

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA INFORMÁTICA



REMERWeb

Proyecto Fin de Carrera - Ingeniería Informática

Autor: Sergio Herranz Huertas

Tutor: David Díez Cebollero

Fecha: 13 de Septiembre de 2010



"Puedes llegar a cualquier parte, siempre que camines lo suficiente."

Lewis Carroll (1832-1898). Matemático y escritor británico.

Agradecimientos

Echando la vista atrás entiendo la importancia que tiene la consecución del proyecto fin de carrera. La tiene porque pone fin a una etapa que probablemente jamás olvide, pero que también ha estado llena de esfuerzo y sacrificio no sólo mío, sino de mucha gente a la que no puedo dejar de agradecerse.

A mis padres sin cuyo esfuerzo, tesón y dedicación constante jamás hubiese tenido la oportunidad de alcanzar este logro.

A mi hermano, que pese a las discusiones habituales siempre me ha hecho notar su cariño y apoyo.

A mis compañeros de clase/trabajo/prácticas, por su ayuda constante y por hacer inolvidable mi camino universitario, haciendo que las prácticas o el trabajo se conviertan en recuerdos bonitos llenos de complicidad y anécdotas.

A mi tutor, David, porque aunque me haya cambiado el enfoque del proyecto una y mil veces y su meticulosidad me haya dado más de un quebradero de cabeza, en todo momento me ha mostrado su predisposición a ayudarme, y siempre lo ha hecho con su buen humor y sus innumerables bromas.

Y por supuesto a ti, Sara, mi mayor ánimo en los momentos duros y mi motivación por seguir adelante y alcanzar metas como esta. En definitiva, la persona que hace que logros como el presente cobren un sentido especial.



Resumen

La Red de Radio Emergencia (REMER) es una comunidad constituida por radioaficionados que prestan su colaboración de manera altruista a la Dirección General de Protección Civil y Emergencias en circunstancias excepcionales o de crisis. La eficacia demostrada por REMER en la gestión de emergencias durante más de veinticinco años ha animado a plantear la posibilidad de trasladar la actividad de esta comunidad al entorno web, proporcionando así contextos de colaboración alternativos a la radiofrecuencia. Sin embargo, la aplicación de esta idea no resulta trivial, siendo necesaria la aplicación de técnicas apropiadas de diseño que permitan establecer las necesidades y forma de trabajo de dicha comunidad.

El presente Proyecto Fin de Carrera describe el proceso seguido para el desarrollo de un prototipo de alto nivel que contemple el proceso básico de actuación de los integrantes de REMER. El propósito de este prototipo es disponer de una herramienta que pueda ser validada por los propios miembros de REMER como parte de un proceso de diseño participativo dirigido a disponer de una herramienta colaborativa que soporte la actividad de dicha comunidad. Esta herramienta colaborativa se encuadraría dentro del proyecto URThey.

A fin de establecer unas bases sólidas para el desarrollo de esta futura herramienta final ya mencionada, el prototipo sigue una serie de requisitos. Entre los mismos, destaca el seguimiento de una metodología de desarrollo web, que gracias a la amplia documentación generada va a facilitar un futuro proceso de rediseño. En cuanto al diseño, mencionar también que se aplican distintas pautas o reglas para garantizar aspectos como la usabilidad y accesibilidad. Por último, respecto a los detalles de implementación, otro requisito establecido ha sido la separación completa de contenido y estilo, y el cumplimiento de estándares en ambas partes.



Abstract

Emergency Radio Network (REMER) is a community composed by radio hams who give their altruistic collaboration to Department of Civil Defence and Protection in exceptional circumstances or crisis situations. The feasibility of REMER in emergency management for more than twenty-five years has encouraged us to transfer the activity of this community to the web context, in order to provide alternative collaborative tools.

However, the application of this idea is not trivial; it requires the application of suitable design techniques for establishing the requirements and working methods of the community.

This Final Degree Project describes the process of developing a high-level prototype that includes the basic interaction process of the members of REMER. The aim of this prototype is to have a tool that may be validated by the members of REMER as part of a participatory design process for providing a collaborative tool that supports the activity of such a community.

A goal of this prototype is to lay the foundations of a future development of the final collaborative system. For these reason, this prototype has been carried out following some quality requirements. Among these requirements we can highlight the use of a web development methodology that manages the development process. In terms of design, also it is important to mention that different guidelines for improving the usability and accessibility have been applied. Finally, about implementation details, another requirement has been the complete separation of content and style, in compliance with standards on both of them.



Índice de contenidos

Glosario de términos.....	9
1 Introducción	11
1.1 Contexto del problema	11
1.2 Planteamiento del problema	14
1.3 Objetivos	15
1.4 Estructura de la documentación	15
2 Gestión de proyecto software	17
2.1 Alcance del proyecto	17
2.2 Plan de trabajo	24
2.3 Gestión de recursos	28
2.4 Gestión de riesgos	29
2.5 Plan de pruebas.....	39
3 Revisión de tecnologías web	44
3.1 Metodología de desarrollo.....	44
3.2 Usabilidad.....	56
3.3 Accesibilidad.....	59
3.4 Estándares Web.....	63
3.5 Herramientas de evaluación	67
4 Solución	70
4.1 Descripción de la solución.....	70
4.2 El proceso de desarrollo.....	72
4.3 El producto del desarrollo.....	113
5 Evaluación.....	118
5.1 Proceso de evaluación	118
5.2 Análisis de resultados	119
6 Conclusión	121
6.1 Aportaciones realizadas.....	121
6.2 Trabajos futuros	122
6.3 Problemas encontrados	122
6.4 Opiniones personales	124
7 Bibliografía.....	125
Anexo I. Control de versiones.....	128
Anexo II. Seguimiento de proyecto fin de carrera	129



Índice de figuras

Figura 1. Planificación diagrama de Gantt.	27
Figura 2. Organigrama equipo de trabajo.	28
Figura 3. Evolución metodología.	45
Figura 4. Fases OOHDM.	45
Figura 5. Ejemplo Diseño Conceptual.	46
Figura 6. ADV y su relación con la interfaz.	47
Figura 7. Ejemplo de navegación.	49
Figura 8. Ejemplo WebML Completo.	49
Figura 9. Proceso de desarrollo ADM.	52
Figura 10. Eficacia-Eficiencia-Satisfacción.	56
Figura 11. Marco estándares web.	64
Figura 12. Regla CSS.	67
Figura 13. Generalización de actores.	86
Figura 14. Diagrama Casos de Uso General.	87
Figura 15. Diagrama Casos de Uso Coordinador.	88
Figura 16. Diagrama Casos de Uso Informador.	89
Figura 17. UID Crear Alerta.	97
Figura 18. UID Buscar alerta por fecha.	98
Figura 19. UID Enviar mensaje.	99
Figura 20. UID Editar usuario.	100
Figura 21. Diagrama conceptual general.	102
Figura 22. Diagrama conceptual usuarios.	103
Figura 23. Diagrama conceptual alertas.	104
Figura 24. Diagrama conceptual comunicación.	104
Figura 25. Esquema de clases navegacionales.	105
Figura 26. Esquema de contextos navegacionales.	106
Figura 27. ADV Alerta.	107
Figura 28. ADV Usuario por alerta.	107
Figura 29. ADV Respuestas por alerta.	108
Figura 30. ADV Usuario.	108
Figura 31. ADV Mensaje.	109
Figura 32. ADV Estadística.	109
Figura 33. Ejemplo importación css.	110
Figura 34. Estructura layout.	111
Figura 35. Mal ejemplo diseño dinámico.	112
Figura 36. Pantalla de inicio.	113
Figura 37. Pantalla gestión de alertas.	114
Figura 38. Pantalla creación de alerta paso 1.	115
Figura 39. Pantalla creación alerta paso 3.	116
Figura 40. Pantalla detalle alerta.	117
Figura 41. Planificación inicial.	130
Figura 42. Planificación final.	133



Índice de tablas

Tabla 1. Clasificación de las fases de una emergencia.....	12
Tabla 2. Distribución de tareas, recursos y dedicación.....	18
Tabla 3. Retribuciones brutas mensuales del personal.....	19
Tabla 4. Bases de cotización 2010.....	20
Tabla 5. Tipos de cotización 2010.....	20
Tabla 6. Cuotas cotización personal.....	21
Tabla 7. Coste final personal.....	21
Tabla 8. Costes tecnológicos.....	22
Tabla 9. Coste total asociado al proyecto.....	23
Tabla 10. Margen de beneficios.....	23
Tabla 11. Margen de riesgo.....	23
Tabla 12. Coste total sin IVA.....	23
Tabla 13. Coste final con IVA.....	24
Tabla 14. Planificación de tareas.....	25
Tabla 15. Fuentes de riesgo.....	30
Tabla 16. Parámetros riesgos.....	30
Tabla 17. Riesgo 01.....	31
Tabla 18. Riesgo 02.....	31
Tabla 19. Riesgo 03.....	31
Tabla 20. Riesgo 04.....	32
Tabla 21. Riesgo 05.....	32
Tabla 22. Riesgo 06.....	32
Tabla 23. Riesgo 07.....	33
Tabla 24. Riesgo 08.....	33
Tabla 25. Riesgo 09.....	33
Tabla 26. Riesgo 10.....	34
Tabla 27. Plan contingencia Riesgo 01.....	34
Tabla 28. Plan contingencia Riesgo 02.....	35
Tabla 29. Plan contingencia Riesgo 03.....	35
Tabla 30. Plan contingencia Riesgo 04.....	36
Tabla 31. Plan contingencia Riesgo 05.....	36
Tabla 32. Plan contingencia Riesgo 06.....	37
Tabla 33. Plan contingencia Riesgo 07.....	37
Tabla 34. Plan contingencia Riesgo 08.....	38
Tabla 35. Plan contingencia Riesgo 09.....	38
Tabla 36. Plan contingencia Riesgo 10.....	39
Tabla 37. Prueba aceptación 01.....	40
Tabla 38. Prueba aceptación 02.....	40
Tabla 39. Prueba aceptación 03.....	40
Tabla 40. Prueba aceptación 04.....	41
Tabla 41. Prueba aceptación 05.....	41
Tabla 42. Prueba aceptación 06.....	41



Tabla 43. Prueba aceptación 07.	42
Tabla 44. Prueba aceptación 08.	42
Tabla 45. Prueba aceptación 09.	43
Tabla 46. Resumen Ariadne.....	53
Tabla 47. Resumen métodos.	56
Tabla 48. Principales versiones XHTML.....	66
Tabla 49. Principales versiones CSS.	67
Tabla 50. Resumen herramientas evaluación.	69
Tabla 51. Requisitos Funcionales I.....	77
Tabla 52. Requisitos Funcionales II.....	79
Tabla 53. Requisitos No funcionales I.....	81
Tabla 54. Requisitos No funcionales II.	82
Tabla 55. Caso de Uso Autenticarse.	89
Tabla 56. Caso de Uso Reenviar contraseña.	89
Tabla 57. Caso de Uso Crear alerta.	90
Tabla 58. Caso de Uso Buscar alerta fecha.....	90
Tabla 59. Caso de Uso Consultar alerta.....	90
Tabla 60. Caso de Uso Editar alerta.	91
Tabla 61. Caso de Uso Cerrar alerta.....	91
Tabla 62. Caso de Uso Asociar informadores.	92
Tabla 63. Caso de uso Generar PDF alerta.	92
Tabla 64. Caso de Uso Responder alerta.....	93
Tabla 65. Caso de Uso Consultar estadísticas.....	93
Tabla 66. Caso de Uso Buscar usuario por zona.	94
Tabla 67. Caso de Uso Consultar perfil.	94
Tabla 68. Caso de Uso Editar perfil propio.	94
Tabla 69. Caso de Uso Editar usuario.....	95
Tabla 70. Caso de Uso Enviar mensaje.....	95
Tabla 71. Caso de Uso Consultar Mensajes recibidos.....	96
Tabla 72. Caso de Uso Consultar ayuda.	96
Tabla 73. Resultado pruebas.	120
Tabla 74. Control de versiones.	128
Tabla 75. Seguimiento del proyecto.	131



Glosario de términos

- **ADM:** *Ariadne Development Method*.
- **ADV:** *Abstract Data View*.
- **Alerta:** situación o aviso de posible peligro que conlleva una acción de seguimiento y vigilancia para una mejor respuesta y control.
- **Braille:** método de lectura y escritura táctil.
- **Case Sensitive:** textos en los que tiene alguna relevancia escribir un carácter en mayúsculas o minúsculas.
- **Comunidad de prácticas:** grupo de personas que comparten unas preocupaciones, una afición por un tema o un conjunto de problemas y que se comunican y comparten información para resolver dichos problemas o aumentar sus conocimientos.
- **CSS:** *Cascading Style Sheets*, CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en (X)HTML.
- **CTIC:** *Centro Tecnológico de la Información y la Comunicación*.
- **Desastre:** hecho que afecta negativamente a la vida y cuyo impacto, o posible impacto, genera una emergencia.
- **Emergencia:** situación producida debido al impacto o posible impacto de un desastre.
- **E-R:** *Entity Relationship*, herramienta para el modelado de datos de un sistema de información.
- **Gestión de Emergencias:** conjunto de decisiones político-administrativas y de intervenciones operativas que se llevan a cabo en las diferentes etapas de una emergencia de cara tanto a la anticipación como a la respuesta del mismo.
- **HDM:** *Hypertext Design Model*.
- **Interfaz de usuario:** consiste de aquellos aspectos del sistema con los que el usuario entra en contacto, físicamente, perceptivamente o conceptualmente.
- **IVA:** *Impuesto sobre el Valor Añadido*.
- **Mantenibilidad:** facilidad con la que un sistema software puede ser modificado para corregir errores.



- **MDA:** *Model Driven Architecture*.
- **MySQL:** sistema de gestión de base de datos relacional, multihilo y multiusuario.
- **RAE:** *Real Academia Española*.
- **REMER:** la Red Radio de Emergencia es la organización estructurada en el ámbito territorial nacional, constituida por los radioaficionados españoles que prestan su colaboración a los servicios oficiales de Protección Civil al ser requeridos para ello, cuando circunstancias excepcionales lo justifiquen.
- **SGML:** *Standard Generalized Markup Language*, sistema para la organización y etiquetado de documentos normalizado en 1986.
- **SMS:** *Short Message Service*, es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos entre teléfonos.
- **Timemap:** diseño gráfico que muestra en una barra horizontal marcada con fechas los eventos en los puntos donde habrían ocurrido.
- **Tooltip:** herramienta visual que funciona al situar el ratón sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra.
- **UML:** *Unified Modeling Language*, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.
- **W3C:** World Wide Web Consortium.
- **WCAG:** *Web Content Accesibility Guidelines*, son un conjunto de pautas que explican cómo hacer que el contenido Web sea accesible para personas con discapacidad.
- **Wizard:** interfaz de usuario presentada como una ventana de dialogo.
- **XHTML:** *EXtensible Hypertext Markup Language* (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web.
- **XML:** *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas muy utilizado en la actualidad.
- **XSLT:** *Extensible Stylesheet Languages Transformations*, estándar que presenta una forma de transformar documentos XML en otros formatos.



1 Introducción

Vivimos en sociedades complejas y globales en las que gestionar una emergencia se ha convertido en una actividad especialmente compleja. Este hecho ha llevado a la creación de nuevos contextos de gestión de emergencias en los cuales cada individuo debe considerarse como parte involucrada en la propia labor de gestión. Un ejemplo que pone en práctica esta idea de colaboración, en la que cada individuo adquiere un papel activo en la gestión de la emergencia, es REMER.

Basándose en esta idea, este documento describe el proceso de diseño de un prototipo de alto nivel que servirá como base para el diseño participativo de una herramienta colaborativa que soporte la actividad de los integrantes de REMER en el entorno web.

El presente capítulo introduce al lector en el trabajo acometido, presentando el contexto del trabajo, detallando la problemática que ha motivado el trabajo, así como el propósito y objetivos del mismo.

1.1 Contexto del problema

El presente trabajo se enmarca en el contexto de la gestión de emergencias. Se define la **gestión de emergencias** como el conjunto de decisiones político-administrativas y de intervenciones operativas que se llevan a cabo en las diferentes etapas de una emergencia a fin de mejorar la gestión tanto de la prevención como de la respuesta ante la misma [1]. La gestión de emergencias, tal y como se puede observar en la Tabla 1, suele dividirse en siete tareas [2], agrupadas a su vez en tres fases:

- **Antes:** en esta fase tienen lugar las actividades realizadas durante la declaración de la emergencia, es decir, antes de que se produzca el desastre, o posible desastre. Consta de las siguientes tareas: prevención, mitigación, preparación y alerta.
- **Durante:** actividades que se llevan a cabo inmediatamente después de ocurrido el desastre como respuesta al mismo. Tiene asociada únicamente la tarea de respuesta.
- **Después:** actividades posteriores al desastre que corresponden en general al proceso de recuperación. Consta de las siguientes dos tareas: rehabilitación y reconstrucción.



	<i>Tarea</i>	<i>Descripción</i>
ANTES	PREVENCIÓN	Comprende el conjunto de medidas cuyo objetivo es evitar que ocurra el desastre. Se pretende evitar que los distintos fenómenos causen desastres.
	MITIGACIÓN	Pretende aminorar el impacto del desastre, es decir, reducir los riesgos y atenuar los daños potenciales sobre la vida y los bienes.
	PREPARACIÓN	Estructura de la respuesta basada en asegurar que existan los planes y sistemas necesarios para el caso en el que el desastre se haga efectivo.
	ALERTA	Declaración formal de una probable ocurrencia de un desastre de manera cercana o inminente con el fin de tomar las precauciones específicas.
DURANTE	RESPUESTA	Acciones de evacuación, de búsqueda y rescate, de asistencia sanitaria y otras, que se realizan durante el tiempo en que la comunidad se encuentra desorganizada y los servicios básicos no funcionan
DESPUÉS	REHABILITACIÓN	Restablecimiento los servicios básicos indispensables a corto plazo e inicio de la reparación del daño causado.
	RECONSTRUCCIÓN	Reparación de la infraestructura y restauración del sistema de producción, a medio o largo plazo

Tabla 1. Clasificación de las fases de una emergencia.



En la actualidad, la complejidad y globalidad de las sociedades hace que gestionar una emergencia se haya convertido en un proceso especialmente complicado. Esto ha llevado a que en algunas de estas fases la participación de cada individuo sea fundamental, involucrándose como parte activa en la gestión de la propia emergencia.

Un ejemplo en el que la colaboración de cada individuo es clave para gestionar una emergencia es REMER. La Red Radio de Emergencia (REMER) [3] está constituida por los radioaficionados españoles que prestan su colaboración a los servicios oficiales de Protección Civil al ser requeridos para ello, cuando circunstancias excepcionales lo justifiquen, vinculándose voluntariamente y de modo altruista a la Dirección General de Protección Civil y Emergencias, una vez seguidos los trámites establecidos por la misma.

Son objetivos de REMER:

- Constituir un sistema de radiocomunicación en base a recursos privados que facilite, cuando sea necesario, la actuación de los de naturaleza pública de la Red Radio de Mando de Protección Civil, complementándola o sustituyéndola, según los casos.
- Articular un mecanismo que permita a los radioaficionados colaborar con la Dirección General de Protección Civil y Emergencias, asumiendo voluntariamente los deberes que, como ciudadanos, les corresponde en los casos en que su actuación se haga necesaria a juicio de las autoridades de Protección Civil.
- Facilitar a los radioaficionados españoles, integrados en la Red, su colaboración a nivel operativo y la coordinación entre ellos, así como la incorporación, en caso necesario, de aquellos otros radioaficionados que, no perteneciendo a la Red, sea necesario pedir su colaboración.

Una actividad significativa dentro de la comunidad REMER es la **alerta temprana (“early warning”)**. La alerta temprana es una actividad incluida dentro de la fase de alerta, encargada de suministrar información oportuna y eficaz ante un peligro declarado, para anticipar y mejorar la toma de decisiones y así reducir su riesgo y preparar una respuesta efectiva. Tal es su importancia, que la Organización Mundial Meteorológica (OMM) ha pedido “implementar sistemas de alerta temprana”, señalando estos sistemas como algo prioritario y que reduciría a la mitad para 2019 las muertes producidas en el período 1994-2003 por desastres meteorológicos [4]. Otra importante



manifestación a favor del uso de estos sistemas, la llevo a cabo en 2005 el entonces Secretario Nacional de las Naciones Unidas, Kofi Annan, que recomendó "el establecimiento de un sistema mundial de alerta temprana para todos los riesgos naturales". [5]

1.2 Planteamiento del problema

La red REMER está constituida como una comunidad en la que la colaboración de cada voluntario es esencial. Gracias a ellos y a la perfecta organización de la red, REMER se ha convertido en un instrumento sumamente eficaz al servicio de los órganos de la dirección y gestión de emergencias, como viene demostrándose durante estos últimos veintiocho años en cada ocasión en la que alguno de sus más de mil voluntarios ha sido requerido para ello. [6]

Basándonos en el éxito de REMER como precedente y punto de partida, se planteo la conveniencia de trasladar la actividad de esta comunidad, en el contexto de alerta temprana, a un entorno Web. Esto permitirá que una comunidad de prácticas como la que forma REMER pueda comunicarse y compartir información de una manera más cómoda y eficaz.

Un precedente en el intento de trasladar las ideas de REMER a entornos virtuales, puede encontrarse en SIGAME [7]. Fruto del convenio de colaboración entre la DGPCE y el grupo DEI, se solicitó llevar la actividad de la comunidad REMER al sistema web SIGAME [7]; sin embargo, a la hora de afrontar el diseño de esta funcionalidad, se constató la existencia de numerosas dudas acerca de cómo llevar a cabo las actividades de colaboración y comunicación dentro de la comunidad. Además, no se puede dejar de tener en cuenta que el concepto de comunidad que colabora en el contexto de alerta temprana es un concepto novedoso y abstracto, cuya aplicación práctica no resulta inmediata [8]. Por estas razones, el trasladar esta idea al contexto web es algo que no puede ser llevado a cabo sin tener en cuenta a la comunidad de prácticas que lo forma, y es por eso que se hace necesario llevar a cabo un diseño participativo en el que el usuario se integre por completo en el proceso de desarrollo, haciéndolo también cómplice y responsable de la solución.



1.3 Objetivos

El objetivo de este Proyecto Fin de Carrera es elaborar un prototipo web de alto nivel centrado en la interfaz de usuario que traslade la actividad de la comunidad REMER a un entorno web. Este prototipo deberá cumplir una serie de requisitos:

- **Metodología de desarrollo web:** seguir un método de desarrollo web permite alcanzar una solución de calidad. Además se obtiene una importante documentación de gran utilidad en el futuro.
- **Principios de diseño:**
 - **Usabilidad:** de esta forma se conseguirá un prototipo fácil de utilizar.
 - **Accesibilidad:** hacer los contenidos lo más accesibles posible independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios es esencial en cualquier desarrollo web.
- **Detalles técnicos de implementación:**
 - **Separación contenido-estilo:** una separación completa del estilo y contenido garantizan una mejor mantenibilidad y simplifican los cambios en el futuro.
 - **Seguir estándares:** respetar los estándares es un concepto clave en la accesibilidad y en la compatibilidad con distintos navegadores.

1.4 Estructura de la documentación

Este apartado detalla la estructura y contenidos del presente documento. Son los siguientes:

- **Capítulo 2. Gestión de proyecto software:** aquí se describe el proceso de gestión de proyecto. En esta fase se describen las tareas de las que consta el proyecto y los recursos que se tienen para solventarlas. A partir de esto, también se detalla el presupuesto y planificación del proyecto, así como la gestión de riesgos y el plan de pruebas.
- **Capítulo 3. Revisión de tecnologías web:** en este apartado se describe cual es la situación de las distintas tecnologías susceptibles de ser utilizadas a lo largo del proyecto. Se estudia también en qué pueden mejorar estas tecnologías las prestaciones del proyecto.



- **Capítulo 4. Solución:** en este capítulo se detalla el proceso de desarrollo hasta llegar a la solución final. En primer lugar se lleva a cabo una descripción genérica de la solución a realizar. A continuación se detalla el proceso de desarrollo, comenzando por el análisis en el que se trata de comprender mejor el problema, siguiendo por el diseño en el que se describe cómo se van a resolver las necesidades expresadas en el análisis, y finalizando con la implementación, en la que se describe como se han codificado las especificaciones expresadas en el diseño. Por último, se muestra algunas características del resultado final del producto.
- **Capítulo 5. Evaluación:** en este capítulo se detalla el proceso de pruebas en el que se pretende garantizar que el producto final cumple con las especificaciones establecidas.
- **Capítulo 6. Conclusión:** en este último apartado se reflejan las conclusiones del autor tras la realización del proyecto. Tiene como objetivo evaluar los objetivos alcanzados, las dificultades encontradas y proponer las líneas de trabajo futuras que mejoren el producto desarrollado.
- **Anexo I. Control de versiones:** en este anexo se detallan las distintas versiones de la memoria.
- **Anexo II. Seguimiento del Proyecto Fin de Carrera:** en este último anexo se puede observar el seguimiento del proyecto, comparando la primera planificación realizada con mi tutor con la planificación real llevada a cabo.



2 Gestión de proyecto software

La gestión de proyecto es una actividad esencial en todo desarrollo software. Esta actividad permite, dado la definición de un proyecto, organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos. En este sentido, en este punto se detallarán las estimaciones de tareas, recursos (tanto humanos como materiales) y presupuesto.

Debido a una petición expresa de mi tutor, la gestión de proyecto ha sido planteada como si se tratase de la gestión de un proyecto software real. Por esta razón, todas las estimaciones serán llevadas a cabo de una forma simulada en el que se podrá contar con todos los recursos, tanto humanos como materiales, que se estimen necesarios para realizar el proyecto. Además, todo presupuesto realista tiene como objetivo una rentabilidad, para ello se sumarán los márgenes de beneficio y riesgo a la oferta, de forma que se asegure su rentabilidad. Después de las estimaciones y planificación de tareas y recursos, se identificarán los posibles riesgos con los que el proceso de desarrollo puede encontrarse, así como medidas para la prevención, detección y gestión de estos riesgos. Para cada riesgo detectado se establece un plan de contingencia asociado cuyo objetivo es minimizar su impacto en el proyecto. Por último, se establecerá un plan de pruebas que describe la estrategia, recursos y planificación de las pruebas de cara a evaluar los objetivos y calidad del sistema construido.

2.1 Alcance del proyecto

Definición del proyecto

El proyecto consiste en elaborar un prototipo web de alto nivel centrado en la interfaz de usuario, que traslade la actividad de colaboración en el contexto de la alerta temprana de la comunidad REMER, a un entorno web. Este prototipo deberá cumplir una serie de requisitos:

- **Seguir una metodología de desarrollo web**
- **Tener en cuenta los siguientes principios de diseño:**
 - **Usabilidad**
 - **Accesibilidad**



- **Detalles técnicos de implementación:**
 - **Separación contenido-estilo**
 - **Cumplir con estándares**

Estimación de tareas y recursos

En la siguiente tabla se resumen las tareas previstas para el proyecto, los recursos asociados a dichas tareas y los porcentajes aproximados de dedicación de cada recurso a cada tarea.

Tarea	Recursos	Dedicación
Estudio Preliminar	Jefe de proyecto	10%
	Analista/Programador	90%
Análisis	Jefe de proyecto	10%
	Analista/Programador	90%
Diseño	Jefe de proyecto	10%
	Diseñador/Tester	90%
Implementación	Analista/Programador	100%
Pruebas	Jefe de proyecto	10%
	Diseñador/Tester	90%

Tabla 2. Distribución de tareas, recursos y dedicación.

Presupuesto

En esta sección se presentarán las estimaciones de coste del proyecto. Los costes estimados se han dividido en dos grupos. En primer lugar se introducirán los costes de personal, que cubrirán tanto los salarios de los empleados como las cuotas de cotización a la seguridad social española. En segundo lugar se detallarán los costes de las nuevas adquisiciones de equipos informáticos y dispositivos electrónicos así como las licencias software y conectividad a Internet.



Consideraciones previas

Según la aprobación de los Presupuestos Generales del Estado Español para el año 2010¹, el tipo impositivo general aplicable en los desarrollos de software (tipo general del IVA), sube del 16% al 18%. Por ello, para todos aquellos pagos o cobros que se encuadren en el ejercicio fiscal de 2010, se aplicará el tipo de IVA general del 18%.

