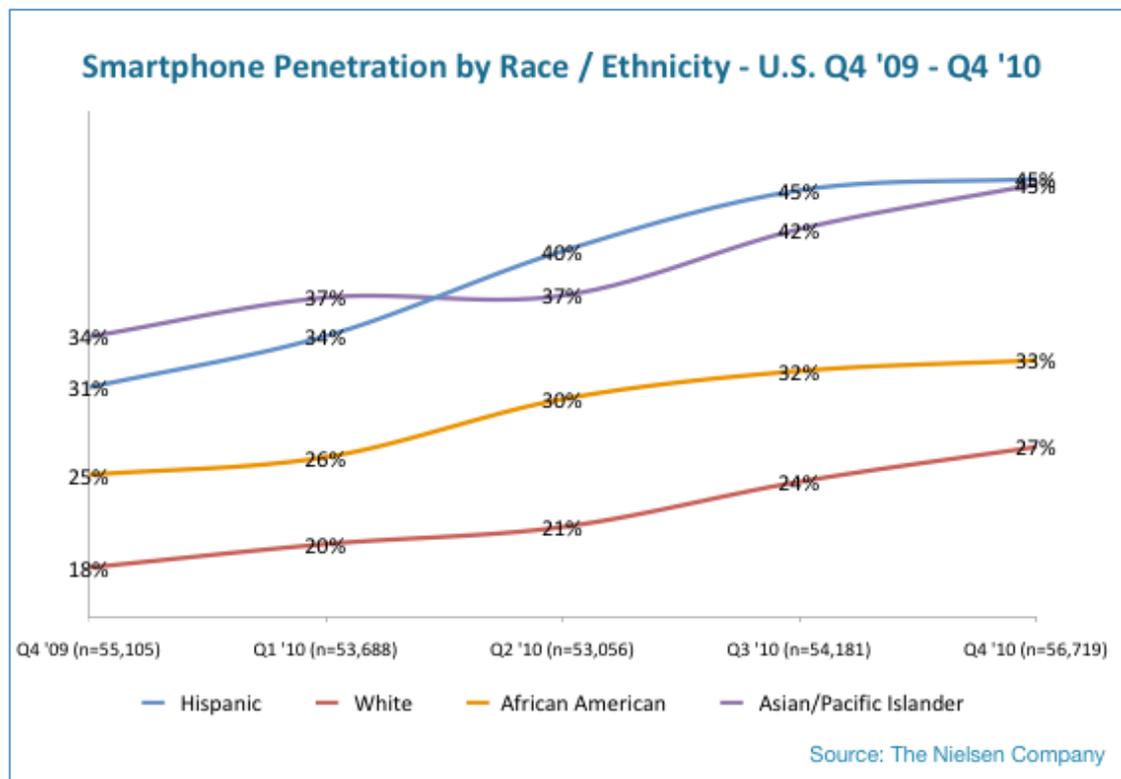


Figura 15: Adopción de smartphones en EE.UU. (blog.nielsen.com)

Dentro del sector de los smartphones existen una variedad de plataformas con cuotas de mercado importantes. Estas plataformas incluyen: iOS de Apple¹, Android de Google², BlackBerry de RIM³ y Windows Phone 7⁴ de Microsoft y Nokia. La siguiente tabla muestra la cuota de mercado de estas plataformas en EE.UU.:

¹ <http://developer.apple.com/devcenter/ios/index.action>

² <http://developer.android.com/index.html>

³ <http://us.blackberry.com/developers/>

⁴ <http://create.msdn.com/>

Top Smartphone Platforms 3 Month Avg. Ending Feb. 2011 vs. 3 Month Avg. Ending Nov. 2010 Total U.S. Smartphone Subscribers Ages 13+ Source: comScore MobiLens			
	Share (%) of Smartphone Subscribers		
	Nov-10	Feb-11	Point Change
<i>Total Smartphone Subscribers</i>	100.0%	100.0%	N/A
Google	26.0%	33.0%	7.0
RIM	33.5%	28.9%	-4.6
Apple	25.0%	25.2%	0.2
Microsoft	9.0%	7.7%	-1.3
Palm	3.9%	2.8%	-1.1

Figura 16: Cuota de mercado de plataformas móviles en EE.UU. (comScore, 2011)

Uno de los aspectos más importantes de estas plataformas es el medio de difusión de aplicaciones para las mismas. Cada una cuenta con una plataforma centralizada donde los usuarios pueden acceder a todas las aplicaciones disponibles, visualizar información sobre cada una y optar por instalarlas (de forma gratuita o pagando). Estos llamados *stores* o *markets* están integrados en la plataforma, y eliminan la inversión del desarrollador a la hora de difundir su aplicación. Esto es un aspecto importante, ya que la facilidad de difusión es algo que favorece la adopción de una aplicación y su viabilidad funcional y económica.

2.3.2 Plataforma iOS

Se trata de la plataforma creada por Apple Inc. en el 2007, inicialmente para el iPhone pero que ahora se ha extendido al iPad y AppleTV también. Se trata de la tercera plataforma en cuanto a cuota de mercado (aunque las tendencias apuntan su llegada inminente al segundo puesto), pero es la primera plataforma en cuanto a número de aplicaciones e ingresos por aplicaciones. Estas ventajas atraen cada vez a más desarrolladores a la plataforma, que lo impulsa aún más en el mercado. Las características particulares de la plataforma son considerados desventajas por unos, y ventajas por otros (Reed, 2008). Entre estas características cabe destacar:

- **Poca variación de hardware:** en los últimos 4 años sólo han existido 4 dispositivos iPhone diferentes en cuanto a hardware, mientras que el software ha ido evolucionando y actualizándose de tal manera que añade funcionalidad nueva a dispositivos anteriores. Los avances resultan significativos en cuanto a prestaciones cada dos años, que coincide con el ciclo habitual de los usuarios de cambiar al modelo más actual. Esto implica que desde el punto de vista del desarrollador no es necesario considerar configuraciones o prestaciones variadas a la hora de crear una aplicación.
- **Estabilidad del software:** las actualizaciones frecuentes y aseguradas para todos los dispositivos conlleva que la gran mayoría de los usuarios tienen su software con la versión más reciente. Desde el punto de vista del desarrollador esto conlleva gran estabilidad, y la confianza de que las aplicaciones seguirán funcionando en el futuro.
- **Entorno de desarrollo avanzado:** el entorno proporcionado por Apple para desarrollar es considerado por el autor como uno de los mejores en el

ámbito de las plataformas móviles. Entre otras herramientas, dispone de un simulador que permite ejecutar las aplicaciones sin necesidad de un dispositivo físico lo cual permite acelerar en gran medida el proceso de desarrollo.

- **Plataforma de venta:** el llamado *App Store* de iOS es el lugar en el que todas las aplicaciones se pueden descargar y comprar. La distribución es extremadamente sencilla, tanto en el caso de aplicaciones gratuitas como de pago. Además de la distribución también se facilita la comunicación con los usuarios finales, lo que puede ayudar a mejorar la aplicación en revisiones futuras o actualizaciones.

De cara a la solución propuesta en este proyecto se consideran estas características como ventajas de la plataforma. La principal desventaja de esta plataforma es la reducida selección de hardware, limitado a sólo un dispositivo nuevo cada año. Algunos aspectos técnicos que faltan en los dispositivos son (hasta la fecha de este proyecto): conectividad 4G, sensores NFC¹, pantallas de 4 pulgadas o mayores, cámaras 3D y ampliación externa de memoria (por tarjetas SD o similar). En definitiva la plataforma iOS destaca por su gran estabilidad, su liderazgo en cuanto a distribución de aplicaciones y su facilidad para el desarrollo - aunque sufre una falta de variedad en cuanto a hardware.

2.3.3 Plataforma Android

La plataforma Android es la impulsada por Google de manera libre y abierta, desarrollada internamente y utilizada en hardware de múltiples fabricantes con una gran variedad de características. Como se pudo observar en el estudio mencionado anteriormente, es la plataforma de mayor alcance en cuanto a cuota de mercado, debido principalmente a la gran variedad de dispositivos disponibles que la utilizan como su sistema operativo. Aunque se definen una serie de requisitos mínimos de hardware para poder utilizar el software, existe margen suficiente para crear dispositivos con un rango amplio de prestaciones. Desde dispositivos sencillos y baratos, hasta aquellos que incorporan las últimas tecnologías como NFC o pantallas qHD². Esta variedad proporciona flexibilidad a la hora de emprender nuevas aplicaciones ya que se reducen en gran medida los límites tecnológicos impuestos por la plataforma. Aun así, suele considerarse como una desventaja de la plataforma desde el punto de vista del desarrollador ya que conlleva una adaptación muy extensa de cualquier aplicación para asegurar su correcto funcionamiento en diferentes dispositivos (Sigal, 2009).

En cuanto a la difusión de aplicaciones existen una variedad de soluciones de tipo *Store* o *Market* tanto para desarrolladores como usuarios, desde la oficial de la plataforma Android hasta otras alternativas como la proporcionada por Amazon. Existe una mayor libertad y menor cantidad de restricciones, que en algunos casos potencia el ecosistema de la plataforma pero en otros actúa en su detrimento. Aunque aparentemente se trata de una plataforma muy similar a la de iOS descrita antes, no se ha conseguido el mismo éxito en cuanto a número de aplicaciones disponibles. Las tendencias de la plataforma son de menor cantidad de ingresos que un desarrollador equivalente en iOS, por lo que la plataforma sufre un crecimiento más lento en este sentido. Hay que añadir además la peor experiencia que existe en Android en cuanto al entorno de desarrollo especialmente a la hora de considerar la gran variedad de dispositivos disponibles con características muy variadas (resolución de pantalla, cámaras, potencia de procesamiento, memoria, etc.).

De todas formas, aunque existen las desventajas descritas la plataforma está en auge y cada vez se considera más madura. Sin lugar a duda es una plataforma a considerar a la hora de desarrollar aplicaciones móviles, y difícil de ignorar dada su gran extensión y adopción entre los consumidores.

¹ NFC: Near Field Communications, una tecnología de comunicación por cercanía o contacto entre dispositivos parecido al Bluetooth (utilizado en algunos dispositivos como medio de pago).

² qHD: quarter HD, una resolución de pantalla de 960x540

2.3.4 Otras plataformas

Aunque iOS y Android son las plataformas dominantes en el mercado existen otras que también se deben considerar, en concreto BlackBerry OS de RIM y Windows Phone 7 de Microsoft. Ambas plataformas ofrecen la posibilidad de crear aplicaciones a medida para sus dispositivos, aunque en este ámbito se encuentran en una gran desventaja por razones distintas. En el caso de RIM, la plataforma BlackBerry tiene una cuota de mercado importante aunque cada trimestre va decayendo principalmente a favor de Android (Paczkowski). BlackBerry está orientado más al sector profesional y empresarial, por lo que se da preferencia a características diferentes que en iOS y Android. Por esta razón no es la plataforma más adecuada para un proyecto orientado al consumidor habitual.

Por otra parte Windows Phone 7 es una plataforma nueva y que aún se encuentra en su infancia (Bleeker, 2010). En principio tiene el mismo potencial que iOS y Android, pareciéndose más al segundo por su separación software-hardware y su variedad de dispositivos. Desde el punto de vista del desarrollador es una plataforma con algo de incertidumbre en estos momentos. Debido a estas desventajas no se ha considerado necesario para este proyecto realizar un estudio más detallado de estas otras plataformas.

2.3.5 Conclusiones

Queda claro por lo tanto que las plataformas móviles tienen un futuro positivo y de crecimiento, y que tanto los factores tecnológicos (comunicación, prestaciones) como los factores económicos (adopción, difusión) muestran su viabilidad como una plataforma de desarrollo para la solución propuesta en este proyecto.

Entre las plataformas analizadas se ha optado por la de Apple (iOS para iPhone) para desarrollar la prueba de concepto. La decisión ha sido impulsada por las ventajas que presenta la plataforma a la hora de facilitar el desarrollo, y por el hecho de que las desventajas en ciertas especificaciones técnicas no tienen ningún impacto negativo en la aplicación tal y como se ha propuesto. Sería perfectamente factible desarrollar la solución para la plataforma Android, o incluso una de las otras plataformas menos extendidas, y en tal caso buena parte del estudio previo, análisis y diseño contenidos en este proyecto servirían como base.

3 Desarrollo de la solución

Como se ha descrito en los apartados anteriores el objetivo del proyecto es crear una solución para la obtención de alertas de emergencia que contemple la agregación de alertas de múltiples fuentes bajo una interfaz unificada con opciones de personalización. Se ha escogido una plataforma móvil de última generación para el desarrollo de la solución ya que reúne una serie de características favorables:

- Movilidad
- Prestaciones tecnológicas
- Extensión y fácil adopción
- Accesibilidad

Se define una plataforma móvil de última generación como aquella en las que los dispositivos contengan al menos las siguientes características:

- Pantalla de al menos 2.5 pulgadas, preferiblemente táctil
- Conectividad de datos 3G o superior
- Creación de aplicaciones a medida (SDK)
- GPS (deseable pero no estrictamente necesario)

Algunos ejemplos de plataformas que cumplen estos requisitos son: iOS de Apple, Android de Google, BlackBerry OS de RIM y Windows Phone 7 de Microsoft. Se puede encontrar un estudio de estas plataformas en el apartado 2.3.

Tras estudiar el estado actual de las tecnologías de difusión y obtención de alertas de emergencia y describir sus deficiencias, se plantea el desarrollo del proyecto en base a los siguientes apartados:

- 1) **Fuentes CAP:** ya que el objetivo del proyecto no incluye el aspecto de generación o transmisión de las alertas se han escogido fuentes existentes de alertas basadas en el estándar CAP. Los mensajes de alerta se forman siguiendo este estándar con una estructura de XML, y son entregados mediante el uso de protocolos de difusión como RSS.
- 2) **Extracción y manipulación de alertas:** una vez obtenidas las alertas se debe proceder a la extracción de la información que contienen, y aplicar cualquier manipulación deseada antes de presentarla al usuario. Un ejemplo de esto puede ser la interpretación de datos de localización para su presentación en una vista de mapa, o la conversión de tiempos a medidas relativas.
- 3) **Presentación de alertas:** una vez preparada la información definitiva de la alerta se puede proceder a su presentación al usuario. Se contemplan varios aspectos a la hora de diseñar y crear la presentación de las alertas, como por ejemplo: los estándares de interfaz de la plataforma, las metáforas conocidas por el usuario, los diferentes métodos de visualización y la importancia de cada parte del mensaje.
- 4) **Personalización de la aplicación:** la aplicación debe permitir un grado de personalización en función de las necesidades del usuario. Esta personalización incluye: filtrado de las alertas, selección de fuentes, opciones de visualización, etc.

Para estudiar una solución partiendo de esta base se ha creado una prueba de concepto, concretamente una aplicación sobre la plataforma iOS para el dispositivo iPhone de Apple. Se ha optado por este método ya que ayuda a entender de primera mano todo lo que conlleva la creación de una aplicación de este estilo. Desde el uso de estándares para la difusión de alertas, hasta las necesidades de usabilidad.

3.1 Análisis y Diseño

Como primera fase de la creación de una aplicación es necesario realizar un análisis de los requisitos y un diseño preliminar. En la fase inicial de análisis se estudió el estándar CAP y las capacidades técnicas de la plataforma iOS para plasmar los requisitos de la aplicación. Para el diseño se ha optado por usar la técnica del prototipado o maquetado rápido para estudiar los aspectos de diseño. En los siguientes apartados se hace referencia a algunos aspectos del estándar CAP descrito en el apartado 2.1.1.

3.1.1 Análisis inicial

Una vez concretado el uso del estándar CAP para los mensajes de alerta de emergencias se pasó a un estudio más detallado de su contenido y su estructura. La aplicación se centra en torno a la presentación de la información contenida en un mensaje CAP, por lo que la funcionalidad estará estrictamente ligada al contenido del mismo. El siguiente listado contiene los puntos clave del análisis resultante que comienzan a especificar los requisitos de la aplicación:

1. **Obtención de alertas:** se debe poder escoger las fuentes de alertas deseadas, sea seleccionando de un conjunto predeterminado o incluso indicando una fuente propia de alertas que siga el estándar CAP.
2. **Presentación de alertas:** aunque una alerta está compuesta por múltiples campos, cada una con un significado concreto, existen algunos campos más significativos que llevan la información más crítica de la alerta y que suelen ser los obligados en el estándar. Estos campos deberían utilizarse para presentar la alerta de una forma muy abreviada en conjunto con otras alertas (v.g. el título de la alerta, su origen y su importancia). A continuación se debe permitir una vista más amplia de la alerta, con un mayor número de campos y de manera aislada de otras alertas. Finalmente deben ser accesibles todos y cada uno de los campos de la alerta sin modificación o manipulación.
3. **Filtrado de alertas:** debe ser posible el filtrado de alertas en varias dimensiones, incluyendo pero no limitado a: su categoría, su urgencia/severidad/certeza y su caducidad.
4. **Concepto de importancia:** el estándar CAP proporciona tres campos que permiten describir la importancia de una alerta de manera más detallada: Urgencia, Severidad y Certeza. Se debe establecer un criterio para unificar los tres valores para obtener un nivel global de importancia de la alerta que simplifique la comprensión para el usuario.
5. **Visualización en mapa:** si una alerta incluye información de localización se debe permitir su visualización sobre un mapa.
6. **Visualización de recursos:** si una alerta viene acompañada de recursos adicionales (multimedia, páginas web) se debe dar acceso directo a ellos desde la aplicación.
7. **Patrones de diseño:** se debe estudiar el uso de patrones conocidos de diseño como el uso de colores, iconos, símbolos, etc. para mejorar la facilidad de uso de la interfaz.

Una vez establecidas estas especificaciones se pasó a una segunda fase de análisis que conllevó el estudio del uso real del estándar por parte de las organizaciones emisoras de alertas. El estándar permite una cierta flexibilidad a la hora de generar las alertas, y es importante comprobar el uso que realmente se hace antes de continuar con la especificación más detallada de la aplicación. Cabe destacar los siguientes resultados del análisis:

- Muchas organizaciones no emiten mensajes en el formato CAP de manera directa, sino que engloban un mensaje CAP dentro de otro formato como RSS. Ya que la estructura de estas fuentes difiere para cada organización, su tratamiento para la extracción de las alertas CAP variará.
- Todas las fuentes encontradas incluyen además del segmento obligado <alert> al menos un segmento <info>. Esto es importante ya que se puede asegurar la disponibilidad de los campos *severity*, *urgency* y *certainty* y *category*, que son obligatorios. Por lo general también se suele incluir el campo *headline* que proporciona una descripción abreviada de la alerta.
- No todas las fuentes incluyen información de localización de la alerta, e incluso cuando la incluyen no respetan por completo el estándar. Éste dice que se puede indicar la localización mediante coordenadas (zona poligonal o circular) o un *geocode* (geoetiqueta). Si se utiliza un geocode se recomienda acompañarlo de coordenadas, ya que no es siempre posible la transformación de una a otra (los geocode se utilizan principalmente cuando los sistemas que se comunican conocen previamente el sistema de codificación).
- Por lo general no se rellenan la mayoría de los campos opcionales de la alerta (según las fuentes estudiadas). Esto implica que no se puede depender de la existencia de todos los valores en un mensaje a la hora de crear la aplicación.

En base a todo el análisis descrito se pasó a la fase de diseño de la aplicación descrita en el siguiente apartado.

3.1.2 Diseño mediante prototipado rápido

En base al análisis realizado, se realizó una definición general por escrito de los distintos aspectos de la interfaz antes de proceder a una maqueta de diseño

- La aplicación contendrá tres vistas globales: Listado de alertas, Mapa de alertas y Ajustes de la aplicación.
- El listado de alertas mostrará todas las alertas (filtradas según lo establecido en los ajustes) con la información mínima necesaria: su título/ID, la fuente, la edad de la alerta y alguna marca de su importancia. El listado se podrá ordenar por fecha o por importancia, y se podrá seleccionar una única categoría o todas.
- Desde el listado se podrá seleccionar una alerta para visualizar una vista resumida de los campos de la misma, pero más detallada que el listado. Debe incluir: título, fecha, fuente, urgencia, severidad, certeza y descripción. Desde esta vista se podrá acceder directamente a los siguientes contenidos si los contiene la alerta: los recursos, la vista en mapa, la página web de la alerta y todos los campos sin formato del mensaje CAP.
- La vista de mapa de alertas mostrará sobre un mapa mundial marcadores con todas las alertas del listado que contengan información de localización. A través de cada marcador se puede acceder a la misma vista resumen que se accede desde el listado.

- En la vista de ajustes de la aplicación se permitirá: seleccionar las fuentes deseadas y definir las opciones de filtrado por urgency, certainty, severity y expiration date.
- Se utilizará la gama de colores Rojo-Naranja-Amarillo-Azul-Gris para representar la importancia de una alerta en base a su severity y urgency. Esta gama de colores está adaptada de la gama utilizada por organizaciones de todo el mundo para representar una escala de importancia, como por ejemplo la mostrada en la **Figura 17** utilizada hasta el 2011 por el departamento de seguridad nacional de EE.UU. (dhs.gov). Se ha cambiado el color verde por gris para representar falta de información, ya que verde puede connotar la ausencia de riesgo. Ya que se están tratando alertas existe una dimensión de riesgo inherente.

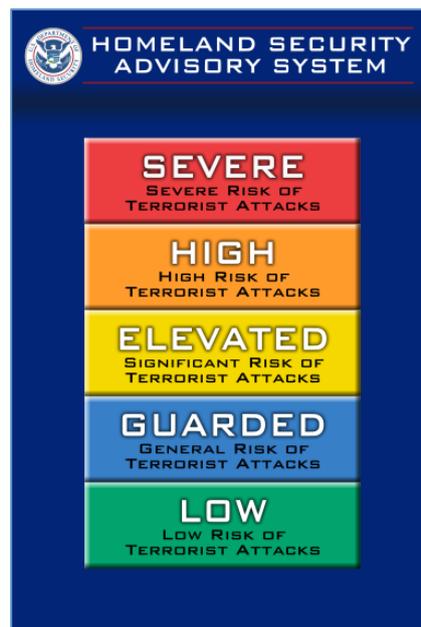


Figura 17: Escala de colores de alerta de seguridad nacional en EE.UU.

