

Diseño de una Aplicación Web Multidispositivo para la Notificación de Eventos

Proyecto Fin de Carrera



Ingeniería Informática

Autor: Carlos Ruano Martínez

Tutor: Fausto J. Sainz de Salces

Octubre 2009



ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	6
1.1. BREVE DESCRIPCIÓN DEL TRABAJO	7
2. ESTADO DEL ARTE	9
2.1. MARCO HISTÓRICO.....	9
2.2. PRINCIPIOS DE DISEÑO PARA DISPOSITIVOS MÓVILES	15
2.2.1. <i>DISEÑO DE DISPOSITIVOS MÓVILES EN APLICACIONES WEB</i>	16
2.2.2. <i>LIMITACIONES DE LOS DISPOSITIVOS MÓVILES</i>	18
2.3. DISEÑO PARA MAYORES: PRINCIPIOS Y RECOMENDACIONES	20
2.3.1. <i>EFFECTOS DEL ENVEJECIMIENTO</i>	20
2.3.2. <i>PÉRDIDA SENSORIAL</i>	21
2.3.3. <i>PÉRDIDA DE LA VISIÓN</i>	21
2.3.4. <i>PÉRDIDA DE LA AUDICIÓN</i>	22
2.3.5. <i>LA PERCEPCIÓN TÁCTIL</i>	25
2.3.6. <i>DETERIORO COGNITIVO</i>	26
2.3.7. <i>TECNOLOGÍA EN EL AMBIENTE ACTUAL</i>	26
3. METODOLOGÍA DE DESARROLLO	30
3.1. DETERMINACIÓN DEL PROBLEMA.....	30
3.2. DESARROLLO DE LA SOLUCIÓN.....	30
3.3. EVALUACIÓN DE LA SOLUCIÓN.....	32
4. DESCRIPCIÓN DEL PROBLEMA.....	33
4.1. PROBLEMAS DE DESARROLLO DEL SISTEMA QUE NOS COMPETE.....	33
4.1.1. <i>PROBLEMAS DEL USUARIO</i>	33
4.1.2. <i>INTERFAZ DE USUARIO</i>	35
4.1.3. <i>USABILIDAD</i>	37
4.1.4. <i>MULTIMODALIDAD</i>	39
4.2. APROXIMACIÓN DE LA SOLUCIÓN AL PROBLEMA.....	43
4.2.1. <i>DIRECTRICES PARA EL DISEÑO HACIA PERSONAS MAYORES</i>	43
5. ANÁLISIS Y DISEÑO DEL SISTEMA	45
5.1. ANÁLISIS DE REQUISITOS.....	45
5.1.1. <i>REQUISITOS FUNCIONALES</i>	46
5.1.2. <i>REQUISITOS NO FUNCIONALES</i>	50
5.2. DISEÑO	53
5.2.1. <i>ANÁLISIS DE LA DOCUMENTACIÓN</i>	54
5.2.2. <i>ARQUITECTURA DEL SISTEMA</i>	55
5.2.3. <i>RECOMENDACIONES DE DISEÑO DE DISPOSITIVOS MÓVILES</i>	57
5.2.4. <i>MODELO CONCEPTUAL</i>	59



6.	IMPLEMENTACIÓN	62
6.1.	HERRAMIENTAS DE DESARROLLO	62
6.1.1.	PLATAFORMA J2ME.....	63
6.1.2.	MIDP: INTERFACES GRÁFICAS DE USUARIO	66
6.1.3.	API MULTIMEDIA J2ME	68
6.2.	DETALLES DE IMPLEMENTACIÓN	70
7.	EVALUACIÓN.....	88
7.1.	TÉCNICA DE EVALUACIÓN EMPLEADA.....	88
7.2.	RESULTADOS	89
8.	ORGANIZACIÓN DEL PROYECTO.....	93
8.1.	PLANIFICACIÓN INICIAL	94
8.2.	PLANIFICACIÓN REAL.....	96
8.3.	PRESUPUESTO – ANÁLISIS ECONÓMICO	99
9.	CONCLUSIONES	102
10.	AMPLIACIONES Y LÍNEAS FUTURAS	104
11.	REFERENCIAS	106
12.	APÉNDICES.....	110
A.	MANUAL DE USUARIO	110
B.	PLANTILLA DE EVALUACIÓN	120
C.	INSTALACIÓN JAVA 2 SDK	122
D.	INSTALACIÓN NETBEANS 6.7.1	126



INDICE DE ILUSTRACIONES

Ilustración 1. Uso de TIC por edades (%).....	7
Ilustración 2. Dispositivos portátiles de tamaño pequeño	10
Ilustración 3. Tipos de dispositivos móviles	15
Ilustración 4. Modelo en cascada.....	32
Ilustración 5. Esquema conceptual de la Usabilidad.....	37
Ilustración 6. Componentes básicos de la interfaz multimodal	40
Ilustración 7. Arquitectura cliente/servidor.....	56
Ilustración 8. Arquitectura peer to peer	56
Ilustración 9. Diagrama UML de actividad	60
Ilustración 10. Arquitectura J2ME.....	64
Ilustración 11. Tipos de pantallas MIDP.....	67
Ilustración 12. Arquitectura API Multimedia	69
Ilustración 13. Diagrama de Clases	70
Ilustración 14. Tareas organizadas al principio del proyecto.....	94
Ilustración 15. Tareas organizadas al final del proyecto	96
Ilustración 16. Interfaz Modo Normal.....	111
Ilustración 17. Interfaz de Información.....	112
Ilustración 18. Interfaz de opciones	113
Ilustración 19. Interfaces gráficas de emergencias.....	114
Ilustración 20. Interfaz de recomendaciones.....	115
Ilustración 21. Interfaz del modo alerta.....	116
Ilustración 22. Interfaz de información de suceso	117
Ilustración 23. Interfaz de recomendaciones del modo alerta	119
Ilustración 24. Ejecutable J2SDK	122
Ilustración 25. Pantalla cuadro de licencia.....	123
Ilustración 26. Pantalla componentes J2SDK	123
Ilustración 27. Pantalla de progreso de instalación	124
Ilustración 28. Pantalla de selección de carpeta de instalación	124
Ilustración 29. Pantalla de instalación correcta	125
Ilustración 30. Pantalla de selección de paquete <i>Netbeans</i>	127
Ilustración 31. Icono para la instalación de <i>Netbeans</i>	127
Ilustración 32. Pantalla de bienvenida de instalación.....	128
Ilustración 33. Pantalla de selección de componentes.....	128
Ilustración 34. Pantalla de bienvenida con nuevos componentes.....	129
Ilustración 35. Pantalla de cuadro de licencia.....	129
Ilustración 36. Pantalla de selección de carpeta destino.....	130
Ilustración 37. Pantalla previa a la instalación	131
Ilustración 38. Pantalla de progreso de la instalación	131
Ilustración 39. Pantalla de instalación completa	132



INDICE DE TABLAS

Tabla 1. La pérdida de audición en términos de decibelios.....	23
Tabla 2. Nivel de decibelios y la intensidad de varias fuentes de sonido.	24
Tabla 3. Varios valores de frecuencia en relación con el oído humano.....	24
Tabla 4. Requisito de Sistema RF-01	46
Tabla 5. Requisito de Sistema RF-02	46
Tabla 6. Requisito de Sistema RF-03	47
Tabla 7. Requisito de Sistema RF-04	47
Tabla 8. Requisito de Sistema RF-05	47
Tabla 9. Requisito de Sistema RF-06	47
Tabla 10. Requisito de Sistema RF-07	47
Tabla 11. Requisito de Sistema RF-08	48
Tabla 12. Requisito de Sistema RF-09	48
Tabla 13. Requisito de Sistema RF-10	48
Tabla 14. Requisito de Sistema RF-11	48
Tabla 15. Requisito de Sistema RF-12	48
Tabla 16. Requisito de Sistema RF-13	48
Tabla 17. Requisito de Sistema RF-14	49
Tabla 18. Requisito de Sistema RF-15	49
Tabla 19. Requisito de Sistema RF-16	49
Tabla 20. Requisito de Sistema RF-17	49
Tabla 21. Requisito de Sistema RF-18	49
Tabla 22. Requisito de Sistema RF-19	49
Tabla 23. Requisito de Sistema RF-20	50
Tabla 24. Requisito de Sistema RF-21	50
Tabla 25. Requisito de Sistema RF-22	50
Tabla 26. Requisito de Sistema RNF-01.....	50
Tabla 27. Requisito de Sistema RNF-02.....	50
Tabla 28. Requisito de Sistema RNF-03.....	51
Tabla 29. Requisito de Sistema RNF-04.....	51
Tabla 30. Requisito de Sistema RNF-05.....	51
Tabla 31. Requisito de Sistema RNF-06.....	51
Tabla 32. Requisito de Sistema RNF-07.....	51
Tabla 33. Requisito de Sistema RNF-08.....	51
Tabla 34. Requisito de Sistema RNF-09.....	52
Tabla 35. Requisito de Sistema RNF-10.....	52
Tabla 36. Requisito de Sistema RNF-11.....	52
Tabla 37. Requisito de Sistema RNF-12.....	52
Tabla 38. Requisito de Sistema RNF-13.....	52
Tabla 39. Requisito de Sistema RNF-14.....	52
Tabla 40. Resultados de la evaluación	90



1. INTRODUCCIÓN

Este Proyecto Fin de Carrera de la titulación Ingeniería Informática tiene como objetivo desarrollar un sistema web multidispositivo para un dispositivo móvil. En este proyecto se llevará a cabo el análisis, diseño e implementación de una interfaz de usuario para la notificación de eventos en situaciones de emergencia.

La mayor complejidad que presenta este trabajo es llegar a crear un sistema que sea compatible con dos aspectos en donde actualmente no hay una relación estrechamente ligada, las personas mayores y el uso de las actuales tecnologías de información.

La motivación de este proyecto conjunta varios aspectos que hizo que su realización fuese satisfactoria e interesante. Por un lado con la realización de este proyecto es posible adentrarse en el conocimiento de una plataforma de desarrollo, como es el diseño y desarrollo de aplicaciones para dispositivos móviles, actualmente muy valorada y de potencial crecimiento en el mercado. Además se tiene la motivación de desarrollar una aplicación en los dispositivos móviles para la notificación de eventos ocurridos debido a una situación de emergencia y por último, hacer que esta herramienta sea útil y fácilmente manejable para personas mayores.

Como se puede observar en la ilustración 1, se muestra una gráfica obtenida del INE¹ donde se refleja un menor porcentaje del uso de las (TIC) Tecnologías de la Información y Comunicación, a medida que se aumenta con la edad. Por esta razón este proyecto ha intentado desarrollar un sistema de información de fácil manejo y aprendizaje para los usuarios y que reúna todas las necesidades y requerimientos de éstos para la notificación de emergencias a personas de la tercera edad principalmente, aunque hay que tener en cuenta que este sistema también es válido para el uso de la población en general.

Además, durante en desarrollo del proyecto se ha puesto especial interés en crear una interfaz de usuario que trate de resolver ciertas características o grados de discapacidades y deficiencias que presenta el sector de usuarios de la población anciana, así como las pérdidas de los distintos grados del funcionamiento cognitivo (perdida de la sensibilidad, de la visión,..etc.).

¹ Instituto Nacional de Estadística. <<http://www.ine.es/>>

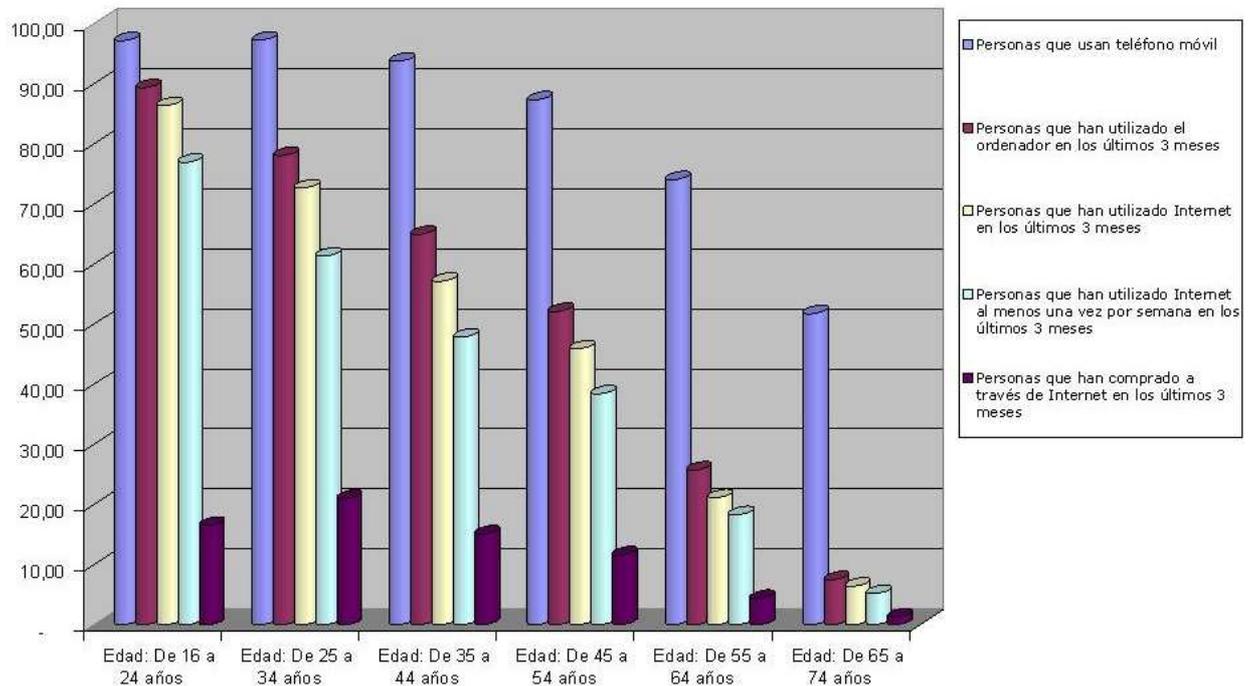


Ilustración 1. Uso de TIC por edades (%)

1.1.BREVE DESCRIPCIÓN DEL TRABAJO

En primer lugar se realiza un estudio completo de la computación ubicua en la actualidad, como ha sido la evolución tecnológica desde los primeros grandes sistemas informáticos concebidos para dar servicio a multitud de usuarios, hasta los dispositivos portátiles que se encuentra hoy día en el mercado que tienen un tamaño más reducido y son también mucho más potentes. Se analiza también en este proyecto el estado de las personas mayores frente a las nuevas tecnologías de información con el objetivo de adaptarlas al perfil de usuario del sistema a desarrollar, así como los problemas que presentan este sector de usuario en sus habilidades cognitivas que pueden conllevar una disminución de la capacidad funcional, con lo que puede afectar al funcionamiento de alguno de sus sentidos.

Puesto que este proyecto está concebido al desarrollo de una aplicación para dispositivos móviles, es necesario realizar un estudio acerca de los principios de diseño de aplicaciones de este tipo de dispositivos y las plataformas de desarrollo que existen actualmente para poder implementar estas aplicaciones.



Una vez que sean estudiados todos los temas anteriormente expuestos, es el momento de enfrentarse con los problemas de desarrollo que existen en la actualidad en este tipo de sistemas de información. Se exponen los problemas encontrados en la actualidad para poder realizar una aplicación web multidispositivo para la notificación de eventos para las personas de la tercera edad correctamente. Además se debe detallar las soluciones llevadas a cabo para solventar estos problemas de desarrollo encontrados.

Finalizado este apartado, se entra en uno de los procesos críticos del desarrollo, el análisis y la especificación de requisitos. Se deben establecer los requisitos que debe cumplir el software del sistema, tanto los requisitos funcionales como no funcionales, es necesario que el conjunto de requisitos sea tan completo, consistente y correcto como sea posible, y así satisfaga a todos los interesados del sistema.

Establecidos todos los requisitos, se iniciará el proceso de diseño del sistema. En este proceso se debe diseñar tanto los aspectos gráficos de la aplicación cómo el esquema de funcionamiento de la misma, para ello emplearemos un modelo conceptual, donde se especificará en detalle el comportamiento que debe tomar el usuario entre el inicio y el final de la ejecución, además de definir la lógica de todas las operaciones del sistema.

Una vez terminada la fase de diseño, se entra en la implementación del sistema, tomando como referencia el diseño del sistema realizado con antelación. Finalizado esta fase de implementación del sistema, se procede a la fase de evaluación del sistema, en la que se debe comprobar si los potenciales usuarios se sienten cómodos al manejar la aplicación y deciden, si está es fácil de usar.

Posteriormente se realiza el estudio acerca de la organización del proyecto, donde se hará un análisis de la planificación realizada al comienzo del proyecto y la planificación real que se obtendrá una vez finalizado el proyecto. Se realiza el presupuesto final del sistema con el objetivo de obtener un mayor realismo en este proyecto

Por último se realizan las pertinentes conclusiones obtenidas, tras el desarrollo y documentación del presente proyecto. Además se exponen las futuras mejoras o ampliaciones que tendría el sistema en caso de continuar con él, una vez presentado el proyecto.



2. ESTADO DEL ARTE

2.1.MARCO HISTÓRICO

La miniaturización de microprocesadores ha supuesto una evolución tecnológica permitiendo abrir nuevas posibilidades de servicios tecnológicos al usuario, a través del empleo y manipulación de información presente en su entorno. Estos avances de la tecnología, han logrado que lo que hace pocos años pareciera ciencia ficción, hoy sea realidad.

Todo esto ha supuesto que se haya entrado en una nueva era o un nuevo avance en las investigaciones en el campo de la computación y de los sistemas de información. Los primeros servicios tecnológicos que encontramos a lo largo del tiempo fueron los grandes sistemas informáticos que fueron concebidos para dar servicio a multitud de usuarios, con un alto costo y una gran complejidad de uso.

En los años 90 nos encontramos con la comercialización de la computación. Fue en esta época en la que más fuerte se empezó a emplear el término hombre-máquina. Esta comercialización trajo consigo un aumento en el empleo de los ordenadores personales de bajo coste. El diseño de estos ordenadores personales estaba destinado hacia el usuario, por lo que su manejo cada vez era mucho más sencillo.

En la actualidad nos encontramos en lo que se denomina “nueva era de la computación”, la era de la computación ubicua². Mediante esta nueva rama de computación, también llamada computación penetrante³, una persona tendrá a su disposición un gran número de dispositivos de computación que no tendrá que manejar, estos trabajarán independientes del usuario, de manera autónoma.

Durante los últimos quince años ha nacido un renovado y creciente interés por el desarrollo y comercialización de aplicaciones ubicuas. En este tipo de aplicaciones se asume un modelo de interacción hombre-máquina diferente al que una persona está acostumbrada en los tiempos actuales, debido en gran manera a la cantidad de dispositivos que tiene a su alcance el usuario.

² tr. *Ubiquitous Computing*

³ tr. *Pervasive Computing*

En su momento, fueron muy pocos los que podían imaginar las posibilidades de semejante avance, siendo posiblemente Mark Weiser quién haya podido tener más clara esta visión sobre la computación ubicua.

Mark Weiser en sus publicaciones [1] y [2], nombraba como la tercera ola o tercera era en la computación a este cambio de paradigma que estaba empezando a surgir. En la primera ola, multitud de individuos compartían un sistema de cómputo de forma análoga al estilo de un mainframe. Posteriormente, con la aparición de ordenadores personales, cada individuo tuvo acceso a su propio computador de escritorio.

Al analizar los numerosos avances de la tecnología, esto no sólo no parece algo fuera del alcance de la imaginación, sino que se trata de una predicción bastante acertada, que pueda llegar a suceder, durante los próximos 10 o 15 años. Hoy se cuenta con una serie de dispositivos portátiles (*handhelds*, *TabletPCs*, *smartphones*, *notebooks*, etc) que se vuelven cada vez más pequeños y mucho más potentes.



Ilustración 2. Dispositivos portátiles de tamaño pequeño

Asimismo, los aparatos realizados mediante el empleo de sensores que alguna vez estuvieron destinados a procesos industriales, empiezan a integrarse en los hogares que habitamos, al mismo tiempo que empiezan a desarrollarse plataformas software en lenguajes de alto nivel para que puedan ser controlados. De igual manera se está trabajando constantemente en lograr que las tecnologías de diseño de sensores superen a las actuales, logrando para ello un aumento en el rendimiento en términos del consumo



de energía y reduciendo el tamaño de los sensores. Con esto se entiende que la tecnología ya está a la altura de las circunstancias actuales, pero aún así, parece que falta una forma de incorporar todos estos dispositivos a nuestra vida diaria.

La tecnología debe convertirse en una herramienta invisible para el usuario, en este sentido debe asistir al usuario y ayudarlo a lograr un mejor estilo de vida, pero bajo ningún concepto debe convertirse en un obstáculo ni en un factor de riesgo para él (el hecho de tener un conjunto dispositivos controlando constantemente las actividades de las personas trae consigo diversos problemas de privacidad de datos). En palabras de Weiser:

"Las tecnologías más profundas son las que desaparecen. Ellas se tejen en la misma tela de la vida cotidiana hasta que son indistinguibles de ella."

Dentro del área de la computación ubicua han surgido diversas ramas, como la mencionada anteriormente computación pervasiva, hogares digitales, aplicaciones sensibles al contexto y personalización de aplicaciones. A pesar de que cada una de estas ramas tiene un objetivo en común, cada una enfoca el problema desde un punto de vista distinto.

Por último M. Weiser [1] definió la computación ubicua mediante cuatro conceptos fundamentales:

- **Empleo eficaz de espacios omnipresentes.** Se basa en la detección del estado de un individuo y de sus necesidades, ya sea en la oficina, clase, domicilio, coche, etc. El espacio omnipresente surge cuando varios de estos dispositivos coinciden en el mismo espacio físico e interactúan colaborativamente para dar soporte a los individuos que se encuentren en él. La domótica, computación ubicua en el domicilio, es la aplicación más popular.
- **Invisibilidad.** Se basa en la completa desaparición de la tecnología de la consciencia del usuario. La invisibilidad va a requerir un cambio drástico en el tipo de interfaces que nos comunican con los ordenadores.



- **Ocultación del acondicionamiento.** La distribución de los servicios ofrecidos puede ser muy poco uniforme, en esta situación el principio de invisibilidad puede no cumplirse ya que el usuario detectaría desagradables transiciones, todo esto puede depender de desarrollo tecnológico disponible, así como su infraestructura. Este requisito es hoy día el más alejado en relación a la situación ideal, los sistemas que incorporan computación ubicua están aislados, sin continuidad entre unos y otros.
- **Localidad.** El concepto de localidad de servicios en computación ubicua es fundamental frente a la universalidad de servicios de Internet. A priori, es natural pensar en la computación ubicua como un traspaso de la filosofía de Internet y cualquier servicio de computación, en cualquier sitio. Nada más alejado de la realidad. Los usuarios disponen de capacidades asociadas al contexto en el que se encuentran.

Los desafíos que se pretende conseguir con la computación ubicua se resumen en:

- **Escalabilidad:** la idea es que soporten cientos, miles e incluso millones de máquinas conectadas a través de internet
- **Facilidad de Uso:** debe ser fácil agregar una nueva máquina al conjunto de dispositivos. Además se debe tener facilidad para el desarrollo de aplicaciones.
- **Tiempos de Ejecución:** Es necesario el empleo de mecanismos para asegurar que se pierda la mínima cantidad de trabajo en caso de fallos.
- **Paralelismo:** el conjunto de máquinas crece y decrece durante la ejecución de una aplicación. El sistema debe ser capaz de reasignar trabajos entre diferentes procesos.
- **Seguridad:** No existe un administrador único para manejar todos los nodos participantes. Los clientes necesitan garantías de que no se permitirán accesos no autorizados a sus máquinas.
- **Heterogeneidad:** involucra múltiples arquitecturas.



- Disponibilidad y alto rendimiento

Multitud aplicaciones hoy en día soportadas a través de Internet se enriquecen y amplían sus servicios: *e-business*, *e-commerce*, *e-government*, *e-banking*, *e-education*. El aumento de la potencia de las actuales agendas digitales personales permitirá a un individuo, que éstas se conviertan en su asistente personal. Este avance tecnológico permitirá que las mencionadas agendas personales manejadas mediante la voz, no sólo gestionarán nuestra agenda, si no que se convertirán, gracias a la gran cantidad de sensores que disponen (biométricos, de localización, etc.), en vigilantes de nuestra salud, consejeros financieros, localizadores, guías turísticos.⁴

El otro gran campo de aplicación de la computación ubicua son los espacios inteligentes. Estos espacios están hoy en nuestras viviendas, salas de reuniones, consultas médicas, centros de educación y entrenamiento, centros de mando militar, gabinetes de crisis, todo tipo de vehículos, etc.

Por otro lado y teniendo en cuenta estos avances tecnológicos a nivel de computación ubicua y uso de dispositivos móviles, que son cada vez más utilizados por el segmento poblacional de la tercera edad, Sainz [3] aboga por la inclusión de las posibles víctimas en caso de emergencias en cuanto la población puede ser ayudada a través de la provisión de información por parte de las autoridades y organismos encargados de responder a la emergencia. Se trataría del principio de:

“Ayudarse por ser ayudado (help us to help you)”.

Por último hay que mencionar el surgimiento de nuevas posibilidades tecnológicas como consecuencia de la interconexión de un número elevado de dispositivos con capacidad de computación, como por ejemplo la computación en *grid*. Esta nueva tecnología consiste en la utilización de múltiples recursos para acometer tareas computacionalmente intensas, durante los tiempos de inactividad de equipos conectados en una red.

Ejemplos de aplicaciones en la vida cotidiana de la computación ubicua son los siguientes:

⁴ NOVATICA. Número especial acerca de la computación ubicua. Octubre 2001.



- APLICACIONES GENERALES
 - ✓ Sistemas de información públicos: guías personalizadas, información adaptada a discapacitados
 - ✓ Hogares inteligentes
 - ✓ Sistemas de señalización para entornos urbanos
 - ✓ Infraestructuras inteligentes: seguridad en carretera
 - ✓ Supervisión de hábitats naturales

- APLICACIONES EN EL AMBITO DE LA SALUD
 - ✓ Asistencia a discapacitados:
 - Adecuación del entorno
 - Señalización y guiado
 - Adaptación de interfaces
 - ✓ Supervisión y cuidado remoto
 - Personas dependientes
 - Enfermos crónicos
 - ✓ Diagnóstico precoz
 - ✓ Hospitales inteligentes

Una vez presentado el marco histórico de la computación ubicua y detallados sus desafíos en el futuro y aplicaciones en la vida actual, se ha descubierto que esta tecnología comienza a ser muy importante en el mercado actual y son numerosas las ventajas que presentan a la sociedad y aportaciones que tiene esta tecnología en el mundo de internet.

En el siguiente apartado, se detallará los principios de diseño en el desarrollo de aplicaciones en dispositivos móviles. Puesto que este proyecto está dirigido a la implementación de una aplicación en un dispositivo móvil, se debe investigar y estudiar las pautas y directivas para el desarrollo de aplicaciones móviles, así como conocer las limitaciones que presenta la implementación en estos dispositivos.

2.2. PRICIPIOS DE DISEÑO PARA DISPOSITIVOS MÓVILES

Hoy día, los dispositivos móviles son una parte importante de la sociedad, cada vez más personas lo utilizan tanto en el trabajo como en el ocio. El diseño de dispositivos móviles se ha convertido en una pieza importante en la actualidad tecnológica.

La generalización en los últimos años de teléfonos móviles, *Smartphone*, PDAs, etc. ha generado una importante necesidad de aplicaciones para este tipo de dispositivos. Las prestaciones de los dispositivos móviles se incrementan día a día, posibilitando la implementación de aplicaciones muy interesantes:

- Acceso a Internet y conectividad mediante Bluetooth
- Envío de mensajes cortos
- Aceleración 2D y 3D
- Reproducción de video y sonido
- Cámara integrada
- Información de localización GPS
- Sensores de orientación



Ilustración 3. Tipos de dispositivos móviles



2.2.1. DISEÑO DE DISPOSITIVOS MÓVILES EN APLICACIONES WEB

El diseño para dispositivos móviles está en crecimiento. Existen multitud de aplicaciones desarrolladas para dispositivos móviles: videojuegos, aplicaciones GPS, agendas electrónicas altamente sofisticadas, software para control de stock...etc. Pero lo que realmente está marcando un nuevo rumbo en este sector tecnológico, está siendo el desarrollo de aplicaciones para la industria del diseño de sitios web. Multitud de compañías buscan desarrolladores que puedan convertir su actual sitio web, en un sitio “móvil compatible”, o la creación de una copia del mismo para ser utilizado exclusivamente por los navegadores de interfaz móvil.

Las aplicaciones Web móviles actuales deben poseer los siguientes principios básicos para realizar un diseño correcto para dispositivos de acceso móvil⁵:

- Si se trata de un servicio Web desarrollado en XHTML, se debe utilizar HTML puro sin estilo. Con el empleo de HTML puro la aplicación Web móvil es fácilmente visible por diversos dispositivos móviles, además el tiempo de carga de las páginas es pequeño, al menos en comparación con las hojas de estilo, que por otro lado, no son fácilmente visibles para todos los navegadores. Por el contrario el hecho de usar HTML puro conlleva a un tamaño de archivo elevado y además se pueden producir problemas de desplazamiento y *scroll* debido a la gran cantidad de contenido.
- Empleo de hojas de estilo para dispositivos de mano. Las hojas de estilo para dispositivos móviles tienen la ventaja de que el desarrollador web sólo tendrá que hacer frente a una nueva hoja de estilos. El gran problema de su uso es que CSS es muy dispar en las diferentes interfaces móviles.
- Se debe desarrollar exclusivamente para dispositivos móviles. En las aplicaciones para este tipo de dispositivos, es obligatorio que las tanto las imágenes como el contenido y los códigos están específicamente diseñados para los mismos.