Todos los costes estarán reflejados en Euros y han sido calculados aplicando redondeo clásico a dos decimales.

Tabla de retribuciones del personal

La siguiente tabla detalla la estimación del salario bruto mensual, calculado sobre un promedio de 8 horas diarias y 22 días al mes. Al no ser un proyecto de una envergadura grande, no se necesitará que cada recurso tenga dedicación completa a este proyecto. A partir de la dedicación estimada de cada recurso a cada tarea, se ha planificado el reparto de la jornada laboral. Se ha aplicado la siguiente fórmula:

$$\text{Bruto mensual} = \text{Euros/hora} \times \text{horas/día} \times \text{días/mes}$$

Cargo	Euros/Hora	Horas/Día	Días/Mes	Bruto mensual
Jefe de Proyecto	27.40	1	22	602.8
Analista/Programador	19.30	4	22	1698.4
Diseñador/Tester	19.30	3	22	1273.8
			TOTAL	3575.00€

Tabla 3. Retribuciones brutas mensuales del personal.

¹ <<http://www.sgpg.pap.meh.es/sitios/sgpg/es-ES/Presupuestos/ProyectoPresupuestos2010/Paginas/proyectoPGE2010.aspx>>



Bases de cotización contingencias comunes. Año 2010²

La siguiente tabla detalla las bases de cotización de contingencias comunes para año 2010, fijadas por el Ministerio de Trabajo e Inmigración español.

Grupo de Cotización	Categorías Profesionales	Bases mínimas	Bases máximas
1	Ingenieros y Licenciados. Personal de alta dirección no incluido en el artículo 1.3.c) del Estatuto de los Trabajadores	1.031.70 Euros/mes	3.198.00 Euros/mes
2	Ingenieros Técnicos, Peritos y Ayudantes Titulados	855.30 Euros/mes	3.198.00 Euros/mes

Tabla 4. Bases de cotización 2010.

Tipos de cotización (%). Año 2010

En el marco del sistema fiscal y tributario español, la empresa contratante debe asumir el pago de un porcentaje de la cotización al sistema de seguridad social público de cada empleado. La siguiente tabla presenta los porcentajes de cotización a la seguridad social asumibles por la empresa y por los propios trabajadores.

Contingencias	Empresa	Trabajadores	Total
Comunes	23.6%	4.7%	28.3%
Horas Extraordinarias Fuerza Mayor	12%	2.0%	14%
Resto Horas Extraordinarias	23.6%	4.7%	28.3%

Tabla 5. Tipos de cotización 2010.

Tabla de cuotas de cotización a la seguridad social

La cuota particular a ingresar es el resultado de aplicar a la base de cotización correspondiente a cada trabajador el tipo o porcentaje que cada año se establece para cada contingencia. Por tanto, puede calcularse esta cuota mediante la siguiente fórmula:

$$\text{Cuota a Ingresar} = \text{Base de Cotización} \times \text{Tipo de Cotización}$$

La siguiente tabla presenta el cálculo de la cuota total por cada trabajador participante en el proyecto.

² Fuente: <http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm?ID=36537>



Cargo	Bruto mensual	Base mínima	Base cotizada	Tipo	Cuota
Jefe de Proyecto	602.8	1.031.70	1.031.70	23.60	243.48
Analista/Programador	1698.4	855.30	1698.4	23.60	400.82
Diseñador/Tester	1273.8	855.30	1273.8	23.60	300.43

Tabla 6. Cuotas cotización personal.

Costes de Personal

La siguiente tabla muestra el coste final total que asume el proyecto por cada empleado, teniendo como duración total del proyecto, expuesto en la planificación, cinco meses.

Cargo	Bruto mensual	Cotización	Coste Total / Mes	Meses	Coste final
Jefe de Proyecto	602.8	243.48	846.28	5	4231.4
Analista/Programador	1698.4	400.82	2099.22	5	10496.1
Diseñador/Tester	1273.8	300.43	1574.23	5	7871.15
				TOTAL	22598.65€

Tabla 7. Coste final personal.

Material Tecnológico

En esta sección se presentan los costes de los diversos materiales tecnológicos adquiridos para el proyecto, así como las licencias y otros gastos de carácter general.



Descripción	Coste/ Unidad	Unidades	Coste total	Periodo de amortización (meses)	Duración del proyecto (meses)	Coste asociado al proyecto
Ordenador Portatil Toshiba Satellite L300D- 23K	499.01	3	1497.03	24	5	311.88
Ratón Logitech Optico M115	12.73	3	38.19	24	5	7.96
Impresora HP Laserjet P1505	169.85	1	169.85	24	5	35.38
Licencia Microsoft Word 2007	229.00	3	687.00	24	5	143.12
Licencia Office Project Standard 2007	599.95	1	599.95	24	5	199.98
Licencia Office Visio Standard 2007	259.95	1	259.95	24	5	54.15
Acceso a Internet	73.05 (MES)	5	584.40	--	--	365.25
Marial de papelería	N/A	N/A	155.00	--	--	155.00
HP Toner Black Laserjet P1505/1505N	68.40	1	68.40	--	--	68.40
					TOTAL	1341.12€

Tabla 8. Costes tecnológicos.



El coste total asociado al proyecto, queda reflejado en la siguiente tabla.

Descripción		Coste total
Personal		22598.65
Material tecnológico, licencias y conectividad		1341.12
TOTAL		23939.77€

Tabla 9. Coste total asociado al proyecto.

La siguiente tabla presenta el beneficio a obtener tras la realización del proyecto, calculado en base a un margen de beneficio del 10% sobre el total de costes. El hecho de no aplicar un margen de beneficio más alto viene dado por que el producto desarrollado es un prototipo que tiene como objetivo un futuro sistema final, en el que sí se espera obtener un mayor beneficio.

Gastos Totales	Margen de Beneficio	Beneficio Total
23939.77	10%	2393.98€

Tabla 10. Margen de beneficios.

Adicionalmente se aplica un margen de riesgo, sobre el total de gastos (sin incluir el beneficio). El margen de riesgo se fija en un 5%.

Gastos Totales	Margen de riesgo	Beneficio Total
23939.77	5%	1196.99€

Tabla 11. Margen de riesgo.

La siguiente tabla presenta el cálculo final (sin IVA), de la cantidad a cobrar al cliente.

Descripción		Euros
Gastos		23785.5
Beneficio		2393.98
Riesgos		1196.99
TOTAL		27376.50€

Tabla 12. Coste total sin IVA.



Con el motivo de dar facilidades al cliente el cobro se realizará en dos cuotas. La siguiente tabla presenta el cálculo final incluyendo el IVA aplicable.

Cuota	Fecha	IVA aplicable	Porcentaje del pago	Total a pagar	IVA Total	Total
Primera cuota	Comienzo del proyecto	18%	30.00%	8212.94	1478.33	9691.27
Cuota final	Finalización del proyecto	18%	70.00%	19163.55	3449.44	22612.99
					TOTAL	32304.26€

Tabla 13. Coste final con IVA.

La oferta por la realización del prototipo REMERWeb es de **TREINTA y DOS MIL TRESCIENTOS CUATRO CON VENTISEIS CÉNTIMOS, 32304.26 €** (IVA incluido).

2.2 Plan de trabajo

Identificación de tareas

Tal y como se avanzó en el apartado de introducción del presente trabajo, se va a seguir un método de desarrollo software en cascada, que cuenta con las siguientes fases o tareas: análisis, diseño, implementación y pruebas. Aparte de estas fases, se necesitará una fase previa en la que se revisen las distintas tecnologías y métodos susceptibles de ser utilizados en la ejecución del proyecto.

Planificación de tareas

La planificación de tareas es un instrumento de vital importancia ya que permite organizar el tiempo del que se dispone, ofreciendo la posibilidad de realizar un seguimiento para determinar si se están realizando las tareas en plazo y límite determinados. Pese a que la planificación puede cambiar a lo largo del desarrollo del proyecto, en cuyo caso se hará necesario replanificar, el objetivo es ajustarse en la mayor medida posible a la planificación realizada.

En la Tabla 14 se muestra el cronograma con las distintas fases del proyecto y sus tareas correspondientes, así como el tiempo estimado para realizar cada una de estas. Esta planificación ha sido realizada manteniendo las siguientes consideraciones:



- Jornada laboral de 8 horas diarias.
- Semana laboral de 5 días.

Tarea	Duración	Comienzo	Fin
Estudio Preliminar	25d	jue 01/04/10	mié 05/05/10
Introducción	5d	jue 01/04/10	mié 07/04/10
Revisión de tecnologías	20d	jue 08/04/10	mié 05/05/10
Análisis	23d	jue 06/05/10	lun 07/06/10
Definición de requisitos	7d	jue 06/05/10	vie 14/05/10
Especificación de Requisitos	16d	lun 17/05/10	lun 07/06/10
Análisis de roles y tareas	3d	lun 17/05/10	mié 19/05/10
Descripción de escenarios	4d	jue 20/05/10	mar 25/05/10
Casos de uso	5d	mié 26/05/10	mar 01/06/10
UID's	4d	mié 02/06/10	lun 07/06/10
Diseño	19d	mar 08/06/10	vie 02/07/10
Diseño Conceptual	6d	mar 08/06/10	mar 15/06/10
Diseño navegacional	8d	mié 16/06/10	vie 25/06/10
Diseño de interfaces abstractas	5d	lun 28/06/10	vie 02/07/10
Implementación	30d	lun 05/07/10	vie 13/08/10
Pruebas	5d	lun 16/08/10	vie 20/08/10
Memoria	56d	mar 08/06/10	mar 24/08/10
Gestión del proyecto software	7d	mar 08/06/10	mié 16/06/10
Conclusiones	2d	lun 23/08/10	mar 24/08/10
Presentación	6d	mié 25/08/10	mié 01/09/10

Tabla 14. Planificación de tareas.



Para una mejor comprensión en la siguiente figura se puede observar una representación gráfica de las tareas en el denominado diagrama de Gantt. Como en él se observa el proyecto tendrá comienzo el día **1 de Abril de 2010** y tiene como fecha prevista de finalización el **1 de Septiembre de 2010**, siendo por tanto la duración del proyecto de cinco meses.

También se puede observar en la imagen como se lleva a cabo el desarrollo del proyecto aplicando el método de desarrollo en cascada, en el que una tarea comienza cuando acaba la anterior, cumpliéndose así lo especificado en la introducción del presente documento.

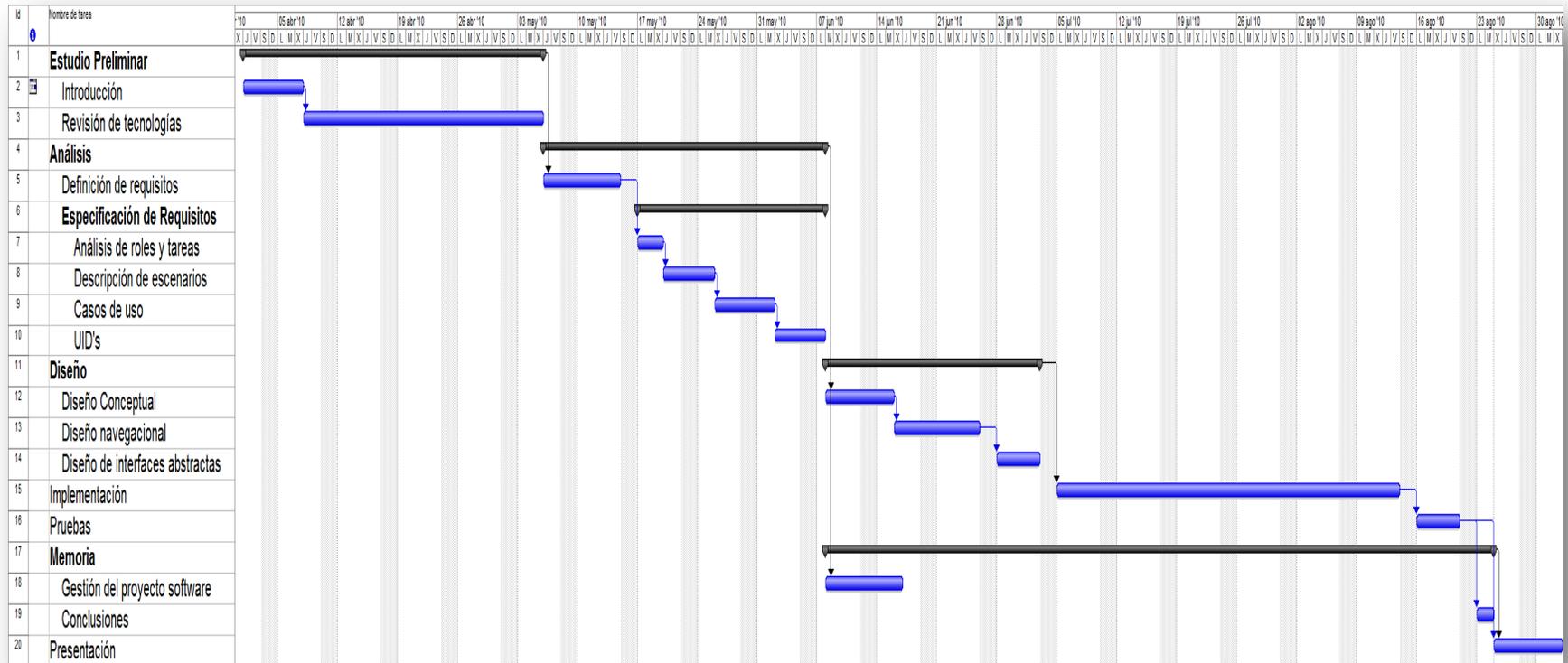


Figura 1. Planificación diagrama de Gantt.

2.3 Gestión de recursos

Especificación de recursos

En cuanto a los recursos, a lo largo del desarrollo del proyecto se necesitarán tanto recursos humanos como tecnológicos. Al tratarse de un proyecto software, serán fundamentales los recursos humanos. En este sentido es imprescindible contar con un equipo de trabajo que sea capaz de entender el dominio del problema y además se encargue de proponer soluciones que cumplan los requerimientos del cliente. En este equipo de trabajo se necesitará ocupar los siguientes roles:

- **Jefe de proyecto:** es el encargado de coordinar, dirigir, controlar e integrar al resto de las personas participantes en el grupo de trabajo.
- **Analista:** el responsable de investigar, planear, coordinar y recomendar opciones tecnológicas para cumplir con los requerimientos de la empresa.
- **Diseñador:** es el encargado de diseñar posibles soluciones al problema que se plantea, bajo la supervisión del jefe de proyecto.
- **Programador:** encargado de trasladar las especificaciones del analista y diseñador al código de la tecnología seleccionada.
- **Tester:** es el encargado de dirigir el proceso de pruebas en el que se verifique que el software cumple con los requisitos de calidad y funcionamiento establecidos.

Debido a la naturaleza del proyecto, de pequeña envergadura, se hace inviable contar con un especialista para cada rol dedicado por completo a este proyecto. A partir de la planificación y estimación de tareas se ha determinado que se cree adecuado y razonable contar con tres personas dedicadas al proyecto. Estas personas deberán ocupar a lo largo del proyecto los roles necesarios, detallados anteriormente. De esta forma, el organigrama del equipo de trabajo asociado a este proyecto quedará definido de la siguiente forma:

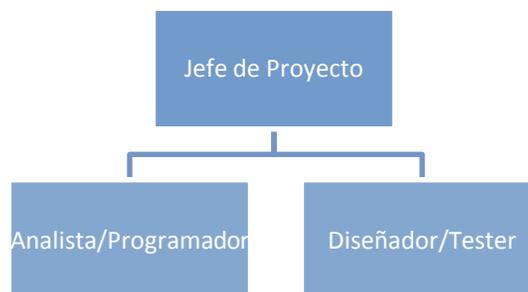


Figura 2. Organigrama equipo de trabajo.



Respecto a los recursos tecnológicos, se necesitará material de oficina y equipos informáticos con distintas licencias, en los cuales se realizará el desarrollo de la aplicación.

Asignación de recursos

Una vez se han determinado los recursos con los que cuenta el proyecto y las tareas que se tienen que llevar a cabo, se hace imprescindible realizar una asignación coherente de estos recursos. Como ya se mencionó en el punto anterior, se cuenta con tres personas que desarrollan, según el proyecto lo vaya necesitando, cinco roles:

- **Jefe de proyecto:** su dedicación al proyecto es mínima, simplemente debe supervisar la labor del resto del equipo.
- **Analista/Programador:** en la primera parte del proyecto, centrará sus esfuerzos en las tareas de estudio preliminar y análisis. Más adelante será el responsable de la implementación.
- **Diseñador/Tester:** se encargará de las fases de diseño y pruebas.

Las tres personas que forman el equipo de trabajo contarán con una jornada laboral proporcional a las tareas asignadas, lo que permitirá que ellos puedan dedicarse a más proyectos y abaratará de esta forma el presupuesto final.

2.4 Gestión de riesgos

Identificación de riesgos

El desarrollo de sistemas de software es un proceso costoso, difícil y con altos índices de fracaso. Siempre está presente de que ciertos eventos se traduzcan en complicaciones para el desarrollo del proyecto. Para intentar evitar o minimizar estas complicaciones se tratarán de identificar los posibles riesgos.

Antes de identificar los riesgos que podrán aparecer se requiere conocer cuáles son las fuentes de las complicaciones y las distintas categorías en las que dichas complicaciones pueden clasificarse. Las principales fuentes de riesgos son:



Fuente	Descripción
Naturales	Son aquellas complicaciones que pueden aparecer como consecuencia de catástrofes naturales, como terremotos, inundaciones, incendios, etc.
Internas	Son los riesgos relacionados con el desarrollo del sistema de información y con los recursos de la empresa (sin incluir al personal).
Externas	Se corresponden con las dificultades que no pertenecen a la empresa, es decir, son ajenas a ella y tampoco son naturales.
Personal	Estos riesgos están relacionados con la documentación, con asuntos personales de los propios miembros del equipo de desarrollo o con la relación con el cliente.

Tabla 15. Fuentes de riesgo.

En la siguiente tabla se detallan los parámetros que se van a especificar en cada uno de los posibles riesgos.

Campo	Descripción
Identificador	Con este campo se pretende realizar una identificación única de un riesgo. Dicho campo tiene una nomenclatura especial: <ul style="list-style-type: none">• RIESGO-XX, donde X se corresponde con el número natural consecutivo que le toca al riesgo (incluyendo el 0).
Nombre	Con este campo se le asigna un nombre común al riesgo.
Fuente	Contiene información sobre alguna de las posibles fuentes del riesgo, mencionadas con anterioridad.
Probabilidad de ocurrencia	Es la probabilidad de que el riesgo se produzca, dicha información vendrá expresada en forma porcentual.
Impacto	Indica el nivel de afectación en el desarrollo software, que supone que el riesgo se produzca. Dicho campo tiene cinco posibles niveles: <ul style="list-style-type: none">• Muy bajo• Bajo• Medio• Alto• Muy alto
Descripción	En dicha sección se realiza una redacción detallada de las características del riesgo. Con ella se pretende dar una información más específica del riesgo al cliente, que aspectos abarca, etc.

Tabla 16. Parámetros riesgos.

Riesgos naturales



RIESGO-01			
Nombre	Fallo eléctrico		
Fuente	Naturales		
Probabilidad de ocurrencia	9%	Impacto	Muy alto
Descripción	Subida de tensión, o bien falta de corriente eléctrica que imposibilita el uso de los equipos informáticos.		

Tabla 17. Riesgo 01.

RIESGO-02			
Nombre	Incendio		
Fuente	Naturales		
Probabilidad de ocurrencia	5%	Impacto	Muy alto
Descripción	Fuego provocado por cortocircuitos, subidas de tensión. Los incendios pueden ser provocados.		

Tabla 18. Riesgo 02.

Riesgos de personal

RIESGO-03			
Nombre	Problemas de comunicación con el cliente		
Fuente	Personal		
Probabilidad de ocurrencia	60%	Impacto	Bajo
Descripción	Falta de entendimiento entre el equipo de desarrollo y el cliente, como consecuencia de una falta de reuniones, de capacidad de expresión por parte del cliente, etc. Esto puede dar lugar a errores en la extracción de requisitos, en la documentación, retrasos en las entregas...		

Tabla 19. Riesgo 03.



RIESGO-04			
Nombre	Baja temporal		
Fuente	Personal		
Probabilidad de ocurrencia	35%	Impacto	Alto
Descripción	Riesgo provocado por la ausencia de algún miembro del personal como consecuencia de enfermedades, accidentes, embarazos... En definitiva toda falta que se produzca en un periodo de tiempo menor al 30% del tiempo de planificación estipulada en el proyecto		

Tabla 20. Riesgo 04.

RIESGO-05			
Nombre	Baja permanente		
Fuente	Personal		
Probabilidad de ocurrencia	5%	Impacto	Muy Alto
Descripción	Cuando un miembro del grupo es despedido; resigna o sufre un suceso que le impide trabajar con el resto del equipo de desarrollo. En definitiva toda falta que se produzca en un periodo de tiempo mayor al 30% del tiempo de planificación estipulada en el proyecto		

Tabla 21. Riesgo 05.

Riesgos externos

RIESGO-06			
Nombre	Modificación de requisitos		
Fuente	Externos		
Probabilidad de ocurrencia	65%	Impacto	Medio
Descripción	Cambio en las exigencias del cliente, o bien en el enfoque del desarrollo del proyecto.		

Tabla 22. Riesgo 06.



RIESGO-07			
Nombre	Cancelación del proyecto		
Fuente	Externos		
Probabilidad de ocurrencia	1%	Impacto	Muy alto
Descripción	El cliente decide retirar su deseo de que no se desarrolle el proyecto.		

Tabla 23. Riesgo 07.

Riesgos internos

RIESGO-08			
Nombre	Error en los requisitos		
Fuente	Internos		
Probabilidad de ocurrencia	20%	Impacto	Muy alto
Descripción	Error grave en un requisito (u omisión del mismo).		

Tabla 24. Riesgo 08.

RIESGO-09			
Nombre	Desviación/Error en la planificación		
Fuente	Internos		
Probabilidad de ocurrencia	50%	Impacto	Medio
Descripción	La asignación de trabajo y distribución del mismo para realizar la fase correspondiente es errónea. También puede aparecer como consecuencia de una desviación en la planificación.		

Tabla 25. Riesgo 09.



RIESGO-10			
Nombre	Fallos de documentación		
Fuente	Internos		
Probabilidad de ocurrencia	80%	Impacto	Medio
Descripción	Errores detectados por el cliente y/o el equipo de desarrollo al revisar el documento.		

Tabla 26. Riesgo 10.

Análisis de riesgos

En este apartado se enumeran los planes de contingencia asociados a cada riesgo identificado en el apartado anterior.

PC-RIESGO-01	
Nombre	Fallo eléctrico
Duración del plan	Dos semanas
Criterio para la ejecución del plan	Cuando no hay electricidad en el edificio
Plan de contingencia	<ul style="list-style-type: none">• Listar los desperfectos producidos por el riesgo, contabilizando las pérdidas que conlleva• Acceder a la última versión correcta que está almacenada en el backup• Recuperar la aplicación a un estado válido, sustituyendo aquella información perdida o dañada

Tabla 27. Plan contingencia Riesgo 01.



PC-RIESGO-02	
Nombre	Incendio
Duración del plan	Un mes
Criterio para la ejecución del plan	Cuando salta la alarma de incendio del edificio
Plan de contingencia	<ul style="list-style-type: none">• Listar los desperfectos producidos por el riesgo, contabilizando las pérdidas que conlleva• Acceder a la última versión correcta que está almacenada en el backup• Recuperar la aplicación a un estado válido, sustituyendo aquella información perdida o dañada

Tabla 28. Plan contingencia Riesgo 02.

Riesgos de Personal

PC-RIESGO-03	
Nombre	Problemas de comunicación con el cliente
Duración del plan	Dos semanas
Criterio para la ejecución del plan	Cuando tras varias entrevistas, no conseguimos obtener lo que el cliente quiere
Plan de contingencia	<ul style="list-style-type: none">• Formalizar nuevas reuniones con el cliente• Intentar ser más cercano al cliente• Despedir a ese miembro del equipo• Repartir la tarea entre el resto de participantes del proyecto, asumiendo los responsables del riesgo esta distribución

Tabla 29. Plan contingencia Riesgo 03.



PC-RIESGO-04	
Nombre	Baja temporal
Duración del plan	Tiempo del empleado en baja
Criterio para la ejecución del plan	Cuando el trabajador remite a la empresa su documento de baja, y esta es de corta duración
Plan de contingencia	<ul style="list-style-type: none">• Reasignación de tareas entre los miembros de la empresa por parte del Jefe de Proyecto

Tabla 30. Plan contingencia Riesgo 04.

PC-RIESGO-05	
Nombre	Baja permanente
Duración del plan	Una semana
Criterio para la ejecución del plan	Cuando el trabajador remite a la empresa su documento de baja, y esta es de larga duración
Plan de contingencia	<ul style="list-style-type: none">• Se asume la baja, repartiendo responsabilidades entre el resto del grupo encargado del proyecto

Tabla 31. Plan contingencia Riesgo 05.

*Riesgos Externos*

PC-RIESGO-06	
Nombre	Modificación de requisitos
Duración del plan	Tres semanas
Criterio para la ejecución del plan	Cuando se detecta que algún requisito no es correcto
Plan de contingencia	<ul style="list-style-type: none">• Reunión urgente• Identificación de los errores cometidos• Nuevas entrevistas con el cliente (si es necesario)• Corregir los errores identificados

Tabla 32. Plan contingencia Riesgo 06.

PC-RIESGO-07	
Nombre	Cancelación del proyecto
Duración del plan	Una semana
Criterio para la ejecución del plan	Cuando el jefe de proyecto, en consenso con el resto del grupo de trabajo, acuerdan que el proyecto es inviable
Plan de contingencia	<ul style="list-style-type: none">• Reunirse con el cliente para explicarle la situación• Llegar a un acuerdo con el cliente• Cancelar el proyecto

Tabla 33. Plan contingencia Riesgo 07.

*Riesgos Internos*

PC-RIESGO-08	
Nombre	Error en los requisitos
Duración del plan	Una semana
Criterio para la ejecución del plan	Cuando se detecta que algún requisito no es valido
Plan de contingencia	<ul style="list-style-type: none">• Recuperar la aplicación a un estado valido• Reunión urgente• Identificación de los errores cometidos• Nuevas entrevistas con el cliente (si es necesario)• Corregir los errores identificados que contiene esta versión con la versión correcta• Confirmar esta versión como la versión a partir de la cual se puede continuar de forma correcta

Tabla 34. Plan contingencia Riesgo 08.

PC-RIESGO-09	
Nombre	Desviación/Error en la planificación
Duración del plan	Un mes
Criterio para la ejecución del plan	Cuando el desarrollo del proyecto no coincide con el estimado
Plan de contingencia	<ul style="list-style-type: none">• Reunión de los responsables de los errores• Identificación de fallos cometidos en la planificación• Calculo de los errores en tiempo y dinero• Volver a planificar la sección y la asignación de trabajo

Tabla 35. Plan contingencia Riesgo 09.



PC-RIESGO-10	
Nombre	Fallos de documentación
Duración del plan	Tres días
Criterio para la ejecución del plan	Cuando algún documento que debería existir en el proyecto, no existe
Plan de contingencia	<ul style="list-style-type: none">• Reunión urgente• Identificación de los errores en la planificación• Calculo de los errores en tiempo y dinero• Volver a planificar las fases del documento

Tabla 36. Plan contingencia Riesgo 10.

2.5 Plan de pruebas

Definición de pruebas

El objetivo de este proyecto es desarrollar un prototipo de alto nivel centrado en la interfaz de usuario. El hecho de que se centre en la interfaz de usuario en lugar de en la funcionalidad hace que las pruebas unitarias y de implantación carezcan de sentido y sólo se opte por llevar a cabo sólo pruebas de aceptación.

Estas pruebas tienen como objetivo verificar que el sistema cumple los requisitos básicos de calidad esperados y permiten que el cliente determine su aceptación del prototipo. Para la especificación de las pruebas realizadas en el siguiente punto se utilizarán tablas compuestas por los siguientes campos:

- **Identificador:** identifica de forma unívoca a una prueba. Debe seguir el siguiente formato: PA-N, siendo N un número entero.
- **Descripción:** breve explicación de la prueba.
- **Proceso:** pasos que deben seguirse para la ejecución de la prueba.
- **Requisitos relacionados:** requisitos relacionados con la prueba.