La técnica de prototipado rápido se basa en la creación de una maqueta que muestra sólo algunos aspectos de la funcionalidad y el diseño de la aplicación, que junto con los comentarios del cliente o equipo de desarrollo va evolucionando para obtener una visión de lo deseado para el producto final. En el caso de este proyecto se realizó el proceso de análisis y diseño consultando con el tutor del proyecto, que tomó el rol tanto de experto técnico como cliente final (aquel que conoce cómo quiere que sea la aplicación final y al que se le entregará).

Para el maquetado se usó una herramienta llamada MockApp (mockapp.com) que consta de un conjunto de plantillas de elementos de interfaz de iOS que junto con una aplicación de presentaciones (PowerPoint de Microsoft® o Keynote de Apple®) permite crear diseños interactivos de una aplicación. En base al análisis y los aspectos de diseño establecidos se creó una maqueta interactiva de la aplicación que facilitaba el proceso de análisis y diseño por varias razones:

1. Permite plasmar muy rápidamente los diseños propuestos, probarlos y modificarlos sin coste alguno. Este método iterativo permite la evolución continua del diseño y asegura la especificación correcta de la aplicación antes de comenzar con su implementación (fase en la que los cambios son más costosos).
2. Sirve como un medio para transmitir las posibilidades técnicas de la plataforma. Al usar sólo elementos de interfaz estándar de la plataforma el diseño planteado presenta un aspecto muy similar a un producto final. Esto reduce las posibles desviaciones del diseño durante el desarrollo. Esto es beneficioso tanto para el cliente (ya que sabe que lo que ve es lo

que se obtendrá) y para el desarrollador (ya que minimiza los cambios durante la implementación que pueden resultar costosos).

3. Muestra el flujo de interacción de la aplicación además de mostrar su diseño. La naturaleza dinámica del maquetado escogido enriquece el proceso de análisis en mayor medida que las maquetas estáticas. Resulta muy sencillo transmitir tanto la relación entre las diferentes interfaces como el uso de la aplicación en sí. Este aspecto es crítico para definir bien una aplicación, y facilitararlo favorece al proyecto.

En las siguientes figuras se muestran algunas de las interfaces generadas en el maquetado dinámico de la aplicación. Se puede observar el icono de enlace (🔗) que se utiliza para crear los aspectos dinámicos de la maqueta.



Figura 18: Maqueta de listado de alertas y selección de categorías



Figura 19: Maqueta de vista resumen y vista detalle de una alerta



Figura 20: Maqueta de vista en mapa de una alerta

Este proceso de maquetado rápido sirvió como punto de partida del proyecto. Se consiguió especificar la aplicación de una manera más general, y en base a los resultados se pasó a un análisis más detallado y una especificación más rigurosa, que se detalla en el apartado Metodología de desarrollo.

3.2 Implementación del prototipo

3.2.1 Metodología de desarrollo

Un aspecto importante de la creación de la prueba de concepto es la selección de la metodología de desarrollo a utilizar. La metodología se refiere a los pasos seguidos para la especificación, el diseño, la implementación y la evaluación de un producto software. Para este proyecto se ha optado por usar una metodología de tipo ágil, adaptada a las condiciones particulares del proyecto (concretamente la de un único desarrollador en vez de un equipo). Existen múltiples metodologías ágiles, pero todas comparten una serie de características comunes que componen el núcleo del movimiento ágil. En los siguientes apartados se describen las metodologías ágiles en general, se detalla una de ellas (SCRUM) y finalmente se explican las adaptaciones realizadas a esta metodología para usarla en el proyecto.

3.2.1.1 Metodologías ágiles

El concepto de metodología ágil fue introducido en el *Manifiesto Ágil* del 2001 (Agile Alliance, 2011), y se puede resumir como aquella metodología de software basada en desarrollo incremental e iterativo. El aspecto incremental conlleva la generación periódica de un producto o solución aunque sea parcial, y un proceso iterativo es aquel que se compone de una serie de fases ordenadas que se repiten cíclicamente hasta llegar a un resultado final. En su página web (agilealliance.org) la alianza ágil se describe de la siguiente manera:

"La Alianza Ágil es una organización sin ánimo de lucro con miembros de todo el mundo entregados al avance de los principios y las prácticas del desarrollo ágil. La Alianza Ágil apoya a aquellas personas que exploran y aplican principios y prácticas ágiles para hacer una industria del software más productiva, humana y sostenible. Compartimos nuestra pasión de entregar mejor software cada día."

Estas metodologías buscan alejarse de aquellas más antiguas modeladas en procesos industriales, como el modelo en cascada. Se busca adaptar la forma de trabajar y desarrollar proyectos teniendo en cuenta las particularidades de la creación del software. El manifiesto ágil es el siguiente:

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

*Individuos e interacciones sobre procesos y herramientas
Software funcionando sobre documentación extensiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan*

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

En base a este manifiesto se han descrito una serie de principios del desarrollo ágil, que a su vez han servido como la base de múltiples metodologías ágiles diferentes:

Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.

Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.

Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.

El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

El software funcionando es la medida principal de progreso.

Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.

La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.

Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.

A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Como se ha mencionado han surgido una multitud de metodologías ágiles diferentes en torno a este movimiento. Algunas proporcionan reglas y procesos más estrictos, e imponen condiciones más específicas sobre su aplicabilidad (como el tamaño del proyecto o del equipo); mientras que otras son más sencillas y simplemente sirven de guía para un desarrollo ágil.

La experiencia personal en el desarrollo de software con metodologías ágiles ha sido siempre positiva. En comparación con metodologías tradicionales, y desde el punto de vista tanto del rol de análisis y diseño como el de desarrollador, las ágiles permiten llevar un proyecto de manera más natural y agradable. Las metodologías tradicionales se definieron basándose en los procesos de ingenierías clásicas, por lo que no consideran las características particulares del desarrollo de software como su mayor flexibilidad y la posibilidad de evolucionar un prototipo sin un gran coste.

A la hora de elegir la metodología concreta se decidió basar la elección en la experiencia personal y en un estudio comparativo breve de algunas de las más significativas. El objetivo era conseguir una metodología que permitiese estructurar y documentar adecuadamente el proceso de desarrollo, y que a la vez se adaptara a las características particulares del proyecto: un único desarrollador y equipo reducido. La siguiente tabla recoge una comparativa de algunas de las metodologías ágiles más significativas y usadas.

Metodología	Características	Roles	Manejo de Requisitos	Seguimiento del trabajo
Scrum	Iterativo e incremental, contempla el equipo completo desde cliente hasta desarrollador y describe el proceso general de análisis y desarrollo.	<i>Scrum Master</i> <i>Equipo de desarrollo</i> <i>Product Owner</i> <i>Stakeholders</i> <i>Managers</i>	<u>Product Backlog</u> : documento principal de requisitos, abierto y flexible <u>Sprint backlog</u> : subconjunto del product backlog a ser realizado en una iteración, se define el "como" y se crean tareas de 4 a 16 horas de trabajo	<u>Daily scrum</u> : diario y corto (15min), sirve para actualizar el progreso de la iteración y resolver problemas <u>Sprint planning</u> : al comienzo de cada iteración, se escogen los requisitos a implementar <u>Sprint review</u> : para comprobar qué se consiguió en la iteración. <u>Sprint retrospective</u> : sirve para reflexionar sobre el método de trabajo, "qué fue bien y qué puede mejorar".
XP	Iterativo con ciclos cortos, describe más a fondo el método de desarrollo. Define valores y principios a seguir, sin entrar en detalles concretos.	<i>Representante del cliente</i> <i>Desarrolladores</i>	Se expresan los requisitos directamente con pruebas automáticas de aceptación. Se utilizan historias del usuario (descripciones de la funcionalidad) para entender las necesidades del cliente Las historias se interpretan y subdividen en tareas de implementación	Se realizan ciclos cortos (1-2 semanas) donde se escucha, diseña, implementa y prueba incrementando el sistema
Crystal	Se centra en las personas del equipo, no tanto en procesos o documentos	Ninguno definido	Ninguno en concreto	Entregas frecuentes de código usable al cliente Reflexión para mejorar el proceso Comunicación constante, preferiblemente en persona
ASD (Adaptive Software Development)	Adaptar el proceso de desarrollo constantemente según las necesidades del proyecto	Ninguno definido	Descomponer el proyecto en pasos a realizar	Tres fases: 1. Especulación (análisis) 2. Colaboración (desarrollo) 3. Aprendizaje (iteración, reflexión)

Metodología	Características	Roles	Manejo de Requisitos	Seguimiento del trabajo
FDD (Feature Driven Development)	El objetivo debe ser entregar software funcional desde el punto de vista del cliente	<p><i>Roles claves:</i> Administrador, Arquitecto jefe, Manager desarrollo, Programador jefe, Propietarios de clases, Experto en dominio</p> <p><i>Roles de Soporte:</i> Administrador de entrega, Guru lenguaje, Ingeniero construcción, Especialista en herramientas, Administrador de sistema</p>	Se centran en funcionalidades que se diseñan, planifican e implementan.	Para cada funcionalidad hay "milestones" que ayudan a llevar el seguimiento del trabajo
DSDM (Dynamic Systems Development Method)	<ul style="list-style-type: none"> -El involucramiento del usuario es imperativo. -Los equipos de DSDM deben tener el poder de tomar decisiones. -El foco está puesto en la entrega frecuente de productos. -La conformidad con los propósitos del negocio es el criterio esencial para la aceptación de los entregables. -El desarrollo iterativo e incremental es necesario para converger hacia una correcta solución del negocio. -Todos los cambios durante el desarrollo son reversibles. -Los requerimientos están especificados a un alto nivel. -Las pruebas son integradas a través del ciclo de vida. -Un enfoque colaborativo y cooperativo entre todos los interesados es esencial. 	<p><i>Executive Sponsor</i></p> <p><i>Visionary</i></p> <p><i>Ambassador User</i></p> <p><i>Advisor User</i></p> <p><i>Project Manager</i></p> <p><i>Technical Co-ordinator</i></p> <p><i>Team Leader</i></p> <p><i>Developer</i></p> <p><i>Tester</i></p> <p><i>Scribe</i></p> <p><i>Facilitator</i></p> <p><i>Specialist Roles Business Architect, Quality Manager</i></p> <p><i>System Integrator</i></p>	Se manejan requisitos de una forma muy parecida a las metodologías tradicionales: funcionales y no-funcionales, en un documento completo	<p>Se siguen 4 fases:</p> <ol style="list-style-type: none"> 1. Estudio (análisis) 2. Iterar el modelo funcional 3. Diseñar y construir prototipo 4. Implementación final del prototipo <p>Se van iterando, dentro de cada una se establece un calendario entre los participantes</p>

Tabla 3: Comparativa de metodologías ágiles

3.2.1.2 SCRUM

En base a la experiencia personal y el estudio comparativo realizado, se optó por la metodología SCRUM para desarrollar el proyecto. Esta metodología es una de las más utilizadas en la Ingeniería de Software Ágil, y es especialmente fiel a los principios del manifiesto ágil en su tratamiento de requisitos y el seguimiento del trabajo.

Scrum es una metodología ágil que resulta apropiada en los proyectos en los que se trabaja con requisitos poco estables, y que requieren rapidez y flexibilidad. Scrum es una metodología de desarrollo muy simple, en la que en el seguimiento se basa en la adaptación continua al contexto de la evolución del proyecto.

El desarrollo se inicia desde la visión general del producto final, dando detalle solo a las funcionalidades de mayor prioridad o necesidad que se van a desarrollar en primer lugar. Se contemplan varios roles divididos en dos grupos:

1. **Product Owner y Stakeholders:** el cliente final y cualquier persona interesada en o afectada por el resultado del proyecto. Son aquellos que conocen la especificación deseada y a quienes se les entregará el producto final.
2. **Equipo técnico:** Scrum master, desarrolladores y mánagers encargados tanto del desarrollo técnico del proyecto como de su gestión y control.

Cada uno de los ciclos de desarrollo es una iteración (*sprint*) que produce un incremento terminado y funcional del producto. Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves de seguimiento en las que todo el equipo revisa el trabajo realizado desde la reunión anterior y el previsto hasta la reunión siguiente.

Requisitos:

El Product Backlog es el inventario de funcionalidades, mejoras y requisitos que deben estar presentes en el producto al final del proyecto. Éstas son las llamadas Historias de Usuario. Representan todo aquello que esperan los clientes, usuarios, y en general los interesados en el producto. Todo lo que suponga un trabajo que debe realizar el equipo de desarrollo tiene que estar reflejado en el Backlog.

El Product Backlog se crea y va evolucionando durante el proyecto siguiendo los siguientes pasos.

1. Definición de las Historias de Usuario
2. Priorización de las Historias de Usuario
3. Estimación del esfuerzo necesario
4. Creación de tareas propias de cada Historia
5. Definición de las pruebas de aceptación para cada Historia
6. Revisión y actualización de Historias

El Sprint Backlog es la lista que descompone las funcionalidades del Product Backlog en las tareas necesarias para construir un incremento: una parte completa y operativa del producto. En el Sprint Backlog se asigna a cada tarea la persona que la va a llevar a cabo, y se indica el tiempo de trabajo tanto estimado como restante. Es útil porque descompone el proyecto en tareas de tamaño adecuado para determinar el avance a diario e identificar riesgos y problemas sin necesidad de procesos complejos de gestión.

Seguimiento:

Scrum propone una serie de reuniones que definen el seguimiento del proyecto en sus diferentes fases. Cada una tiene unas características y unos objetivos particulares:

1. Sprint planning meeting: se trata de la reunión inicial de cada iteración, en la que se planean las tareas a completar en la misma.
2. Daily Scrum meeting: una reunión diaria, corta y dinámica en la que se detalla el progreso desde el día anterior y lo planificado para el día actual.
3. Sprint review: la reunión de fin de iteración, en la que se recogen los resultados de la misma y se comparan con los objetivos establecidos. Puede servir para revisar las estimaciones de Historias o influenciar la siguiente iteración.
4. Sprint retrospective: segunda reunión mantenida al final de la iteración, pero cuyo objetivo no es analizar el "qué" se ha conseguido sino el "cómo". Se analizan las interacciones del equipo, la comunicación y el estado general del proyecto. Sirve para mejorar el proceso global.

Se ha escogido Scrum como metodología base debido a su definición clara pero sencilla de los procesos y documentos, y por su fácil adaptación al contexto particular. En el siguiente apartado se detalla la metodología final utilizada para este proyecto que estuvo altamente influenciada por Scrum. Toda la información acerca de Scrum se puede consultar en el documento Scrum Guide ([scrum.org](https://www.scrum.org)).

3.2.1.3 Metodología definitiva

Una vez escogida la metodología Scrum como base para el proyecto se realizaron algunas adaptaciones para adecuarla al caso particular de un equipo efectivo de dos personas (con un único desarrollador). A la hora de adaptar los roles propuestos se ha realizado la siguiente asignación:

- **Tutor del Proyecto**: ha tomado el rol de Product Owner (aquel que especifica el producto y al que se le entrega), además de un rol de mánager general durante el desarrollo participando en las reuniones de seguimiento y control.
- **Autor del Proyecto**: ha tomado el rol de desarrollador y mánager técnico del proyecto. El rol de mánager ha sido principalmente durante la especificación dónde se centró en mantener los requisitos técnicamente viables.

También se ha optado por transformar la definición de la iteración o sprint como un periodo de tiempo fijo a un periodo basado en hitos. Debido a que no se podía dedicar necesariamente el mismo tiempo cada semana a la tarea resultaba más adecuado definir cada iteración en base a un avance significativo y demostrable del proyecto. Se comprobó que en la práctica resultaron iteraciones desde 2 hasta 6 semanas.

Una vez adaptados los roles se unificaron los documentos de la metodología (Product Backlog y Sprint Backlog) en un mismo documento en formato de tabla. En el primer apartado se definieron las Historias de Usuario con un simple título e identificador. A continuación cada Historia fue definida con más detalle incluyendo una descripción y un caso de uso. En todo momento se mantiene un lenguaje que resulta familiar a cualquier persona no-técnica (como sería el caso del cliente).

Una vez especificadas las historias se pasó a su priorización en base a tres criterios:

1. **Riesgo:** representa la incertidumbre en cuanto al esfuerzo requerido para implementar una historia, la pobreza de la especificación o la utilización de técnicas, herramientas o librerías desconocidas. Es un criterio definido por el equipo técnico del proyecto, en este caso el desarrollador.
2. **Importancia para la arquitectura del sistema:** representa el grado de dependencia del sistema en el requisito, o su necesidad para que el sistema funcione de manera global. Es el segundo criterio definido por el equipo técnico, o en este caso el desarrollador.
3. **Criticidad para el usuario:** utilizado para definir la importancia de una historia para el producto final desde el punto de vista del cliente. Este criterio es definido por el Product Owner.

Desde el punto de vista del desarrollador cada historia se prioriza en base al riesgo de que se complete como se desea (Riesgo) y su impacto en otras historias si se retrasa (Importancia para la Arquitectura). El cliente contempla otros objetivos a la hora de definir la criticidad de cada historia, como la utilidad de la misma para la mayoría de los usuarios, la utilidad para un grupo de usuarios importantes o la cohesión de la historia con las demás.

Una vez definidos los criterios se deben asignar valores a cada uno para cada historia, y posteriormente reunir esos valores para obtener la prioridad global de la historia. Para los criterios definidos por el equipo técnico (Riesgo e Importancia para la Arquitectura) se establecen valores entre 0 y 3, de menor a mayor riesgo o importancia respectivamente. En el caso del criterio de criticidad para el usuario se ha optado por la valoración mediante las reglas MoSCoW, que define los siguientes valores para cada criterio más fáciles de entender:

1. **Must Have:** historia fundamental para el cliente. Se corresponde con un valor numérico de 3.
2. **Should Have:** historia importante para el cliente. Se corresponde con un valor numérico de 2.
3. **Could Have:** historia que podría quedarse fuera si no hay más remedio. Se corresponde con un valor numérico de 1.
4. **Won't Have:** historias que son valoradas pero pueden esperar a actualizaciones futuras. Se corresponde con un valor numérico de 0.

Una vez valorado cada criterio se reúnen en un único valor mediante una suma ponderada, en la que se da mayor peso a la criticidad para el usuario. La fórmula es la siguiente:

$$\text{Prioridad Final} = [\text{Riesgo}] + [\text{Importancia}] \times 2 + [\text{Criticidad}] \times 3$$

Esta fórmula produce valores desde 0 hasta 18 para cada historia de usuario, y en base a esos valores quedan priorizadas en orden de ejecución las mismas. Se ha optado por suprimir el proceso de estimación de esfuerzo de cada historia ya que al tener un único desarrollador e iteraciones flexibles no es necesario dividir el trabajo en base al esfuerzo estimado. El esfuerzo de las historias se suele calcular como el número de horas necesarias para implementarla considerando las estimaciones de todos los desarrolladores involucrados.

El último paso de la documentación conllevó la división de cada historia de usuario en tareas individuales. Estas tareas fueron definidas por el desarrollador de tal manera que fueran equivalentes a un esfuerzo de entre 2 y 6 horas aproximadamente. Las tareas sirven como guía de trabajo a la vez que sirven de seguimiento del proyecto, ya que analizando el número de tareas completadas se puede estimar el grado de avance total. En el Anexo B del documento se puede consultar toda la documentación generada y asociada a la metodología de desarrollo.

A la hora de realizar el seguimiento del proyecto se han simplificado las reuniones mantenidas. La reunión de Sprint Planning no era aplicable al caso al

existir un único desarrollador, por lo que simplemente se han ido completando los requisitos en orden de prioridad. Las reuniones de Daily Scrum y Sprint Review se han agrupado en una única reunión que se ha mantenido al final de las iteraciones (de duración entre 2 y 6 semanas). En esta reunión se presentaba la funcionalidad implementada desde la última reunión, se discutían cambios o adaptaciones a los requisitos y se proyectaba la siguiente fase de desarrollo. Estas reuniones se mantenían en persona, con una estructura informal y una duración de entre 30 y 60 minutos. Finalmente la reunión Sprint Review no se ha mantenido explícitamente, aunque durante el final de las reuniones de seguimiento se ha dedicado una parte del tiempo a una serie de objetivos similares. Se analizaba el progreso global del proyecto, la efectividad de la metodología y las tareas restantes (incluyendo aquellas ajenas al desarrollo como tal, como la evaluación o la documentación del proyecto).

Siguiendo la metodología descrita se han llegado a cumplir los objetivos de desarrollo del proyecto de manera efectiva y flexible. En el caso de tratarse de un proyecto de mayor envergadura o un equipo más amplio se hubiera aproximado más a lo descrito en la metodología Scrum.

3.2.2 Implementación

La creación de una prueba de concepto como solución al problema propuesto ha conllevado la programación de un prototipo funcional: una aplicación en la plataforma iOS. Este aspecto del proyecto se ha tratado con la misma importancia que los demás, ya que es una parte integral de la carrera de Informática. En los siguientes apartados se presentan los aspectos más destacados del desarrollo de la aplicación.

3.2.2.1 Arquitectura iOS: Modelo-Vista-Controlador

A la hora de comenzar un proyecto de programación es esencial establecer de antemano la arquitectura general del sistema que se va a implementar. Esto puede ir desde una definición en términos generales de la agrupación de módulos y su funcionalidad, hasta la especificación concreta de clases y su contenido.