⁵ < <http://blogemarketer.wordpress.com/2008/07/07/disenio-de-interfaces-para-dispositivos-moviles> >



Cuando se emplea el término de Web Móvil, lo que se pretende es hacer una referencia a una Web en la que un usuario puede acceder a la información desde cualquier ubicación, independientemente del tipo de dispositivo que utilice para ello.

En la mayoría de la ocasiones, la experiencia del usuario es poco satisfactoria al encontrarse con numerosos problemas para acceder a la Web desde los dispositivos móviles.

El W3C (*World Wide Web Consortium*), ha puesto en marcha la "Iniciativa de Web Móvil"⁶. Con esta idea se pretende conseguir el objetivo de convertir el acceso a la Web desde un dispositivo móvil en algo tan sencillo y cómodo como lo es desde los equipos de escritorio. Se busca resolver los problemas de usabilidad y interoperabilidad que actualmente dificultan el acceso a la Web desde un dispositivo móvil y hacer así posible uno de los objetivos principales de la organización que consiste en alcanzar una Web única.

Es de gran importancia en este tipo de aplicaciones, y por tanto habrá que tenerlo muy en cuenta, el crear programas separados por cada uno de los posibles usos que se le dé a la aplicación. Haciéndolo de esta forma, se conseguirá que cada programa sea más pequeño y se adapte mucho mejor a las características de los dispositivos móviles.

Por otro lado, en relación con el diseño de la interfaz de usuario, la primera decisión que se debe tomar es la correspondencia entre la pantalla y la aplicación. El diseño de la interfaz no se debe realizar pensando cómo un usuario operará con un dispositivo móvil para llevar a cabo una determinada tarea. El diseño debe concentrarse únicamente en el objetivo asignado a la pantalla y en la tarea que permitirá llevar a cabo ese objetivo.

Una solución sencilla y eficaz para no caer en este error, recomendada por el consorcio *Sun Microsystems*⁷, es el empleo de *story boards*. Lo ideal sería que cada viñeta únicamente contuviera los requerimientos correspondientes para la pantalla.

⁶ <<http://www.w3c.es/Divulgacion/GuiasBreves/webmovil>>

⁷ <<http://www.sun.com/>>



Otra consideración en cuanto al diseño de dispositivos móviles es el almacenamiento persistente. Si el dispositivo móvil posee un sistema de ficheros, lo más recomendable sería optar por la creación de ficheros que tuviesen una estructura más o menos compleja y acceder a ellos mediante el empleo de bibliotecas de Java.

Otra alternativa para el almacenamiento persistente puede ser el empleo sistemas de gestión de bases de datos relacionales. Esta solución no es la más recomendable, y solo se emplearía en los siguientes casos:

- Tener que almacenar un gran volumen de datos
- Realizar una gran cantidad de accesos.

Debido a que en este proyecto no es necesario un almacenamiento de un elevado volumen de datos, se ha optado por emplear la primera alternativa y emplear directivas Java para el acceso a los ficheros. Por último, se debe tener en cuenta dentro del diseño de aplicaciones para dispositivos móviles, diversos aspectos relacionados con la conectividad y con la E/S (entrada / salida), ya que estos puntos son muy importantes y determinarán la portabilidad de la aplicación en otras plataformas.

2.2.2. LIMITACIONES DE LOS DISPOSITIVOS MÓVILES

El diseño de los dispositivos móviles presenta una serie de limitaciones en cuanto al desarrollo de aplicaciones móviles. Los problemas relacionados con el desarrollo de una aplicación móvil se pueden enumerar de la siguiente manera:

- Diferente soporte: Los distintos S.O. y arquitecturas admitidas por los dispositivos móviles ofrecen un soporte diferente. El problema que supone esto, es que una aplicación móvil deberá ser desarrollada teniendo en cuenta el soporte correspondiente.
- Diferentes formularios: Los distintos tamaños de pantalla, su orientación y los tipos de pantalla (color, escala de grises, blanco y negro) son variables que afectan a la paginación del contenido y el tipo de gráficos que se podrían visualizar.



- Diferentes capacidades de los dispositivos móviles: Memoria, velocidad de procesamiento, tiempo de respuesta, servicios ofrecidos...
- La necesidad de ahorro de energía obliga a emplear procesadores con una baja frecuencia de reloj y capacidad de cómputo limitada.
- El uso intensivo de gráficos, sonidos, cálculos, conexiones inalámbricas, es decir, la multitud de servicios que ofrecen los dispositivos móviles, limita la vida de la batería.
- La memoria disponible comparada con de un ordenador ordinario es muy limitada. Además el almacenamiento de datos persistente también es muy limitado: memoria interna + tarjetas SD o similar.

Por otro lado el diseño de interfaces para teléfonos móviles suele ser difícil debido a las siguientes causas:

- El gran número y variedad de dispositivos móviles en el mercado actual.
- La amplia gama de navegadores que utilizan en sus aplicaciones.
- Los límites de entrada y salida.
- La falta de directrices y recursos en materia de diseño.
- La falta de normas.

Por último existen también limitaciones dentro de los navegadores de Internet móvil, como son las dificultades de navegación y el tiempo de carga de todos los elementos del sitio web. Estos inconvenientes unidos al hecho de que el equivalente al ratón de un ordenador es un dedo de la mano para estos dispositivos, se traduce en dificultades para el diseño de enlaces. Otro problema de los enlaces se produce en su tamaño, en si estos enlaces son demasiado pequeños o demasiado grandes. Este último punto, hace difícil al desarrollador determinar qué tamaño de fuente es el óptimo para hacer los vínculos. Se ha de tratar de buscar una solución con respecto al tamaño, que satisfaga a todos los usuarios.



2.3.DISEÑO PARA MAYORES: PRINCIPIOS Y RECOMENDACIONES

En estos días la población de la tercera edad está convirtiéndose en un gran segmento de la población. Este grupo de la población está aumentando rápidamente en todo el mundo desarrollado.⁸ Algunos datos de las Naciones Unidas muestran que esta tendencia va a continuar en los próximos siglos. Esta tendencia es particularmente fuerte en Europa:

*"Durante el próximo cuarto de siglo se prevé que Europa mantenga su título de «más antigua» en la región del mundo. Actualmente las personas mayores representan alrededor del 20% de la población total y se prevé que represente el 25% de aquí a 2020."*⁹

En muchos países, el sector de las personas de edad muy avanzada (los mayores de 75 años) es de los que más rápido crecen en la población actual. Sin embargo, como ha sido previsto por la OMS [4] no se trata sólo de que Europa está experimentando una transición demográfica que va de la alta mortalidad / alta fecundidad a un modelo de baja tasa de fecundidad / baja mortalidad, sino que se trata de un patrón de transición mundial.

2.3.1. EFECTOS DEL ENVEJECIMIENTO

El proceso de envejecimiento está generalmente relacionado con una disminución de la funcionalidad y la pérdida sensorial, así como con una desaceleración cognitiva. Estas características que definen a este segmento de la población, son también la causa de las deficiencias que podemos experimentar en cualquier momento durante el curso de la vida, y que son inesperadas, a priori, como fue indicado por Edwards en 2001¹⁰:

"Todos tenemos diferentes capacidades y debilidades, todos podemos ser discapacitados por nuestro medio ambiente y estadísticamente todos tendremos algún momento en nuestras vidas (como nosotros tendremos una vida larga) que nos convertiremos en discapacitados de alguna manera."

⁸ La esperanza de vida en el Reino Unido y en la Unión Europea ha aumentado considerablemente desde 1980, con un aumento de casi 5 años para los hombres y mujeres (Eurostat).

⁹ Estadísticas 2002

¹⁰ Edwards, A. "Enabling technology for users with special needs". Proceedings of the IHM-HCI, Lille.



Al mismo tiempo, la población de personas mayores es un grupo muy heterogéneo con una amplia variedad de características, esta heterogeneidad hace que sea difícil generalizarla.

2.3.2. PÉRDIDA SENSORIAL

Uno de los efectos más evidentes del envejecimiento es la pérdida sensorial [5]. Algunas habilidades de las personas mayores tienden a permanecer intactas o ligeramente afectadas, mientras que otras sufren una gran cantidad de pérdida aunque también depende en gran medida de la persona. La disminución de la capacidad funcional pueden afectar el funcionamiento de:

- Los órganos de los sentidos (vista, oído, olfato, sensación táctil, sabor).
- La capacidad de procesamiento de la información, la inteligibilidad de la palabra.
- Reducción de la velocidad y el aumento de la varianza en el momento preciso de los movimientos.
- Aumento del tiempo necesario para recuperar la información de la memoria.

2.3.3. PÉRDIDA DE LA VISIÓN

Los cambios en la percepción visual, (referido a los cambios en la agudeza visual, el contraste de la agudeza, la profundidad de percepción, la agudeza visual) son también notables desde principios de la edad adulta según lo expuesto por el proyecto INCLUDE [6]. La capacidad de la agudeza disminuye rápidamente entre los 50 a los 90 años de edad. La percepción de los colores y la capacidad de ver el atardecer también disminuye con la edad. Lo mismo sucede con la disminución en la adaptación para ver las distancias cercanas, un problema que comienza a veces tan pronto como a la edad de 20 años, así como la adaptación de la visión para ver en condiciones de poca luz. Las personas mayores tienen una disminución de la sensibilidad con el contraste espacial a altas frecuencias. La sensibilidad del contraste de las personas mayores también es de 1,5 a 4 veces inferior que una persona joven. En las personas de 30 a 80 años, la percepción de la profundidad también se va convirtiendo en más pobre, con un aumento en el umbral de 100 a 300 segundos por arco [6].



Como ejemplo de la amplia disminución de la agudeza visual, el envejecimiento en los seres humanos hace necesario el empleo de lentes correctoras, que se convierten en casi universalmente necesarias para una disponer de una buena agudeza visual a una distancia óptica inferior a 0,5 metros en torno a la edad de 40 años. Una vez más, esto apoya la idea de que el diseño para las personas mayores no sólo debe satisfacer sus necesidades sino también las de una amplia gama de la población, que sufren cualquier tipo de dificultad en el uso de la tecnología debido a las deficiencias (tanto naturales como circunstancialmente impuestas).

Algunas capacidades visuales tienden a disminuir con la edad, ya sea debido a los efectos del envejecimiento o la aparición de enfermedades más frecuentes con el avance de la edad. Cataratas, glaucoma, degeneración macular y la retinopatía diabética son otras causas que se desarrollan con la edad. También se sabe que la visión de las personas mayores es mucho menos sensible sobre los objetos de grano fino con superficies y objetos en movimiento, a la visión de las personas jóvenes.

2.3.4. PÉRDIDA DE LA AUDICIÓN

El oído no es una excepción a la progresiva degeneración de los tejidos del cuerpo que viene unido con el envejecimiento. El grado de pérdida auditiva, aunque no es un proceso estándar y hay una considerable diferencia en la pérdida de la capacidad de una persona a otra, puede darse el caso de completar la sordera total, en la que no se oye ningún sonido o el caso de tener una ligera dificultad para seguir una conversación en una situación de ruido externo.

El sentido de la audición se ve afectado de diversas maneras durante el proceso de envejecimiento, desde la disminución de la capacidad para localizar fuentes de sonido, así como sordera y dureza de la audición que pueden llevar a la pérdida total de esta sensibilidad. Algunos de estos procesos, como la presbiacusia (pérdida de sensibilidad) pueden comenzar tan temprano como a los 25 años de edad.

Los problemas de audición son familiares para todos nosotros cuando se trata de las personas mayores. Fozard en 1993 [7] informó que los umbrales de tonos puros en las frecuencias altas aumentan regularmente a un ritmo de alrededor de 0,3 dB de un año a



otro a través de la edad de 60 años, pero de los 80 a los 95 años de edad aumenta rápidamente, a 1,4 dB por año.

Para dar una idea clara del efecto de dicho déficit a largo plazo, la Tabla 1 muestra los niveles de pérdida auditiva en términos de decibelios.

Umbral Auditivo	Grado de pérdida auditiva	Capacidad para escuchar
0–25 dB	Ninguna	Dificultad no significativa
26–40 dB	Leve	Dificultad leve para escuchar a distancia
41–55 dB	Moderada	Dificultad para seguir una conversación
56–70 dB	Moderada a grave	El discurso debe ser alto. Dificultad para entablar una conversación de grupo
71–90 dB	Grave	El discurso debe ser muy alto. Solo puede seguir una conversación si están gritando o hablando muy fuerte
91+ dB	Profunda	No puede entender un discurso con un tono fuerte.

Tabla 1. La pérdida de audición en términos de decibelios

Los seres humanos escuchan en una amplia gama de frecuencias que oscilan entre unos 20 Hz a 20.000 Hz. El tono es nuestra percepción de la tasa o de la frecuencia de las vibraciones, que componen las ondas sonoras. Las vibraciones más rápidas crean tonos más altos y las vibraciones más lentas crean tonos inferiores.

Sin embargo, también existen algunas diferencias en la forma en que afecta el tiempo de decaimiento en la capacidad de nuestra audición, por ejemplo para frecuencias superiores a 1000 Hz, la pérdida de audición es aproximadamente la misma para hombres y mujeres de la misma edad, pero para frecuencias por debajo de este punto, la pérdida de la audición en los hombres es, en general, mayor que la que sufren las mujeres. Por otra parte, en las frecuencias por debajo de 1000 Hz, la pérdida de la audición en general es mayor para las mujeres que para los hombres. Los hombres mayores tienen peor audición en altas frecuencias que las mujeres mayores. Esto se debe muy probablemente a causa de los riesgos profesionales y la exposición al ruido. La Tabla 2 muestra los niveles de presión acústica que nos encontramos en situaciones cotidianas.



Origen	Nivel de intensidad	# Horas Mayor que TOH *	Intensidad
Umbral de audición (TOH)	0 dB	10^0	$1 \cdot 10^{-12} \text{ W/m}^2$
Respiración normal	10 dB	10^1	$1 \cdot 10^{-11} \text{ W/m}^2$
Susurro	20 dB	10^2	$1 \cdot 10^{-10} \text{ W/m}^2$
Música muy suave	30 dB	10^3	$1 \cdot 10^{-9} \text{ W/m}^2$
Nevera tarareando	40 dB	10^4	$1 \cdot 10^{-8} \text{ W/m}^2$
Conversación tranquila	50 dB	10^5	$1 \cdot 10^{-7} \text{ W/m}^2$
Conversación normal	60 dB	10^6	$1 \cdot 10^{-6} \text{ W/m}^2$
Calle con tráfico	70 dB	10^7	$1 \cdot 10^{-5} \text{ W/m}^2$
Aspirador	80 dB	10^8	$1 \cdot 10^{-4} \text{ W/m}^2$
Gritos, cortadora de césped	90 dB	10^9	$1 \cdot 10^{-3} \text{ W/m}^2$
Gran orquesta	98 dB	$10^{9.8}$	$6.3 \cdot 10^{-3} \text{ W/m}^2$
Nivel Máximo en el walkman, motos de nieve	100 dB	10^{10}	$1 \cdot 10^{-2} \text{ W/m}^2$
Primeras filas de concierto de rock	110 dB	10^{11}	$1 \cdot 10^{-1} \text{ W/m}^2$
Hélice del avión despegando	120 dB	10^{12}	$1 \cdot 10 \text{ W/m}^2$
Umbral del dolor	130 dB	10^{13}	$1 \cdot 10^1 \text{ W/m}^2$
Avión militar despegando	140 dB	10^{14}	$1 \cdot 10^2 \text{ W/m}^2$
Instantánea de la perforación de tímpano	160 dB	10^{16}	$1 \cdot 10^4 \text{ W/m}^2$

Tabla 2. Nivel de decibelios y la intensidad de varias fuentes de sonido. * Umbral de audición

La tendencia general es que los altos tonos son oídos gradualmente peor que los bajos, así como las diferencias sexuales como se comentó antes.

Por ejemplo, las notas de un tambor, una profunda voz masculina o los sonidos vocales son los sonidos de tono más bajo, mientras que las notas en una flauta, la voz de los niños y algunos sonidos consonantes tales como 'f', 's', 'sh', son tonos agudos. La Tabla 3 muestra el valor en Hz típico de algunos sonidos. Aunque el tono se relaciona directamente con la frecuencia esto no es lo mismo, ya que la frecuencia es un fenómeno físico, mientras que el tono es una percepción (psicoacústicas, cognitivas o psicofísica).

Origen	Frecuencia baja (Hz)	Frecuencia alta (Hz)
Piano	27.5	4,186
Voz femenina	140	500
Voz masculina	80	240
Compact Disc	0	22,050
Oído humano	20	20,000

Tabla 3. Varios valores de frecuencia en relación con el oído humano.



Lysons [8] apunta en la misma dirección en cuanto a la importancia de la exposición al ruido como detonante para la presbiacusia, ya que es más pronunciada en algunas personas que otros.

"Una primitiva tribu africana que vive en un entorno en el que los niveles de ruido sólo de vez en cuando superó 40dB, indicó que un hombre entre 70-79 años de edad tenía un oído más penetrante que un estadounidense de una edad comprendida entre 30-39 años de edad que había sido expuesto a los ruidos de la civilización moderna".

A medida que nuestra sociedad comienza a estar por encima de los niveles de ruido que nos rodea y en la mayor parte del tiempo del día, la presbiacusia parece ser inevitable para la gran mayoría de nuestra población. Lysons [8] apunta también a las personas de avanzada edad relacionados con la pérdida:

"Se ha estimado que aproximadamente el 25 por ciento de las personas de 65 años de edad tienen más dificultad para oír que cuando tenía 30 años de edad. A 70 y 80 años el porcentaje aumenta al 33,3 y 50 por ciento, respectivamente. "

2.3.5. LA PERCEPCIÓN TÁCTIL

La disminución de la percepción sensorial es también notable en el sentido del tacto. Si hay una patología asociada, el problema es agravado notablemente como Sathian [9], demostró en pacientes con enfermedad de Parkinson, los cuales mostraron un aumento doble del umbral táctil espacial.

Asimismo, estos pacientes fueron perjudicados en su capacidad para discriminar el rallado de la rugosidad, con diferencias de más de tres veces superior al de un grupo de personas mayores sin la enfermedad de Parkinson. En las personas mayores también ha sido probado que sus facultades se ven reducidas para discriminar diferencias táctil, la orientación de las líneas y la longitud de las líneas.



2.3.6. DETERIORO COGNITIVO

En la parte superior del deterioro sensorial, también hay ciertos problemas cognitivos adquiridos con el envejecimiento. Estas deficiencias tienen efectos en el tratamiento de la información, tales como tiempo de respuesta lento, la velocidad en que se procesa la información, etc.

"Un usuario mayor típicamente tiene limitaciones de memoria a corto plazo, menor capacidad de coordinación, menor capacidad sensorial, y una más lenta capacidad de reacción." (Dong, 2002).

Los cambios en las funciones cognitivas en la vejez afectan a la velocidad de procesamiento de la información y a la memoria, que pueden causar problemas si el tiempo fuera de los procedimientos operativos son demasiado cortos. Las personas mayores funcionan peor que los adultos más jóvenes en términos de latencia de respuesta y precisión en tareas variadas (velocidad de percepción, la memoria de trabajo, el seguimiento, la toma de decisiones, memoria explícita y de procesamiento de múltiples). Este hecho apunta a un descenso constante en la calidad de la ejecución de una amplia variedad de tareas en la vida.

También es importante notar que la capacidad cognitiva varía considerablemente entre las personas mayores, así como dentro de los adultos más jóvenes. La compensación por disminución del cerebro puede ayudar a explicar por qué algunas personas ancianas envejecen con "gracia" mientras que en otros disminuyen rápidamente sus capacidades cognitivas.

2.3.7. TECNOLOGÍA EN EL AMBIENTE ACTUAL

En su artículo publicado en 1991 Weiser [1] se expande en la idea de los equipos ideales están tan incrustados en la vida cotidiana que son "indistinguibles de ella". Esta idea de llevar el equipo a un segundo plano, no sólo en la oficina de medio ambiente, sino también en el hogar, se ha tomado con gran aceptación, ya que un ordenador portátil no es una barrera para la comunicación humana y la interacción y en muchos casos lo facilita.



En 1996 Venkatesh [9] ya había señalado la tecnología y los cambios sociales ocurridos a finales de los años 80 y principios de los 90 como precursores y adoquines para la introducción de las modernas tecnologías en el entorno familiar. La predicción hecha por los hogares de Venkatesh con tareas automatizadas, aparatos inteligentes y elegantes en las casas del futuro parece estar sucediendo en este momento más que en el futuro. Independientemente del nivel de complejidad incorporado en la casa, las posibilidades que ofrece esta tecnología se consideran esenciales para algunas personas en cuanto a su felicidad y bienestar.

Las dificultades en el funcionamiento de electrodomésticos, así como la realización de otras tareas diarias son parte de los efectos que, en muchos casos, cultivan a las personas mayores tanto sensorial como cognitivamente.

Aparte de los factores económicos y las motivaciones, los recientes esfuerzos en el desarrollo de hardware y software dan soluciones a estos problemas. Mediante el desarrollo de lo que se denomina tecnología de asistencia, se muestra la importancia de mejorar la calidad de vida de las personas mayores y las personas con discapacidad. Casas inteligentes, teleasistencia y la telemedicina son ejemplos de esta práctica en la aplicación de la tecnología para el hogar, todos orientados a permitir a las personas, permanecer en sus hogares con una buena calidad de vida. Ayudas de memoria y ejercicios de reminiscencia son ejemplos de estas posibilidades. Con la tecnología de asistencia, las personas mayores pueden permanecer más tiempo en sus propios hogares, ahorrando dinero en costosos cuidados. Estas nuevas tecnologías también pueden disminuir los riesgos de accidente, los daños físicos existentes en el entorno familiar, ofrecer nuevas respuestas a viejos problemas y tener un impacto en un área de la vida diaria de gran preocupación para las personas mayores de población: la seguridad.

La introducción y el rápido desarrollo de nuevas soluciones tecnológicas facilitan la comunicación entre aparatos con dispositivos de control, con otros tipos de aparatos y también con los usuarios. Los sistemas móviles se utilizarán en un futuro en las casas para la prestación de servicios ya que son conscientes de las necesidades de los usuarios. Ifukube [10] sugiere que las herramientas inteligentes sean capaces de reconocer tanto verbal como no verbal la información.



Las nuevas tecnologías pueden aumentar las funciones corporales y permiten a los usuarios interactuar con su entorno de nuevas y diferentes maneras. También pueden hacer que los usuarios con discapacidad, sean los principales beneficiarios de estos nuevos avances proporcionando la capacitación, la independencia y también alternativas en su vida cotidiana. Es importante indicar que estos nuevos acontecimientos no se perciben para marginar o estigmatizar, sino para "normalizar" las herramientas. Abordar el estigma y la sola discapacidad, como fue señalado por Hanson [11], son dos de los problemas de las tecnologías de asistencia que siguen presentes hoy en día.

Es importante ver los beneficios sociales, así como de los psicológicos. Existen importantes beneficios psicológicos cuando las personas están sujetas a permanecer el mayor tiempo posible en la vivienda como hogar, entornos donde los objetos y el entorno interactúan para dar a la persona una identidad fuerte y libre. A pesar de que algunas casas de cuidado o residencias para las personas mayores, hacen que los residentes se sientan como en casa y pasen sus últimos años de vida en un ambiente cálido, otras instituciones continúan apareciendo más como un lugar para morir más que un lugar donde verdaderamente un ser humano vive. La necesidad de desplazarse a una residencia institucional ya ha sido identificado como un hecho por las personas mayores a evitarse si es posible, ya que tiene una amplia gama de efectos negativos para las personas mayores. Entre estos efectos negativos se encuentran la depresión, un sentimiento de aislamiento y una mayor reducción de la motivación para el auto cuidado.

Mann [12] encontró que en las personas mayores frágiles se provoca un descenso funcional experimentado en el tiempo, como se esperaba. Sin embargo, también encontró que la tasa de declive podría ser más lenta e institucional y que determinados costes de personal en su hogar pueden ser reducidos por la introducción de la tecnología de asistencia. En un estudio previo también encontró que un mayor uso de tecnología de asistencia se relaciona con una mayor independencia funcional. También sugiere que la reducción de los costos podría estar relacionada con la prevención de las lesiones debidas al uso de la tecnología de asistencia.

Existen varios requisitos de una casa que las personas mayores deben disponer. Estos requisitos comprenden a la protección, la seguridad, confort, comunicación y ahorro de



energía. Estos requisitos fueron establecidos para los futuros usuarios de la hipotética casa, y de entre ellos, el principal es la idea de la seguridad. El objetivo final de estos esfuerzos será el desarrollo de viviendas que no están diseñadas especialmente para las personas mayores o con discapacidad, pero si para que las casas se adapten a sus necesidades con el tiempo.

Por último, y en cuanto a resumen de este apartado del documento, hay que considerar la forma en que la población está envejeciendo en todo el mundo, y cómo el proceso de envejecimiento afecta a los distintos sentidos, así como las capacidades cognitivas. Se ha podido comprobar cómo el resultado del proceso de envejecimiento afecta la forma en que las personas interactúan con su medio ambiente y afecta a sus vidas cotidianas. Por lo tanto hay que reflejar la importancia de ayudar a las personas que viven de forma independiente el mayor tiempo posible. Al mismo tiempo el uso de las nuevas tecnologías en el hogar va en aumento y este puede influir positivamente en las condiciones de vida de las personas mayores y personas con discapacidad, así como la población en general.



3. METODOLOGÍA DE DESARROLLO

Para planificar y realizar un proyecto que cumpla todas las expectativas puestas en él y no deje ningún detalle suelto, es necesaria la realización de una metodología de desarrollo. En este apartado se plantea que es lo que se va hacer en este proyecto y cómo se realizará. A continuación se explicará la metodología a seguir para la realización del proyecto que ahora comienza.

Para la realización de este proyecto, se ha seguido una metodología que se puede dividir en tres fases principalmente. En una primera fase se presenta la determinación del problema, donde se realiza un análisis de los problemas que presenta la realización de este proyecto. En segundo lugar se debe realizar el proceso de desarrollo de la solución o proceso de diseño, donde se determina la forma de resolver los problemas encontrados en la fase anterior. Por último se realizará la evaluación del sistema.

3.1. DETERMINACIÓN DEL PROBLEMA

Desde el inicio del proyecto en donde, mediante la realización del estudio de documentación o las entrevistas mantenidas con el tutor, se lleva a cabo la determinación del problema que se presenta con la realización de este proyecto. La determinación del problema requiere un estudio acerca de los problemas actuales que existen entre las personas mayores y las nuevas tecnologías de información, así como el nuevo rumbo que está teniendo actualmente la computación ubicua en la sociedad actual. Dentro de esta fase se lleva a cabo la identificación del problema que se desea resolver mediante el desarrollo de un sistema de información. Por tanto en esta fase se deberá determinar el alcance del sistema y así poder comenzar en la elaboración de una posible solución al problema.

3.2. DESARROLLO DE LA SOLUCIÓN

En esta fase se realiza el desarrollo de la solución, en donde se resolverán todos los problemas encontrados para la realización del proyecto. Mediante el análisis y establecimiento de los requisitos que debe cumplir el software del sistema se emprenderá esta etapa del proyecto, en donde se comenzará a proponer soluciones que resuelvan el



problema. En esta fase se realiza el diseño del sistema, en donde se profundizará en el desarrollo de la solución al problema planteado en la fase anterior. El proceso seguido para la elaboración de la solución se determina a través de un modelo de desarrollo en cascada, que se define a continuación.

Para comenzar con la explicación del modelo de cascada empleado en este proyecto, hay que tener en cuenta una serie de cosas antes de comenzar la definición de la metodología seguida en este proyecto. En primer lugar, en el ciclo de vida de un proyecto software se debe definir un marco de referencia en el que se incluyan los procesos, actividades y tareas que forman parte del proceso de desarrollo, mantenimiento y despliegue de un producto software. Esto debe englobar todas las fases del proyecto, desde la primera definición del proceso hasta la finalización del producto final.

Se debe definir un modelo de ciclo de vida que contemple el estado de las fases que componen este proyecto, indicando además el orden en el que se deben realizar las tareas que lo conforman. Existen varios modelos de ciclo de vida, en este proyecto se ha decidido seguir un modelo en cascada con el objetivo de adaptarse con mayor exactitud a la realización de este proyecto.

El modelo en cascada define una perspectiva metodológica que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la etapa inmediatamente anterior. Con el empleo de este modelo de desarrollo se pretende que en la realización de este proyecto se realice una correcta organización y no se mezclen las diferentes fases.

Un modelo en cascada es ideal para proyectos donde se especifiquen muy bien los requerimientos y requisitos, por lo que el análisis de requisitos del sistema, requiere de una vital importancia para la correcta realización de este proyecto. Además este modelo es muy sencillo de entender ya que se siguen los pasos intuitivos necesarios a la hora de desarrollar el software del sistema. Este modelo puede verse reflejado en la siguiente ilustración:

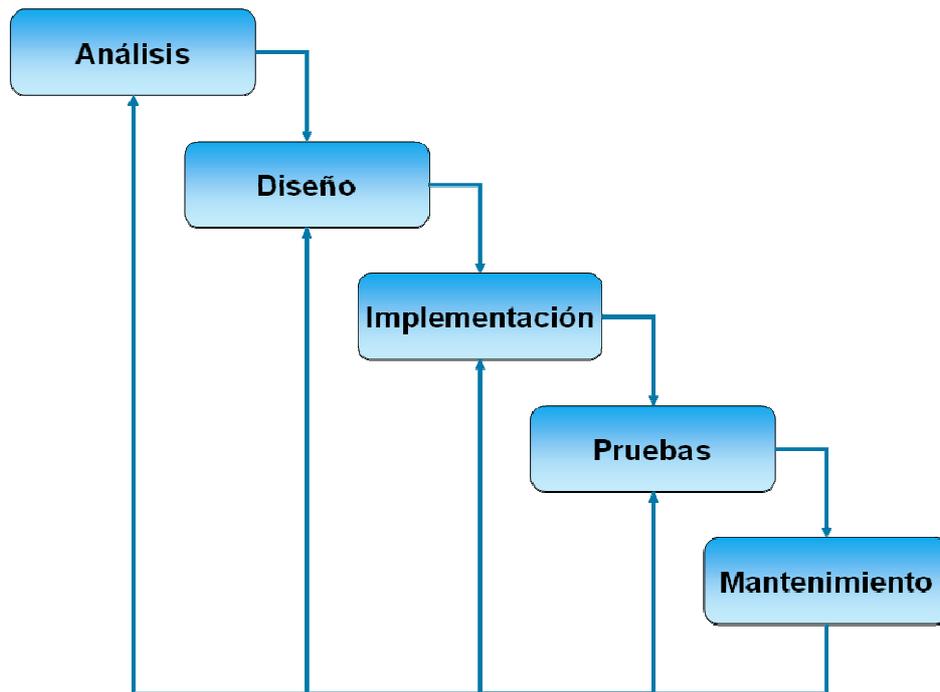


Ilustración 4. Modelo en cascada

3.3.EVALUACIÓN DE LA SOLUCIÓN

Una vez completadas las fases anteriores, es necesaria una evaluación del sistema, para comprobar su funcionamiento correcto, se debe realizar un sistema de uso sencillo y fácil manejo, ya que este proyecto va dirigido principalmente hacia personas mayores. Por tanto es necesario realizar una evaluación completa del sistema, para determinar si el sistema final cumple todos los requisitos y requerimientos que se establecieron en el análisis de la solución del problema.