Especificación de pruebas



Las pruebas de aceptación son las siguientes:

PA-01	
Descripción	Separación de estilo y contenido.
Proceso	<ol style="list-style-type: none">1. Acceder al prototipo.2. Cambiar los archivos css y dejar los mismos archivos de contenido(html)3. Volver a acceder al prototipo y verificar que el contenido es el mismo pero con otro estilo
Requisitos relacionados	RNF-05

Tabla 37. Prueba aceptación 01.

PA-02	
Descripción	Cumplimiento estándar XHTML 1.0
Proceso	<ol style="list-style-type: none">1. Acceder a la herramienta de validación de la W3C (http://validator.w3.org/).2. Introducir la url o dominio del prototipo3. Verificar que no reporta ningún error
Requisitos relacionados	RNF-06

Tabla 38. Prueba aceptación 02.

PA-03	
Descripción	Cumplimiento estándar CSS 2.1
Proceso	<ol style="list-style-type: none">1. Acceder a la herramienta de validación de la W3C (http://jigsaw.w3.org/css-validator)2. Introducir la url o dominio del prototipo3. Verificar que no reporta ningún error
Requisitos relacionados	RNF-07

Tabla 39. Prueba aceptación 03.



PA-04	
Descripción	Cumplimiento de los principios de accesibilidad web de la WCGA
Proceso	<ol style="list-style-type: none">1. Acceder a la herramienta de validación TAW basada en los principios de accesibilidad web de la WCGA (http://www.tawdis.net/)2. Introducir el dominio del prototipo3. Verificar los errores reportados
Requisitos relacionados	RNF-07

Tabla 40. Prueba aceptación 04.

PA-05	
Descripción	Cumplimiento de los principios de accesibilidad web
Proceso	<ol style="list-style-type: none">1. Acceder a la herramienta de validación Wave (http://wave.webaim.org/)2. Introducir el dominio del prototipo3. Verificar los errores reportados
Requisitos relacionados	RNF-14

Tabla 41. Prueba aceptación 05.

PA-06	
Descripción	Layout uniformes y constantes
Proceso	<ol style="list-style-type: none">1. Acceder a al prototipo2. Verificar que la disposición del layout es uniforme en las distintas páginas del prototipo3. Modificar el tamaño de la ventana4. Comprobar que el layout permanece constante y que todos los contenidos siguen estando disponibles
Requisitos relacionados	RNF-12

Tabla 42. Prueba aceptación 06.



PA-07	
Descripción	Prevención y recuperación de errores
Proceso	<ol style="list-style-type: none">1. Acceder a al prototipo2. Acceder a la tarea de creación de alerta3. Verificar que existe suficiente información a la hora de introducir un dato4. Intentar introducir algún dato erróneo5. Dejar algún dato sin introducir6. Comprobar que el prototipo no ha dejado introducir un dato erróneo e informa adecuadamente de que se ha dejado algún dato sin introducir
Requisitos relacionados	RNF-10, RNF-11

Tabla 43. Prueba aceptación 07.

PA-08	
Descripción	Feedback adecuado en el que el usuario siempre sabe en qué punto de la aplicación se encuentra
Proceso	<ol style="list-style-type: none">1. Acceder a al prototipo2. Navegar por las distintas zonas del prototipo3. Comprobar que siempre permanece visible la situación del usuario en el prototipo tanto en el breadcrumb como en el título de la página
Requisitos relacionados	RNF-08

Tabla 44. Prueba aceptación 08.



PA-09	
Descripción	Feedback adecuado cuando el usuario lleva a cabo una tareas
Proceso	<ol style="list-style-type: none">1. Acceder a al prototipo2. Acceder a la tarea de creación de alerta3. Comprobar que el usuario siempre conoce cuantos pasos tiene la tarea y en cuál de ellos se encuentra
Requisitos relacionados	RNF-08

Tabla 45. Prueba aceptación 09.

PA-10	
Descripción	Compatibilidad con distintos navegadores
Proceso	<ol style="list-style-type: none">1. Acceder a al prototipo en distintos navegadores2. Probar la tarea de creación de una alerta en distintos navegadores3. Comprobar que el comportamiento y estilo es el mismo en todos los navegadores y el usuario siempre puede llevar a cabo la tarea
Requisitos relacionados	RNF-16



3 Revisión de tecnologías web

Las aplicaciones desarrolladas para la web, como es el caso de los sistemas de alerta temprana, tienen características especiales que hacen que los mecanismos de ingeniería empleados sean distintos. En esta dirección, se define la ingeniería de la web como la disciplina que estudia el proceso de creación, implantación y mantenimiento de aplicaciones y sistemas web de alta calidad [9]. Esta disciplina, de carácter relativamente moderno (1998), está orientada a solucionar los problemas derivados de una proliferación de sistemas web de baja calidad, realizados con una carencia completa de proceso. Esta posible carencia de calidad y proceso, hace necesaria la presente revisión de las tecnologías web que permita establecer una serie de requisitos mínimos que debe cumplir la solución desde un punto de vista tecnológico.

Este apartado se dividirá en varios bloques, identificados con diversas categorías de estudio dentro del contexto de ingeniería de la web. En primer lugar se estudiarán distintos métodos de desarrollo de aplicaciones web, con la finalidad de ayudar a seleccionar cuál es el que mejor se ajusta a la problemática a resolver. En los dos siguientes puntos, se detallarán los conceptos de usabilidad y accesibilidad que más adelante serán perseguidos en la solución. A continuación, se llevará a cabo una revisión de los distintos estándares web con el objetivo de, posteriormente, elegir los estándares más adecuados para cumplir los objetivos de la solución, en sus versiones más estables. Por último, se realizará una revisión de las distintas herramientas de evaluación, que más adelante servirán para verificar que la solución cumple los requisitos tecnológicos establecidos.

3.1 Metodología de desarrollo

En los últimos años existe una tendencia a considerar el desarrollo web con un enfoque de proceso de ingeniería (del software), lo que ha llevado al desarrollo de distintos métodos de diseño web que van a permitir a los desarrolladores conseguir productos web de alta calidad, reusables y fáciles de mantener.

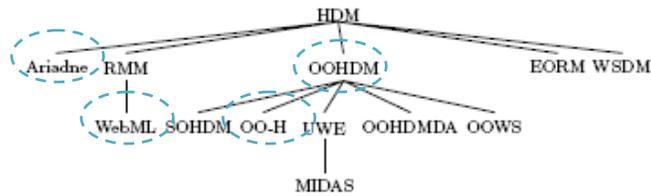


Figura 3. Evolución metodológica.

De los distintos métodos enfocados al desarrollo web que son mostrados en la [Figura 3](#), van a ser objeto de estudio algunos de los más conocidos y utilizados como OOHDM y WebML; y otros que aunque no tengan tanta influencia y se haya encontrado menos literatura merecen la pena ser estudiados, como son Ariadne y OO-H. Con el estudio de estos métodos, se podrá obtener una referencia básica del estado en el que se encuentran dichos métodos de desarrollo web de una forma comparativa.

OOHDM

Object Oriented Hypermedia Design Method (OOHDM), es un método orientado a objetos que extiende de HDM y que propone un proceso de desarrollo de cuatro fases donde se combinan notaciones gráficas UML con otras propias del método. Las cuatro fases son las siguientes [10]:



Figura 4. Fases OOHDM.

- **Diseño Conceptual:** se construye un modelo del dominio de la aplicación, a través de las técnicas del modelado orientado a objetos. Se identifican las clases y sus relaciones, que pueden ser de asociación, agregación, composición, y generalización-especialización. Tal y como se observa en la [Figura 5](#), el resultado de esta etapa es un modelo estructural similar al diagrama de clases de Unified Modelling Language (UML).

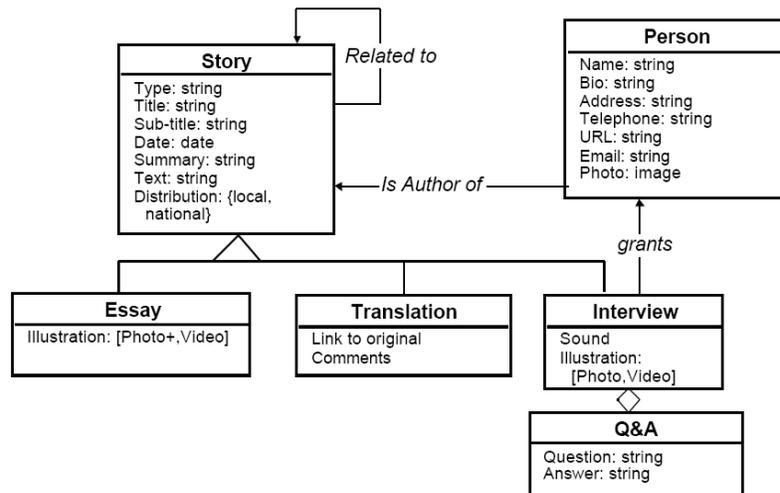


Figura 5. Ejemplo Diseño Conceptual.

- **Diseño Navegacional:** en esta etapa se pretende desarrollar una topología navegacional que permita a la aplicación ejecutar todas las tareas requeridas por el usuario. El modelo navegacional está integrado por el diagrama de clases navegacionales y el diagrama de contextos.

El diagrama de clases navegacionales se obtiene a partir del diagrama conceptual y define los objetos navegables de la aplicación hipertexto, cuyas clases reflejan la vista elegida por el dominio de aplicación. En OOHDM hay varios tipos de clases navegacionales predefinidas: nodos, links y estructuras de acceso. Los nodos semánticos y links son usuales en las aplicaciones de hipertexto, y las estructuras de acceso representan posibles maneras de acceder a nodos.

El diagrama de contextos organiza el espacio de navegación en conjuntos homogéneos que pueden ser recorridos en un orden particular; ellos deberían definirse de tal modo que contribuyan a que el usuario realice su tarea prevista.

- **Diseño de la interfaz abstracta:** en esta etapa se define la forma en la que serán percibidos los objetos a través de la interfaz de usuario y también la apariencia que tendrán. La separación de diseño navegacional y de la interfaz de usuario permite dividir las tareas del desarrollo, así como tener diferentes interfaces para un mismo modelo navegacional. En OOHDM se utilizan vistas abstractas de datos (abstract data views, ADV). Mediante un ADV se representa la estructura estática de la interfaz, la composición de



objetos y los eventos a los que responden. En la Figura 6, se puede observar un ejemplo de correspondencia entre los elementos de la interfaz y los elementos de un ADV.

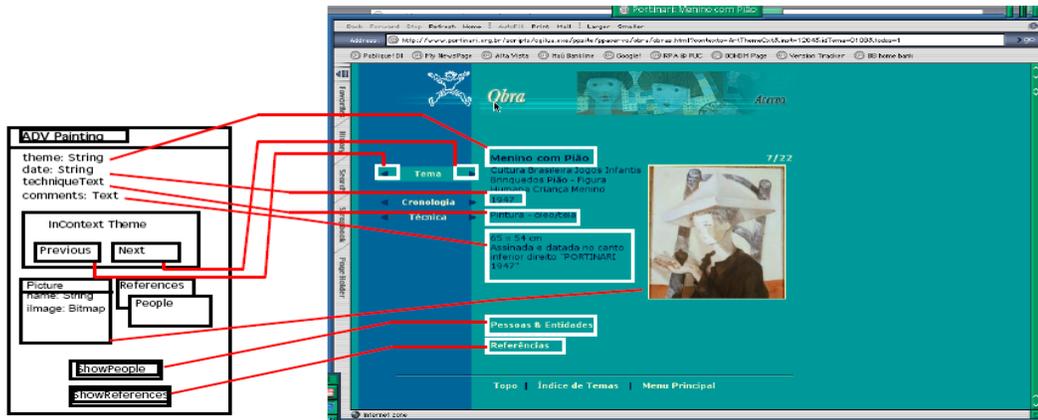


Figura 6. ADV y su relación con la interfaz.

- **Implementación:** es la última etapa, en la que, a partir de los modelos diseñados, se deben escoger las correspondencias con los objetos concretos de la plataforma de implementación.

Inicialmente, OOHDM contaba con las cuatro fases que han sido detalladas anteriormente. Sin embargo, en 2001 recibió una propuesta orientada a la ingeniería de requisitos denominada User Interaction Diagrams (UID) [11]. Esta propuesta añadía la obtención de requisitos como una fase previa al diseño conceptual, y al ser esta fase de obtención de requisitos algo crucial en cualquier proyecto informático, subdividía esta etapa en las siguientes subetapas [12]:

- **Identificación de roles y tareas:** esta etapa consiste en identificar los diferentes roles que podrían cumplir cada uno de los potenciales usuarios de la aplicación así como sus tareas.
- **Especificación de escenarios:** cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea.
- **Especificación de casos de uso:** en esta etapa se representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes.
- **Especificación de UIDs:** en esta fase se representa de forma rápida y sencilla los casos de uso generados en la etapa anterior organizado en interacciones.



- **Validación de los casos de uso y de los UIs:** en esta última etapa el desarrollador deberá interactuar con cada usuario para validar los casos de uso y UIs obtenidos, mostrando y explicando cada uno de ellos para ver si el o los usuarios están de acuerdo.

La principal ventaja de este método viene dada por su alto grado de aceptación y uso que hacen de él un modelo experimentado y con recorrido. Además cuenta con una gran completitud en la documentación en la que el usuario es el elemento esencial para la validación del producto. Es conveniente usar este método con aplicaciones con una extensión media-alta, aunque diversos autores han propuesto herramientas para simplificar la más tediosa de sus fases que es el modelo navegacional.

WebML

WebML utiliza XML para representar los modelos generados en cada etapa del desarrollo, lo que permite que sean fácilmente consumidos por generadores de código fuente que usan transformaciones XSLT. WebML cuenta con una herramienta de desarrollo llamada WebRatio que permite construir sus modelos y generar el código de la aplicación hipertexto mediante transformaciones XSLT [13].

El proceso de desarrollo que propone se compone de la especificación de requisitos, el diseño del modelo de datos E-R, diseño del hipertexto, diseño de la arquitectura e implementación, cerrando el ciclo con una etapa de evaluación y pruebas. Como el proceso es iterativo, cada una de las etapas del proceso puede retroalimentar a las anteriores. Las etapas de desarrollo de WebML se centran en la construcción de cuatro modelos:

- **Modelo estructural:** representa las entidades y atributos del dominio del problema en notación E-R. Los datos redundantes pueden ser fácilmente especificados usando WebML-OQL.
- **Modelo de hipertexto:** llamado site view, que describe las vistas del modelo estructural que estarán publicadas en el sitio web. El site view se divide en dos modelos, el de navegación y el de composición.
El modelo de composición define qué páginas estarán contenidas en el sitio web, y qué contenido tiene cada página.



El modelo de navegación sirve para especificar los enlaces entre páginas, tal y como se muestra en la [Figura 7](#).

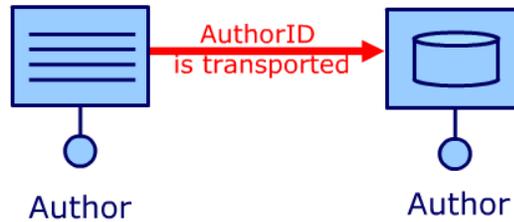


Figura 7. Ejemplo de navegación.

- **Modelo de presentación:** se efectúa a través de la aplicación de hojas de estilo XSL a documentos XML que representan una instancia del modelo navegacional, y el resultado de la transformación es un documento HTML
- **Modelo de personalización:** tiene en cuenta las operaciones de autenticación y las características o relaciones de cada usuario o grupo de usuarios a nivel de diseño de datos.

En la siguiente imagen se puede observar un ejemplo completo en WebML, en el que se puede comprobar cómo en este método se sigue una notación particular que no tiene ninguna relación con UML o con otras notaciones conocidas.

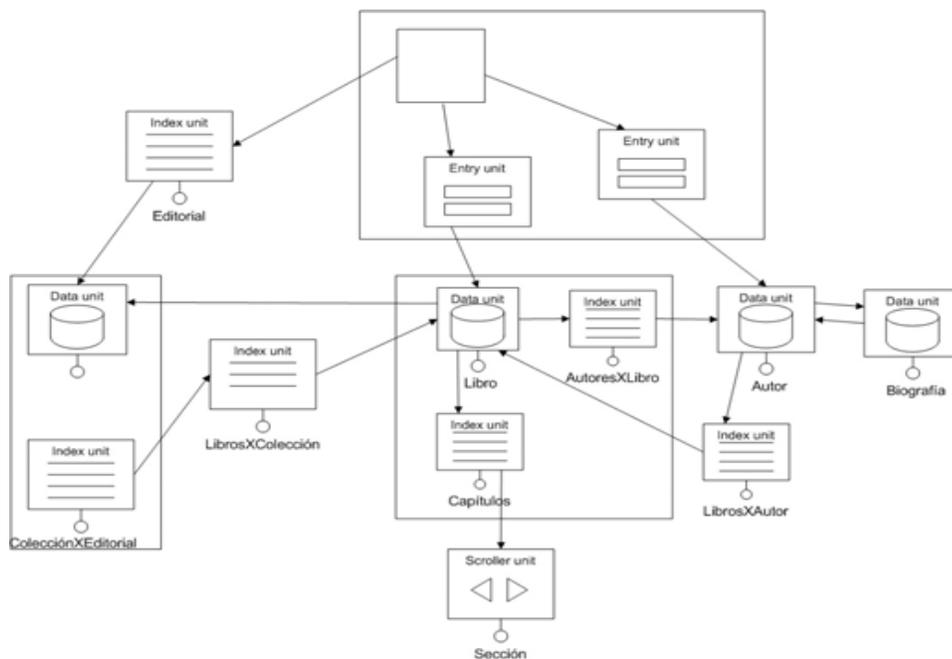


Figura 8. Ejemplo WebML Completo.



La principal ventaja de este método es que utiliza XML, lo que permite que sus productos sean consumidos por generadores de código que utilizan este estándar. Otras ventajas son que posee una notación bien definida y que cuenta con una herramienta que genera el código de la aplicación mediante transformaciones XSLT. Sobre la conveniencia de su uso, WebML es especialmente interesante cuando se trata con aplicaciones grandes que tratan con gran cantidad de información.

OO-H

OO-H, Object Oriented Hypermedia Method, inicia su análisis mediante casos de uso. OO-H reconoce que además de la navegación que procede de la estructura de datos, hay otra derivada de la funcionalidad del sistema, y por ello, clasifica la navegación en estructural y semántica. OO-H puede verse como una extensión a OO-HDM con casos de uso y enlaces de servicio. De esta forma, a través de casos de uso, y su técnica de encadenar la ejecución de éstos, llega a establecer un modelo parecido al de proceso [14].

El proceso de OO-H se compone de las siguientes fases:

- **Análisis del sistema:** los requisitos del usuario son plasmados en casos de uso; a partir de ellos, a través de técnicas conocidas de análisis orientado a objetos se deriva el modelo estructural, que se representa como un diagrama de clases UML.
- **Definición de unidades semánticas:** un grupo de casos de uso que tiene como fin realizar un determinado requisito es llamado unidad semántica de navegación. Para cada grupo de usuarios, se determina en qué unidades semánticas participa; las unidades semánticas a las que puede navegar un usuario se agrupan a través de un menú.
- **Diseño navegacional:** la navegación semántica se deriva de los casos de uso en los que participa el usuario, mientras que la navegación estructural se deriva de las relaciones de los datos. La navegación estructural no se refleja en un cambio en la intención de efectuar un cierto caso de uso. El orden de navegación entre las unidades semánticas se define mediante los enlaces semánticos. Este modelo es llamado modelo de análisis de la navegación.



- **Diseño de presentación:** la presentación es diseñada con vistas abstractas de datos, las cuales definen una jerarquía de elementos gráficos que permiten diseñar la presentación de las clases navegacionales.

Al ser considerado una extensión de OOHDM, conserva muchas de las ventajas de éste y al añadir los casos de uso llega a establecer un modelo similar al de proceso. El problema viene dado porque al modelar la interacción de diferentes usuarios puede ser mejor utilizar el propio modelo de proceso porque tiene una notación más expresiva para el flujo de control.

ADM

ADM tiene como enfoque principal el modelado de contenido multimedia, de requisitos funcionales, de usuarios y de políticas de seguridad. Esto es remarcado porque Ariadne considera que los métodos anteriores no les tienen en cuenta suficientemente.

ADM combina el proceso de diseño centrado en el usuario y flexible con una aproximación de ingeniería de modelos hipermedia basado en el proceso de desarrollo MDA, puesto que los productos resultantes de cada fase son modelos de diferentes vistas de diseño (estructura, navegación, presentación, comportamiento, procesos y acceso), abordadas desde diferentes niveles de abstracción. Por lo tanto, una ventaja de ADM es que no está orientado a la implementación en una tecnología específica, sino que ofrece productos de diseño generales que pueden ser trasladados a otras tecnologías en la fase de implementación [15].

En cuanto a su sintaxis y semántica, se podría decir que están bien definidas proporcionando de esta manera una independencia entre el sistema y su implementación. Ambas son proporcionadas por el meta-modelo Labyrinth, cuyas entidades y relaciones representan los componentes de cualquier sistema hipermedia (nodos, contenidos, enlaces, anclas, atributos, eventos y usuarios) así como las construcciones básicas (función de localización, mecanismo de composición, sistema de anclado, restricciones temporales y espaciales) ofreciendo un lenguaje común válido para diferentes plataformas de implementación.

Tal y como se observa en la [Figura 9](#), el proceso de desarrollo de Ariadne cuenta con las siguientes fases, que no tienen necesariamente que seguir un orden secuencial:

- **Diseño conceptual:** está enfocado en la identificación de tipos abstractos de datos de componentes, funciones y relaciones, pero siempre de una forma abstracta.



- **Diseño detallado:** especificación muy concreta de características y comportamiento del sistema, a un nivel de detalle tal que se pueda trasladar de manera semi-automática a la implementación
- **Evaluación:** se verifica la validez y usabilidad del sistema y prototipos con el objetivo de mejorar el diseño conceptual o detallado.



Figura 9. Proceso de desarrollo ADM.

Cada fase está compuesta por una serie de actividades que generarán una serie de productos, como se puede apreciar en la siguiente tabla (Tabla 46).



Fase	Actividad	Producto
Diseño Conceptual	Definición de la estructura lógica Estudio de la funcionalidad Especificación de entidades Modelado de usuario Políticas de seguridad	Diagrama estructural Diagrama navegacional Especificación funcional Diagramas internos Catálogo de atributos Catálogo de eventos Diagrama de usuarios Tabla de categorías Tabla de acceso
Diseño Detallado	Identificación de instancias Especificación funciones de acceso Especificación de instancias Definición de las características de presentación	Diagrama nodos de instancia Diagrama usuarios instanciados Especificación de estructura Especificación de funciones Diseño detallado de los diagramas internos Reglas de autorización Asignación de usuarios Especificación de la presentación
Evaluación	Desarrollo del prototipo Evaluación	Prototipo Documento de evaluación Reporte de conclusiones

Tabla 46. Resumen Ariadne.

Destacar que ADM cuenta con una herramienta de automatización, AriadneTool, que permite llevar a cabo transformaciones semi-automáticas y asistidas entre los diferentes niveles de abstracción, proporcionando así la generación de prototipos ligeros en lenguajes de marcado como HTML, XML, SMIL y RDF/S.



Como ventajas de este método cabría mencionar que hace especial énfasis en aspectos importantes como los requisitos funcionales, de usuarios y de políticas de seguridad. Además cuenta con un proceso sistemático, definido en fases bien delimitadas, y flexible puesto que la secuencia se deja a elección del desarrollador. Por último destacar que la independencia de plataforma se ha conseguido derivando los productos de diseño como abstracciones de un modelo formal para hipermedia denominado Labyrinth.

Comparativa

Una vez se han detallado los distintos métodos que se van a estudiar en el presente estado de la cuestión, se va a pasar a realizar una comparativa que establezca las similitudes y diferencias entre cada uno de ellos. Puesto que todos parten de HDM, todos siguen de alguna u otra forma algunos pasos básicos: modelo conceptual, diseño navegacional, diseño de la presentación e implementación.

Respecto al modelo conceptual, WebML lo lleva a cabo mediante un modelo relacional con notación E-R. En cambio, OOHDm y OO-H son métodos orientados a objetos que siguen una notación UML. Este hecho de seguir una notación tan aceptada y entendida como es UML puede ser visto como una ventaja respecto a otros métodos. En cuanto a Ariadne, su diseño conceptual contiene varias actividades y elementos que siguen su propia notación. El hecho de modelar cosas como los requisitos de políticas de seguridad y usuarios, que otros métodos no hacen, conlleva que se obtenga un modelo más completo aunque también más complejo y extenso.

Sobre el diseño navegacional cabe destacar que cada método define su propia notación. La de OOHDm está muy consolidada aunque carece de un metamodelo bien definido. Por otro lado, la notación de WebML y Ariadne parecen estar definidas de una forma más clara, aunque cuentan con la desventaja de no ser muy seguidas y reconocidas.

En cuanto al diseño de la presentación todas siguen o parten de la idea de interfaces abstractas propuesta por OOHDm. La implementación aunque se menciona en la mayoría de los métodos como una fase, no es abordada por ninguna de ellas. Sólo en algún caso, se cuenta con alguna herramienta que permite generar código ejecutable.



En la siguiente tabla se muestra un resumen de los métodos estudiados en este punto así como de las fases con las que cuenta, los objetos de salida de cada fase y las herramientas que existen para su implementación.

Método	Fases	Salidas	Herramientas
Ariadne	Diseño Conceptual	Diagrama estructural Diagrama navegacional Especificación funcional Diagramas internos	Ariadne Tool
	Diseño Detallado	Diagrama nodos de instancia Diagrama usuarios instanciados Prototipo	
	Evaluación	Reporte de conclusiones	
OO-H	Análisis con casos de uso	Casos de uso	Visual-Wade
	Modelo conceptual		
	Diseño navegacional	Diagrama de clases	
	Diseño de presentación	Diagrama navegacional	
	Implementación	ADV	
OOHDM	(Identificación de roles y tareas)		OOHDM-WebHyperSD
	(Descripción de escenarios)		
	(Especificación casos de uso)		
	(Especificación UIDs)		
	Modelo conceptual	Diagrama de interacción	
	Diseño navegacional	Diagrama de clases Diagrama de contextos	
	Diseño de presentación	Diagrama navegacional	



WebML	Implementación	ADV	
	Análisis de requisitos		WebRatio
	Diseño conceptual de datos	Diagrama E-R	
	Diseño de hipertexto		
	Diseño de arquitectura	Site View	
	Implementación		
	Pruebas		
	Puesta en producción		
Mantenimiento			

Tabla 47. Resumen métodos.

3.2 Usabilidad

El concepto de usabilidad tiene como factores principales la facilidad de aprendizaje, la efectividad en el uso y la satisfacción con la que un usuario es capaz de realizar sus tareas en un determinado sistema. Con este enfoque, se define usabilidad en la ISO 9241-11 como la medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.



Figura 10. Eficacia-Eficiencia-Satisfacción.

Otra definición muy aceptada es la que propone Jakob Nielsen, para muchos el padre de la usabilidad, que la define como un atributo de calidad que mide lo fáciles de usar que son las interfaces web. Este autor valora la usabilidad en base a las siguientes cualidades [16]:



- **Facilidad de aprendizaje:** define en cuánto tiempo un usuario, que nunca ha visto una interfaz, puede aprender a usarla bien y realizar operaciones básicas.
- **Eficiencia:** determina la rapidez con que se pueden desarrollar las tareas, una vez que se ha aprendido a usar el sistema.
- **Facilidad de recordar cómo funciona:** se refiere a la capacidad de recordar las características y forma de uso de un sistema para volver a utilizarlo a futuro.
- **Frecuencia y gravedad de errores:** plantea el apoyo que se le entrega a los usuarios cuando se enfrentan a los errores que cometen al usar el sistema.
- **Satisfacción:** indica lo satisfechos que quedan los usuarios cuando han usado el sistema, gracias a la facilidad y simplicidad de uso de sus pantallas.