La plataforma iOS se basa en el lenguaje de programación orientado a objetos *Objective-C*, y proporciona un SDK (Kit de Desarrollo de Software) con una serie de clases y paquetes existentes a disposición del desarrollador. El SDK se basa en la arquitectura llamada MVC (Modelo-Vista-Controlador), que se puede ver representado esquemáticamente en la siguiente figura:

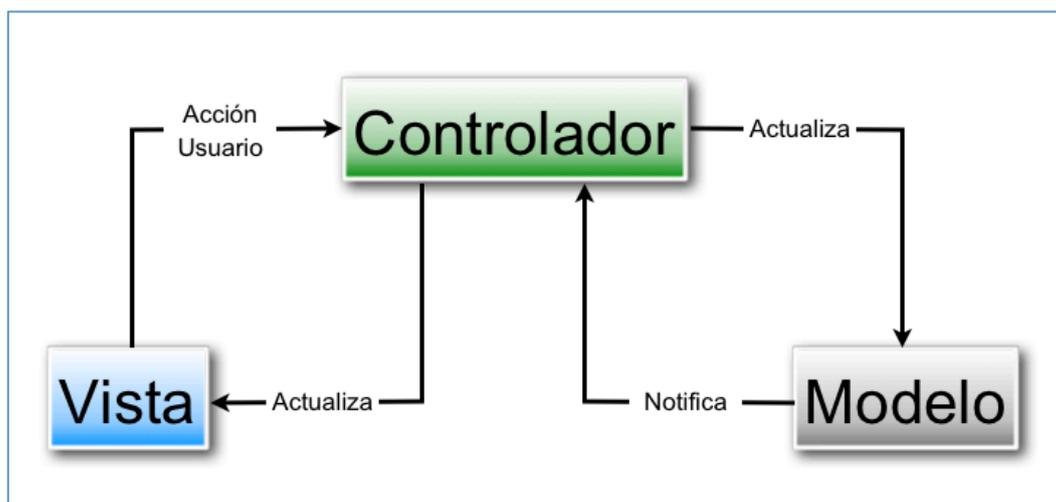


Figura 21: Arquitectura MVC

Esta arquitectura aboga por una separación de la funcionalidad del código en tres dimensiones distintas:

1. **Vista:** todo aquello que se muestra o con lo que interacciona el usuario de la aplicación. Suele relacionarse con la interfaz gráfica de la aplicación, desde las ventanas hasta los controles como botones o listas.
2. **Modelo:** todo aquello relacionado con información almacenada o tratada por la aplicación. Incluyen tanto el acceso, modificación y recuperación de datos como su manipulación antes de ser presentados. Un modelo puede basarse en algo tan sencillo como estructuras de datos o clases serializadas hasta bases de datos relacionales.
3. **Controlador:** todo aquello que contiene la lógica de la aplicación, y que además sirve de puente entre las vistas y los modelos. Desde las vistas recibe acciones o eventos, y responde con actualizaciones en la interfaz. Los modelos pueden notificar de cambios al controlador, que a su vez puede manipular los datos que contienen.

La clave de esta arquitectura es la separación de la lógica de la aplicación de los datos y de la interfaz de usuario. La gran mayoría de las clases proporcionadas en el SDK de iOS siguen esta arquitectura, y se recomienda mantenerla a la hora de desarrollar aplicaciones ya que fomenta la reutilización del código, la alta cohesión y el bajo acoplamiento.

A la hora de desarrollar la aplicación para este proyecto se ha respetado esta arquitectura. En los siguientes diagramas se muestran de manera esquemática las clases principales de la aplicación. Se puede observar que la mayoría son controladores que tienen automáticamente asociados una vista (no representadas en el diagrama), y cuyo nombre termina con "ViewController". Aquellas clases cuyo nombre sólo termina en "View" son del tipo Vista. Las clases sin "View" y "ViewController" son aquellas de tipo Modelo siendo las principales las relacionadas con las alertas CAP, su obtención y filtrado.

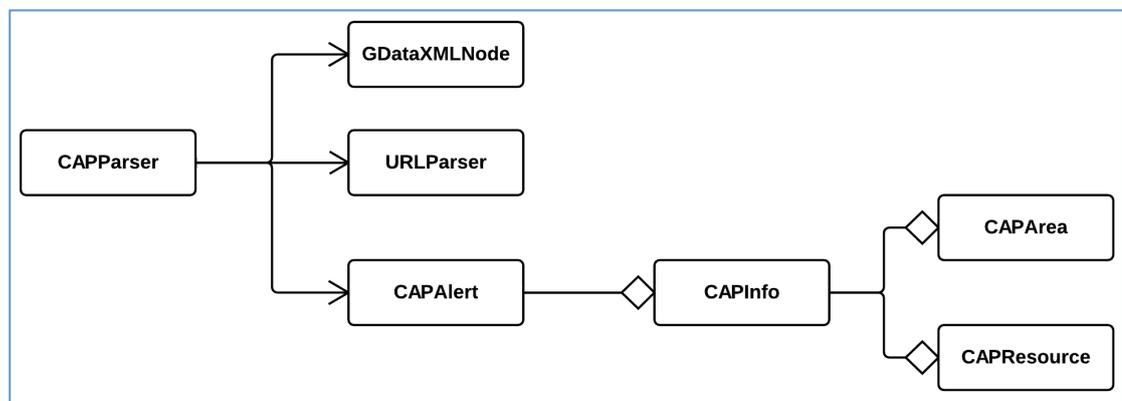


Figura 22: Diagrama de clases principales de modelo

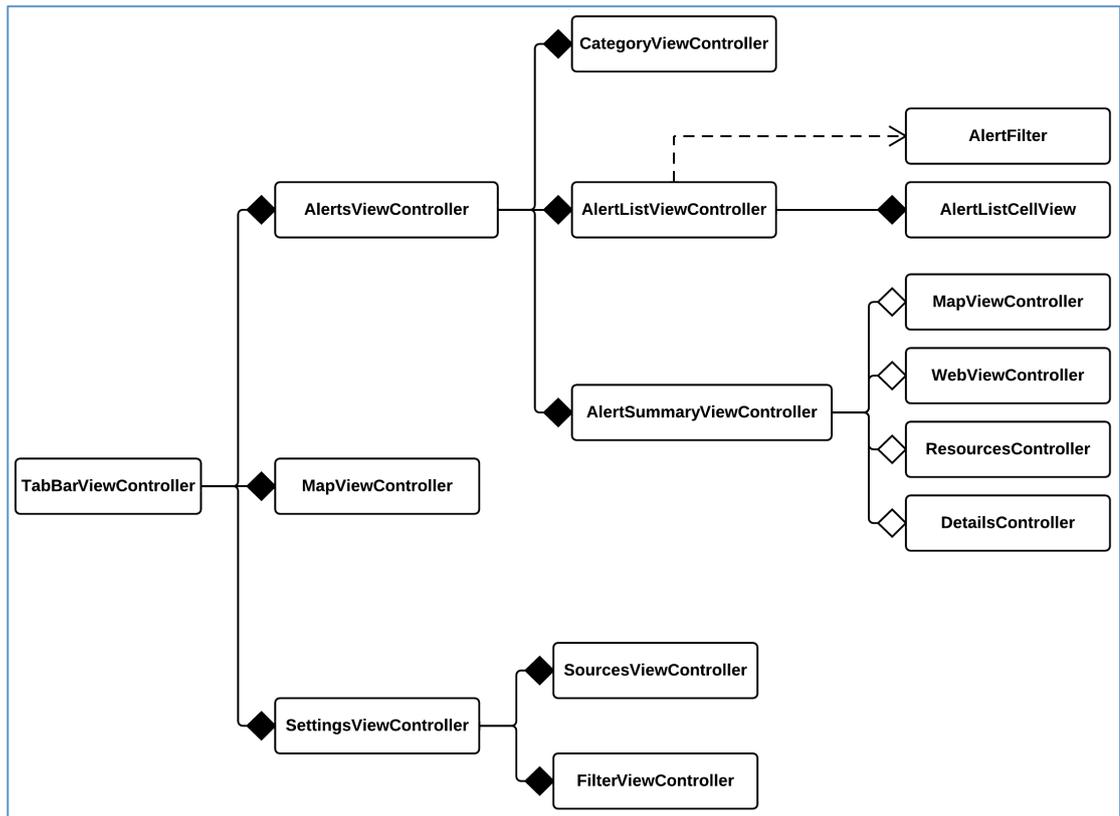


Figura 23: Diagrama de clases principales de controlador

En la siguiente tabla se detallan las clases más importantes del proyecto junto con una breve descripción de su utilización y la reutilización posible de la misma:

Clase	Tipo	Descripción	Reutilización posible
CAPAlert CAPInfo CAPArea CAPResource	Modelo	Representan una alerta CAP con todos sus campos. Incluyen métodos para la comparación, la serialización y el cálculo de la importancia.	Cualquier aplicación que trate alertas CAP, tan sólo es necesario cambiar el algoritmo de importancia si fuera diferente.
AlertFilter	Modelo	Representa la clase modelo principal de la aplicación, aloja las alertas almacenadas por la misma además de la lógica de filtrado. Las alertas se obtienen de manera dinámica con el filtrado y la ordenación deseada. En cualquier momento se puede extender esta clase para modificar el método de almacenamiento o filtrado sin tener ningún efecto sobre el resto del código.	Reutilizable por cualquier aplicación que desee filtrar alertas de manera dinámica, la lógica de filtrado queda totalmente separada del resto de la aplicación reduciendo el acoplamiento.
AlertSummary	Vista/Cont.	Conjunto Vista-Controlador encargado de representar la vista resumen de una alerta - una de las más importantes de la aplicación.	Esta Vista-Controlador sólo depende de obtener una alerta CAPAlert, y puede integrarse en cualquier otra aplicación con relativa facilidad.
AlertListCell	Vista	Vista utilizada dentro del listado de alertas para representar cada alerta. Consta de un diseño a medida para representar la información más representativa de la alerta.	Esta vista se puede incorporar a cualquier listado.
MapView	Vista/Cont.	Conjunto Vista-Controlador que permite visualizar una o varias alertas sobre un mapa en base a su información de localización.	Poca reutilización posible ya que es un ajuste de una clase genérica a las necesidades particulares de la aplicación.
GDataXMLNode	Modelo	Clase modelo perteneciente a un framework externo de software libre para la gestión de datos en formato XML - creado por Google	Alta reutilización, al pertenecer a un framework de uso libre.
URLParser	Modelo	Clase encargada de extraer las alertas o URL's de las alertas de las diferentes fuentes existentes según el formato particular de cada una.	Reutilización posible si se desean tratar las mismas fuentes de alertas.
CAPParser	Modelo	Clase encargada de generar un conjunto de alertas en base a las fuentes, y en forma de objetos CAPAlert. Se apoya en las clases de URLParser y GDataXMLNode.	Menos reutilización posible debido a su alta dependencia en otras clases.

Tabla 4: Clases más importantes de la aplicación

3.2.2.2 Interfaces y Vistas

En este apartado se presentan las interfaces de la aplicación creada que son el resultado del proceso de análisis, de los requisitos generados y de los cambios producidos al evaluar e iterar progresivamente. Para cada interfaz o vista de la aplicación se incluye una descripción breve de su funcionalidad además de un resumen de los detalles destacados y las decisiones de diseño involucradas.

La siguiente interfaz es la primera que se presenta al usuario al abrir la aplicación. En ella se pueden observar los primeros aspectos importantes de la aplicación. En la parte inferior existe una fila de tres botones que alternan entre pestañas de la aplicación (*Tab Bar*), esta fila está siempre presente en la interfaz. Los iconos usados se han seleccionado para transmitir con claridad el contenido de cada pestaña: un megáfono para representar las alertas, un mundo para representar la vista en mapa y un engranaje para representa los ajustes de la aplicación (icono utilizado de manera estándar en iOS). En la parte superior se pueden observar varios botones: un control segmentado para permitir la ordenación de la lista por fecha o importancia, un botón de recarga de alertas con un icono estándar del sistema y un botón de navegación hacia atrás para acceder a la vista de categorías. No se presenta ninguna alerta inicialmente ya que no se han cargado, aunque en sucesivas aperturas aparecerán las alertas almacenadas desde el último uso de la aplicación.

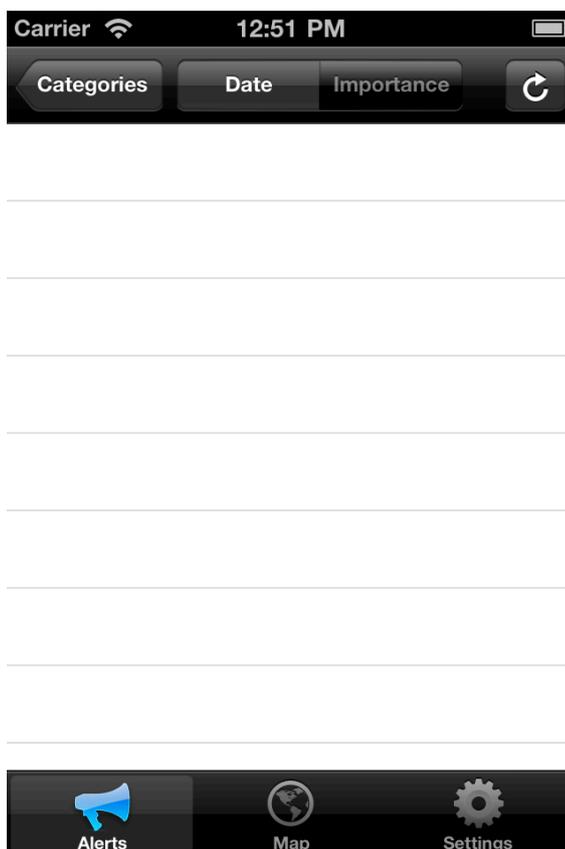


Figura 24: Listado de alertas vacío

Esta interfaz ha sufrido algunos cambios leves desde el comienzo del proyecto. Inicialmente la barra superior presentaba el aspecto siguiente:



Figura 25: Barra superior inicial del listado de alertas

Se optó por cambiar el color a negro para unificar el aspecto de la aplicación, ya que la barra inferior de pestañas debía ser de ese color, y el control segmentado cambió de funcionalidad. Inicialmente se utilizaba para alternar entre mostrar todas las alertas descargadas o sólo aquellas no caducadas, pero tras una evaluación de la interfaz se cambió para definir la ordenación del listado (por fecha o por importancia).

Para cargar las alertas de las fuentes escogidas se pulsa el botón en la parte derecha de la barra superior, y la interfaz cambia para mostrar una barra de progreso y un botón de parada. Durante la carga de alertas el listado se va poblando dinámicamente y las alertas se pueden seleccionar:

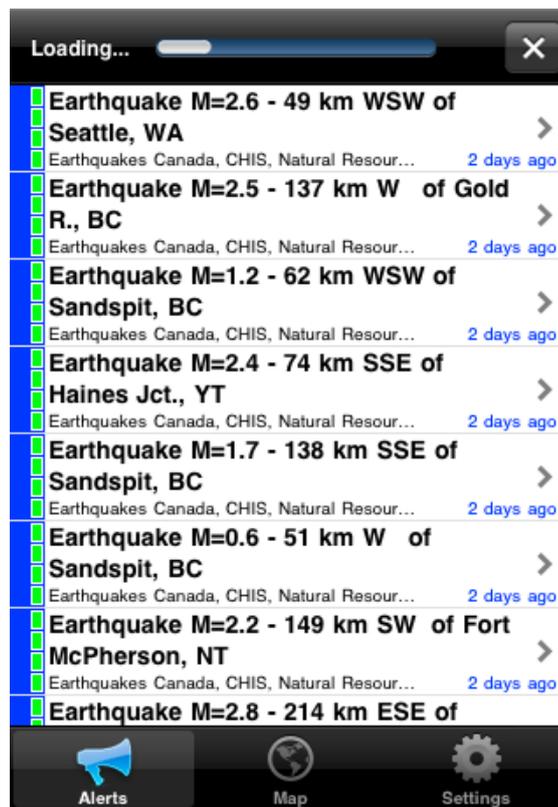


Figura 26: Listado cargando alertas

Esta interfaz sufrió una evolución desde su diseño inicial, en el que solamente se incluía un marcador de cargado genérico sin barra de progreso ni botón de parada.



Figura 27: Vista inicial de cargado de alertas

Una vez cargadas las alertas cada una se representa dentro del listado con una celda que contiene información representativa de la misma. Se incluye un título de alerta, el origen, el tiempo desde su emisión, una barra de color para representar su importancia y un indicador de nivel de certeza (de cero a cuatro barras verdes sobre el color):



Figura 28: Celda de alerta individual en el listado

Cabe destacar de esta parte de la interfaz su similitud con otros listados de mensajes presentes en la plataforma, como la lista de correos electrónicos o mensajes de texto. Presentar dos campos de la alerta con fuentes de diferente impacto y utilizar una fuente de color azul para el tiempo ayudan al usuario a familiarizarse rápidamente con la interfaz y entender su funcionalidad.

En cuanto a la selección del color de una alerta se diseñó una escala basada en dos aspectos de la alerta: su Urgencia y Severidad. Ambos campos pueden tomar 5 valores diferentes, y para cada posible combinación (25 en total) se ha establecido el color correspondiente en base a los siguientes criterios:

- La importancias tomarán uno de los siguientes valores: Muy Importante, Importante, Serio, Informativo o Desconocido.
- Si los valores de Urgencia o Severidad está en sus valores más altos, la alerta será Muy Importante
- Si la Urgencia es "Pasado" sólo podrá ser informativa la alerta.
- Se seguirá una progresión desde Muy Importante hasta Desconocido para el resto.

La siguiente tabla recoge la selección de colores según los valores de Urgencia y Severidad:

Urgency\Severity	Extreme	Severe	Moderate	Minor	Unknown
Immediate	Red	Red	Orange	Yellow	Yellow
Expected	Red	Orange	Yellow	Blue	Blue
Future	Orange	Yellow	Blue	Blue	Blue
Past	Blue	Blue	Blue	Blue	Blue
Unknown	Orange	Yellow	Yellow	Yellow	Grey

Tabla 5: Colores de nivel de importancia

Volviendo a la parte superior del listado se puede observar un botón que permite acceder a la vista de categorías:

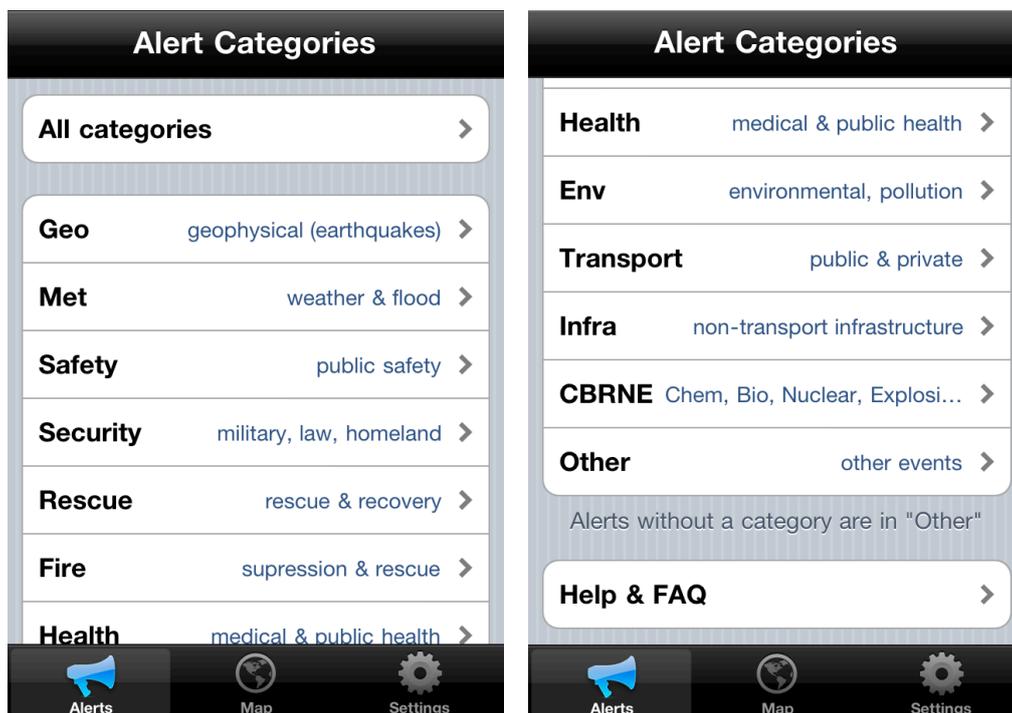


Figura 29: Vista de categorías

Desde esta vista se puede seleccionar una única categoría de alertas para mostrar, o mostrar todas las alertas. Las categorías vienen definidas por el estándar CAP con términos a veces poco descriptivos como "Infra" o "CBRNE", por lo que el listado de categorías incluye un texto breve de detalle para cada una que permite entender mejor su significado. En la parte inferior del listado se puede acceder a un apartado de ayuda y preguntas frecuentes que contiene toda la información necesaria para que un usuario conozca los detalles de la aplicación en caso de necesitarlo.