Con la evaluación del sistema se pretende además recoger opiniones y sugerencias de los usuarios potenciales, con el fin de recabar información que ayude a los desarrolladores a realizar un sistema que cumpla todas las expectativas de los usuarios. Todas las opiniones de los usuarios son muy importantes, por lo tanto se debe poner especial interés en realizar una buena evaluación, ya que puede determinar el futuro inmediato de este sistema de información.



4. DESCRIPCIÓN DEL PROBLEMA

En el desarrollo de este proyecto se deberá tener en cuenta una serie de problemas, que deberán ser abordados y solucionados con la mayor precisión posible. Estos problemas podrían dividirse en diversas áreas como la usabilidad, problemas específicos de los usuarios potenciales..., etc.

4.1. PROBLEMAS DE DESARROLLO DEL SISTEMA QUE NOS COMPETE

Como problemas de desarrollo, se puede encontrar la dificultad del usuario con la navegación por la interfaz de usuario. En la navegación habrá que tener un especial cuidado, con la presentación de la información, por ejemplo, el tamaño del texto no deberá ser pequeño, el tiempo de aprendizaje de la interfaz deberá ser mínimo, en definitiva, problemas que produzcan al usuario frustración y le hagan desear del uso de este sistema.

Por otro lado, se ha de tener en cuenta que el usuario final del sistema, es una persona de edad avanzada, por lo que habrá que tener especial cuidado en diseñar una interfaz de usuario sencilla en su utilización, ya que este sector de la población no está acostumbrado a interactuar con este tipo de tecnologías. Se deberá tener también cuidado en las salidas del sistema, tanto la textual como la sonora (al tratarse de un sistema multimodal, esto deberá tenerse muy en cuenta).

En conclusión se debe desarrollar un sistema en el que se puedan llevar a cabo todas las actividades que ofrece, con la mayor simplicidad, sencillez, fiabilidad y eficacia como sea posible. Se resumirá en una serie de apartados los distintos problemas a los que nos enfrentamos en la realización de esta aplicación web móvil para la notificación de eventos a las personas de la tercera edad.

4.1.1. PROBLEMAS DEL USUARIO

Como se ha comentado anteriormente en el apartado general “Problemas de desarrollo del sistema que nos compete”, un problema importante que es necesario reflexionar a la hora de diseñar e implementar la interfaz de usuario de esta aplicación web, es que se tenga en cuenta las diferentes particularidades de los usuarios a los que va destinado



este sistema. En este caso el sector de usuarios finales, se encuentra entre los de edad avanzada, por tanto hay ciertas características o grados de incapacidades y deficiencias que se deben de tener en cuenta, así como los distintos grados del funcionamiento cognitivo (perdida de la sensibilidad, de la visión,..etc.). Todas estas peculiaridades referidas a este sector de usuarios, han sido recogidas en el apartado 2.3 de este documento.

Por todo lo comentado anteriormente, se debe realizar una aplicación centrada en las necesidades del sector de la población a la que va dirigida este proyecto. Se debe crear una aplicación web fácil e intuitiva de utilizar para tratar de resolver todas las dificultades que encuentren los usuarios potenciales en su uso.

Es importante darse cuenta que la introducción de las personas mayores a las nuevas tecnologías de la información supone un punto de ruptura con el estereotipo que existe actualmente en nuestra sociedad. Actualmente se vive con una cierta separación o frontera generacional entre la población joven y la anciana, debido a la fuerte relación que existe entre la informática o servicios tecnológicos actuales con la población joven.

Las personas mayores tienden a no enfrentarse a los retos y experiencias de las nuevas tecnologías de la información ya que son desconocidas por ellos y supone un cambio radical en su vida y en la forma de ver las cosas de su entorno. Esto supone un gran obstáculo a la hora de conseguir que esta población se familiarice con estas tecnologías y haga que el uso de estas herramientas tecnológicas no suponga un gran impacto en sus vidas.

Eva Rosado [13], periodista especializada en nuevas tecnologías pone de manifiesto en un artículo la lenta pero grave brecha que se abre entre la tercera edad y las nuevas tecnologías. Acusa a la misma tanto de la falta de simplicidad de algunos aparatos como de los prejuicios sociales existentes frente a la actual revolución de Internet, que la hacen percibir como algo negativo.

"Muchos ancianos asocian directamente el concepto de 'tecnología' con el de 'complejidad'"(E. Rosado).



4.1.2. INTERFAZ DE USUARIO

En primer lugar surgen los problemas derivados tras la creación de la interfaz de usuario. La interfaz de usuario es el medio por el cual se comunica la máquina y el usuario, la parte del sistema con la que el usuario interactúa. Al enfrentarse en esta aplicación con el diseño de un sistema interactivo, hay que tener en cuenta varios aspectos fundamentales.

En el diseño de interfaces de usuario es necesario tener en cuenta las habilidades cognitivas de las personas, así como la capacidad de percepción, por tanto es necesario adaptar el programa a estas capacidades teniendo en cuenta tanto el dispositivo como las características específicas de la máquina. Christine Faulkner expone en su libro “*The essence of Human-Computer Interaction*” [14], la existencia de un gran debate sobre la naturaleza de la memoria de las personas, de la misma manera que de la mente humana. Estudios muy completos acerca de la adquisición y retención de habilidades cognitivas, reflejan una complejidad muy grande, debido a la existencia de una gran diversidad de perspectivas y acercamientos realizados al tema.

Con la inclusión de las tecnologías de información en los estudios de las habilidades cognitivas, ha sido posible desarrollar en el procesamiento de la información unas consideraciones especiales dentro del modelo de interacción hombre-máquina. A pesar de la multitud de modelos existentes y la infinidad de teorías elaboradas, la meta que se pretende obtener en el análisis de las habilidades cognitivas en HCI es, como define de la siguiente forma Christine Faulkner:

“Desarrollar una firme comprensión de lo que un usuario puede razonablemente esperar recordar cuando trabaja con un sistema de información.”

Faulkner aporta una guía de consideraciones que deben ser tomadas para el desarrollo de Interfaces Gráficas de Usuarios:

- Las interfaces de usuario y los sistemas de información deben ofrecer un mayor *feedback* acerca de lo que está ocurriendo en todo momento.



- Las interfaces de usuario deben ser consistentes.
- Todo requisito de información ha de ser solicitado en un formato que debe ser común para el usuario.
- No se debe esperar a que la memoria humana realice tareas que lógicamente serían más fáciles realizarlas por la memoria de un ordenador.
- No se debe pensar en la interacción entre el usuario y los sistemas de información de manera aislada, debe existir un contexto de fondo.

Uno de los puntos más importantes que una interfaz debe hacer, es reducir la dependencia que tienen las personas de su propia memoria, no hay que forzarles a recordar cosas innecesarias como el contenido que apareció en una pantalla anterior. Tampoco se debe forzar al usuario a repetir operaciones ya realizadas como la introducción un mismo dato repetidas veces. Otros aspectos importantes son la interactividad y la eficiencia de la aplicación final. Para ello se deberá buscar la forma idónea para trabajar sobre aspectos psicológicos del usuario, sociales,..etc.

Las personas tienen unas habilidades distintas de los ordenadores, por tanto estos deben utilizar las suyas para prevenir las de las personas. Se debe tener en cuenta:

- **Velocidad de Aprendizaje.** Es necesario que el usuario aprenda a usar el sistema lo más pronto posible.
- **Velocidad de Respuesta.** Tiempo mínimo para realizar una operación en el sistema de información.
- **Tasa de errores.** El porcentaje de fallos o errores que comete un usuario navegando por un sistema de información.
- **Retención.** Capacidad que tiene usuario para saber cuánto recuerda sobre el uso del sistema en un período de tiempo.
- **Satisfacción.** Determina que el usuario esté a gusto con el sistema.

Todos estos aspectos están recogidos dentro de la metodología conocida como “Diseño centrado en el usuario”. Esta metodología es la filosofía y proceso de diseño en el que las necesidades, los deseos y las limitaciones del usuario final de una interfaz o documento, toman una atención y relevancia considerable en cada nivel del proceso de diseño. Las personas podrán llevar a cabo sus tareas o actividades productivamente, con fiabilidad, eficacia, comodidad, simplicidad y fiabilidad.

4.1.3. USABILIDAD

Se puede definir la usabilidad como la medida en la cual un producto puede ser usado por usuarios determinados para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso.



Ilustración 5. Esquema conceptual de la Usabilidad

En el año 2002 J.M. Carrol [16] definió la usabilidad como la calidad de un sistema respecto a:

- **Facilidad de uso.** Múltiples formas de interactuar información entre el usuario y el sistema
- **Satisfacción del usuario.** Se debe incluir un soporte al usuario capaz de garantizar sus fines
- **Facilidad de aprendizaje.** Se debe garantizar una interacción efectiva incluyendo las máximas prestaciones



A la hora de adentrarse en el diseño de la aplicación web se deben seguir una serie de principios de usabilidad que aseguren una facilidad de aprendizaje y satisfacción del usuario. En este proyecto se decidió emplear los principios de usabilidad de Jakob Nielsen [17] que constituyen la base de la evaluación heurística. Se ha decidido emplear estos principios puesto que son las directrices más conocidas en cuanto a usabilidad web y se adaptan perfectamente a la idea de sistema que se quiere desarrollar.

Los 10 principios desarrollados por Nielsen son los siguientes:

- **Visibilidad del estado del sistema.** El sistema debe siempre mantener informado a los usuarios de lo que ocurre, con un correcto *feedback* en un tiempo razonable.
- **Correspondencia entre el sistema y el mundo real.** El sistema debe hablar el lenguaje de los usuarios con palabras, frases y conceptos familiares. Sigue las convenciones del mundo real. Haz que la información aparezca en forma natural y lógica.
- **Control y libertad del usuario.** Los usuarios frecuentemente eligen opciones por error, por eso indica una salida clara a esas situaciones no deseadas sin necesidad de pasar por extensos diálogos.
- **Consistencia y estándares.** Los usuarios no tienen que adivinar que las diferentes palabras, situaciones o acciones significan lo mismo.
- **Evitar errores.** Un diseño cuidado que previene problemas es mejor que unos buenos mensajes de error.
- **Reconocimiento.** Haz objetos, acciones y opciones visibles. El usuario no tiene que recordar información de una parte a otra. Las instrucciones de uso del sistema deben estar visibles o ser fácilmente recuperables.
- **Diseño minimalista.** No hay que mostrar información irrelevante. Cada pedazo de información extra compete con la importante y disminuye su relativa visibilidad.



- **Flexible y eficiente.** Diseña un sistema que pueda ser utilizado por un rango amplio de usuarios. Brinda instrucciones cuando sean necesarias para nuevos usuarios sin dificultar el camino de usuarios avanzados. Permite a los usuarios avanzados ir directamente al contenido que buscan.
- **Reconocer, diagnosticar y recuperarse de los errores.** Para ayudar a los usuarios, los mensajes de error deben estar escritos en lenguaje sencillo, indicar el problema de forma precisa e indicar una solución.
- **Ayuda y documentación.** Facilita siempre una documentación o ayuda. La información debe ser fácil de encontrar, está dirigida a las tareas de los usuarios, lista los pasos concretos para hacer algo y es breve.

4.1.4. MULTIMODALIDAD

El concepto de multimodalidad según la W3C¹¹ consiste en la realización de un proceso en el cual diversos dispositivos y personas son capaces de llevar a cabo una interacción (auditiva, visual, táctil y gestual) conjunta desde cualquier sitio, en cualquier momento, utilizando cualquier dispositivo y de forma accesible, incrementando así la interacción entre personas, y entre dispositivos y personas.

El concepto de interacción multimodal ofrece la posibilidad de extender la interfaz de usuario para permitir múltiples modelos de interacción, ofreciendo a los usuarios la alternativa de utilizar la voz o cualquier dispositivo válido para introducir datos, como pueden ser un teclado, lápiz..., etc. Como respuesta, los usuarios podrán escuchar mensajes de audio, comandos hablados y a la vez ver la información gráficamente. En ese sentido, la capacidad por parte del usuario de poder especificar el modo o el dispositivo a emplear para una interacción particular, está pensada con el objetivo de llegar a mejorar la interfaz del usuario, así como su accesibilidad y fiabilidad.

Un sistema multimodal ofrece numerosas ventajas para la realización de este proyecto, para empezar ayuda a la usabilidad especialmente en el grupo de personas mayores, ya que se les ofrecen varios canales para obtener la información. Con el empleo de varios

¹¹ Guía Breve de la Interacción Multimodal

<<http://www.w3c.es/Divulgacion/Guiasbreves/Multimodalidad>>

canales de información, si por alguna deficiencia o discapacidad, una persona anciana no pudiese obtener información o tuviera dificultad para comprenderla, la multimodalidad ofrece al usuario el empleo de otros canales, que hagan que el usuario obtenga la información a pesar de tener alguna discapacidad.

Por otro lado, la multimodalidad ofrece la posibilidad de elegir al usuario el empleo de un canal concreto de información o varios a la vez y minimiza el esfuerzo asociado al proceso de aprendizaje, que siempre existe. De manera conceptual se podría describir los componentes básicos de la interfaz multimodal siguiendo el esquema representado en la Ilustración 6.

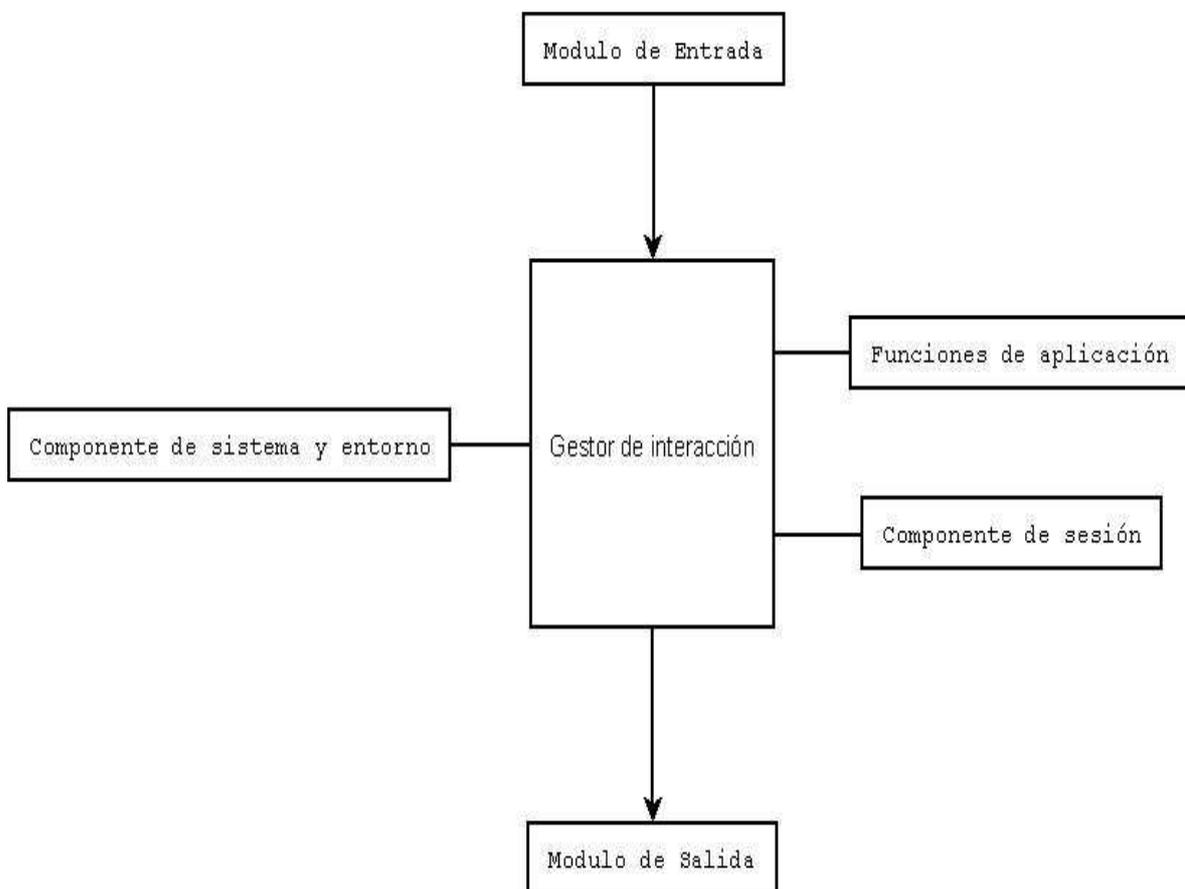


Ilustración 6. Componentes básicos de la interfaz multimodal



Se observa en la ilustración anterior que existen seis componentes básicos dentro de la interfaz multimodal:

- **Módulo de entrada.** En este módulo se especifican las diferentes formas que el usuario puede interactuar con el sistema. En este módulo se hacen los diferentes procesos de:
 - ✓ Reconocimiento: Se encargan de capturar la entrada de datos desde el usuario y transformarlos para su posterior procesamiento.
 - ✓ Interpretación: Se encargan de procesar aún más los resultados generados por los componentes de reconocimiento. Cada componente de interpretación identifica el *significado* o *semántica* deseada por el usuario.
 - ✓ Integración: Se encargan de combinar la salida de datos desde distintos componentes de interpretación. Algunas o todas las funcionalidades de este componente podrían ser implementadas como parte de los componentes de reconocimiento, interpretación e interacción

- **Módulo de salida.** En este módulo se proporciona al usuario la información de la manera solicitada por este, como puede ser por voz, texto, vídeo...etc. En este módulo se hacen los diferentes procesos de:
 - ✓ Generación: Son los modos de salida que se van a utilizar para presentar la información desde el administrador de interacción hasta el usuario. El componente de generación puede seleccionar un sólo modo de salida o puede seleccionar modos complementarios o suplementarios.
 - ✓ Estilo: Se encargan de añadir datos sobre cómo se va a mostrar la información.
 - ✓ Conversión: Se encargan de transformar la información del componente de estilo en un formato que será de fácil comprensión para el usuario.



- **Gestor de interacción.** Este componente se encarga de la gestión del flujo de los diferentes componentes de la interfaz. Asocia los diferentes objetos asociados a la interfaz con los módulos de entrada y salida.
- **Funciones de aplicación.** En este componente se definen las funciones que el usuario espera del sistema.
- **Componente de sesión.** Este componente es el encargado de soportar las diferentes sesiones persistentes y temporales de las aplicaciones multimodales. Además permite al usuario guardar el estado de la sesión en todo momento.
- **Componente de sistema y entorno.** Este componente es el encargado de gestionar los cambios que se producen en el gestor de interacción a través del terminal del usuario.

Se pueden encontrar varios problemas con respecto a la multimodalidad. Hay tres problemas claramente destacables, que el desarrollador deberá evitar:

- La complejidad que supone la integración de las diferentes tecnologías de interacción.
- Las costosas necesidades computacionales que requiere algunos modos de interacción.
- La dificultad que requiere el control temporal o el sincronismo de los eventos producidos por los diferentes modos de interacción.



4.2. APROXIMACIÓN DE LA SOLUCIÓN AL PROBLEMA

Presentados los problemas de desarrollo que se han de superar, en este punto se presentan las líneas a seguir para solucionar dichos problemas. Se presentarán unas directrices o recomendaciones de diseño que se han de tener en cuenta para solucionar la usabilidad del sistema así como los problemas del sector de usuario al que va referido el sistema.

4.2.1. DIRECTRICES PARA EL DISEÑO HACIA PERSONAS MAYORES

En su artículo publicado, Hilary Brown [18] aborda la idea de la necesidad de una serie de pasos para lograr la accesibilidad universal de los computadores y de Internet para las personas mayores y, de hecho, para todas las personas.

Para los ancianos, las directrices más importantes se refieren a la necesidad de que las imágenes y los modelos se adapten a sus deficiencias en la visión, audición, habilidades motoras y la memoria de trabajo. Algunas de estas directrices que se describen a continuación, tienen claras ventajas para una amplia población de usuarios. Brown en su artículo recogió dos grandes consorcios tecnológicos que proporcionan estas directrices comentadas.

El Consorcio *World Wide Web* en su Iniciativa de Accesibilidad Web proporciona un conjunto de directrices para los diseñadores de páginas web para dar cabida a una variedad de discapacidades y deficiencias. Las siguientes directrices son referidas especialmente al grupo de las personas mayores:

- Proporcionar alternativas equivalentes para el contenido visual y auditivo.
- Proporcionar mecanismos claros de navegación.
- No confiar en el color por sí solo (proporcionar señales redundantes).
- Proporcionar información de contexto y de orientación.
- Asegurar que los documentos sean claros y sencillos



Microsoft distribuye un libro en su página web sobre la "Tercera Edad y la tecnología"¹². Este documento describe una serie de directrices de diseño de páginas web específicamente dirigido a las personas mayores para hacer frente a deficiencias en la visión y habilidades motoras. El diseño, estilo, color, contraste y todas las opciones son fuente de debate.

Las recomendaciones incluyen:

- Utilizar grandes áreas de espacio en blanco y pequeños bloques de texto.
- Proporcionar mayores gráficos y objetivos donde un usuario únicamente tenga que hacer un *clic*.
- Uso de contraste y colores de fondo plano.
- Minimizar el número de imágenes y los efectos de animación.
- Usar un tamaño al menos 12 de letra y evitar numerosas fuentes de letra.

Hawthorn [5], prevé una aún más extensa serie de recomendaciones para abordar las deficiencias en la visión, el habla y la audición, la atención y la memoria en el diseño de cualquier tipo de interfaz gráfica para los ancianos:

- Utilizar un diseño basado en la simplicidad, claridad y coherencia.
- Empleo de bajos tonos de frecuencia de sonidos.
- Diseño de software de reconocimiento de voz.
- Utilizar sólo simples y relevantes gráficos.
- Evitar los retrasos y distracciones para reducir al mínimo la pérdida de la memoria a corto plazo.
- Empleo de textos cortos o listas frente a párrafos de texto.

¹² <http://www.microsoft.com/business/smb/es-es/internet/pagina_web.msp>



5. ANÁLISIS Y DISEÑO DEL SISTEMA

El objetivo de este apartado es obtener una especificación detallada del sistema que se quiere desarrollar. El análisis establece con precisión las necesidades y condiciones del sistema a desarrollar. Es una fase esencial, que es tomada como punto de partida para la realización del diseño de la aplicación.

La fase de diseño es un proceso iterativo a través del cual se traducen los requisitos del sistema en una representación del software. El diseño debe implementar todos los requisitos especificados en la fase de análisis y debe acumular todos los requisitos implícitos que desea el cliente.

5.1. ANÁLISIS DE REQUISITOS

El objetivo del análisis de los requisitos es analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar y obtener el conjunto de requisitos que debe cumplir el software del sistema. El conjunto ha de ser tan completo, consistente y correcto como sea posible. El resultado de la definición de requisitos debe satisfacer a todos los interesados del sistema. Todas las partes serán consultadas para minimizar los errores y reducir el riesgo de que la definición del sistema sea incompleta.

El proceso de análisis exige continuas realimentaciones. El primer catálogo de requisitos probablemente no será el definitivo y a medida que se profundice en el análisis surgirán nuevos requisitos que no se tuvieron en cuenta, o se modificarán requisitos que estaban mal planteados.

En este punto se muestra el catálogo de requisitos de sistema que definen las acciones que se pueden llevar a cabo en la aplicación. Estos han sido extraídos conforme a las funcionalidades establecidas por el usuario, añadiendo algunos otros que se ha considerado necesario para poder satisfacer las peticiones del usuario. El catálogo de requisitos que se va a establecer, está únicamente referido a la interfaz de usuario del sistema, no se tendrá en cuenta requisitos relativos al dispositivo físico en cuestión, como por ejemplo la duración de la batería,..etc.



El identificador de cada requisito será RX-NN, donde:

- X es F si el requisito es de funcionalidad
- X es NF si el requisito es no funcional
- NN es el número de requisito, secuencial empezando en 01.

Los requisitos se van a medir entorno a una serie de atributos:

- Necesidad: indica la negociación de la realización del requisito entre el desarrollador y el cliente:
 - ✓ Esencial: no es posible la negociación, ya que es imprescindible.
 - ✓ Deseable: el usuario desea que se realice lo pedido.
 - ✓ Opcional: la implementación del mismo es opcional y no implica perjuicio para el buen funcionamiento del sistema.
- Fuente: Cada requisito tendrá como origen al cliente o al desarrollador del sistema.
- Descripción: se describe al requisito de una forma clara y concisa.

5.1.1. REQUISITOS FUNCIONALES

Los requisitos funcionales de una aplicación especifican “qué” tiene que hacer el software. Están relacionados con los procesos que el sistema ha de realizar o con la información que el sistema debe contener.

1. Id	RF-01	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
El sistema debe avisar al usuario cada vez que suceda una emergencia.					

Tabla 4. Requisito de Sistema RF-01

1. Id	RF-02	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
El sistema va dirigido a las personas mayores, por tanto la información debe ser adaptada para este sector de población.					

Tabla 5. Requisito de Sistema RF-02



1. Id	RF-03	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
Existen 5 tipos de emergencias: <ul style="list-style-type: none">▪ Riesgo de inundaciones▪ Riesgo de incendios▪ Riesgo nuclear▪ Riesgo sísmico▪ Atentados terroristas					

Tabla 6. Requisito de Sistema RF-03

1. Id	RF-04	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
El sistema ofrecerá un modulo de ayuda al usuario sobre la emergencia que haya ocurrido.					

Tabla 7. Requisito de Sistema RF-04

1. Id	RF-05	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
La información ofrecida al usuario será en forma de mensajes.					

Tabla 8. Requisito de Sistema RF-05

1. Id	RF-06	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
Los mensajes estarán redactados en lenguaje castellano, de manera clara, breve y concisa para no generar dudas.					

Tabla 9. Requisito de Sistema RF-06

1. Id	RF-07	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
Los mensajes deben seguir las recomendaciones de semántica de expertos en el área de diseño de información.					

Tabla 10. Requisito de Sistema RF-07



1. Id	RF-08	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
Los mensajes tendrán siempre el mismo formato, aparecerán siempre en la misma posición dentro de la pantalla y únicamente se diferenciarán en el identificador del tipo de emergencia.					

Tabla 11. Requisito de Sistema RF-08

1. Id	RF-09	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
El sistema es multimodal, por lo tanto ofrecerá al usuario la posibilidad de recibir la información mediante un mensaje de voz.					

Tabla 12. Requisito de Sistema RF-09

1. Id	RF-10	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
El sistema permitirá al usuario poder ajustar el volumen de los sonidos.					

Tabla 13. Requisito de Sistema RF-10

1. Id	RF-11	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
El sistema ofrecerá un modulo de configuración al usuario para los correspondientes parámetros de la aplicación. Estos parámetros son el tamaño del texto de los mensajes del texto y el control del volumen de los mensajes de voz.					

Tabla 14. Requisito de Sistema RF-11

1. Id	RF-12	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
La interfaz de usuario será diseñada siguiendo los 10 principios de diseño desarrollados por Jakob Nielsen recogidas en el punto 3.1.3 de este documento.					

Tabla 15. Requisito de Sistema RF-12

1. Id	RF-13	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
El diseño de la aplicación se realizará siguiendo las pautas y recomendaciones de la “Iniciativa de Web Móvil” de W3C recogidas en el punto 2.2.1 de este documento.					

Tabla 16. Requisito de Sistema RF-13



1. Id	RF-14	2. Necesidad	Esencial	3. Fuente	Desarrollador
4. Descripción					
La navegación por el sistema será sencilla y fácil de recordar para el usuario.					

Tabla 17. Requisito de Sistema RF-14

1. Id	RF-15	2. Necesidad	Opcional	3. Fuente	Desarrollador
4. Descripción					
Los controles e iconos de la interfaz de usuario tendrán un texto asociado que indique claramente su significado o función.					

Tabla 18. Requisito de Sistema RF-15

1. Id	RF-16	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
El contenido de los mensajes sobre las directivas y protocolos de actuación tendrá la información adecuada sobre “qué” hacer en caso de la emergencia sucedida.					

Tabla 19. Requisito de Sistema RF-16

. Id	RF-17	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
El sistema ofrecerá al usuario la posibilidad de deshacer los efectos de acciones no intencionadas.					

Tabla 20. Requisito de Sistema RF-17

1. Id	RNF-18	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
El sistema ofrecerá un aspecto visual de la emergencia sucedida al usuario mediante una imagen del evento ocurrido, antes de indicarle las directivas y protocolos de actuación a seguir.					

Tabla 21. Requisito de Sistema RF-18

1. Id	RF-19	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
La información sobre cualquier error del sistema se le comunicará al usuario mediante mensajes y serán enviados constantemente hasta que el usuario realice su lectura.					

