



Universidad de Valladolid

E.T.S. DE INGENIERIA INFORMATICA

Ingeniería Técnica en Informática de Sistemas

Aplicación para la monitorización remota de pacientes

Alumno: Jose Antonio González Vesga

***Tutores: Javier Finat Codes
Miguel Angel Laguna Serrano***

INDICE

INTRODUCCION AL PROYECTO	7
1. INTRODUCCIÓN.....	8
2. OBJETIVOS Y MOTIVACIONES.....	8
3. DESCRIPCIÓN DEL PROYECTO.....	9
3.1. <i>Situación Actual</i>	9
3.2. <i>Solución Planteada</i>	9
4. ESTRUCTURA DEL DOCUMENTO	10
MONITORIZACION REMOTA DE PACIENTES.....	13
1. INTRODUCCIÓN.....	14
2. ORIGEN. PROYECTO AIVI	14
2.1. <i>Introducción</i>	14
2.2. <i>Objetivos</i>	14
2.3. <i>Motivaciones del Proyecto</i>	15
2.4. <i>Otras Iniciativas</i>	16
2.5. <i>Hogar Digital</i>	16
2.6. <i>TeleAsistencia</i>	18
3. PFC MONITORIZACIÓN REMOTA DE PACIENTES.....	19
3.1. <i>Introducción</i>	19
3.2. <i>Monitorización Remota de Pacientes</i>	19
3.2.1. Contexto. Complejo Residencial.....	20
3.2.2. Objetivos.....	20
ESTUDIO DE LA TECNOLOGIA DISPONIBLE.....	23
1. INTRODUCCIÓN.....	24
2. HERRAMIENTAS DE PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES	24
2.1. Tipos de Arquitectura de Programación.....	25
3. DISPOSITIVOS MÓVILES.....	26
3.1. Tipos de dispositivos móviles.	26
4. SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES	28
5. SENSORES.....	28
5.1. Características de los sensores	28
5.2. Tipos de Sensores.....	29
6. REDES	31
6.1. Bluetooth.....	31
6.2. Wifi	32
7. CONCLUSIONES Y TECNOLOGÍA ELEGIDA	33
7.1. Windows Mobile, Visual Studio .Net y .Net Compact Framework	33
7.2. Comunicaciones.....	34
7.3. Hardware. PDA y Sensores.....	35
ANALISIS.....	39
1. INTRODUCCIÓN.....	40
2. REQUISITOS SOFTWARE.....	40
2.1. <i>Descripción General de la Aplicación</i>	40

2.2. <i>Funcionalidad de la Aplicación</i>	43
2.3. <i>Requisitos Funcionales</i>	44
2.4. <i>Requisitos No Funcionales</i>	46
3. CASOS DE USO	47
3.1. <i>Actores</i>	48
3.2. <i>Casos de Uso</i>	49
3.2.1. Descripción de los Casos de Uso	51
2. MODELO DE ANÁLISIS	63
2.1 <i>Introducción</i>	63
2.2 <i>Diagrama de Clases Inicial</i>	63
2.2.1. Descripción de Clases de Análisis	64
2.2.2. Realización de Casos de Uso	65
2.3. <i>Modelo de Dominio</i>	68
2.3.1. Diagrama de Clases	68
DISEÑO	71
1. INTRODUCCIÓN	72
2. MODELO ESTRUCTURAL.....	72
2.1. <i>Proyecto Global Monitorización Pacientes</i>	72
2.2. <i>Diagrama de Paquetes para Monitorización PDA</i>	73
3. DISEÑO DE CAPA DE PRESENTACIÓN. INTERFAZ	74
3.1. Diagrama de Clases	74
4.1. <i>Diagramas de Secuencia</i>	78
4.2. <i>Diagrama de Clases Detallado</i>	82
5. DISEÑO DE LA CAPA DE PERSISTENCIA.....	84
6. MODELO DE DATOS.....	85
6.1. <i>Datos de Sensores y Notificaciones de Situaciones de Riesgo</i>	85
6.2. <i>Ficheros Configuración Pda y Paciente</i>	86
IMPLEMENTACION	93
1. INTRODUCCIÓN	94
2. REQUISITOS SOFTWARE.....	94
3. OPTIMIZACIÓN USO DE BATERÍA.....	94
4. FICHEROS DE CONFIGURACIÓN	95
4.1. <i>Fichero de Configuración General</i>	95
4.2. <i>Ficheros de Configuración de Paciente</i>	96
PRUEBAS	99
1. INTRODUCCIÓN	100
2. CASOS DE PRUEBA	100
3. DURACIÓN DE BATERÍA	108
MANUAL DE USUARIO	111
1. INTRODUCCIÓN	112
2. MANUAL DE USUARIO.....	112
2.1. Inicio de la aplicación.....	112
2.2 Pantalla Estado Paciente.....	114
2.3. Pantalla Configuración General.....	116
2.4. Pantalla Estado Sensores	117

2.5. Pantalla Estado GPS.....	118
2.6. Pantalla Configuración Paciente	119
CONCLUSIONES.....	123
BIBLIOGRAFIA.....	125
APENDICES	127
APÉNDICE A. MICROSOFT .NET Y NET COMPACT FRAMEWORK	128
APÉNDICE B. SERVICIOS WEB XML	137

INTRODUCCION AL PROYECTO

1. Introducción

Este Proyecto Fin de Carrera (PFC) nace de la colaboración de la Universidad de Valladolid en el proyecto Ambientes Inteligentes para la Vida Independiente (AIVI). El proyecto AIVI tiene como principal objetivo promover un salto cualitativo en la implantación de servicios de inteligencia ambiental en el hogar dirigidos a personas dependientes (ancianos y discapacitados), con la finalidad de extender al máximo el tiempo que dichas personas pueden permanecer en sus hogares, bien realizando autónomamente el mayor número posible de tareas cotidianas, bien siendo asistidas por familiares o cuidadores; en cuyo caso la inteligencia ambiental supone una ayuda complementaria muy valiosa.

En el proyecto AIVI se cubren todas las áreas de conocimiento relacionadas con el diseño e implantación de inteligencia ambiental en hogares de personas dependientes: aspectos arquitectónicos, técnicas de construcción y equipamiento, redes de sensores, interfaces, servicios de teleasistencia, etc. Por otra parte, se cuenta con la colaboración de asociaciones y organizaciones de apoyo a personas dependientes, que establecerán las líneas prioritarias de investigación, validarán las aplicaciones y servicios desarrollados, y darán su asesoramiento en las cuestiones éticas que surjan durante el transcurso del proyecto.

Es en una de estas áreas donde tiene lugar el presente proyecto, el cual consiste en el desarrollo de una aplicación, que instalada en un dispositivo PDA, se encarga de recoger la información de los sensores que un paciente tiene colocados en su cuerpo; con objeto de, primero analizarlos en busca de situaciones de riesgo, y posteriormente enviarlos a un servidor central para que sean interpretados por el personal médico.

Este PFC se engloba en parte de un sistema ya desarrollado, y por tanto únicamente hemos documentado el funcionamiento de la parte correspondiente a la aplicación que funciona sobre el PDA.

2. Objetivos y Motivaciones

Las metas abordadas en la elaboración de este PFC, son los siguientes:

1. Desarrollo de la aplicación móvil sobre el PDA

Aprender los conocimientos necesarios para el desarrollo de aplicaciones sobre PDA, incrementando mis conocimientos sobre la herramienta Visual Studio 2005 y ampliar mis conocimientos en general sobre la arquitectura Microsoft .Net

2. Ampliación de los conocimientos en comunicaciones

Una particularidad interesante del proyecto es la gran variedad de comunicaciones que están implicadas; aprender estos conocimientos sobre estas tecnologías de comunicación, y aplicarlo al desarrollo ha sido un reto, que inicialmente fue complicado, pero que siempre ha sido muy interesante.

3. Motivación social del proyecto

Desde el inicio ha sido motivador el objetivo del proyecto y su aplicación en la vida real, sobre todo, conociendo la evolución del envejecimiento de la población y el aumento de costes que

va a suponer en un futuro esta nueva realidad. Ser parte de la solución, aunque con una pequeña contribución, ha supuesto una motivación adicional en su desarrollo.

4. Desarrollo del Servidor Central. Servicio Web.

Inicialmente la aplicación objeto del PFC, se comunicaba contra un servidor central proporcionado por miembros del proyecto AIVI. Finalmente, debido a la falta de especificación de dicho servidor, hemos tenido que desarrollar un servidor central propio, que permitiera probar el buen funcionamiento de la aplicación. Aunque este servidor no forma parte de la presente documentación, adquirir los conocimientos sobre su desarrollo, ha sido interesante.

5. Artículos Presentados

Como consecuencia de los desarrollos realizados en este trabajo, se ha redactado un trabajo que ha sido presentado y publicado en los Proceedings de la 7th European Conference on Product and Process Modelling (ECPMP) que ha tenido lugar del 10 al 12 de Septiembre de 2008 en Marsella. La valoración positiva de los referees es un índice del interés por la aplicación desarrollada, no sólo en el ámbito de Salud al que va dirigido, sino también de sus potenciales aplicaciones a otros sectores productivos.

3. Descripción del Proyecto

3.1. Situación Actual

En los complejos asistenciales actuales, los pacientes viven en sus propias habitaciones individuales, a lo sumo acompañados de una segunda persona; una situación que complica enormemente la monitorización, o bien complica la intimidad del paciente.

La monitorización de las personas que se encuentran en situaciones de riesgo, se realiza a través de equipos que debido a su tamaño y características, no son móviles, o bien su transporte se hace complicado. En el caso de personas, cuya situación de riesgo no es elevada, ni siquiera se monitoriza debido o bien al coste de los dispositivos, o bien a que el paciente no quiere perder calidad de vida, por estar conectado continuamente a una máquina.

Otro problema es la falta de personal en los complejos residenciales para atender a todos los pacientes de una forma activa, ya que habitualmente con el objeto de reducir costes, la relación entre pacientes y personal médico es muy alta, y esto penaliza la atención a los pacientes.

3.2. Solución Planteada

La solución planteada proporciona una solución a una parte del problema, permitiendo una monitorización remota y móvil de cada uno de los pacientes, consiguiendo que el paciente gane en calidad de vida.

La calificación de remota significa que el personal médico será capaz de conocer todos los datos que recogen los sensores de cada uno de los pacientes en un único punto de manera no-intrusiva,

sin tener que acercarse a cada uno de los pacientes para comprobar sus valores biométricos y facilitando así una mayor intimidad del paciente. La recepción de esta información puede ser también recibida en un terminal móvil, permitiendo que la monitorización también sea continua.

Hemos indicado que la monitorización también será móvil. La solución planteada se basa en dispositivos hardware de pequeño tamaño, portátiles e inalámbricos. De este modo, el paciente no está “atado” a una máquina fija, o bien no tiene por qué cargar con pesados dispositivos. Este punto es una limitación de la solución, ya que la monitorización depende de los dispositivos portátiles disponibles.

Por supuesto, el sistema es adaptativo, es decir, el desarrollo aportará no solo información de los valores biométricos, sino que se podrán configurar reglas para cada uno de los pacientes. De este modo, no sólo se le harán llegar los valores biométricos del paciente al personal médico, sino que se le harán llegar notificaciones sobre las situaciones de riesgo que se den al paciente.

Por tanto la solución consiste en el desarrollo de una aplicación para un dispositivo PDA, el cual se encargará de recoger la información de los sensores configurados por el personal médico para el paciente. Estos datos serán almacenados y evaluados en busca de Situaciones de Riesgo en el paciente. Finalmente, estos datos y las notificaciones de situaciones de riesgo serán enviadas al servidor central para que estén disponibles al personal médico. Detallaremos en capítulos posteriores la solución de forma más detallada.

4. Estructura del Documento

La memoria aquí presentada contiene los siguientes contenidos

1. Introducción

Se presenta una breve descripción del conjunto del documento, realizando una breve descripción del contenido de las soluciones aportadas en el proyecto.

2. Monitorización Remota de Pacientes

Esta sección presenta una descripción de las soluciones de la medicina asistencial actual basada en monitorización remota de pacientes, para finalmente describir en detalle el funcionamiento de la solución planteada.

3. Tecnología Disponible

Se realiza una breve descripción de las diferentes alternativas en cuestión de tecnología disponibles en el mercado, para finalmente justificar las elegidas para el presente proyecto.

4. Análisis

Este documento describe la especificación de requisitos obtenida a partir del punto 2, y así comenzar a describir los diferentes diagramas de análisis hasta obtener el modelo de dominio del sistema.

5. Diseño

Este documento permite, a partir del documento de análisis, detallar el funcionamiento de la aplicación y adaptarlo a la tecnología escogida en el punto 3.

6. Implementación

Se trata del documento de despliegue de la aplicación, en el cual se pueden comprobar los diferentes elementos que intervienen en el funcionamiento de ésta.

7. Pruebas

Se trata del documento que describe las pruebas de caja negra realizadas sobre la aplicación desarrollada.

8. Manual de Usuario

Documento que nos ilustra sobre el funcionamiento de la aplicación.

9. Conclusiones

En esta sección se describirán las conclusiones obtenidas en el desarrollo del producto y las aplicaciones prácticas del mismo. Además se describirán futuras ampliaciones sobre el desarrollo en cuestión.

10. Bibliografía

Los documentos, libros y referencias web aplicados en el desarrollo del producto.

11. Apéndice

Información útil que se ha aplicado en el desarrollo del proyecto.

MONITORIZACION REMOTA DE PACIENTES

1. Introducción

En este capítulo se define con mayor detalle el proyecto de Monitorización Remota de Pacientes. Este punto de la documentación comienza con la descripción más amplia del proyecto AIVI, el cual en el desarrollo de una de sus líneas, la Telemedicina, se engloba el presente proyecto. Por último, se define el proyecto concreto de Monitorización Remota de Pacientes.

2. Origen. Proyecto AIVI

2.1. Introducción

El proyecto tiene su origen en la colaboración de la Universidad de Valladolid (UVA) en el proyecto “Ambientes Inteligentes para la Vida Independiente” (AIVI), el cual tiene como objetivo promover un salto cualitativo en la implantación de servicios de inteligencia ambiental en el hogar dirigidos a personas dependientes (ancianos y discapacitados), con la finalidad de extender al máximo el tiempo que dichas personas pueden permanecer en sus hogares, bien realizando autónomamente el mayor número posible de tareas cotidianas, bien siendo asistidas por familiares o cuidadores, en cuyo caso la inteligencia ambiental supone una ayuda complementaria muy valiosa.

En el proyecto AIVI se cubren todas las áreas de conocimiento relacionadas con el diseño e implantación de inteligencia ambiental en hogares de personas dependientes: aspectos arquitectónicos, técnicas de construcción y equipamiento, redes de sensores, interfaces, servicios de teleasistencia, etc. Por otra parte, se cuenta con la colaboración de asociaciones y organizaciones de apoyo a personas dependientes, que establecerán las líneas prioritarias de investigación, validarán las aplicaciones y servicios desarrollados, y darán su asesoramiento en las cuestiones éticas que surjan durante el transcurso del proyecto.