Volviendo al listado de alertas, una vez seleccionada una alerta se pasa a la vista resumen de la misma con el siguiente aspecto:

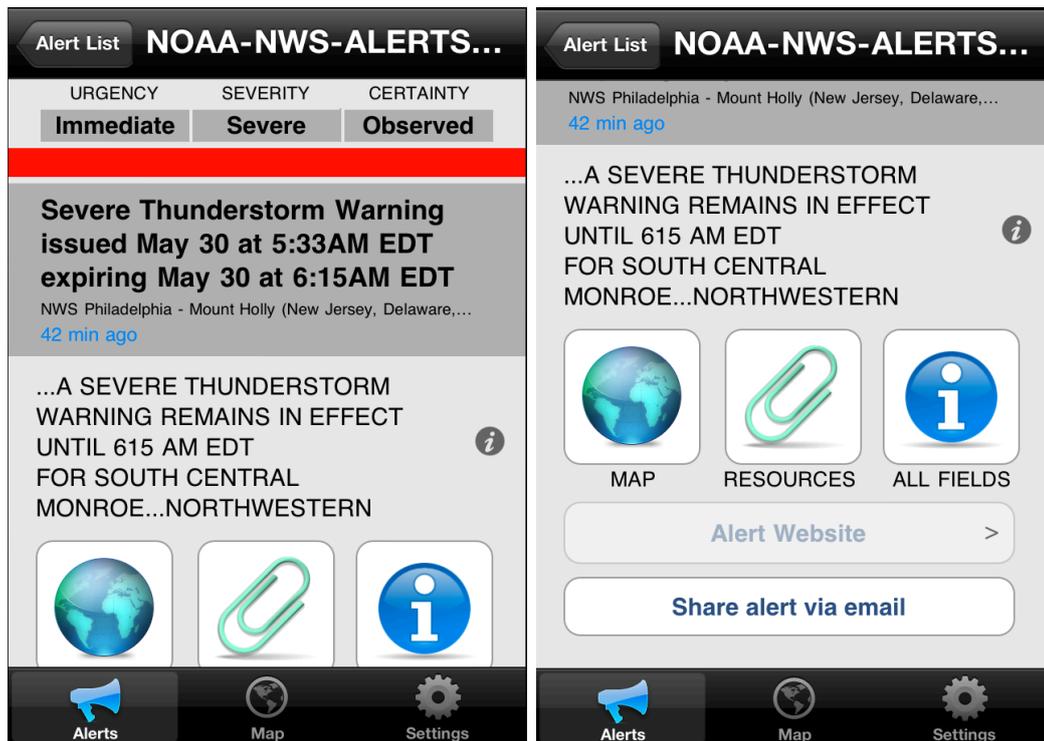


Figura 30: Vista resumen de una alerta

Empezando por la parte superior se pueden observar los valores de Urgencia, Severidad y Certeza de la alerta, además del color de nivel de importancia. A continuación se presenta de nuevo el título, el origen y la edad de la alerta (en esta vista con más espacio). Lo siguiente es la descripción de la alerta, con un botón de detalle marcado con 'i' que permite leer la descripción completa (se fija un número máximo de líneas de descripción en esta vista para mantener las proporciones fijas). Después de la descripción se pueden observar tres botones con icono: Mapa, Recursos y Todos los Campos. Al pulsar Todos los Campos se muestra el contenido íntegro de la alerta CAP recibida. Los iconos de Mapa y Recursos se presentan sin color en caso de que la alerta no contenga la información pertinente:

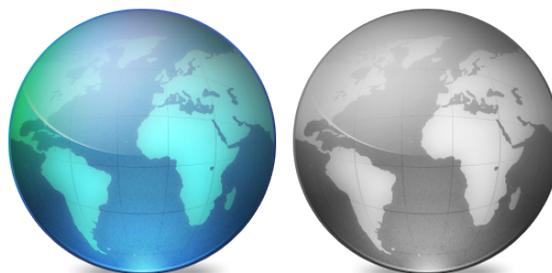


Figura 31: Icono de mapa disponible y no disponible

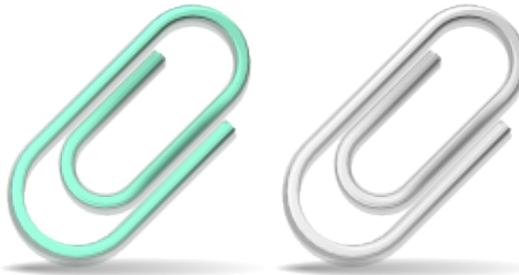


Figura 32: Icono de recursos disponibles y no disponibles

Por último la vista resumen (Figura 30) presenta un botón para acceder a la página web asociada a la alerta en caso de existir, y un botón para compartir la alerta mediante correo electrónico. En ambos casos las acciones se desarrollan dentro del ámbito de la aplicación y se vuelve a la vista resumen al finalizar.

The screenshot shows the USGS Earthquake Hazards Program website. The main heading is "earthquake.usgs.gov/eq...". The page displays information for a magnitude 2.2 earthquake in Baja California, Mexico, on Monday, May 30, 2011 at 09:08:17 UTC. The earthquake details table is as follows:

Magnitude	2.2
Date-Time	Monday, May 30, 2011 at 09:08:17 UTC Monday, May 30, 2011 at 02:08:17 AM at epicenter Time of Earthquake in other Time Zones
Location	32.477°N, 115.646°W
Depth	1.4 km (~0.9 mile) (poorly constrained)
Region	BAJA CALIFORNIA, MEXICO
Distances	26 km (16 miles) SSW (21°) from Calexico, CA 26 km (16 miles) SW (221°) from Mexicali, Baja California, Mexico 30 km (19 miles) SSW (201°) from Heber, CA 129 km (80 miles) E (92°) from Tijuana, Baja California, Mexico
Location Uncertainty	horizontal +/- 1.7 km (1.1 miles); depth +/- 2 km (1.2 miles)
Parameters	Np= 27, Dmin=32 km, Rmss=0.36 sec, Gp=234°, M-type=local magnitude (ML), Version=0
Source	California Integrated Seismic Net; USGS Caltech CGS UCB UCSD UNR
Event ID	ci14993428

Below the table, there is a section titled "Did you feel it?" with a link to report shaking and damage at your location. At the bottom of the page, there are three icons: Alerts, Map, and Settings.

Figura 33: Vista de la página web contenida en una alerta

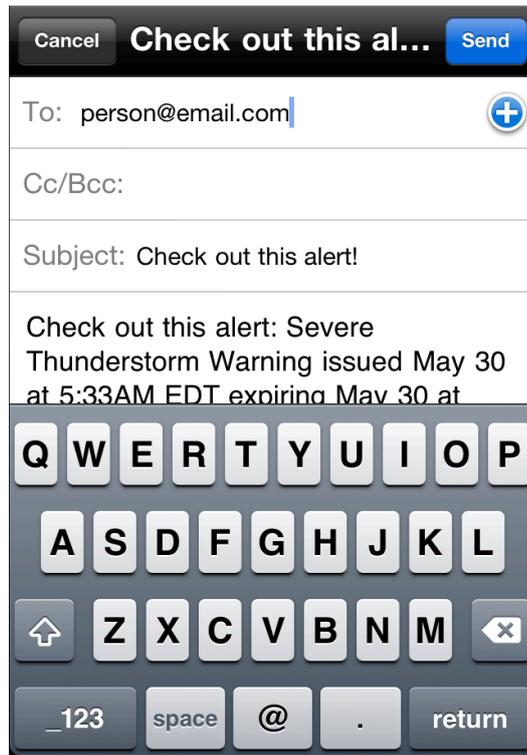


Figura 34: Vista de envío de una alerta por correo electrónico

En las siguientes figuras se muestran las vistas de Mapa, Recursos y Todos los Campos que se muestran al pulsar el botón correspondiente. En el caso de la vista en mapa, ésta se diferencia de la vista en mapa general de la aplicación (accesible desde la pestaña inferior) en que muestra la localización solamente de la alerta en cuestión y no todas las del listado.

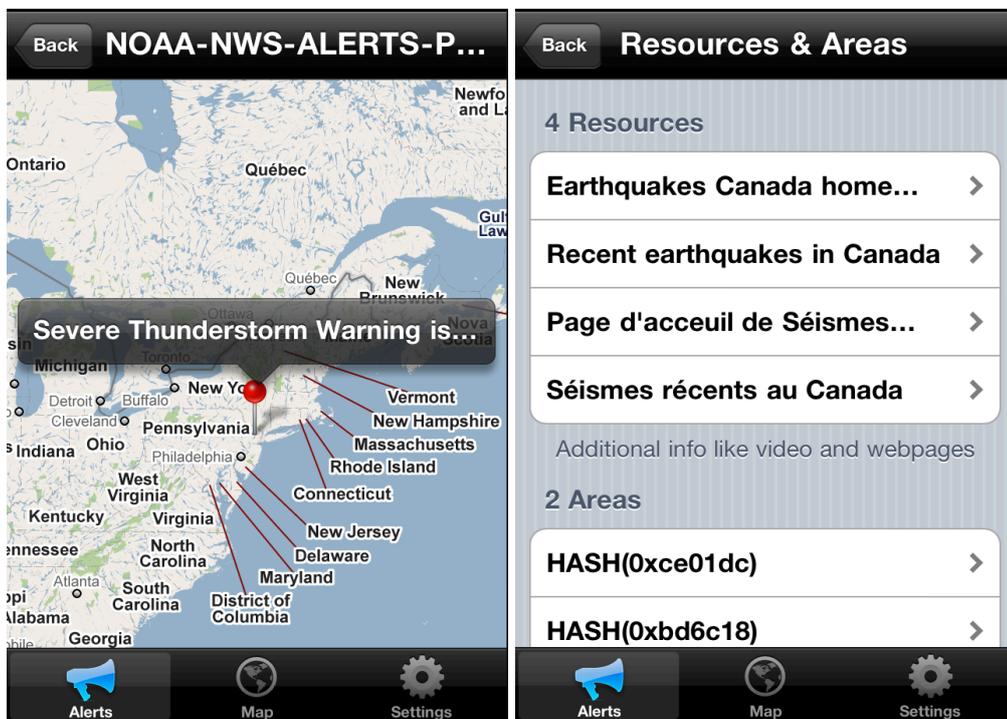


Figura 35: Vista en mapa y vista de recursos de una alerta

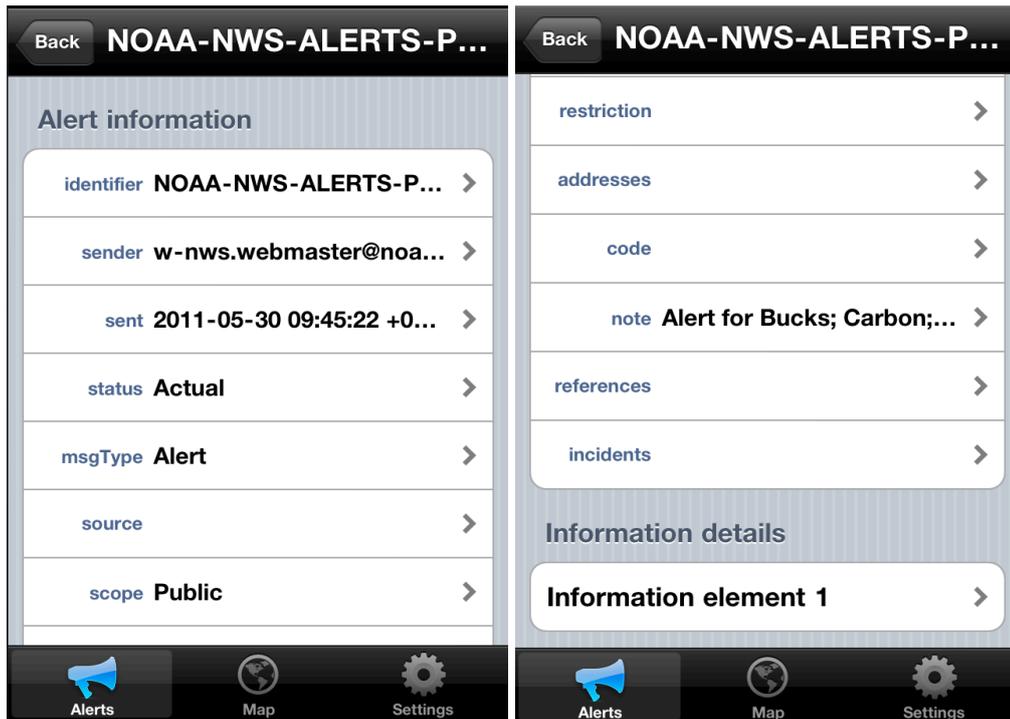


Figura 36: Vistas de todos los campos de una alerta I



Figura 37: Vistas de todos los campos de una alerta II

Cabe destacar que la vista resumen de una alerta ha sufrido una evolución importante en cuanto a su diseño. El diseño inicial se muestra a continuación:

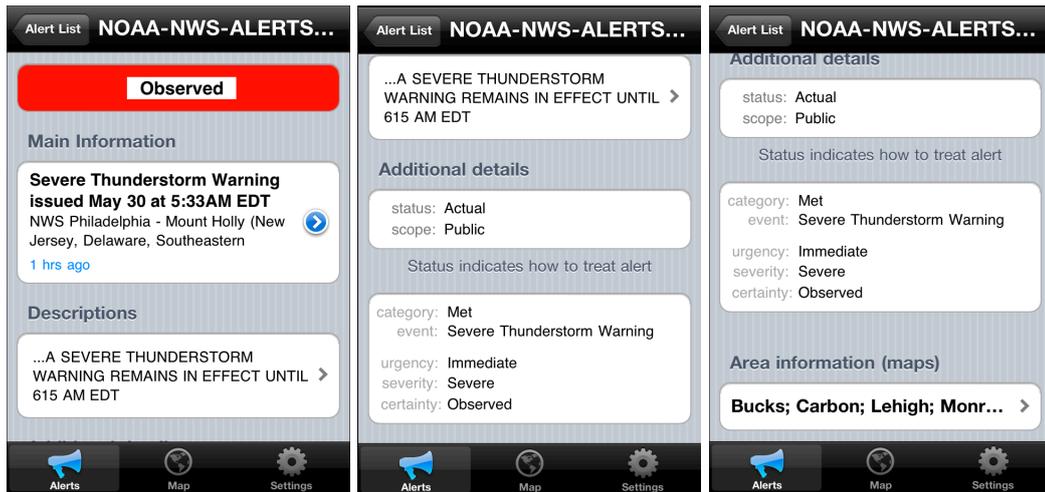


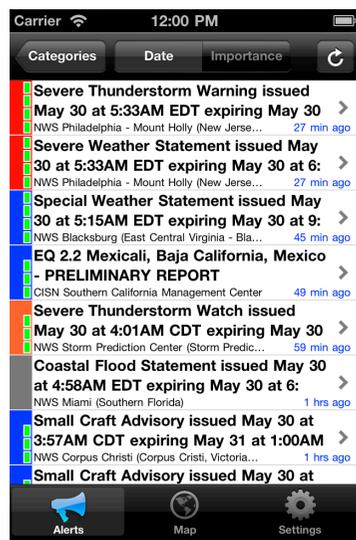
Figura 38: Vista inicial de resumen de alerta

Se puede observar que incluye todos los mismos elementos de interfaz, pero utilizando un diseño en listado levemente modificado. Se representan todos los campos del resumen, incluyendo el elemento de color, y se da acceso a los detalles de la alerta incluyendo sus recursos, su vista en mapa y todos los campos originales. Aun así el diseño desperdicia mucho espacio de la pantalla, algo a evitar en los dispositivos móviles que poseen pantallas con espacio limitado. El uso de iconos en la vista rediseñada además permite interpretar fácilmente la funcionalidad, y son visibles desde el momento en el que se abre la vista resumen permitiendo una interacción muy rápida del usuario que ya conoce la aplicación.

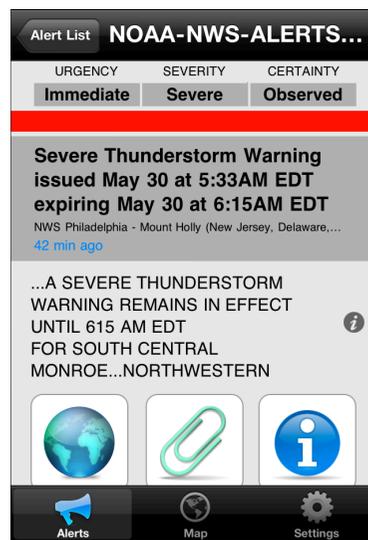
Todas las vistas de la primera pestaña de la aplicación forman parte de una estructura de vistas en formato de navegación de izquierda a derecha. Este estilo de interacción es uno de los más usados en la plataforma iOS y las plataformas móviles en general. Se pasa desde vistas más generales (a la izquierda) hasta vistas más en detalle (a la derecha). El siguiente diagrama muestra el flujo de interacción según esta estructura de navegación:



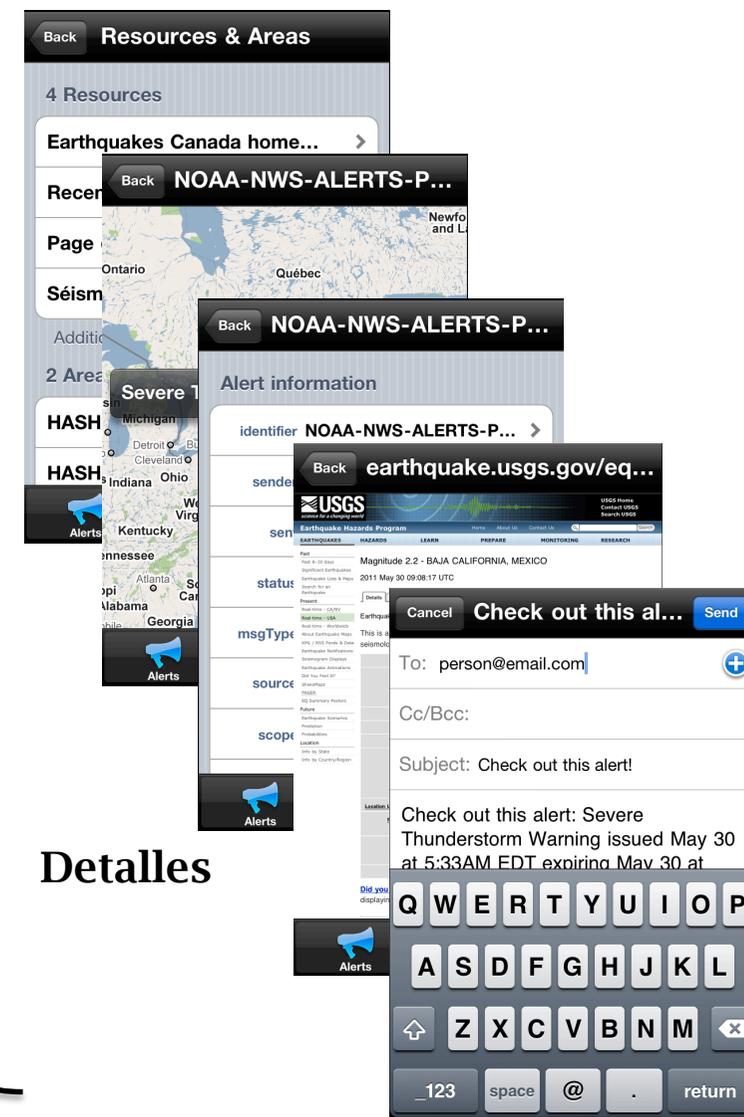
Categorías



Alertas



Una Alerta



Detalles

Figura 39: Diagrama de flujo de interacción

La siguiente vista principal de la aplicación, accesible desde la pestaña central de la barra inferior, es la vista general en mapa. En esta vista se representan todas las alertas del listado que proporcionen información de localización apropiada. Es importante destacar que cualquier filtrado que se haya aplicado al listado (por selección de categoría o en ajustes) se refleja en el mapa. Sólo se mostrarán las alertas que son visibles en ese momento en el listado.



Figura 40: Vista en mapa general

El mapa comienza en un estado de visibilidad máxima, con un nivel de detalle global que permita mostrar la gran mayoría del mundo y por tanto de las alertas. Posee un botón en la parte superior derecha para alternar entre los tres tipos de mapa disponibles (Normal, Satélite e Híbrido).



Normal



Satélite



Híbrido

Figura 41: Tres tipos de mapa

Al seleccionar el marcador de una alerta se muestra una etiqueta con el título de la misma y, a diferencia de la vista en mapa de una alerta individual, se muestra un botón de acceso a la vista resumida de la alerta:



Figura 42: Marcador de una alerta en la vista en mapa

En efecto la vista en mapa general es una alternativa a la visualización en listado. Mientras que en el listado se distinguen las alertas por campos textuales como título y fuente, en la vista en mapa el aspecto clave es la localización. Esto permite al usuario escoger la manera de búsqueda de alertas más adecuada a sus necesidades.

Por último la aplicación posee una vista de ajustes a la que se accede mediante la última pestaña de la barra inferior. Esta vista permite escoger las fuentes de alertas deseadas, ajustar el tipo de filtrado de alertas y habilitar/deshabilitar un límite máximo de alertas para descargar de cada fuente para reducir el tiempo de carga (especialmente apropiado en caso de conexiones lentas o mala cobertura).

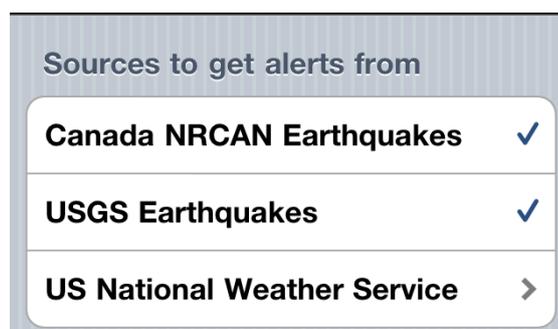


Figura 43: Ajustes de selección de fuentes

Las fuentes actuales incluyen terremotos de NRCAN y USGS, y alertas de temporal de NWS (pudiendo seleccionar estados individuales o todas las zonas).

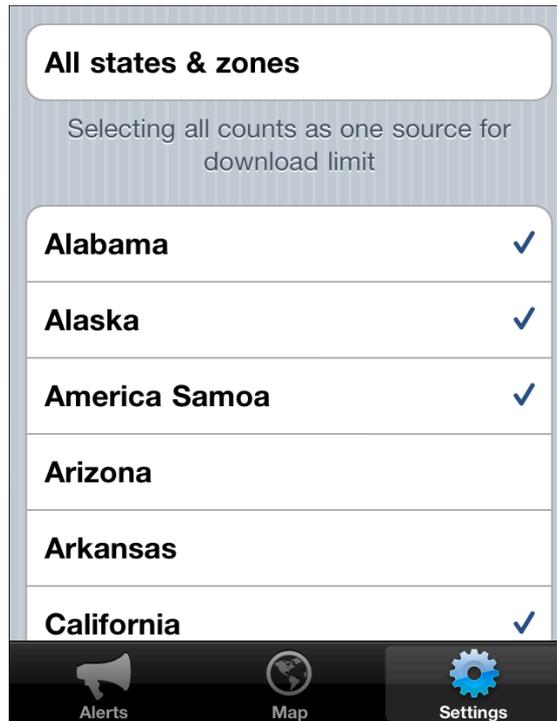


Figura 44: Selección de fuentes individuales de NWS

Existe la opción de ocultar alertas caducadas. Cada alerta puede llevar una fecha de caducidad indicada, pero en caso de no llevarla se considera como fecha de caducidad 7 días después de la fecha efectiva de emisión de la alerta.

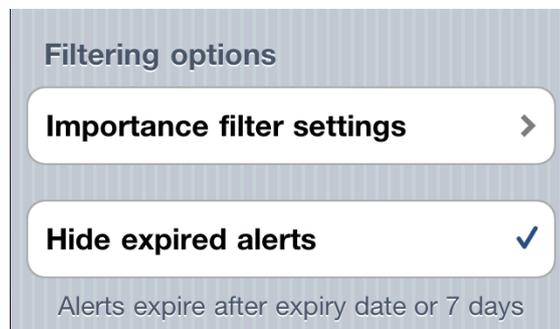


Figura 45: Ajustes de filtrado y ocultación de alertas caducadas

Se pueden escoger los valores de Urgencia, Severidad y Certeza que debe poseer una alerta para mostrarse. Debajo de cada apartado de ajustes se incluye una explicación breve de lo que representa cada valor.