Tabla 22. Requisito de Sistema RF-19



1. Id	RF-20	2. Necesidad	Esencial	3. Fuente	Desarrollador
4. Descripción					
La navegación por la aplicación móvil se realizará mediante el empleo de los botones del dispositivo móvil.					

Tabla 23. Requisito de Sistema RF-20

1. Id	RNF-21	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
Se debe diseñar un software que minimice los pasos que un usuario debe seguir para realizar una operación.					

Tabla 24. Requisito de Sistema RF-21

1. Id	RF-22	2. Necesidad		3. Fuente	
4. Descripción					
Los mensajes de emergencia avisarán al usuario de la fecha y situación geográfica del suceso.					

Tabla 25. Requisito de Sistema RF-22

5.1.2. REQUISITOS NO FUNCIONALES

Los requisitos no funcionales se refieren a características del sistema relacionadas con la plataforma, el rendimiento, etc. Se utilizan para delimitar los requisitos funcionales al imponer condiciones a los mismos.

1. Id	RNF-01	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
La aplicación será portable y por tanto ejecutable en cualquier dispositivo móvil.					

Tabla 26. Requisito de Sistema RNF-01

1. Id	RNF-02	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
La aplicación se construirá mediante el empleo de la tecnología J2ME (<i>Java 2 Micro Edition</i>).					

Tabla 27. Requisito de Sistema RNF-02



1. Id	RNF-03	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
No se requiere de ninguna autenticación para entrar en el sistema.					

Tabla 28. Requisito de Sistema RNF-03

1. Id	RNF-04	2. Necesidad	Esencial	3. Fuente	Cliente
4. Descripción					
El sistema garantizará la integridad y disponibilidad de los datos de la emergencia o del protocolo de actuación.					

Tabla 29. Requisito de Sistema RNF-04

1. Id	RNF-05	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
La salida de sonido del sistema que alertará al usuario de una nueva emergencia, será implementada mediante grabaciones de voz.					

Tabla 30. Requisito de Sistema RNF-05

1. Id	RNF-06	2. Necesidad	Deseable	3. Fuente	Cliente
4. Descripción					
El sistema ofrecerá en cualquier momento al usuario la opción de salir del sistema.					

Tabla 31. Requisito de Sistema RNF-06

1. Id	RNF-07	2. Necesidad	Esencial	3. Fuente	Desarrollador
4. Descripción					
La interfaz de usuario estará organizada de tal forma que minimice los errores por parte del usuario, botones grandes y con nombres descriptivos, que defina de forma clara la funcionalidad que realizará ese botón.					

Tabla 32. Requisito de Sistema RNF-07

1. Id	RNF-08	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
El sistema proporcionará un mecanismo de <i>feedback</i> , con el fin de asegurarse que el usuario ha leído o comprendido las instrucciones a seguir en cada paso de la navegación por el sistema.					

Tabla 33. Requisito de Sistema RNF-08



1. Id	RNF-09	2. Necesidad	Opcional	3. Fuente	Desarrollador
4. Descripción					
El tipo de letra a utilizar para el contenido de los mensajes de información será <i>Sans Serif</i> .					

Tabla 34. Requisito de Sistema RNF-09

1. Id	RNF-10	2. Necesidad	Opcional	3. Fuente	Desarrollador
4. Descripción					
El tamaño de tipo de letra principal para el contenido de los mensajes de información será de 12 <i>px</i> .					

Tabla 35. Requisito de Sistema RNF-10

1. Id	RNF-11	2. Necesidad	Opcional	3. Fuente	Desarrollador
4. Descripción					
El texto del contenido de los mensajes de información presentará una alineación izquierda.					

Tabla 36. Requisito de Sistema RNF-11

1. Id	RNF-12	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
Los iconos de la aplicación tendrán un tamaño grande, permitiendo una mayor facilidad en su uso y una mejor visibilidad de los mismos.					

Tabla 37. Requisito de Sistema RNF-12

1. Id	RNF-13	2. Necesidad	Deseable	3. Fuente	Desarrollador
4. Descripción					
La información de los mensajes sobre los protocolos y directivas de actuación estará dividido en puntos para una mayor legibilidad al usuario.					

Tabla 38. Requisito de Sistema RNF-13

1. Id	RNF-14	2. Necesidad	Esencial	3. Fuente	Desarrollador
4. Descripción					
La información de los mensajes sobre los protocolos y directivas de actuación será recogida de las recomendaciones establecidas por protección civil en este tipo de emergencias.					

Tabla 39. Requisito de Sistema RNF-14



5.2. DISEÑO

Una vez definidos los requisitos de sistema, tanto funcionales como no funcionales, se entra en la fase de diseño que servirá al desarrollador de guía para realizar una correcta implementación. En esta fase, se seguirá un orden de las actividades que se han de realizar para poder realizar un diseño correcto y no caer en errores, que luego puedan causar problemas en la implementación.

En primer lugar se realizará un estudio intensivo de toda la documentación recogida hasta el momento, con el fin de aplicar todos esos conocimientos adquiridos a lo largo del proyecto al sistema desarrollado.

En segundo lugar se detallarán las posibles arquitecturas que presentan los sistemas para dispositivos móviles y se decidirá por una de ellas. Para la realización de la aplicación existen varias posibles arquitecturas a utilizar que satisfagan los objetivos del sistema, por ello la elección de la arquitectura del sistema que cumpla todos los requisitos es una labor muy compleja y, una buena elección será muy importante en la fase de diseño.

Posteriormente se planteará una serie de pautas y recomendaciones sobre el diseño de aplicaciones en dispositivos móviles que ayudará a diseñar e implementar el sistema final, siguiendo las pautas y normas establecidas en este tipo de aplicaciones móviles. El desarrollador que implemente el sistema, deberá estudiar esta serie de directrices para no caer en errores o fallos durante la fase de implementación.

Por último, en este punto del proyecto se debe realizar el diseño detallado del sistema, el cuál ha sido seguido para realizar la implementación del sistema. La implementación deberá realizarse basándose en el modelo conceptual. Este modelo conceptual muestra al usuario la arquitectura de la aplicación. Con el empleo de este modelo, se detalla el comportamiento que un usuario toma, desde el inicio de la aplicación hasta la ejecución final de esta.



5.2.1. ANÁLISIS DE LA DOCUMENTACIÓN

Durante todo el proyecto se ha ido realizando un análisis de toda la documentación encontrada relacionada con el proyecto: artículos, proyectos, *URLs* de información electrónica..., etc.

Las dificultades de aprendizaje, familiarización con las nuevas tecnologías de información y problemas que se presentan en el desarrollo de aplicaciones hacia personas mayores, requieren de un análisis y estudio exhaustivo con la finalidad de crear un sistema adaptado a las necesidades de este sector de población. Además de adaptar este sistema a las necesidades de los usuarios finales, es necesario solventar las deficiencias que presentan las personas de edad avanzada y mejorar su adaptación a las nuevas tecnologías de información. Por tanto, se ha realizado un análisis exhaustivo de la documentación, la cual se puede dividir en 3 partes para así realizar una mejor organización de la información encontrada:

1. **Aprendizaje de las personas mayores.**

Es de gran importancia recoger toda la documentación sobre estudios realizados acerca de la dificultad en el aprendizaje de las personas mayores en las nuevas tecnologías de información, el impacto que produce estas tecnologías en la vida de las personas de edad avanzada y lo más importante, conocer las deficiencias que posee este sector de la población a la hora de interactuar con un ordenador o un dispositivo móvil. Toda esta información es necesaria para realizar la descripción del problema, proponer soluciones para paliar o minimizar estos problemas de desarrollo y para realizar el análisis y diseño del sistema.

2. **Dispositivos móviles**

Por otro lado también es prioritario el estudio de la documentación sobre los dispositivos móviles, su funcionamiento y componentes, así como las diferentes recomendaciones de diseño para el desarrollo de aplicaciones móviles. Toda la información obtenida sobre los dispositivos móviles será tomada en cuenta en la fase de diseño, puesto que para realizar un diseño óptimo de una aplicación móvil,



es necesario analizar las recomendaciones y normas que se han establecido como guía a los desarrolladores de aplicaciones en dispositivos móviles.

3. **Plataforma de desarrollo**

Por último se ha de realizar un estudio sobre la documentación de la plataforma de desarrollo escogida para realizar la implementación. Por tanto se analizará la información obtenida sobre la plataforma elegida para tener una buena base de cara a la implementación del sistema, y no caer en errores ni confusiones que haga al desarrollador emplear más tiempo del estrictamente necesario en realizar dicha implementación.

5.2.2. **ARQUITECTURA DEL SISTEMA**

Las aplicaciones móviles ofrecen a los usuarios la posibilidad de actualizar en tiempo real los datos que mantiene compartidos con el sistema central cuando la red no se encuentra disponible o por el contrario no se puede acceder a los servidores de datos.

Esto ha hecho que en estos últimos tiempos haya surgido la replicación de los datos como la manera lógica de permitir a los diferentes clientes escribir, leer y compartir todos estos datos. En los esquemas optimistas cada cliente se guarda una copia de los datos en su caché, leyendo y escribiendo para ello de forma local, sin comunicárselo en ningún momento a otras máquinas.

Cuando varios clientes intentan actualizar los mismos datos, esas actualizaciones entran en conflicto, por lo tanto dichas actualizaciones deberán de ser integradas, o de otra forma, una de ellas deberá ser desechada si se desea eliminar este conflicto.

Para resolver los problemas de actualización de los datos transmitidos entre diferentes dispositivos móviles se definen dos arquitecturas que tratan de resolver este conflicto. Una vez presentadas estas dos arquitecturas, se realizará un análisis para determinar cuál es la arquitectura que mejor se adapta a las necesidades y requerimientos del sistema.

- Arquitectura cliente / servidor

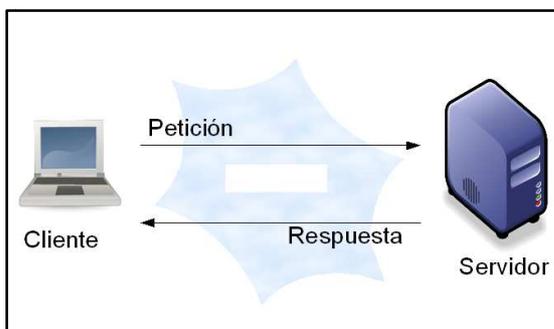


Ilustración 7. Arquitectura cliente/servidor

Una Arquitectura cliente/servidor como la de la Ilustración 7 se caracteriza en que los clientes móviles almacenan dentro de una caché los datos que recibe del servidor. Estos datos son actualizados en local y se sincronizan cuando se vuelven a conectar al servidor. De esta forma los clientes nunca pueden intercambiarse los datos entre sí.

- Arquitectura P2P (*peer to peer*)

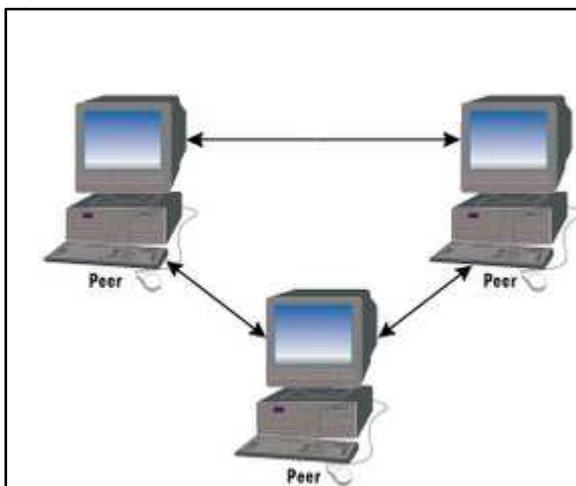


Ilustración 8. Arquitectura peer to peer

Una arquitectura *peer to peer* como la de la Ilustración 8 se caracteriza en que un cliente puede intercambiar sus actualizaciones con otro cliente. Las aplicaciones que se basan en esta arquitectura no suelen proporcionar suficientes garantías, cuando un grupo de clientes intenta actualizar el mismo grupo de datos a la vez.



Al tratarse de un sistema donde un organismo o institución (en este caso protección civil) manda mensajes de información al usuario notificándole la ocurrencia de una emergencia, la arquitectura que mejor se acopla a esta idea es la de cliente/servidor. En esta arquitectura, el organismo o institución actuaría como servidor mandando información al cliente sobre las emergencias, mientras que el usuario, que tiene su aplicación instalada en su dispositivo móvil para recibir los mensajes de información actuaría como cliente. La arquitectura *peer to peer* ha sido desechada, ya que en este sistema no existe la comunicación entre diferentes clientes, y únicamente es el servidor el que se conecta con los clientes, por tanto la elección de arquitectura más acertada es la de cliente/servidor.

5.2.3. RECOMENDACIONES DE DISEÑO DE DISPOSITIVOS MÓVILES

Para desarrollar aplicaciones Web aptas para pequeños dispositivos conectados a Internet, como teléfonos móviles, monitores o PDAs, se siguen una serie de pautas para indicar a los desarrolladores web, como debe realizarse el diseño de una aplicación:

- Es recomendable definir *URLs* cortas, debido a la dificultad de búsqueda y carga sobre dispositivos móviles.
- Debe asegurarse que el tamaño total de la página sea apropiado a las limitaciones de la memoria del dispositivo. Si las páginas son demasiado pesadas pueden tomar mucho tiempo para cargarse. Además, los dispositivos móviles en general tienen restricciones para visualizar páginas grandes.
- Al igual que ocurre con las aplicaciones desarrolladas para los equipos escritorio, hay navegadores que no respetan las intenciones del proveedor del contenido. Hay diferencias en la interpretación entre los navegadores y hay también deficiencias en la puesta en práctica, es decir, no se pueden soportar las características obligatorias de un estándar o una recomendación relevante como otros *bugs* o errores en la puesta en práctica.



- No utilizar tablas, a no ser que se sepa de antemano que el dispositivo las soporta.
- Evitar tener que hacer entrar al usuario a un subdirectorío dentro de un dominio por la misma dificultad.
- Es también muy recomendable proporcionar solamente una navegación mínima en la parte superior de la página. Cualquier otro elemento de navegación secundario se puede poner al pie de la página si realmente es necesario.
- No generar cambios de ventana o *pop ups* sin avisarle al usuario. Muchos dispositivos móviles no pueden soportar más de una ventana abierta.
- Se recomienda testear¹³ el sitio Web desarrollado en los navegadores que utilizan los dispositivos móviles y sobre los dispositivos móviles reales, ya que los emuladores de navegadores móviles en la práctica no funcionan exactamente igual que dentro del propio dispositivo.
- Proporcionar los enlaces básicos en una sola línea.
- Informar sobre enlaces que apunten a contenido en otro formato, de modo que los usuarios no se descarguen contenido que su dispositivo no puede visualizar o interpretar.
- No usar imágenes para espaciar. El mecanismo popular de utilizar un píxel no funciona en una variedad de pantallas.
- No se debe utilizar los mapas de imagen a menos que el dispositivo los soporte con eficiencia.
- Se debe asegurar que la información proporcionada a color es decodificada correctamente en pantallas monocromáticas.

¹³ tr. Arg. y Chile. Someter algo a un control o prueba.



- No se debe utilizar *frames*, ya que muchos dispositivos móviles no los interpretan.
- Asegurarse de proporcionar un buen contraste entre figuras y fondo. Los dispositivos móviles a menudo no tienen buen contraste de color y se utilizan en general en condiciones de la iluminación poco ideales.
- Muchos dispositivos móviles no soportan *scripts* y en muchos casos no es posible que los usuarios descarguen el *plug-in* correspondiente. El contenido se debe diseñar con esta limitación en mente. Aún cuando el dispositivo soporte *scripts*, no lo utilizarlos a menos que no haya otra manera de lograr sus objetivos. Los *scripts* aumentan el consumo de energía disminuyendo la vida de la batería.
- Por último, evitar las medidas absolutas para permitir que el navegador se adapte el contenido a la medida de pantalla.

5.2.4. MODELO CONCEPTUAL

Antes de comenzar con el diseño, se ha de detallar el modelo conceptual con el fin de mostrar al desarrollador la estructura de la aplicación. Hacer un buen modelo conceptual del sistema será una clave importante para el desarrollo posterior de la interfaz de usuario.

Mediante un diagrama UML de actividad se especificará en detalle el comportamiento que debe tomar el usuario entre el inicio y el final de la ejecución, y además definir la lógica de todas las operaciones del sistema. Con este diagrama se entra en un diseño más detallado del sistema, donde se especifica en cada momento la navegación por cada una de las pantallas.

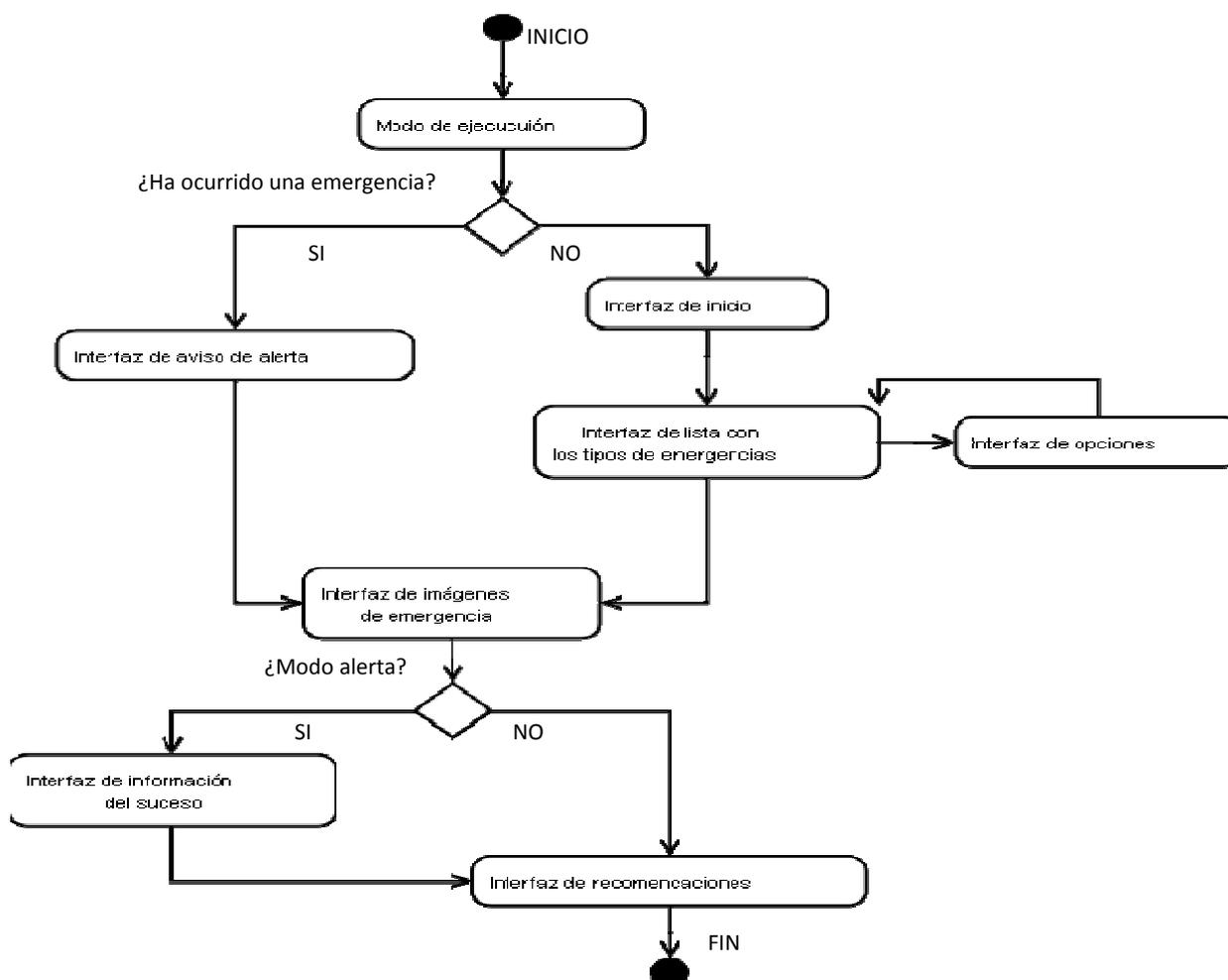


Ilustración 9. Diagrama UML de actividad

Como se observa en el diagrama UML de actividad de la aplicación, existen dos modos de ejecución, dependiendo del suceso de una emergencia o no. Por un lado, si se produjese una situación de emergencia el sistema se inicializaría en un modo alerta, en este caso se mostraría al usuario una interfaz que indicaría al usuario una situación de alerta. El otro modo de ejecución que existe en este sistema, es el modo normal, el usuario entra en el sistema sin que le llegue ningún aviso de alerta y únicamente quiere obtener información sobre las recomendaciones en caso de que suceda una emergencia.

En el caso de que el usuario entrase al sistema por el modo de alerta, se le daría la opción de finalizar la ejecución de la aplicación inmediatamente (porque ya conoce los pasos a seguir en ese tipo de emergencias o no desea conocerlas) o la opción de ir a la pantalla de suceso, donde se le informaría específicamente de la emergencia y su ubicación, en este caso habría una pantalla intermedia que le mostraría al usuario un



aspecto gráfico de la emergencia ocurrida. Una vez en esta pantalla se le ofrece al usuario la posibilidad de ir a la pantalla de recomendaciones, donde se le explica que hay que hacer en el caso de encontrarse en una situación de emergencia.

Si por el contrario la aplicación se inicializara en modo normal, se accedería a una pantalla de información donde se le mostraría al usuario una lista de los tipos de emergencia que contempla el sistema. Si se seleccionara alguno de éstos la aplicación iría a la pantalla de recomendaciones, donde se explica que hay que hacer en el caso de encontrarse en una situación de emergencia, de igual manera que en el modo alerta, en este caso también habría una pantalla intermedia que le mostraría al usuario un aspecto gráfico de la emergencia ocurrida. En esta pantalla de información, se daría la opción al usuario de modificar las opciones del dispositivo móvil. El usuario puede cambiar el tamaño de letra de los mensajes de recomendaciones y controlar el volumen de los mensajes de voz, ya que se trata de un sistema multimodal.

Por último, se le da al usuario la opción desde la pantalla de recomendaciones de finalizar la ejecución en caso de conocer ya las recomendaciones, o volver a la pantalla de información, si el usuario quisiese obtener más información sobre otros tipos de emergencias. En el apartado siguiente, se detalla cómo se ha implementado este modelo conceptual.



6. IMPLEMENTACIÓN

Una vez finalizada la etapa de diseño, se procede a realizar la implementación del sistema, tomando como referencia el diseño realizado con antelación. A lo largo de este apartado se comentan los detalles de implementación que se han considerado relevantes.

Antes de entrar de lleno en la explicación de la implementación del sistema, se detalla las herramientas de desarrollo que han sido utilizadas para la implementación de la solución, así como la forma en la cual se han resuelto los problemas de desarrollo de la interfaz de usuario y de la multimodalidad, y como han sido integradas las distintas partes de la aplicación.

Por último se realiza una explicación detallada de todas las clases determinadas en el diseño, además se da una explicación de cómo se ha llevado a cabo la implementación la los diferentes componentes del sistema.

6.1. HERRAMIENTAS DE DESARROLLO

Existen varias plataformas de desarrollo para aplicaciones móviles actualmente, las características de cada modelo o plataforma, determinará la elección de dicha plataforma para la implementación del sistema final. A continuación se realiza una descripción breve de las plataformas más importantes actualmente y se explicará por que se ha decido elegir la plataforma J2ME para la implementación de la aplicación web móvil. Una vez presentadas las plataformas de desarrollo más importantes actualmente, se realizará una descripción detallada de la opción escogida.

- .NET Compact Framework. Plataforma de desarrollo utilizada por Microsoft. Emplea XML junco con código controlado para el desarrollo y ejecución de aplicaciones web en dispositivos móviles. .NET Compact Framework es un subconjunto de la biblioteca de clases .NET Framework que también contiene clases diseñadas expresamente para él. Ejecuta programas independientes del hardware y el sistema operativo. Facilita el diseño y la optimización de los recursos de sistema limitados.



- ASP .NET Mobile Controls. Plataforma de desarrollo excelente para el diseño y ejecución de páginas web en el entorno Visual Studio .NET. Permite adaptar la interpretación y soporte de varias páginas web a un dispositivo concreto dentro del mismo modelo de programación.
- J2ME. Plataforma de desarrollo Java, enfocada a los dispositivos móviles. Permite la compatibilidad de aplicaciones entre cualquier dispositivo J2ME, la portabilidad de los programas desarrollados en diferentes dispositivos, rápido y fácil desarrollo y ofrece una gran robustez para las interfaces de usuario. J2ME es la tecnología del futuro para la industria de los dispositivos móviles. Esta plataforma es la elegida para realizar la implementación del sistema, ya que todas sus características hacen que sea la opción idónea para elaborar la aplicación de este proyecto.

6.1.1. PLATAFORMA J2ME

La plataforma J2ME (*Java 2 Micro Edition*) es un conjunto amplio de especificaciones donde se definen diversas versiones minimizadas de la plataforma de desarrollo Java 2. Estas versiones son usadas por los desarrolladores para programar multitud de aplicaciones en dispositivos electrónicos, desde teléfonos móviles, PDAs, hasta en tarjetas inteligentes, etc. Todos estos dispositivos tienen en común la poca potencia en el procesamiento y el no empleo de abundante memoria, además tampoco disponen de todo el soporte que proporciona J2SE (*Java 2 Standard Edition*), la plataforma estándar de Java 2 empleada en los sistemas de escritorio y servidores.

Esta plataforma es la única opción real que permite un desarrollo multiplataforma para aplicaciones en dispositivos móviles. Por ello la principal característica que tiene J2ME se basa en la posibilidad de crear aplicaciones independientes a la plataforma donde van a ser ejecutadas. Tiene una estructura altamente modular que le permite adaptarse a las características de cada dispositivo. Su arquitectura queda reflejada en la Ilustración 10.

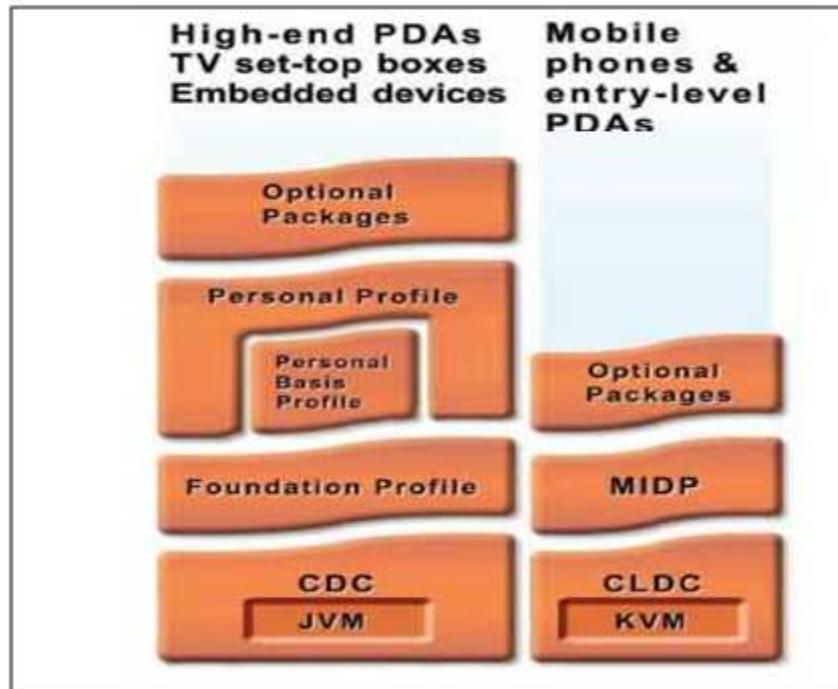


Ilustración 10. Arquitectura J2ME

Las aplicaciones Java ME toman como referencia esta arquitectura, la cual se desarrolla a partir de una combinación que se explica a continuación:

- Máquina Virtual: Dentro de la plataforma J2ME se pueden encontrar dos máquinas virtuales de Java ME. Cada una de estas máquinas virtuales fue pensada para diferentes tipos de dispositivos móviles:
 - ✓ CVM (*Compact Virtual Machine*). Esta máquina virtual soporta las mismas características que la Máquina Virtual de Java SE (JVM). Esta fundamentalmente pensada para dispositivos electrónicos con procesadores de 32 bits de gama alta y en torno a 2 MB de memoria RAM.
 - ✓ KVM (*Kilobyte Virtual Machine*). Esta máquina virtual se trata de una implementación reducida de una máquina virtual y esta fundamentalmente pensada para dispositivos de 16 o 32 bits, con al menos 25 MHz de velocidad y hasta 512 Kb de memoria total disponible.



- Configuraciones: Las configuraciones definen el entorno mínimo necesario para la ejecución de aplicaciones java en un grupo amplio de dispositivos móviles (CVM/KVM + paquetes básicos). Dentro de J2ME existe dos configuraciones:
 - ✓ CDC (*Connected Device Configuration*). Está enfocada a dispositivos con muchos más recursos, como PDAs de última generación o *smartphones* avanzados. Los dispositivos que utilizan el CDC admiten tres perfiles:
 - El *Foundation Profile* proporciona soporte básico de red y E/S.
 - El *Personal Basis Profile* contiene todas las clases del *Foundation Profile* más soporte limitado de *Javabeans* y AWT.
 - El *Personal Profile* incluye los dos perfiles anteriores más soporte completo de AWT y *Javabeans*.
 - ✓ CLDC (*Connected Limited Device Configuration*). Para dispositivos con potencia de cálculo, memoria, batería y acceso a red limitados.
 - MIDP (*Mobile Information Device Profile*) es el perfil más común con CLDC. Permite la implementación de aplicaciones con la siguiente funcionalidad:
 - Interfaz de usuario sencilla, adaptada a móviles
 - API especializado para juegos
 - Persistencia básica
 - Conexión por red
 - Sonido
 - La mayoría de los móviles actuales soportan CLDC +MIDP. Mucho éxito por la implementación de juegos
- Los perfiles definen paquetes adicionales para soportar funcionalidades básicas imprescindibles de cada familia de dispositivos
- Los paquetes opcionales sirven para cubrir funcionalidades específicas: *Bluetooth*, soporte 3D, etc.