A partir de esta concepción de usabilidad, Nielsen también define las diez reglas heurísticas más importantes, a su juicio, para obtener un sistema usable:

- **Visibilidad del estado del sistema:** el sistema debe siempre mantener a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.
- **Utilizar el lenguaje de los usuarios:** el sistema debe hablar el lenguaje de los usuarios, con las palabras, las frases y los conceptos familiares, en lugar de que los términos estén orientados al sistema. Utilizar convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico.
- **Control y libertad para el usuario:** los usuarios eligen a veces funciones del sistema por error y necesitan a menudo una salida de emergencia claramente marcada. Es importante disponer de deshacer y rehacer.
- **Consistencia y estándares:** los usuarios no deben tener que preguntarse si las diversas palabras, situaciones, o acciones significan la misma cosa. En general siga las normas y convenciones de la plataforma sobre la que está implementando el sistema.
- **Prevención de errores:** es importante prevenir la aparición de errores mejor que generar buenos mensajes de error.



- **Minimizar la carga de la memoria del usuario:** el usuario no debería tener que recordar la información de una parte del diálogo a la otra. Es mejor mantener objetos, acciones, y las opciones visibles que memorizar.
- **Flexibilidad y eficiencia de uso:** las instrucciones para el uso del sistema deben ser visibles o fácilmente accesibles siempre que se necesiten. Es importante que el sistema permita personalizar acciones frecuentes.
- **Los diálogos estéticos y diseño minimalista:** no deben contener la información que sea inaplicable o se necesite raramente. Cada unidad adicional de la información en un diálogo compete con las unidades relevantes de la información y disminuye su visibilidad relativa.
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** que los mensajes de error se deben expresar en un lenguaje claro, se debe indicar exactamente el problema, y deben ser constructivos.
- **Ayuda y documentación:** aunque es mejor si el sistema se puede usar sin documentación, puede ser necesario disponer de ayuda. Ésta tiene que ser fácil de buscar, centrada en las tareas del usuario, teniendo información de las etapas a realizar y que no sea muy extensa.

Todas estas guías van a ayudar, según Nielsen, a contar con un sistema más usable y por tanto a disfrutar de los beneficios que esto conlleva. Pese a que Nielsen es en cuanto a la usabilidad el autor más reconocido, existen otros autores que también han aportado mucho a este tema, como es el caso de Shneiderman. Este autor enunció las siguientes ocho reglas de oro para el diseño [17]:

- 1. Luchar por la coherencia:** secuencias coherentes de acciones han de ser necesarias en situaciones similares, terminología idéntica debería ser utilizada en avisos, en menús y pantallas de ayuda, y la coherencia en los comandos debe ser empleada en todas partes.
- 2. Permita a los usuarios frecuentes utilizar accesos directos:** si la frecuencia de uso aumenta, también lo hacen los deseos del usuario por reducir el número de interacciones y aumentar el ritmo de la interacción.



3. Ofrecer retroalimentación informativa: por cada acción del operador, debe haber alguna información del sistema. Para las acciones frecuentes y de menor importancia, la respuesta puede ser modesta, mientras que para las acciones infrecuentes y graves, la respuesta debe ser más sustancial.

4. Diseño de diálogo en el cierre de una acción: las secuencias de acciones deben ser organizadas en grupos con un principio, medio y final. La retroalimentación informativa en la realización de un conjunto de acciones da a los operadores la satisfacción de logro, una sensación de alivio.

5. Ofrecer una manipulación simple de errores: en la medida de lo posible, diseñar el sistema para que el usuario no pueda llevar a cabo un error grave. Si se comete un error, el sistema debe ser capaz de detectar el error y ofrecer de manera simple mecanismos comprensibles para su manejo.

6. Permitir una “vuelta atrás” simple de las acciones: esta prestación alivia el la ansiedad, ya que el usuario toma conciencia de que los errores se pueden deshacer, y esto lo alienta a la exploración de opciones desconocidas.

7. Apoyo interno a un enfoque de control total: los usuarios experimentados desean el sentido de que están a cargo del sistema y que el sistema responde a sus acciones.

8. Reducir la carga de memoria a corto plazo: la limitación de procesamiento de la información humana en la memoria a corto plazo, exige que las pantallas se muestren sencillas.

3.3 Accesibilidad

“El poder de la Web está en su universalidad. Un aspecto esencial es el acceso para todo el mundo sin importar la discapacidad” (Tim Berners-Lee, Director de la W3C e inventor de la World Wide Web). Este aspecto esencial no siempre se cumple, ya que aunque el uso de Internet permite el acceso a todo tipo de recursos, servicios, cultura, educación..., Internet no es aún una herramienta para todos puesto que su desarrollo está manifestando el hecho de que las barreras habituales en nuestra sociedad se manifiestan también en este medio.



Accesibilidad significa proporcionar flexibilidad para acomodarse a las necesidades de cada usuario y a sus preferencias y/o limitaciones [18].

Los seres humanos son diferentes entre sí y en un mundo ideal todas las interfaces de usuario deberían acomodarse a esas diferencias de tal modo que cualquier persona fuera capaz de utilizarlas sin problemas, sin que nadie se vea limitado en el uso de algo por causa de esas diferencias personales. Es necesario evitar diseñar solamente atendiendo a características de grupos de población específicos, imponiendo barreras innecesarias que podrían ser evitadas prestando más atención a las limitaciones de estos.

Esta accesibilidad total en la que un usuario con cualquier tipo de discapacidad pueda acceder a todos los contenidos, no es algo que se busque o esté dentro de los objetivos del presente proyecto. La accesibilidad también puede beneficiar a personas sin discapacidades, por ejemplo a personas que no pueden acceder a un contenido web porque usan un navegador determinado, o tienen una conexión lenta, o simplemente no dominan el idioma en el que están redactados los contenidos. Por eso, y debido a la naturaleza de la problemática, la accesibilidad a la que se va a enfocar el proyecto, tiene más que ver con una accesibilidad relacionada con los dispositivos donde se muestran los contenidos web, que con una accesibilidad relacionada con discapacidades de usuarios.

El W3C, y en especial su grupo de trabajo Web Accessibility Initiative (WAI), proporcionan una serie de principios denominados Pautas de Accesibilidad Web (WCAG) que ayudan al desarrollador a hacer contenidos accesibles. En 1999 se publicó la versión 1.0, que con el paso del tiempo se ha convertido en un referente internacionalmente aceptado. En diciembre de 2008 fue aprobada la versión 2.0, que se basa en la primera, de la que toma la mayoría de sus principios de accesibilidad con la diferencia de que los acerca más a las nuevas tecnologías que podemos encontrar hoy en la web. Estas pautas tienen el propósito de hacer la web más accesible al usuario, cualquiera que sea la aplicación o plataforma que se está utilizando o las limitaciones bajo las que opere.

La evaluación de la accesibilidad de un sitio web puede ser llevada a cabo con diversas herramientas basadas en diversas normativas. Una de las herramientas más aceptadas y más reconocidas es TAW (Test de Accesibilidad Web), que es una herramienta, desarrollada por la



Fundación CTIC, que permite comprobar de forma automática ciertos aspectos de la accesibilidad Web. Para el desarrollo del análisis automático, el TAW se basa en las ya nombradas Pautas de accesibilidad al Contenido Web, en concreto en la versión 1.0 WCAG del W3C. Dichas pautas son 14, con 65 puntos de verificación, divididos a su vez en 3 prioridades:

- **Prioridad 1:** hace referencia a los puntos de verificación “obligatorios”, aquellos que el desarrollador tiene que satisfacer; si no, algunos grupos de personas serán incapaces de acceder al contenido de un sitio web.
- **Prioridad 2:** el desarrollador debería satisfacerla; sin ello alguien encontrará muchas dificultades para acceder a la información. Satisfacer este punto salvará importantes barreras para acceder al contenido web.
- **Prioridad 3:** el desarrollador puede satisfacerla; de lo contrario, algunas personas hallarán dificultades para acceder a la información. Ello mejoraría notablemente el acceso a los documentos web.

La especificación tiene tres "niveles de adecuación" con estas directrices:

- El nivel "A": se satisfacen todos los puntos de prioridad 1.
- El nivel "AA": se satisfacen todos los puntos de prioridad 1 y 2.
- El nivel "AAA": se satisfacen todos los puntos de prioridad 1, 2 y 3.

Destacar que también existe una versión beta del TAW (Test de Accesibilidad Web) para la versión 2.0 de las Pautas de accesibilidad al contenido web.

Muy probablemente, la accesibilidad es vista hoy en día cómo una dificultad añadida al diseño de aplicaciones. No obstante, existen motivos y razones por los que se debería diseñar de una manera accesible:

- **Un beneficio a nivel tecnológico:** el diseño accesible fomenta el uso de diversas utilidades de los sistemas operativos y de los navegadores.
- **Un beneficio económico:** la accesibilidad ofrece el potencial necesario para que las organizaciones y empresas adquieran nuevos clientes y nuevos mercados.



Además de por todos estos beneficios, es necesario diseñar de manera accesible porque la ley nos obliga a ello, ya que es un aspecto regulado. Muchos países cuentan con legislación sobre la accesibilidad de las aplicaciones informáticas que prestan servicios públicos. Además, también se están desarrollando normativas y códigos éticos y de buenas prácticas para la industria y el comercio electrónico. En España nos incumben las siguientes normativas [19]:

- Ley 49/2007, de 26 de diciembre, por la que se establece el régimen de infracciones y sanciones en materia de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad.
- Real Decreto 1494/2007, de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a la sociedad de la información.
- Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico (LSSICE). En esta ley se puede encontrar la siguiente disposición, que puede llegar a tener especial interés de cara al problema que se quiere resolver, puesto que habla de las Administraciones Públicas, "Las Administraciones Públicas adoptarán las medidas necesarias para que la información disponible en sus respectivas páginas de Internet pueda ser accesible a personas con discapacidad y de edad avanzada de acuerdo con los criterios de accesibilidad al contenido generalmente reconocidos antes del 31 de diciembre de 2005. Asimismo, podrán exigir que las páginas de Internet cuyo diseño o mantenimiento financien apliquen los criterios de accesibilidad antes mencionados."
- **Normativa UNE 139803:2004:** Llamada "Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web", proporciona soluciones accesibles para los desarrolladores Web, creando un listado de recursos que permiten definir las características que han de cumplirse en materia de los contenidos Web en Internet, Intranets y en cualquier otro tipo de redes informáticas.
- **Normativa CWA 15554:2006:** Llamada "Especificaciones para el esquema de la evaluación de la conformidad y marca de calidad sobre accesibilidad Web".

En Europa existen otras disposiciones legales que también nos afectan:

- **eEurope 2002.** "Accesibilidad de los sitios Web públicos y de su Contenido".



- **eEurope 2005.** "Una sociedad de la información para todos".

3.4 Estándares Web

Los estándares web son un conjunto de recomendaciones dadas por el World Wide Web Consortium (W3C) y otras organizaciones internacionales acerca de cómo crear e interpretar documentos basados en la Web. Si se tiene en cuenta que el punto de partida y primer objetivo de la Web es permitir que los usuarios intercambien información, documentos y conocimiento a través de la red que ofrece la Web, se entiende mejor que sea necesario hablar un lenguaje común, aceptado y respetado por todos. Por esta razón, el primer objetivo de los estándares web es que sean reconocidos y respetados por todos, de forma que entre todos se pueda hablar el mismo lenguaje. Algunos de los beneficios obtenidos al trabajar con estándares son los siguientes [20]:

- **Mayor consistencia visual:** seguir estándares como XHTML o CSS permite separar el contenido de la apariencia, con lo que se puede transformar la apariencia visual de un sitio, realizando pocos y muy localizados cambios.
- **Mantenibilidad y uso de menos cantidad de código:** la ya mencionada separación de código, permite reutilizar estilos para diferentes contenidos, lo que proporciona una reducción de la cantidad de código de la web y aumenta la mantenibilidad de la aplicación.
- **Mayor relevancia para los motores de búsqueda:** la mayoría de los motores de búsqueda penalizan el no cumplimiento de los estándares web en su algoritmo de recuperación de información.
- **Mayor compatibilidad con distintos navegadores:** un sitio basado en estándares tendrá una mayor compatibilidad con los navegadores existentes y con las versiones futuras.
- **Mejora de la accesibilidad:** un sitio que tiene un código XHTML semánticamente correcto, proporciona un texto equivalente alternativo a las imágenes, y a los enlaces se les da un nombre significativo, esto permite a los usuarios ciegos utilizar lectores de pantalla o líneas Braille para acceder a los contenidos.

Dos de los estándares web más importantes son XHTML y CSS. El primero de ellos se centra más en la estructuración del contenido del sitio web, y el segundo en la apariencia del mismo.



XHTML

XHTML (eXtensible Hypertext Markup Language) es un lenguaje de marcado destinado a la creación de páginas web. XHTML tiene como punto de partida HTML, que fue originalmente escrito usando SGML. Con XML en escena, se creyó conveniente una reformulación de HTML como una aplicación XML (eXtensible Markup Language). El resultado de este esfuerzo originó XHTML, que muchos autores sitúan como un puente entre HTML y XML. [21]

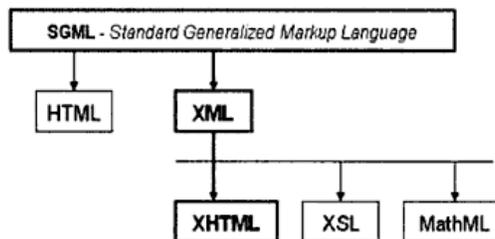


Figura 11. Marco estándares web.

XHTML posee, al igual que HTML, tres variantes que son las siguientes [22]:

- **Transitoria:** se utiliza en caso de que haya que usar características de presentación de XHTML, si no se pueden utilizar hojas de estilo.
- **Estricta:** no permite marcas de presentación, es la que se utiliza con hojas de estilo.
- **Con marcos:** se utiliza sólo cuando se emplean frames.

Como ya se mencionó, aunque XHTML toma la mayoría de los elementos de HTML, también posee algunas diferencias respecto al mismo centradas en su mayoría en conseguir un mayor rigor en el código que rompa con la laxitud y mal uso que se hace en la actualidad de HTML. Algunas reglas sencillas que se deberá respetar en la escritura de etiquetas XHTML y que garantizarán este rigor en el código, son las siguientes:

- Todas las etiquetas y atributos deben ir obligatoriamente en minúsculas (case-sensitive). Con HTML se podía escribir sin problemas <Etiqueta>, <ETIQUETA> o <etiqueta>. En XHTML sólo sería válida la última.
- Cualquier etiqueta abierta <etiqueta> debe cerrarse </etiqueta>. Pero hay excepciones, las etiquetas únicas incluyen también un signo de cierre <etiqueta />.



- Las etiquetas deben cerrarse en el orden correcto.
Mal ejemplo: `<div>...</div>`
Buen ejemplo: `<div>...</div>`
- Es obligatorio usar comillas en los atributos. De esta manera se impone mayor rigor de sintaxis.
- No se permiten formas abreviadas de los atributos. Por ejemplo usando HTML podían utilizarse atributos sin valores asociados como `checked`, `disabled`... En XHTML se debería especificar (`checked="checked"`).
- El atributo `name` queda sustituido por `id`.

Estas diferencias sobre HTML han permitido corregir numerosos problemas que conllevaba el uso de este estándar y además añadir otras ventajas:

- **Estandarización y rigor en el código:** HTML ha sido siempre un lenguaje normalizado, pero la norma ha sido ampliamente ignorada por los autores de contenido y compañías de software que crean los navegadores y herramientas de autor. Esta laxitud en el uso y la implementación, acaba con XHTML, puesto que todos están obligados a cumplir con las reglas de sintaxis.
- **Extensibilidad:** HTML tenía un conjunto de etiquetas fijas que lo hacían no extensible. En cambio, XHTML es una aplicación XML que es inherentemente extensible.
- **Compatibilidad** “hacia atrás” con los navegadores HTML.
- **Introducción de las hojas de estilo:** ahora es posible separar el contenido del documento de los aspectos relacionados con la apariencia del mismo.

Todas estas ventajas con las que cuenta XHTML, hacen que el uso de este estándar sea muy recomendable. Aún así, se pueden establecer unas guías o principios de diseño sobre cuándo es recomendable utilizar XHTML [23]:

- Cuanto más grande sea el sitio, se puede concluir que es más importante el uso CSS y XHTML. El primero hace que sea fácil de aplicar, editar y actualizar el formato de todo el sitio y el último dota a la página de una estructura que debe asegurar que tenga una duración en el futuro.



- Sitios que necesiten cubrir estándares de forma estricta para asegurar requisitos de accesibilidad o usabilidad hacen recomendable el uso de un estándar estricto como XHTML.
- Los sitios pequeños o personales puede ser recomendable aprovechar la sintaxis fácil de HTML junto con su formato de gran alcance.

Posterior a XHTML, y quizá como sucesor natural del mismo, se puede encontrar HTML5, quinta revisión del importante lenguaje HTML. El principal problema de este estándar, y lo que ha hecho que se aleje de ser objeto de estudio en este documento, es el hecho de sus numerosas incompatibilidades con los distintos navegadores. Estas incompatibilidades hacen que todavía no se tenga una especificación concreta sobre HTML5 [24] y que se deba dar un cierto tiempo para poder utilizar el estándar con garantías.

Por último, se destaca que las especificaciones de la W3C más importantes encontradas sobre el estándar XHTML son las siguientes [25]:

XHTML	Estado	Fecha
1.0	Recomendación	26/10/2000
2.0	Borrador	26/07/2006

Tabla 48. Principales versiones XHTML.

CSS

Hojas de estilo en cascada (CSS), es un mecanismo simple para agregar estilo (por ejemplo: fuentes, colores, espaciado) a los documentos web, describiendo cómo se va a mostrar un documento en la pantalla. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos. En cuanto su funcionamiento, CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más reglas aplicadas a un documento HTML o XHTML. Como se puede observar en la [Figura 12](#), la regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.



```
Selector {declaración (propiedad : valor)}
```

Figura 12. Regla CSS.

Las ventajas de usar CSS son múltiples, y la mayoría de ellas tienen que ver con la separación de la apariencia que proporciona este estándar respecto al contenido. Esta separación ayuda a tener documentos bien estructurados, con un marcado que describe con precisión su contenido. Este marcado hace que las computadoras sean más capaces de comprender la semántica del contenido y esta semántica se considera necesaria para un futuro cercano en la web, en la que poco a poco se introduce lo que se denomina como web semántica. Por otro lado, y como ventaja esencial que proporciona tener la apariencia de manera separada en hojas de estilo, hay que destacar la facilidad de rediseño y mantenibilidad que aporta un estándar como CSS, que convierte dicho rediseño en una tarea relativamente sencilla. En lugar de cortar y pegar contenido de las tablas, simplemente bastaría con escribir una nueva hoja de estilos y aplicarlo al documento.

En cuanto al estándar CSS, se destacan las siguientes versiones y sus estados [26]:

CSS	Estado	Fecha
1.0	Recomendación	17/12/1996
2.1	Candidata a recomendación	08/11/2009
3.0 ³	Borrador	23/05/2001

Tabla 49. Principales versiones CSS.

3.5 Herramientas de evaluación

En este apartado se llevará a cabo una revisión de distintas herramientas de evaluación de estándares, accesibilidad y usabilidad que ayudarán en un futuro a comprobar si el sistema desarrollado cumple con los requisitos establecidos en la fase de análisis y diseño.

La W3C proporciona distintas herramientas automáticas para evaluar si una aplicación sigue sus estándares y recomendaciones:

³ Introduction to CSS3



- **(X)HTML Validator:** es una herramienta de validación del correcto cumplimiento del estándar X (HTML). <http://validator.w3.org/>
- **CSS Validator:** es una herramienta de validación del estándar CSS. <http://jigsaw.w3.org/css-validator/>

Por otro lado, existen también herramientas online para evaluar la accesibilidad de un sistema web. Gran parte de ellas basan su análisis en las guías WCAG 1.0, y algunas han empezado a introducir como referencia WCAG 2.0 [27]:

- **TAW:** es una familia de herramientas para el análisis automático de la accesibilidad de sitios web, alcanzando de una forma integral y global a todos los elementos y páginas que lo componen. <http://www.tawdis.net/>
- **Wave:** herramienta automática de evaluación de la accesibilidad que muestra la página analizada con iconos que nos muestran algunos aspectos sobre su accesibilidad. <http://wave.webaim.org/>
- **Cynthia says:** revisa de forma automática la conformidad de las páginas Web con los distintos niveles de accesibilidad propuestos por las Directrices de Accesibilidad para el Contenido Web 1.0 del WAI o con la Sección 508 (enmienda a la Rehabilitation Act de EE.UU.), a elección del usuario. También puede revisar la propiedad del uso de los textos alternativos. <http://www.cynthiasays.com>
- **Hera:** es una utilidad para revisar la accesibilidad de las páginas web, que realiza un análisis automático previo de la página e informa si se encuentran errores (detectables en forma automática) y qué puntos de verificación de las pautas deben ser revisados manualmente. <http://www.sidar.org/hera/>
- **A-prompt:** herramienta para verificar la accesibilidad de las páginas y corregir los errores encontrados. <http://aprompt.snow.utoronto.ca/>

En la Tabla 50. Resumen herramientas evaluación. Tabla 50 se puede observar un resumen de las distintas herramientas de evaluación encontradas, los objetivos que persiguen y las direcciones web dónde se pueden encontrar.



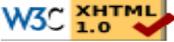
Herramienta	Objetivo	URL
 Validador XHTML	Validar el cumplimiento del estándar XHTML	http://validator.w3.org/
 Validador CSS	Validar el cumplimiento del estándar CSS.	http://jigsaw.w3.org/css-validator/
 TAW	Revisión automática del grado de accesibilidad web.	http://www.tawdis.net/
 Wave	Verificación automática de la corrección de los textos alternativos y de la estructura de la página web.	http://wave.webaim.org/
 Cynthia Says	Revisión automática de la conformidad de las páginas Web con los distintos niveles de accesibilidad.	http://www.cynthiasays.com
 Hera	Revisión manual de la accesibilidad mediante la aplicación de hojas de estilo en cascada (CSS).	http://www.sidar.org/hera/
 A-prompt	Herramienta reparadora usada para verificar la accesibilidad de las páginas y corregir los errores encontrados.	http://aprompt.snow.utoronto.ca/

Tabla 50. Resumen herramientas evaluación.



4 Solución

En este apartado se explicará de forma detallada la solución llevada a cabo para resolver la problemática planteada, incidiendo de forma especial en el proceso de desarrollo de la solución propuesta.

4.1 Descripción de la solución

Como ya se introdujo en apartados anteriores, se pretende desarrollar un prototipo de alto nivel centrado en las interfaces que dé soporte a la actividad básica de REMER, pero trasladando su forma de trabajo a un contexto web.

La razón por la que se elabora este prototipo es disponer de una herramienta que pueda ser validada por la comunidad REMER como parte de un futuro proceso de diseño participativo dirigido a disponer de una herramienta colaborativa, encuadrada en el marco del proyecto URThey, que soporte la actividad de dicha comunidad. En este sentido, el prototipado aportará las siguientes ventajas durante el futuro proceso de diseño [28]:

- **Involucra a los usuarios en el proceso de diseño:** el hecho de poder ver e interactuar con el prototipo, va a proporcionarnos más y mejor retroalimentación de los usuarios, evitando los comunes malentendidos y mejorando la comunicación entre el desarrollador y los usuarios.
- **Mayor satisfacción de los usuarios con el resultado:** al involucrarse en el proceso de desarrollo, los usuarios se sienten partícipes de la solución.
- **Optimiza el proceso de desarrollo:** debido a que la necesidad de cambios o errores se van a detectar en etapas tempranas en las que el rediseño es menos costoso, los tiempos y costes generados por el producto se verán sensiblemente reducidos.
- **Facilita el aprendizaje del usuario:** al ser el usuario partícipe de la solución hace que le sea más sencillo aprender a usarla.

Desde un punto de vista tecnológico, y tal y como se detalló en el planteamiento del problema, son objetivos clave del diseño de la solución el cumplimiento de los estándares y los principales principios de usabilidad y accesibilidad, de forma que se obtenga un prototipo de calidad.



Para desarrollar el prototipo web que se pretende con las garantías de éxito y calidad requeridas, se hace necesario seguir una metodología de desarrollo que guíe la solución al problema ya planteado. De las distintas alternativas expuestas en la revisión de las mismas llevada a cabo en apartados anteriores, se ha decidido seguir la metodología OOHDm debido a los siguientes motivos:

- Es la más utilizada y aceptada en la actualidad. Esto también contribuye a que haya más literatura acerca de esta metodología.
- Utiliza como notación UML, uno de los lenguajes de modelado más conocidos que es seguido y respaldado por numerosas organizaciones.
- Las vistas abstractas de datos (ADV) que proporciona esta metodología, permiten representar la estructura estática de la interfaz de una manera sencilla, completa e independiente de plataforma.
- Es una metodología orientada a objetos, lo que facilitará una futura integración en lenguajes orientados a objeto.

Además se va a seguir la propuesta del modelo denominada OOHDm-UID, que está más orientada a la ingeniería de requisitos. Esta propuesta añade algunas técnicas durante la fase de obtención de requisitos que van a ayudar a completar una solución al problema aún más completa y detallada. El único inconveniente encontrado tanto en esta metodología, como en muchas otras, es la excesiva documentación que genera. Dado que el producto a generar es un prototipo, se seguirá la metodología pero no se profundizará en exceso debido a la naturaleza de la solución.

Respecto a la usabilidad, se pretende diseñar un prototipo en el que sus interfaces web traten la usabilidad como un factor crítico de calidad, permitiendo que el usuario trabaje de una manera sencilla y fácil en el logro de sus objetivos, sintiéndose satisfecho de haber utilizado el sistema. Para conseguir este objetivo, la solución seguirá las reglas heurísticas de Nielsen ya expuestas en anteriores apartados, puesto que son las más reconocidas y aceptadas. A partir de estas reglas y en base a ellas, se establecerán una serie de requisitos no funcionales durante el análisis, que servirán para realizar una evaluación de la usabilidad de la solución mediante la verificación analítica de los mismos.



Otro objetivo clave de la solución es la accesibilidad, por ello el diseño tendrá como objetivo cumplir las Pautas de Accesibilidad al Contenido en la Web 1.0 (WCAG), con el propósito de hacer accesibles los contenidos del prototipo, especialmente con independencia del dispositivo y aplicación que los soporte. Para la evaluación de la accesibilidad a partir de dichas pautas, se utilizará el Test de Accesibilidad Web (TAW) que nos reportará el nivel de accesibilidad, siendo el objetivo alcanzar los puntos de prioridad 1, es decir, un nivel A de accesibilidad. Además de esta herramienta, se utilizarán otras complementarias como son Wave y Hera, que permitirán completar una evaluación global y completa de la solución.

Por último, es necesario destacar que la solución contará con el contenido y la apariencia separada mediante hojas de estilo. Para el contenido se seguirá el estándar XHTML 1.0 puesto que es la única versión estable que existe, ya que la siguiente se encuentra aún como borrador. Para los aspectos relacionados con la apariencia y estilo del mismo, se seguirá el estándar CSS 2.1, debido a que es la versión superior que tiene un estado estable. A pesar de que la siguiente versión de CSS (CSS3) incorpora una mejoras para definir el estilo de una web, se ha decidido no seguirlo en la solución porque, tal y como se expuso en la revisión de estándares realizada, se encuentra aún como borrador y existen una gran cantidad de navegadores que no soportan una o varias de sus funcionalidades.

4.2 El proceso de desarrollo

En este punto se determinará el proceso de desarrollo del sistema propuesto en la solución, siendo este proceso la base para que el proyecto se realice de forma correcta. Este proceso viene ya delimitado por la metodología de diseño escogida (OOHDM-UID), y consta de una primera parte relativa a la ingeniería de requisitos sobre los que se construirá el software. Esta parte será crucial e influirá de forma determinante en las siguientes fases del proceso de desarrollo. En ella, se realizará una especificación de los requisitos mediante distintas técnicas detalladas en OOHDM-UID. A partir de lo anterior, se seguirán las fases de la metodología OOHDM original, a saber: diseño conceptual, diseño navegacional, diseño de la interfaz abstracta e implementación. Por último, se realizarán pruebas y evaluaciones necesarias que indiquen que se cumple todo lo establecido en los requisitos y análisis.



Análisis

Este apartado tiene como meta final especificar las características y objetivos que el nuevo sistema deberá proporcionar así como su análisis y clasificación. En esta fase, y siguiendo el modelo OOHDM-UID, se especificarán: las necesidades del usuario en forma de requisitos, una identificación de roles y tareas que aclaren el intercambio de información con el sistema por parte de los usuarios, una descripción de los escenarios en los que se encuentra cada una de estas tareas, los casos de uso y una representación gráfica de los mismos en forma de diagrama de interacción de usuario (UID).