Urgency levels to display	Severity levels to display	Certainty levels to display
Immediate ✓	Extreme ✓	Observed (100%) ✓
Expected ✓	Severe ✓	Likely (more than 50%) ✓
Future ✓	Moderate ✓	Possible (less than 50%) ✓
Past ✓	Minor ✓	Unlikely (not expected) ✓
Unknown ✓	Unknown ✓	Unknown ✓

When responsive action must be taken How extensive the effects are How certain the event is

Figura 46: Ajustes de filtrado de Urgencia, Severidad y Certeza

Por último se puede establecer un límite de número de alertas a descargar de cada fuente (15 alertas más recientes) para reducir el tiempo de obtención de las mismas. La naturaleza de las conexiones móviles puede llevar a situaciones con velocidades bajas en las cuales no es apropiado que se descargue un número elevado de alertas.

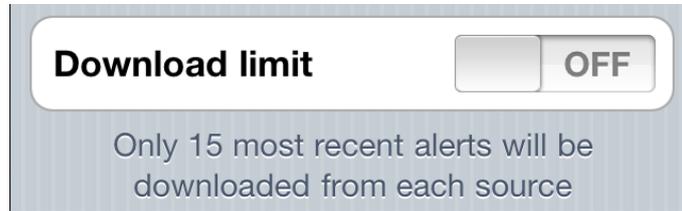


Figura 47: Ajuste de límite de descarga

3.3 Evaluación

Una parte indispensable de cualquier proceso de implementación de una aplicación informática es la evaluación de la misma. La evaluación permite estudiar el éxito del proyecto en varias dimensiones, desde el cumplimiento de la especificación hasta las opiniones de expertos y usuarios sobre su utilidad. En los siguientes apartados se describen las distintas evaluaciones realizadas durante el desarrollo del prototipo.

3.3.1 Evaluación de requisitos

El primer aspecto evaluable del prototipo es el cumplimiento de los requisitos o historias de usuario definidas en la fase de análisis (y actualizadas en cada iteración). Aunque se cumpla estrictamente la especificación propuesta es posible que el resultado final desvele deficiencias en esta especificación, y por ello se considere que no se ha cumplido. En definitiva, la prueba final del cumplimiento de la especificación es la aprobación o rechazo del cliente o interesado en el producto final (en el caso de la metodología de este proyecto el rol de Product Owner). Es por este aspecto, entre otros, por lo que se ha optado por una metodología iterativa e incremental que proporciona un nivel de flexibilidad y adaptabilidad suficiente para subsanar cualquier resultado inesperado.

Durante el proyecto se mantuvieron reuniones de seguimiento al final de cada iteración, en las cuales una parte del tiempo se dedicó a la entrega de la funcionalidad implementada hasta el momento. Esta funcionalidad estaba ligada a una serie de requisitos de la especificación (recogidos en el Product Backlog), que se recordaban a la vez que se presentaba la funcionalidad resultante. Tras la presentación el que tomaría el rol de Product Owner puede tomar una de las siguientes decisiones:

1. **Aprobar la funcionalidad:** se considera que la funcionalidad implementada refleja fielmente el requisito correspondiente y se cumple con la expectativa
2. **Rechazar la funcionalidad:** se considera que la funcionalidad no es la adecuada para cumplir con el requisito en correspondiente. Se debe actualizar el requisito adecuadamente con la nueva especificación y volver a incluirlo en la siguiente iteración.

Además de la aprobación o el rechazo existe la posibilidad de que surjan nuevos requisitos que se estudian y se incorporan al conjunto de requisitos. Este conjunto nunca se considera firme y cerrado, puede modificarse en cualquier momento.

3.3.2 Evaluación del diseño con un experto

NOTA: esta reunión se mantuvo durante el proceso de desarrollo, por lo que los diseños mostrados pueden variar de los diseños finalmente escogidos.

Después de tomar algunas de las decisiones de diseño descritas en el apartado 3.1 se plantearon una variedad de opciones de interfaz para conseguir el efecto deseado. Para ayudar en la evaluación de las diferentes posibilidades, además de obtener una opinión experta sobre la interfaz completa de la aplicación, se preparó una reunión con un experto en el ámbito de la Interacción Persona-Computador. Se trata de un profesor titular de la Universidad Carlos III de Madrid perteneciente al laboratorio de sistemas interactivos (DEI) del departamento de Informática. Posee una amplia experiencia académica y profesional en el ámbito de la interacción persona-ordenador, y actualmente imparte cursos de interfaces de usuario y contenidos multimedia/hipermedia en las titulaciones del departamento de Informática.

Antes de la reunión se facilitó al experto un documento con una descripción del proyecto, los objetivos de la reunión y del proceso de evaluación que se iba a llevar a cabo. La primera parte de dicho documento se transcribe a continuación:

Contexto

Actualmente se está desarrollando una aplicación para la plataforma iOS (específicamente iPhone) que sirva como lector de alertas de emergencia con formato CAP. El objetivo es desarrollar una aplicación que permita extraer la mayor cantidad de información útil de las alertas y permita la comprensión efectiva de esa información. Además de simplemente mostrar los campos de una alerta se están estudiando diferentes procesamientos de la información contenida para transmitirla de una manera más intuitiva al usuario. Algunos ejemplos de este procesamiento son:

- Interpretar los grados de severidad, urgencia y certeza de una alerta utilizando escalas de colores, iconos o elementos gráficos*
- Filtrar las alertas por diferentes campos como categorías, niveles de urgencia, fechas de efectividad o caducidad, etc.*
- Resumir los aspectos más relevantes de una alerta para su rápida comprensión*
- Mostrar información de localización y extensión de alertas sobre un mapa*

La aplicación actualmente se encuentra en fase de desarrollo. Se está desarrollando siguiendo los principios de las metodologías ágiles, por lo que existe un prototipo que se va evaluando y actualizando durante el proceso. Parte de esta evaluación debe tratar las interfaces gráficas de la aplicación y su diseño.

Objetivos de la reunión

En esta reunión se pretende presentar el prototipo en su estado actual para evaluar el diseño de algunas de las interfaces, además de consultar sobre diferentes propuestas de diseño que se han preparado pero que aún no están incluidas en el prototipo.

Las dos grandes preguntas en las que se centra la evaluación son:

- 1. Desde el punto de vista de un usuario, ¿es intuitiva y efectiva la interfaz?*
- 2. Dada la información que se desea transmitir (alerta CAP), ¿se ha procesado, agrupado y mostrado de la manera más eficaz?*

Se buscan tanto comentarios sobre los aspectos positivos como críticas y propuestas de cambio o mejora.

La reunión se documentó de dos maneras: grabación de audio en la sala (previo consentimiento de los participantes) y anotación por escrito de respuestas, comentarios y observaciones del experto.

Se agrupó la reunión en dos fases. En la primera fase se mostró la aplicación describiendo la funcionalidad brevemente y permitiendo la interacción del experto con la misma. Durante esta fase se realizaron las siguientes preguntas al experto:

- ¿Es adecuada la interfaz de barra de pestañas para agrupar las diferentes vistas de la aplicación?
- ¿Es clara la interfaz del listado de alertas, echa en falta alguna información? (No considerar elemento de color ni urgencia, severidad, certeza)
- ¿Se comprende bien el uso y efecto del botón segmentado de “*All/Not-Expired*”?
- ¿Queda claro el uso del selector de categorías del listado?
- ¿Queda claro la navegación jerárquica Categorías>Listado>Resumen de Alerta?
- ¿Es clara la interfaz de resumen de la alerta, echa en falta alguna información?
- ¿Se comprende la diferencia entre el resumen y todos los campos originales de la alerta?
- ¿El comportamiento de la apertura de la página web de la alerta fuera de la aplicación le parece correcto?
- ¿El comportamiento de la selección de un recurso le parece adecuado?
- ¿Se comprenden bien las opciones disponibles en los ajustes?
- ¿Queda claro el efecto de los ajustes sobre el listado de alertas?
- ¿Qué otra funcionalidad le gustaría tener en la aplicación?

La gran mayoría de las consultas obtuvieron una respuesta positiva. Aun así, cabe destacar los siguientes comentarios del experto:

- Se propuso cambiar la funcionalidad del botón segmentado del listado de alertas (en la barra superior). En el momento de la reunión el botón permitía ocultar o mostrar alertas caducadas, y la propuesta fue cambiar la funcionalidad a la selección de la ordenación de la lista. Esta ordenación podría ser por fecha o por importancia de la alerta (ligada al color).
- Se debería considerar siempre el contraste de los elementos de la interfaz a la hora de tratar con colores, o texto sobre fondo de color para asegurar una buena legibilidad. Incluso podría considerarse la inclusión de opciones de inversión de contraste o ajuste de colores para distintas situaciones (visibilidad nocturna) o usuarios con necesidades especiales (daltonismo).
- No pareció acertado utilizar iconos para representar los diferentes tipos de recursos que acompañan una alerta. Se recomendó simplemente un valor textual del tipo basado en el MIME-Type.
- No pareció correcto el comportamiento de la aplicación a la hora de abrir una página web relacionada con la alerta. Si es posible se debería evitar interrumpir el flujo de interacción y que el usuario tenga que salir de la aplicación al menos que sea totalmente imprescindible.

Tras esta fase se pasó a una segunda fase en la que se realizaron consultas sobre diferentes propuestas para aspectos de la interfaz y se pedía la opinión del experto y sus preferencias. Se presentaron una serie de diapositivas con las propuestas y se anotaron las respuestas. A continuación se muestran las diapositivas junto con las preguntas realizadas y las respuestas obtenidas.

Aquí se presentó la propuesta para usar la escala de colores mencionada anteriormente para representar la importancia de una alerta dada por su Severidad y Urgencia (**Figura 48**). Aparte de mostrar la escala de colores y su significado, se mostró una tabla en la que se recogían las distintas combinaciones posibles de Severidad y Urgencia y el color asignado a cada una. Se preguntó si ambas parecían adecuadas y el experto contestó afirmativamente. Reafirmó la selección de la escala de colores y le pareció correcta la asignación a las diferentes combinaciones de Severidad y Urgencia.

Propuesta: colores para severidad y urgencia

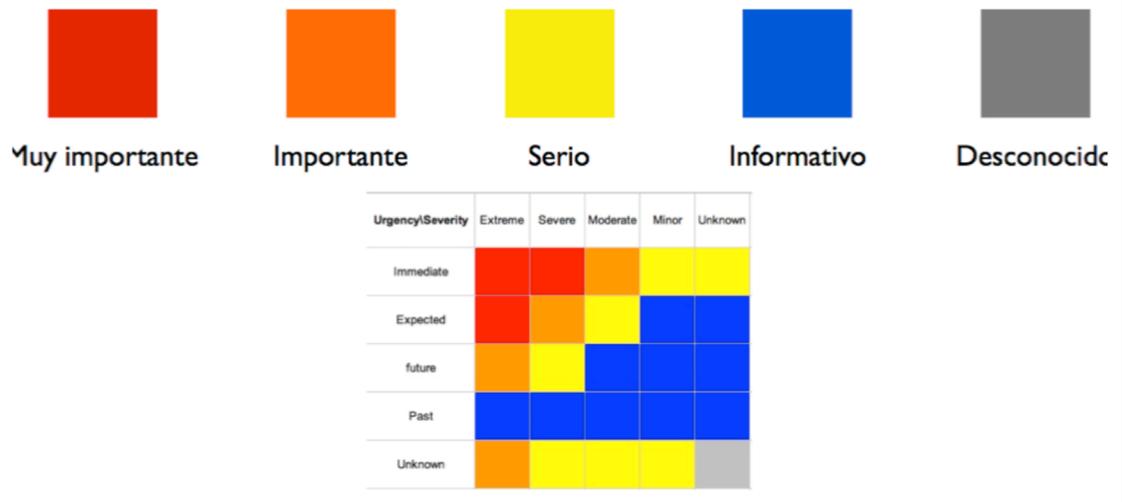
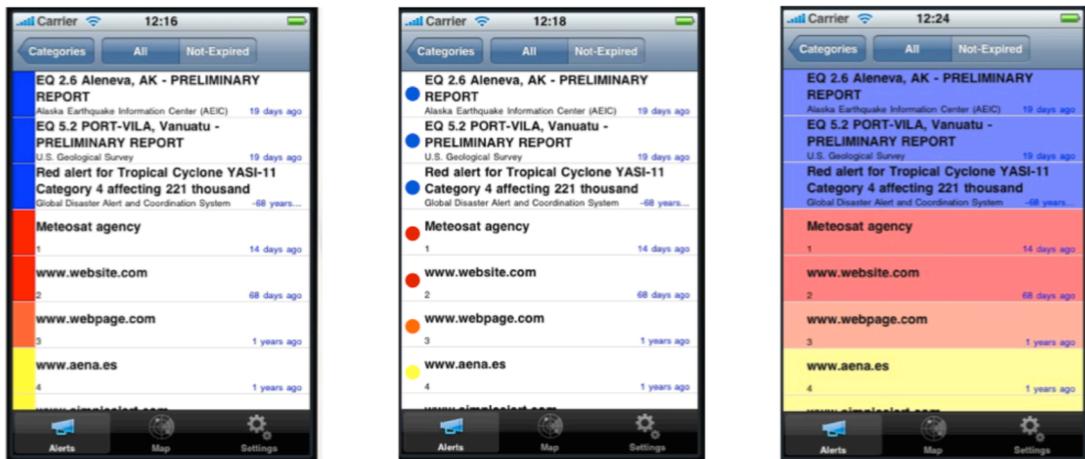


Figura 48: Propuesta de colores para severidad y urgencia

En la siguiente diapositiva (**Figura 49**) se mostraron varias opciones para mostrar el color de una alerta dentro del listado de alertas. El experto seleccionó como opción preferida la primera (marca de color en bloque) por resultar claro y en acuerdo con la información que se desea transmitir. La opción de fondo de color se consideró la menos apropiada, por sufrir problemas de contraste para la lectura y por asignar al color una relevancia demasiado alta cuando es sólo una parte de la información representativa de la alerta.

Opciones para mostrar color (listado):



Bloque

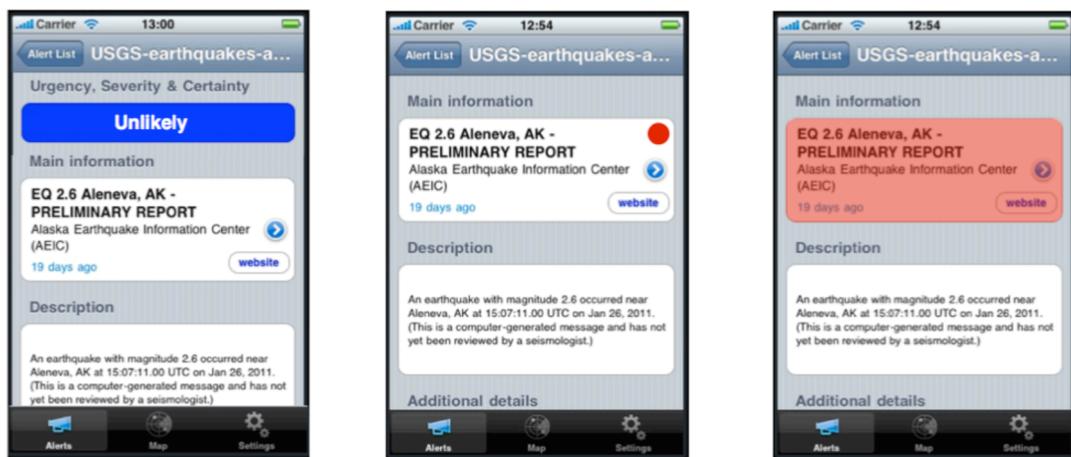
Círculo/Punto

Fondo

Figura 49: Opciones para mostrar color en el listado de alertas

La siguiente consulta realizada trataba del uso del color en la vista resumen de una alerta (Figura 50). De nuevo el experto seleccionó la opción de color en bloque por mantener una consistencia con la vista anterior en cuanto a la primera información mostrada en el resumen.

Opciones para mostrar color (resumen):



Bloque

Círculo/Punto

Fondo

Figura 50: Opciones para mostrar color en la vista resumen de alerta

En la siguiente diapositiva (Figura 51) se presentaron opciones para representar la certeza de una alerta, tanto en la vista de listado como de resumen. Estas propuestas se detallaban en la diapositiva de la Figura 52 donde se mostraron

ejemplos gráficos de las mismas. El experto opinó que una visualización en formato de texto era la más clara. Observó que el uso de un icono podría también resultar apropiado además del texto. El icono podría utilizar la metáfora del relleno o completitud para representar niveles crecientes de certeza, al estilo de marcadores de nivel típicos (como cobertura de red o nivel de carga).

Opciones para mostrar certeza:

- Incluir el texto literal: Observed, Likely, Possible, etc.
- Cambiar textura del color Severidad/Urgencia
- Utilizar iconos

Figura 51: Opciones para mostrar valor de certeza

Propuesta: textura de color para certeza

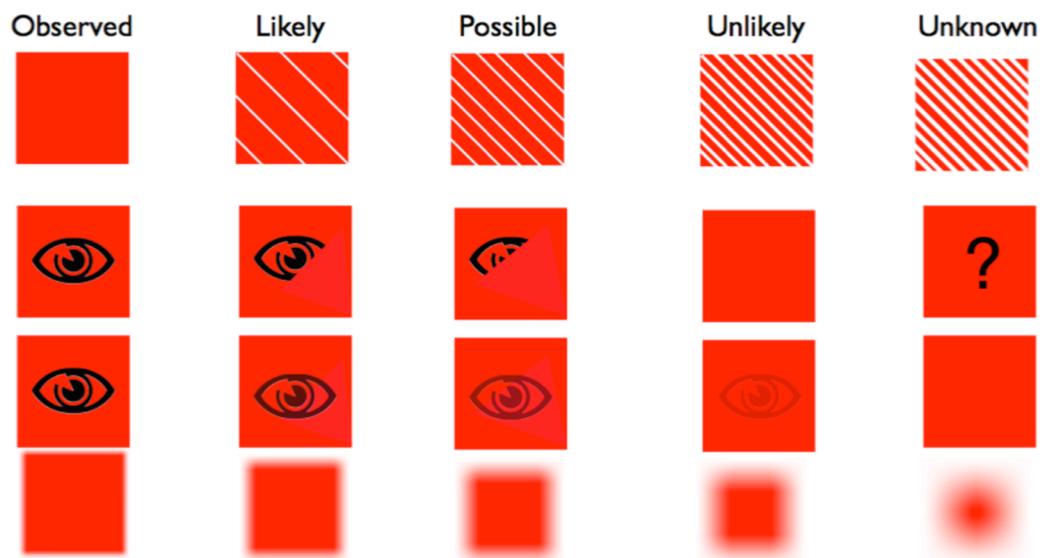


Figura 52: Propuesta de textura como muestra de certeza

La última propuesta (**Figura 53**) trataba la posibilidad de incluir una visualización en forma de gráfico de barras de los niveles de Urgencia, Severidad y Certeza. El experto seleccionó la primera opción con las barras en el mismo color no relacionado con los colores de la escala de importancia.

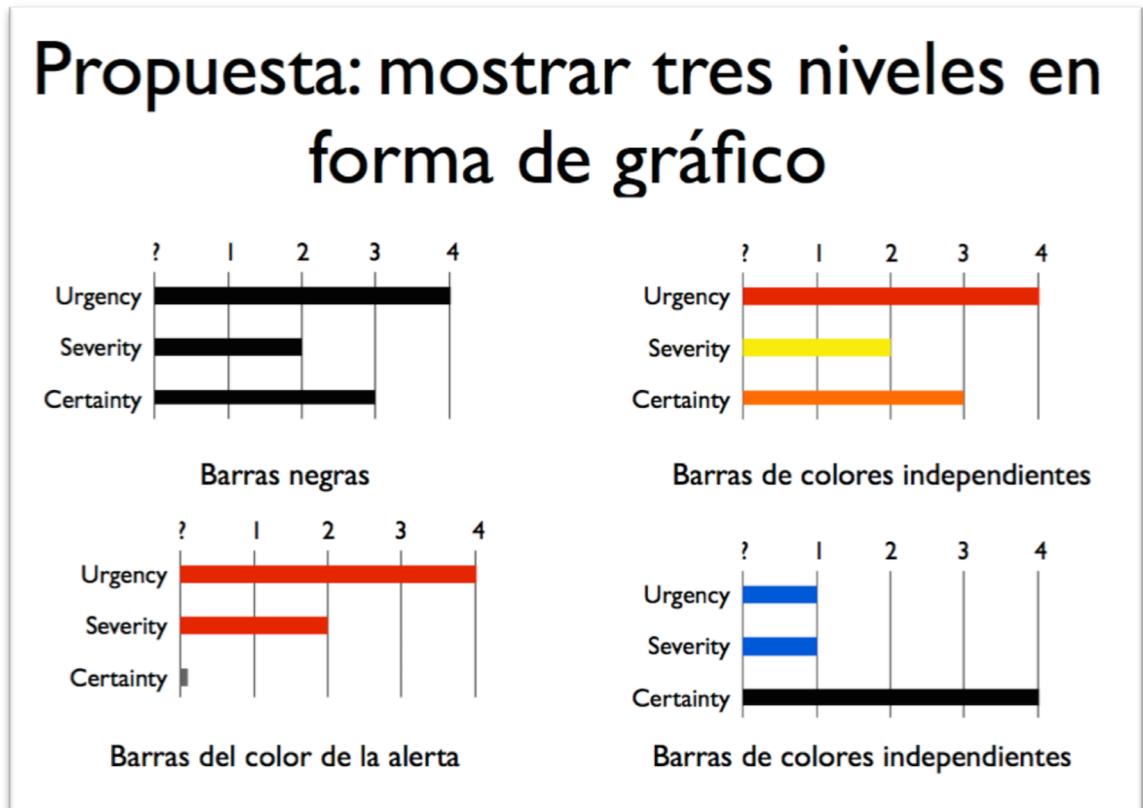


Figura 53: Propuesta de gráfico de niveles de Urgencia, Severidad y Certeza

Con los resultados de la evaluación se realizaron revisiones a los requisitos y a los diseños de interfaz de la aplicación. El objetivo de la reunión era conseguir la opinión de un experto ajeno al proyecto en cuanto a las decisiones de diseño tomadas, consultar su preferencia entre diferentes propuestas de diseño y conocer su opinión general sobre la aplicación.