Los beneficios que ofrece la plataforma J2ME a los desarrolladores en el diseño de aplicaciones para dispositivos móviles son los siguientes:

- Interfaces robustas, aumento de la flexibilidad y acceso completo a las funcionalidades de los dispositivos móviles.
- Compatibilidad de aplicaciones entre cualquier dispositivo J2ME
- Fácil distribución y rápido desarrollo
- Simplicidad en la integración *Wireless*(HTTP)

En el apartado siguiente, se detalla la tecnología que proporciona J2ME para la construcción de interfaces gráficas de usuario. Mediante esta forma se ha solventado el problema de desarrollo que se explicó en el apartado “4.1.2 Interfaz de usuario” de este documento.

6.1.2. MIDP: INTERFACES GRÁFICAS DE USUARIO

MIDP (*Mobile Information Device Profile*) ofrece una forma sencilla de construir interfaces gráficas de usuario adaptándolas a las limitaciones de pantalla y de batería de los dispositivos móviles, así como las que presentan la potencia de cálculo. La variedad y número de componentes existentes que ofrece MIDP es muy pequeño, muy por debajo de otros *toolkits* como *Java Swing*.

La interfaz de usuario se construye en torno a un conjunto de pantallas, en lugar de ventanas o diálogos. Las acciones que realiza un usuario en el sistema definen la transición de una pantalla a otra. Por supuesto es de gran importancia tener muy en cuenta el pequeño tamaño de la pantalla del dispositivo móvil a la hora de diseñar la interfaz.

La mayoría de los dispositivos móviles dispone de un par de botones de acción cuyo efecto se puede programar para cada pantalla. Estos botones de acción, también llamados comandos, son los que el usuario empleará durante su navegación por la aplicación desarrollada.

Las clases que implementan la interfaz de presentación son las siguientes (representadas por la interfaz de usuario que definen en su creación):

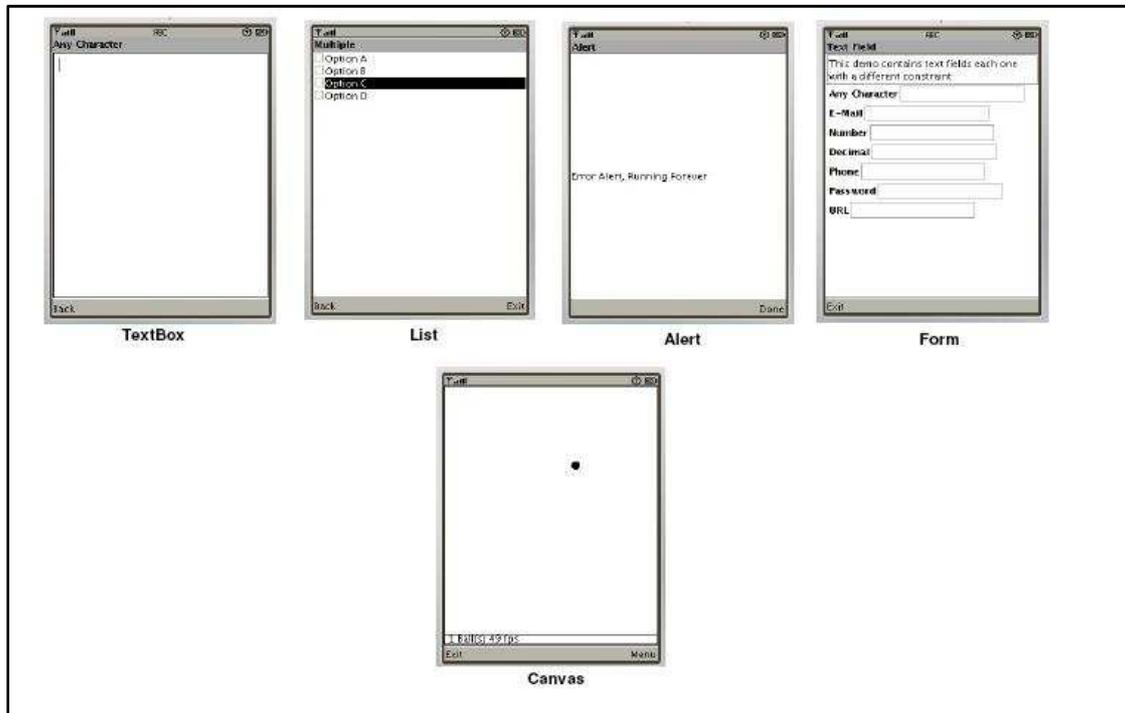


Ilustración 11. Tipos de pantallas MIDP

El *displayable Form* (Formulario) es el más característico de las clases que implementan la interfaz *Displayable*. Permite definir una pantalla con múltiples *Item* (o componentes):

- *StringItem*. Similar a un *label* de *Swing*
- *Spacer*. Un espacio con un ancho y alto determinado. Útil para distribuir los componentes
- *TextField*. Un editor de texto con una etiqueta asociada
- *ImageItem*. Una imagen con una etiqueta
- *DateField*. Un editor que permite introducir una fecha/hora



- *Gauge*. Sirve para representar de manera gráfica un valor entero
- *ChoiceGroup*. Sirve para seleccionar valores de una lista predeterminada. Puede ser múltiple, exclusiva o *popup*
- Cualquier *Item* definido por el usuario

En el apartado siguiente, se detalla el API que proporciona J2ME para la implementación de la multimodalidad dentro del sistema. Mediante esta herramienta se ha solventado los diversos problemas de desarrollo que fueron explicados en el apartado “4.1.4 Multimodalidad” de este documento.

6.1.3. API MULTIMEDIA J2ME

El API Multimedia de J2ME constituye una interfaz potente, flexible y sencilla para la manipulación y empleo de las capacidades multimedia. La plataforma J2ME proporciona tres paquetes en su API para soportar la generación, reproducción y grabación de contenido multimedia:

- El paquete *javax.microedition.media*
- El paquete *javax.microedition.media.control*
- El paquete *javax.microedition.media.protocol*

La arquitectura del API multimedia de J2ME se define en 3 conceptos básicos que se muestran a continuación:

- *Player* (Reproductor): Utilizado para reproducir unos datos multimedia concretos. Por ejemplo se necesita uno para reproducir archivos de audio, otro para tonos simples, etc.
- *Control* (Controlador): Se emplea para modificar el comportamiento de un reproductor concreto. A partir de una instancia de un reproductor podemos obtener su controlador asociado.

- *Manager* (Gestor): Este objeto es empleado para cohesionar todos los anteriores y sirve como punto de entrada al sistema, debido a que dispone de métodos estáticos que permiten crear tanto reproductores como fuentes de datos.

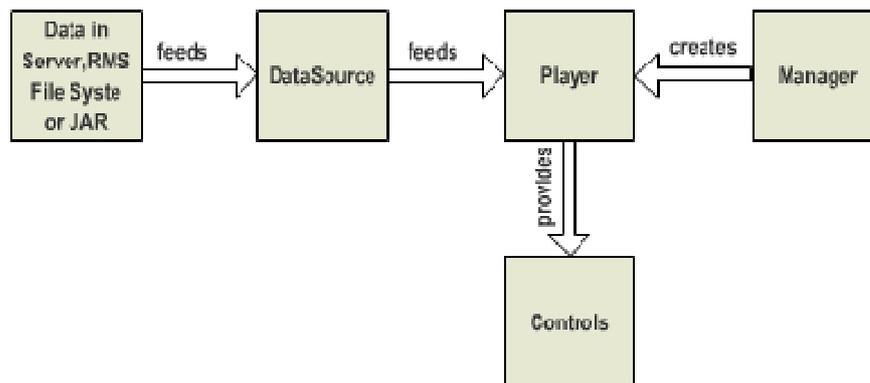


Ilustración 12. Arquitectura API Multimedia

Como se ve en la Ilustración 12 la arquitectura del API multimedia se divide en cuatro partes principales que realizan el procesado multimedia:

- *Player*: Lee y procesa los datos proporcionados por el objeto *DataSource*. Se encarga de controlar y ejecutar el contenido multimedia. Además *Player* nos permite no sólo reproducir audio sino que además permite controlar dicha reproducción e incluso su propio ciclo de vida.
- *DataSource*. Permite al objeto *Player* tratar el contenido, encapsulando y abstrayendo para ello la gestión del protocolo.
- *Manager*: Permite controlar los recursos de audio disponibles dentro del dispositivo y es solicitado por la aplicación para la creación de objetos *Player*.
- *Control*: Permite acceder a los diferentes controles proporcionados por el objeto *Player*.

6.2. DETALLES DE IMPLEMENTACIÓN

El sistema ha sido implementado mediante el empleo de un único *MIDlet* y varias clases Java anidadas que proporcionará al sistema un control de la complejidad, permitirá la visibilidad de las clases, permitiendo al sistema poseer un diseño robusto basado en la orientación a objetos. Además, con el empleo de estas clases anidadas se evitará un gasto de memoria muy alto, necesario para estos tipos de dispositivos, que tiene el inconveniente de poseer una capacidad de memoria muy reducida.

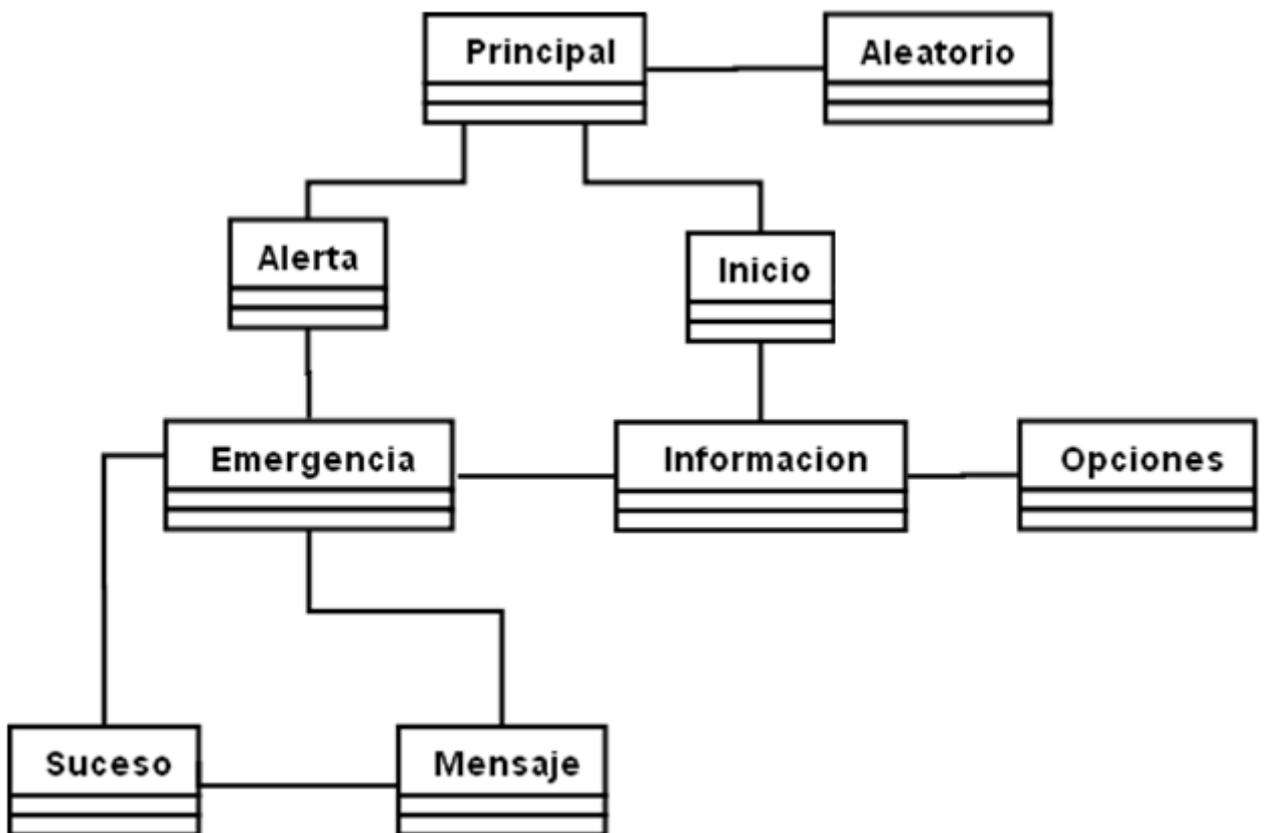


Ilustración 13. Diagrama de Clases

La organización de las clases implementadas del sistema queda reflejada en la Ilustración 13. Como se puede observar la clase principal del sistema es *Principal.java*, esta clase es la única que hereda de la clase *MIDlet* y por lo tanto servirá como referencia o base para las demás clases implementadas.



Además también hay otras 8 clases. De ellas hay siete clases que forman la interfaz de usuario del sistema divididas en dos grupos, las interfaces del API de J2ME de alto nivel y las de bajo nivel:

- API alto nivel: Estas clases heredan de la clase *Form*. En este sistema las clases de este grupo son *Informacion.java*, *Inicio.java*, *Mensaje.java* y *Opciones.java*.
- API bajo nivel: Estas clases heredan de la clase *Canvas*. En este sistema las clases de este grupo son *Alerta.java*, *Suceso.java* y *Emergencia.java*.

A continuación se detalla cómo ha sido implementada cada una de estas clases y cuál es su misión dentro del sistema, se mostrará la parte de código que hacer referencia a cada operación de la misión asignada a esa clase.

Principal.Java

Esta clase es la principal del sistema, por lo tanto hereda de *javax.microedition.midlet.MIDlet*. En esta clase, se gestionarán los diferentes estados de vida del *MIDlet* a través de los métodos *startApp()*, *pauseApp()* y *destroyApp()*. Además se debe encargar de los diferentes eventos de la aplicación.

```
public class Principal extends MIDlet {
```

Todos los eventos iniciales de la aplicación móvil se lanzan a través del método *startApp()*. Previamente a estos eventos se han de gestionar una serie de elementos. Lo principal es obtener el objeto *Display*, correspondiente al *MIDlet* de la aplicación. Esta operación se realiza en el constructor de la clase, empleando para ello el método *getDisplay()*, como se muestra a continuación.

```
public Principal() {  
    this.display = Display.getDisplay(this);  
    this.inicio = new Inicio(this, display);  
}
```



Una vez establecida la configuración inicial del *MIDlet* y creados todos los objetos existentes en la aplicación en el constructor de la clase, se dispone realizar en lanzamiento de dicha aplicación. En el método *startApp()* se realizará este lanzamiento. En primer lugar se hace una diferencia de la forma en que se debe iniciar la aplicación. Como se explico en el modelo conceptual, existen dos modos para lanzar la aplicación:

- Modo Alerta. Si ha ocurrido ninguna emergencia y se quiere notificar al usuario.
- Modo Normal. Si por el contrario, no ha ocurrido ninguna emergencia y el usuario únicamente entra al sistema en busca de información.

En este proyecto al no simular la notificación de una emergencia ocurrida por la red, se realizará dicha implementación haciéndonos ayuda de la aleatoriedad. En un 75% de las probabilidades se lanzará la aplicación en modo alerta y en un 25% restante se lanzará en modo normal. Como existen 5 tipos de emergencias en este sistema, si diese la casualidad de que el sistema se iniciará en modo alerta, habría una probabilidad equitativa que fuese una de los 5 tipos de emergencia.

Mediante el empleo de la clase Aleatorio (se explicara a continuación), obtenemos un número aleatorio y dependiendo del valor de ese número se iniciará la aplicación de un modo o de otro.

Si se iniciase en modo alerta, como se ha explicado hay un número equitativo de probabilidad de que saliese un tipo u otro, en esta aplicación se ha decidido asignar los siguientes números a los tipos de emergencia:

- 0: Incendio
- 1: Terremoto
- 2: Riesgo nuclear
- 3: Inundación
- 4: Atentado

A continuación, se muestra como ha sido implementada esta operación de inicio aleatorio.



```
Aleatorio a = new Aleatorio();
aleatorio = a.getAleatorio(0, 4);

if (aleatorio == 1) {
    this.display.setCurrent(inicio);
} else {
    aleatorio = a.getAleatorio(0, 4);
    switch (aleatorio) { //opcion del menu
        case 0:
            this.alerta = new Alerta(this, display, inicio, 0);
            this.display.setCurrent(alerta);
            break;
        case 1:
            this.alerta = new Alerta(this, display, inicio, 1);
            this.display.setCurrent(alerta);
            break;
        case 2:
            this.alerta = new Alerta(this, display, inicio, 2);
            this.display.setCurrent(alerta);
            break;
        case 3:
            this.alerta = new Alerta(this, display, inicio, 3);
            this.display.setCurrent(alerta);
            break;
        case 4:
            this.alerta = new Alerta(this, display, inicio, 4);
            this.display.setCurrent(alerta);
            break;
    }
}
```

Aleatorio.Java

Esta clase es utilizada únicamente para proporcionar un número aleatorio. En el constructor de la clase se inicializa un atributo de tipo entero como una variable aleatoria y mediante el método *getAleatorio()* se obtiene el numero aleatorio deseado, ofreciendo un límite inferior y superior que definirá el intervalo del numero aleatorio a obtener.

```
public Aleatorio() {
    aleatorio = new Random();
}

public int getAleatorio(int limiteInferior, int limiteSuperior) {
    return limiteInferior + Math.abs(aleatorio.nextInt() % (limiteSuperior - limiteInferior));
}
```



Inicio.Java

Esta clase representa el inicio de modo normal. Es una de las clases anidadas que extiende de *Canvas*. Además implementa la interfaz *CommandListener* presentando dos comandos: “Entrar” y “Salir”.

```
public class Inicio extends Canvas implements CommandListener {
```

Puesto que esta clase hereda de *Canvas* deberá implementar el método *paint()* que pintará en la pantalla todo lo que se implemente en este método. La pantalla de inicio presenta una imagen que se empleara de fondo de pantalla y una imagen en el medio de la pantalla que representa el logo del sistema:”Sistema de Alertas”.

```
protected void paint(Graphics g) {  
    g.setColor(0, 0, 255);  
    g.fillRect(0, 0, getWidth(), getHeight());  
    g.drawImage(fondo, getWidth() / 2, getHeight() / 2, Graphics.VCENTER | Graphics.HCENTER);  
    g.drawImage(logo, getWidth() / 2, getHeight() / 2, Graphics.VCENTER | Graphics.HCENTER);  
}
```

De igual forma, dado que esta clase implementa la interfaz *CommandListener* deberá implementar el método *commandAction()*. Como se ha comentado la pantalla de inicio contiene dos comandos (“Entrar” y “Salir”), al pulsar cualquiera de estos comandos el evento es manejado en este método. Si se pulsará el comando “Entrar” se accedería a la pantalla de información.

Si por el contrario se pulsará el comando “Salir” se finalizaría la ejecución del *MIDlet*, es decir, de la aplicación móvil. Para ello se haría una llamada al método *salirSistema()*, que finalizaría la ejecución llamando a su vez al método *destroyApp()*.

```
public void commandAction(Command c, Displayable d) {  
    if (c == salir) {  
        this.principal.salirSistema();  
    } else {  
        this.display.setCurrent(info.getMenu());  
    }  
}
```



Informacion.Java

Esta clase representa la información del sistema, es decir, el menú de los diferentes tipos de emergencias que hay en el sistema. Si se pulsará sobre uno de estos elementos de una lista de tipos de emergencia se accedería a la pantalla de recomendaciones en caso de suceder esta emergencia.

La clase Información hereda de la clase *Form*, que representa una pantalla de tipo formulario. Además implementa la interfaz *CommandListener* presentando dos comandos: “Atrás” y “Opciones”. También se definirá un comando por cada uno de los elementos de la lista de tipos de emergencia.

```
public class Informacion extends Form implements CommandListener {
```

Como esta clase implementa la interfaz *CommandListener* deberá implementar el método *commandAction()*. En este método se tomarán 3 caminos dependiendo del comando que se pulse:

- Si el comando seleccionado es un elemento de la listas de tipos de emergencias, se accederá a la pantalla correspondiente del tipo de emergencia que el usuario haya seleccionado.

Además se enviará la información sobre el tamaño del texto de los mensajes sobre los protocolos de actuación en caso del suceso de una emergencia y el volumen del mensaje de voz de estos mensajes. Por defecto se tendrá una configuración de tamaño de letra medio y un volumen de 50 (el rango del volumen va desde 0 a 100). Si se accediera a la pantalla de opciones y se modificará estos valores, la configuración cambiaría (se explicará más adelante en la clase Opciones).

- Si el comando seleccionado fuera “Salir” se saldría del sistema y concluiría la ejecución de la aplicación.
- Si el comando fuera “Opciones” se accedería a la pantalla de opciones.



```
public void commandAction(Command c, Displayable d) {
    // TODO Auto-generated method stub
    if (c == List.SELECT_COMMAND) { // Si selecciono
        switch (menu.getSelectedIndex()) { //opcion del menu
            case 0:
                emergencia = new Emergencia(this, this.display, 0);
                break;
            case 1:
                emergencia = new Emergencia(this, this.display, 1);
                break;
            case 2:
                emergencia = new Emergencia(this, this.display, 2);
                break;
            case 3:
                emergencia = new Emergencia(this, this.display, 3);
                break;
            case 4:
                emergencia = new Emergencia(this, this.display, 4);
                break;
        }

        emergencia.ponerTamanyo(opcion.getTamanyo());
        emergencia.ponerVolumen(opcion.getVolumen());

        this.display.setCurrent(emergencia);

    } else if (c == salir) {
        this.inicio.getPrincipal().salirSistema();
    } else {
        this.display.setCurrent(opcion);
    }
}
```

Opciones.Java

La clase Opciones es un objeto de tipo formulario que hereda de *Form*. En esta clase se establecerán las opciones de configuración de la aplicación, es decir, el tamaño del texto y el volumen de los mensajes de voz. Además esta clase implementa la interfaz *CommandListener*, con un único comando para volver a la pantalla anterior de información y la interfaz *ItemStateListener*. Con esta última interfaz se realizará la configuración de las opciones. El objeto empleado para manejar el tamaño de texto es un *ChoiceGroup* y para el control del volumen se ha empleado un objeto *Gauge*.

```
public class Opciones extends Form implements CommandListener, ItemStateListener {
```



Puesto que esta clase implementa la interfaz *ItemStateListener*, deberá implementar el método *itemStateChanged()*. En este método se diferenciará del tipo de *ítem* que queremos representar.

Si estamos en el caso del control del volumen, mediante un objeto de tipo *Gauge* se realizará este control. Un objeto *Gauge*, es un tipo de objeto visual que representa perfectamente la idea de volumen, haciendo un intervalo de sus posibles valores. En este caso se tienen 6 valores: 0, 20, 40, 60, 80 y 100 (de menor a mayor volumen). Mediante el empleo del método *setVolumen()*, se realiza la actualización de la configuración del volumen del sistema.

```
public void itemStateChanged(Item item) {
    // TODO Auto-generated method stub

    if (item == controlVolumen) { //Controlo el Gauge

        switch (((Gauge) item).getValue()) {

            case 0:
                this.setVolumen(0);
                break;
            case 1:
                this.setVolumen(20);
                break;
            case 2:
                this.setVolumen(40);
                break;
            case 3:
                this.setVolumen(60);
                break;
            case 4:
                this.setVolumen(80);
                break;
            case 5:
                this.setVolumen(100);
                break;
        }
    }
}
```



Si por el contrario estamos en el caso del tamaño del texto, se empleará un objeto de tipo *ChoiceGroup*. Este objeto representa una lista exclusiva de valores. En este caso disponemos de tres tipos de valores: 0 (pequeño), 1 (medio) y 2 (grande).

Mediante el empleo del método *setTamanyo()*, se realiza la actualización de la configuración del tamaño del texto del sistema.

```
    } else if (item == controlTexto) { //Controlo el Choice
        switch (((ChoiceGroup) item).getSelectedIndex()) {
            case 0:
                this.setTamanyo (Font.SIZE_SMALL);
                break;
            case 1:
                this.setTamanyo (Font.SIZE_MEDIUM);
                break;
            case 2:
                this.setTamanyo (Font.SIZE_LARGE);
                break;
        }
    }
}
```

Emergencia.Java

Esta clase anidada representa el paso intermedio entre el aviso de emergencia (en modo alerta) o la elección del tipo de emergencia en la pantalla de información (en modo normal) y la pantalla de recomendaciones. Esta clase hereda de *Canvas*, ya que se quiere insertar una imagen en la pantalla que represente la emergencia sucedida e implementa la interfaz *Runnable*.

```
public class Emergencia extends Canvas implements Runnable {
```

Como se ha comentado antes, esta clase implementa la interfaz *Runnable*, por tanto se crea un *Thread* (un hilo), con el objetivo de presentar la imagen en la pantalla de la emergencia sucedida durante dos segundos, para después acceder inmediatamente a la pantalla de recomendaciones si estamos en modo normal o la pantalla de sucesos, si estamos en modo alerta. El usuario por tanto no tendrá nada más que hacer que esperar a que el tiempo pase para ver las recomendaciones o más información sobre el suceso.



Mediante el método *run()* se realiza esta operación comentada anteriormente. Se duerme la pantalla 2000 milisegundos y posteriormente, cuando haya pasado este tiempo se accederá automáticamente a la pantalla de recomendaciones o de suceso.

```
public void run() {
    try {
        Thread.sleep(2000);

        if(emergencia){
            suceso = new Suceso(principal, info, display, alerta, tamanyoFuente, volumen);
            this.display.setCurrent(suceso);
        }else{
            mensaje = new Mensaje(info, display, alerta, tamanyoFuente,volumen);
            this.display.setCurrent(mensaje);
        }

        t.join();
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

Puesto que esta clase hereda de *Canvas*, se deberá implementar el método *paint()*. En este método es donde es llamando el *start()* del hilo. Al llamar a este método se inicia la ejecución del hilo, se llama al método *run()* y es dentro de ese método donde se implementa lo que se quiere hacer durante la ejecución del hilo. En este caso, se desea insertar una imagen de fondo que representará la emergencia sucedida de la siguiente manera:

```
protected void paint(Graphics g) {
    //Se pone el fondo de pantalla
    g.drawImage(fondo, getWidth() / 2, getHeight() / 2, Graphics.VCENTER | Graphics.HCENTER);
    this.start();
}
```



Suceso.Java

Esta clase representa la pantalla con la información asociada a una emergencia. La clase Suceso hereda de *Canvas*. Además implementa la interfaz *CommandListener* presentando dos comandos: “Entrar” y “Salir”.

```
public class Suceso extends Canvas implements CommandListener {
```

Puesto que esta clase hereda de *Canvas* deberá implementar el método *paint()* que mostrará en la pantalla todo lo que se implemente en este método. La pantalla suceso tiene una imagen que se empleara de icono de la emergencia sucedida, para dar un mayor realismo a la información del suceso. Además presenta un título informando al usuario el tipo de emergencia, y un texto debajo del icono, separado en dos secciones mediante una línea, informándole de la información específica de la emergencia y la situación donde ha ocurrido. A continuación se muestra el método *paint()*. Como se observa, mediante el atributo *alerta*, se diferencia el título con el aviso del tipo de emergencia, que posteriormente será incrustado en la pantalla, debajo del icono de información de emergencia. Mediante el método *escribirSuceso()* se escribe toda la información correspondiente al suceso y en el método *paint()* se pinta en la pantalla.

```
g.setColor(0, 0, 0);
g.fillRect(0, 0, getWidth(), getHeight());

g.drawImage(fondo, getWidth() / 2, getHeight() / 2, Graphics.VCENTER | Graphics.HCENTER);
g.drawImage(icono, getWidth() / 2, 22 * getHeight() / 100, Graphics.VCENTER | Graphics.HCENTER);

g.drawLine(0, 35 * getHeight() / 100, getWidth(), 35 * getHeight() / 100);
g.drawLine(0, 70 * getHeight() / 100, getWidth(), 70 * getHeight() / 100);

this.escribirSuceso();
g.drawString("!!! " + tipoAlerta + " !!!", getWidth() / 2, 7 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[0].getText(), 4 * getWidth() / 100, 40 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[1].getText(), 8 * getWidth() / 100, 48 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[2].getText(), 8 * getWidth() / 100, 56 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[3].getText(), 8 * getWidth() / 100, 64 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[4].getText(), 4 * getWidth() / 100, 75 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[5].getText(), 8 * getWidth() / 100, 83 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[6].getText(), 8 * getWidth() / 100, 90 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
g.drawString(si[7].getText(), 8 * getWidth() / 100, 97 * getHeight() / 100, Graphics.BASELINE | Graphics.LEFT);
```



Por último y dado que esta clase implementa la interfaz *CommandListener* deberá implementar el método *commandAction()*. Como se ha comentado la pantalla suceso contiene dos comandos (“Entrar” y “Salir”), al pulsar cualquiera de estos comandos el evento es manejado en este método. Si se pulsará el comando “Entrar” se accedería a la pantalla de recomendaciones, con la información de “que” hacer en caso de suceder dicha emergencia.

Si por el contrario se pulsará el comando “Salir” se finalizaría la ejecución del *MIDlet*, es decir, de la aplicación móvil. Para ello se haría una llamada al método *salirSistema()*, que finalizaría la ejecución llamando a su vez al método *destroyApp()*.

```
public void commandAction(Command c, Displayable d) {  
  
    if (c == salir) {  
        this.principal.salirSistema();  
  
    } else {  
        mensaje = new Mensaje(principal, info, display, alerta, this, tamanyoFuente, volumen);  
        this.display.setCurrent(mensaje);  
    }  
}
```

Mensaje.Java

Esta clase representa la pantalla de protocolo y normas de actuación en caso de suceder una emergencia. La clase Mensaje implementa la interfaz *CommandListener* para el empleo de los diferentes comandos, la interfaz *PlayerListener* para reproducir los mensajes de voz con las recomendaciones y la interfaz *Runnable*, para poder sincronizar y reproducir a la vez el mensaje de voz grabado y mostrar a usuario el texto con las recomendaciones.