Definición de requisitos

La definición de requisitos expresa las necesidades y la problemática a la que se debe dar solución. En esta fase se dividen los requisitos en dos tipos:

- **Requisitos funcionales:** expresan funcionalidades del sistema, indicando qué debe hacer el sistema y nunca dirigiendo su atención a aspectos técnicos de cómo debe hacerlo. Se identifican también por Requisitos de Capacidad.
- **Requisitos no funcionales:** expresan la forma, el cómo, se deben alcanzar los objetivos fijados con los requisitos funcionales. Tienen también por función recoger las limitaciones del software. Se conocen también por Requisitos de Restricción.

En la [Tabla 52](#) y la [Tabla 54](#) se muestra la definición de requisitos de acuerdo a la siguiente notación:

- **Identificador:** identifica el requisito de forma unívoca, con el formato: R(F|NF)-X donde “R” representa la palabra requisito, “F” o “NF” representan el tipo que puede ser funcional o no funcional y “X” indica el número de secuencia del requisito.
- **Descripción:** breve explicación del requisito en cuestión.
- **Estabilidad:** puede ser *estable*, cuando el cliente asegura que, en principio, no se va a modificar, o *inestable* cuando el requisito puede variar en función de las sucesivas etapas del proyecto.



- **Necesidad:** puede tomar el valor *esencial* cuando el cliente lo considera un requisito imprescindible, *deseable* cuando el requisito es importante pero se puede negociar u *opcional* cuando el requisito es prescindible.
- **Prioridad:** puede ser *alta, media o baja*, según el momento en que se debe implementar durante el desarrollo.
- **Fuente:** indica el principal origen del requisito.



Requisitos Funcionales

IDENTIFICADOR	NOMBRE	DESCRIPCIÓN
<i>RF-01</i>	Roles de usuario	El sistema proporcionará los siguientes roles de usuario: coordinador general, coordinador autonómico e informador.
<i>RF-02</i>	Coordinador Autonómico	Los usuario del tipo coordinador autonómico e estarán asociados a una comunidad autónoma, en la cual se centrarán su participación.
<i>RF-03</i>	Coordinador General	El usuario con rol coordinador general podrá participar en todas las comunidades autónomas.
<i>RF-04</i>	Informador	Los usuarios con rol informador estarán asociados a una zona.
<i>RF-05</i>	Tipos de alerta	Las alertas podrán ser de tipo Lluvia o Nieve.
<i>RF-06</i>	Estado de las alerta	Las alertas podrán estar en estado abierto o cerrado, y dentro del estado abierto, podrán tener el valor actualizada o no actualizada.
<i>RF-07</i>	Zonas de alerta	Una alerta debe estar asociada a una zona.
<i>RF-08</i>	Id alertas y respuestas	El identificador de las alertas y respuestas será una cadena de texto codificada en un formato proporcionado por Protección Civil.
<i>RF-09</i>	Autenticación	Se proporcionará una interfaz para autenticarse al entrar al sistema mediante usuario y contraseña.
<i>RF-10</i>	Últimas alertas	Una vez autenticado, el sistema proporcionará información de las últimas alertas relacionadas con el usuario, que estará asociado con la pestaña "Principal" del menú de navegación.
<i>RF-11</i>	Gestión de alertas	El sistema contará con una gestión de alertas.



RF-12	Localización gráfica de alertas	Todas las alertas podrán ser localizadas geográficamente en un mapa proporcionado por el sistema.
RF-13	Información geo-temporal	El sistema mostrará información geo-temporal de las alertas.
RF-14	Alta de alerta	El sistema permitirá dar de alta una alerta a los usuarios de tipo coordinador.
RF-15	Formulario alta alerta	Para el alta de una alerta el sistema proporcionará un formulario donde se tendrán que rellenar una serie de campos.
RF-16	Editar Asociación Informadores	El sistema permitirá a los usuarios de tipo coordinador editar la asociación de informadores en las alertas que ellos mismos han creado.
RF-17	Notificación de asociación a alerta	Una vez un usuario de tipo informador sea asociado a una alerta, se le notificará que ha sido asociado a nuevas alertas, la siguiente vez que acceda al sistema.
RF-18	Detalle de la alerta	El sistema mostrará el detalle de la alerta.
RF-19	Cerrar alerta	El sistema permitirá a los usuarios de tipo coordinador cerrar una alerta.
RF-20	Editar alerta	El sistema permitirá a los usuarios de tipo coordinador editar una alerta.
RF-21	PDF alerta	El sistema permitirá a los usuarios de tipo coordinador generar un archivo de tipo PDF con la información acerca de la alerta.
RF-22	Responder a alerta	El sistema dará soporte para que los usuarios de tipo informador puedan responder a una alerta.
RF-23	Gestión de usuarios	El sistema proporcionará una gestión de usuarios.
RF-24	Edición perfil propio	El sistema permitirá a los usuarios editar su perfil.



RF-25	Modificación Usuario Coordinador General	El sistema permitirá a los usuarios de tipo coordinador general modificar los datos de todos los usuarios.
RF-26	Modificación Usuario Coordinador Autónomo	El sistema permitirá a los usuarios de tipo coordinador autónomo modificar los datos de los usuarios asociados a su comunidad autónoma.
RF-27	Autorización Visualizar Perfil Informador	El sistema dispondrá de información acerca de si los usuarios de tipo informador autorizan a que otros usuarios con este mismo rol puedan consultar su perfil.
RF-28	Visualizar usuario	El sistema proporcionará una visualización del perfil de los usuarios que lo hayan autorizado.
RF-29	Estadísticas	El sistema permitirá consultar las estadísticas relativas al histórico de alertas (Comparativa entre meses, Comparativa entre tipos de alerta y Zonas con gran densidad de alertas)
RF-30	Envío Mensajes	El sistema permitirá el envío de mensajes entre los usuarios.
RF-31	Consultar mensajes	El sistema dispondrá de una interfaz donde el usuario podrá consultar los mensajes que se le han enviado.
RF-32	Ayuda	El sistema contará elementos de ayuda al usuario.
RF-33	Pérdida de contraseña	El sistema permitirá que el usuario pueda solicitar el envío de su contraseña a su dirección E-mail.
RF-34	Búsqueda en la web	El sistema proporcionará un motor de búsqueda para el sitio web.

Tabla 51. Requisitos Funcionales I



IDENTIFICADOR	PRIORIDAD	FUENTE	NECESIDAD	ESTABILIDAD
<i>RF-01</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-02</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-03</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-04</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-05</i>	ALTA	OTRAS APLICACIONES	ESENCIAL	ESTABLE
<i>RF-06</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-07</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-08</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-09</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-10</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-11</i>	ALTA	OTRAS APLICACIONES	ESENCIAL	ESTABLE
<i>RF-12</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-13</i>	MEDIA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-14</i>	MEDIA	OTRAS APLICACIONES	ESENCIAL	ESTABLE
<i>RF-15</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-16</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-17</i>	ALTA	OTRAS APLICACIONES	DESEABLE	ESTABLE
<i>RF-18</i>	MEDIA	CLIENTE	ESENCIAL	ESTABLE



<i>RF-19</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-20</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-21</i>	ALTA	CLIENTE	OPCIONAL	ESTABLE
<i>RF-22</i>	ALTA	OTRAS APLICACIONES	ESENCIAL	ESTABLE
<i>RF-23</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-24</i>	MEDIA	CLIENTE	DESEABLE	ESTABLE
<i>RF-25</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-26</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-27</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-28</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-29</i>	ALTA	OTRAS APLICACIONES	OPCIONAL	ESTABLE
<i>RF-30</i>	MEDIA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-31</i>	ALTA	CLIENTE	DESEABLE	ESTABLE
<i>RF-32</i>	ALTA	CLIENTE	DESEABLE	ESTABLE
<i>RF-33</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RF-34</i>	BAJA	OTRAS APLICACIONES	DESEABLE	ESTABLE

Tabla 52. Requisitos Funcionales II



Requisitos No Funcionales

IDENTIFICADOR	NOMBRE	DESCRIPCIÓN
<i>RNF-01</i>	Coordinador General	El sistema sólo podrá contar con un usuario con rol coordinador general.
<i>RNF-02</i>	Coordinador Autonómico	Por cada comunidad autónoma, el sistema solo podrá contar con un usuario con rol coordinador autonómico.
<i>RNF-03</i>	Estado cerrado	Las alertas con estado actualizado sólo formarán parte del sistema para conformar las estadísticas, en ningún caso podrán ser actualizadas o editadas alertas en este estado.
<i>RNF-04</i>	Envío de mensajes	El sistema permitirá el envío de mensajes a través de distintos medios: email, sms, mensaje a través de la aplicación.
<i>RNF-05</i>	Separación completa contenido-estilo	El código del sistema deberá tener una separación completa del estilo y contenido.
<i>RNF-06</i>	XHTML	La aplicación deberá superar validaciones acerca del cumplimiento del estándar XHTML 1.0.
<i>RNF-07</i>	CSS	La aplicación deberá superar validaciones acerca del cumplimiento del estándar CSS 2.1.
<i>RNF-08</i>	Feedback adecuado	El sistema proporcionará al usuario un feedback adecuado de dónde se encuentra dentro de la estructura de páginas del sitio.
<i>RNF-09</i>	Lenguaje familiar	Las interfaces del sistema deberán usar un lenguaje familiar para los usuarios.
<i>RNF-10</i>	Prevención de errores	El sistema deberá minimizar la posibilidad de que el usuario entre en un estado inconsistente o de error.
<i>RNF-11</i>	Errores descriptivos	Los mensajes de error del sistema deberán ser descriptivos ayudando al usuario a salir de dicho error.



<i>RNF-12</i>	Coherencia en Layouts	La aplicación mantendrá una coherencia en el uso de layouts en todas sus interfaces.
<i>RNF-13</i>	Ayuda siempre visible	El link para consultar la ayuda deberá estar siempre visible para el usuario.
<i>RNF-14</i>	Accesibilidad	Las interfaces deberán seguir las Pautas de accesibilidad al Contenido Web 1.0 WCAG.
<i>RNF-15</i>	Formato fecha	Las fechas que aparezcan en el sistema deberán seguir el siguiente formato dd-mm-aaaa.
<i>RNF-16</i>	Compatibilidad con distintos navegadores	El sistema deberá ser compatible, al menos, con los navegadores: Internet Explorer, Google Chrome, Opera, Mozilla Firefox. Todos ellos en su última versión

Tabla 53. Requisitos No funcionales I



IDENTIFICADOR	PRIORIDAD	FUENTE	NECESIDAD	ESTABILIDAD
<i>RNF-01</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-02</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-03</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-04</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-05</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-06</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-07</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-08</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-09</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-10</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-11</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-12</i>	ALTA	CLIENTE	DESEABLE	ESTABLE
<i>RNF-13</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-14</i>	ALTA	CLIENTE	DESEABLE	ESTABLE
<i>RNF-15</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE
<i>RNF-16</i>	ALTA	CLIENTE	ESENCIAL	ESTABLE

Tabla 54. Requisitos No funcionales II.



Análisis de roles y tareas

Esta fase tiene como objetivo identificar los diferentes roles que podrían cumplir cada uno de los potenciales usuarios de la aplicación y detallar las tareas que va a soportar el sistema para cada uno de estos roles.

El rol **coordinador general** tiene como tareas:

- Crear una alerta en cualquier zona.
- Consultar información de las alertas de cualquier zona.
- Gestionar una alerta en cualquier zona.
- Gestionar usuarios de cualquier comunidad autónoma.
- Consultar estadísticas.
- Comunicarse con otros usuarios.
- Consultar mensajes recibidos
- Consultar la ayuda del sistema.

El rol **coordinador autonómico** tiene como tareas:

- Crear una alerta en zonas pertenecientes a la comunidad autónoma a la que están asociados.
- Consultar información de las alertas en zonas pertenecientes a la comunidad autónoma a la que están asociados.
- Gestionar una alerta en zonas pertenecientes a la comunidad autónoma a la que están asociados.
- Gestionar usuarios en la comunidad autónoma a la que están asociados.
- Comunicarse con otros usuarios.
- Consultar mensajes recibidos.
- Consultar la ayuda del sistema.

El rol **informador** tiene como tareas:

- Consultar información de una alerta a la que esté asociado.
- Responder a una alerta a la que esté asociado.
- Gestionar su usuario.
- Consultar información del perfil de otros usuarios.



- Comunicarse con otros usuarios.
- Consultar mensajes recibidos.
- Consultar la ayuda del sistema.

Descripción de escenarios

Los escenarios son descripciones narrativas de cómo la aplicación puede ser usada para que los actores lleven a cabo cada tarea. En este punto, se deberán especificar todos los escenarios que describen las tareas expuestas anteriormente.

- 1.1 **Autenticación:** el usuario debe introducir login y password para acceder al sistema y sus funcionalidades.
- 1.2 **Creación de una alerta:** las alertas son creadas rellenando un formulario que el sistema proporciona, y que deberá incluir toda la información necesaria relativa a la alerta. Estarán asociadas a una zona.
- 1.3 **Visualizando alertas:** las alertas serán visualizadas en un mapa de localización en el que, también de forma gráfica, se denotará el tipo de alerta del que se trata (según un icono).
- 1.4 **Visualizando alertas según su fecha de creación:** en la visualización de alertas, el usuario dispondrá de un timemap dónde indicará el periodo en el que quiere visualizar alertas. De esta forma, en el mapa de localización de alertas sólo aparecerán aquellas cuya fecha de inicio prevista este dentro del periodo seleccionado.
- 1.5 **Consultando información de una alerta:** el usuario podrá consultar en detalle la información de las alertas visualizadas, identificadas por una cadena de texto codificada a partir de los datos de la misma. Esta información contará con campos como localización geográfica individual, respuestas, detalles técnicos...
- 1.6 **Edición de una alerta:** dentro de la gestión de alertas, las alertas creadas deberán poder ser editadas para cambiar alguno de sus datos.
- 1.7 **Cierre de una alerta:** dentro de la gestión de alertas, las alertas creadas podrán cerrarse cuando se considere que dicha alerta ha dejado de tener validez. Estas alertas seguirán apareciendo en el sistema, pudiendo consultar información sobre la misma. La diferencia será que no se seguirá completando la información sobre estas ni se podrá realizar ninguna operación.



- 1.8 Asociación de informadores a una alerta:** tanto durante la creación de la alerta, como una vez creada, el sistema permitirá asociar usuarios con rol informador a las alertas dentro de la gestión de alertas. Los usuarios asociados deberán pertenecer a alguna de las zonas de la comunidad autónoma dónde se sitúa la alerta.
- 1.9 Generación de un PDF sobre la alerta:** dentro de la tarea de gestión de alertas, el usuario podrá generar un PDF con los detalles de la alerta que desee y que posteriormente podrá ser impreso o guardado.
- 1.10 Respondiendo a una alerta:** una vez la alerta ha sido creada y se han asociado a ella los usuarios de tipo informador oportunos, se podrá responder a la alerta indicando datos acerca de cómo evoluciona la misma.
- 1.11 Consultando estadísticas:** el usuario podrá consultar una serie de estadísticas acerca de las alertas, como las zonas con más densidad de alertas, el tipo de alertas más ocurridas y una comparativa entre meses y alertas ocurridas.
- 1.12 Buscar usuario por comunidad autónoma:** en la gestión de usuarios, se podrá dada una comunidad autónoma, conocer los usuarios que pertenecen a esa comunidad autónoma.
- 1.13 Consultando información del perfil un usuario:** en la gestión de usuarios, se podrá consultar información personal sobre los usuarios que lo permitan, como su localización u otros datos personales.
- 1.14 Edición de usuarios:** además de consultar información, en la gestión de usuarios del sistema, se podrán seleccionar usuarios y, dentro de los mismos, modificar sus datos.
- 1.15 Comunicación con otros usuarios:** el sistema permitirá que un usuario envíe mensajes a otro/s usuario/s a través de diversos medios de comunicación como sms, email y mensaje en la aplicación.
- 1.16 Consultando mensajes recibidos y enviados:** los usuarios podrán consultar los mensajes tanto recibidos como enviados.
- 1.17 Consultar ayuda:** el sistema cuenta con una ayuda que explica las diversas funcionalidades del sistema y cómo se utilizan.

Modelo de casos de uso

Los escenarios deben ser agrupados en unidades funcionales definidas como casos de uso. Con estos, se pretende mostrar la forma en que los distintos usuarios de la aplicación interactúan con



la misma, proporcionando una mejor comprensión del sistema a implementar. En este punto, en primer lugar se mostrarán de forma gráfica los casos de uso para los distintos roles de usuario, y posteriormente se detallará la descripción textual de los mismos.

Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el usuario con el sistema, representando la secuencia de interacciones que se desarrollan entre el sistema y sus actores. Un actor es toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. En el sistema a diseñar y tal y como se expuso en la descripción del sistema se cuenta con los siguientes roles de usuario:

- Coordinador
 - Coordinador General
 - Coordinador Autónomico
- Informador

Estos roles se representan jerárquicamente con la estructura de generalización UML de actores que se observa en la [Figura 13](#).

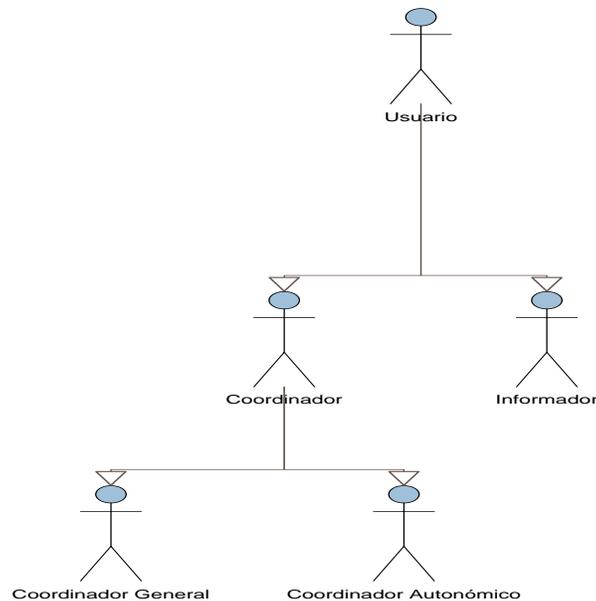


Figura 13. Generalización de actores.

Para una mejor comprensión del diagrama de casos de uso, se van a mostrar los casos de uso agrupados por el rol de los actores que pueden llevar a cabo dichos casos de uso. En primer lugar, se visualizarán en la [Figura 14](#) los casos de uso comunes a todos los roles.

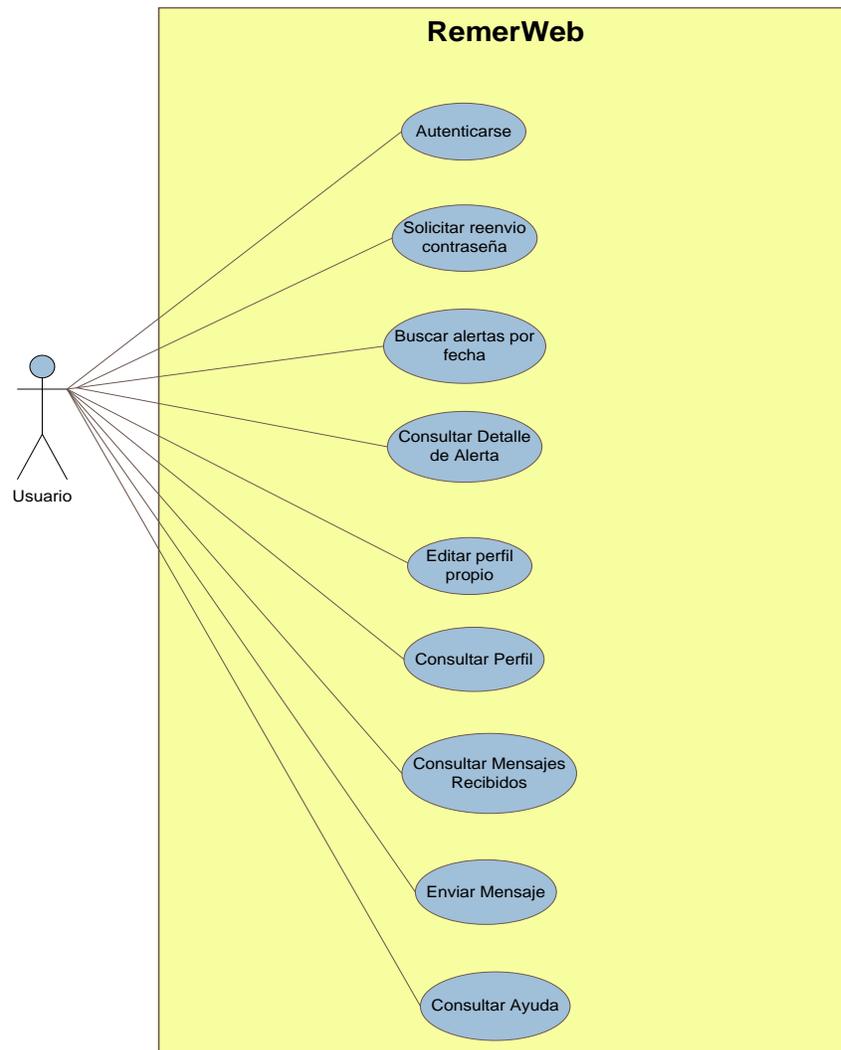


Figura 14. Diagrama Casos de Uso General.

A continuación, se pueden observar en la [Figura 15](#) los casos de uso que pueden realizar los usuarios con rol coordinador. Tanto el usuario con rol coordinador general como el usuario con rol coordinador provincial, realizan las mismas operaciones con la única diferencia del ámbito de aplicación. Mientras que el primero se ocupa de la gestión de alertas y usuarios de forma general, el segundo se encarga de esta misma gestión pero solamente en las zonas que pertenezcan a la provincia a la que está asociado.

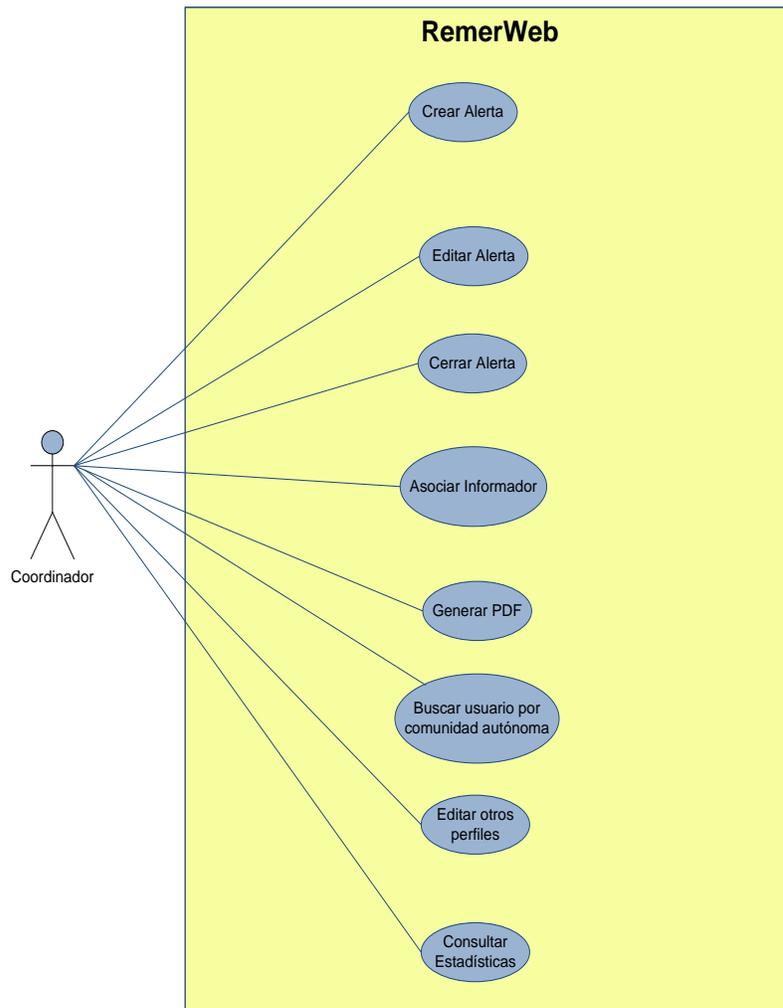


Figura 15. Diagrama Casos de Uso Coordinador.

Por último, más adelante se puede observar el diagrama de casos de uso para los usuarios con rol informador. Estos usuarios tienen como única función particular la respuesta a las alertas a las que han sido asociadas.

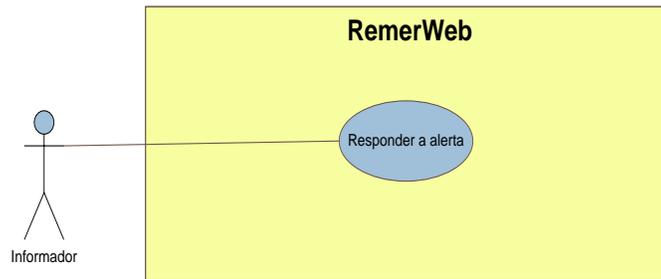


Figura 16. Diagrama Casos de Uso Informador.

Como complemento y apoyo a los diagramas de casos de uso anteriores, se va a realizar una descripción textual de los casos de uso que mejore la comprensión formal de los mismos.

Caso de Uso	Autenticarse
Escenarios	1.1
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario introduce login y password.2. El usuario hace click en el botón de Entrar.3. Si el login y el password es correcto el sistema permitirá al usuario acceder al mismo.

Tabla 55.Caso de Uso Autenticarse.

Caso de Uso	Reenviar contraseña
Escenarios	1.1
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario selecciona la opción "Olvidó su contraseña".2. El usuario introducirá su cuenta de correo.3. El sistema enviará por email a este correo, el login y usuario asociado a la cuenta de correo introducida.

Tabla 56. Caso de Uso Reenviar contraseña.

Caso de Uso	Crear una alerta
Escenarios	1.2
Actores	Coordinador General, Coordinador Autonómico
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción "Gestión de alertas".2. El sistema le llevará a la interfaz de gestión de alertas.3. En está interfaz el usuario hará click en el botón de crear una alerta.4. El sistema abrirá un formulario para que el usuario introduzca los datos relativos a la alerta.



	<ol style="list-style-type: none">5. El sistema mostrará una lista de usuarios propuestos para ser asociados a la alerta6. El usuario asociará aquellos usuarios que consideré a la alerta7. Una vez todos los datos relativos a la alerta estén introducidos el usuario confirmará la creación de la alerta.8. El sistema informará de la correcta creación de la alerta
--	--

Tabla 57. Caso de Uso Crear alerta.

Caso de Uso	Buscar alertas por fecha
Escenarios	1.3, 1.4
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción "Gestión de alertas".2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. En usuario seleccionará en el timemap el periodo de tiempo en el que quiere buscar la alerta.4. El sistema mostrará solamente las alertas cuya fecha prevista de inicio este dentro del periodo seleccionado.

Tabla 58. Caso de Uso Buscar alerta fecha.

Caso de Uso	Consultar en detalle la información de una alerta
Escenarios	1.3, 1.4, 1.5
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción "Gestión de alertas".2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. El usuario buscará la alerta concreta que se quiere consultar y la seleccionará.4. El sistema le mostrará la información en detalle de la alerta seleccionada.

Tabla 59. Caso de Uso Consultar alerta.



Caso de Uso	Editar una alerta
Escenarios	1.3, 1.4, 1.5, 1.6
Actores	Coordinador General, Coordinador Autonómico
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción "Gestión de alertas".2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. El usuario buscará la alerta concreta que se quiere consultar y la seleccionará.4. El sistema le mostrará la información en detalle de la alerta seleccionada.5. El usuario hará click en el botón de editar la alerta.6. El sistema le mostrará una interfaz donde cambiar los datos de la alerta.7. Una vez cambiados, el usuario guardará la nueva información.8. El sistema informará de la correcta edición de la alerta.

Tabla 60. Caso de Uso Editar alerta.

Caso de Uso	Cerrar una alerta
Escenarios	1.3, 1.4, 1.5, 1.7
Actores	Coordinador General, Coordinador Autonómico
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción "Gestión de alertas".2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. El usuario buscará la alerta concreta que se quiere consultar y la seleccionará.4. El sistema le mostrará la información en detalle de la alerta seleccionada.5. El usuario hará click en el botón de cerrar la alerta.6. El sistema le informará que la alerta ha sido cerrada adecuadamente.