2.2. Objetivos

El objetivo general del proyecto AIVI se desglosa en los siguientes objetivos concretos:

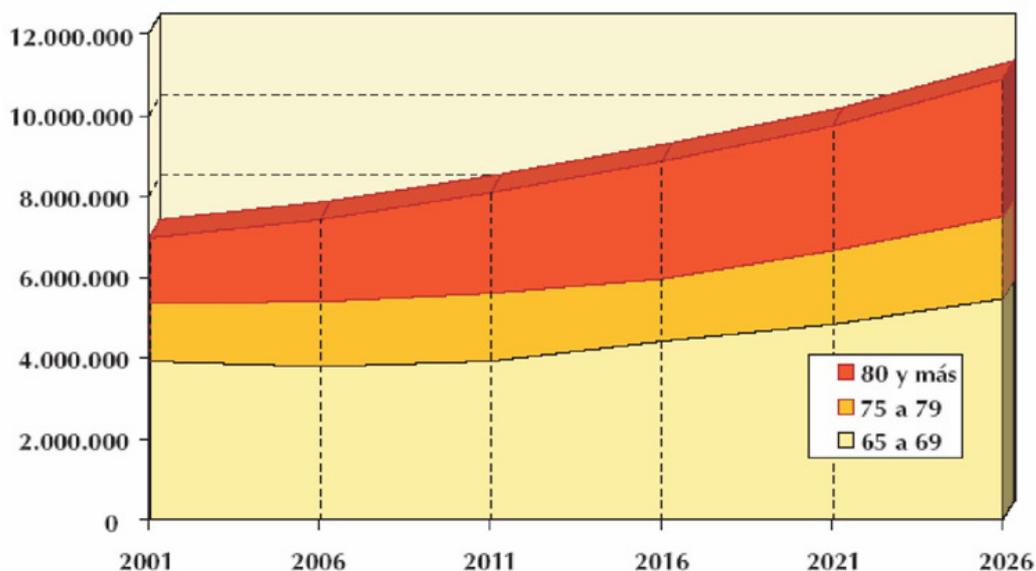
- Definición y desarrollo de una plataforma abierta sobre la que se puedan implantar variados servicios de carácter avanzado e innovador: comunicaciones, teleasistencia, gestión domótica, etc. La integración de todos estos servicios es lo que constituye el ambiente inteligente.
- Desarrollo de servicios de teleasistencia y de resolución de incidencias. El sistema se encarga de monitorizar cualquier anomalía, ofreciendo ayuda en la resolución de problemas.
- Personalización de servicios, que engloba conceptos tales como gestión de la identidad, localización, reconocimiento y adaptación al contexto, autoaprendizaje de las necesidades y preferencias del usuario, etc.
- Realización de experiencias piloto que permitan conocer la problemática real de la implantación de inteligencia ambiental en los hogares, desde el punto de vista tanto del instalador, como del usuario final y de los proveedores de servicios. Aunque el proyecto AIVI está enfocado principalmente al hogar, no se descartan para las experiencias piloto otros ámbitos de interés para las personas dependientes: residencias, hoteles, etc.

- Validación técnica y funcional por parte de los usuarios finales de todos los elementos y componentes que se van a desarrollar en el proyecto, asegurando la correcta integración de los mismos. En dicha validación se cuenta con la ayuda y asesoramiento de asociaciones y organizaciones de apoyo a personas dependientes.

2.3. Motivaciones del Proyecto

El Consejo de Europa define la dependencia como "la necesidad de ayuda o asistencia importante para las actividades de la vida cotidiana", o, más concretamente, como "un estado en el que se encuentran las personas que por razones ligadas a la falta o la pérdida de autonomía física, psíquica o intelectual, tienen necesidad de asistencia y/o ayudas importantes a fin de realizar los actos corrientes de la vida diaria y, de modo particular, los referentes al cuidado personal". Por otra parte, aunque los datos son ya algo antiguos, la Encuesta sobre Discapacidades, Deficiencias y Estado de Salud de 1999 (EDDES 99) cifraba en 3.528.221 el número total de personas con alguna discapacidad o con limitaciones que han causado o pueden llegar a causar discapacidades, lo cual representa aproximadamente un 9% de la población española.

El envejecimiento de la población española es un factor que incrementará significativamente en el futuro el porcentaje de población dependiente, ya que existe una estrecha relación entre dependencia y edad. Según las proyecciones de población del INE, el número de personas mayores de 65 años supondrá en veinte años más del 20% de la población total española, y dentro de ese grupo de población se producirá un fuerte aumento de las personas mayores de 80 años, edad a partir de la cual la tasa de prevalencia de dependencia crece notablemente.



Fuente: INE, Proyecciones de población calculadas a partir del Censo de 2001.

Figura 1 . Evolucion de la Poblacion Española

Hasta ahora han sido principalmente las familias, especialmente las mujeres, las que se han encargado del cuidado de las personas dependientes. Sin embargo, los cambios que han ido aconteciendo en la sociedad, tales como la incorporación masiva de la mujer al mercado de trabajo, la caída de la fecundidad, la movilidad geográfica o las transformaciones de las relaciones familiares hacen que este modelo de “asistencia informal” se vuelva insostenible a medio plazo.

2.4. Otras Iniciativas

Paralelamente al proyecto AIVI, existen otras iniciativas que se encaminan en la misma línea, de tal forma que ayudan a que las personas con discapacidad puedan mantener una calidad de vida aceptable. Así, actualmente se están desarrollando iniciativas como sistemas de telefonía de textos, pasarela de textos, automatización de subtítulos, pasarelas multimodales, etc., que permitan a personas en situación de dependencia la utilización de las posibilidades que brinda Internet (Teleaprendizaje, Teletrabajo, Teleocio, Teleasistencia, Telemedicina, Teledemocracia) y así aumentar su calidad de vida.

También existen iniciativas para el desarrollo de Sistemas de Teleformación y Teletrabajo para personas en situación de dependencia. Las tecnologías de la información y el uso, cada vez mayor, de las redes telemáticas está generando nuevas formas de enseñar, de trabajar y, como no, de comunicar, que han dado lugar a nuevos conceptos que implican nuevos estilos de vida, como son, la teleformación, el teletrabajo, la teleasistencia o los sistemas multimedia. La formación informática especializada tiene como objeto paliar la escasez de especialistas en tecnologías de la información. En España este problema se agudiza por el retraso en la alfabetización digital y la escasez en infraestructura de redes y equipamiento, lo que incide en la pérdida de oportunidades para el progreso.

2.5. Hogar Digital

El desarrollo de los teleservicios y comunicaciones se ven favorecidos por la plena irrupción de Internet en el hogar y, en general, las denominadas TIC (Tecnologías de la Información y las Comunicaciones). Se ha forjado una nueva forma de entender la aplicación de la tecnología en la vivienda, mucho más positivista y realista, dónde lo único importante es el propio usuario y no ésta. Es decir, de la tecnología por la tecnología se ha pasado a asegurar la consecución de las necesidades o deseos de los usuarios a través de servicios, donde evidentemente la tecnología adquiere un papel de soporte muy importante. Con ello, la tecnología debe ser algo transparente para el usuario, que no está interesado en las innovaciones técnicas, sino en la capacidad de estas innovaciones para resolver su problema, necesidad o deseo.

Paralelamente a este fenómeno, la disponibilidad de arquitecturas software y la aparición de dispositivos digitales móviles que se conectan a diferentes redes (PDA, laptops...) están transformando profundamente la vida cotidiana de los usuarios. Como consecuencia, puede verse una futura convergencia entre las redes de área extensa y las redes privadas del hogar. A pesar de que dicha convergencia va a abrir nuevos mercados y áreas de negocio, introduce también grandes retos, abriendo áreas de investigación relacionadas con la seguridad, la personalización y la distribución de datos.

Dichos fenómenos hacen surgir la necesidad de dar el paso decisivo siguiente para potenciar el mercado español, europeo y mundial de productos domésticos: Asegurar el desarrollo de un

mercado horizontal, donde exista una convergencia entre los sectores involucrados en la vivienda, hasta el momento independientes o no interrelacionados. Evidentemente, el desarrollo de este nuevo mercado horizontal requiere asegurar la capacidad de comunicación e interacción entre todos los equipos domésticos de la vivienda, los domóticos incluidos, creando lo que suele conocerse “equipos comunicantes”, diferenciándose del poco afortunado término “inteligente”, muy utilizado durante la década de los noventa.

En el mercado europeo existen numerosas maneras de denominar a esta nueva forma de concebir la comunicación en la vivienda o al propio hogar (Digital Homes, Connected Homes, eHomes, Smart Homes, iHomes, etc.). En España, se está popularizando el nombre de “Hogar Digital” como más relevante, impulsado por grandes entidades promotoras de este proyecto.

Dichos procesos sentarán las bases para el desarrollo de la denominada “Inteligencia Ambiental”, concepto muy ligado a tecnologías clave como son la computación y comunicación ubicua y los interfaces de usuario inteligentes y multi-modales. A través de ellos se crea un entorno donde el usuario puede interactuar con una gran diversidad de dispositivos de una manera totalmente transparente, permitiendo un aumento de servicios y aplicaciones

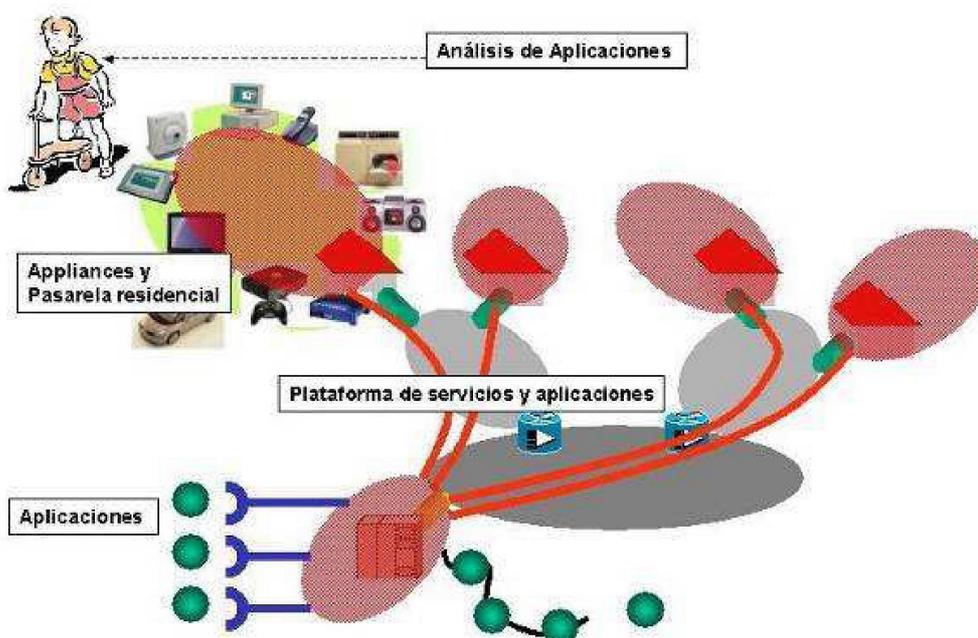


Figura 2. Provisión de servicios en el hogar.

Los ambientes inteligentes sitúan al usuario en el centro de futuros desarrollos para una sociedad basada en el conocimiento que incluya a todos. Este proceso proporciona grandes oportunidades al aumentar el rango de servicios posibles en el hogar.

De igual manera, la evolución de los equipos terminales hacia una mayor integración y menor tamaño hace más difícil su manejo, pero al mismo tiempo las mejoras en prestaciones abren el

camino a la búsqueda de nuevas soluciones que permitan el acceso rápido y sencillo a las nuevas aplicaciones y servicios que están empezando a surgir en el mercado. Por ejemplo, el teclado, como elemento de diálogo en estos nuevos dispositivos (teléfonos móviles, PDA), plantea dificultades de integración. Se hace necesaria la búsqueda de nuevas interfaces que faciliten el uso de dispositivos portátiles, y que permitan la utilización rápida de los mismos en cualquier situación. Las interfaces multi-modales se ofrecen así como un complemento del teclado (que debe quedar incluido como un canal más de la nueva interfaz). Estas interfaces resultan sencillas de utilizar, ya que proporcionan una forma natural de comunicación con el terminal, incluso sin necesidad de tener que mirar a una pantalla.

2.6. TeleAsistencia

Actualmente se pretende obtener un servicio de “ambiente inteligente” orientado a la teleasistencia como un modelo de desarrollo social de cara al futuro y como un servicio de prevención y atención a la dependencia. Un servicio integral destinado al servicio de las personas dependientes permite, que éstas tengan un contacto directo y continuo con profesionales de la asistencia social, al mismo tiempo que se encuentran perfectamente comunicados en caso de emergencia.

Es clave para las personas dependientes la coordinación de los diferentes niveles asistenciales y la continuidad en el servicio, los profesionales del sector y las instituciones deben tener en cuenta los retos que plantea el aumento de la población dependiente. La asistencia domiciliaria es enormemente cara, es necesario llevar al médico al domicilio del paciente y en muchas ocasiones sin necesidad. La gestión integral del ambiente de la persona dependiente es una solución idónea para abaratar las partidas presupuestarias y mejorar la calidad de vida de las personas dependientes. Actualmente cerca de 50.000 españoles reciben este servicio.

Si se destaca la eficacia de la teleasistencia a la hora de resolver emergencias no se debe olvidar su importancia como sistema preventivo, ya que disminuye las angustias del usuario ante el problema de encontrarse solo ante cualquier eventualidad, se le avisa de determinados hechos puntuales, como la hora de tomarse las pastillas o de una cita con el médico y realizando un seguimiento de cada usuario. El servicio de teleasistencia está pensado fundamentalmente para personas mayores, pero también pueden beneficiarse de este sistema otro tipo de usuarios dependientes como discapacitados físicos o psíquicos, enfermos crónicos, mujeres embarazadas, personas convalecientes en postoperatorio, menores de edad que pasan muchas horas solas en casa, etc. Es en este punto donde se comienza a integrar el desarrollo objeto de esta PFC, el cual presentamos a continuación.

3. PFC Monitorización Remota de Pacientes

3.1. Introducción

Como se ha indicado anteriormente, el proyecto de monitorización Remota de Pacientes, se engloba en el concepto de TeleAsistencia, y trata de resolver uno de los problemas planteados en este nuevo tipo de asistencia medica; y aunque nuestro proyecto se encuentra bajo el contexto de complejos residenciales, este se puede aplicar a atenciones domiciliarias de otra índole.

3.2. Monitorización Remota de Pacientes

El proyecto pretende la monitorización de un paciente a través de una red de sensores, que comuniquen los valores biométricos del paciente a un dispositivo móvil que acompañara a este. Este dispositivo se encargara de acumular todos los datos de todos los sensores, para cada cierto intervalo de tiempo, poder enviarlos a un Servidor Central donde sean procesados y almacenados, de tal forma que estén a disposición del personal medico.

Se trata de una monitorización “continua” del paciente, de tal forma que el personal medico, desde su puesto de trabajo o bien desde cualquier punto con acceso a los datos recogidos, pueda conocer en cada momento los valores biométricos del paciente. El conocimiento en tiempo real de la evolución del paciente, permite que el personal medico pueda detectar cuadros de riesgo sobre los datos del paciente, y pueda realizar un diagnostico preventivo de la situación de este. Esto permite que la medicina pase a una fase preventiva, conociendo con anticipación lo que le puede ocurrir a cada paciente.