4 Presupuesto

El producto de software generado como resultado de este proyecto es una aplicación sobre la plataforma iOS para el dispositivo iPhone de Apple. Se trata de una aplicación que permite la obtención y visualización de alertas de emergencia basadas en el estándar CAP, directamente de organizaciones especializadas en diferentes tipos de emergencias. La aplicación permite la selección de fuentes, el filtrado de alertas en base a su contenido, la visualización textual de la alerta en diferentes niveles de detalle, la visualización sobre un mapa de las alertas que lleven información de localización y el acceso a cualquier recurso que acompañe a la alerta siguiendo el estándar CAP.

La aplicación se ha desarrollado como una solución propuesta al siguiente problema: la falta de un sistema que integre las características mencionadas y siga un estándar para el formato de alertas. Tras un análisis y diseño se establecieron los requisitos del proyecto de tal manera que se cubriera al menos de manera demostrativa los aspectos más importantes de la solución propuesta. Aunque existen otros aspectos adicionales que se proponen como mejoras futuras a la solución, el presupuesto que a continuación se expone se ciñe estrictamente al desarrollo realizado durante este proyecto. Este desarrollo se caracteriza por lo siguiente:

- Desarrollo sobre la plataforma iOS para iPhone (aunque sería equivalente en otras plataformas móviles avanzadas estudiadas en el estado del arte, para dispositivos de tipo móvil - no tableta o PC portátil).
- Metodología de desarrollo de tipo ágil con una estructura flexible de iteraciones cortas y un producto incremental.
- Un único desarrollador, con experiencia general en el desarrollo de aplicaciones y programación pero no en la plataforma móvil (iOS en este caso).
- Un equipo reducido de análisis y diseño, en la que se incluye el desarrollador y un representante del cliente o dueño final.
- Acceso al hardware necesario para el desarrollo y familiaridad con la plataforma (Mac OS X).

El presupuesto podría ampliarse para contemplar un proyecto similar de mayor envergadura, especialmente en el caso de añadir el desarrollo simultáneo en múltiples plataformas móviles en base a un mismo análisis, diseño y evaluación.

4.1 Presupuesto resumido

Para el desarrollo completo de la aplicación iPhone descrita, contemplando un periodo de desarrollo de 3 meses amortizando el coste de dispositivos apropiadamente, se estima un coste total de **12.662,37€ (Doce mil seiscientos sesenta y dos euros con treinta y siete céntimos)**.

Concepto	Descripción	Coste
<i>Personal</i>	Incluye el coste de desarrollador(es) y el equipo de análisis, diseño y evaluación.	11.264,50€
<i>Material</i>	Incluye dispositivos de hardware para el desarrollo.	227,00€
<i>Licencias</i>	Cualquier licencia o coste asociado con el desarrollo.	19,75€
Coste total (aplicando un 10% de costes indirectos):		12.662,37€

Tabla 6: Presupuesto resumido

4.2 Presupuesto detallado

En la siguiente tabla se pueden observar cada uno de los conceptos individuales presupuestados para el proyecto, incluyendo su periodo de amortización. El presupuesto contempla una duración del proyecto de 3 meses con respecto a la amortización de material o dispositivos. Las estimaciones de coste por mes de los diferentes roles se han tomado de los salarios brutos medios indicados en salarios.infojobs.net, aplicando una cuota de 30% por seguridad social y una distribución en 12 pagas.

Concepto	Coste por mes (€)	Tiempo total de participación (meses)	Coste (€)
Desarrollador (Programador Junior)	2.166	3	6.498,00
Analista	3.250	1	3.250,00
Evaluador (Programador Senior)	3.033	0.5	1.516,50
Coste total personal:			11.264,50
Concepto	Coste total (€)	Periodo de amortización (meses)	Coste (€)
1 Dispositivo iPhone 4 16Gb	500	12	125,00
1 Mac Mini (configuración base)	700	24	87,00
1 Pantalla LCD de 20 pulgadas	118	36	10,00
1 Ratón óptico	30	36	2,50
1 Teclado QWERTY	30	36	2,50
Coste total material:			227,00
Concepto	Coste total (€)	Periodo de amortización (meses)	Coste (€)
1 Licencia de desarrollador iOS	79	12	19,75
Coste total licencias:			19,75
Coste total:			11.511,25€
Costes indirectos 10%:			1.151,12€
Precio final:			12.662,37€

Tabla 7: Presupuesto detallado

5 Solución y Conclusiones

Como cierre del proyecto se presenta a continuación la solución final propuesta para el problema definido en el apartado 1.1 de este documento. Los apartados siguientes contienen un resumen de la funcionalidad de la aplicación, una comparativa con los objetivos planteados y una serie de mejoras o ampliaciones futuras propuestas. Se termina con una descripción breve de la contribución total del proyecto y unas conclusiones personales.

5.1 Solución final

5.1.1 Presentación de la aplicación

La solución propuesta por este proyecto al problema planteado consta de una aplicación para la plataforma iOS de Apple®. La aplicación presenta una funcionalidad y unas características que alivian en gran parte las deficiencias asociadas con la manera actual de obtener alertas de emergencia directamente de organizaciones especializadas (RSS, aplicaciones web y móviles poco configurables). Las funcionalidades más destacadas de la aplicación son las siguientes:

- **Selección de fuentes** de las cuales se desea recibir alertas. En cualquier momento se puede modificar la selección.
- **Agregación de todas las alertas** en dos vistas (ambas permiten seleccionar una alerta para verla en mayor detalle):
 - *Listado*: vista en la cual se muestran las alertas recibidas en forma de texto, con una información breve y representativa de cada alerta.
 - *Mapa*: vista en la cual se muestran marcadores para las alertas recibidas sobre un mapa.
- **Visualización resumen** de una alerta: una vista homogénea para todas las alertas que muestra la información más relevante. Además da acceso a los recursos que acompañan a una alerta, a la información completa de ésta y a una vista individual en mapa de su ubicación. Se permite compartir el resumen de una alerta por correo electrónico desde esta vista.
- **Filtrado y organización** de alertas por sus campos como: severidad, urgencia, certeza, categoría, importancia o fecha de emisión.
- **Uso de colores e iconos** para reflejar la importancia relativa de una alerta y su certeza, permitiendo una interpretación más rápida de la información asociada a ésta.

Se puede observar que las características de la aplicación ponen solución a las desventajas detalladas en el apartado 1.1 de la siguiente manera:

1. **Fragmentación de fuentes**: ya no es necesario acudir individualmente a cada fuente de alertas, ni descubrir qué fuentes existen. Basta con seleccionar las fuentes proporcionadas en la aplicación.
2. **Presentación heterogénea**: todas las alertas se muestran de la misma manera en listado, resumen y detalle independientemente de su fuente.
3. **Falta de personalización**: se permite el filtrado por varios campos de la alertas, además de visualizaciones en listado o mapa.
4. **Comunicación unidireccional**: la aplicación no resuelve del todo esta desventaja, aunque forma una base para futuras mejoras que puedan contemplar la comunicación de un usuario hacia las organizaciones. La aplicación permite la redifusión de la alerta mediante un correo electrónico desde la aplicación que se puede considerar como una extensión de la difusión de la alerta, pero en el mismo sentido.

5.1.1.1 Manual de uso

Lo primero que se debe conocer de la aplicación son sus tres vistas principales: Listado de Alertas, Mapa y Ajustes (*Alerts*, *Map* y *Settings*) accesibles mediante la barra inferior:

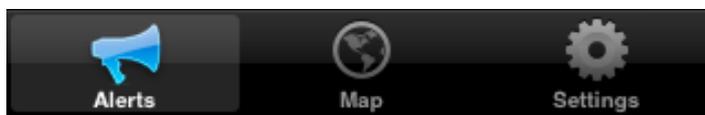


Figura 54: Barra inferior de la aplicación

La primera vez que se utiliza la aplicación se deben seleccionar las fuentes deseadas en *Settings*. En caso de no seleccionar ninguna fuente la aplicación lo avisará cuando se intenten recargar las alertas.

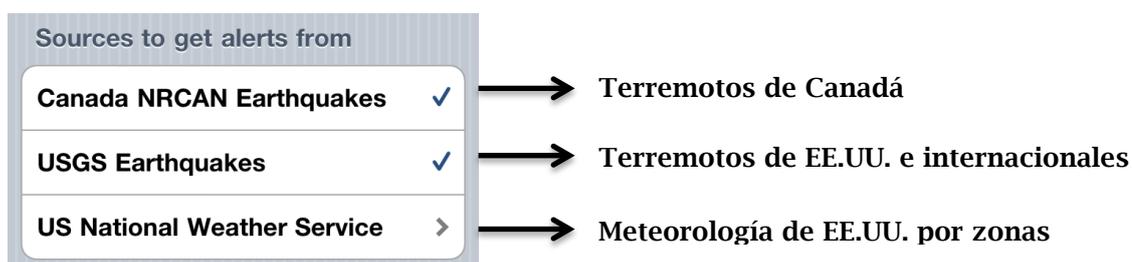


Figura 55: Selección de fuentes en la aplicación

Desde la vista de *Alerts* se pulsa el botón de recarga en la esquina superior derecha y comienza la descarga de alertas. Se puede detener en cualquier momento la descarga pulsando el botón de parada que se muestra. Al cerrar la aplicación se almacenan las alertas descargadas que se muestran de nuevo al abrir.



Figura 56: Acción de descarga de alertas y parada

Las alertas van poblando el listado y se muestra la siguiente información de cada una: título, origen, tiempo desde emisión, un color de nivel de importancia y un indicador de certeza de cero a cuatro barras verdes.

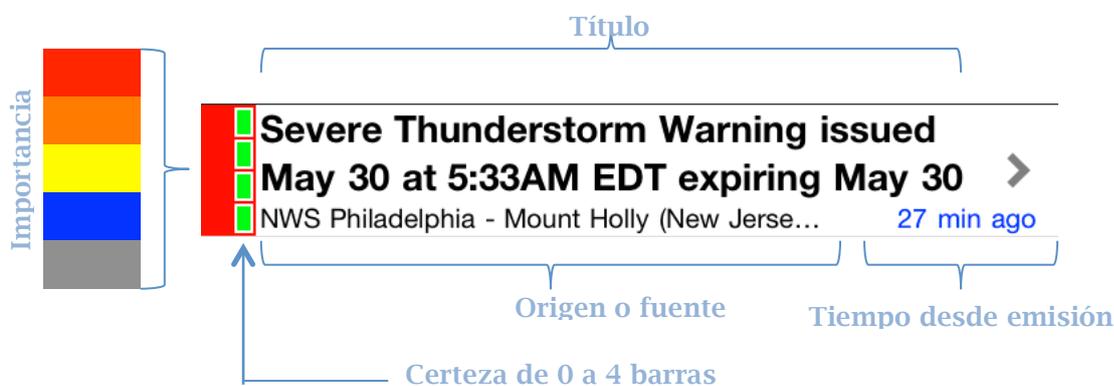


Figura 57: Descripción de alerta en listado

Al seleccionar una alerta del listado se pasa a la vista resumen de la misma, donde se muestran campos adicionales como la descripción. Desde esta vista se puede acceder a los siguientes aspectos de la alerta: detalle de todos los campos (tal y como llegaron en la alerta), vista en mapa, vista de recursos que acompañan la alerta y acceso a la página web de la alerta. Es posible que la alerta no contenga la información de mapa, página web o recursos y en tal caso los botones correspondientes aparecerán desactivados.



Figura 58: Descripción de vista resumen de alerta

Desde la vista resumen también se puede optar por compartir la alerta por medio de correo electrónico, en cuyo caso se despliega la interfaz del correo y se vuelve a la aplicación una vez enviado.

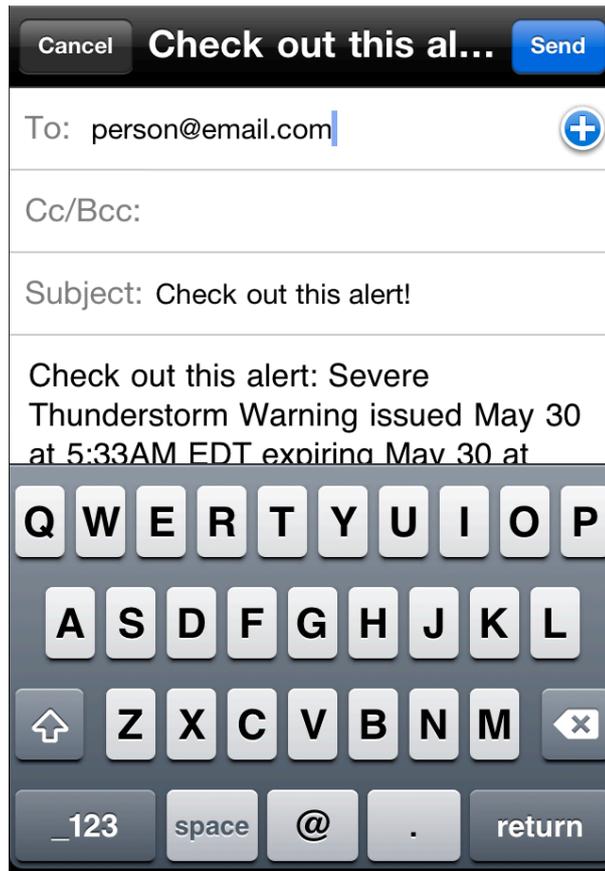


Figura 59: Vista de compartir alerta por correo

Volviendo al listado de alertas se puede observar el botón segmentado en la parte superior que permite definir la ordenación de la lista: por importancia o por fecha de emisión.



Figura 60: Botón de ordenación de listado de alertas

Las alertas que se muestran en el listado están filtradas en función a tres opciones: categoría (accesible desde el listado), caducidad (*Settings > Hide expired alerts*) y valores de importancia (*Settings > Importance filter settings*).

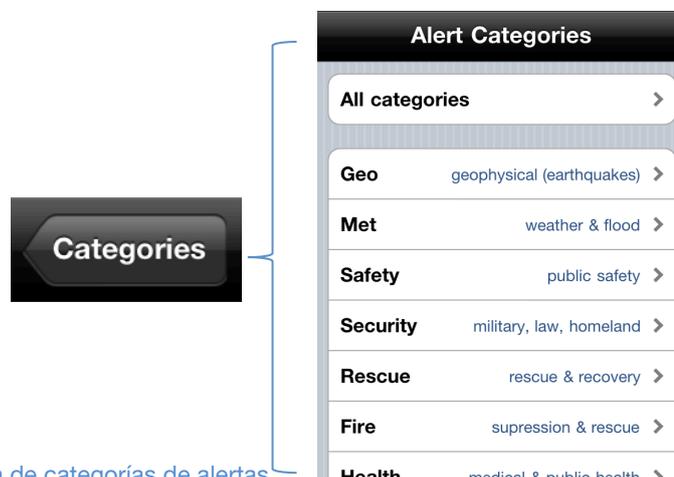


Figura 61: Selección de categorías de alertas

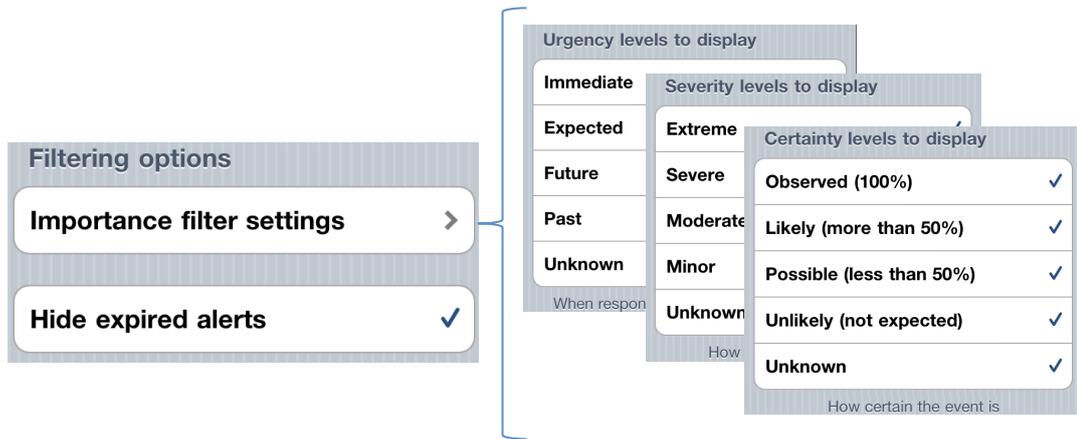


Figura 62: Selección de opciones de filtrado de alertas

En cualquier momento se puede acceder a la segunda vista principal de la aplicación, *Map*, en la cual se mostrarán todas las alertas del listado que contengan información de localización. Por lo tanto las opciones de filtrado afectan de la misma manera a las alertas mostradas en el mapa. La vista posee un control para alternar entre los distintos tipos de mapa (Normal, Satélite e Híbrido).



Figura 63: Vista en mapa de alertas

Al pulsar el marcador de una alerta en el mapa el botón azul de acceso permite acceder a la vista resumen de una alerta al igual que seleccionar una del listado.

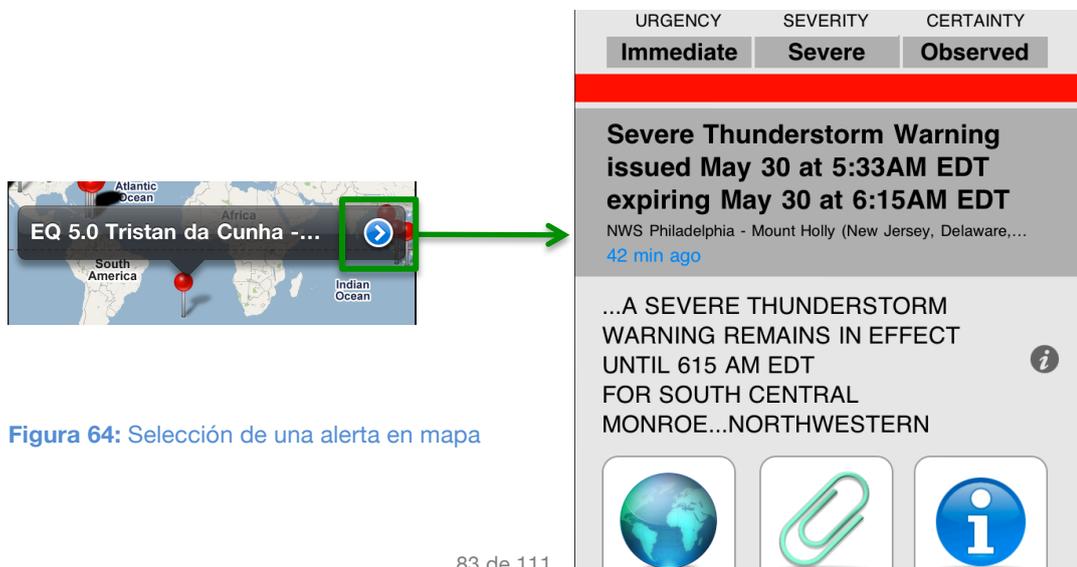


Figura 64: Selección de una alerta en mapa

Adicionalmente, cabe destacar las siguientes funcionalidades de la aplicación:

La sección de ayuda disponible debajo del listado de categorías:



Figura 65: Sección de ayuda de la aplicación

Por último, se provee un ajuste en *Settings* para limitar el número de alertas descargadas de cada fuente (*Download limit*), indicado para situaciones de conectividad baja o velocidades reducidas de conexión de datos:

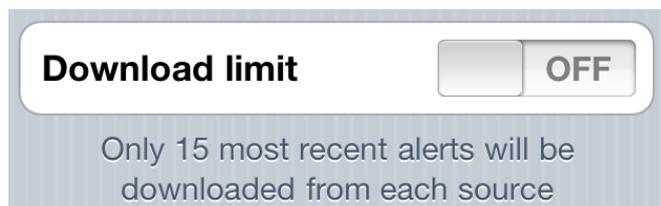


Figura 66: Opción de limite de descarga de la aplicación

5.1.2 Cumplimiento de objetivos del proyecto

Al comienzo del proyecto se establecieron una serie de objetivos que se incluyen a continuación:

1. Estudiar el estado actual de la tecnología de difusión de alertas de emergencia. Esto incluye aplicaciones e iniciativas existentes, además del uso de estándares para permitir interoperabilidad de sistemas.
2. Diseñar una plantilla de solución que reduzca o elimine las deficiencias existentes en la difusión de alertas. Este diseño contemplará tanto la naturaleza de la información tratada (alertas) como aspectos de diseño de interfaces y usabilidad.
3. Crear un prototipo como prueba de concepto de la solución, proporcionando una muestra no sólo del producto final sino también del proceso seguido hasta obtenerlo. Este prototipo puede servir de plantilla para futuras soluciones ampliadas o incluso podría llegar a ser desplegado en el mercado de aplicaciones para la plataforma escogida.

En el apartado 2. Estado del arte se recoge todo aquello relacionado con el primer objetivo. Se ha estudiado tanto el estado de las alertas de emergencia y los estándares asociados, como las organizaciones e iniciativas en el mundo de las alertas de emergencia. Se han analizado aplicaciones existentes en plataformas de escritorio, web y móvil para conseguir una comparativa con la solución propuesta. Finalmente, se han estudiado las tecnologías móviles como tal, tanto para asegurar que proporcionan plataformas viables para el desarrollo de la solución propuesta como para comparar plataformas existentes y elegir la más apropiada para la prueba de concepto.

El segundo objetivo se resuelve en la mayor parte en el apartado 3.1 Análisis y Diseño donde se analizan los aspectos claves para la creación de una solución del tipo propuesto (aplicación móvil). Aunque se centra en torno a la plataforma escogida, se considera que las decisiones tomadas son aplicables a otras plataformas similares y sufrirían sólo leves cambios en función de las capacidades técnicas disponibles.