Tabla 61. Caso de Uso Cerrar alerta.



Caso de Uso	
	Asociar informadores a una alerta
Escenarios	1.2, 1.3, 1.4, 1.5, 1.8
Actores	Coordinador General, Coordinador Autonómico
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de alertas”.2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. El usuario buscará la alerta concreta que se quiere consultar y la seleccionará.4. El sistema le mostrará la información en detalle de la alerta seleccionada.5. El usuario hará click en el botón de asociar informadores.6. El sistema le mostrará una lista de usuarios con rol informador, pertenecientes a alguna de las zonas de la comunidad autónoma a la que está asociada la alerta.7. El usuario seleccionará a los que considere oportuno y confirmará los nuevos usuarios asociados a la alerta.8. El sistema informará de la correcta asociación de los usuarios a la alerta.

Tabla 62. Caso de Uso Asociar informadores.

Caso de Uso	
	Generar PDF de una alerta
Escenarios	1.3, 1.4, 1.5, 1.9
Actores	Coordinador General, Coordinador Autonómico
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de alertas”.2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. El usuario buscará la alerta concreta que se quiere consultar y la seleccionará.4. El sistema le mostrará la información en detalle de la alerta seleccionada.5. El usuario hará click en el botón de generar PDF.6. El sistema abrirá un archivo PDF con la información de la alerta.7. El usuario podrá guardar o cerrar el archivo.

Tabla 63. Caso de uso Generar PDF alerta.



Caso de Uso		Responder a una alerta
Escenarios	1.3, 1.4, 1.5, 1.10	
Actores	Informador	
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de alertas”.2. El sistema le llevará a la interfaz de gestión de alertas, donde se mostrarán todas las alertas.3. El usuario buscará la alerta concreta que se quiere responder y la seleccionará.4. El sistema le mostrará la información en detalle de la alerta seleccionada.5. El usuario hará click en el botón de responder.6. El sistema mostrará una interfaz en la que el usuario podrá introducir los datos relativos a la respuesta.7. Una vez introducida la respuesta el usuario podrá guardarla.8. El sistema informará del correcto almacenamiento de la respuesta.	

Tabla 64. Caso de Uso Responder alerta.

Caso de Uso		Consultar estadísticas
Escenarios	1.11	
Actores	Coordinador General, Coordinador Autonómico	
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Estadísticas”.2. El sistema le llevará a la interfaz donde se mostrarán las distintas estadísticas que genera el sistema: zonas con más densidad de alertas ocurridas, tipo de alerta más ocurrida y comparativa entre meses y alertas ocurridas.3. El usuario seleccionará el tipo de estadística que quiere consultar.4. El sistema le mostrará dicha estadística.	

Tabla 65. Caso de Uso Consultar estadísticas.



Caso de Uso	
Buscar usuario por comunidad autónoma	
Escenarios	1.12
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de usuarios”.2. El usuario seleccionará la comunidad autónoma sobre la que quiere consultar los perfiles de usuario.3. El sistema le mostrará una lista con los usuarios asociados a dicha provincia.

Tabla 66. Caso de Uso Buscar usuario por zona.

Caso de Uso	
Consultar perfil de un usuario	
Escenarios	1.12, 1.13
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">4. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de usuarios”.5. El sistema le llevará a la interfaz donde se mostrarán una lista de usuarios que han autorizado a que su perfil sea visto.6. El usuario buscará y después seleccionará el usuario concreto sobre el que quiere consultar la información.7. El sistema le mostrará el perfil de dicho usuario.

Tabla 67. Caso de Uso Consultar perfil.

Caso de Uso	
Editar perfil propio	
Escenarios	1.14
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de usuarios”.2. El sistema le llevará a la interfaz donde se mostrarán una lista de usuarios que han autorizado a que su perfil sea visto.3. El usuario seleccionará su propio perfil de usuario.4. El sistema mostrará una interfaz para que el usuario pueda editar la información de su propio perfil.5. El usuario confirmará estos cambios.6. El sistema informará de la correcta edición de su perfil.

Tabla 68. Caso de Uso Editar perfil propio.



Caso de Uso		Editar un usuario
Escenarios	1.12, 1.14	
Actores	Coordinador General, Coordinador Autonómico	
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Gestión de usuarios”.2. El sistema le llevará a la interfaz donde se mostrarán una lista de usuarios con rol informador cuyos datos pueden ser editados.3. El usuario buscará y después seleccionará el usuario concreto sobre el que quiere editar la información.4. El sistema mostrará una interfaz para que el usuario pueda editar la información del usuario seleccionado.5. El usuario confirmará estos cambios.6. El sistema informará de la correcta edición del perfil del usuario seleccionado.	

Tabla 69. Caso de Uso Editar usuario.

Caso de Uso		Enviar mensaje
Escenarios	1.15	
Actores	Coordinador General, Coordinador Autonómico, Informador	
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Comunicación”.2. El usuario hará click en la el botón de enviar mensaje.3. El sistema le llevará a la interfaz donde se mostrarán una lista de usuarios ordenados por provincia, una lista de medios de comunicación y un cuadro de texto donde introducir el texto a enviar.4. El usuario seleccionará los usuarios con los que se quiere comunicar, los medios de comunicación que elige para establecer la comunicación y por último introducirá el texto a enviar.5. El usuario confirmará el envío.6. El sistema informará del correcto envío del texto.	

Tabla 70. Caso de Uso Enviar mensaje.



Caso de Uso Consultar mensajes recibidos	
Escenarios	1.16
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Comunicación”.2. El sistema le llevará a la interfaz donde se mostrarán dos listas una con los mensajes recibidos por el usuario y otra con los enviados.3. El usuario seleccionará la lista según el tipo de mensajes que se quiera consultar: recibidos o enviados.4. Dentro de la lista, el usuario seleccionará el mensaje que desea consultar.5. El sistema mostrará la información relativa al mensaje.

Tabla 71. Caso de Uso Consultar Mensajes recibidos.

Caso de Uso Consultar ayuda	
Escenarios	1.17
Actores	Coordinador General, Coordinador Autonómico, Informador
Descripción	<ol style="list-style-type: none">1. El usuario autenticado hace click en el menú de navegación en la opción “Ayuda”.2. El sistema le llevará a la interfaz donde se mostrará la ayuda del sistema.

Tabla 72. Caso de Uso Consultar ayuda.

Especificación de UID's

De acuerdo a UML, existen diversos diagramas, como los diagramas de secuencia, de colaboración y de estado, que son capaces de representar y ampliar la información de un caso de uso. Sin embargo, la especificación de casos de usos usando estas técnicas es un amplio trabajo y puede anticiparse inesperadamente a tomar algunas decisiones de diseño. Por ello, en la metodología utilizada se propone la utilización de la herramienta llamada UID, que permite representar la información de los casos de uso de una forma rápida y sencilla.

Un diagrama de interacción de usuario (UID), representa la interacción entre el usuario y el sistema describiendo el intercambio de información entre los mismo sin tener en cuenta aspectos



relacionados con la interfaz de usuario. [29] Estos diagramas se utilizarán para mejorar el entendimiento de algunos de los casos de uso más representativos del sistema.

Caso de Uso Crear Alerta

El proceso que describe la creación de una alerta consta de dos grandes fases, la primera en la que se detallan los datos de la alerta y la segunda en la que se asocian usuarios con rol informador a la misma. En la [Figura 17](#) se detalla el diagrama de interacción de esta tarea. En él, se observa como en la primera y segunda interacción, el usuario debe introducir algunos datos relativos a la alerta. En la primera, datos más generales como la fecha, tipo, localización o, de forma opcional, la duración estimada de la alerta. En cambio, en la segunda interacción se introducen datos más técnicos sobre la alerta. En la tercera interacción, el sistema mostrará una lista de usuarios propuestos por el sistema en función de la proximidad a la alerta de los cuales se podrán asociar a la alerta uno o varios. Recordar que estos usuarios tendrán rol informador y estarán asociados a alguna de las zonas de la Comunidad Autónoma en la que se sitúa la alerta. Para conocer más información del usuario a seleccionar con el objetivo que evaluar su idoneidad para ser asociado a la alerta, se puede consultar de forma opcional el detalle de la información de un usuario durante el proceso de creación. En la última iteración, el sistema mostrará la información final de la alerta con el objetivo de que el usuario la verifique y confirme la creación de la alerta.

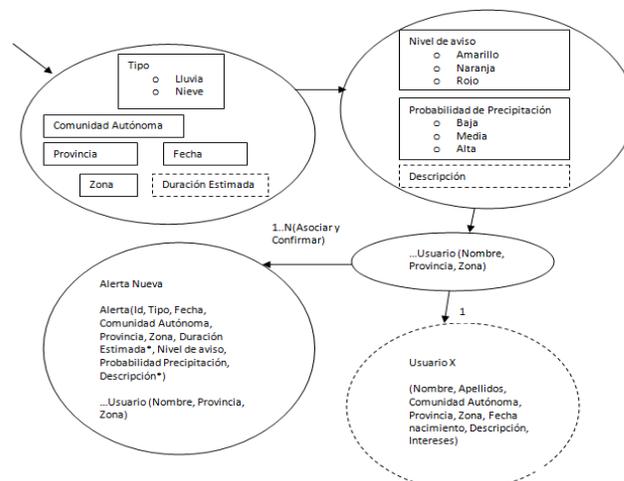


Figura 17. UID Crear Alerta.



Caso de Uso Buscar alerta por fecha

En este caso de uso, el usuario puede realizar la búsqueda de alertas en un periodo comprendido por una fecha inicial y otra final. Tal y como se observa en la [Figura 18](#), en la primera interacción el usuario introducirá estas dos fechas y en la siguiente el sistema le devolverá una lista de alertas comprendidas entre las mismas. Por último, el usuario podrá consultar el detalle de dichas alertas, seleccionando cada una de ellas.

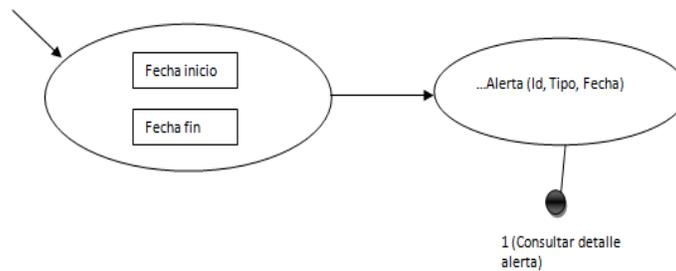


Figura 18. UID Buscar alerta por fecha.

Caso de Uso Enviar mensaje

En la [Figura 19](#) se puede observar el diagrama de interacción para el caso de uso en el que el usuario se comunica con otro mediante el envío de un mensaje. En la primera interacción del mismo se observa como el sistema muestra una lista de usuario, ordenados por Comunidad Autónoma, de los que se tienen que seleccionar uno o varios, que serán los destinatarios del mensaje. Para ayudar a escoger estos destinatarios, el sistema permitirá acceder de forma opcional al detalle del perfil de los usuarios. En la siguiente interacción, y una vez han sido seleccionados los usuarios destinatarios, el usuario remitente del mensaje deberá elegir el medio de envío del presente mensaje así como el texto a enviar. También de forma opcional podrá introducir un adjunto al mensaje.

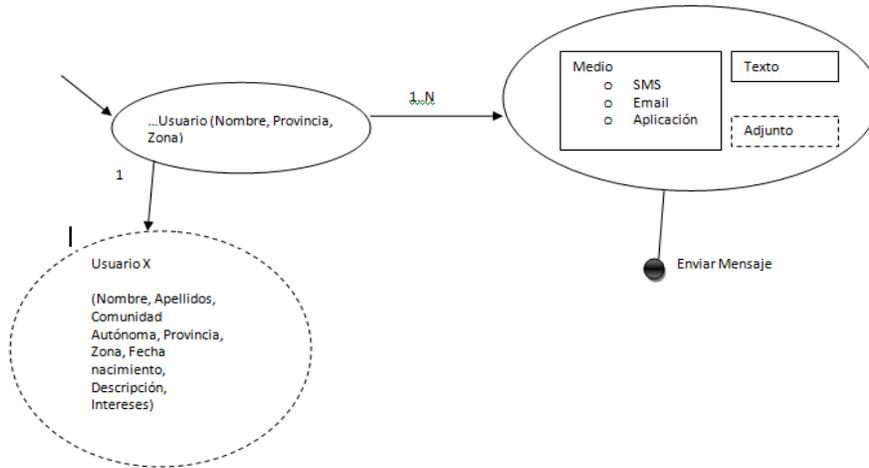


Figura 19. UID Enviar mensaje.

Caso de Uso Editar Usuario

Para llevar a cabo el proceso de edición de un usuario, primero se hace necesario buscarlo. Tal y como se muestra en los casos de uso, dicha búsqueda se puede llevar a cabo buscando al usuario por la comunidad autónoma a la que este asociado o simplemente buscando entre una lista con todos ellos. En la primera interacción de la [Figura 20](#), se observa cómo, si el usuario lo cree oportuno, de la lista de comunidades autónomas que muestra el sistema, puede seleccionar una y se le mostrará una lista con todos los usuarios asociados a esa comunidad. A partir de ahí, y tal y como se observa en la tercera interacción se visualizarán los datos del perfil del usuario seleccionado (consulta) y se podrán cambiar (edición) guardando los cambios realizados. Destacar que la zona y provincia son opcionales puesto que sólo están asociados a una zona los usuarios con rol informador.

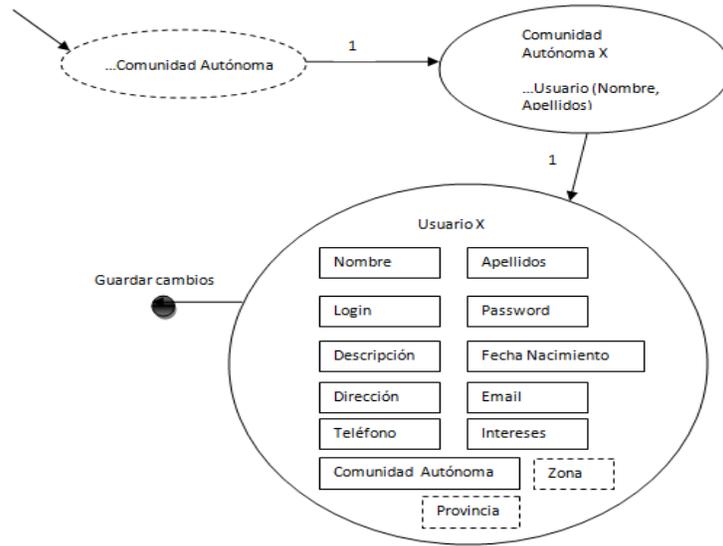


Figura 20. UID Editar usuario.



Diseño

Una vez se ha simplificado el problema para su mejor comprensión en el análisis, se va a pasar a diseñar la solución del mismo. El diseño es el primer paso en la fase de desarrollo de cualquier producto de ingeniería, cuyo objetivo es anticipar y guiar el proceso de producción, produciendo modelos o representaciones de aquellas entidades que posteriormente se van a construir.

En base al método de desarrollo seguido en la solución, OOHDM, la fase de diseño quedará dividida de la siguiente forma:

- **Diseño Conceptual:** tendrá como resultado el modelo conceptual, que mostrará la estructura de información que va a ser presentada a los usuarios.
- **Diseño Navegacional:** detallará la topología de navegación que va a tener el sistema para soportar cada tarea. Está formado por el diagrama de clases navegacionales y el diagrama de contexto navegacional.
- **Diseño de interfaces abstractas:** especificará la manera en la que el usuario va a percibir los objetos navegacionales mediante los distintos diagramas de interfaces abstractas.

Diseño Conceptual

Es el punto de comienzo del proceso de diseño y tiene como meta obtener un diagrama, en notación UML, que especifique el conjunto de clases del dominio, así como sus atributos y relaciones. De esta manera se mostrará la estructura de información que va a ser presentada a los usuarios, con especial atención a la semántica del dominio de la solución.

En la [Figura 21](#) se puede observar una visión general y global del modelo. Más adelante, y para su mejor comprensión, se irán mostrando imágenes de los distintos módulos del sistema: gestión de alertas, gestión de usuarios y comunicación.

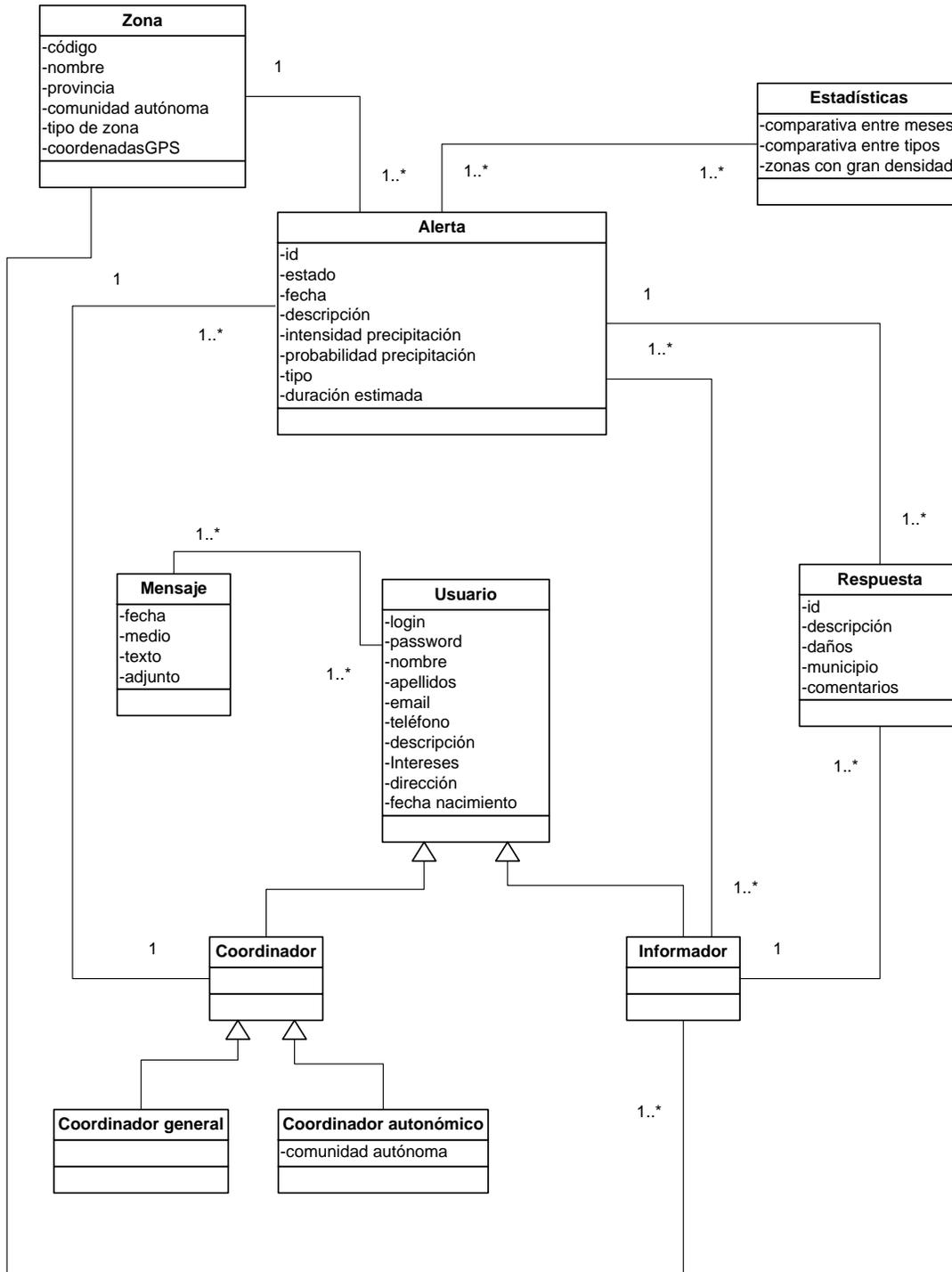


Figura 21 . Diagrama conceptual general.



En la [Figura 22](#), se focaliza la atención en la estructura de usuarios del sistema. Tal y como se observa en la figura, los usuarios pueden ser de dos tipos:

- **Coordinadores:** aquellos que pueden gestionar una alerta, desde su creación hasta su cierre. Dentro de este rol, existen dos tipos de coordinadores:
 - **Coordinador General:** pueden participar en todo el territorio.
 - **Coordinador Autónomico:** limitan su participación en la comunidad autónoma a la que están asociado.
- **Informadores:** son los que, realizando repuestas sobre las alertas a las que son asociados, van conformando una información completa sobre la misma. Están asociados a una zona concreta.

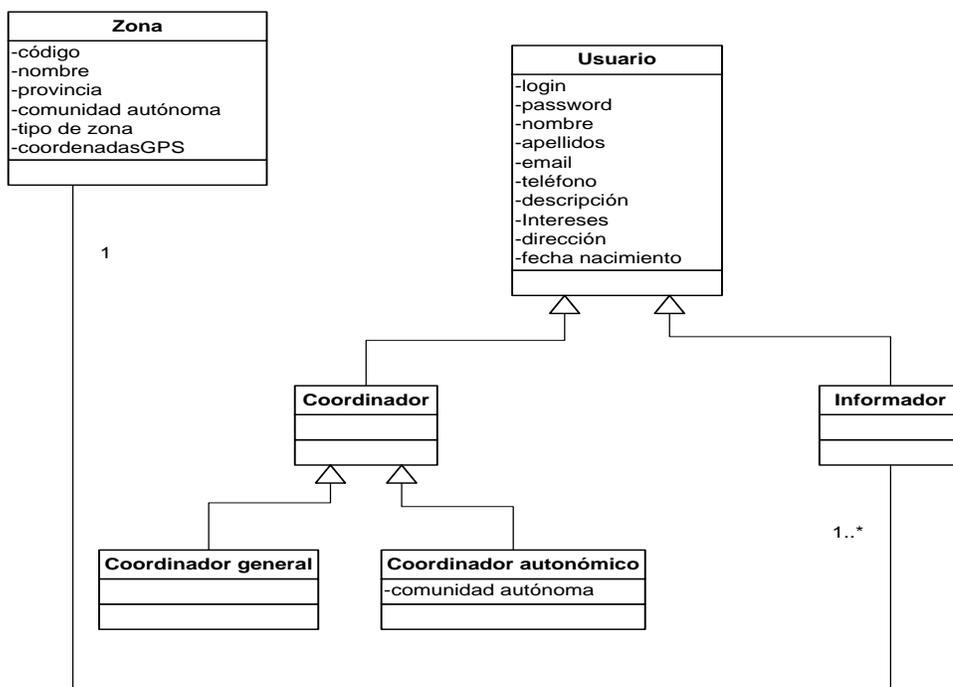


Figura 22. Diagrama conceptual usuarios.

En la [Figura 23](#) se centra la atención del modelo en la gestión de alertas. En esta figura, se puede observar como un usuario de tipo coordinador puede haber dado de alta una o varias alertas, y un usuario informador puede tener asociadas también una o varias alertas. En cambio, mientras que una alerta sólo puede tener un usuario coordinador como creador, sí puede tener asociados a



varios usuarios de tipo informador. Estos usuarios pueden realizar una o varias respuestas a las alertas, pero cada respuesta estará asociada a una única alerta y tendrá como autor a un único usuario informador. Por otro lado, se observa como una alerta está asociada a una zona, y en una zona puede haber también una o varias alertas. Por último destacar que las alertas también están asociadas a estadísticas.

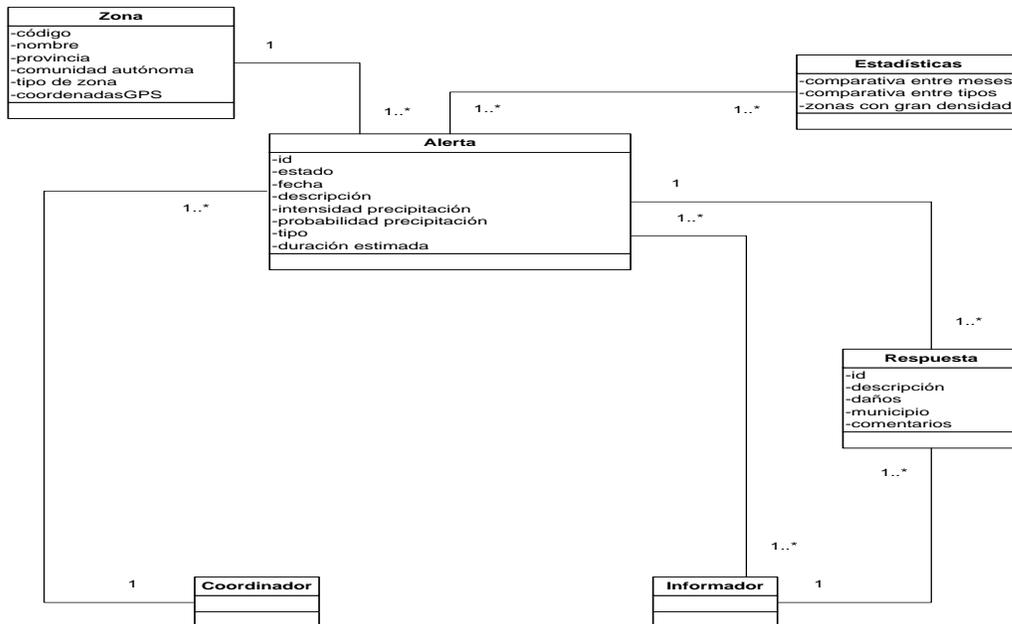


Figura 23. Diagrama conceptual alertas.

En esta última figura de modelo conceptual se observa como un usuario puede haber sido autor de uno o varios mensajes y el mensaje puede estar asociado a varios usuarios destinatarios.

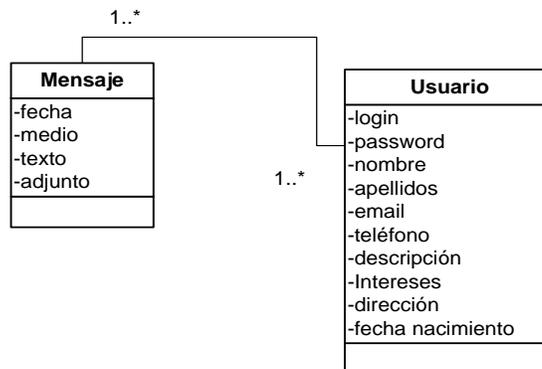


Figura 24. Diagrama conceptual comunicación.



Diseño Navegacional

En OOADM, la navegación se considera un paso crítico en el diseño de aplicaciones. Un modelo navegacional se construye como una vista del modelo conceptual, proveyendo una vista subjetiva del mismo.

El diseño navegacional da como resultado dos diagramas: el esquema de clases navegacionales y el esquema de contextos navegacionales.

El esquema de clases navegacionales parte de la idea de que el usuario no navega por objetos conceptuales, sino por otros (objetos navegacionales) contruidos a partir de uno o varios objetos conceptuales. La siguiente figura muestra este esquema en el que se distinguen los objetos navegacionales del sistema.

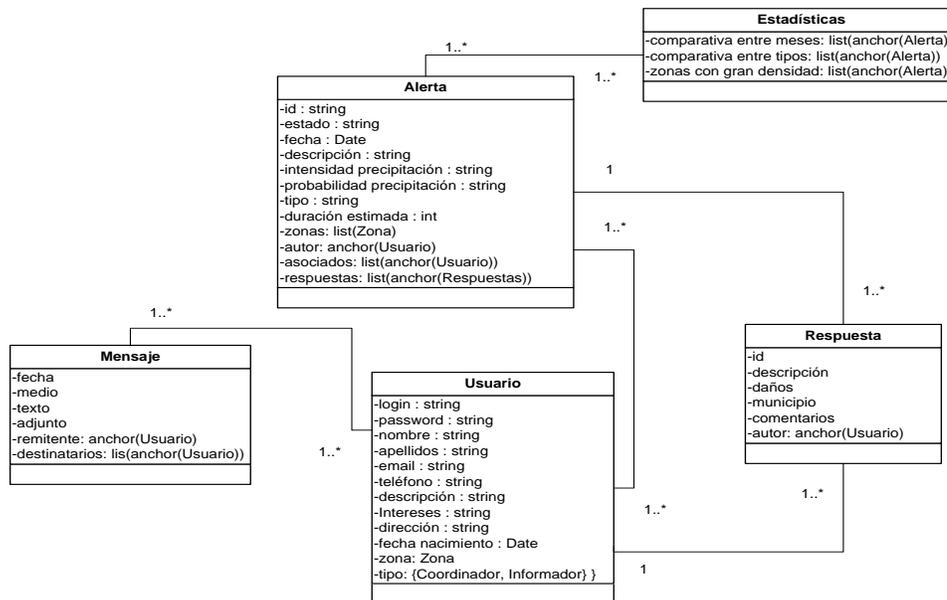


Figura 25. Esquema de clases navegacionales.