Por otra parte, el proyecto pretende dotar de inteligencia al dispositivo encargado de realizar la tarea de captura de datos, de tal forma, que a través de reglas simples de comparación de datos, el dispositivo pueda determinar el estado del paciente. En el caso de que se llegue a la conclusión de que el paciente se encuentre en una situación de riesgo, el dispositivo deberá informar al personal medico a través de un mecanismo de alarma, para que este actúe si lo considera necesario.

Este último punto permite que el tiempo de intervención en una situación de riesgo de un paciente se reduzca al mínimo, de tal forma que según esta ocurriendo la situación de riesgo, el personal medico queda avisado de inmediato. De la misma forma, al ser avisado el personal medico, se eliminan los tiempos y personal empleado en hacer rondas para comprobar el estado de los pacientes.

Hay que tener en cuenta, que el proyecto AIVI define la importancia de la independencia de los pacientes y de su aumento de su calidad de vida. Este punto marca un requisito muy importante en la definición del proyecto, y es que esta monitorización debe permitir la movilidad del paciente, de tal forma que tanto si el paciente se encuentra en el complejo residencial, como si quiere salir fuera del recinto de este, la monitorización debe continuar, y los datos, y sobre todo las notificaciones sobre situaciones de riesgo, deben, en la medida de lo posible, seguir llegando al personal medico.

Esta ultima medida, no solo se trata de un requisito a tener en cuenta en la aplicación, sino también del hardware a utilizar y sobre todo de los sensores. La psicología del paciente, es un factor que influye enormemente en la calidad de vida del paciente, y tener un cuerpo lleno de cables no ayuda

a que el paciente se sienta libre. Por tanto se ha definido como requisito adicional, que para que la monitorización permita una completa movilidad, los sensores deberán transmitir de forma inalámbrica los valores biométricos recogidos del paciente.

3.2.1. Contexto. Complejo Residencial.

Hemos definido como se va a realizar la monitorización. Ahora toca definir el contexto de uso, con el objetivo de optimizar los requisitos en función de esta situación.

El desarrollo objeto de la PFC se va a utilizar en complejos residenciales. Los complejos residenciales son centros de asistencia donde los pacientes residen, y donde tenemos personal medico que se encarga de su asistencia. Estos complejos residenciales son abiertos, y los pacientes pueden salir a las zonas de paseo habilitadas dentro del complejo, y por supuesto fuera del recinto de este.

Este ultimo punto es importante, ya que un paciente que se encuentra en una situación de riesgo, y que se encuentre fuera del complejo, es importante tenerlo localizado, ya que determinados pacientes, en función de su enfermedad, pueden desorientarse y será necesario tenerlos localizados.

Tanto los sensores, como los dispositivos serán colocados por el personal medico del complejo residencial; y serán el mismo personal medico los que recibirán los datos y notificaciones recogidos por los dispositivos. Este personal no esta especialmente cualificado para el uso de PDAs, y por tanto la aplicación a desarrollar deberá ser sencilla e intuitiva.

3.2.2. Objetivos

Exponemos a continuación los objetivos del proyecto, afines a los objetivos del proyecto AIVI, y por tanto del desarrollo:

- Mejora en Calidad de Vida e Independencia

Este objetivo pretende que el paciente se sienta mas seguro en el desarrollo de su vida personal, gracias a que sabe que se encuentra monitorizado en todo momento, y que esta monitorización minimiza el impacto en su vida cotidiana. Se trata de un objetivo ambicioso, con respecto al cual, no solo se orienta el desarrollo de la aplicación, sino que se debe orientar todos los componentes que intervienen en el proyecto.

- Monitorización Remota y Continua

El paciente podrá moverse con total libertad y deberá seguir siendo monitorizado, de tal forma que la información se haga llegar, en la medida de lo posible, al personal medico. En el caso de que el dispositivo sea incapaz de ponerse en contacto con el servidor, los datos se deberán almacenar localmente, para ser enviados posteriormente.

- Detección de Situaciones de Riesgo

El dispositivo deberá ser capaz, a través del análisis de los datos, de detectar situaciones de riesgo del paciente. Las situaciones de riesgo serán configuradas por el personal medico en función del paciente que este utilizando el dispositivo; y en el caso de ser detectadas, generaran notificaciones que deberán hacerse llegar al personal medico con una mayor prioridad sobre

los datos.

- Reducción de Costes

La monitorización debe ser pasiva, en la cual no haga falta la intervención del personal medico; mientras que la detección de situaciones de riesgo debe ser activa, de tal forma que cuando se detecte esta situación, el personal medico deberá ser avisado. Este nuevo método de supervisión, permite que un solo miembro del personal medico pueda supervisar un mayor numero de pacientes.

- Localización

En el caso de que el paciente salga fuera del complejo residencial, y dentro de la medida que sea posible, este deberá estar continuamente monitorizado, pero también localizado. De tal forma que la localización de los pacientes también será transmitida al personal medico.

ESTUDIO DE LA TECNOLOGIA DISPONIBLE

1. Introducción

En este capítulo se va a realizar un estudio de la tecnología disponible para la correcta ejecución del proyecto, abordando tanto la parte software como hardware de la solución. Este estudio es vital debido a que una mala elección en el software puede conllevar a disponer de una muy buena aplicación con un hardware inadecuado, y viceversa; y llevar al fracaso la totalidad del proyecto.

El estudio de la tecnología disponible se puede dividir en dos partes bien diferenciadas, la plataforma para el desarrollo, donde vamos a primar la experiencia de las plataformas al tipo de solución que tenemos que abordar; y por otra parte, los dispositivos hardware a utilizar, donde primaremos la capacidad para soportar la plataforma elegida, pero más importante aun será la ergonomía que aporte al usuario de la aplicación, teniendo en cuenta tanto en el peso de los dispositivos, como en la duración de las baterías.

2. Herramientas de Programación para Dispositivos Móviles

Hasta hace no demasiado tiempo, no existían demasiadas herramientas para la programación de dispositivos portátiles, y la gran mayoría eran propietarias del hardware, y no permitían la portabilidad entre sistemas, de tal forma que una aplicación desarrollada en un sistema, debía volver a programarse si se quería implementar en otro sistema, aun siendo similar. El avance de la tecnología y la reducción de costes de los dispositivos, provocaron el auge en la aparición de nuevos dispositivos y de las posibilidades que estos podían ofrecer, haciendo necesaria la aparición de nuevas herramientas de programación, que solventaran los problemas anteriormente citados.

En este punto, fue vital el esfuerzo que hicieron los fabricantes en el desarrollo de sistemas operativos de uso general, que permitió a su vez la aparición de lenguajes y compiladores sobre los estos. De esta forma sistemas operativos como Symbian, Pocket PC o Palm, abrieron las puertas al desarrollo de nuevas aplicaciones.

Presentamos a continuación un gráfico que expone las diferentes tecnologías descritas bajo el fabricante Microsoft. También se acompaña en el gráfico de los entornos de programación necesarios para la realización de los desarrollos.

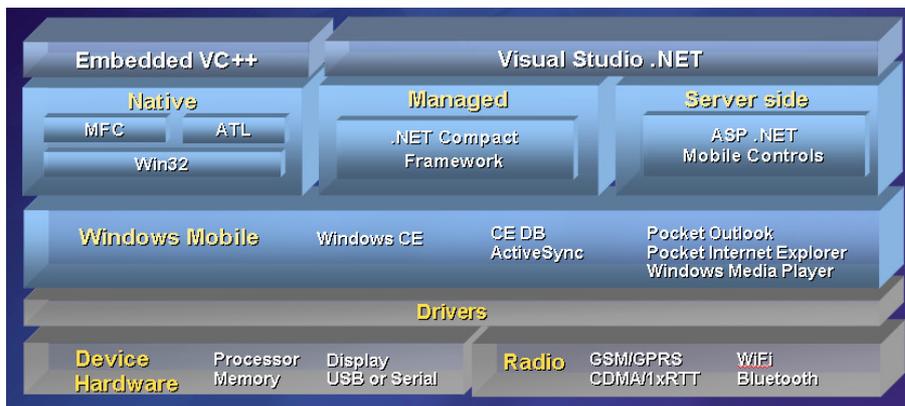


Figura 3. Aplicaciones Nativas y Administradas

2.1. Tipos de Arquitectura de Programación

A continuación se presentan diferentes tipos de arquitecturas de programación actualmente disponibles y que permiten el desarrollo de aplicaciones.

- Basadas en lenguaje de marca y separación.

Esta tecnología permite construir aplicaciones on-line, basadas en navegación, mediante un browser que se ejecuta en el dispositivo y que interpreta un lenguaje de marcas para generar la interfaz con el usuario. Toda la lógica de programación reside en el lado del servidor. Ejemplos de esta tecnología son WAP e i-Mode en teléfonos convencionales, y de lenguajes basados en HTML para teléfonos más avanzados o PDAs, que disponen de navegadores estándar.

La gran ventaja de este tipo de lenguajes es la gran portabilidad de aplicaciones, ya que éstas realmente no residen en el dispositivo, sino en un servidor remoto. Por el contrario, la gran desventaja, es la falta de integración de la aplicación con el hardware, no permitiendo integrar la aplicación con el hardware del dispositivo.

- De aplicaciones y bibliotecas “nativas”.

Son SDKs e IDEs que permiten desarrollar aplicaciones nativas que acceden a los servicios proporcionados por el sistema operativo mediante APIs. Ejemplos de esta tecnología son Embedded Visual C++, SDK Symbian, y SDK PalmOS (estos dos últimos para el desarrollo de aplicaciones para dispositivos bajo los sistemas operativos Symbian y PalmOS respectivamente). Una aplicación nativa es una aplicación que ejecuta código máquina para el procesador y sistema operativo concreto de una familia de dispositivos móviles. Dicho código se encuentra dentro de un fichero ejecutable residente en el sistema de ficheros del dispositivo. Este tipo de aplicaciones se escriben en un lenguaje de alto nivel, normalmente C o C++, y normalmente se dispone de IDEs y entornos de emulación.

Los servicios accesibles desde las APIs C o C++ son, entre otros, la gestión de conexiones, la gestión de intercambios, los infrarrojos, el módem, el acceso a la red, Internet, las comunicaciones serie, las notificaciones, la interfaz de usuario, la telefonía, el SMS, el Bluetooth, y el WLAN.

La principal ventaja que presenta este tipo de aplicaciones es que permiten al desarrollador el acceso a todos los recursos y servicios que ofrece el sistema operativo. La gran desventaja es su falta de portabilidad entre diferentes sistemas operativos y dispositivos.

- Aplicaciones Administradas – Entornos de Ejecución.

Esta tecnología permite desarrollar programas que se ejecutan en el dispositivo sobre un entorno de ejecución estandarizado. Se trata de aplicaciones que se ejecutan sobre un software que actúa de intermediario y que abstrae la capa física del dispositivo, proporcionando un entorno de programación uniforme independientemente del hardware que estemos utilizando.

Estas arquitecturas de programación ponen a disposición del programador un conjunto de bibliotecas que implementan las funciones típicas de los sistemas operativos, pudiendo realizar de forma muy sencilla tareas que serían muy complicadas con las bibliotecas nativas, vistas en el punto anterior.

Su ventaja principal su portabilidad, ya que solo es necesario disponer de un entorno de ejecución compatible para cada dispositivo para poder ejecutar la aplicación. Por el contrario la utilización de este tipo bibliotecas genéricas, no permite optimizar el rendimiento de las aplicaciones provocando

que se requiera una mayor disponibilidad de recursos, que con una aplicación nativa.

Dentro de este tipo de plataformas tenemos J2ME (Java 2 Micro Edición) y .Net Compact Framework. Es en este último en el lenguaje que se ha desarrollado el proyecto, y sobre el cual ampliaremos información en posteriores capítulos.

3. Dispositivos Móviles

Esta sección presenta los dispositivos existentes en el mercado, que cumplen con los requisitos de movilidad y comunicaciones que requiere el proyecto. Se ha decidido incluir en este capítulo los diferentes sistemas operativos que corren en estos dispositivos, ya que en función de estos, las capacidades de programación serán muy diferentes.

Por tanto a continuación se exponen una clasificación de los dispositivos disponibles, y posteriormente una clasificación de los sistemas operativos que nos podemos encontrar.

3.1. Tipos de dispositivos móviles.

No es fácil encontrar una clasificación para discriminar los diferentes dispositivos móviles existentes en el mercado, ya que, en ocasiones, no está clara la diferencia entre ellos. Por tanto hemos decidido clasificarlos por la funcionalidad principal para la que son empleados. A continuación se muestra la clasificación elaborada:

- Teléfono móvil. El teléfono móvil consiste en un dispositivo de comunicación electrónico con las mismas capacidades básicas de un teléfono de línea telefónica convencional. Además de ser portátil, es inalámbrico al no requerir cables conductores para su conexión a la red telefónica.
Características principales: Orientado a voz. Incorpora SMS, EMS, MMS.
- Smartphone. Un Smartphone (en español teléfono inteligente) es un dispositivo electrónico de mano que integra la funcionalidad de un teléfono móvil o similar, con otra serie de funciones para comunicarse a través de Wifi y Bluetooth. Así se dispone de conexión a Internet, y este soporte permite el envío de mensajería y e-mails.
Características principales: Muy similar a un teléfono móvil, pero con tecnologías mejoradas de mensajería y navegación por Internet.



Figura 4. Diferentes Modelos de SmartPhones

- PDA, del inglés Personal Digital Assistant, (Ayudante personal digital) es un computador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura. Hoy en día se puede usar como una computadora doméstica (ver películas, crear documentos, juegos, correo electrónico, navegar por Internet, escuchar música, etc.). Características principales: Orientado a datos. Pantalla grande, entrada mejorada de datos, amplio rango y sistemas de comunicaciones.



Figura 2. Diferentes Modelos de PDA

Hemos decidido no incluir en la clasificación otros dispositivos, o bien por no cumplir con requisitos de movilidad, tamaño reducido o bien por su falta de procesamiento. En particular, se han excluido ordenadores portátiles, TabletPC, teléfonos móviles sin capacidad para ejecutar programas, agendas simples, etc.

4. Sistemas Operativos para Dispositivos móviles

A continuación se exponen los diferentes Sistemas Operativos disponibles para dispositivos portátiles. Todos ellos se acompañan de herramientas para el desarrollo de aplicaciones, aunque la madurez de estas herramientas es muy diferente entre sí, debido en ocasiones al tiempo que el sistema operativo se encuentra en el mercado, pero principalmente, al apoyo que le han brindado los desarrolladores.

- Palm OS: Hoy en día mantenido casi en solitario por Palm, pero que hasta hace poco ha tenido importantes fabricantes como Sony.
- Symbian: Se trata del sistema que podemos encontrar en los últimos modelos de Nokia, y que ha sufrido una gran evolución en los últimos años.
- Windows Mobile: Denominado Pocket Pc en sus versiones previas, se trata del sistema operativo de Microsoft para dispositivos portátiles. Se trata de un sistema muy extendido, y que dispone de versiones tanto para PDA como para SmartPhones. Es utilizado por los principales fabricantes como HP, Acer, Dell o HTC.
- Blackberry: Sistema operativo desarrollado por Research In Motion, más orientado a Smartphones que a PDAs, pero que han copado una parte importante del mercado corporativo durante muchos años, y han contribuido con importantes avances tecnológicos en el desarrollo de tecnologías móviles
- Linux: Sistema operativo omnipresente, que también podemos encontrar instalado en PDAs. En concreto en el modelo Zaurus de la marca Sharp.
- Iphone OS: Se trata de uno de los últimos sistemas operativos que han salido al mercado, aunque solo se distribuye con el dispositivo Iphone de la marca Apple, su salida al mercado ha sido toda una revolución.