```
public class Mensaje extends Form implements CommandListener, PlayerListener, Runnable {
```

Para mostrar en pantalla el mensaje que contiene la información sobre las recomendaciones en el caso de que suceda una emergencia se implementa el método *escribirMensaje()*. Los mensajes se presentarán en puntos, con un punto por cada



recomendación. Además a su vez habrá diferentes secciones en la presentación de los mensajes, que diferenciará la situación del usuario en el momento de la emergencia.

En este método lo primero que se realiza es una diferencia de fuentes del texto. Existirá una fuente que establecerá el estilo y el tamaño del encabezado de cada sección, es decir, la ubicación del usuario. Habrá otra fuente para los diferentes puntos con la información de las recomendaciones. El tamaño de esta fuente vendrá determinado por el valor obtenido en la pantalla de opciones.

Mediante el atributo *alerta* se podrá diferenciar el tipo de emergencia que ha sucedido, y por lo tanto en función del valor de este número se mostrará en pantalla las recomendaciones correspondientes.

```
private void escribirMensaje() {  
  
    si = new StringItem[20];  
    fuente = Font.getFont(Font.FACE_SYSTEM, Font.STYLE_PLAIN, this.tamanyoFuente);  
    fuente2 = Font.getFont(Font.FACE_SYSTEM, Font.STYLE_BOLD, this.tamanyoFuente);  
  
    for (int i = 0; i < 20; i++) {  
        si[i] = new StringItem("", "");  
        si[i].setFont(this.fuente);  
        this.append(si[i]);  
    }  
  
    switch (this.alerta) { //opcion del menu  
        case 0:  
  
            si[0].setFont(fuente2);  
            si[4].setFont(fuente2);  
            si[9].setFont(fuente2);  
  
            si[0].setText("Si está en el bosque \n");  
            si[1].setText("\t Tápese la nariz y la boca con un pañuelo húmedo para no respirar e  
            si[2].setText("\t Aléjese del fuego, siempre por zonas carentes de vegetación, carre  
            si[3].setText("\t Intente atravesar a las zonas ya quemadas, pero siempre por donde
```

Como se ha comentado antes, esta clase implementa la interfaz *Runnable*, por tanto se crea un *Thread* (un hilo), con el objetivo de reproducir un mensaje de voz con las recomendaciones a la vez que el usuario observa en su móvil el texto con estas recomendaciones.



Mediante el método *run()* se realiza esta operación comentada anteriormente. Usando el atributo *alerta* se seleccionará el tipo de emergencia sucedida, se almacenará en una variable *InputStream* y posteriormente se llamará al método *play()* para reproducir este mensaje.

```
public void run() {
    switch (this.alerta) { //opcion del menu

        case 0:
            is = getClass().getResourceAsStream("/sonidos/Incendio.wav");
            break;

        case 1:
            is = getClass().getResourceAsStream("/sonidos/Terremoto.wav");
            break;

        case 2:
            is = getClass().getResourceAsStream("/sonidos/Nuclear.wav");
            break;

        case 3:
            is = getClass().getResourceAsStream("/sonidos/Inundacion.wav");
            break;

        case 4:
            is = getClass().getResourceAsStream("/sonidos/Atentado.wav");
            break;

    }
    this.play(is);
}
```

Puesto que implementa la interfaz *PlayListener* se ha de implementar el método *play()*. Mediante el método *play()* se reproduce el mensaje grabado. Para empezar se debe obtener el objeto de tipo *Manager*, este será utilizado para el control y la reproducción multimedia, con la única restricción de que los datos manipulados deben pertenecer específicamente al tipo sobre el que se creó el objeto.

Una vez hecho esto, se asignará el volumen seleccionado en la interfaz de opciones y se comenzará con la reproducción del audio.



```
public void play(InputStream is) {

    try {
        this.estadoPlayer();
        // create a player instance
        player = Manager.createPlayer(is, "audio/x-wav");
        player.addPlayerListener(this);

        // realize the player
        player.realize();

        //Se asigna el volumen
        this.controlVolumen();

        //Se reproduce
        player.prefetch();
        player.start();

    } catch (Throwable t) {
        this.reset();
    }

}
```

Mediante el empleo del método *controlVolumen()* se realiza la configuración del volumen de los mensajes de voz. Si el valor del volumen seleccionado por el usuario en la pantalla de opciones es 0, el reproductor no se escuchará y se pondrá en *Muted*. Si tuviese un valor mayor que 0, se le asignará al reproductor mediante el método *setLevel()*.

```
public void controlVolumen() {

    this.vc = (VolumeControl) player.getControl("VolumeControl");

    if (vc != null) {
        if (this.volumen == 0) {
            vc.setMute(true);
        } else {
            vc.setMute(false);
            vc.setLevel(this.volumen);
        }
    }

}
```



Por último y dado que esta clase implementa la interfaz *CommandListener* deberá implementar el método *commandAction()*. En esta pantalla se diferencian los comandos, dependiendo desde que pantalla se haya accedido a esta (se ha implementado mediante el uso de una variable booleana que es pasada en el constructor desde la pantalla anterior a esta)

- Si se accede desde la pantalla inicial del modo alerta (el valor de la variable booleana *emergencia* es *true*) se tendrán los comandos “Atrás” que volvería a la pantalla de sucesos, y el comando “Información” que accedería a la pantalla de información, con la lista de los diferentes tipos de emergencia.
- Si se accede desde la pantalla de información (el valor de la variable booleana *emergencia* es *false*) se tendrá únicamente el comando “Atrás” que retornaría a la pantalla de información, con la lista de los diferentes tipos de emergencia.

Además en cada comando va añadido el método *stopPlayer()* con el fin de dejar de reproducir el mensaje de voz de forma inmediata.

```
public void commandAction(Command c, Displayable d) {
    if (emergencia) {
        if (c == atras) {
            this.display.setCurrent(suceso);
            this.stopPlayer();
        } else if (c == infor) {
            this.display.setCurrent(info.getMenu());
            this.stopPlayer();
        }
    } else {
        this.display.setCurrent(info.getMenu());
        this.stopPlayer();
    }
}
```



Alerta.java

Esta clase representa el inicio de modo alerta. La clase *Alerta* hereda de *Canvas*. Además implementa la interfaz *CommandListener* presentando dos comandos: “Entrar” y “Salir”. También implementa la interfaz *PlayerListener* para reproducir un mensaje de voz que alerte al usuario del suceso de la emergencia de forma sonora y la interfaz *Runnable*, para poder sincronizar y reproducir a la vez el mensaje de voz de alerta y mostrar a usuario el aviso gráficamente.

```
public class Alerta extends Canvas implements CommandListener, PlayerListener, Runnable {
```

Puesto que esta clase hereda de *Canvas* deberá implementar el método *paint()* que mostrará en la pantalla todo lo que se implemente en este método. La pantalla de alerta tiene una imagen que se empleará de fondo de pantalla de color negro metalizado. Además presenta un texto anunciando al usuario la ocurrencia de una alerta, una imagen en el medio de la pantalla con el símbolo de peligro de tamaño grande, para dar una mayor visión al usuario del motivo de alerta y un texto en la parte inferior de la pantalla indicando el tipo de emergencia ocurrida. Este último texto, se implementa mediante el atributo *alerta* que servirá de ayuda para saber que alerta ha sucedido.

```
protected void paint(Graphics g) {  
    // TODO Auto-generated method stub  
    String tipoAlerta = "";  
    switch (this.alerta) { //opcion del menu  
        case 0:  
            tipoAlerta = "INCENDIO";  
            break;  
        case 1:  
            tipoAlerta = "TERREMOTO";  
            break;  
        case 2:  
            tipoAlerta = "RIESGO NUCLEAR";  
            break;  
        case 3:  
            tipoAlerta = "INUNDACION";  
            break;  
        case 4:  
            tipoAlerta = "ATENTADO TERRORISTA";  
            break;  
    }  
}
```



```
g.setColor(255, 128, 0);
g.fillRect(0, 0, getWidth(), getHeight());
g.drawImage(fondo, getWidth() / 2, getHeight() / 2, Graphics.VCENTER | Graphics.HCENTER);

//Escribimos el texto en la pantalla
fuente = Font.getFont(Font.FACE_SYSTEM, Font.STYLE_PLAIN, 16);
g.setFont(fuente); //Seleccionamos la fuente activa.
g.drawString("¡¡ ATENCION: ALERTA !!", getWidth() / 2, 10 * getHeight() / 100, Graphics.BASELINE | Graphics.HCENTER);
g.drawImage(simbolo, getWidth() / 2, getHeight() / 2, Graphics.VCENTER | Graphics.HCENTER);
g.drawString(tipoAlerta, getWidth() / 2, 90 * getHeight() / 100, Graphics.BASELINE | Graphics.HCENTER);
```

Por último y dado que esta clase implementa la interfaz *CommandListener* deberá implementar el método *commandAction()*. Como se ha comentado la pantalla de alerta contiene dos comandos (“Entrar” y “Salir”), al pulsar cualquiera de estos comandos el evento es manejado en este método. Si se pulsará el comando “Entrar” se accedería a la pantalla de información del suceso, con la información específica de la emergencia ocurrida.

Si por el contrario se pulsará el comando “Salir” se finalizaría la ejecución del *MIDlet*, es decir, de la aplicación móvil. Para ello se haría una llamada al método *salirSistema()*, que finalizaría la ejecución llamando a su vez al método *destroyApp()*.

Además en cada comando va añadido el método *stopPlayer()* con el fin de dejar de reproducir el mensaje de voz de forma inmediata.