Por último el tercer objetivo se cumple con lo descrito en los apartados 3.2, 3.3 y 4. Se describe tanto el prototipo creado como prueba de concepto como el proceso seguido para su desarrollo y evaluación. En el apartado 5.2 se describe el despliegue de una aplicación basada en el prototipo como contribución al sector.

5.1.3 Mejoras futuras

Tomando como base el prototipo generado durante el proyecto existen una serie de mejoras o ampliaciones futuras que se proponen a continuación. Estas mejoras no se han incluido en el proyecto actual por no formar parte de la funcionalidad básica buscada, por limitaciones temporales o por restricciones ajenas al proyecto. Las mejoras propuestas son:

- **Ampliación de fuentes:** según se adopte de manera más extensa el estándar CAP se podrán añadir fuentes a la aplicación para mejorar su utilidad. Iniciativas como las descritas en el apartado 2.1.3 favorecen la adopción del estándar.
- **Notificaciones en tiempo real:** aparte de servir como lector pasivo de alertas la aplicación podría incorporar un sistema de aviso dinámico de alertas. Estos avisos podrían depender de la importancia de la alerta o el lugar geográfico del usuario.
- **Contestación a las alertas:** con la participación de las organizaciones, o mediante un desarrollo adicional al de la aplicación móvil, se podría crear un sistema para responder a las alertas. Esto podría ir desde un simple aviso de recepción hasta avisos por parte de aquellos afectados.

5.2 Contribución

Este proyecto pretende contribuir una solución innovadora para la obtención de alertas de emergencia en un dispositivo móvil avanzado (smartphone). La naturaleza de los smartphones aumentan la efectividad de la difusión de alertas: están siempre con el usuario, mantienen conexiones constantes a Internet y poseen las características técnicas para mostrar todos los aspectos posibles de una alerta (mapas, multimedia, notificaciones en tiempo real, etc.). Se pueden agrupar las contribuciones en dos grandes bloques:

1. **Estudio del estado del arte:** todo el estudio del estado del arte y los procesos de análisis y diseño pueden servir tanto de referencia como base para proyectos futuros. Del estudio han surgido ideas como la representación de la importancia de una alerta mediante una gama de color, representación de la certeza por medio de iconos de nivel, etc. Son conceptos basados en ideas existentes, pero extendidos para adaptarse al estándar CAP.
2. **Análisis y Diseño:** de un mecanismo para la obtención de alertas de emergencia en la forma de una aplicación funcional y eficiente que aprovecha las características de un dispositivo móvil de última generación para enriquecer la obtención de alertas mediante acceso móvil a Internet, acceso a contenido multimedia, geo-localización, etc.

En definitiva la innovación de este proyecto se centra en torno al uso de una plataforma móvil para la obtención alertas de emergencia, al uso exclusivo del estándar CAP (que se encuentra en auge) y la posibilidad de visualizar toda la información que acompaña a una alerta bajo una misma interfaz, desde mapas hasta recursos multimedia. Se cede más control al usuario que ahora puede ajustar la configuración del sistema en función de sus objetivos o necesidades.

5.2.1 World Alerts App

Se ha creado una aplicación real en base al prototipo del proyecto que se llama World Alerts. Existen dos aspectos principales en las que se diferencia del prototipo. El primero es la propia imagen de la aplicación, su nombre y su icono:



Figura 67: Icono de World Alerts

La inclusión en la App Store conlleva un proceso de identificación y descripción de la aplicación para el público, además de un proceso de verificación por parte de Apple en la que se revisa el código, la funcionalidad y la apariencia de la aplicación en base a una serie de requisitos.

El segundo aspecto es la monetización de la aplicación. La licencia de desarrollador tiene un coste de 79€ anuales, y en caso de querer recuperar esa cantidad por ingresos de la aplicación existen dos posibilidades: hacer la aplicación de pago o incluir publicidad en la aplicación. Se ha optado por la segunda opción, ya que permite que todo aquel interesado en la aplicación la pueda obtener de manera gratuita, y los ingresos por publicidad irán relacionados con el uso que se le da. La publicidad se ha incluido mediante el servicio AdMob de Google (www.admob.com) y consta de una sección rectangular (*banner*) que se incluye en la vista resumen de una alerta como se muestra a continuación:

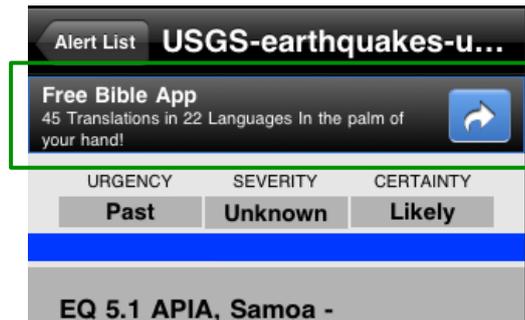


Figura 68: Zona de publicidad en la vista resumen de alerta

El objetivo es mantener la aplicación disponible, realizando mejoras incrementales en el futuro en base al uso de la aplicación por parte de los usuarios.

5.3 Conclusiones personales

Al finalizar este proyecto estoy muy satisfecho con el resultado. Siento que es un proyecto completo que cubre muchos aspectos de la carrera de Informática, y que me ha permitido aprender aún más durante su realización. Uno de las motivaciones personales para realizar el proyecto ha sido aprender a desarrollar en la plataforma iOS. He conseguido ese objetivo con creces, ya que además el desarrollo me ha permitido lanzar mi primera aplicación al mercado. Además he mejorado mis habilidades de documentación, investigación, estudio, análisis, diseño y evaluación. No podría pensar mejor forma de clausurar mi carrera de Ingeniería en Informática, y espero realizar una aportación de algún valor.

6 Referencias

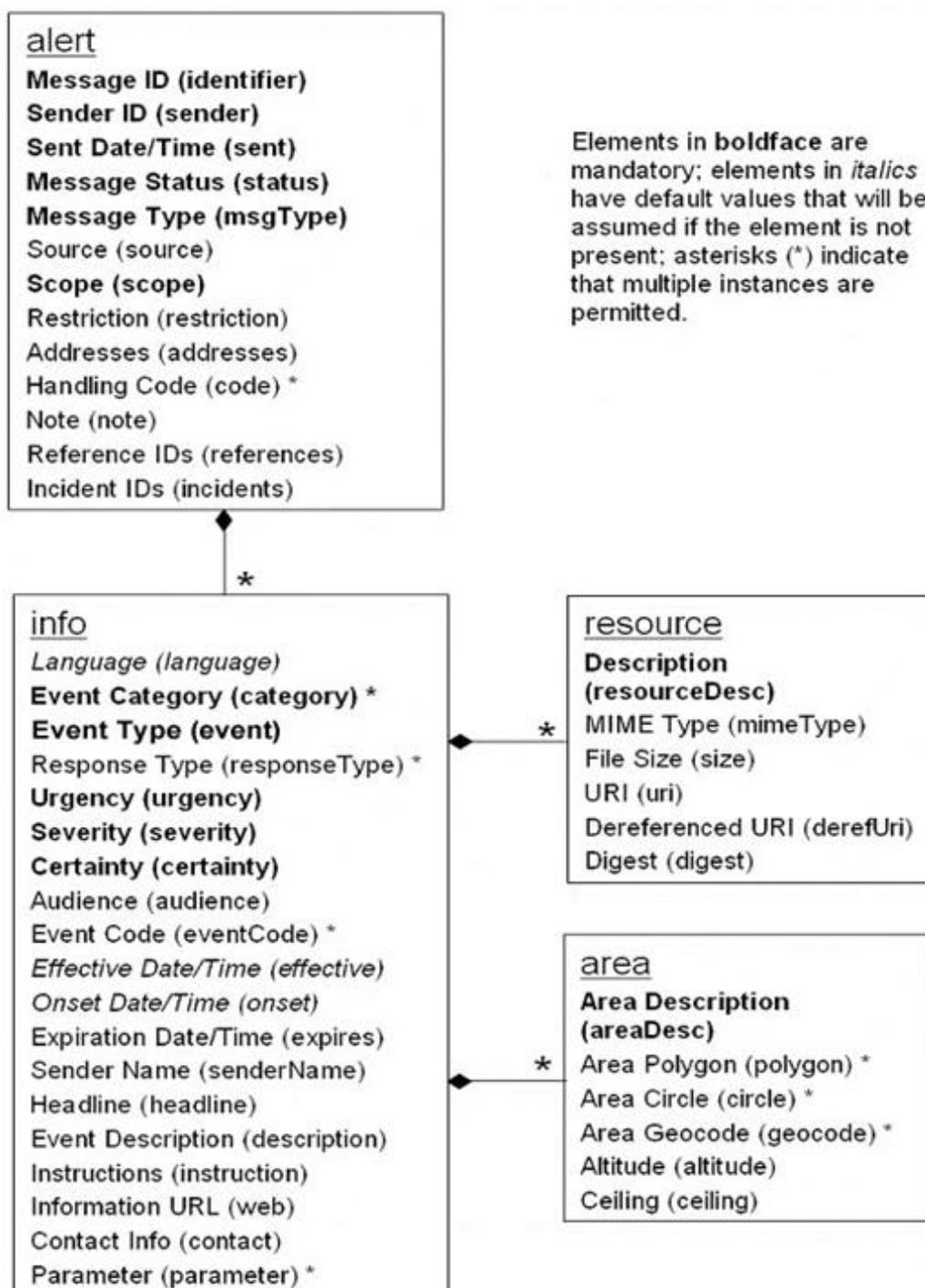
6.1 Referencias y enlaces

- Agile Alliance. (2011). *The agile manifesto*. Retrieved 2011 from Agile Alliance: <http://www.agilealliance.org/the-alliance/the-agile-manifesto/>
- Apple Inc. (2011). *Apple Developer*. Retrieved 2011 from Apple: <http://developer.apple.com/>
- Bleeker, E. (09 de 2010). *Microsoft's Billion Dollar Question*. From fool.com: <http://www.fool.com/investing/value/2010/09/04/microsofts-billion-dollar-question.aspx>
- comScore. (2011). From <http://www.prnewswire.com/news-releases/comscore-releases-inaugural-report-the-2010-mobile-year-in-review-116157639.html>
- ComScore. (02 de 2011). *February 2011 U.S. Mobile Subscriber Market Share Report*. From http://www.comscore.com/Press_Events/Press_Releases/2011/4/comScore_Reports_February_2011_U.S._Mobile_Subscriber_Market_Share?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+comscore+%28comScore%2C+Inc.%29
- EDIS. (n.d.). From <http://hisz.rsoe.hu/alertmap/>
- FEMA. (n.d.). From <http://www.fema.gov/>
- google.org. (n.d.). From <http://alert-hub.appspot.com>
- IPAWS. (n.d.). From <http://www.fema.gov/emergency/ipaws/>
- ITU. (2009). From <http://www.itu.int/ITU-D/ict/newslog/Spain+Ends+2008+With+A+Subscriber+Base+Of+5089Mn.aspx>
- Machine 2 Machine. (n.d.). From <http://m2m.tmcnet.com/news/2009/06/12/4222607.htm>
- OASIS. (2011). *CAP 1.2 Standard*. Retrieved 2011 from Organization for the Advancement of Structured Information Standards: <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>
- Paczkowski, J. (n.d.). *Invasion of the Market Share Snatchers: BlackBerry Losing Share to Android?* From digitaldaily.allthingsd.com: <http://digitaldaily.allthingsd.com/20100512/blackberry-losing-to-droid-piper/>
- Pacific Disaster Center. (n.d.). From <http://www.pdc.org/>
- scrum.org. (n.d.). *Scrum Guide*. From <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf>
- Sigal, M. (02 de 2009). *Android vs. iPhone: Why Openness May Not Be Best*. From gigaom: <http://gigaom.com/2009/02/22/is-being-%E2%80%9Copen%E2%80%9D-an-absolute-in-mobile/>
- Reed, B. (04 de 2008). *SDK showdown: iPhone vs. Android*. From Macworld: http://www.macworld.com/article/133152/2008/04/iphone_android.html
- The Nieslen Company. (02 de 2011). *Smartphone study*. From <http://blog.nielsen.com/nielsenwire/?p=25901>

Anexo A: Estructura de mensaje CAP

Los siguientes apartados detallan tanto la estructura de un mensaje CAP como el significado de cada uno de los campos y las restricciones impuestas. Las tablas se han obtenido directamente de la especificación del estándar por lo que están en Inglés. Para conocer el estándar CAP completo se puede acudir a la página oasis-open.org.

Modelo de objeto de documento



Diccionario de datos

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
"alert" Element and Sub-elements			
alert	cap. alert. group	The container for all component parts of the alert message (REQUIRED)	<p>(1) Surrounds CAP alert message sub-elements.</p> <p>(2) MUST include the xmlns attribute referencing the CAP URN as the namespace, e.g.:</p> <pre><cap:alert xmlns:cap="urn:oasis:names:tc:emergency:cap:1.2"> [sub-elements] </cap:alert></pre> <p>(3) In addition to the specified sub-elements, MAY contain one or more <info> blocks.</p>
identifier	cap. alert. identifier. identifier	The identifier of the alert message (REQUIRED)	<p>(1) A number or string uniquely identifying this message, assigned by the sender.</p> <p>(2) MUST NOT include spaces, commas or restricted characters (< and &).</p>
sender	cap. alert. sender. identifier	The identifier of the sender of the alert message (REQUIRED)	<p>(1) Identifies the originator of this alert. Guaranteed by assigner to be unique globally; e.g., may be based on an Internet domain name.</p> <p>(2) MUST NOT include spaces, commas or restricted characters (< and &).</p>
sent	cap. alert. sent. time	The time and date of the origination of the alert message (REQUIRED)	<p>(1) The date and time SHALL be represented in the DateTime Data Type (See Implementation Notes) format (e.g., "2002-05-24T16:49:00-07:00" for 24 May 2002 at 16:49 PDT).</p> <p>(2) Alphabetic timezone designators such as "Z" MUST NOT be used. The timezone for UTC MUST be represented as "-00:00".</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
status	cap. alert. status. code	The code denoting the appropriate handling of the alert message (REQUIRED)	Code Values: "Actual" - Actionable by all targeted recipients "Exercise" - Actionable only by designated exercise participants; exercise identifier SHOULD appear in <note> "System" - For messages that support alert network internal functions "Test" - Technical testing only, all recipients disregard "Draft" - A preliminary template or draft, not actionable in its current form
msgType	cap. alert. msgType. code	The code denoting the nature of the alert message (REQUIRED)	Code Values: "Alert" - Initial information requiring attention by targeted recipients "Update" - Updates and supercedes the earlier message(s) identified in <references> "Cancel" - Cancels the earlier message(s) identified in <references> "Ack" - Acknowledges receipt and acceptance of the message(s) identified in <references> "Error" - Indicates rejection of the message(s) identified in <references>; explanation SHOULD appear in <note>
source	cap. alert. source. identifier	The text identifying the source of the alert message (OPTIONAL)	The particular source of this alert; e.g., an operator or a specific device.
scope	cap. alert. scope. code	The code denoting the intended distribution of the alert message (REQUIRED)	Code Values: "Public" - For general dissemination to unrestricted audiences "Restricted" - For dissemination only to users with a known operational requirement (see <restriction>, below) "Private" - For dissemination only to specified addresses (see <addresses>, below)

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
restriction	cap. alert. restriction. text	The text describing the rule for limiting distribution of the restricted alert message (CONDITIONAL)	Used when <scope> value is "Restricted".
addresses	cap. alert. addresses. group	The group listing of intended recipients of the alert message (CONDITIONAL)	<p>(1) Required when <scope> is "Private", optional when <scope> is "Public" or "Restricted".</p> <p>(2) Each recipient SHALL be identified by an identifier or an address.</p> <p>(3) Multiple space-delimited addresses MAY be included. Addresses including whitespace MUST be enclosed in double-quotes.</p>
code	cap. alert. code. code	The code denoting the special handling of the alert message (OPTIONAL)	<p>(1) Any user-defined flag or special code used to flag the alert message for special handling.</p> <p>(2) Multiple instances MAY occur.</p>
note	cap. alert. note. text	The text describing the purpose or significance of the alert message (OPTIONAL)	The message note is primarily intended for use with <status> "Exercise" and <msgType> "Error".
references	cap. alert. references. group	The group listing identifying earlier message(s) referenced by the alert message (OPTIONAL)	<p>(1) The extended message identifier(s) (in the formsender, identifier, sent) of an earlier CAP message or messages referenced by this one.</p> <p>(2) If multiple messages are referenced, they SHALL be separated by whitespace.</p>
incidents	cap. alert. incidents. group	The group listing naming the referent incident(s) of the alert message (OPTIONAL)	<p>(1) Used to collate multiple messages referring to different aspects of the same incident.</p> <p>(2) If multiple incident identifiers are referenced, they SHALL be separated by whitespace. Incident names including whitespace SHALL be surrounded by double-quotes.</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
"info" Element and Sub-elements			
info	cap. alertInfo. info. group	The container for all component parts of the info sub-element of the alert message (OPTIONAL)	<p>(1) Multiple occurrences are permitted within a single <alert>. If targeting of multiple <info> blocks in the same language overlaps, information in later blocks may expand but may not override the corresponding values in earlier ones. Each set of <info> blocks containing the same language identifier SHALL be treated as a separate sequence.</p> <p>(2) In addition to the specified sub-elements, MAY contain one or more <resource> blocks and/or one or more <area> blocks.</p>
language	cap. alertInfo. language. code	The code denoting the language of the info sub-element of the alert message (OPTIONAL)	<p>(1) Code Values: Natural language identifier per [RFC 3066].</p> <p>(2) If not present, an implicit default value of "en-US" SHALL be assumed.</p> <p>(3) A null value in this element SHALL be considered equivalent to "en-US."</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
category	cap. alertInfo. category. code	The code denoting the category of the subject event of the alert message(REQUIRED)	<p>(1) Code Values:</p> <ul style="list-style-type: none"> “Geo” - Geophysical (inc. landslide) “Met” - Meteorological (inc. flood) “Safety” - General emergency and public safety “Security” - Law enforcement, military, homeland and local/private security “Rescue” - Rescue and recovery “Fire” - Fire suppression and rescue “Health” - Medical and public health “Env” - Pollution and other environmental “Transport” - Public and private transportation “Infra” - Utility, telecommunication, other non-transport infrastructure “CBRNE” – Chemical, Biological, Radiological, Nuclear or High-Yield Explosive threat or attack “Other” - Other events <p>(2) Multiple instances MAY occur within an <info> block.</p>
event	cap. alertInfo. event. text	The text denoting the type of the subject event of the alert message(REQUIRED)	

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
responseType	cap. alertInfo. responseType. code	The code denoting the type of action recommended for the target audience (OPTIONAL)	<p>(1) Code Values:</p> <p>“Shelter” – Take shelter in place or per <instruction></p> <p>“Evacuate” – Relocate as instructed in the <instruction></p> <p>“Prepare” – Make preparations per the <instruction></p> <p>“Execute” – Execute a pre-planned activity identified in <instruction></p> <p>“Avoid” – Avoid the subject event as per the <instruction></p> <p>“Monitor” – Attend to information sources as described in <instruction></p> <p>“Assess” – Evaluate the information in this message. (This value SHOULD NOT be used in public warning applications.)</p> <p>“AllClear” – The subject event no longer poses a threat or concern and any follow on action is described in <instruction></p> <p>“None” – No action recommended</p> <p>(2) Multiple instances MAY occur within an <info> block.</p>
urgency	cap. alertInfo. urgency. code	The code denoting the urgency of the subject event of the alert message (REQUIRED)	<p>(1) The <urgency>, <severity>, and <certainty> elements collectively distinguish less emphatic from more emphatic messages.</p> <p>(2) Code Values:</p> <p>“Immediate” - Responsive action SHOULD be taken immediately</p> <p>“Expected” - Responsive action SHOULD be taken soon (within next hour)</p> <p>“Future” - Responsive action SHOULD be taken in the near future</p> <p>“Past” - Responsive action is no longer required</p> <p>“Unknown” - Urgency not known</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
severity	cap. alertInfo. severity. code	The code denoting the severity of the subject event of the alert message (REQUIRED)	<p>(1) The <urgency>, <severity>, and <certainty> elements collectively distinguish less emphatic from more emphatic messages.</p> <p>(2) Code Values:</p> <p>“Extreme” - Extraordinary threat to life or property</p> <p>“Severe” - Significant threat to life or property</p> <p>“Moderate” - Possible threat to life or property</p> <p>“Minor” – Minimal to no known threat to life or property</p> <p>“Unknown” - Severity unknown</p>
certainty	cap. alertInfo. certainty. code	The code denoting the certainty of the subject event of the alert message (REQUIRED)	<p>(1) The <urgency>, <severity>, and <certainty> elements collectively distinguish less emphatic from more emphatic messages.</p> <p>(2) Code Values:</p> <p>“Observed” – Determined to have occurred or to be ongoing</p> <p>“Likely” - Likely (p > ~50%)</p> <p>“Possible” - Possible but not likely (p <= ~50%)</p> <p>“Unlikely” - Not expected to occur (p ~ 0)</p> <p>“Unknown” - Certainty unknown</p> <p>(3) For backward compatibility with CAP 1.0, the deprecated value of “Very Likely” SHOULD be treated as equivalent to “Likely”.</p>
audience	cap. alertInfo. audience. text	The text describing the intended audience of the alert message (OPTIONAL)	

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
eventCode	cap. alertInfo. eventCode. code	A system-specific code identifying the event type of the alert message (OPTIONAL)	<p>(1) Any system-specific code for event typing, in the form:</p> <pre><eventCode> <valueName>valueName</valueName> > <value>value</value> </eventCode></pre> <p>where the content of "valueName" is a user-assigned string designating the domain of the code, and the content of "value" is a string (which may represent a number) denoting the value itself (e.g., valueName = "SAME" and value = "CEM").</p> <p>(2) Values of "valueName" that are acronyms SHOULD be represented in all capital letters without periods (e.g., SAME, FIPS, ZIP).</p> <p>(3) Multiple instances MAY occur within an <info> block.</p>
effective	cap. alertInfo. effective. time	The effective time of the information of the alert message (OPTIONAL)	<p>(1) The date and time SHALL be represented in the DateTime Data Type (See Implementation Notes) format (e.g., "2002-05-24T16:49:00-07:00" for 24 May 2002 at 16:49 PDT).</p> <p>(2) Alphabetic timezone designators such as "Z" MUST NOT be used. The timezone for UTC MUST be represented as "-00:00".</p> <p>(3) If this item is not included, the effective time SHALL be assumed to be the same as in <sent>.</p>
onset	cap. alertInfo. onset. time	The expected time of the beginning of the subject event of the alert message (OPTIONAL)	<p>(1) The date and time SHALL be represented in the DateTime Data Type (See Implementation Notes) format (e.g., "2002-05-24T16:49:00-07:00" for 24 May 2002 at 16:49 PDT).</p> <p>(2) Alphabetic timezone designators such as "Z" MUST NOT be used. The timezone for UTC MUST be represented as "-00:00".</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
expires	cap. alertInfo. expires. time	The expiry time of the information of the alert message (OPTIONAL)	<p>(1) The date and time SHALL be represented in the DateTime Data Type (See Implementation Notes) format (e.g., "2002-05-24T16:49:00-07:00" for 24 May 2002 at 16:49 PDT).</p> <p>(2) Alphabetic timezone designators such as "Z" MUST NOT be used. The timezone for UTC MUST be represented as "-00:00".</p> <p>(3) If this item is not provided, each recipient is free to set its own policy as to when the message is no longer in effect.</p>
senderName	cap. alertInfo. senderName. text	The text naming the originator of the alert message (OPTIONAL)	The human-readable name of the agency or authority issuing this alert.
headline	cap. alertInfo. headline. text	The text headline of the alert message (OPTIONAL)	A brief human-readable headline. Note that some displays (for example, short messaging service devices) may only present this headline; it SHOULD be made as direct and actionable as possible while remaining short. 160 characters MAY be a useful target limit for headline length.
description	cap. alertInfo. description. text	The text describing the subject event of the alert message (OPTIONAL)	An extended human readable description of the hazard or event that occasioned this message.
instruction	cap. alertInfo. instruction. text	The text describing the recommended action to be taken by recipients of the alert message (OPTIONAL)	An extended human readable instruction to targeted recipients. If different instructions are intended for different recipients, they should be represented by use of multiple <info> blocks.