5. Sensores

Este capítulo muestra los diferentes tipos de sensores que hemos encontrado en el mercado y que permiten recoger diferentes valores biométricos de los pacientes. Hay que decir que existen un gran número de sensores, pero solo hay unos pocos que nos permitan establecer una comunicación inalámbrica, y menos aun con una comunicación inalámbrica que no utilice un protocolo propietario y sobre la cual podamos realizar una programación. Por tanto, debido a la escasez de los sensores encontrados, a continuación se muestran los productos más significativos.

5.1. Características de los sensores

Los sensores necesarios para el proyecto deben cumplir unos requisitos específicos a la tarea que van a desempeñar, tengamos en cuenta que el tipo de monitorización a realizar, va a ser una

monitorización continua en el tiempo y que debe ser ergonómica, y por tanto los dispositivos deben tener características adecuadas para su buen funcionamiento:

- **Tamaño reducido**

El sensor debe ser de un tamaño no demasiado grande, incluida la alimentación necesaria. Este tamaño podrá ser diferente, en función del lugar donde se deba colocar el sensor; de tal forma que el perjuicio en el día a día del paciente sea lo menor posible.

- **Inalámbrico**

Se trata de una característica fundamental desde el punto de vista psicológico del paciente. El objetivo es que el paciente se sienta lo mas cómodo posible, y la falta de cables entre el sensor y el dispositivo PDA, se orienta en ese sentido. En la siguiente sección hablaremos de las tecnologías inalámbricas y de sus características.

- **Inocuo**

La monitorización debe ser continua, por tanto no es recomendable sensores que puedan provocar, por ejemplo lesiones o quemaduras, por aplicación prolongada en el paciente.

- **Consumo mínimo de batería**

No es recomendable un dispositivo que nos obligue a conectarnos a alimentación cada poco tiempo y perder así el requisito de movilidad del proyecto. Por tanto el sensor debe optimizar al máximo la batería, tanto en la detección de los valores biométricos, como en el envío vía radio de dichos valores.

5.2. Tipos de Sensores.

En este estudio, hemos encontrado diferentes tipos de sensores, los cuales son capaces de leer diferentes tipos de valores biométricos del paciente. A continuación se muestra una descripción de las funcionalidades de dichos sensores:

- **EMG/GSR**

La electro miografía, EMG o miograma es una técnica de diagnóstico médico consistente en un estudio neurofisiológico de la actividad bioeléctrica muscular. Clásicamente, el mismo término EMG engloba también a la electro neurografía (el estudio de los nervios que transmiten la orden motora al aparato muscular) si bien en la actualidad se usa cada vez más en este sentido la palabra electroneuromiografía (ENMG). La técnica consiste en la aplicación de pequeños electrodos de bajo voltaje en forma de agujas en el territorio muscular que se desea estudiar, midiendo la respuesta y la conectividad entre los diferentes electrodos.

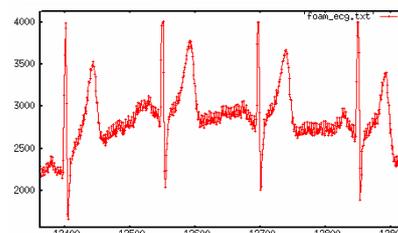


Figura 5. Cardiograma

- EMG/GSR Detección de estrés

Se trata de la tecnología anterior que combinada con otras mediciones permite la detección del estrés. Se utiliza la medición de la tensión muscular en reposo como indicativo de estrés; acompañándolo de la medición sobre la resistencia de la piel (GSR) que varía con la producción involuntaria del sudor como resultado de la tensión.

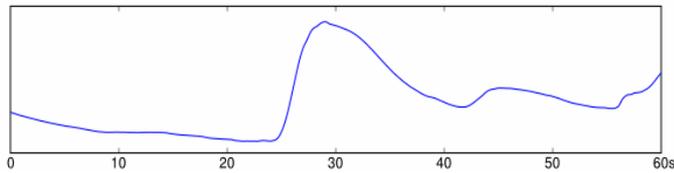


Figura 6. Grafico GSR

- Pulsioximetro

Como su nombre indica, este dispositivo permite la medición de varios valores del paciente. Por una parte es capaz de medir las pulsaciones por minuto, y por otra parte, permite detectar la saturación de oxígeno de hemoglobina arterial. La combinación de estas lecturas permite correlacionar la respiración con la frecuencia cardiaca, y detectar un amplio rango de problemas cardiacos, además de detectar hipo e hiper volemia.



Figura 7. HealthePod

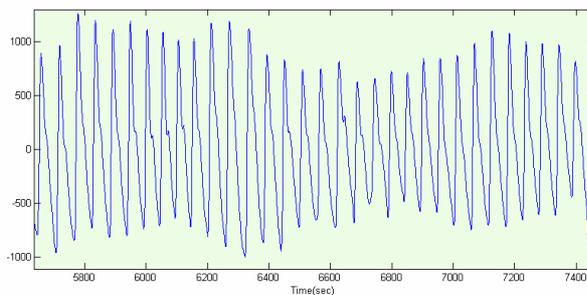


Figura 8. Grafico Pulsioximetro

- **Acelerómetro 3 ejes**

Se trata de un sensor que permite detectar la movilidad de un paciente, incluso detectar la orientación del paciente. Suele utilizarse para la detección de caídas, o para la detección de los niveles de actividad del paciente.

- **Audio**

Se trata del conjunto de sensores que permite la grabación de sonidos del corazón, pulmones, etc. que permiten la detección de cuadros de asma a través del análisis de frecuencias.

- **Termómetro**

Permiten monitorizar la temperatura interna del cuerpo, y que por si solo no aportan demasiada información, pero que combinados con otros sensores, pueden detectar diferentes cuadros médicos.

Por supuesto, también existen en el mercado, dispositivos multisensor aunque suelen estar comercializados para cumplir con un objetivo particular, como por ejemplo programas de adelgazamiento, o de entrenamiento.

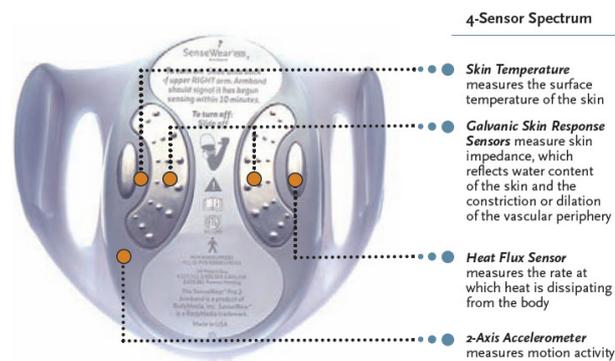


Figura 9. SenseWear BMS

6. Redes

Presentamos a continuación los tipos de redes que nos podemos encontrar en el mercado, y que son aplicables a nuestro proyecto, en función de los tipos de PDA y sensores expuestos anteriormente. Bluetooth y Wifi cubren necesidades distintas en los entornos domésticos actuales, desde la creación de redes y las labores de impresión, a la transferencia de ficheros entre PDAs y ordenadores personales.

6.1. Bluetooth

Bluetooth se utiliza en un gran número de productos tales como teléfonos, impresoras, módems y

auriculares. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda, creando una Personal Area Network (PAN). Su uso más común está integrado en teléfonos y PDAs, bien por medio de unos auriculares Bluetooth o en transferencia de ficheros.

Bluetooth tiene la ventaja de simplificar el descubrimiento y configuración de los dispositivos, ya que éstos pueden indicar a otros los servicios que ofrecen, lo que redundaría en la accesibilidad de los mismos sin un control explícito de direcciones de red, permisos y otros aspectos típicos de redes tradicionales.

Una de las principales ventajas de Bluetooth es su bajo consumo, lo que le permite estar integrado en dispositivos de pequeño tamaño, con un uso de baterías en línea con el tamaño del dispositivo. Por el contrario, una de sus desventajas es el alcance, ya que en su configuración estándar supone aproximadamente unos 10 metros; aunque existen implementaciones que disponen de alcances de hasta 100 metros, por supuesto penalizando el consumo de la batería. Además esta tecnología trabaja en una banda libre, y por tanto no es necesario el pago de licencias por su uso.

6.2. Wifi

Wifi es similar a la red Ethernet tradicional, y como tal, el establecimiento de comunicación necesita una configuración previa. Utiliza el mismo espectro de frecuencia que Bluetooth con una potencia de salida mayor que conlleva a conexiones más sólidas. A veces se denomina a Wifi a la "Ethernet sin cables". Aunque esta descripción no es muy precisa, da una idea de sus ventajas e inconvenientes en comparación a otras alternativas. Se adecua mejor para redes de propósito general: permite conexiones más rápidas, un rango de distancias mayor y mejores mecanismos de seguridad.

Actualmente Existen diversos tipos de Wifi, basado cada uno de ellos en un estándar IEEE 802.11 aprobado. Son los siguientes:

- Los estándares IEEE 802.11b e IEEE 802.11g disfrutan de una aceptación internacional debido a que la banda de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbps y 54 Mbps, respectivamente.
- En la actualidad ya se maneja también el estándar IEEE 802.11a, conocido como Wifi 5, que opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpios. La banda de 5 GHz ha sido recientemente habilitada y, además no existen otras tecnologías (Bluetooth, microondas, ZigBee, WUSB) que la estén utilizando, por lo tanto existen muy pocas interferencias. Su alcance es algo menor que el de los estándares que trabajan a 2.4 GHz (aproximadamente un 10%), debido a que la frecuencia es mayor (a mayor frecuencia, menor alcance).
- Un primer borrador del estándar IEEE 802.11n que trabaja a 2.4 GHz a una velocidad de 108 Mbps. Sin embargo, el estándar 802.11g es capaz de alcanzar ya transferencias a 108 Mbps, gracias a diversas técnicas de aceleramiento. Actualmente existen ciertos dispositivos que permiten utilizar esta tecnología, denominados Pre-N, sin embargo, no se sabe si serán compatibles ya que el estándar no está completamente revisado y aprobado.

Para la conexión de dispositivos en una red Wifi, existe la posibilidad de poder configurar estos dispositivos para el acceso a la red, definiendo permisos de acceso, o encriptación de

comunicaciones. A partir de este punto, la comunicación entre dispositivos se puede regir mediante el uso de varios protocolos, siendo el más habitual el TCP/IP, de la misma forma que en las redes cableadas.

Una de las principales ventajas de Wifi es su alcance, siendo este de hasta 300 metros en su estándar, y de kilómetros en algunas implementaciones avanzadas. Por el contrario, tiene un gran inconveniente en el consumo, ya que una comunicación Wifi penaliza la batería de los dispositivos.

7. Conclusiones y Tecnología Elegida

Como se ha podido observar, disponemos en el mercado de numerosas alternativas para el desarrollo del proyecto, pero finalmente solo podemos elegir una, y por tanto hemos elegido la que consideramos que optimiza los objetivos del proyecto. De ahí, que estudiado los objetivos del proyecto, y obtenidas los requisitos que estudiaremos en el proceso de análisis, nos hemos decantado por la tecnología que presentamos a continuación.

Aunque se presenta la tecnología de forma independiente, hay que decir que la decisión se ha realizado de forma conjunta, debido a que el hardware, el sistema operativo y el entorno de desarrollo están muy ligados entre si. Esta ligazón es más acusada aún para dispositivos móviles

7.1. Windows Mobile, Visual Studio .Net y .Net Compact Framework

Se trata de la plataforma de Microsoft para el desarrollo de aplicaciones, no solo para dispositivos móviles. Se compone del Entorno de Desarrollo (IDE) Visual Studio .NET y de .NET Compact Framework, el cual nos proporciona un conjunto de bibliotecas para el desarrollo de aplicaciones, y del motor de ejecución necesario para la ejecución de las aplicaciones en el PDA. En el apéndice A se incluye una descripción más extensa de esta tecnología.

La elección de esta plataforma se debe principalmente a dos motivos.

- El primero y mas importante, es la alta experiencia y gran numero de desarrolladores que utilizan esta plataforma, lo cual ha provocado una rápida evolución de la arquitectura en un periodo corto de tiempo. Además esa base de desarrolladores garantiza la continuidad de la arquitectura a medio plazo.
- El segundo motivo es el gran numero de dispositivos existentes en el mercado con sistemas operativos compatibles con la arquitectura, y en este caso de la familia Windows Mobile o Windows CE .Net. Hay que destacar que estos dispositivos son comercializados por grandes fabricantes de hardware como HP, Acer, Dell, HTC, etc.

Estos motivos, nos dan a entender que se la plataforma de desarrollo elegida, es una buena opción y que permitirá la continuidad del desarrollo a medio y largo plazo.

7.2. Comunicaciones

En función de la lectura de los objetivos del proyecto, disponemos dos escenarios bien diferenciados de comunicaciones; la comunicación entre el sensor y PDA, y la comunicación entre PDA y servidor central. Se trata de dos comunicaciones bien diferenciadas y que requieren especificar requisitos bien diferentes, donde nos debemos estudiar 3 variables principalmente:

- Cobertura

Entendemos la cobertura como el radio de funcionamiento de la comunicación entre dos dispositivos. En este sentido no tendremos problemas en la cobertura entre sensor y PDA, ya que ambos estarán siempre acompañando al paciente; pero en la comunicación entre PDA y servidor, podemos tener problemas en función de la posición del paciente.

- Cantidad Datos

Se tratara de un objetivo que se intentará minimizar siempre, pero la tecnología de comunicación elegida puede ser, o no, un cuello de botella en las transmisiones.

- Consumo de Energía

Probablemente sea el factor más influyente de los 3, ya que en los dispositivos PDA, este recurso es más bien escaso, y habrá que condicionar la temporalidad del número de comunicaciones, para ahorrar el máximo de energía.

- Coste

No se trata de una variable técnica, pero que debe tenerse en cuenta para el desarrollo del proyecto.

7.2.1. Comunicación entre Sensor y PDA - Bluetooth

Por supuesto los criterios arriba indicados, se deben tener en cuenta tanto para el PDA como para el Sensor, de ahí que hayamos elegido la comunicación Bluetooth para esta comunicación. Esta tecnología cumple con los 3 requisitos arriba indicados, y sobre todo con el último, el consumo de energía, haciendo que los dispositivos sensor y PDA, no tengan que consumir gran cantidad de este recurso para comunicarse.

7.2.2. Comunicaciones entre PDA y Servidor Central

La mejor opción de comunicaciones en este punto, hubiera sido una comunicación 3G/GPRS, la cual cumple con todos los requisitos, incluso el de cobertura, excepto con el requisito del coste. Por tratarse de una tecnología que se factura en función de los Kilobytes enviados y en una aplicación cuya principal función es la monitorización, se ha estimado que el coste iba a ser elevado y por ello se ha relegado a un segundo término. No obstante, se trata de una tecnología de comunicaciones muy valida, por lo que estaremos pendientes de la evolución de sus costes en un futuro. Estas consideraciones han motivado que finalmente nos hayamos decantado por Wifi, que es la siguiente tecnología con mayor cobertura de comunicaciones.

7.3. Hardware. PDA y Sensores

Forman parte de este punto, el PDA y los sensores.