```
public void commandAction(Command c, Displayable d) {
    // TODO Auto-generated method stub
    if (c == salir) {
        this.stopPlayer();
        this.principal.salirSistema();
    } else {
        this.stopPlayer();
        emergencia = new Emergencia(this.principal, this.inicio, this.display, this.alerta);
        this.display.setCurrent(emergencia);
    }
}
```



7. EVALUACIÓN

En este proyecto se ha desarrollado una aplicación web multidispositivo interactiva, por lo tanto es de vital importancia realizar una evaluación de la usabilidad del sistema con los usuarios potenciales. El objetivo que se pretende conseguir mediante la realización de la evaluación es comprobar si los usuarios potenciales se sienten cómodos en el manejo de la aplicación y si piensan realmente, que ésta es fácil de usar. Por lo tanto se ha de realizar una evaluación completa del sistema, ya que es fundamental que se verifique que la aplicación funciona correctamente y cumple las exigencias de los usuarios.

7.1. TÉCNICA DE EVALUACIÓN EMPLEADA

La técnica que se ha empleado para realizar la evaluación de la usabilidad del sistema ha sido la técnica de recorrido cognitivo. Existen otras técnicas como el análisis de tareas, uso de escenarios, grupos de interés o talleres, software de registros... etc. Se ha decidido escoger esta técnica ya que su principal característica es que se basa en hacer al usuario aprender a manejar la aplicación de forma exploratoria, para poder así evaluar correctamente el sistema. El usuario avanza lentamente a través de la tarea de un escenario, haciendo para cada acción del usuario, un análisis detallado de sus intenciones, su conocimiento, procesos mentales e interpretaciones. Hay que focalizar un único objetivo de la usabilidad: "la facilidad de aprendizaje". La gran virtud de esta técnica es que al estar tan enfocada a la resolución de tareas concretas se consigue detectar un elevado número de problemas, inconsistencias y mejoras.

En la evaluación del sistema se le realiza a cada usuario una entrevista semi-formal, empleando para ello una plantilla de evaluación (en el apéndice B se muestra esta plantilla). En esta plantilla se explica de forma detallada para que se puede utilizar este proyecto y como es su manejo, además si el usuario tuviese alguna duda, se le explicaría todo para aclarar confusiones. Una vez comprendido todo por el usuario, se le dejará al usuario que interactúe y aprenda a manejar la aplicación, explorándola un tiempo indefinido hasta que puedan ver todas sus funcionalidades y aprender a utilizarlo correctamente.



Además al final de la entrevista, se pedirá al usuario una opinión sobre los distintos temas que conforman el sistema, así se podrá obtener una visión más real de sus valoraciones acerca de las necesidades y requerimientos que tiene cada usuario sobre la notificación de emergencias.

7.2. RESULTADOS

Una vez realizada la entrevista con cada usuario, se procede a realizar las conclusiones oportunas a partir de los resultados obtenidos. Los usuarios analizados para esta evaluación fueron 5, en los que se puede dividir en dos grupos, jóvenes con conocimientos en las tecnologías de información y personas de edad avanzada si ningún o poco conocimiento acerca de las tecnologías actuales. Se realizó la evaluación a dos personas con edad comprendida entre los 23 y 28 años, y tres personas de edad avanzada entre 55 y 65 años.

Los resultados de la plantilla anterior obtenidos son los que se recogen en la siguiente tabla (como se explica en la plantilla de evaluación ubicada en el apéndice B del presente documento, cada respuesta puede tomar un valor de B (bueno), R (regular) y M (malo), cada pregunta de la evaluación tiene esa enumeración).

Se podría haber usado otro tipo de escala para las respuestas de la pregunta, una escala que fuera del 1 al 10, una escala de tipo *Likert*¹⁴, u otras más complejas, pero se ha decidido utilizar este tipo de escala, ya que las personas mayores se adaptan mas a este tipo de respuestas y al tratarse de la evaluación de un sistema no muy complejo, es ventajoso realizar una escala tan sencilla.

Además lo más importante de esta evaluación son las opiniones o sugerencias de los usuarios acerca de la usabilidad del sistema, por tanto no hace falta hacer un estudio exhaustivo sobre las respuestas del cuestionario. Una vez observados los resultados obtenidos en la siguiente tabla, se realizará un estudio sobre las respuestas de los usuarios y sobre sus comentarios acerca de la usabilidad del sistema.

¹⁴ Se trata de una escala psicométrica empleada comúnmente en todo tipo de cuestionarios. Esta es la escala de mayor empleo en las encuestas para la investigación. Cuando se responde a un elemento de un cuestionario elaborado con la técnica de *Likert*, lo que realmente se está haciendo, es especificar el nivel de acuerdo o desacuerdo con una declaración (elemento, ítem o reactivo).



	DATOS GENERALES			DATOS ESPECÍFICOS DE LA APLICACIÓN									DATOS ESPECÍFICOS DE LOS MENSAJES						
	1	2	3	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7
EV-01	23	M	D	B	R	B	B	B	R	B	R	B	B	B	B	B	B	B	B
EV-02	65	H	C	B	R	B	B	B	B	R	R	B	B	B	B	B	B	B	B
EV-03	60	M	B	B	B	B	B	B	B	R	R	B	B	B	B	B	B	B	R
EV-04	28	H	D	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	R
EV-05	56	H	C	B	R	B	B	B	B	B	R	B	B	B	B	B	B	B	B

Tabla 40. Resultados de la evaluación

Como se observa en los resultados obtenidos, a pesar de con contar con alguna respuesta “Regular” se puede considerar que en términos generales, los resultados obtenidos fueron realmente satisfactorios, todos valoran positivamente la navegabilidad por la aplicación, el tamaño del texto, la distinción de colores, así como la funcionalidad de la aplicación y de los diferentes comandos que en ella existen. En cuanto a las sugerencias y opiniones de cada uno de los usuarios evaluados, se pueden sacar conclusiones muy interesantes con ellas. Todo los usuarios se centraron en aspectos más concretos como se les pidió al principio y mostraron sus sugerencias sobre la aplicación.

Un comentario generalizado por todos los usuarios, fue el manejo sencillo y práctico del sistema. Todos los usuarios determinaron que resulta una aplicación muy útil en la sociedad actual y dejaron comentarios buenos al respecto. En cuanto a los demás aspectos del sistema, cada usuario tuvo su opinión sobre el sistema. Como era de esperar, los comentarios que se obtuvieron al final de la entrevista, tomaron dos caminos no muy diferentes pero si variantes, dependiendo de los 2 grupos anteriormente explicados. Esto refleja lo que dijo Eva Rosado en su artículo, sobre la lenta pero grave brecha que se abre entre tercera edad y nuevas tecnologías (apartado “4.1.1 Problemas del usuario” del presente documento).

Los usuarios del grupo joven no tuvieron ningún problema en el manejo de la aplicación, una vez leído lo expuesto en la plantilla no tuvieron mayor problema en navegar por la aplicación. Por el contrario los usuarios del sector de personas mayores si tuvieron más dificultades para manejar la aplicación como era de esperar. A estos usuarios se les ayudo



en la navegación del sistema, hasta que todos supiesen interactuar con la aplicación por sí solos.

Este último grupo destacó del sistema, el tamaño del texto de los diferentes botones de comandos, ya que les resultó pequeño, aunque se podía distinguir lo que ponía. En su afán de ayudar a mejorar el sistema en un futuro, los usuarios dejaron los comentarios específicos sobre el sistema y posibles mejoras sobre la usabilidad del mismo. Estos se resumen en los siguientes puntos (diferenciando entre los dos grupos):

- ✓ Usuarios del grupo de personas de edad avanzada:
 - No intuitivo, aplicar lógica de aparatos electrónicos comúnmente conocidos, como televisiones, ordenadores..., etc.
 - Añadir imágenes explicativas en los mensajes de información, que le darían un mayor realismo.
 - Aumento del tamaño del texto de los botones de comandos. Ya que en muchas ocasiones tenían que acercarse al dispositivo, para poder comprender exactamente lo que quería decir ese comando.

- ✓ Usuarios del grupo joven:
 - No se da la opción de no escuchar el mensaje de voz, salta automáticamente.
 - No se da la opción de cambiar la voz del mensaje de voz.
 - No se da la opción de cambiar el contraste (cambiar fuente, color, etc.).
 - Recomendado empleo de sistema de ayuda, con el que el usuario pueda saber en todo momento que hacer para navegar en la aplicación.



- Dar la opción de elegir la ubicación del usuario en recibir las alertas. Es decir, si un usuario por diferentes motivos tiene que trasladar su actual ubicación a otra comunidad de España o país, se le dé la opción de elegir la ubicación exacta de las emergencias que ocurran por esas zonas que desea recibir por su dispositivo móvil. Además el usuario recomendó la utilización de un mapa GIS (*Geographic Information System*).

Por último y en cuanto a los mensajes de información, todos los usuarios de los dos grupos, reflejaron el acierto de presentar la información en diferentes puntos con la información concreta, sin el empleo de interminables párrafos que dejarían a un usuario de leer más sobre el tema. También expresaron la ayuda que supondría el empleo de la conocida frase *¿Está seguro que de que quieres salir?*, antes de aprobar la salida del sistema mediante el empleo del comando salir. Esto ayudaría al usuario a no salir de la aplicación, al equivocarse éste por torpeza o nerviosismo pulsar el comando salir.



8. ORGANIZACIÓN DEL PROYECTO

La organización del proyecto es una parte muy importante de la planificación y programación del proyecto. Se trata de uno de los factores más importante en el desarrollo de un proyecto informático. Todo proyecto está compuesto de tareas o de actividades. Para que el proyecto tenga éxito, es necesario planificar con cuidado las tareas y luego organizar éstas en orden de prioridad, según sea más importante o menos su ejecución.

El diagrama de GANTT es una herramienta que le permite al usuario modelar la planificación de las tareas necesarias para la realización de un proyecto. La lectura de los diagramas de GANTT es relativamente sencilla, esta herramienta es utilizada en numerosos proyectos en todos los sectores del área de desarrollo. El diagrama de GANTT permite realizar una representación gráfica del progreso del proyecto y además es un buen medio de comunicación entre las diversas personas involucradas en un proyecto.

En este proyecto se han realizado dos diagramas de GANTT. El primero se realizó al comienzo del proyecto, para planificar el tiempo esperado en la realización de cada una de las tareas del proyecto. El segundo diagrama de GANTT ha sido realizado con la planificación real del proyecto, es decir, un diagrama que contiene los tiempos reales de duración de cada tarea.

Una vez vistos los dos diagramas de Gantt, se realizará un estudio acerca de lo que pasó con la planificación final. En este apartado se comenta si se siguió escrupulosamente el plan original, si hubo retrasos, a que fueron debidos y diferentes tipos de preguntas sobre la planificación real.

Por último se realiza un análisis económico del proyecto, que contiene todos los gastos asociados al proyecto y ofrece un presupuesto inicial con el coste total del proyecto.



8.1.PLANIFICACIÓN INICIAL

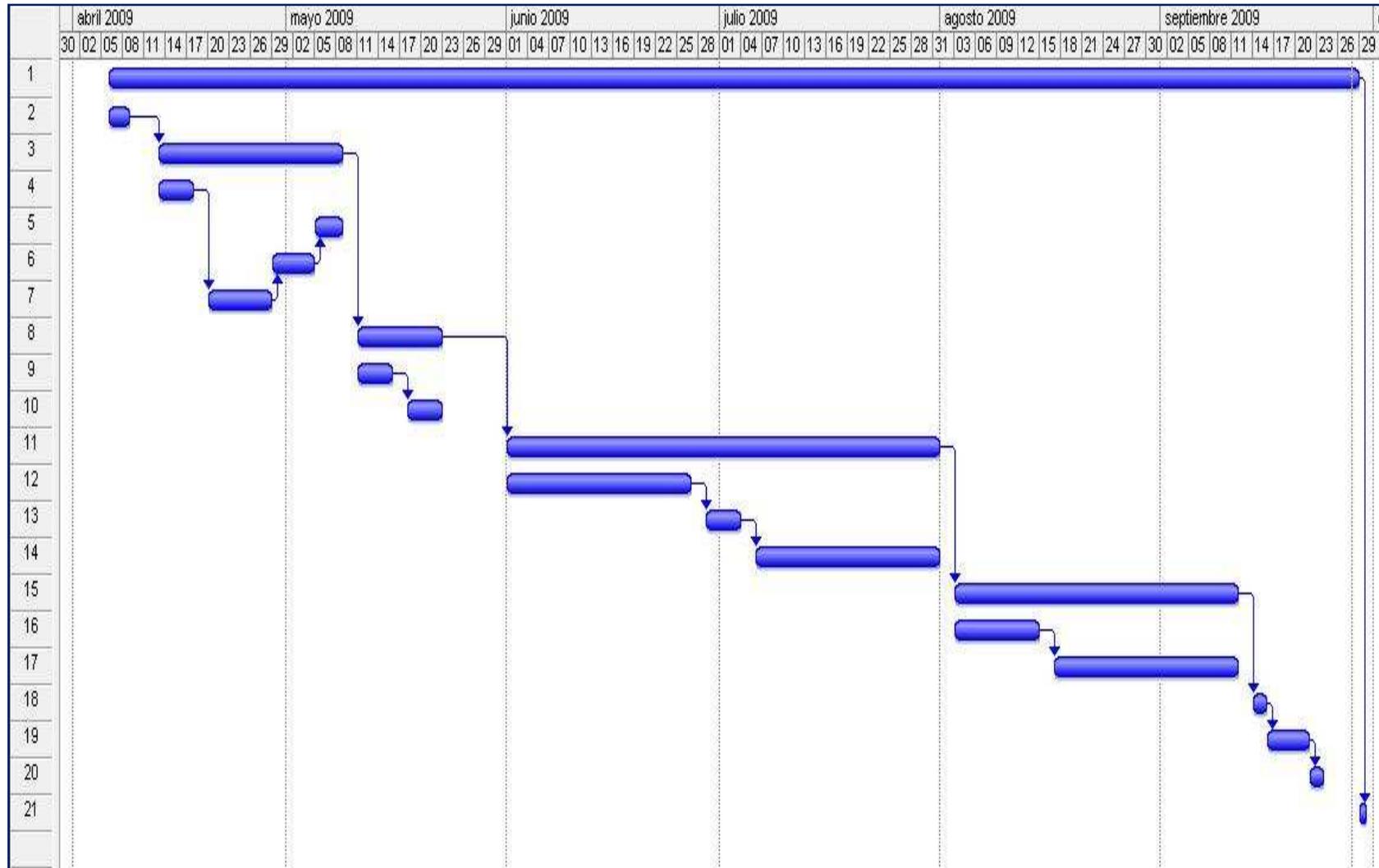
Tareas del proyecto con la duración de tiempo estimada

	Nombre de tarea	Duración	Comienzo
1	Sistema Multimedia de Informacion	126 días	lun 06/04/09
2	- Introduccion del Proyecto	3 días	lun 06/04/09
3	- Estado del Arte	20 días	lun 13/04/09
4	<i>Lectura sobre temas del proyecto</i>	5 días	lun 13/04/09
5	<i>Marco histórico</i>	4 días	mar 05/05/09
6	<i>Diseño dispositivos móviles</i>	4 días	mié 29/04/09
7	<i>Diseño para Personas Mayores</i>	7 días	lun 20/04/09
8	- Descripción del Problema	10 días	lun 11/05/09
9	<i>Problemas que se nos plantean</i>	5 días	lun 11/05/09
10	<i>Aproximación de una posible solución</i>	5 días	lun 18/05/09
11	- Análisis y Diseño del Sistema	45 días	lun 01/06/09
12	<i>Análisis y Especificación de Requisitos</i>	20 días	lun 01/06/09
13	<i>Modelo Conceptual de la Aplicación</i>	5 días	lun 29/06/09
14	<i>Diseño de la Aplicación</i>	20 días	lun 06/07/09
15	- Implementación	30 días	lun 03/08/09
16	<i>Iniciación en lenguaje de Programación</i>	10 días	lun 03/08/09
17	<i>Implementación del sistema</i>	20 días	lun 17/08/09
18	- Organización de proyecto	2 días	lun 14/09/09
19	- Evaluación	4 días	mié 16/09/09
20	- Conclusiones	2 días	mar 22/09/09
21	- Presentación del Proyecto	1 día	mar 29/09/09

Ilustración 14. Tareas organizadas al principio del proyecto



Diagrama de Gantt Inicial





8.2.PLANIFICACIÓN REAL

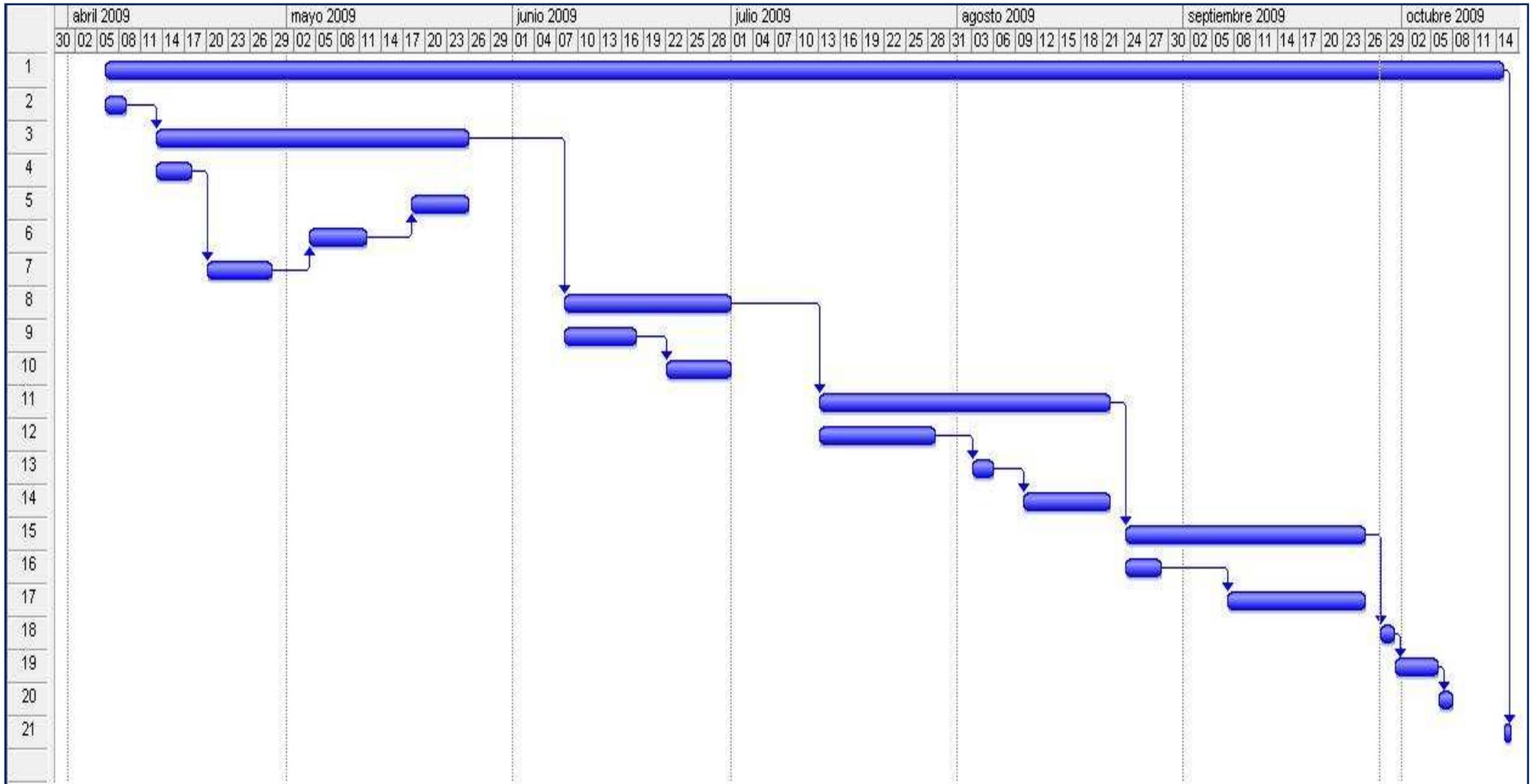
Tareas del proyecto con la duración de tiempo real

	Nombre de tarea	Duración	Comienzo
1	Sistema Multimedia de Informacion	138 días	lun 06/04/09
2	- Introduccion del Proyecto	3 días	lun 06/04/09
3	- Estado del Arte	31 días	lun 13/04/09
4	<i>Lectura sobre temas del proyecto</i>	5 días	lun 13/04/09
5	<i>Marco histórico</i>	6 días	lun 18/05/09
6	<i>Diseño dispositivos móviles</i>	6 días	lun 04/05/09
7	<i>Diseño para Personas Mayores</i>	7 días	lun 20/04/09
8	- Descripción del Problema	17 días	lun 08/06/09
9	<i>Problemas que se nos plantean</i>	8 días	lun 08/06/09
10	<i>Aproximación de una posible solución</i>	7 días	lun 22/06/09
11	- Análisis y Diseño del Sistema	30 días	lun 13/07/09
12	<i>Análisis y Especificación de Requisitos</i>	12 días	lun 13/07/09
13	<i>Modelo Conceptual de la Aplicación</i>	3 días	lun 03/08/09
14	<i>Diseño de la Aplicación</i>	10 días	lun 10/08/09
15	- Implementación	25 días	lun 24/08/09
16	<i>Iniciación en lenguaje de Programación</i>	5 días	lun 24/08/09
17	<i>Implementación del sistema</i>	15 días	lun 07/09/09
18	- Organización de proyecto	2 días	lun 28/09/09
19	- Evaluación	4 días	mié 30/09/09
20	- Conclusiones	2 días	mar 06/10/09
21	- Presentación del Proyecto	1 día	vie 16/10/09

Ilustración 15. Tareas organizadas al final del proyecto



Diagrama de Gantt Final





Como se puede observar comparando la planificación que se realizó al comienzo del proyecto y la planificación real realizada al acabar el mismo, no se ha seguido escrupulosamente el plan original, pero aunque ha habido un desfase en cuanto a la fecha de finalización del proyecto de dos semanas, este retardo no se puede considerar muy grande y se puede decir que no se ha realizado una mala organización del proyecto.

Debido a los exámenes de Junio y Septiembre, se han perdido dos semanas de trabajo que en el plan inicial no se tuvo en cuenta, lo que hace que se haya tenido que retrasar la fecha de presentación del proyecto dos semanas como se ha comentado anteriormente. En cuanto a los retrasos y adelantos de las diferentes tareas programadas para la realización del proyecto, el capítulo de estado del arte se ha retrasado más tiempo del que se esperaba inicialmente, ya que toda la recogida de documentación y el hecho que gran parte de esta estuviera en inglés, lo que suponía una pequeña complejidad añadida, ha llevado al retraso de esta tarea. Por el contrario en cuanto al capítulo de análisis y diseño del sistema, se planificó que duraría un número elevado de días y finalmente se ha podido comprobar, que en menos días de lo que se esperaba, se ha podido realizar este apartado. Por último en cuanto al apartado de la implementación, más o menos se ha empleado el tiempo que se estipuló al comienzo, por lo que no ha variado en gran cantidad de días su planificación inicial.

Por otro lado, se ha seguido exactamente el orden de las actividades a realizar, ya que era un requisito esencial al inicio del proyecto, si se quería terminar en un tiempo razonable y no dejar ningún detalle sin realizar. Por tanto lo único que se ha modificado de la planificación inicial a la planificación real, ha sido el tiempo dedicado a estas actividades y el hecho de no contemplar el periodo de exámenes del alumno, tanto en Junio como en Septiembre, en el cual no ha sido posible dedicarle tiempo a la realización del proyecto y ha provocado el retraso de dos semanas de la finalización del mismo.



8.3. PRESUPUESTO – ANÁLISIS ECONÓMICO

Dentro de este apartado se van a especificar los costes que producirá el proceso de desarrollo del proyecto. Los gastos que se tendrán en cuenta son: de personal y de material, dentro de este tipo se distinguen tres subtipos: hardware, software y material fungible. En primer lugar se van a analizar los gastos producidos por el equipo de trabajo. Para este tipo de recurso se muestra el coste que supone una hora de trabajo por cada puesto de trabajo en la siguiente tabla.

Categoría	Coste/hora
Jefe de Proyecto	60 €
Analista	50 €

Tabla 41. Coste / hora por categorías

Conocidos los gastos asociados por cada rol, se procede al detalle con el coste total de cada categoría. Se debe aclarar primero que de la categoría de analistas se tienen dos miembros y solamente existe un jefe de proyecto. En la siguiente tabla se muestra este detalle, teniendo en cuenta que la duración del proyecto, como se estipula en la planificación inicial en el apartado 8.1 de este documento, se tienen 25 semanas de trabajo, es decir 125 días laborables y se supone que para una jornada laboral se emplea como máximo 8 horas.

	Jefe de Proyecto	Analista
Precio por hora	60 €	50 €
Semanas de trabajo	25 semanas	25 semanas
Días laborales	125 días	125 días
Horas totales	1000 horas	1000 horas
TOTAL	60.000 €	50.000 €

Tabla 42. Costes de personal

El coste total derivado de la contratación de personal, asciende a:

160.000,00 €

Una vez calculado el coste de personal, se debe calcular el coste de material del desarrollo del proyecto, en las siguientes tablas se incluyen todos los recursos materiales empleados con el coste de cada uno de ellos. Estos costes son de distintos tipos:



hardware, *software* y material fungible. A continuación, mostramos los costes derivados de la compra de equipos hardware y software para el desarrollo del proyecto.

	Coste/Unidad	Cantidad	Coste Total
Ordenador de sobremesa	1100 €	3	3300 €
1 Dispositivo móvil	350 €	1	350 €

Tabla 43. Detalle de materiales

Los recursos software utilizados dentro del desarrollo del proyecto para el despliegue de la aplicación, son *open – source* y por tanto no tiene coste imputable al proyecto, asique el único coste asociado son los equipos de ordenador que tendrán los 3 miembros del equipo, y el dispositivo móvil donde se realizará la implantación del sistema y las pruebas pertinentes. Antes de proceder a la parte del riesgo y el beneficio aplicado al coste total, se debe detallar el coste derivado de la adquisición de material fungible. Dicho material es aquel que se compra y gasta según va avanzando el proyecto.

	Coste/Semana	Total
Folios	5,00 €	85,00 €
CD's, cartuchos...	15,00 €	255,00 €
Material de oficina (bolígrafos, tipex...)	20,00 €	340,00 €

Tabla 44. Coste derivado del material fungible

El coste total derivado de la adquisición de material a lo largo del proyecto, asciende a:

4330,00 €

Por último, se va a detallar el coste proporcional al riesgo aplicado al proyecto y el beneficio aplicado al mismo. En este caso, se aplica un 20% de riesgo y un 15% de beneficios. Antes de detallar el coste, se debe indicar el coste total sin el riesgo y los beneficios aplicados:

164.330,00 €

Riesgo 20%	Beneficio 15%
32.866,00 €	24.649,5 €

Tabla 45. Coste asociado al beneficio y al riesgo



Para concluir con el presupuesto, se indica el coste total, con el 16% de IVA, que supondrá el desarrollo de este proyecto para la notificación de emergencias hacia las personas mayores. En la siguiente tabla se ilustra el coste asociado.

Presupuesto Final	
Concepto	Coste
Costes totales	164.330,00 €
Riesgo (20%)	32.866,00 €
Beneficio (15%)	24.649,50 €
Total (sin IVA)	221.854,50 €
I.V.A. (16%)	35.495,28 €
TOTAL	257.349,78 €

Tabla 46. Presupuesto final

El proceso a embolsar por el cliente (sin IVA) será de:

221.854,50 €

(Doscientos veintiún mil ochocientos cincuenta y cuatro euros con cincuenta céntimos de euro)

A este precio es necesario aplicarle el IVA (base imponible: 16%), con lo cual obtenemos el precio final para el cliente de:

257.349,78 €

(Doscientos cincuenta y siete mil trescientos cuarenta y nueve euros con setenta y ocho céntimos de euro)



9. CONCLUSIONES

Al comienzo del proyecto el conocimiento que tenía sobre el tema de las diferentes discapacidades o problemas que tenían las personas mayores era muy básico, así como el desarrollo de sistemas orientados a las personas mayores, por lo que suponía un reto para mí enfrentarme a este tema. A medida que fue pasando el tiempo desde que empecé el proyecto, fui adquiriendo un gran conocimiento sobre este tema a través de artículos, tesis relacionadas con las personas mayores y otros proyectos que se habían realizado, que despertó un gran interés en mí realizar un proyecto muy bien documentado sobre el tema.

El motivo principal por lo que quise realizar este proyecto, además de por qué ya conocía al tutor de una asignatura de la carrera, fue realmente introducirme en el desarrollo de sistemas para dispositivos móviles, un tema que desconocía del todo y que actualmente en el mercado está en un gran crecimiento. Esta es una de las principales razones por la que una vez finalizado este proyecto, esté muy satisfecho de conocer las técnicas de diseño hacia dispositivos móviles y conocer en profundidad la tecnología de desarrollo J2ME, la cual desconocía al principio del proyecto y ahora puedo decir que conozco al detalle su API.

En cuanto al tema de notificación de emergencias, es un tema que requiere de mucho cuidado y por tanto es necesaria una profunda documentación sobre él. El desarrollo de sistemas de emergencias para usuarios es un tema muy interesante, que ha supuesto en mi una expansión de conocimientos sobre su empleo, que lo valoro positivamente. La importancia y responsabilidad que supone la realización de un proyecto para ayudar a usuarios, en este caso personas mayores, en situaciones delicadas supone un gran reto.

El objetivo que me propuse al embarcarme en este proyecto, fue realizar una aplicación de fácil manejo y gran uso para las personas mayores. Se puede decir que se ha cumplido este objetivo tal y como corroboran los resultados obtenidos en la evaluación, aunque como es obvio hay varios detalles del sistema desarrollado que se deben mejorar o ampliar, pero en términos generales, se ha cumplido con creces las expectativas que puse al comienzo del proyecto. Además con la realización de la evaluación del sistema,



me he dado cuenta, las diferencias sobre el conocimiento de las tecnologías actuales que existen entre las personas jóvenes y mayores. Visto desde fuera parece obvia la diferencia que hay entre estos dos grupos, pero hasta que no profundizas en el tema y realizas una evaluación como la que se hizo en su momento, no puedes asegurar con absoluta rotundidad o certeza la existencia de esta diferencia tecnológica en la sociedad actual.

Para finalizar las conclusiones que se han extraído con la realización de este proyecto, he de comentar los aspectos ocurridos en cuanto a la organización del mismo. A pesar de haberse retrasado 15 días la finalización del mismo, se ha cumplido el objetivo de acabar nada más acabar el verano, que era el objetivo principal que acordamos el tutor y yo antes de empezar el proyecto. Además con el seguimiento de la planificación inicial realizada al comienzo del proyecto, me he dado cuenta de lo importante que es llevar una buena organización en este tipo de proyectos, porque de lo contrario, su finalización seguramente se hubiese retrasado mucho más y no hubiera podido completar todos los detalles que presentaba la realización de este proyecto.

También hay que destacar que a pesar de que en los últimos cursos de la carrera, sobre todo en quinto, se nos enseña a redactar la documentación de un sistema informático, ha sido con la realización de este proyecto con el que verdaderamente he aprendido a redactar una memoria. Por esta razón he de aportar que para realizar un buen proyecto, es importantísima una profunda documentación del mismo, y además es tan importante la implementación del mismo como una buena memoria.



10. AMPLIACIONES Y LÍNEAS FUTURAS

Este proyecto puede ser si se sigue trabajando con él, un posible proyecto de gran alcance, ya que la idea de que la notificación de emergencias sucedidas en una determinada zona pueda llegar a una gran multitud de personas por medio de un simple mensaje al móvil, y además se les presente una serie de recomendaciones que deben de tomar en caso de que suceda esa situación de emergencia, es un proyecto muy interesante y que obtendría numerosos beneficios.

Pero para que este proyecto pudiese hacerse realidad y extenderse a toda la población, se deben realizar una serie de mejoras (algunas de ellas han sido sugeridas por las personas a las que se le realizó la evaluación del sistema). Las posibles mejoras que se podría plantear son las siguientes:

- La aplicación podría exportarse a otras plataformas de desarrollo para dispositivos móviles que no sea J2ME y así poder tener un mayor mercado de comercialización. En el apartado “6.1 Herramientas de desarrollo” de este documento, se hace una referencia a las plataformas de desarrollo para dispositivos móviles más importantes en la actualidad, por tanto una posible mejora sería, desarrollar esta aplicación para estas plataformas.
- En cuanto al software del sistema, podría aumentarse el número de emergencias, ya que las emergencias que ocurren hoy en día, son más de las que este sistema ofrece al usuario. Una emergencia que no están contemplada en el sistema y me di cuenta al final del proyecto, son los accidentes de tráfico, que actualmente sucede una cantidad elevada diariamente. El trabajo de aumentar el número de emergencias, no supondría una labor muy compleja.
- Si se quisiese llevar esta aplicación a la vida real, se debería de implementar un módulo de conexión por red, con algún organismo o institución que enviará información de las emergencias. Mediante este módulo en vez de iniciarse el sistema por aleatoriedad como se hace en el prototipo de este proyecto, las autoridades mandaría por red estos mensajes y a los usuarios les llegaría a sus dispositivos móviles inmediatamente después.