La principal estructura primitiva del espacio navegacional es la noción de contexto navegacional. En la [Figura 26](#) se muestra el diagrama contextual que contiene los distintos contextos navegacionales del sistema. Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada.



Como se observa en el siguiente diagrama, existen diversos caminos navegacionales para acceder a los distintos contextos. Al contexto de alerta se puede acceder por el id de la alerta, por fecha (a través de la estructura de acceso Timemap) y por estadística. Desde el contexto de alerta se puede acceder al contexto de respuesta, ya que las respuestas se clasifican por alerta y al contexto de usuario, puesto que las alertas tienen asociados usuarios. Al contexto de usuario se puede acceder a través de la comunidad autónoma a la que están asociados, según el rol que tienen, o a través de los contextos de alerta (usuarios asociados a la alerta) y mensaje (usuarios destinatarios). Los contextos de mensaje y estadística pueden ser accedidos por fecha y tipo respectivamente.

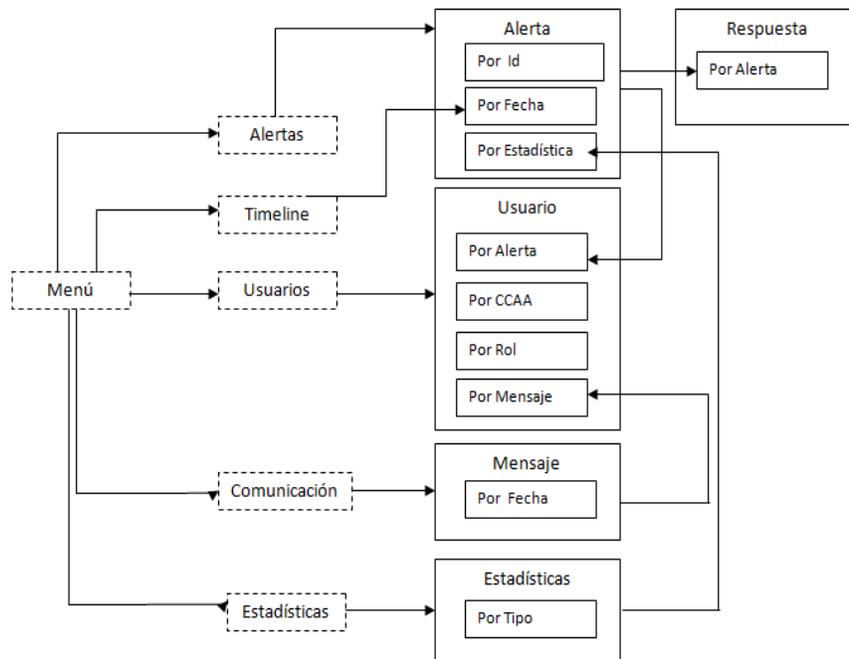


Figura 26. Esquema de contextos navegacionales.

Diseño de interfaz abstracta

Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz, esto significa definir la forma en la cual los objetos navegacionales pueden aparecer. Una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.



El modelo de interfaz ADVs (Vista de Datos Abstracta) especifica la organización y comportamiento de un contexto de navegación en la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación.

En la [Figura 28](#) se muestra la interfaz abstracta del contexto de navegación de alerta. En ella, se observan los distintos atributos que contiene el concepto de alerta así como otros ADV's correspondiente a los contextos "Usuarios por Alerta" y "Respuestas por Alerta". En el primero ([Figura 28](#)) se muestra una lista de enlaces a los usuarios asociados a la alerta. El segundo tiene la misma estructura pero con las respuestas generadas sobre la alerta ([Figura 29](#)).

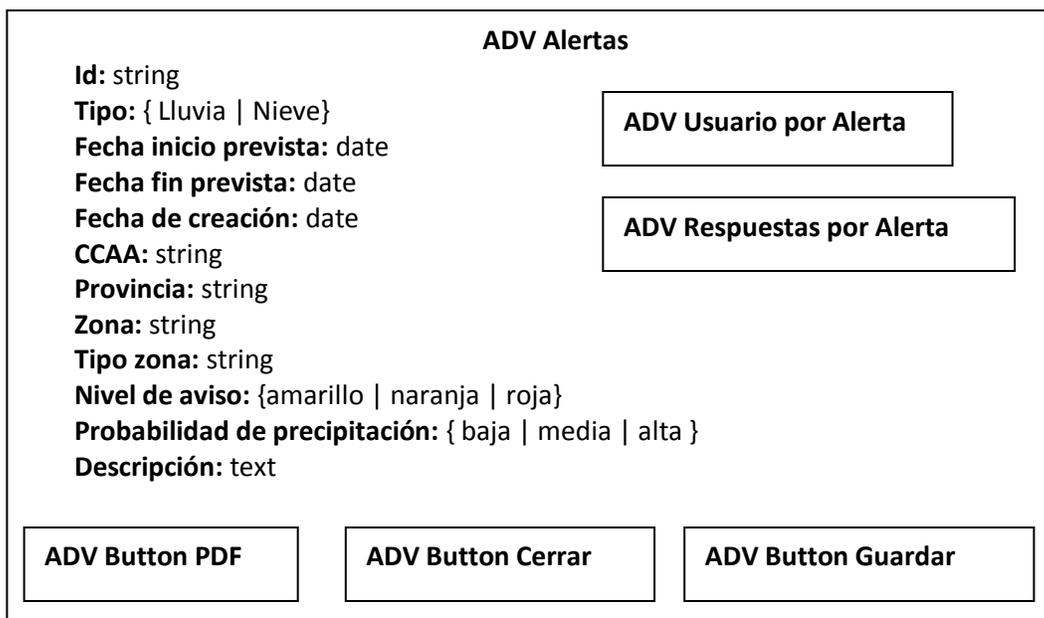


Figura 27. ADV Alerta.

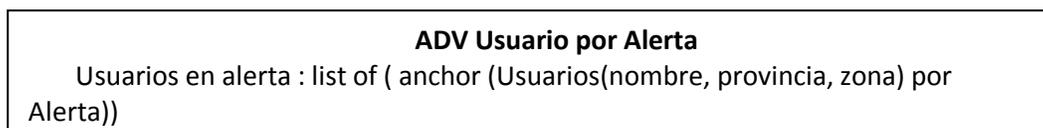


Figura 28. ADV Usuario por alerta.

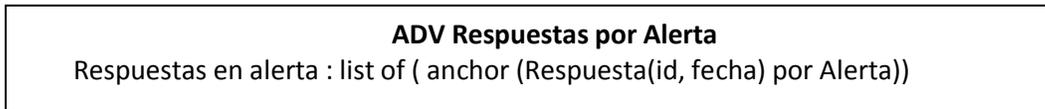


Figura 29.ADV Respuestas por alerta.

A continuación se muestra el contexto de navegación de usuarios, en la siguiente figura se pueden observar los atributos del concepto de usuario.

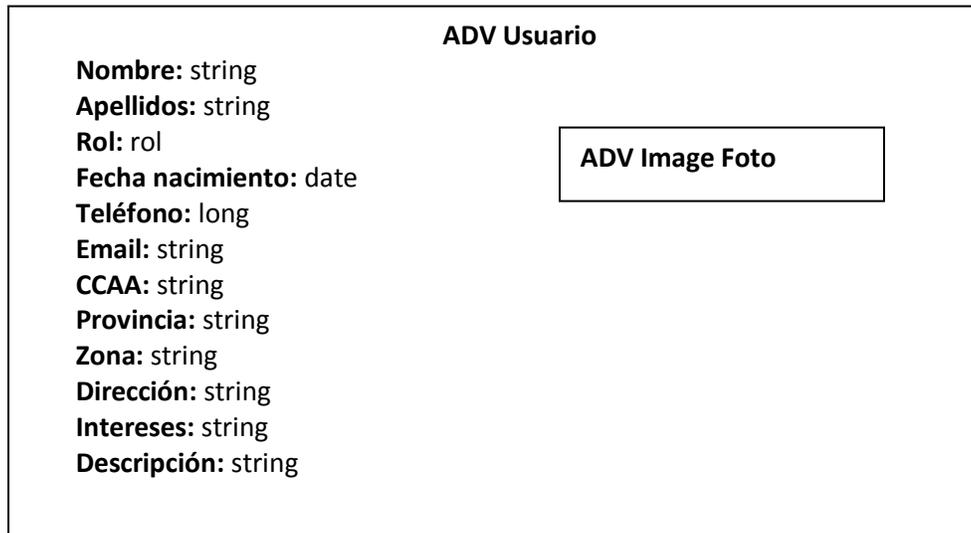


Figura 30. ADV Usuario.



Ya por último se visualizan los dos últimos contextos, mensajes y estadísticas. En estos conceptos, más simples, destaca la navegabilidad desde el ADV mensaje al ADV usuario por medio del destinatario. Otra opción de navegabilidad la tiene el ADV estadística que, por medio de las alertas que conforman dicha estadística, permite acceder al ADV de estas alertas.

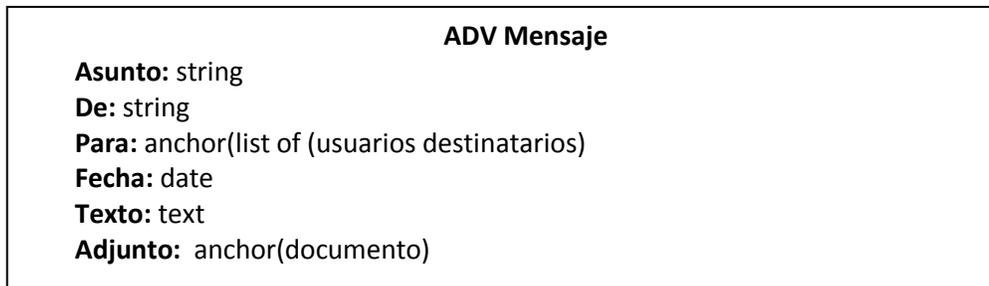


Figura 31. ADV Mensaje.

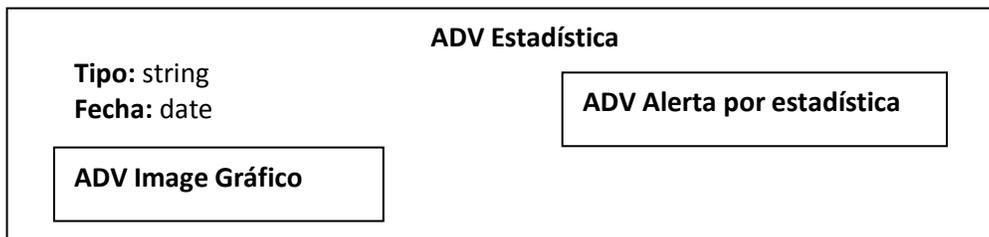


Figura 32. ADV Estadística.



Implementación

En esta fase se explica cómo se ha llevado a cabo la codificación de las ideas y modelos desarrollados en las fases de análisis y diseño. En cuanto a los detalles técnicos de implementación, destacar que se identificó como requisito la separación de contenido y estilo. Para cumplir este requisito se ha tenido un riguroso cuidado de que todo el contenido se encuentre en los archivos HTML, y toda la codificación de estilos se encuentre en los archivos CSS. De esta forma los archivos HTML importan el estilo a aplicar en sus contenidos cargando el archivo CSS a utilizar, tal y como se muestra en la siguiente figura.

```
<link rel="stylesheet" href="css/menus.css" type="text/css"/>
```

Figura 33. Ejemplo importación css.

Otra especificación detallada en los requisitos consistía en que se siguiesen estándares. En este sentido, los archivos de contenido respetan el estándar XHTML Strict 1.0 y los archivos de estilo el estándar CSS 2.1. Además, para una mejor organización los estilos han sido separados en cinco archivos CSS:

- **Layout.css:** contiene el estilo de los layout, de las cajas contenedoras (div) y los elementos genéricos.
- **Tables.css:** contiene el estilo de todas las tablas del prototipo.
- **Timemap.css:** contiene los estilos referentes al timemap.
- **Menus.css:** en él se encuentran los estilos de los diferentes menus aplicados en el prototipo.
- **headerFooter.css:** en él se hallan los estilos referentes a las zonas del header y el footer del layout.

Un aspecto muy importante en cuanto al estilo es la estructura de las pantallas. Conviene destacar que tal y como se estableció en los requisitos todas las páginas del prototipo tienen la misma estructura de layout, respetando por tanto la coherencia de la estructura en todo el prototipo. El esquema básico del layout se puede observar en la [Figura 34](#).

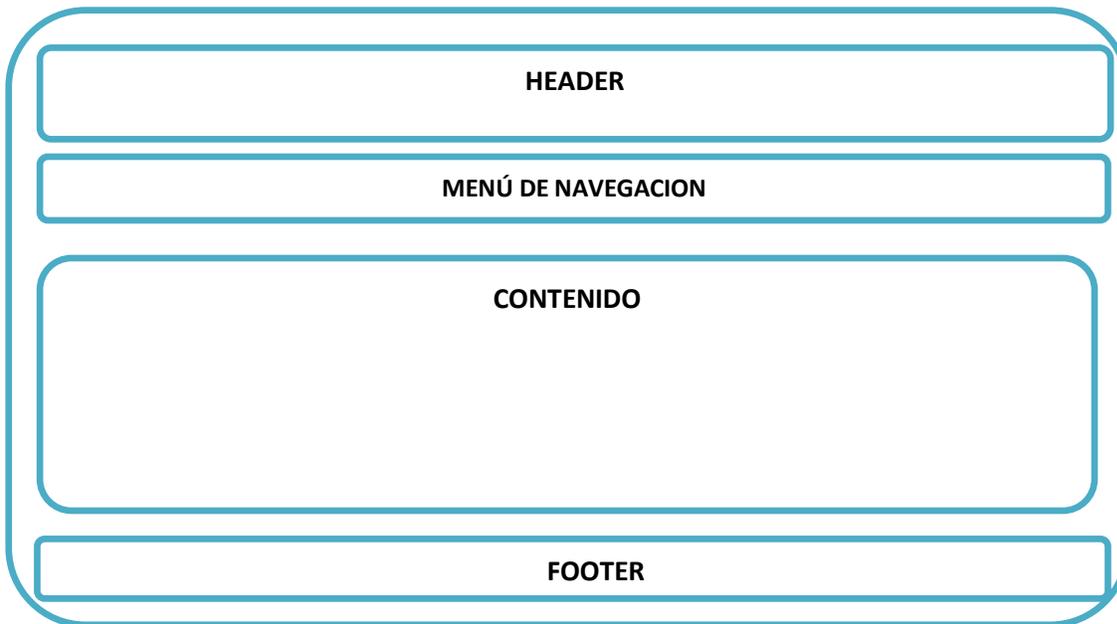


Figura 34. Estructura layout.

Las únicas excepciones a este diseño del layout, son la página principal, que sólo contiene header y footer, y los wizard. Se ha optado por utilizar wizards para las pantallas en las que el usuario vaya a realizar una tarea o meta que esté dividida en pasos.

Una vez descrita la estructura del layout, se hace necesario hablar del tipo de diseño de éste en términos de adaptación a distintas resoluciones. Se ha optado por un diseño semielástico en el que la anchura de todos los elementos sea dinámica, pero que cuente con los atributos min-width y min-height que garanticen que a partir de una medida el layout no disminuya, asegurando de esta forma la legibilidad y claridad en todos los contenidos. No sé ha optado por desarrollar un diseño completamente elástico porque al tratarse de un desarrollo en el que la información es el principal capital y tiene una importancia básica, era imprescindible garantizar la legibilidad y claridad de la información. En este sentido, un diseño elástico o dinámico puede acarrear problemas de legibilidad cuando se usan resoluciones más bajas de lo optimizado, tal y como se observa en la siguiente figura.



Figura 35. Mal ejemplo diseño dinámico.

En cuanto a las herramientas utilizadas, dado que es un prototipo que no tiene implementada la capa servidora, sólo se han utilizado herramientas que se ejecuten en el cliente, como es el caso de Javascript. Además, dentro de este lenguaje de scripting, se han utilizado las siguientes librerías:

- **Dojo Toolkit:** framework que contiene APIs para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Se ha utilizado de forma programática para aplicar drag & drop y tooltips de ayuda en los formularios.
- **Jscalendar:** está librería permite utilizar calendarios mensuales para introducir las fechas.
- **Sorttable:** clase que ha sido utilizada para ofrecer la ordenación de distintas tablas en función de un parámetro escogido.
- **Timemap:** librería Javascript que con ayuda de Google maps y SIMILE Timeline permite localizar sucesos en un mapa y en un timeline a la vez. Los datos de los sucesos se leen de un fichero KML.

Por último, hay que mencionar que la organización del proyecto se estructura a partir de una carpeta raíz que contiene los siguientes elementos:

- **Páginas html**
- **Carpeta css:** contiene todos los archivos css.
- **Carpeta script:** en ella se encuentran todos los archivos de javascript excepto los relativos al timemap.
- **Carpeta timemap:** contiene todos los archivos relativos al timemap.



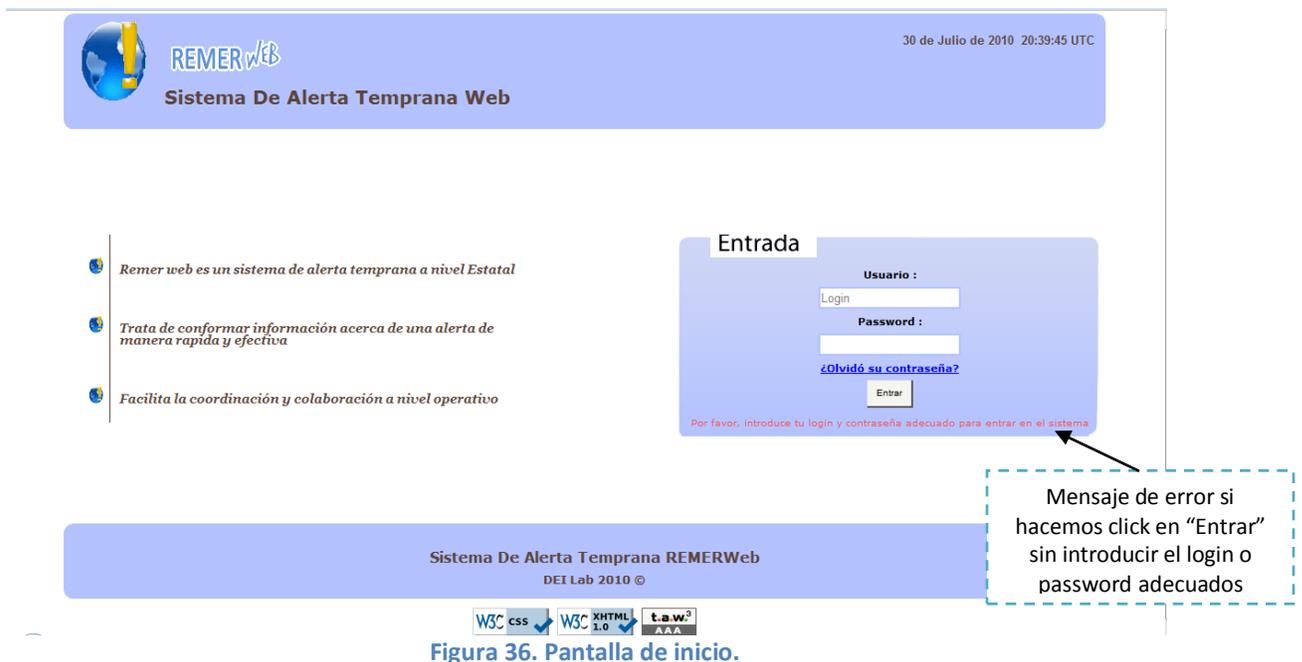
- **Carpeta images:** contiene todas las imágenes del proyecto.

4.3 El producto del desarrollo

En este apartado se va a hacer un breve repaso por el producto resultante de tal forma que el usuario pueda tener una obtener una visión global del resultado final del mismo.

El producto de desarrollo obtenido consiste en un prototipo web de alto nivel que, en un contexto de alerta temprana, aplique la idea de REMER como sistema en el que la colaboración entre distintos individuos de forma altruista.

En la [Figura 36](#), se puede observar la pantalla principal del prototipo, dónde el usuario debe autenticarse para poder acceder al sistema.



En la siguiente figura se observa la interfaz de gestión de alertas, en la que el usuario ve las tareas que tiene asociadas (rol informador), las tareas de su comunidad autónoma (rol coordinador autonómico) o todas las tareas (rol coordinador general). En el caso de la [Figura 37](#), tal y como podemos observar en la parte superior derecha, se trata de un usuario con rol coordinador



autonómico que está asociado a la Comunidad Autónoma Valenciana. Para acceder al detalle de las alertas se tienen distintas opciones:

- **Timemap:** es el diseño gráfico situado a la izquierda que muestra en una barra horizontal marcada las fechas de las alertas en los puntos donde se prevee que ocurra y su correspondencia gráfica en un mapa. De esta manera podemos buscar las alertas entre una fecha de inicio y otra de fin, y observar su localización en el mapa. Pinchando sobre las marcas tanto en la barra temporal como en el mapa se puede acceder al detalle de la alerta.
- **Tabla de alertas:** a la derecha del timemap se observa una tabla resumen con todas las alertas de la Comunidad Autónoma Valenciana. Pichando sobre ellas accedemos al detalle de la alerta, y pasando el ratón sobre ellas se abre una ventana que nos indica en el mapa su localización.

The screenshot shows the 'Sistema De Alerta Temprana Web' interface. The header includes the logo, the title 'Sistema De Alerta Temprana Web', and user details: 'Usuario: Sergio Herranz Huertas', 'Rol: Coordinador', 'CCAA: Comunidad Valenciana', and the date '31 de Julio de 2010 11:50:49 UTC'. The menu contains 'Gestión de alertas', 'Gestión de usuarios', 'Mi perfil', 'Comunicación', 'Estadísticas', and 'Ayuda'. The main content area features a map of the Iberian Peninsula with a timeline from 2009 to 2013. A table titled 'Alertas Activas En C.Valenciana' lists three active alerts:

Id	Tipo	Fecha
Alerta I-SU-T11	Lluvia	30-10-2010
Alerta I-SR-T22	Lluvia	14-12-2010
Alerta I-H-T24	Nieve	14-01-2011

The footer contains 'Sistema De Alerta Temprana REMERWeb' and 'DEI Lab 2010 ©'.

Figura 37. Pantalla gestión de alertas.

Una tarea fundamental en la alerta temprana es la propia declaración de alerta. Para crear una alerta, el usuario haría click en la pantalla anterior en el botón "Crear Alerta", este botón abrirá un wizard en el que completar la tarea. Se ha decidido usar wizard para todas las tareas que se dividan en pasos, tal y como recomiendan diversos autores [30].



En la Figura 38 se observa como en la parte superior del wizard se cuenta con una barra de progreso, que indica al usuario el número y una breve descripción de los pasos que tiene que llevar a cabo para finalizar la tarea, así como el paso en el que se encuentra. De esta forma se le proporciona una información adecuada de su situación dentro del sistema, y dentro de la tarea en cuestión.

Además hay que destacar que el formulario está diseñado para prevenir que el usuario pueda cometer errores al completarlo. Todos los campos de entrada cuentan con un valor por defecto que indica lo que se espera en el campo, un tooltip que explica brevemente qué se debe introducir y todos los campos obligatorios están marcados con los símbolos “(*)”. Aún así, si algún campo obligatorio se queda sin rellenar, el sistema avisa y señala los campos que faltan por introducir, tal y como se muestra en la siguiente figura.

Otra ventaja de utilizar wizard, es que proporciona al usuario un control y libertad mayor, ya que si llega un punto de la tarea en el que se equivoca o simplemente quiere salir sin acabar, simplemente cerrando la ventana volverá a la pantalla de alertas. También para mejorar el control, a partir del segundo paso se cuenta con botones de “Volver”, que permiten al usuario hacer y deshacer cambios antes de finalizar la tarea.

Crear Alerta

Paso 1. Datos Generales | Paso 2. Datos Especificos | Paso 3. Asociar Usuarios | Paso 4. Confirmación

Existen datos obligatorios sin introducir

(*) Tipo de alerta: Lluvia (*) Fecha inicio prevista: fechalnicio ...

Fecha fin prevista: fechaFin < Seleccione la fecha prevista para el fin de la alerta

Localización

(*) CCAA : ..Seleccione CCAA (*) Provincia : Castellón

(*) Zona : ..Seleccione Zona

Siguiente

Sistema De Alerta Temprana REMERWeb
DEI Lab 2010 ©

Figura 38. Pantalla creación de alerta paso 1.



En el paso tres de este mismo proceso de creación de una alerta, el creador de la misma debe asociar a la alerta una serie de usuarios que son los que van a conformar la respuesta a la misma. Para ello, el sistema propondrá los usuarios ordenados por proximidad geográfica a la alerta. En la [Figura 39](#) se puede observar como en la tabla de la izquierda se muestran una serie de usuarios que se pueden asociar a la alerta, en la tabla de la derecha tanto el título (“Usuarios seleccionados”), como el texto de la tabla (“Arrastre los usuarios hasta aquí”) dan una información bastante completa de cómo asociar los usuarios a la alerta en este paso, que básicamente consiste en utilizar el mecanismo denominado drag&drop (arrastrar y soltar). Aún así, y para evitar posibles errores, la pantalla cuenta con un tooltip de ayuda que explica más en detalle cómo se puede completar este paso.

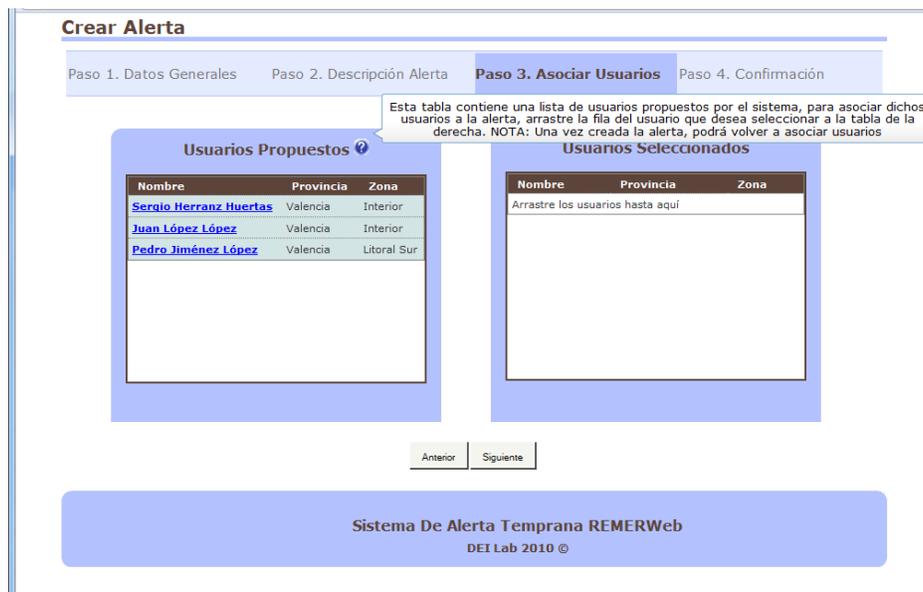


Figura 39. Pantalla creación alerta paso 3.

Imaginemos que una vez finalizado correctamente el proceso de creación de la alerta, se autentica en el sistema un usuario de rol informador que ha sido asociado a dicha alerta. Éste observaría la pantalla de la [Figura 37](#) sólo con las alertas asociadas que éste tiene asociadas, en este caso con una nueva alerta que se acaba de crear y a la que ha sido asociado.

El siguiente paso sería suministrar por parte de los informadores información sobre la alerta. Para poder hacerlo, lo primero que necesita conocer es la información acerca de la misma, para ello debe hacer click en la alerta en cuestión, ya sea en la tabla o en el timemap. Una vez esto ocurra



se visualizará la pantalla mostrada en la [Figura 40](#). En esta pantalla se muestra toda la información de la alerta, a la izquierda la información propia de la alerta (Id, fechas, localización, detalles técnicos) y a la derecha los usuarios asociados y las respuestas con las que cuenta la alerta. Destacar como detalles de diseño el breadcrumb que mejora la navegabilidad además de proveer al usuario de información de su situación en el sistema. Otro detalle que ayuda a que el usuario identifique su lugar en la aplicación es que todas las páginas cuentan con un título descriptivo.