En el caso del PDA es requisito que utilizara un sistema operativo compatible con la Arquitectura .Net Framework, o bien, que tuviera un entorno de ejecución para esta arquitectura. Finalmente nos decidimos por el modelo HTC Touch Cruise, de la marca HTC (High Tech Computer). Se trata de un PDA con características avanzadas que dispone de las comunicaciones necesarias para el proyecto. De esta forma, se dispone de Bluetooth, Wifi y GPS. Se trata además de un dispositivo que dispone de GSM/GPRS/3G, que permite realizar también llamadas, envío de mensajes SMS y conexión a Internet mediante 3G o GPRS, características no necesarias para el proyecto, pero que serán de gran utilidad para una posterior versión de este.



Figura 10. HTC Touch Cruise

A continuación se detallan las especificaciones técnicas del dispositivo PDA.

- Sistema Operativo: Windows Mobile® 6 Professional
- Procesador: Qualcomm® MSM7200™, 400MHz
- Memoria: ROM: 256MB ,RAM: 128MB DDR SDRAM
- Teléfono Integrado: HSDPA/UMTS/GSM/GPRS/EDGE
- Conectividad Inalámbrica: Bluetooth® 2.0, Wi-Fi®: IEEE 802.11 b/g
- Ranuras de Expansión: MicroSD™ HC
- GPS Integrado: Si
- Puertos: USB, Bluetooth 2, WiFi 802.11b/g, MicroSD
- Cámara: 3.0 mega-pixel auto focus / VGA CMOS
- Pantalla: 2.8
- batería: Extraíble, polímero de litio, 1350 mAh
- Autonomía: PDA 10h; Conversación 4h; en espera 200h.
- Dimensiones: 110 x 58 x 15,5 mm
- Peso: 130g
- Colores: 65535

En el caso del sensor, optamos por un Pulsioxímetro Nonim 4100, el cual permite la detección del pulso cardíaco y el nivel de saturación de oxígeno en sangre. Se trata de un dispositivo que permite la comunicación mediante Bluetooth, y uno de los únicos que encontramos en el que el interfaz de comunicación era abierto, y por tanto fue posible obtener sus especificaciones.



Figura 11. Pulsioxímetro Nonim

Detallamos a continuación las especificaciones técnicas del sensor:

- Rango de Saturación de Oxígeno(SpO₂): 0 a 100%
- Rango Frecuencia Cardíaca: 18 a 300 pulsos por minuto.
- Precision SpO₂ 70-100%
- Precision Frecuencia Cardíaca: 40 a 240
- Temperatura: -20° a +50° C°
- Humedad: 10% a 90% sin condensación
- baterías: 2AA
- duración batería: 120 horas operación continua.
- Especificación Bluetooth: 1.1
- Dimensiones: 3" x 2.74" x 1.34"
- Peso: 125g con baterías instaladas

ANALISIS

1. Introducción

El análisis consiste en la comprensión y modelado de la aplicación y del dominio en el cual se desempeña. La entrada inicial de la fase de análisis es una descripción del problema que hay que resolver y proporciona una descripción general del sistema propuesto.

Inicialmente se estudia la especificación del sistema, teniendo en cuenta el contexto en el cual será aplicado. El objetivo de este apartado es reconocer los elementos básicos del programa tal como lo percibe el usuario.

La salida del análisis es un modelo formal, que muestra de una forma abstracta y resumida, los diferentes elementos del dominio, y que sirve como base para la fase de diseño del sistema.

2. Requisitos Software

El objeto de este apartado es el de identificar la funcionalidad de la aplicación que vamos a desarrollar. Para ello se deben obtener los requisitos que esta debe cumplir, obtenidos estos a través de las indicaciones de los tutores del proyecto.

El objeto del PFC es obtener una aplicación que permita la monitorización pasiva y remota de pacientes, y que esta monitorización se realice de la forma menos intrusiva posible, haciendo que el paciente mantenga una calidad de vida adecuada.

El siguiente apartado proporciona una idea descripción general del proyecto que nos permite introducirnos en la funcionalidad que debe desempeñar y por tanto obtener la especificación de requisitos. Se define también el contexto de uso, así como perfil de usuario final de la aplicación y perfil de los pacientes que se van a monitorizar.

Posteriormente, comprendida la problemática a resolver, pasaremos a describir uno a uno los requisitos funcionales y no funcionales, actores y casos de uso, hasta obtener un modelo conceptual de la aplicación a desarrollar.

2.1. Descripción General de la Aplicación

De la entrevista con los profesores se extrajeron multitud de datos que permitieron definir la funcionalidad de la aplicación, y por tanto hacerme una amplia idea del alcance de ésta. Si bien, previo a un segunda entrevista, se tuvo que realizar un sondeo de mercado sobre los diferentes sensores no intrusivos que se disponían, con el fin de poder orientar un segundo grupo de requisitos. A continuación explicamos el funcionamiento general de la aplicación

La aplicación será utilizada en complejos residenciales donde el objetivo es la monitorización pasiva de cada paciente. Los pacientes llevarán adosados a su cuerpo los diferentes sensores que recogerán sus valores biométricos. Esta información será recogida por la aplicación a desarrollar, que correrá en un dispositivo Personal Data Assistant (PDA), y posteriormente será enviada a un servidor central (a partir de ahora Central). A través de esta Central, el personal medico podrá conocer el estado de cada uno de sus pacientes.

Además el PDA podrá ser configurado con Situaciones de Riesgo, las cuales, a través de la evaluación de los datos recogidos, y comprobando que estos no alcancen ciertos límites, permitirán determinar el estado de los pacientes. Si se da el caso de que el estado del paciente se agrava, entonces el PDA podrá enviar una Notificación con el nuevo estado del paciente, para que sea conocido por el personal médico.

A continuación mostramos una ilustración que define toda la infraestructura de la solución, desde los sensores del paciente, hasta que el personal médico recibe la información. Si bien el objeto del proyecto es únicamente el desarrollo de la aplicación que correrá en el PDA del paciente.

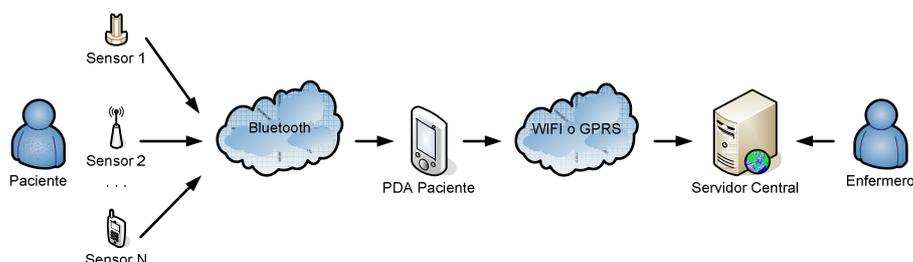


Figura 1. Descripción General del PFC

Como se ha indicado anteriormente, esta PFC es originaria del proyecto AIVI, el cual nos debía proporcionar la especificación de la interfaz de la Central. Finalmente por retardos en la entrega de la documentación y de la Central en sí, decidimos realizar una aplicación que nos resolviera esta necesidad. De la misma forma, los diccionarios de datos sobre la configuración de los pacientes, el registro de datos de los sensores, y los cambios en las notificaciones también son simulados, no cumpliendo la especificación del proyecto AIVI.

2.1.1. Contexto de la aplicación

Uno de los objetivos es que la monitorización continua del paciente sea lo más ergonómica posible, y permita al paciente mantener su calidad de vida. Por tanto esta monitorización se realizará tanto en recintos cerrados, como en los alrededores del complejo residencial.

Esto supone que la disponibilidad de redes de comunicación no será siempre la adecuada, y que deberemos de disponer de almacenamiento en el PDA donde poder almacenar una cola de envío de los datos y notificaciones.

De la misma forma, y aprovechando las características de Global Positioning System (GPS) que actualmente disponen algunos PDAs, cuando el paciente se encuentre fuera de recintos cerrados, el PDA recogerá la información sobre la posición del paciente, como si de un sensor adicional se tratara, con el objeto de notificar al personal médico de la localización de este.

2.1.2. Roles

A partir de la información recogida en la entrevista se detectaron dos roles en el uso de la aplicación: Personal Sanitario y Paciente.

El personal sanitario será quien configure las características del paciente en el PDA, y quien evalúe la correcta colocación de los dispositivos en el paciente. No serán personal especialmente cualificado, pero se supone que recibirán la formación específica en el uso de PDA y en el uso de la aplicación. Se trata de personal que conocen el estado de cada paciente y por tanto de los valores físicos a controlar en cada uno de ellos; y por tanto el tipo de sensores más apropiado para utilizar en cada caso.

El paciente no necesitará conocer necesariamente el uso de la aplicación, y por supuesto su configuración. No obstante, podrá tener la oportunidad de conocer su estado en todo momento, tras recibir una formación sobre manejo de PDA y la aplicación.

2.1.3. Servidor Central

La Central esta compuesta por un servicio web que se encargara principalmente de dos tareas. Por una parte dispone de la información de todos los sensores y de todos los pacientes, y proporcionará la configuración de cada uno de los pacientes. Esta configuración se registrará en un fichero XML, que dispone de la información del paciente, con los sensores que le van a monitorizar y las situaciones de riesgo a controlar.

Por otra parte, se encargará de recoger todos los datos recogidos por el PDA para cada paciente, así como las notificaciones sobre cambios en el estado del paciente. De esta forma el personal medico tendrá históricos sobre los datos del paciente, y el estado real del paciente en cada momento; además de la localización del paciente.

2.1.4. Dispositivos Móviles

Son dos los dispositivos hardware a utilizar: los sensores y el PDA. Los sensores son dispositivos que recogerán los valores biométricos del paciente. Como hemos visto antes, existe una gran variedad, y con un amplio espectro de valores a recoger. Por otra parte, tenemos el PDA el cual se encargará de recopilar la información de los sensores y almacenarlos localmente, hasta que se puedan enviar a la Central.

Uno de los requisitos no funcionales del PFC, es la ergonomía del paciente y el mantenimiento de la calidad de vida del paciente, la cual se obtiene a través de su movilidad, y aunque en menor medida minimizar el cableado sobre el cuerpo.

Para cumplir con estos requisitos, se ha optado por la utilización de sensores inalámbricos, que permiten que el sensor sea utilizado en cualquier zona del cuerpo, independientemente de donde se coloque el PDA. La comunicación inalámbrica se realizara a través de Bluetooth, que permitirá desplegar una PAN (Personal Area Network) donde podrán intercambiar datos los sensores y el PDA.

La principal ventaja de esta comunicación, es el bajo uso de la batería que utiliza, lo cual permite

que el desarrollo de la comunicación se pueda realizar durante horas. Por otra parte, no se trata de una comunicación con gran ancho de banda, pero suficiente para los dispositivos que hemos encontrado en el mercado.

En el caso del PDA, este deberá disponer, además de la comunicación bluetooth, de comunicación inalámbrica para comunicarse con la Central. Y en este caso hemos optado por la comunicación Wifi para el envío de datos.

2.2. Funcionalidad de la Aplicación

Analizadas los condicionantes externos que intervienen en la aplicación, vamos a realizar una descripción detallada del funcionamiento de esta, con el fin de obtener la especificación de requisitos.

En un comienzo el personal sanitario deberá configurar el PDA en función de cada uno de los pacientes. Para ello tendrá una aplicación muy sencilla que previa introducción del identificador del paciente, se cargará dicha configuración en el PDA. Esta configuración se concentrará en un fichero, e indicará de forma muy sencilla la configuración correspondiente a los sensores, de tal forma que el personal sanitario identifique que sensores tiene que colocar a cada paciente. Además, el personal sanitario deberá tener la posibilidad de comprobar el correcto funcionamiento de los sensores.

Adicionalmente a cada paciente se le podrán configurar situaciones de riesgo, las cuales estarán compuestas por reglas de riesgo. Estas reglas de riesgo son comparaciones de los datos obtenidos de los sensores contra valores límite configurados en estas.

Una situación de riesgo será activada cuando la evaluación de todas sus reglas de riesgo sea verdadera. En caso de que existan reglas activas, pero otras no, la situación de riesgo se encontrará en un estado intermedio.

Configurado el PDA, comenzará la monitorización del paciente, la cual se realizará en dos vías principalmente: Lectura de Sensores y evaluación.

El PDA tendrá una lista de los sensores a la cuales tendrá que recoger información, cada uno de los cuales configurado con un intervalo de tiempo entre dos lecturas. Llegado el momento el PDA realizará una conexión bluetooth con el sensor, para recoger el dato que dispone en ese momento. Si la lectura ha sido satisfactoria, el PDA procederá a almacenar el dato internamente para que sea enviado posteriormente.

La segunda parte de la monitorización evalúa el estado de las Situaciones de Riesgo, de tal forma que el dato recogido es comparado contra un valor límite para determinar el estado del paciente. En el caso de que la comparación sea positiva, y se haya detectado una situación de riesgo en el paciente, se realizará una notificación al personal sanitario.

Por otra parte, y como si fuera un sensor adicional, en el intervalo de tiempo fijado, el PDA consultará al GPS la localización del paciente, la cual se almacenará localmente para ser enviada posteriormente.

Por último, tenemos el envío de datos y notificaciones al personal médico. El PDA tendrá otro

intervalo de tiempo configurado para el envío de datos a la Central, de tal forma que cuando este se alcance, el PDA procederá a leer uno a uno los datos almacenados para que sean enviados. Una vez que queden confirmados los envíos, los datos serán borrados.

Las notificaciones sobre el cambio de estado del paciente son información mas urgente, y aunque previamente serán almacenadas localmente, su envío será inmediato con el fin de que lleguen lo antes posible al personal medico.