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
web	cap.alertInfo.web.identifier	The identifier of the hyperlink associating additional information with the alert message (OPTIONAL)	A full, absolute URI for an HTML page or other text resource with additional or reference information regarding this alert.
contact	cap.alertInfo.contact.text	The text describing the contact for follow-up and confirmation of the alert message (OPTIONAL)	
parameter	cap.alertInfo.parameter.code	A system-specific additional parameter associated with the alert message (OPTIONAL)	<p>(1) Any system-specific datum, in the form:</p> <pre><parameter> <valueName>valueName</valueName> <value>value</value> </parameter></pre> <p>where the content of "valueName" is a user-assigned string designating the domain of the code, and the content of "value" is a string (which may represent a number) denoting the value itself (e.g., valueName ="SAME" and value="CIV").</p> <p>(2) Values of "valueName" that are acronyms SHOULD be represented in all capital letters without periods (e.g., SAME, FIPS, ZIP).</p> <p>(3) Multiple instances MAY occur within an <info> block.</p>
"resource" Element and Sub-elements			
resource	cap.alertInfoResource.resource.group	The container for all component parts of the resource sub-element of the info sub-element of the alert element (OPTIONAL)	<p>(1) Refers to an additional file with supplemental information related to this <info> element; e.g., an image or audio file.</p> <p>(2) Multiple instances MAY occur within an <info> block.</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
resourceDesc	cap. alertInfoResource. resourceDesc. text	The text describing the type and content of the resource file (REQUIRED)	The human-readable text describing the type and content, such as “map” or “photo”, of the resource file.
contentType	cap. alertInfoResource. contentType. identifier	The identifier of the MIME content type and sub-type describing the resource file (REQUIRED)	MIME content type and sub-type as described in [RFC 2046].(As of this document, the current IANA registered MIME types are listed at http://www.iana.org/assignments/media-types/)
size	cap. alertInfoResource. size. integer	The integer indicating the size of the resource file (OPTIONAL)	(1) Approximate size of the resource file in bytes. (2) For <uri> based resources, <size> SHOULD be included if available.
uri	cap. alertInfoResource. uri. identifier	The identifier of the hyperlink for the resource file (OPTIONAL)	A full absolute URI, typically a Uniform Resource Locator that can be used to retrieve the resource over the Internet OR a relative URI to name the content of a <derefUri> element if one is present in this resource block.

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
derefUri	cap. alertInfoResource. derefUri. data	The base-64 encoded data content of the resource file(CONDITIONAL)	(1) MAY be used either with or instead of the <uri> element in messages transmitted over one-way (e.g., broadcast) data links where retrieval of a resource via a URI is not feasible. (2) Clients intended for use with one-way data links MUST support this element. (3) This element MUST NOT be used unless the sender is certain that all direct clients are capable of processing it. (4) If messages including this element are forwarded onto a two-way network, the forwarder MUST strip the <derefUri> element and SHOULD extract the file contents and provide a <uri> link to a retrievable version of the file. (5) Providers of one-way data links MAY enforce additional restrictions on the use of this element, including message-size limits and restrictions regarding file types.
digest	cap. alertInfoResource. digest. code	The code representing the digital digest ("hash") computed from the resource file (OPTIONAL)	Calculated using the Secure Hash Algorithm (SHA-1) per [FIPS 180-2] .
"area" Element and Sub-elements			
area	cap. alertInfoArea. area. group	The container for all component parts of the area sub-element of the info sub-element of the alert message (OPTIONAL)	(1) Multiple occurrences permitted, in which case the target area for the <info> block is the union of all the included <area> blocks. (2) MAY contain one or multiple instances of <polygon>, <circle> or <geocode>. If multiple <polygon>, <circle> or <geocode> elements are included, the area described by this <area> block is represented by the union of all the included elements.
areaDesc	cap. alertInfoArea. areaDesc. text	The text describing the affected area of the alert message (REQUIRED)	A text description of the affected area.

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
polygon	cap. alertInfoArea. polygon. group	The paired values of points defining a polygon that delineates the affected area of the alert message (OPTIONAL)	<p>(1) Code Values: The geographic polygon is represented by a whitespace-delimited list of [WGS 84] coordinate pairs. (See WGS 84 Note at end of this section)</p> <p>(2) A minimum of 4 coordinate pairs MUST be present and the first and last pairs of coordinates MUST be the same.</p> <p>(3) Multiple instances MAY occur within an <area> block.</p>
circle	cap. alertInfoArea. circle. group	The paired values of a point and radius delineating the affected area of the alert message (OPTIONAL)	<p>(1) Code Values: The circular area is represented by a central point given as a [WGS 84] coordinate pair followed by a space character and a radius value in kilometers. (See WGS 84 Note at end of this section)</p> <p>(2) Multiple instances MAY occur within an <area> block.</p>

Element Name	Context. Class. Attribute. Representation	Definition and (Optionality)	Notes or Value Domain
geocode	cap. alertInfoArea. geocode. code	The geographic code delineating the affected area of the alert message (OPTIONAL)	<p>(1) Any geographically-based code to describe a message target area, in the form:</p> <pre><geocode> <valueName>valueName</valueName> <value>value</value> </geocode></pre> <p>where the content of "valueName" is a user-assigned string designating the domain of the code, and the content of "value" is a string (which may represent a number) denoting the value itself (e.g., valueName = "SAME" and value="006113").</p> <p>(2) Values of "valueName" that are acronyms SHOULD be represented in all capital letters without periods (e.g., SAME, FIPS, ZIP).</p> <p>(3) Multiple instances MAY occur within an <area> block.</p> <p>(4) This element is primarily for compatibility with other systems. Use of this element presumes knowledge of the coding system on the part of recipients; therefore, for interoperability, it SHOULD be used in concert with an equivalent description in the more universally understood <polygon> and <circle> forms whenever possible.</p>
altitude	cap. alertInfoArea. altitude. quantity	The specific or minimum altitude of the affected area of the alert message (OPTIONAL)	<p>(1) If used with the <ceiling> element this value is the lower limit of a range. Otherwise, this value specifies a specific altitude.</p> <p>(2) The altitude measure is in feet above mean sea level per the [WGS 84] datum.</p>
ceiling	cap. alertInfoArea. ceiling. quantity	The maximum altitude of the affected area of the alert message (CONDITIONAL)	<p>(1) MUST NOT be used except in combination with the <altitude> element.</p> <p>(2) The ceiling measure is in feet above mean sea level per the [WGS 84] datum.</p>

Anexo B: Documentación del Desarrollo

Historias de Usuario (Product Backlog)

ID	Título	Tipo	Prioridad	Estado
R01	Listado principal básico de alertas	Requisito Usuario	16	Completado
R02	Colores e iconos para mostrar urgencia, severidad y certeza de una alerta	Requisito Usuario	7	Completado
R03	Estándar mensajes CAP 1.2	Requisito Usuario	17	Completado
R04	Fuentes de mensajes CAP por URL o similar	Requisito Usuario	13	Completado
R05	Filtrado de alertas por caducidad, categoría, urgencia, certeza, severidad	Requisito Usuario	15	Completado
R06	Visualización resumida de una alerta con los campos más relevantes	Requisito Usuario	16	Completado
R07	Visualización detallada de la alerta con todos sus campos	Requisito Usuario	12	Completado
R08	Visualización en mapa de una alerta	Requisito Usuario	13	Completado
R09	Visualización en mapa de todas las alertas	Requisito Usuario	8	Completado
R10	Visualización de ajustes y preferencias de la aplicación	Requisito Usuario	8	Completado
R11	Búsqueda dentro de listado	Requisito Usuario	6	Sin comenzar
R12	Visualización de recursos MIME de una alerta	Requisito Usuario	16	Completado
R13	Utilización del modelo MVC de arquitectura	Requisito Técnico	4	Completado
R14	Código estructurado de manera limpia, organizada y bien comentada	Requisito Técnico	5	Completado
R15	Optimización del consumo de recursos de la aplicación	Requisito Técnico	6	Completado
R16	Uso de patrones de diseño	Requisito Técnico	3	Completado
R17	Demo guiada de aplicación	Requisito Usuario	2	Sin comenzar
R18	Ayuda	Requisito Usuario	4	Completado
R19	Accesibilidad de la aplicación	Requisito Usuario	2	Sin comenzar

Se puede observar que algunas de las historias están marcadas "Sin comenzar". Las historias se fueron realizando en orden de prioridad hasta agotar el tiempo disponible para el desarrollo. Se plantearon de forma intencionada algunas historias aunque se contemplaba la posibilidad de no realizar todas en el periodo de desarrollo proyectado.

Priorización de Historias

ID	Título	MoSCoW (0,1,2,3)			Prioridad
		Riesgo	Importancia para la arquitectura del sistema	Criticidad para el usuario	
R01	Listado principal básico de alertas	1	3	3	16
R02	Colores e iconos para mostrar urgencia, severidad y certeza de una alerta	1	0	2	7
R03	Estándar mensajes CAP 1.2	2	3	3	17
R04	Fuentes de mensajes CAP por URL o similar	3	2	2	13
R05	Filtrado de alertas mostradas por los campos: caducidad, categoría, urgencia, certeza, severidad	2	2	3	15
R06	Visualización resumida de una alerta con los campos más relevantes	1	3	3	16
R07	Visualización detallada de la alerta con todos sus campos	1	1	3	12
R08	Visualización en mapa de una alerta	3	2	2	13
R09	Visualización en mapa de todas las alertas	3	1	1	8
R10	Visualización de ajustes y preferencias de la aplicación	1	2	1	8
R11	Búsqueda dentro de listado	3	0	1	6
R12	Visualización de recursos MIME de una alerta	3	2	3	16
R13	Utilización del modelo MVC de arquitectura	0	2	0	4
R14	Código estructurado de manera limpia, organizada y bien comentada	1	2	0	5
R15	Optimización del consumo de recursos de la aplicación	2	2	0	6
R16	Uso de patrones de diseño	1	1	0	3
R17	Demo guiada de aplicación	2	0	0	2
R18	Ayuda	1	0	1	4
R19	Accesibilidad de la aplicación	2	0	0	2

Tareas individuales (Sprint Backlog Unificado)

ID	Estado	Tarea	Historia
T01	Completado	Crear nuevo proyecto para la aplicación	R01
T02	Completado	Crear clase de listado principal de alertas	R01
T03	Completado	Diseñar celda de información del listado principal	R01
T04	Completado	Diseñar esquema de colores asociado a combinaciones de Severidad y Urgencia	R02
T05	Completado	Diseñar incorporación del color en el listado de alertas	R02
T06	Completado	Diseñar la incorporación del color en la vista resumida de la alerta	R02
T07	Completado	Diseñar los iconos representativos de la certeza de una alerta-Niveles en barra de color	R02
T08	Completado	Diseñar la incorporación de los iconos de certeza en el listado y la vista resumida	R02
T09	Completado	Escoger clase de interpretación XML adecuada	R03
T10	Completado	Crear clases de objetos para cada elemento posible de las alertas CAP	R03
T11	Completado	Crear clase de intérprete de alertas para cargar en objetos	R03
T12	Completado	Incorporar clase de intérprete a la aplicación para que cargue las alertas en el listado	R03
T13	Completado	Buscar y estudiar fuentes existentes de alertas CAP en internet	R04
T14	Completado	Escoger las fuentes deseadas y modificar la clase de intérprete de mensajes para que las utilice	R04
T15	Completado	Incorporar en los ajustes un listado para poder seleccionar las fuentes deseadas	R04
T16	Completado	Crear una clase filtro que se convierta en la fuente de datos para la aplicación (Patrón ?)	R05
T17	Completado	Actualizar listado para incorporar botón segmentado para filtrar por caducidad	R05
T18	Completado	Incorporar en las preferencias la selección de niveles de Certeza, Urgencia y Severidad deseadas	R05
T19	Completado	Añadir una vista en el listado en la que se escoge una categoría de alertas, o todas	R05
T20	Completado	Redefinir caducidad de alertas, y nos mostrar alertas antes de fecha efectiva	R05
T21	Completado	Diseñar y crear la vista de resumen de una alerta, escogiendo los campos más relevantes	R06
T22	Completado	Incorporar esta vista para que sea accesible desde el listado de alertas para cada alerta	R06
T23	Completado	Añadir acceso a los recursos de una alerta a la vista resumida	R06
T24	Completado	Añadir acceso a las áreas de una alerta a la vista resumida	R06
	Completado	Evaluar vista resumen de alerta con experto y actualizar diseño	R06
T25	Completado	Crear las vistas necesarias para mostrar de manera jerárquica todos los campos de una alerta	R07
T26	Completado	Crear vista de mapa e incluir en barra de pestañas	R08
T27	Completado	Crear clase de conversión de datos de Area de una alerta a coordenadas en mapa	R08
T28	Completado	Enlazar Vista resumida de alerta con mapa al pulsar Area	R08
T29	Completado	Añadir elemento de interfaz para activar todas las alertas en el mapa	R09
T30	Completado	Crear vista de preferencias e incluirla en la barra de pestañas	R10
T31	Completado	Implementar almacenamiento y recuperación de valores de ajustes	R10

T32	Sin comenzar	Modificar la vista para incorporar una caja de búsqueda en la parte superior	R11
T33	Sin comenzar	Implementar el algoritmo de búsqueda	R11
T34	Completado	Añadir tratamiento de recursos "image"	R12
	Completado	Añadir tratamiento de recursos "text"	R12
	Completado	Añadir tratamiento de recursos "audio"	R12
	Completado	Añadir tratamiento de recursos "video"	R12
	Completado	Mover código de carga de datos a la vista del recurso y añadir indicador de progreso	R12
	Completado	Añadir tratamiento de errores de carga de recursos - falta para video	R12
T36	Completado	Utilizar plantillas de controladores-vista para iniciar clases nuevas	R13
T37	Completado	Generar diagramas de clases	R13
T38	Completado	Refactorizar clases para agrupar métodos y secciones de código adecuadamente	R14
T39	En curso	Revisar y añadir comentarios a todas las clases y métodos	R14
T40	En curso	Agrupar clases en paquetes o carpetas estructuradas	R14
T41	Completado	Ejecutar la aplicación con el debugger y detector de fugas de memoria, repasar gestión de memoria	R15
T42	Completado	Realizar pruebas de estrés	R15

Definición detallada de Historias de Usuario

ID:	R01
Título:	Listado principal básico de alertas
Descripción:	La aplicación debe mostrar un listado de notificaciones recibidas con su título, el emisor, el tiempo desde su emisión y alguna representación de la severidad, certeza y urgencia en forma de color o icono (ver R02)
Caso de uso:	El listado estará en formato de tabla de entradas, navegable de forma vertical mediante scroll. Será la primera vista de la barra de pestañas (que dará acceso a las demás vistas de la aplicación)

ID:	R02
Título:	Colores e iconos para mostrar urgencia, severidad y certeza de una alerta
Descripción:	Según la importancia de un mensaje se marcará con un color/icono diferente en el listado. La severidad y la urgencia se evalúan juntas para definir el color, desde rojo para lo más alto hasta azul para informativo, y gris para desconocido La importancia se define como la suma de los valores de Severidad y Urgencia, que van desde 0 a 4 respectivamente Se contempla el uso de algún icono para representar la certeza de una alerta Se contempla el uso de iconos para representar la categoría de una alerta

ID:	R04
Título:	Fuentes de mensajes CAP por URL o similar
Descripción:	Se podrán escoger fuentes de mensajes CAP definidas por una URL/URI de un listado de fuentes predefinidas. La aplicación obtendrá los mensajes de dichas fuentes para mostrar, que se mantendrá actualizado

ID:	R05
Título:	Filtrado de alertas mostradas por los campos: caducidad, categoría, urgencia, certeza, severidad
Descripción:	La aplicación filtrará los mensajes entrantes según las preferencias definidas por el usuario en relación a los siguientes parámetros: -Fecha: mostrar u ocultar caducados. Nos mostrar alertas cuya fecha efectiva no ha llegado. Considerar alertas caducadas a la semana si no incluyen fecha de caducidad -Categorías: mostrar todas las alertas o sólo las de una categoría -Urgencia, Severidad y Certeza: elegir qué valores se quieren ver
Caso de uso:	-El listado de alertas tendrá control segmentado con valores "Fecha" y "Importancia", que cambiará la ordenación de las alertas en el listado -El listado tendrá una vista anterior en la que se selecciona una categoría o todas -En la vista de ajustes, que es la última vista de la barra de pestañas, se podrán seleccionar los valores de Urgencia, Severidad y Certeza que se desean ver. además de poder ocultar alertas caducadas

ID:	R06
Título:	Visualización resumida de una alerta con los campos más relevantes
Descripción:	Se mostrará una visualización con los campos más relevantes de una alerta, con un formato que permita rápidamente interpretar la información de la misma Esta vista incluirá acceso a los recursos y áreas que acompañen a una alerta, que se cargarán en el momento de seleccionarlas

ID:	R07
Título:	Visualización detallada de la alerta con todos sus campos
Descripción:	Desde la vista resumida de la alerta se permite acceder a un listado con todos los campos de la alerta, en la jerarquía en la que se ha presentado, y sin formato alguno

ID:	R08
Título:	Visualización en mapa de una alerta
Descripción:	La vista de mapa se incluye en la barra de pestañas, y si una alerta viene acompañada de una área que defina su localización se permitirá visualizar sobre este mapa dicha información al pulsarla desde la vista resumida de la alerta

ID:	R09
Título:	Visualización en mapa de todas las alertas
Descripción:	Una vista permitirá mostrar mensajes activos/todos sobre un mapa, con los correspondientes iconos en su caso

ID:	R10
Título:	Visualización de ajustes y preferencias de la aplicación
Descripción:	La última vista de la barra de pestañas será la de ajustes y preferencias de la aplicación. Contiene: -Las preferencias de filtrado de Urgencia, Severidad y Certeza de alertas
Caso de uso:	-Las preferencias se guardan cuando se cierra la aplicación, y se establecen al abrirla

ID:	R11
Título:	Búsqueda dentro de listado
Descripción:	Se permitirá realizar una búsqueda sobre los listados de notificaciones, se mostrará una caja de búsqueda sobre el listado que irá filtrando en tiempo real según se teclean los términos

ID:	R12
Título:	Visualización de recursos MIME de una alerta
Descripción:	Dentro de la vista resumida de la alerta se dará acceso a los recursos que la acompañan. Estos recursos pueden ser de los siguientes tipos: -application -audio -example -image -message -model -multipart -text -video
Caso de uso:	-El recurso no se carga hasta que el usuario decide mostrarlo

ID:	R13
Título:	Utilización del modelo MVC de arquitectura
Descripción:	El diseño de las clases de la aplicación seguirán el modelo arquitectónico MVC (Model-View-Controller) en todo lo posible

ID:	R14
Título:	Código estructurado de manera limpia, organizada y bien comentada
Descripción:	El código debe estar refactorizado, debe seguir un mismo estándar de codificación en todas las clases además de estar correctamente dividido para aumentar la cohesión y reducir el acoplamiento. Si fuera posible se deben establecer qué clases serían fácilmente reutilizables para futuros proyectos Se comentará de manera extensa el código, y se generará la documentación necesaria para su correcta comprensión

ID:	R15
Título:	Optimización del consumo de recursos de la aplicación
Descripción:	Se someterá la aplicación a una serie de pruebas de recursos, además de utilizar mediciones para asegurar el correcto uso de recursos y evitar fugas de memoria

ID:	R16
Título:	Uso de patrones de diseño
Descripción:	Se utilizarán en la medida de lo posible patrones de diseño, especialmente siguiendo las guías de interfaces móviles de Apple y otros estándares

ID:	R17
Título:	Demo guiada de aplicación
Descripción:	Al iniciar la aplicación por primera vez se dará la opción al usuario de realizar una demo guiada de la aplicación y sus funciones

ID:	R18
Título:	Ayuda
Descripción:	La aplicación contará con un apartado denominado "Ayuda" donde se explicarán las funciones de la aplicación, así como los diferentes elementos de interfaz y su significado

ID:	R19
Título:	Accesibilidad de la aplicación
Descripción:	La aplicación permitirá ajusta elementos de la interfaz para mejorar la accesibilidad: -Cambiar el contraste de la aplicación -Aumentar el tamaño del texto mostrado