Título descriptivo

REMERWeb | Gestión Alertas | Ver Alerta

REMERWeb | Gestión Alertas | Ver Alerta

Usuario: Sergio Herranz Huertas
Rol: Informador
CCAA: Comunidad Valenciana
31 de Julio de 2010 12:58:57 UTC

Gestión de alertas | Gestión de usuarios | MI perfil | Comunicación | Estadísticas | Ayuda

Gestión de alertas > Ver PCL 31102010 2100 Alerta I-SU-T11

Breadcrumb

10 2100 Alerta I-SU-T11

Fecha inicio prevista: 31 de Octubre de 2010 21:00 UTC
Fecha fin prevista: 31 de Noviembre de 2010
Fecha Creación : 25 de Septiembre de 2010 12:00 UTC

Localización

CCAA : Comunidad Valenciana
Provincia : Valencia
Zona : Interior
Tipo : Semilurbano
[Detalle de localización](#)

Descripción De La Alerta

Nivel de aviso : Naranja

Probabilidad de precipitación : Entre el 40% y el 70%

Descripción : la gota fría es una perturbación atmosférica extratropical no frontal que puede provocar precipitaciones excepcionalmente violentas e intensas durante unas horas o días, acompañadas de numerosos rayos y de granizo

Usuarios Asociados

Nombre	Provincia	Zona
Sergio Herranz Huertas	Valencia	Interior
Juan López López	Valencia	Interior
Pedro Jiménez López	Valencia	Litoral Sur

Respuestas

Id	Municipio	Fecha
13-C	Chesle	25/09/2010

Sistema De Alerta Temprana.REMERWeb

Figura 40. Pantalla detalle alerta.



5 Evaluación

La construcción de cualquier herramienta software tiene como objetivo satisfacer una necesidad planteada. En este apartado se evalúa si el prototipo satisface esa necesidad planteada mediante una serie de pruebas que lo verifiquen.

5.1 Proceso de evaluación

Plan de pruebas

En términos generales, se pueden distinguir dos tipos de evaluaciones durante el proceso de desarrollo: verificaciones y validaciones. Según el IEEE éstas se definen como:

- **Verificación:** Proceso de determinar si los productos de una determinada fase del desarrollo de software cumplen o no los requisitos establecidos durante la fase anterior.
- **Validación:** Proceso de evaluación del software al final del proceso de desarrollo para asegurar el cumplimiento de las necesidades del cliente.

La verificación tiene que ver típicamente con errores de los desarrolladores que no han construido bien el producto. Mientras que la validación tiene que ver con errores de los desarrolladores al malinterpretar las necesidades del cliente y por tanto no están construyendo correctamente la funcionalidad requerida. Estos últimos no son evaluables en esta solución puesto que se trata de un prototipo en el que la funcionalidad requerida por el cliente va a ser evaluada por él mismo más adelante realizando un diseño participativo. Por ello sólo se han establecido una serie de pruebas que permitan verificar que el sistema cumple los requisitos básicos de calidad establecidos. El plan de pruebas se puede consultar en la gestión de proyecto en el apartado Especificación de pruebas.



5.2 Análisis de resultados

En este apartado se detalla el resultado de las pruebas de aceptación identificadas en la gestión de proyecto.

Prueba	Descripción	Resultado	Comentarios
PA-01	Separación contenido-estilo	Positivo	Al cambiar los archivos .css se verifica que cambia el estilo aplicado a los contenidos.
PA-02	Cumplimiento estándar XHTML 1.0	Positivo	Los resultados del validador confirman que ninguna página del prototipo contiene ningún error ni aviso respecto al cumplimiento del estándar XHTML 1.0
PA-03	Cumplimiento estándar CSS 2.1	Positivo	Los resultados del validador confirman que ninguna página del prototipo contiene ningún error ni aviso respecto al cumplimiento del estándar CSS 2.1
PA-04	Cumplimiento de los principios de accesibilidad web de la WCGA	Positivo	Los resultados del validador indican un nivel de accesibilidad AAA.
PA-05	Cumplimiento de los principios de accesibilidad web Wave	Positivo	Los resultados del validador confirman que ninguna página del prototipo contiene ningún error ni aviso respecto al análisis de este validador
PA-06	Layout uniformes y constantes	Positivo	Se verifica que los layouts son uniformes en todo el prototipo a excepción de la página principal y los wizard. Estas excepciones han sido argumentadas.
PA-07	Prevención y recuperación de errores	Positivo	Se previenen errores gracias a los valores por defecto y tooltips de los inputs en todos los formularios. En caso de cometer un error, el prototipo avisa de dicho error y le sitúa siempre con el mismo formato (color rojo).
PA-08	Feedback adecuado en el que el usuario siempre sabe en qué punto de la aplicación se encuentra	Positivo	El breadcrumb y los títulos ofrecen siempre información sobre dónde se encuentra el usuario dentro del prototipo.
PA-09	Feedback adecuado cuando el usuario lleva a cabo una	Positivo	Se observa como todas las tareas son llevadas a cabo en wizards, y éstos siempre contienen una barra que indica qué pasos tiene dicha tarea y en cuál de ellos se encuentra el



	tareas		usuario.
PA-10	Compatibilidad distintos navegadores	Positivo	Se observa como todas las la funcionalidad y el estilo es el mismo en la última versión de los siguientes navegadores: Internet Explorer, Mozilla Firefox, Google Chrome y Opera.

Tabla 73. Resultado pruebas.



6 Conclusión

Como punto final a este documento, este último apartado aborda las conclusiones globales obtenidas por el autor a lo largo del desarrollo del proyecto. En él se analizan las aportaciones realizadas, las posibles líneas de mejora y ampliación, y las dificultades encontradas a lo largo del proyecto.

6.1 Aportaciones realizadas

La principal aportación de este proyecto es la elaboración de un prototipo web de alto nivel, centrado en la interfaz de usuario, que traslada la actividad de la comunidad REMER a un entorno web, proporcionando así un contexto de colaboración alternativo a la radiofrecuencia. Este prototipo debía cumplir con los siguientes requisitos:

- **Seguir una metodología de desarrollo web:** se ha seguido la metodología de desarrollo web OOHD, haciendo de esta forma del análisis y diseño del prototipo, un proceso claro y bien documentado.
- **Principios de diseño:**
 - **Usabilidad:** a partir de los principios heurísticos de Nielsen, se han elaborado una serie de requisitos que posteriormente han sido verificados en el plan de pruebas.
 - **Accesibilidad:** se han seguido las Pautas de Accesibilidad Web 1.0 (WCAG). Este requisito se ha validado con el Test de Accesibilidad Web (TAW), que no ha identificado ningún error automático para un nivel de accesibilidad AAA, y con la herramienta Wave que tampoco arrojo ningún error en términos de accesibilidad. Destacar también que se ha conseguido una accesibilidad tecnológica en la que el prototipo es accesible, al menos, desde los siguiente navegadores en sus últimas versiones: Internet Explorer, Mozilla Firefox, Google Chrome, Safari y Opera.
- **Detalles técnicos de implementación:**
 - **Separación contenido-estilo:** todo el código referente al estilo del prototipo se encuentra en archivos CSS de forma independiente.



- **Seguir estándares:** para el estilo se ha cumplido el estándar CSS 2.1 y para el contenido el estándar XHTML Strict 1.0. Esto ha sido validado por las distintas herramientas que proporciona la W3C.

Todos estos requisitos de calidad han establecido las bases para aportar al futuro sistema final las importantes características de interoperabilidad, portabilidad y accesibilidad.

6.2 Trabajos futuros

Todos los posibles trabajos futuros están relacionados con el desarrollo de una herramienta colaborativa definitiva encuadrada en el proyecto URThey. En este sentido, el prototipo elaborado tiene como fin servir como base para diseñar esta futura herramienta colaborativa. Para completar este objetivo, se deberán seguir los siguientes pasos:

1. **Validar el prototipo con los miembros de la comunidad REMER como parte de un proceso de diseño participativo:** esto servirá para identificar el proceso y forma de trabajo de la comunidad REMER en la web, permitiendo además detectar fallos o aspectos susceptibles de mejorar en lo que se refiere a su uso.
2. **Desarrollar servicios para soportar la actividad de comunidades virtuales:** estos servicios estarían orientados a mejorar las características de comunicación, cooperación y coordinación. Mejorando estos tres puntos, el objetivo es llegar a desarrollar una comunidad virtual en el contexto de la alerta temprana, en el que en una situación de posible emergencia cada individuo de la comunidad sea importante para reducir los riesgos y prepararse para actuar eficazmente ante un posible peligro derivado de dicha alerta.

Por último, otra línea de trabajo futuro de gran utilidad y que puede facilitar y hacer más cómoda la colaboración de la comunidad, podría ser **llevar a cabo una versión del sistema adaptada al uso de dispositivos móviles** que satisfaga el importante número de usuarios de Internet que utilizan dispositivos móviles para navegar, que además crece día a día.

6.3 Problemas encontrados



Aunque el resultado global del proyecto es positivo y se han cumplido los objetivos establecidos, éste no ha estado exento de problemas y dificultades, que se describen a continuación.

- **Falta de información de los procesos de REMER:** el principal problema encontrado, y una de las cosas que se pretende resolver con este proyecto, es la falta de información acerca de cómo lleva a cabo REMER los procesos de gestión de alertas y comunicación. La solución a este problema vino dada mediante la aplicación al diseño de las tareas de la máxima coherencia y flexibilidad a partir de la poca información con la que se contaba, a la espera de llevar a cabo un diseño participativo con los usuarios de la comunidad, en el que se evaluase el prototipo y refinasen los procesos de gestión de alertas y comunicación del mismo.
- **Desconocimiento de la metodología OOHDM:** OOHDM es un método de desarrollo web extenso y de gran envergadura. Debido a mi inexperiencia con este método fue necesario dedicar algún tiempo para comprender los aspectos fundamentales del mismo y poder aplicarlo de esta forma al proyecto.
- **Dojo Toolkit y el cumplimiento de estándares:** Dojo Toolkit es un framework de Javascript que ha proporcionado una gran utilidad al sistema desarrollado. Este framework se puede utilizar de dos maneras: programáticamente y declarativamente. Para utilizarlo declarativamente, se hace necesario utilizar atributos como “dojoType”, que por supuesto no son validados con el estándar XHTML Strict. Es por esto por lo que se ha utilizado la forma programática en la que en un método Javascript se van asociando los elementos Dojo a los elementos HTML, ejecutándose este método al cargar la página en cuestión.
- **Accesibilidad:** cumplir con las pautas de accesibilidad web de la W3C es una tarea difícil que requiere ser muy cuidadoso con el código HTML. Por poner algunos ejemplos, para cumplir una accesibilidad AAA, todas las tablas deben contar con el elemento “Summary”, todos los input deben tener un valor por defecto, no se deben utilizar tablas para maquetar el espacio web, se deben utilizar unidades porcentuales para el tamaño de las fuentes, proporcionar contenidos equivalentes al contenido visual.... En resumen, diseñar e implementar contenidos accesibles aumenta inevitablemente la complejidad y el tiempo dedicado a la solución. A cambio, por supuesto, se obtienen numerosas ventajas ya comentadas en puntos anteriores.



- **Timemap:** esta librería Javascript basada en Google Maps y SIMILE Timeline puede leer los datos de los eventos directamente en el archivo HTML o en distintos formatos como JSON o KML. Se optó por leer los datos de un fichero KML, puesto que proporcionaba una independencia entre los datos y la presentación de los mismos. Esto aumentó el tiempo dedicado a la implementación del Timemap.

6.4 Opiniones personales

Como es lógico, la consecución de un proyecto fin de carrera es una empresa de gran envergadura que ha supuesto un gran esfuerzo y dedicación. En él he tenido que aprender a planificar mi tiempo, tomar mis propias decisiones y superar los distintos problemas con los que me he ido encontrando.

Debido a las condiciones personales a las que me enfrentaba, en las que contaba con un periodo de tiempo ajustado para llevar a cabo el proyecto, he tenido que organizar mejor el tiempo disponible y dedicarle las horas que fuesen necesarias para cumplir con los estrictos plazos marcados.

Por otra parte, destacar que partía con el objetivo personal de que el proyecto, aparte de un requisito para acabar la carrera, debía suponer un enriquecimiento personal. Por esta razón he intentado escoger, dentro de las posibilidades existentes, tecnologías que no conocía con el propósito de aprenderlas. Por poner un ejemplo, podía haber escogido la metodología de desarrollo Ariadne, que estudié este año en la asignatura “Diseño y Evaluación de Sistemas Hipermedia” y me hubiese ahorrado mucho tiempo de aprendizaje, pero decidí escoger OOHDM porque me iba a permitir conocer una nueva e importante metodología.

Como conclusión final que me ha aportado el proyecto, destacaría que me ha permitido entender que un desarrollo web no consiste sólo en implementar algo con un buen diseño gráfico y una aceptable organización. Para desarrollar una web de calidad hoy en día, hay que tener en cuenta aspectos como la usabilidad, accesibilidad, mantenibilidad...puesto que van a ser claves en el éxito de un desarrollo web.



7 Bibliografía

- [1]. Karlos Pérez de Armiño, Diccionario de Acción Humanitaria y Cooperación al desarrollo, 2005. <http://dicc.hegoa.efaber.net/listar/mostrar/119>
- [2]. CONRED, El ciclo de los desastres.
<http://www.conred.gob.gt/boletines/comunicados-tecnicos/006.pdf>
- [3]. Dirección General de Protección Civil y Emergencias, Red Radio de Emergencias (REMER)- Información General.
http://www.proteccioncivil.org/es/DGPCE/Informacion_y_documentacion/catalogo/carpeta02/carpeta24/informacion.html
- [4]. Sistema Nacional de Información para la Prevención y Atención de Desastres.
<http://sinadeci.indeci.gob.pe/PortalSINPAD/Default.aspx?ItemId=94>
- [5]. Global Survey of Early Warning Systems, United Nations.
<http://www.unisdr.org/ppew/info-resources/ewc3/Global-Survey-of-Early-Warning-Systems.pdf>
- [6]. Vademécum Remer, Prólogo
http://www.proteccioncivil.org/es/DGPCE/Informacion_y_documentacion/catalogo/carpeta02/carpeta24/vademecum/vade01.htm
- [7]. SIGAME: Web-based System for Resources Management on Emergencies. Laura Montells, Susana Montero, Paloma Díaz, Ignacio Aedo.
- [8]. Virtual Communities of Practice:
Design Directions for Technology-mediated Collaboration in the Early Warning Activity.
D.Díez, 2010
- [9]. S. Murugesan, Y. Deshpande, S. Hansen, A. Ginige. Web Engineering : A New Discipline for Development of Web-Based Systems, 2001.
<http://www.ldc.usb.ve/~abianc/materias/ci4712/murugesan.pdf>
- [10]. SCHWABE, D. ROSI, G.; BARBOSA, S. Systematic Hypermedia Application Design with OOHDM. Proceedings of the ACM International Conference on Hipertext (Hypertext'96). Washington DC, March, 16-20, 1996.
<http://www-di.inf.puc-rio.br/schwabe/papers/TAPOSRevised.pdf>



- [11]. A Conference Review System in OOHDm. D. Schwabe, P. Vilain, G. Rossi, 2001.
<http://users.dsic.upv.es/~west/iwwost01/files/presentations/DanielSchwabe/AConfReviewSystemInOOHDm%28present%29.pdf>
- [12]. Propuesta de un modelo navegacional para el desarrollo de aplicaciones basadas en OOHDm. Ricardo Soto De Giorgis, Wenceslao Palma Muñoz, Silvana Roncagliolo De La Horra.
http://www.inf.ucv.cl/~rsoto/papersPUCV/Propuesta_de_un_modelo_navigacional.pdf
- [13]. S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (WebML): a modeling language for designing web sites. Computer Networks (Amsterdam, Netherlands: 1999), 2000.
- [14]. Modeling Web Business Processes with OO-H and UWE1 NORA KOCH AND ANDREAS KRAUS.
- [15]. ADM, Método de diseño para la generación de prototipos web rápidos a partir de modelos. Susana Montero, Ignacio Aedo y Laura Montells.
<http://users.dsic.upv.es/workshops/dsdm06/files/dsdm06-03-Montero.pdf>
- [16]. Jakob Nielsen, Introduction to Usability
<http://www.useit.com/alertbox/20030825.html>
- [17]. Ben Shneiderman, Designing the User Interfaces.
- [18]. La Ingeniería de la Usabilidad y de la Accesibilidad aplicada al diseño y desarrollo de sitios web, Universidad de Lleida.
http://griho2.udl.cat/mpiuu/wiki/images/IngUsabWEBxJLores_TGranollersx.pdf
- [19]. Legislación Española sobre la Accesibilidad, SIDAR.
<http://www.sidar.org/recur/direc/legis/espa.php>
- [20]. Luc van Lancker. XHTML 1 y CSS 1 y 2.1: los nuevos estándares de la web 2.0, 2007
- [21]. Platinum Edition Using XHTML™, XML, and Java™ 2By: Eric Ladd; Jim O'Donnell; Mike Morgan; Andrew H. Watt
- [22]. Vocabularios XML: XHTML y WML, Daniel Gayo, Universidad de Oviedo
<http://petra.euitio.uniovi.es/~labra/cursos/ext02/xhtmlWml.PDF>
- [23]. HTML, XHTML & CSS, Visual Quick Start Guide, Elizabeth Castro



- [24]. W3C, <http://www.w3.org/QA/TheMatrix>
- [25]. W3C, <http://www.w3.org/TR/>
- [26]. W3C,CSS <http://www.w3.org/Style/CSS/>
- [27]. Mari-Carmen Marcos et al.. *Evaluación de la usabilidad en sistemas de información terminológicos online* [en línea]. "Hipertext.net", núm. 4, 2006. <http://www.hipertext.net>
- [28]. C. Melissa McClendon , Larry Regot , Gerri Akers. , <http://www.umsl.edu/~sauter/analysis/prototyping/intro.html>
- [29]. A Diagrammatic Tool for Representing User Interactions, Patricia Vilain, Daniel Schwave.
- [30]. Designing interfaces: Patterns for Effective Interaction Design, Jeniffer Tidwell



Anexo I. Control de versiones

En este anexo se incluyen las diferentes versiones que se han realizado de la memoria, para poder hacer un seguimiento de las distintas modificaciones que se han realizado.

Versión	Fecha	Descripción
1.0	15/07/2010	Versión inicial.
1.1	30/07/2010	Cambio en la revisión de las tecnologías, en la que se añaden las herramientas de evaluación. Cambios también en los requisitos.
1.3	23/08/2010	Cambios en la gestión de proyecto, se pretende una gestión de proyecto simulada.
2.0	30/08/2010	Versión Final. Cambios en las conclusiones

Tabla 74. Control de versiones.



Anexo II. Seguimiento de proyecto fin de carrera

La planificación de tareas es un instrumento de esencial en la gestión de cualquier proyecto ya que permite organizar el tiempo del que se dispone, ofreciendo la posibilidad de realizar un seguimiento para determinar si se están realizando las tareas en plazo y límite determinados. Destacar que tanto las planificaciones como el seguimiento son herramientas reales que, con la ayuda de mi tutor, he utilizado para organizar el tiempo dedicado al proyecto.

Planificación inicial

A continuación se muestra la planificación que estimé, junto a mi tutor, al comienzo del proyecto.

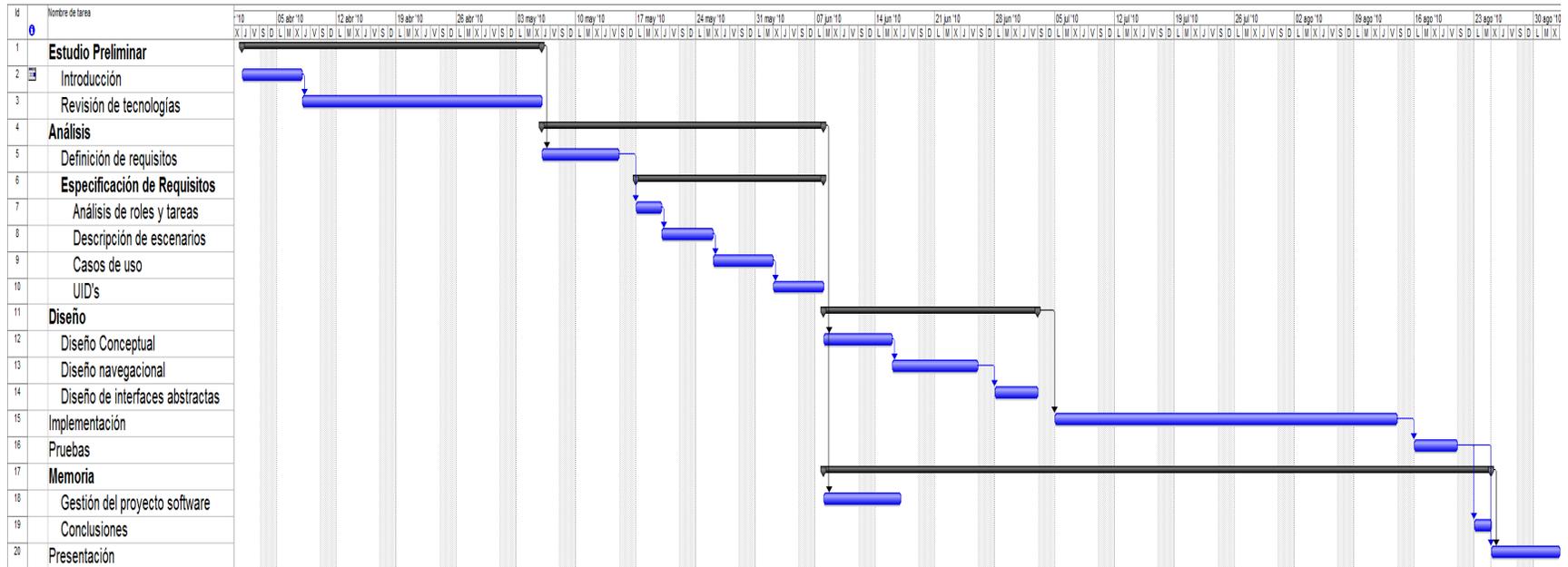


Figura 41. Planificación inicial.

Seguimiento

Tareas	Planificado		Real		Avance	Tiempo	Observaciones
	Desde	Hasta	Desde	Hasta	%	dedicado(días)	
Desarrollo	1/4/2010	20/08/2010			100		
Análisis	6/5/2010	7/6/2010	5/5/2010	11/6/2010	100		
Definición de requisitos	6/5/2010	14/5/2010	5/5/2010	14/05/2010	100	8	Análisis de los requisitos
Análisis de tareas	17/5/2010	19/05/2010	17/5/2010	19/05/2010	100	3	Definición de tareas de usuario
Definición de escenarios	20/5/2010	25/5/2010	20/5/2010	27/5/2010	100	6	Diseño de escenarios
Casos de uso	26/5/2010	1/6/2010	26/5/2010	7/6/2010	100	7	Definición de casos de uso
UID	2/6/2010	7/6/2010	8/6/2010	11/6/2010	100	4	Definición de diagramas de interacción
Diseño	8/6/2010	2/7/2010	14/6/2010	19/7/2010	100		Diseño del sistema
Diseño Conceptual	8/6/2010	15/6/2010	14/6/2010	22/6/2010	100	7	Diseño del modelo conceptual UML
Diseño de navegación	16/6/2010	25/6/2010	23/6/2010	8/7/2010	100	12	Diseño del modelo navegacional
Diseño de pantallas	28/6/2010	2/7/2010	9/7/2010	19/7/2010	100	7	Diseño de interfaces abstractas
Implementación	5/7/2010	13/8/2010	20/7/2010	25/8/2010	100	27	Programación de la solución.
Evaluación	16/8/2010	20/8/2010	26/8/2010	1/9/2010	100	5	Fase de evaluación de la interfaz
Defensa	1/4/2009	1/9/2010			100		
Memoria	1/4/2010	24/08/2010	1/4/2010	2/9/2010	100		Redacción de la memoria del proyecto.
Introducción	1/4/2010	7/4/2010	1/4/2010	8/4/2010	100	6	Introducción al proyecto. Objetivos.
Estudio del problema	8/4/2010	5/5/2010	9/4/2010	4/5/2010	100	18	Estudio del contexto del proyecto y descripción del problema
Gestión del proyecto	8/6/2010	16/06/2010	14/6/2010	23/06/2010	100	8	Forma en la cual se ha realizado el proyecto.
Conclusiones	23/8/2010	24/8/2010	2/9/2010	2/9/2010	100	1	Conclusiones personales
Presentación	25/8/2010	1/9/2010	3/9/2010	13/9/2010	100	7	Elaboración de la presentación.

Tabla 75. Seguimiento del proyecto.

Planificación final

Aunque se ha cumplido en su mayoría con la planificación inicial, no siempre ha sido posible, ya que ha habido ciertas tareas que han llevado más tiempo del inicialmente previsto. Este es el caso del diseño, que al emplear una metodología desconocida en un principio como es OOHDM, ha sufrido los diversos retrasos propios del estudio y documentación sobre la misma hasta que ha sido posible aplicarla.

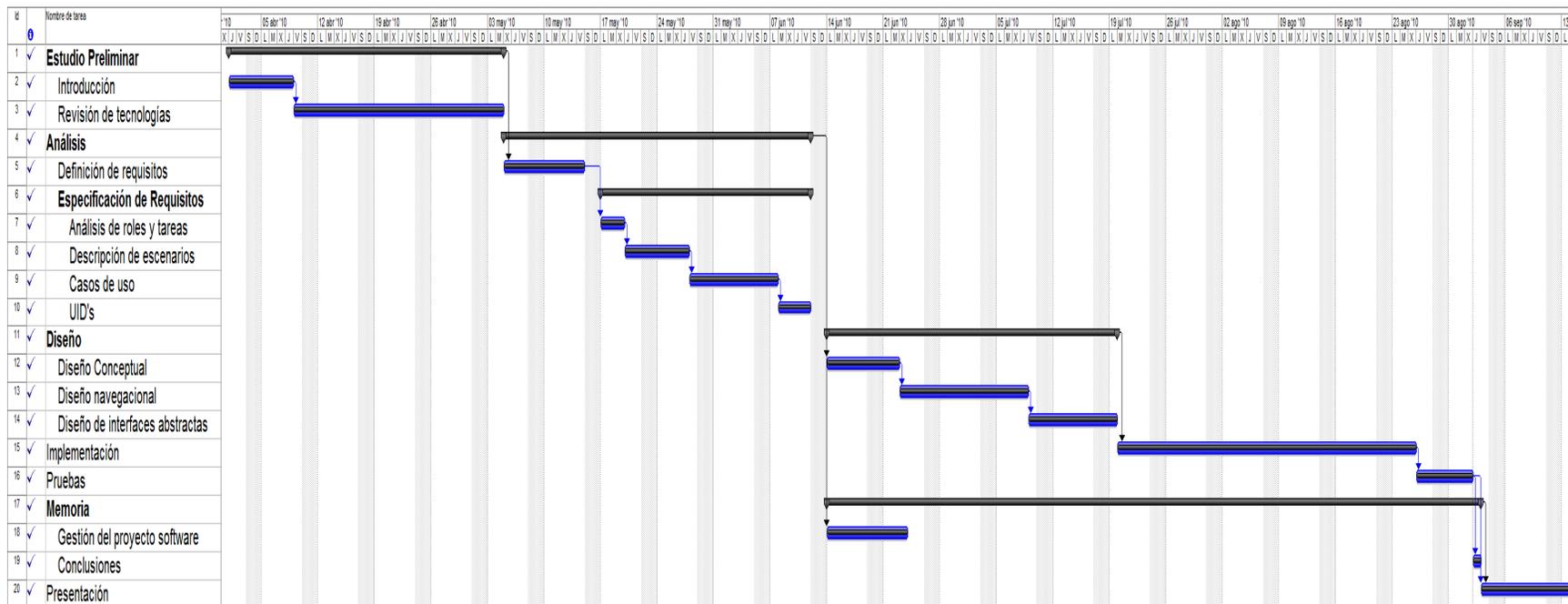


Figura 42. Planificación final