2.3. Requisitos Funcionales

FRQ-0001	Datos de Paciente
Dependencias	Ninguno
Descripción	El sistema deberá <i>almacenar el identificador del paciente.</i>

FRQ-0002	Datos de Sensores
Dependencias	Ninguno
Descripción	El sistema deberá <i>almacenar los datos relativos a cada uno de los sensores, como identificador de sensor, tipo, descripción, dirección bluetooth, pin bluetooth e intervalo.</i>

FRQ-0003	Datos de Situaciones de Riesgo
Dependencias	Ninguno
Descripción	El sistema deberá <i>almacenar los datos relativos a cada una de las situaciones de riesgo, como Identificación de la situación de riesgo, descripción, destinatario, el texto del mensaje y sus correspondientes reglas.</i>

FRQ-0004	Datos de Reglas de Riesgo
Dependencias	• [FRQ-0003] Datos de Situaciones de Riesgo
Descripción	El sistema deberá <i>almacenar la información relativa a las reglas de riesgo, como el identificador de la regla de riesgo, identificador de sensor, nombre del dato, operador y el valor límite.</i>

FRQ-0005	Configuración Remota de Paciente
Dependencias	• [FRQ-0001] Datos de Paciente

	<ul style="list-style-type: none"> • [FRQ-0003] Datos de Situaciones de Riesgo • [FRQ-0004] Datos de Reglas de Riesgo • [FRQ-0002] Datos de Sensores
Descripción	El sistema deberá <i>permitir la configuración de toda la información relativa al paciente, recogiéndola de la Central y aplicándola en el PDA.</i>

FRQ-0006	Configuración Local Aplicación
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir la configuración del PDA, excluyendo la configuración del paciente, de forma local a través de un fichero.</i>

FRQ-0007	Cambiar Estado Aplicación
Dependencias	<ul style="list-style-type: none"> • [FRQ-0006] Configuración Local Aplicación
Descripción	El sistema deberá <i>permitir el cambio de estado de la aplicación, activando y deteniendo el reloj de ésta.</i>

FRQ-0008	Recogida información Sensores
Dependencias	<ul style="list-style-type: none"> • [FRQ-0002] Datos de Sensores
Descripción	El sistema deberá <i>recoger la información de los sensores configurados en el intervalo configurado en cada uno de ellos.</i>

FRQ-0009	Almacenamiento Local de Datos de Sensores
Dependencias	<ul style="list-style-type: none"> • [FRQ-0010] Recoger información de GPS • [FRQ-0008] Recogida información Sensores
Descripción	El sistema deberá <i>almacenar localmente la información recogida de cada uno de los sensores.</i>

FRQ-0010	Recoger Información de GPS
Dependencias	<ul style="list-style-type: none"> • [FRQ-0006] Configuración Local Aplicación
Descripción	El sistema deberá <i>tener localizado, siempre que sea posible, al paciente, recogiendo información de este y almacenándola como si fuera un dato de un sensor.</i>

FRQ-0011	Detección de Situaciones de Riesgo
Dependencias	<ul style="list-style-type: none"> • [FRQ-0009] Almacenamiento Local de Datos de Sensores

Descripción	El sistema deberá <i>evaluar las reglas de riesgo asociadas a cada situación de riesgo, y establecer el estado de ésta.</i>
--------------------	---

FRQ-0012	Almacenamiento Local de Notificaciones
Dependencias	• [FRQ-0011] Detección de Situaciones de Riesgo
Descripción	El sistema deberá <i>almacenar localmente las notificaciones derivadas de los cambios de estado de las situaciones de riesgo.</i>

FRQ-0013	envío de Datos a Servidor Central
Dependencias	• [FRQ-0008] Recogida información Sensores
Descripción	El sistema deberá <i>enviar la información almacenada de los sensores al servidor central en el intervalo configurado de tiempo.</i>

FRQ-0014	Envío de Notificaciones al servidor central
Dependencias	• [FRQ-0012] Almacenamiento Local de Notificaciones
Descripción	El sistema deberá <i>enviar la información almacenada de las notificaciones al servidor central inmediatamente después de que la situación de riesgo cambie su estado.</i>

2.4. Requisitos No Funcionales

NFR-0001	Confidencialidad del Paciente
Dependencias	Ninguno
Descripción	El sistema deberá <i>permitir que solo el personal sanitario autorizado pueda acceder a los datos del servidor central.</i>

NFR-0002	PDA con GPS, Wifi y Bluetooth
Dependencias	Ninguno
Descripción	El sistema deberá <i>disponer de las tecnologías de GPS para la localización geográfica del paciente; de Wifi para la comunicación con el servidor central; y de Bluetooth para la comunicación con los diferentes sensores.</i>

NFR-0003	Arquitectura Microsoft .Net y C#
-----------------	---

Dependencias	Ninguno
Descripción	El sistema deberá ser programado bajo la arquitectura Microsoft .Net y el lenguaje C#, debido a la alta compatibilidad de esta arquitectura con el Sistema Operativo Windows Mobile; el cual esta implantado en la mayoría de las PDA disponibles en el mercado

NFR-0004	Servicios Web como interfaz del Servidor Central
Dependencias	Ninguno
Descripción	El sistema deberá permitir la comunicación entre el PDA y el Servidor Central mediante la utilización de Web Services, con el objeto de estandarizar la comunicación

NFR-0005	Facilidad de Uso
Dependencias	Ninguno
Descripción	El sistema deberá facilitar la ergonomía en el uso de la aplicación, con el objeto que este pueda ser utilizado por personal no especializado.

NFR-0006	Abstracción de Sensores
Dependencias	Ninguno
Descripción	El sistema deberá permitir la integración de cualquier sensor Bluetooth, así como permitir el almacenamiento de sus datos y su evaluación en las Situaciones de Riesgo.

NFR-0007	Optimización de la batería
Dependencias	Ninguno
Descripción	El sistema deberá optimizar el uso de la batería, y maximizar el tiempo de uso del dispositivo

3. Casos de Uso

A continuación expongo el Modelo de Casos de Uso de la aplicación para la monitorización de pacientes que expone toda la funcionalidad recogida de los requisitos.

El documento se organiza de la siguiente forma. Primero se exponen los actores participantes en los casos de uso, para acto seguido continuar con la presentación de estos casos en dos diagramas en función de la fuente de los casos de uso. Por ultimo se exponen el detalle de cada uno de los casos de uso.

3.1. Actores

Los actores participantes en los casos de uso son los siguientes:

ACT-0001	Personal Médico
Descripción	Este actor representa <i>el usuario quien se encargará de configurar la aplicación y comprobar su funcionamiento a través del estado de cada uno de sus sensores.</i>

ACT-0002	Reloj
Descripción	Este actor representa <i>a la parte correspondiente del sistema que se encargará de marcar la temporalidad de todos los casos de uso automáticos (sin intervención directa del paciente o el personal medico), y por tanto el que desencadenará éstos.</i>

ACT-0003	Sensor
Descripción	Este actor representa <i>a cada uno de los sensores Bluetooth que dispondremos en la aplicación y de los cuales obtendremos la información para su tratamiento posterior.</i>

ACT-0004	Sensor GPS
Descripción	Este actor representa <i>al sensor que nos proporcionará la información de la localización del paciente.</i>

ACT-0005	Central
Descripción	Este actor representa <i>a la Central donde disponemos las configuraciones de los pacientes y donde entregaremos los datos recogidos.</i>

3.2. Casos de Uso

Presentamos a continuación los casos de uso en función del actor que los desencadena.

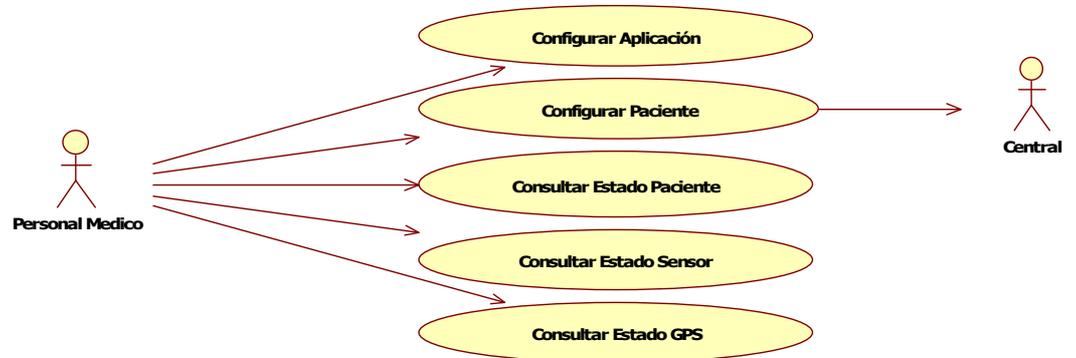


Figura 1. Casos de Uso de Personal Medico

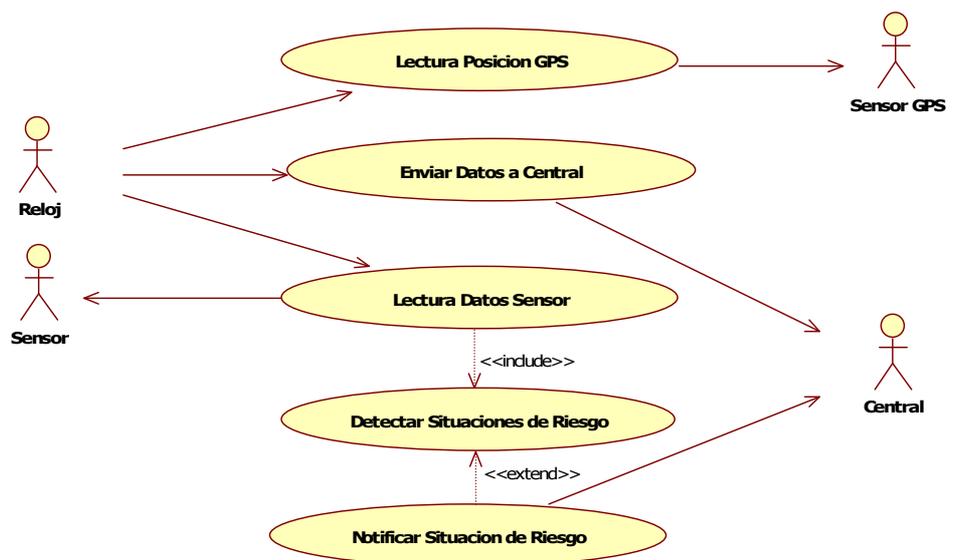


Figura 2. Casos de Uso de Reloj

Aplicación para la monitorización remota de pacientes

Se presenta a continuación un resumen de cada uno de los casos de uso.

- UC-0001: Configurar Aplicación

Este caso de uso representa la personalización de la aplicación con configuración propia de la PDA, independientemente de la configuración del paciente; y que sustituirá a la configuración por defecto de la aplicación.

- UC-0002: Configurar Paciente

Este caso de uso representa la personalización de la aplicación para cada uno de los pacientes de la aplicación, proporcionando configuración sobre sensores e intervalos de lectura; además de situaciones de riesgo con sus reglas correspondientes.

- UC-0003: Consultar Estado Paciente

Este caso de uso representa el interés por parte del personal médico de conocer el estado del paciente a través del estado de las situaciones de riesgo que tienen configuradas.

- UC-0004: Consultar Estado Sensor

Este caso de uso representa el interés en conocer el buen funcionamiento de los sensores asociados al paciente, conociendo su estado en todo momento.

- UC-0005: Consultar Estado GPS

Este caso de uso representa el interés en conocer el buen estado de funcionamiento del GPS, conociendo su estado en todo momento y conociendo la última posición recogida.

- UC-0006: Lectura de Datos de Sensor

Este caso de uso representa la lectura por parte del sistema de los datos de los diferentes sensores y su almacenamiento local.

- UC-0007: Lectura de Posición de GPS

Este caso de uso representa la lectura por parte del sistema de la posición del sensor GPS y su almacenamiento local.

- UC-0008: Detectar Situaciones de Riesgo

Este caso de uso represente la evaluación de los datos recogidos en el sistema con el fin de identificar cambios en las situaciones de riesgo y por tanto cambios de estado del paciente.

- UC-0009: Enviar Datos a la Central

Esta caso de uso representa el envío de los datos recogidos en el PDA para que sean interpretados por el personal médico

- UC-0010: Notificar Situación de Riesgo

Este caso de uso representa el envío de los cambios de estado de las situaciones de riesgo, y por tanto la notificación de alerta al personal médico de los cambios de estado del paciente.

3.2.1. Descripción de los Casos de Uso

Se detalla a continuación cada uno de los casos de uso con sus diagramas de secuencia.

UC-0001	Configurar Aplicación	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se quiera configurar parámetros correspondientes a la aplicación, que son independientes del paciente a monitorizar.</i>	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor Personal Médico (ACT-0001) iniciará la aplicación en el dispositivo Pda.
	2	El sistema <i>cargará el fichero de configuración general de la aplicación</i>
	3	El sistema <i>validará las opciones en busca de errores o de alguna inconsistencia</i>
	4	El sistema <i>aplicará las opciones en el sistema</i>
Postcondición	El sistema estará en funcionamiento	
Excepciones	Paso	Acción
	3	Si <i>alguna de las opciones no es válida</i> , el sistema <i>no tendrá en cuenta el valor de la opción</i> , a continuación este caso de uso continúa
Comentarios	Ninguno	

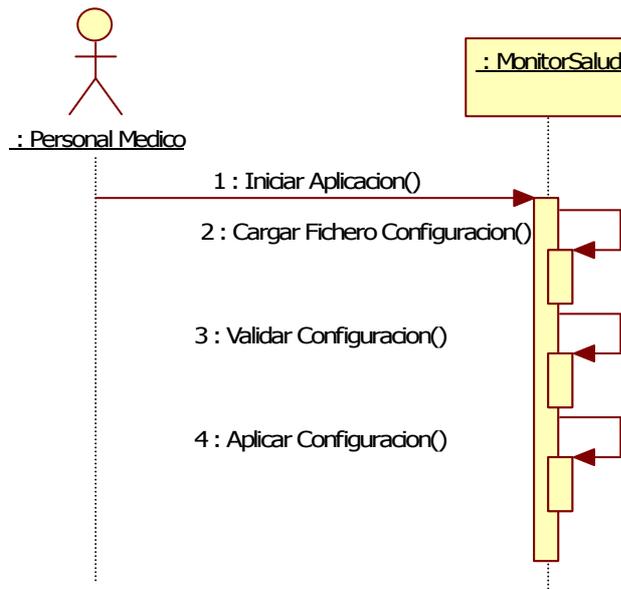


Figura 3. Diagrama Secuencia UC-0001

UC-0002	Configurar Paciente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se configure los parámetros asociados a un paciente, como sensores e intervalos de lectura, y por situaciones de riesgo con sus correspondientes reglas.	
Precondición	El sistema esta en estado iniciado con un paciente configurado	
Secuencia normal	Paso	Acción
	1	El actor Personal Médico (ACT-0001) indicará al sistema que quiere configurar un paciente en la aplicación.
	2	El sistema solicitará a la Central el fichero de configuración para el paciente indicado
	3	El sistema validará la configuración en busca de errores o de alguna inconsistencia
4	El sistema aplicará la configuración del paciente en la aplicación.	
Postcondición	El sistema queda en estado detenido y con la nueva configuración aplicada.	