- Habría también que pensar, si es la mejor opción que este en sistema se desarrollará mediante una aplicación web. Es decir, si sucediese una catástrofe como un terremoto o inundación, que hiciese que el edificio donde está alojado el servidor del sistema sufriera daños que pudiese hacer que el servidor resultase dañado o inoperativo por este motivo, no se podría avisar a los usuarios de esa emergencia. Por tanto habría que determinar si es más eficaz una aplicación web o una aplicación diferente. Este problema también se podría subsanar, empleando varios servidores de información en otros lados.

- Además, todas las opiniones por los usuarios evaluados recogidas en el apartado “7. Evaluación” se tendrían en cuenta como mejoras para aumentar el potencial de esta aplicación móvil. Sobre todo la idea de introducir un mapa *GIS* para seleccionar la ubicación de la zona donde quiere el usuario que le llegué la notificación de emergencias de esa ubicación, sería una ampliación muy interesante.

- Por último y dado que no se ha podido tener la oportunidad, en este proyecto podría haberse realizado un módulo de accesibilidad para las personas discapacitadas y haber podido evaluar a alguna persona con discapacidad. Esta es otras de las posibles mejoras que se quedarían pendientes para un futuro



11. REFERENCIAS

- [1] WEISER, Mark. "The Computer for the 21st Century", Sci. Amer., Sept. 1991
- [2] WEISER, M. "Some Computer Issues in Ubiquitous Computing", CACM. Julio 1993.
- [3] SAINZ SALCES, F.J. "Pattern design for the development of ubiquitous emergency notification systems. In HCI and the Older Population". 2008.
- [4] KALACHE, A; LUNENFELD, B. "Health and the ageing male". 2001.
- [5] HAWTHORN, D. "Possible implications of aging for interface designers. Interacting with Computers". 2000.
- [6] INCLUDE. (2002). Inclusion of Disabled and Elderly people in telematics. Retrieved 2 December 2002, from <http://www.stakes.fi/include/>
- [7] FOZARD, J. L.; GORDON-SALANT, S.; SCHEIBER, F.; WEIFFENBACH, J. M. "Sensory and perceptual considerations in designing environments for the elderly". Paper presented at the Life-Span Design for Residential. 1993
- [8] LYSONS, K. "Understanding hearing loss. Jessica Kingsley". 2006.
- [9] SATHIAN, K. "Tactile spatial acuity and roughness discrimination: impairments due to aging and Parkinson's disease. Neurology". 1997
- [9] VENKATESH, A. "Computers and other interactive technologies for the home". 1996.
- [10] IFUKUBE, T. "A neuroscience-based design of intelligent tools for the elderly and disabled". Proceedings of the WUAUC'01, Alcacer do Sal. 2001
- [11] HANSON, V. L. "Web ACcess for elderly citizens". Proceedings of the WUAUC'01, Alcacer do Sal, Portugal. 2001



- [12] MANN, W. C. “Effectiveness of assistive technology and environmental interventions in maintaining independence and reducing home care cost for the frail elderly”.1999.
- [13] ROSADO, Eva. Artículo sobre las personas mayores y las tecnologías actuales.
<http://www.consumer.es/web/es/tecnologia/internet/2007/04/19/161812.php>
- [14] FAULKNER, Christine, “The essence of human-computer interaction”. Prentice Hall, 1998.
- [15] CARROL, J.M. “Usability Engineering: scenario-based development of HCI”. 2002.
- [17] NIELSEN, Jacob. <http://www.useit.com/>. Jacob Nielsen´s Website. 2003.
- [18] BROWN, Hilary. “Accessibility and Usability of Information Technology by the Elderly”. 2000.
- [19] COLADO RODRIGUEZ, Cesar. “Diseño y desarrollo de aplicaciones web Multidispositivo”. Germinius XXI. Febrero 2003.
- [20] MUÑOZ MARQUEZ, Luis David. “Las personas mayores ante las tecnologías de la información y la comunicación. Estudio Valorativo”. 2002.
- [21] MARCUS, Aaron. “Universal, Ubiquitous, esuer-Interface Design for the Disabled and Elderly”. 2003
- [22] KNUTSEN Lars; CONSTANTIOU, Johanna; DAMSGAARD, Jan. “The Four Incremental Steps Toward ADVANCED MOBILE SERVICE ADOTION: Exploring mobile device user adoption patterns and market segmentation”. 2007.
- [23] BENFORD, Steve; LJUNGRASTAND, Peter. “Brifging the PHYSICAL AND DIGITAL in pervasive Gaming”. 2005.
- [24] DEKLEVA, Sasha; KNOERZER, Geofrey. “Evolution and Emerging Issues in Mobile Wireless Networks”. 2007.



[25] RENAUD, Kareb; VAN BIJON, Judy. “Predicting Technology Acceptance and Adoption by the elderly: A Qualitative study”. 2008.

[26] NEHMER, Jurgen; BECKER, Martin; KARSHMER, Arthur; LAHM, RoseMarie. “Living Assistance Systems – An Ambient Intelligence Approach”. 2006.

[27] BRIGEIRO, Mauro. “Envejecimiento exitoso y Tercera Edad: Problemas y retos para la promoción de la salud”.

[28] W3C World Wide Web Consortium. <http://www.w3c.es/>

[29] DEPTO. INGENIERÍA TELEMÁTICA. “Multimedia en MIDP”. Universidad Carlos III de Madrid.

[30] Desarrollo de aplicaciones para dispositivos móviles utilizando J2ME. Universidad de Guadalajara.

[31] Dirección General de Protección Civil. <http://www.proteccioncivil.org/es/index.html>

[32] TELEFÓNICA. “Las telecomunicaciones y la movilidad en la sociedad de la información”. Capitulo 17: Las interfaces Multimodales

[33] Aplicaciones para móviles con JavaME. <http://blog.wixel.es/2008/09/aplicaciones-para-moviles-con-javame/#more-10>

[34] Tutorial: herramientas de programación JAVA de dispositivos móviles. <http://leo.ugr.es/J2ME/TOOLS/index.html>

[35] J2ME Standard Edition. <http://books.google.es/books?vid=ISBN8468847046>

[36] GALVEZ ROJAS, Sergio. Java a tope: Java 2 Micro Edition. Universidad de Málaga.

[37] CALERO MONTEAGUDO, Antonio. “Tecnologías móviles con Java”. Revista del Instituto Tecnológico de Informática.

[38] MORALES INDIAS, ALEJANDRO. “Implementación de un Sistema de Notificación para Dispositivos Móviles orientado a usuarios de la Tercera Edad con uso de Multimedia”. Proyecto Fin de Carrera. Universidad Carlos III de Madrid. 2009



[39] GUERRERO CORDOBA, Maria Luz. “Sistema avanzado para la gestión de información utilizando dispositivos móviles de Bolsillo”. Proyecto Fin de Carrera. Universidad Carlos III de Madrid. 2005

[40] ARANAZ TUDELA, Jaime. “Desarrollo de aplicaciones para dispositivos móviles sobre la plataforma android de google”. Proyecto Fin de Carrera. Universidad Carlos III de Madrid. 2009



12. APÉNDICES

A. MANUAL DE USUARIO

En este manual de usuario se procederá a realizar una explicación de la utilización del sistema, que servirá al usuario como guía en su afán de interactuar con el sistema. Como se explicó en el modelo conceptual del sistema, la aplicación puede arrancar de dos modos:

- Si ha ocurrido ninguna emergencia, el sistema arrancará en Modo Alerta.
- Si por el contrario, no ha ocurrido ninguna emergencia, el usuario entrará al sistema en Modo Normal.

En la vida real si ocurriera una emergencia se le enviaría al usuario por red la notificación de dicha emergencia y el sistema arrancaría en modo alerta. Si no hubiera ninguna emergencia y el usuario quisiese entrar en el sistema, entraría en modo normal.

El hecho de que en este proyecto no tenemos en cuenta los avisos por la red, esta función se ha implementado de tal forma que se inicie en modo normal o en modo alerta de forma aleatoria. Un 75% de las veces iniciará en modo alerta y un 25% de las veces en normal. Cuando se inicie en modo alerta, habrá una probabilidad equitativa de que el evento que se le notifica al usuario como emergencia sea cualquiera de los 5 tipos de emergencias.

En primer lugar se detallará un ejemplo de navegación por el sistema, si este arrancará en modo normal. Una vez explicado este modo, se procederá a la explicación del modo alerta de igual forma. En cada pantalla se detallará textualmente lo que hace la interfaz de usuario correspondiente, así como la función de los diferentes comandos que en ella hay. Además se mostrará mediante una imagen la pantalla de cada paso de la navegación del sistema, para dar una idea mucho más clara de la aplicación.

A.1 MODO NORMAL

INTERFAZ INICIAL



Ilustración 16. Interfaz Modo Normal

La ilustración 16 muestra la pantalla del modo normal. Esta pantalla será la inicial en caso de que no haya sucedido una emergencia y únicamente el usuario, quiera entrar en el sistema para obtener información sobre los diferentes tipos de emergencias y como actuar en caso de que estos sucedan. En el medio de la pantalla se puede ver el logo del sistema “Sistema de Alertas”. En esta pantalla se puede observar la presencia de dos comandos:

- Salir: Cierra la aplicación.
- Entrar: Entra en la aplicación e irá a la interfaz de información sobre los posibles casos de emergencias que existen en este sistema.

INTERFAZ DE INFORMACION



Ilustración 17. Interfaz de Información

La Ilustración 17 muestra la interfaz de la información. En esta interfaz aparecen los 5 tipos de emergencia. Pulsando sobre uno de ellos se accederá primero a una pantalla que muestra una imagen del tipo de emergencia y a los 2 segundos se irá a una pantalla que mostrará las recomendaciones propias del suceso seleccionado. El tamaño de los diferentes elementos de la lista es el máximo posible. Lo recomendable sería que ocupase toda la pantalla y su distribución fuese más ecuánime, pero no se puede hacer el tamaño de la letra más grande, ya que eso forma parte de la configuración del dispositivo móvil en donde se instale la aplicación. Además en esta interfaz se puede apreciar la presencia de dos comandos:

- **Atrás:** Vuelve hacia la pantalla anterior (Ilustración 16).

- Opciones: Dirige al usuario hacia la pantalla de opciones, donde se podrá definir el perfil o la configuración de usuario preferida, eligiendo el tamaño del texto de los mensajes de información y el volumen de los mensajes de voz.

INTERFAZ DE OPCIONES



Ilustración 18. Interfaz de opciones

La Ilustración 18 muestra la interfaz de opciones. En esta pantalla el usuario podrá configurar a su gusto las diferentes opciones del sistema como son: el tamaño del texto de los mensajes de información y el volumen de los mensajes de voz. Esta pantalla solo puede ser accedida desde la interfaz de información. Tiene un comando:

- Aceptar: Almacena los valores configurados por el usuario y vuelve a la pantalla de información (Ilustración 17)

INTERFAZ DE IMÁGENES DEL TIPO DE EMERGENCIA

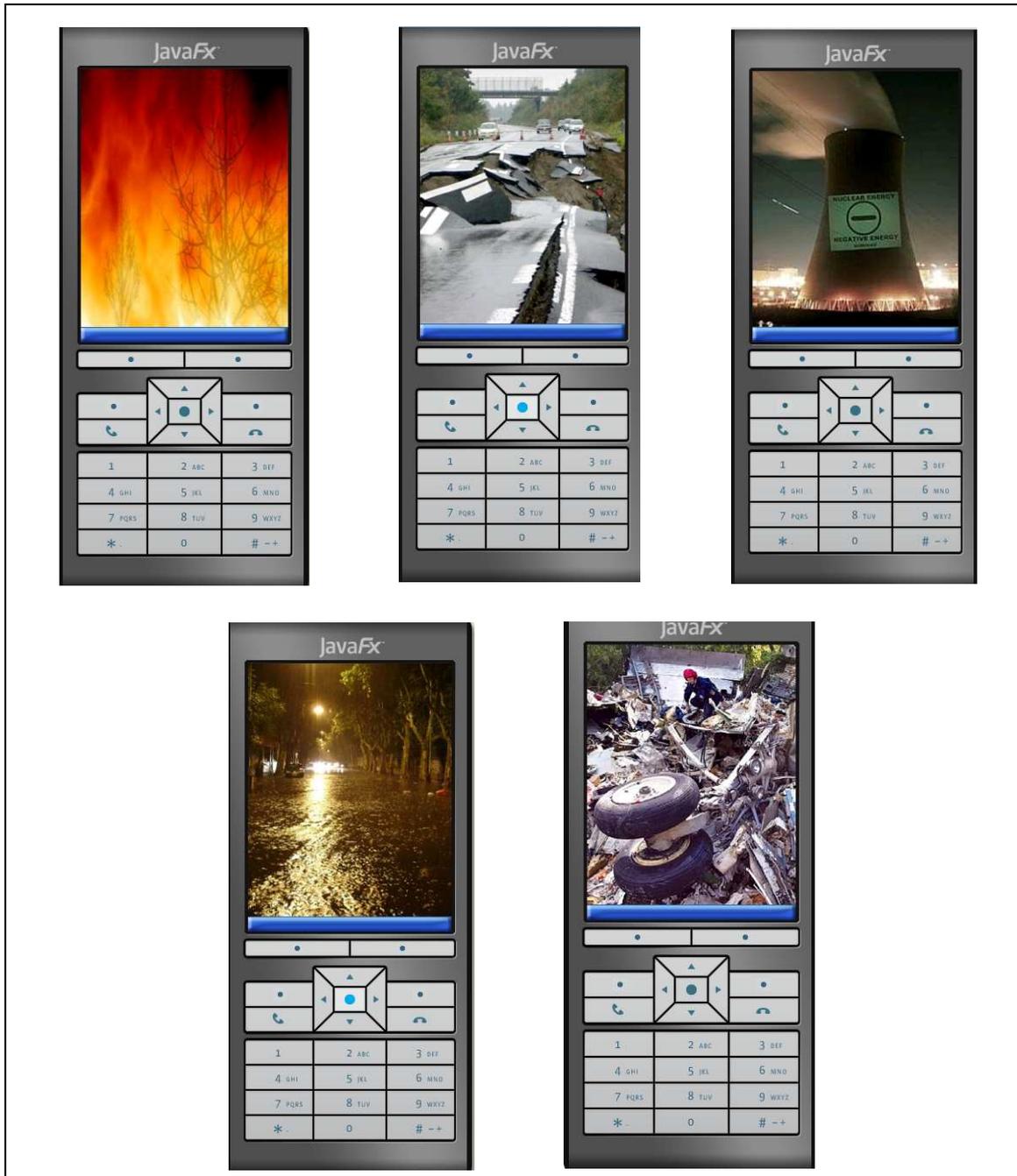


Ilustración 19. Interfaces gráficas de emergencias

Las Ilustración 19 muestra las interfaces intermedias entre la pantalla de información de las diferentes emergencias y las recomendaciones de cada una de estas en caso de ocurrir tal suceso. Estas pantallas permiten mostrar al usuario un aspecto grafico del tipo de incidencia. A los 2 segundos de mostrar una la imagen del suceso ocurrido, se redireccionará automáticamente hacia la pantalla de recomendaciones.

INTERFAZ DE RECOMENDACIONES



Ilustración 20. Interfaz de recomendaciones

La Ilustración 20 muestra la pantalla de recomendaciones. En esta pantalla se avisará al usuario de “que” hacer en caso de que suceda la emergencia. En este caso la emergencia sucedida es un incendio. Como se puede apreciar, las recomendaciones se muestran divididas en puntos y a su vez están subdivididas en secciones, dependiendo de la situación de la persona. Además se podrán escuchar estas recomendaciones mediante un mensaje grabado de voz. Todas estas recomendaciones han sido recopiladas de la web de Protección Civil, como se explica en uno de los requisitos de sistema. Esta pantalla tiene también un comando:

- Atrás: Vuelve a la interfaz de información (Ilustración 17).

A.2 MODO ALERTA

INTERFAZ DE ALERTA



Ilustración 21. Interfaz del modo alerta

La ilustración 21 muestra la pantalla del modo alerta, donde se avisa al usuario de que ha sucedido una emergencia, indicándole el tipo (Texto debajo de la imagen de alerta), mostrándole una señal de peligro que alerte al usuario del peligro que puede correr y un mensaje de voz grabado avisándole del evento ocurrido. Se tienen dos comandos:

- **Salir:** Cierra la aplicación, en caso de que el usuario conozca ya las indicaciones oportunas o decida no saber más información sobre la emergencia sucedida. Este comando ha sido puesto, para darle la oportunidad al usuario de no tener que acceder a la pantalla de recomendaciones y darle la posibilidad de no tener que acceder obligatoriamente a la pantalla los protocolos de actuación en este tipo de

emergencias, si por algún motivo propio no desease obtener esa información (porque ya la conoce, por vaguería..., etc.)

- Entrar: Si se pulsa entrar, el sistema irá a la pantalla con toda la información del suceso. Previamente se pasará por la imagen del tipo de emergencia, y a los 2 segundos se accederá a la pantalla del suceso.

INTERFAZ DEL SUCESO

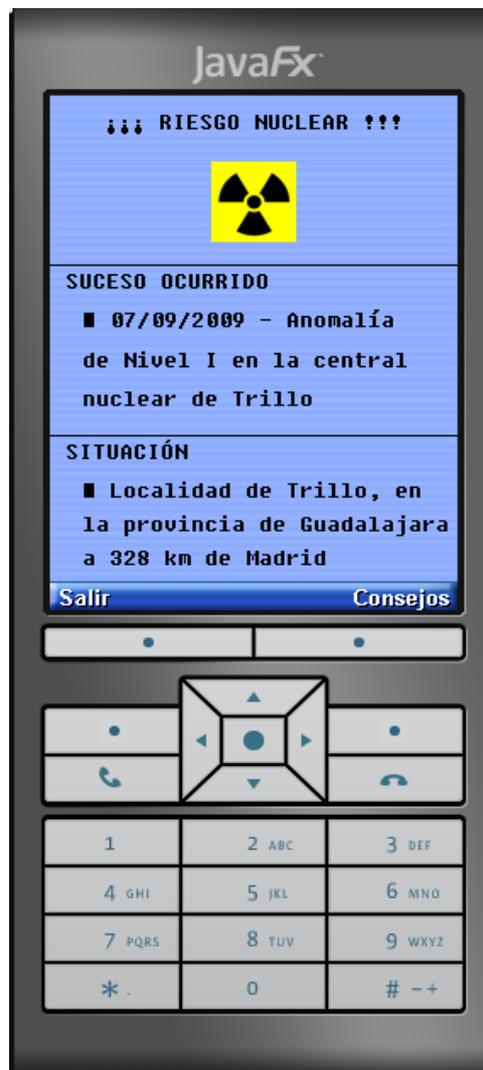


Ilustración 22. Interfaz de información de suceso



La ilustración 22 muestra la pantalla del suceso ocurrido, donde se avisa al usuario de que de la información exacta de la emergencia y la situación geográfica de donde ha sucedido. Además se le muestra un icono para ayudar al usuario a asociar fácilmente la emergencia.

En este sistema la ubicación del usuario es muy importante, ya que el organismo que manda los mensajes de información al usuario, le alerta de las emergencias que suceden cerca de donde el usuario vive. Por tanto el mensaje con la información del suceso, contendrá la fecha del suceso de la emergencia y la ubicación exacta de donde ha sucedido.

Como se observa en la ilustración 22, se tienen dos comandos:

- Salir: Cierra la aplicación, en caso de que el usuario conozca ya las indicaciones oportunas o decida no saberlas. Este comando ha sido puesto, para darle la oportunidad al usuario de no tener que acceder a la pantalla de recomendaciones y de igual manera que en el modo normal, darle la posibilidad de no tener que acceder obligatoriamente a la pantalla los protocolos de actuación en este tipo de emergencias, si por algún motivo propio no desee obtener esa información (porque ya la conoce, por vaguería..., etc.)
- Consejo: Si se pulsa el sistema irá a las recomendaciones que debe tomar el usuario en caso de esa emergencia. En esta pantalla se le indicará al usuario que recomendaciones debe tomar en caso de que suceda la emergencia. Como se explicará a continuación, esta pantalla de recomendaciones es la misma pantalla de recomendaciones del modo normal, aunque difieren en el número y el significado de los comandos que tienen.

INTERFAZ DE RECOMENDACIONES ALERTA

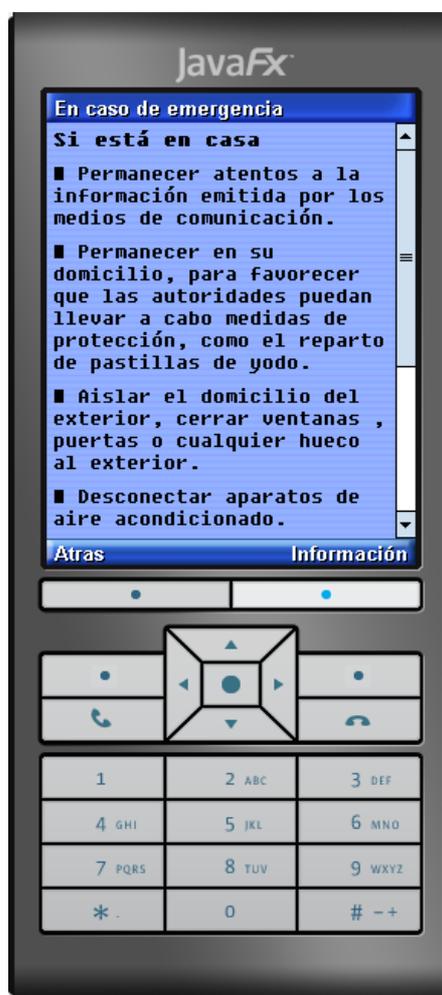


Ilustración 23. Interfaz de recomendaciones del modo alerta

La ilustración 23 muestra la pantalla de recomendaciones. Esta interfaz es exactamente igual que la interfaz de recomendaciones del modo normal (Ilustración 20), se diferencia de ella en los comandos:

- **Atrás:** Volverá a la pantalla de suceso.
- **Información:** Accederá a la pantalla de información, por si el usuario desea conocer más información sobre otro tipo de emergencia.

Además, tanto en esta pantalla de recomendaciones como en la del modo normal, al final de las recomendaciones se ofrece al usuario la dirección web de protección civil y el número de teléfono, por si el usuario quisiese obtener más información sobre la emergencia.



B. PLANTILLA DE EVALUACIÓN

EVALUACIÓN DEL PROYECTO FIN DE CARRERA DE

CARLOS RUANO MARTÍNEZ

Este cuestionario tiene por objeto obtener las opiniones de los participantes en la realización de la evaluación de este proyecto fin de carrera. A continuación se le hará una descripción del funcionamiento del sistema y el objetivo de la realización del mismo. Una vez haya comprendido todo esto, se le ofrecerá una plantilla con una serie de cuestiones para evaluar el sistema mientras usted esta navegando por él.

Este proyecto tiene como objetivo ofrecer al usuario una plataforma para un dispositivo móvil que le alerte mediante el envío de un mensaje a su dispositivo de situaciones de emergencia que ocurran en la zona donde usted reside y le ofrezca una serie de recomendación que usted debe tomar en el caso de que suceda una emergencia. Además de ofrecerle la información textualmente, se le ofrece un módulo de audio al usuario, que mediante mensajes de voz le informará también de la emergencia sucedida.

La navegación por la aplicación se realiza mediante los comandos del emulador del teléfono móvil situados en la parte superior derecha e izquierda. Para la evaluación del sistema se realizarán dos modos de ejecución, uno simulando el envío de una situación de alerta, donde el usuario entrará al sistema en busca de información sobre “qué” hacer en caso del suceso de la emergencia. Y otro modo denominado normal, en el que el usuario entrará a la aplicación con el único objetivo de obtener información sobre las diferentes emergencias que ofrece el sistema. Se le mostrará una pantalla con todos los tipos de emergencias contemplados en el sistema y accediendo a uno de ellos se le mostrará la información correspondiente. En esta pantalla de información usted puede configurar las diferentes opciones del sistema como pueden ser el tamaño de texto de los mensajes de información o el control de volumen de los mensajes de voz.

A continuación se le presenta una serie de cuestiones que usted debe rellenar para poder evaluar correctamente el sistema.



Datos generales de la población.

1. ¿Qué edad tiene usted?
2. Sexo (Hombre o mujer)
3. ¿Con que frecuencia se conecta Ud. a Internet?
 - a. Ninguna frecuencia, no tengo conexión
 - b. No me conecto, aunque tengo conexión
 - c. Varias veces al mes
 - d. Varias veces a la semana

Datos específicos de la aplicación. Responda con M (mal), R (regular) y B (bien).

1. ¿Se comprende para que sirve esta aplicación web móvil (Funcionalidad)?
2. ¿Describe el logo del sistema la funcionalidad correcta de la aplicación?
3. ¿Los colores se distinguen con claridad?
4. ¿El tamaño del texto es el adecuado?
5. ¿Se entiende el icono que acompaña a cada botón?
6. ¿Se entiende la funcionalidad de cada botón?
7. Se puede responder en todo momento la pregunta ¿Dónde me encuentro?
¿Dónde puedo ir?
8. ¿La navegación entre los distintos apartados de la aplicación resulta sencilla?
9. ¿Le resulta sencillo salir de la aplicación?

Datos específicos de los mensajes de información. Responda con M (mal), R (regular) y B (bien).

1. ¿El tipo de fuentes es el apropiado?
2. ¿El tamaño del texto es el adecuado?
3. ¿Le parece correcto la organización de la información (por puntos)?
4. ¿Le parece correcto la división de información según la ubicación del usuario?
5. ¿El lenguaje utilizado es apropiado?
6. ¿Le parece correcto el empleo de mensajes de voz con las recomendaciones?
7. ¿Le resulta cómodo la voz grabada en los mensajes de voz?



C. INSTALACIÓN JAVA 2 SDK

Java 2 SDK (*Software Development Kit*) es un kit de desarrollo de software formado por un conjunto de herramientas útiles, que permiten a los desarrolladores crear aplicaciones escritas en el lenguaje de programación Java y que se ejecutan en la plataforma Java. Estas herramientas están diseñadas para que se utilicen desde la línea de comandos.

Este kit también proporciona también información sobre la documentación de la plataforma Java 2, contiene especificaciones API, descripciones de funciones, guías de desarrollador, páginas de referencia de utilidades y herramientas de SDK, demostraciones y enlaces con información relacionada. Esta documentación también está disponible en un paquete de descarga que puede instalar en su máquina. Además se puede consultar el API de plataforma Java, que proporciona una breve descripción de la API haciendo énfasis en las especificaciones y no tanto en ejemplos de código.

En cuanto a la instalación, lo primero que se tiene que hacer es descargar el kit j2sdk versión 1.5 como mínimo. Desde el siguiente enlace se puede realizar la descarga:

- <http://java.sun.com/javase/downloads/index.jsp>

Una vez descargado el paquete del j2sdk, se hará doble clic en el programa de instalación del j2sdk, en este caso se ha descargado el kit j2sdk versión (1.6.0_16):

- `jdk-6u16-windows-i586`



Ilustración 24. Ejecutable J2SDK

Una vez hecho el doble clic, aparece el cuadro de licencia, se pulsa *Accept*>.

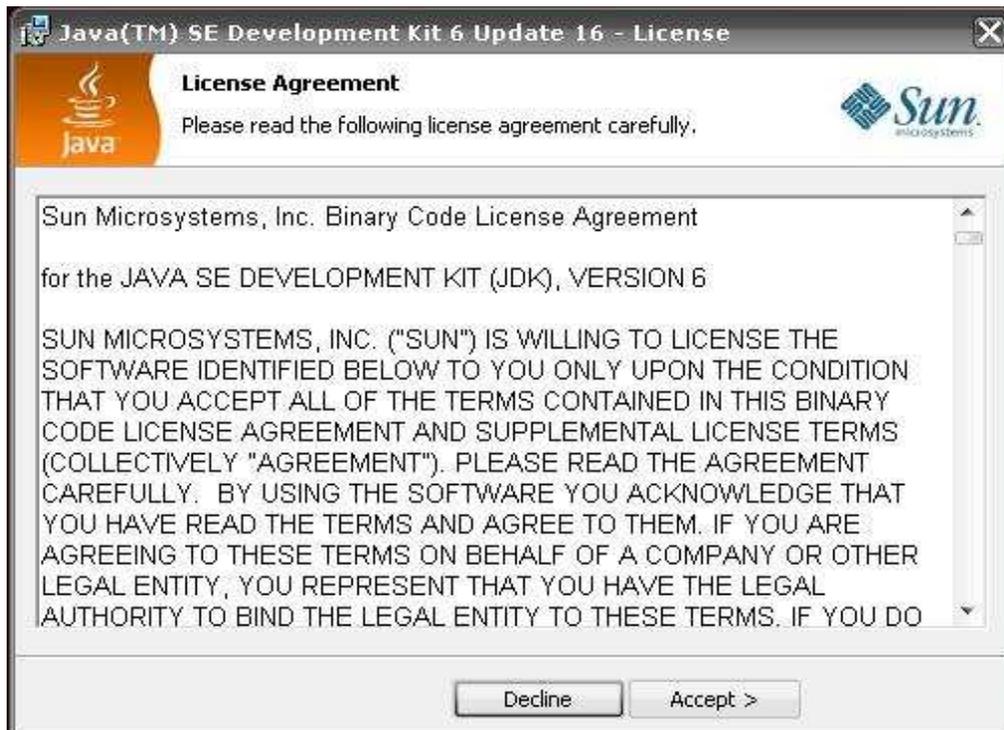


Ilustración 25. Pantalla cuadro de licencia

Posteriormente aparece la siguiente pantalla, se debe pulsar el botón *Next*.



Ilustración 26. Pantalla componentes J2SDK



Una vez seleccionados los componentes (se dejan por defecto), se procede al siguiente paso de la instalación. En este paso se debe esperar a que se copien todos los ficheros en la pantalla de progreso de la instalación. Puede tardar varios minutos.



Ilustración 27. Pantalla de progreso de instalación

Cuando se hayan copiado todos los ficheros, aparecerá la siguiente pantalla, donde se seleccionará la carpeta de destino de la instalación (se deja por defecto). Pulse *Next >*.

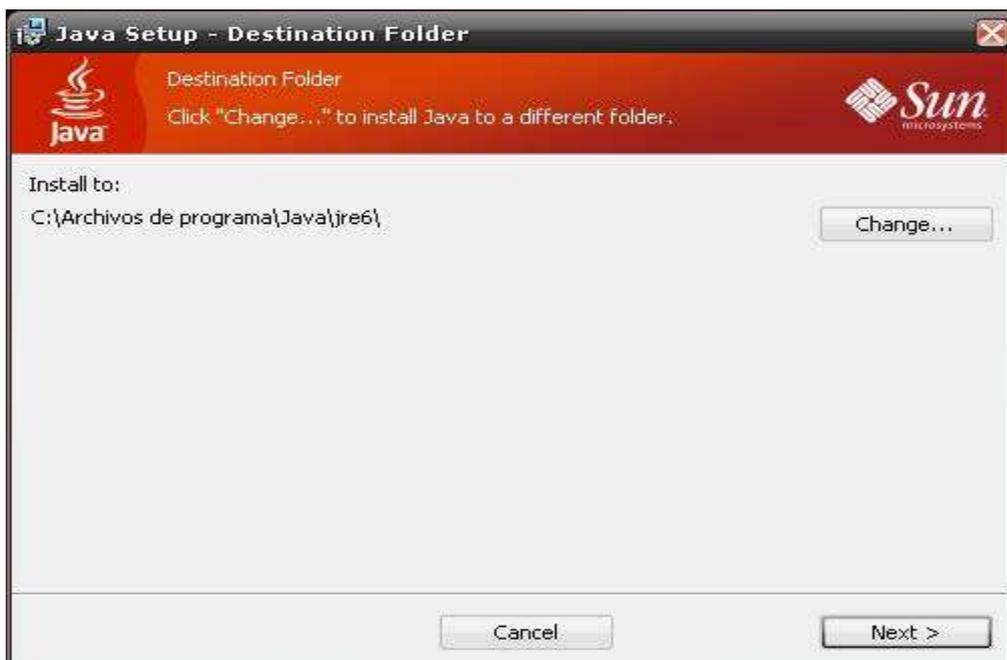


Ilustración 28. Pantalla de selección de carpeta de instalación

Una vez terminado este paso se acaba la instalación del j2sdk. Aparecerá la pantalla de instalación correcta. El usuario pulsara el botón *Finish* y se reiniciara el ordenador para la correcta instalación de este kit.



Ilustración 29. Pantalla de instalación correcta



D. INSTALACIÓN NETBEANS 6.7.1

NetBeans es una plataforma para el desarrollo de aplicaciones de escritorio usando el lenguaje Java y un entorno de desarrollo integrado (IDE) para desarrollar bajo esta plataforma. Su aprendizaje se ha convertido en fundamental para quienes están interesados en el desarrollo de aplicaciones multiplataforma.

La plataforma *NetBeans* permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con las API de *NetBeans* y un archivo especial (*manifest file*) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.

Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma *NetBeans* pueden ser extendidas fácilmente por otros desarrolladores de software.

Mediante *NetBeans* es posible diseñar aplicaciones con solo arrastrar y soltar objetos sobre la interfaz de un formulario. Si ya te familiarizaste con entornos como .NET no te costará trabajo aprender los nuevos *JLabels*, *JButtons* o *JTextFields*. *NetBeans* pone a disposición de los usuarios decenas de módulos a través de su página web, que podrás integrar en él para conseguir mejores aplicaciones.

Puesto que esta versión de *Netbeans* proporciona una plataforma de desarrollo para aplicaciones móviles, será la ideal para realizar la implementación del sistema.

En cuanto a la instalación, lo primero que se tiene que hacer es descargar el paquete *Netbeans* que incluye la versión J2ME. Desde el siguiente enlace se puede realizar la descarga:

- <http://www.netbeans.org/downloads/>

En la siguiente ilustración se muestra que paquete de descarga de *Netbeans* es el que el usuario debe seleccionar, mediante un encuadrado rojo.

Paquetes de descarga de NetBeans IDE

Tecnologías *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
Java SE	•	•	•				•
JavaFX		•					•
Java Web y EE			•				•
Java ME			•				•
Ruby				•			•
C/C++					•		•
Groovy			•				•
PHP						•	•
Servidores incluidos							
Sun GlassFish Enterprise Server v2.1			•				•
Sun GlassFish Enterprise Server v3 Prelude			•	•			•
Apache Tomcat 6.0.18			•				•
	Download Libre, 47 MB	Download Libre, 90 MB	Download Libre, 238 MB	Download Libre, 59 MB	Download Libre, 30 MB	Download Libre, 26 MB	Download Libre, 302 MB

Ilustración 30. Pantalla de selección de paquete *Netbeans*

Una vez descargado el paquete de *Netbeans*, se hará doble clic en el programa de instalación de *Netbeans*, en este caso se ha descargado versión 6.7.1:

- *netbeans-6.7.1-ml-java-windows*



Ilustración 31. Icono para la instalación de *Netbeans*

A continuación aparecerá la pantalla de bienvenida de instalación, mostrando los componentes que contiene el paquete descargado. Puesto que para este proyecto no es necesario paquetes que no sean el de J2ME, seleccionaremos el botón *Customize* para seleccionar los paquetes específicos para esta aplicación. En la ilustración siguiente se muestra esta pantalla.

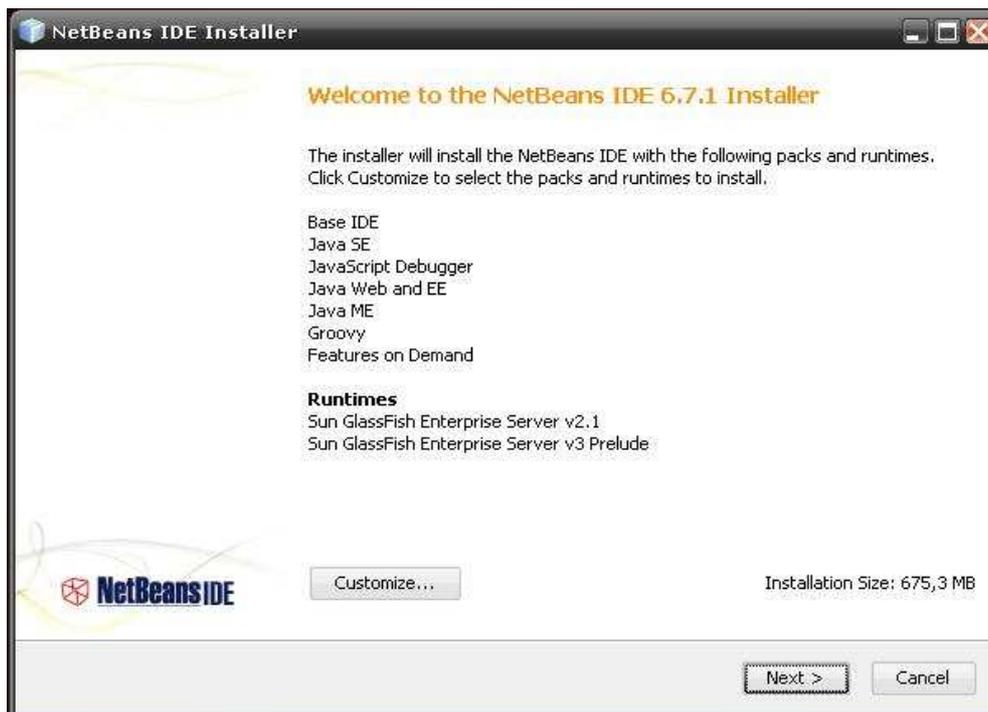


Ilustración 32. Pantalla de bienvenida de instalación

Una vez entrado en la selección de componentes a instalar, seleccionamos los componentes necesarios para esta aplicación. Se seleccionan los seleccionados en la ilustración siguiente y se pulsa el botón *OK*.



Ilustración 33. Pantalla de selección de componentes

Una vez seleccionados los componentes, se vuelve a la pantalla de bienvenida de instalación, esta vez con los nuevos componentes. Se selecciona el botón *Next>*.

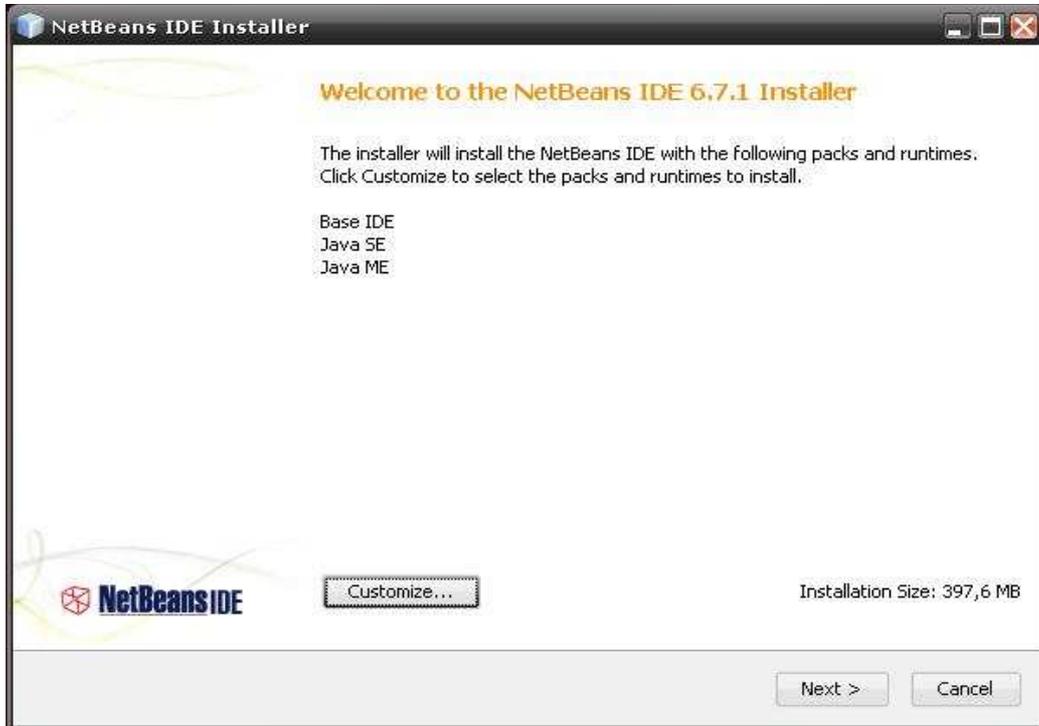


Ilustración 34. Pantalla de bienvenida con nuevos componentes

A continuación, aparece el cuadro de licencia, se acepta y se pulsa *Next>*.

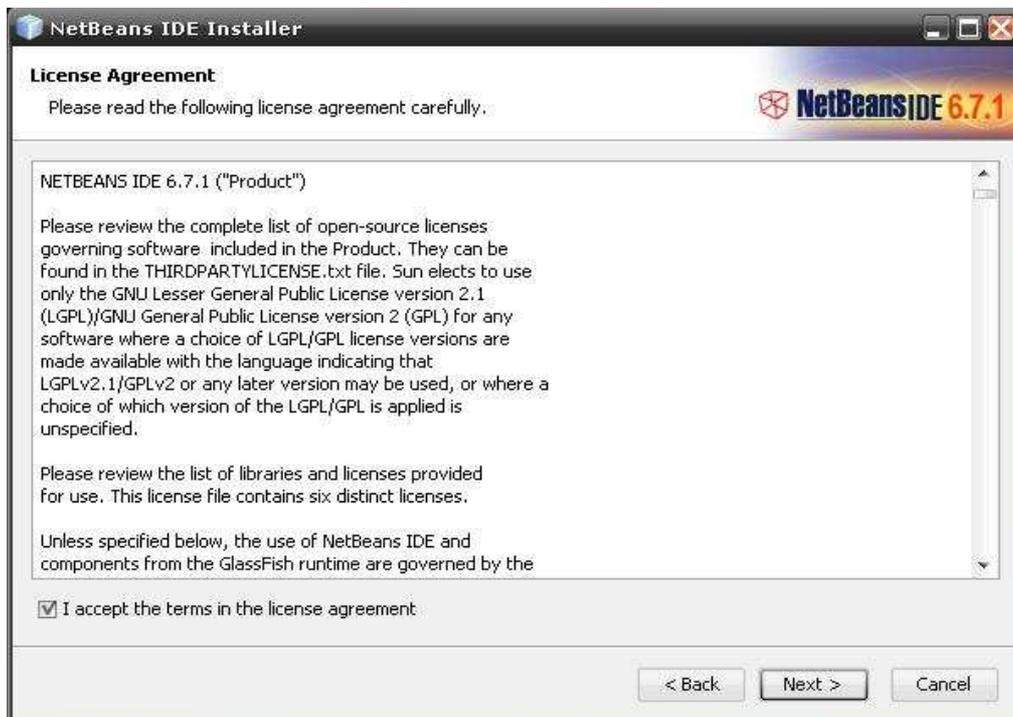


Ilustración 35. Pantalla de cuadro de licencia

Posteriormente aparecerá la pantalla de selección de carpeta de destino de la instalación de la herramienta de desarrollo, se deja por defecto la que aparece.

También se ofrece al usuario la selección del paquete Java instalado, por defecto *Netbeans* conoce la ruta de la carpeta de instalación de Java, por tanto el usuario dejara por defecto la dirección que aparece.

En la siguiente ilustración aparece dicha pantalla.



Ilustración 36. Pantalla de selección de carpeta destino

Una vez concluido este paso, se acaba la configuración del paquete de instalación de *Netbeans*. Únicamente quedará comenzar la instalación de esta herramienta de desarrollo y ya habrá concluido todo.

En la siguiente ilustración se muestra al usuario la información del paquete que se va a instalar, es decir, el tamaño que ocupa en disco y la carpeta de destino. El usuario debe pulsar el botón *Install* y comenzará la instalación.



Ilustración 37. Pantalla previa a la instalación

A continuación aparece la pantalla de progreso de instalación. En este paso se debe esperar a que se copien todos los ficheros en esta pantalla. Puede tardar varios minutos.



Ilustración 38. Pantalla de progreso de la instalación



Una vez terminado este paso se acaba la instalación de la herramienta de desarrollo *Netbeans*. Aparecerá la pantalla de instalación completada. El usuario pulsara el botón *Finish* y se reiniciara el ordenador para la correcta instalación de este kit.

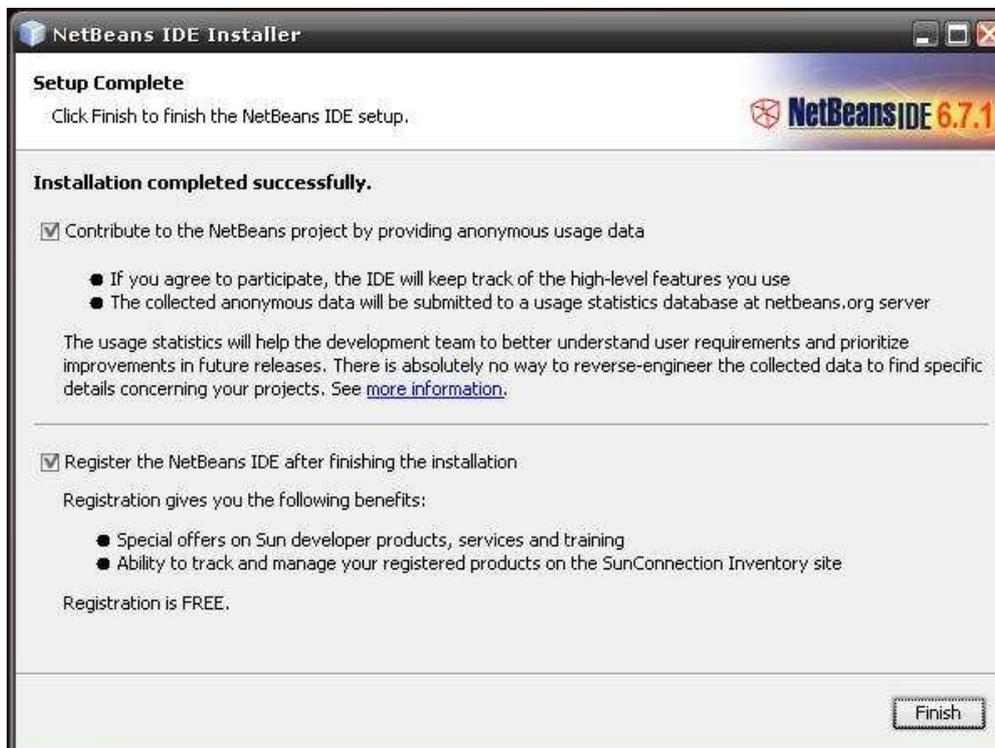


Ilustración 39. Pantalla de instalación completa