Excepciones	Paso	Acción
	3	Si la configuración no es válida, el sistema el sistema descartará el fichero de configuración, a continuación este caso de uso queda sin efecto
Comentarios	Ninguno	

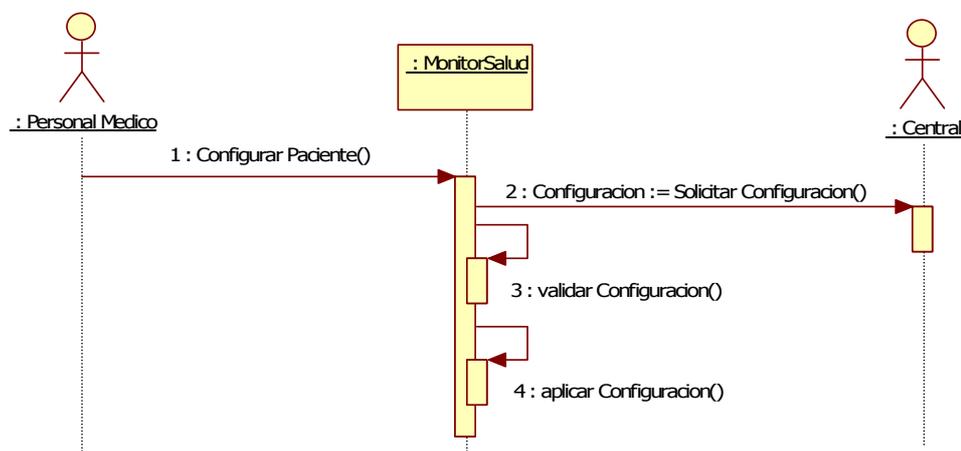


Figura 4. Diagrama de Secuencia UC-0002

UC-0003	Consultar Estado Paciente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se desee consultar el estado de un paciente, estado que estará determinado por el estado de las situaciones de riesgo asociados a éste.	
Precondición	El sistema esta en estado iniciado y con un paciente configurado	
Secuencia normal	Paso	Acción
	1	El actor Personal Médico (ACT-0001) indica que quiere conocer el estado del paciente
	2	El sistema, para cada situación de riesgo, recoge el estado de cada una de las reglas de riesgo que la componen.
	3	El sistema calcula el estado de la situación de riesgo, en función del estado recogido de las reglas de riesgo de la que se componen la situación.
	4	El sistema calcula el estado del paciente, en función del estado

	de las situaciones de riesgo que tiene configuradas.
Postcondición	
Comentarios	Ninguno

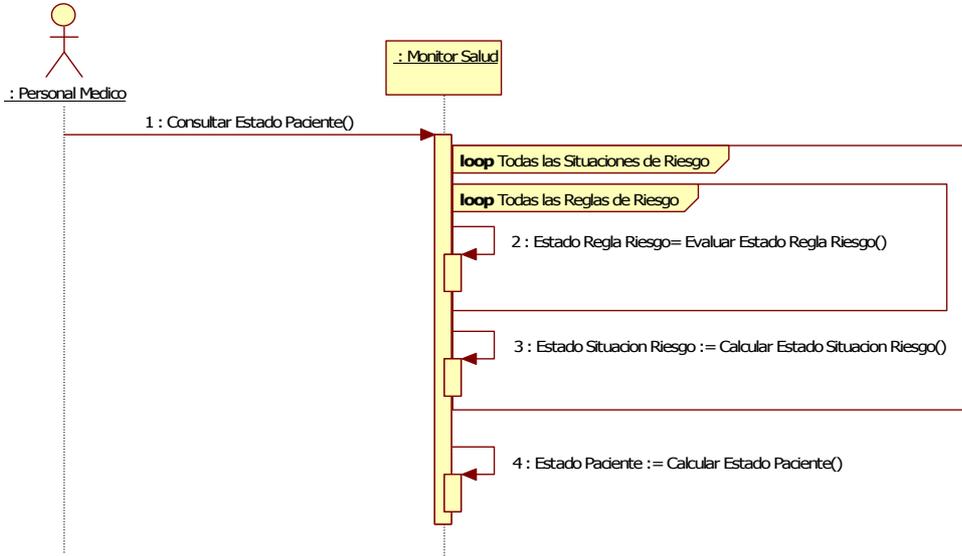


Figura 5. Diagrama de Secuencia UC-0003

UC-0004	Consultar Estado Sensores	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se desee consultar el estado de los sensores, el cual estará determinado en función de la tarea que estén realizando en cada momento y del resultado de su última lectura.</i>	
Precondición	El sistema esta en estado iniciado y con un paciente configurado	
Secuencia normal	Paso	Acción
	1	El actor Personal Médico (ACT-0001) indica que quiere conocer el estado de los diferentes sensores.
	2	El sistema consulta a cada uno de los sensores por su estado actual
Postcondición		
Excepciones	Paso	Acción
	-	-

Comentarios	Ninguno
--------------------	---------

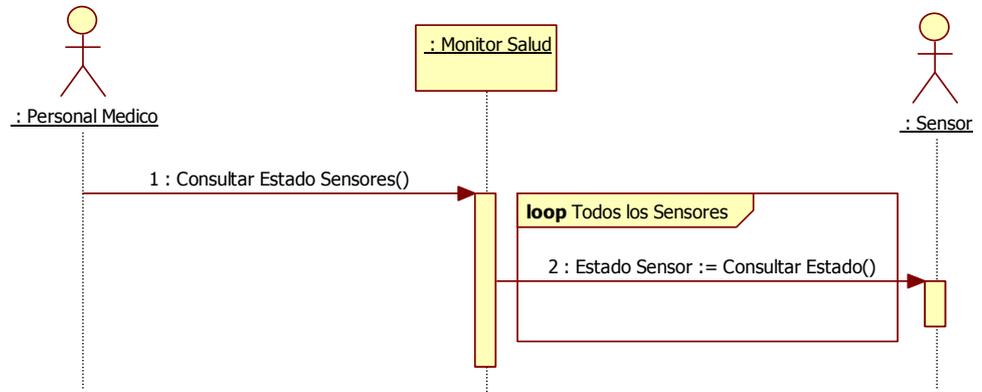


Figura 6. Diagrama de Secuencia UC-0004

UC-0005	Consultar Estado GPS	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se desee consultar el estado del GPS, el cual estará determinado en función de la tarea que este realizando en cada momento y del resultado de su última lectura.</i>	
Precondición	El sistema esta en estado iniciado, con un paciente y Gps configurado	
Secuencia normal	Paso	Acción
	1	El actor Personal Médico (ACT-0001) indica que quiere conocer el estado del GPS.
	2	El sistema consulta al GPS su estado actual
Postcondición		
Excepciones	Paso	Acción
	-	-
Comentarios	Ninguno	

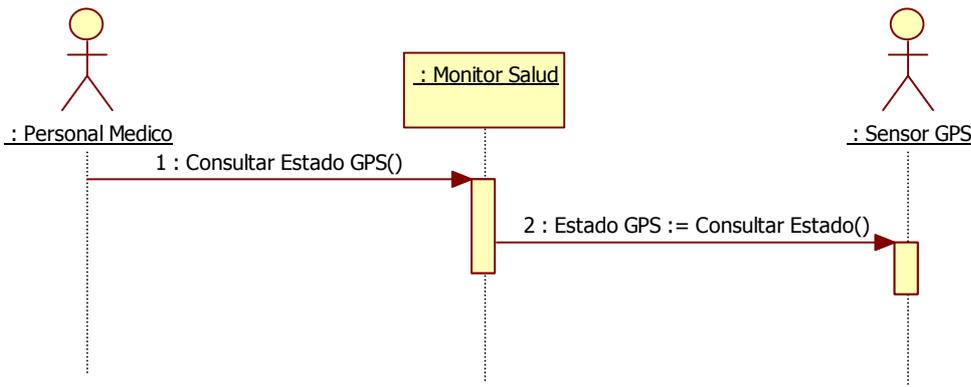


Figura 7. Diagrama de Secuencia UC-0005

UC-0006	Lectura de Datos Sensor	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el reloj indique que se debe realizar la lectura de un sensor en particular, y que terminara cuando dicho dato quede almacenado en caso de tener una lectura satisfactoria.</i> o durante la realización de los siguientes casos de uso: [UC-0008] Detectar Situaciones de Riesgo	
Precondición	El sistema esta en estado iniciado y con un paciente configurado	
Secuencia normal	Paso	Acción
	1	El actor Reloj (ACT-0002) indica que el sensor correspondiente tiene que comenzar a leer.
	2	El sistema consulta el estado del sensor para comprobar si esta disponible
	3	El sistema consulta al sensor por el ultimo dato que ha recogido
	4	Si la lectura es correcta, el sistema almacena la información en la base de datos local
	5	Se realiza el caso de uso Detectar Situaciones de Riesgo (UC-0008)
Postcondición	El sensor se encuentra en estado de lectura Ok y el dato queda almacenado.	
Excepciones	Paso	Acción
	2	Si el sensor no responde, el sistema indica que el sensor no esta al alcance, a continuación este caso de uso queda sin efecto

	3	Si se produce un fallo de comunicación, el sistema indica que ha habido un fallo en la comunicación con el sensor, a continuación este caso de uso queda sin efecto
Comentarios	Ninguno	

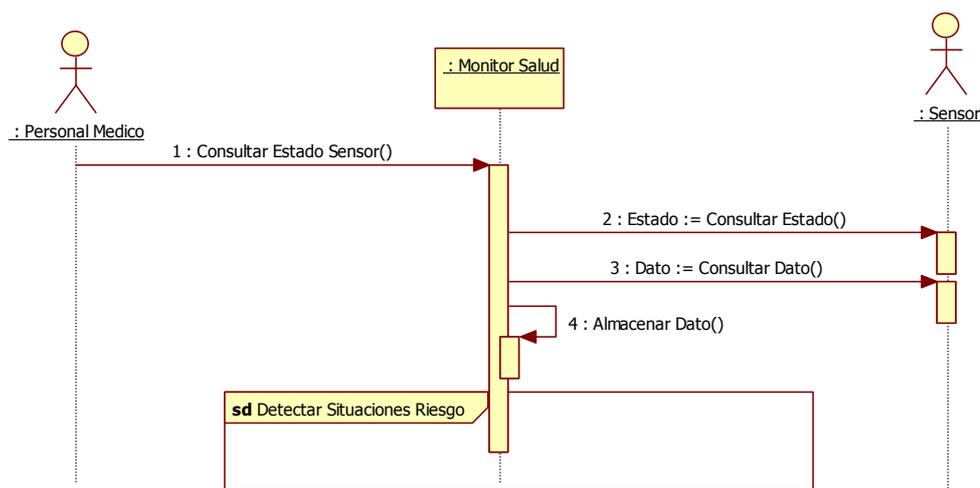


Figura 8. Diagrama de Secuencia UC-0006

UC-0007	Lectura Posición GPS	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el reloj indique que se debe realizar la lectura del GPS, y que terminará cuando dicho dato quede almacenado en caso de tener una lectura satisfactoria.	
Precondición	El sistema esta en estado iniciado y con un paciente y Gps configurado	
Secuencia normal	Paso	Acción
	1	El actor Reloj (ACT-0002) indica que el GPS tiene que comenzar a leer.
	2	El sistema consulta el estado del GPS con el objeto de comprobar su disponibilidad.
	3	El sistema consulta al GPS la información sobre la posición actual del paciente

	4	Si la lectura es correcta, el sistema almacena la posición del GPS en la base de datos local
Postcondición	El GPS se encuentra en estado de lectura Ok y la posición queda almacenada	
Excepciones	Paso	Acción
	2	Si el GPS no responde, el sistema indica que el GPS no está al alcance, a continuación este caso de uso continúa
	3	Si se produce un fallo de comunicación con el GPS, el sistema indica de lo ocurrido, a continuación este caso de uso queda sin efecto
Comentarios	Ninguno	

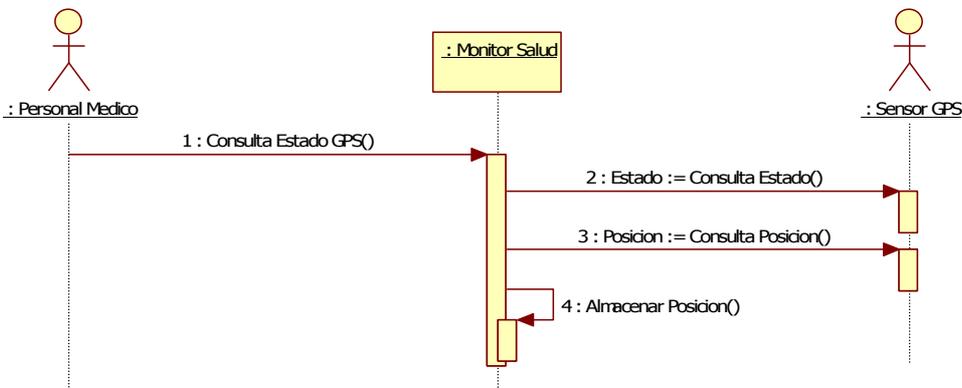


Figura 9. Diagrama de Secuencia UC-0007

UC-0008	Detectar Situaciones de Riesgo	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se ha detectado un nuevo dato en el sistema, con el objeto de que sea evaluado en las situaciones de riesgo en las que corresponda. o durante la realización de los siguientes casos de uso: [UC-0006] Lectura de Datos Sensor	
Precondición	Disponemos de un dato de un sensor para evaluar	
Secuencia normal	Paso	Acción
	1	El sistema evalúa las situaciones de riesgo, tras el caso de uso UC-0006 Lectura de Datos Sensor

	2	El sistema <i>evalúa las reglas de riesgo de las situaciones de riesgo afectadas por el dato almacenado en el UC-0006</i>
	3	El sistema <i>calcula el estado de la situación de riesgo en función del estado de las diferentes reglas de que se compone.</i>
	4	El sistema <i>calcula el estado del paciente, en función del estado de las diferentes situaciones de riesgo configuradas.</i>
	5	El sistema <i>almacena la notificación en la base de datos local</i>
Postcondición	La notificación sobre el cambio en la Situación de Riesgo queda almacenada.	
Excepciones	Paso	Acción
	2	Si <i>el dato no pertenece a ninguna regla</i> , el sistema <i>deja el dato sin evaluar</i> , a continuación este caso de uso <i>queda sin efecto</i>
	4	Si <i>no hay cambios de estado en situaciones de riesgo</i> , el sistema <i>no realiza ninguna acción</i> , a continuación este caso de uso <i>queda sin efecto</i>
Comentarios	Ninguno	

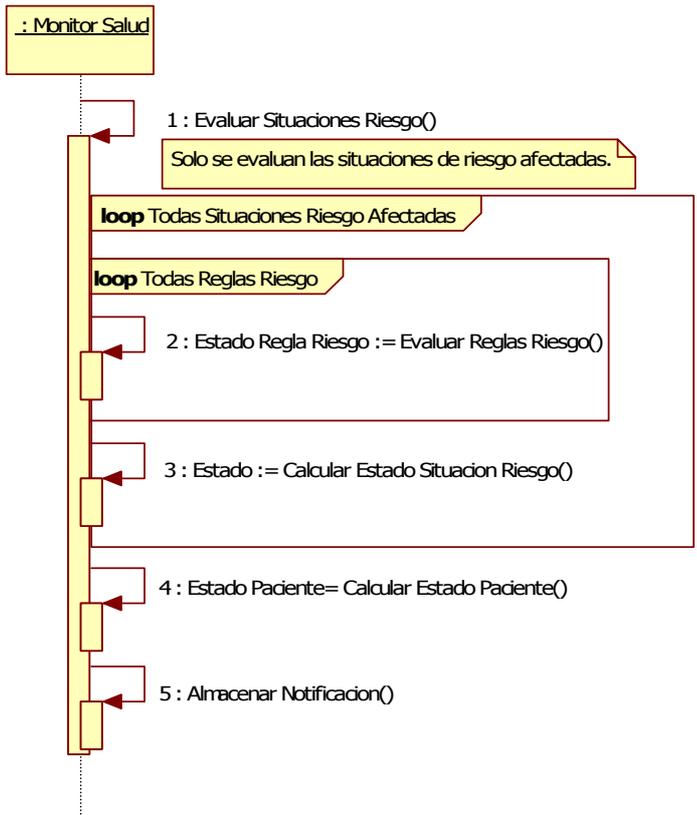


Figura 10. Diagrama de Secuencia UC-0008

UC-0009	Enviar Datos a la Central	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el reloj indique que se debe realizar el envío de los datos almacenados en local.</i>	
Precondición	El sistema esta en estado iniciado	
Secuencia normal	Paso	Acción
	1	El actor Reloj (ACT-0002) indica que se ha cumplido el intervalo de tiempo que se realice el envío de los datos
	2	El sistema <i>consulta el estado del servidor, para comprobar su disponibilidad</i>
	3	El sistema <i>envía los datos pendientes al servidor central</i>

		<i>recogiendo el estado de cada envío</i>
	4	<i>Si hay datos pendiente de enviar, el sistema borra el dato enviado</i>
Postcondición	La cola de datos de envío queda vacía	
Excepciones	Paso	Acción
	3	<i>Si hay un fallo de comunicación, el sistema establece el estado del envío como erróneo, a continuación este caso de uso continúa</i>
Comentarios	Ninguno	

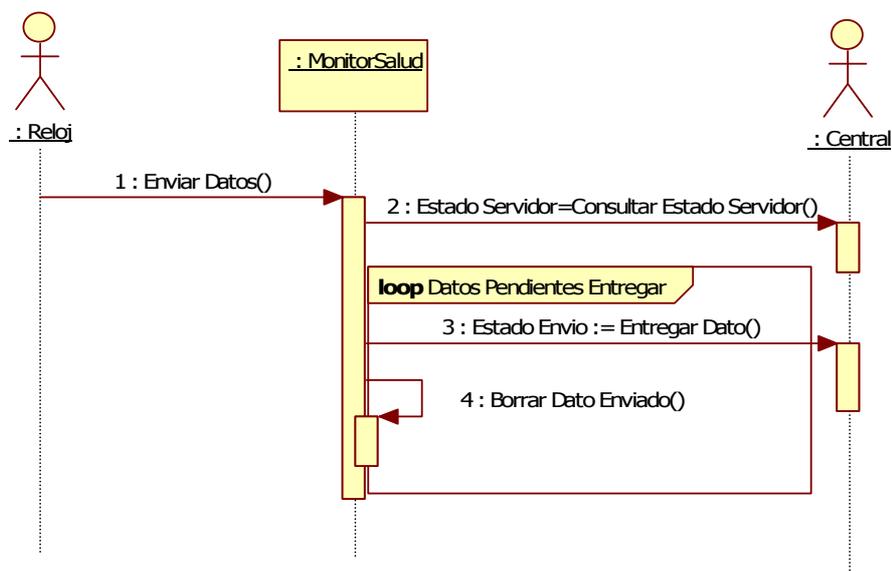


Figura 11. Diagrama de Secuencia UC-0009

UC-0010	Notificar Situación de Riesgo
Dependencias	Ninguno
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el reloj indique que se ha cumplido el intervalo de tiempo para el envío de notificaciones, o bien se haya detectado el cambio en una situación de riesgo.</i>

Precondición	el sistema esta en estado iniciado	
Secuencia normal	Paso	Acción
	1	El actor <u>Reloj (ACT-0002)</u> indica que se ha cumplido el intervalo de tiempo para que se realice el envío de notificaciones pendientes
	2	El sistema <i>comprueba la disponibilidad del servidor</i>
	3	El sistema <i>envía las notificaciones pendientes de enviar que se encuentran en la base de datos local, y obtiene el estado del envío</i>
	4	El sistema <i>borra la notificación que se acaba de enviar</i>
Postcondición	la cola de notificaciones queda vacía	
Excepciones	Paso	Acción
	3	Si se produce un fallo de comunicación, el sistema establece el estado de envío como error, a continuación este caso de uso queda sin efecto
Comentarios	Ninguno	

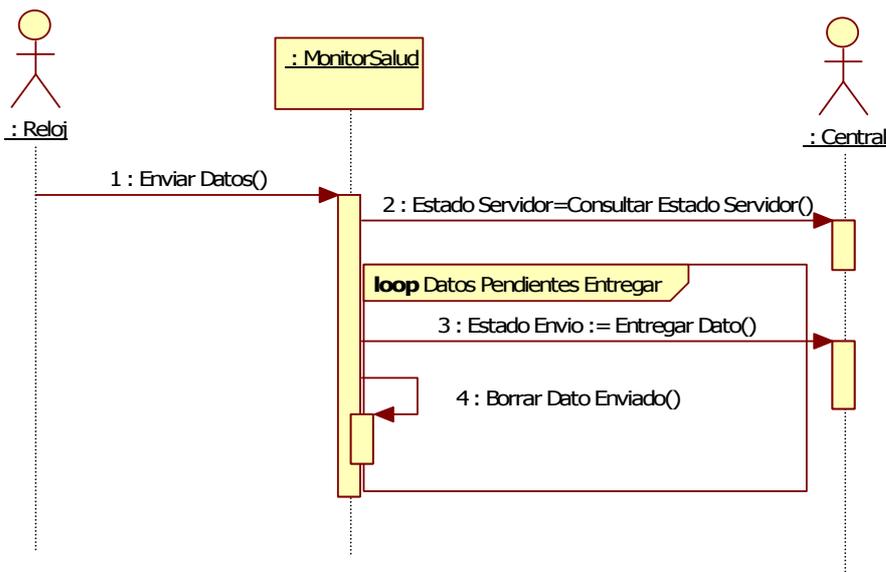


Figura 22. Diagrama de Secuencia UC-0010

2. Modelo de Análisis

2.1 Introducción

Este documento describe el diagrama de clases inicial obtenido para el proyecto Monitorización Remota de Pacientes, donde se ofrece una breve descripción de las clases detectadas en el análisis, y por tanto, que se encuentran en el Modelo de Dominio.

Este documento se limita a dar una descripción de la funcionalidad de sistema, de tal forma que se satisfagan todos y cada uno de los requisitos software desarrollados en los apartados anteriores. Se describen las clases que componen la aplicación y las relaciones entre ellas, así como una breve descripción de la responsabilidad de cada una de ellas.

Este documento se compone de 3 secciones, el diagrama de clases inicial, la realización de los casos de uso, y finalmente el modelo de dominio.

2.2 Diagrama de Clases Inicial

En función de los casos de uso descritos en la sección anterior, presentamos a continuación el diagrama de clases inicial.

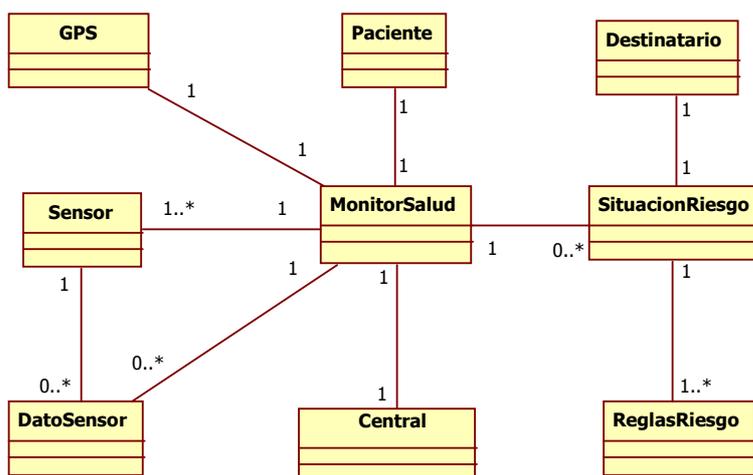


Figura 13. Diagrama de Clases Inicial

En el diagrama se han incluido las clases que cubren los diferentes conceptos que nos hemos encontrado en los casos de uso, y que van a permitir al sistema desarrollar la funcionalidad requerida en estos.

Se ha centralizado toda la funcionalidad y gestión de la aplicación sobre la clase MonitorSalud. Esta clase tendrá dos funcionalidades bien diferenciadas. Por una parte se encargara de temporizar

la lectura de los diferentes sensores y del GPS, así como de almacenar sus datos y gestionar su envío al servidor. Y por otra parte, se encargará de gestionar la evaluación de los datos obtenidos por los sensores en busca de situaciones de riesgo, y de gestionar los posibles cambios en el estado de estas, para así notificar al personal medico correspondiente de los cambios de estado del paciente.

También cabe destacar la clase `ServidorCentral`, la cual será el interfaz de comunicación para la recepción de configuración de pacientes, envío de datos y notificaciones.

Uno de los problemas principales de la aplicación, es homogeneizar los diferentes tipos de datos que nos pueden aportar los sensores. Para ello se ha incluido la clase `DatoSensor`, que nos permitirá que los datos de los sensores sean recogidos, y que posteriormente sean transformados en un tipo más homogéneo de datos.

2.2.1. Descripción de Clases de Análisis

Presentamos a continuación una breve descripción de cada una de las clases propuestas:

- **MonitorSalud**
Es la clase principal del modelo, la cual coordina al resto de las clases para la captura de los datos de los diferentes sensores y la evaluación de las situaciones de riesgo. Será la encargada de conocer en todo momento el estado del sistema y el estado del paciente, y por tanto la encargada de gestionar la notificación de cambios del paciente al personal medico.
- **Paciente**
Se trata de la clase que representa al paciente a monitorizar, y en el cual dispondremos de su estado de este en todo momento.
- **GPS**
Representa al GPS del sistema, a partir del cual podremos determinar la localización del paciente.
- **Sensor**
Representa cada uno de los sensores bluetooth del sistema, a partir de los cuales podremos recoger los diferentes valores biométricos del paciente con el fin de conocer su estado
- **DatoSensor**
Esta clase permite que los sensores puedan almacenar sus resultados en ella, y posteriormente generar una cadena xml para abstraer al dato del sensor.
- **SituacionRiesgo**
Describe un posible situación de riesgo para el paciente, y define al destinatario de la alarma, a quien hay que avisar en caso que se de esta situación de riesgo. La evaluación de la situación de riesgo, se realizará a través de la evaluación de sus reglas, de tal forma que

cuando todas las reglas de una situación de riesgo estén activadas, se activará la situación de riesgo.

- ReglaRiesgo**
 Cada una de las reglas que componen una situación de riesgo. Estas reglas tienen una regla básica de comparación, donde se permite comparar el valor obtenido de un sensor contra un valor límite configurado en la regla.
- Destinatario**
 Se trata de la persona a comunicar se ha producido un cambio en la situación de riesgo. Habitualmente será parte del personal medico.
- ServidorCentral**
 Representa el interfaz del Servidor Central contra el que se van a realizar todas las comunicaciones de la aplicación. Gracias a esta clase podremos recoger la configuración de cada paciente, enviar los datos recogidos y las notificaciones por cambio de estado de la situación de riesgo.

2.2.2. Realización de Casos de Uso

Para dotar de mayor detalle al diagrama de clases inicial, vamos a desarrollar los diagramas de secuencia de los casos de uso, con las nuevas clases arriba presentadas. No hemos incluido todos los casos de uso, pero si los mas importantes y que permiten mostrar la mayor parte de funcionalidad de la aplicación.

- UC-0006: Lectura de Datos de Sensor

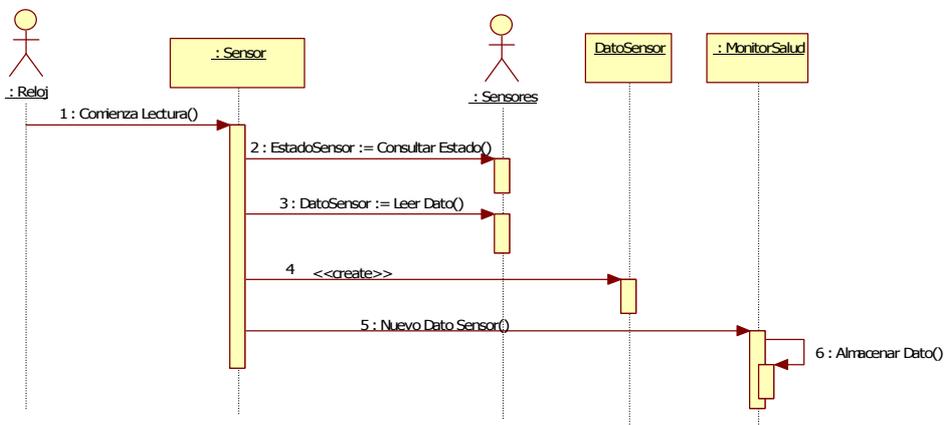


Figura 14. Diagrama de Secuencia UC-0006

- UC-0007: Lectura de Posición de GPS

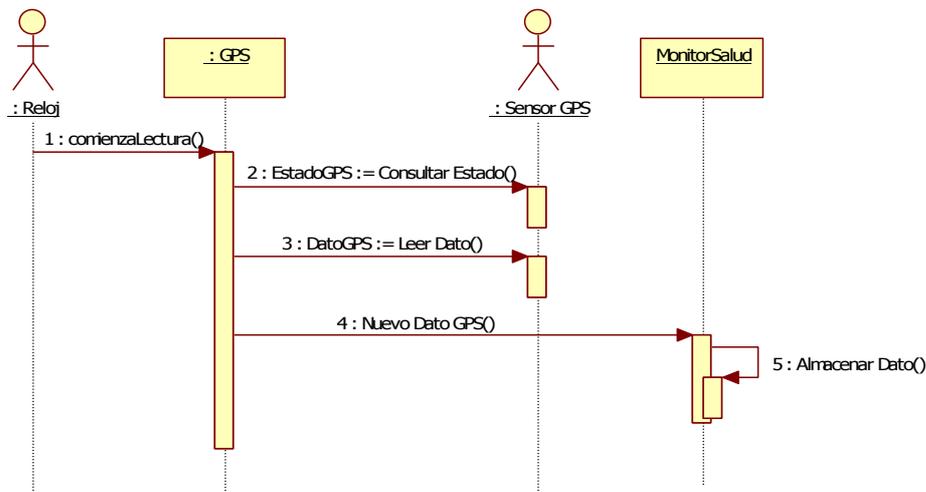


Figura 15. Diagrama de Secuencia UC-0007

- UC-0008: Detectar Situaciones de Riesgo

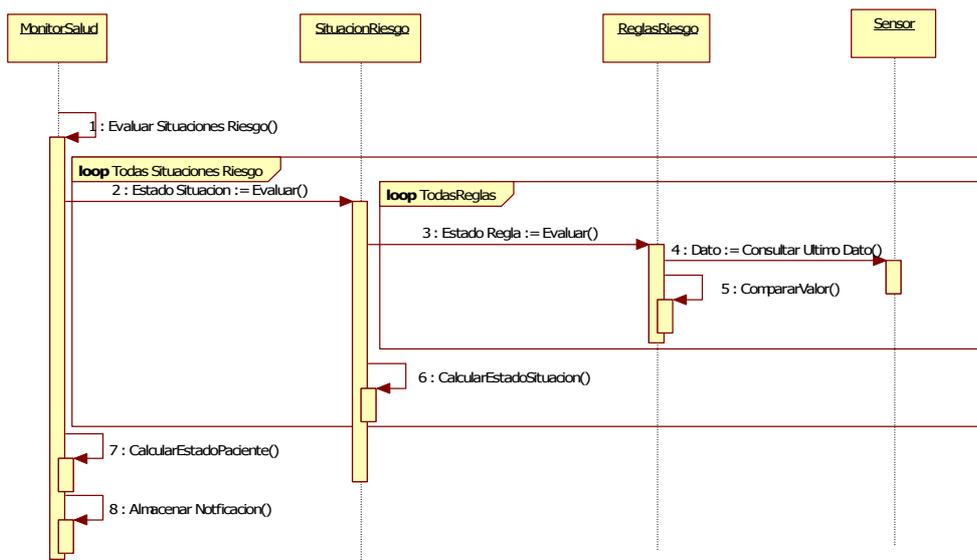


Figura 16. Diagrama de Secuencia UC-0008

- UC-0009: Enviar Datos al Servidor Central

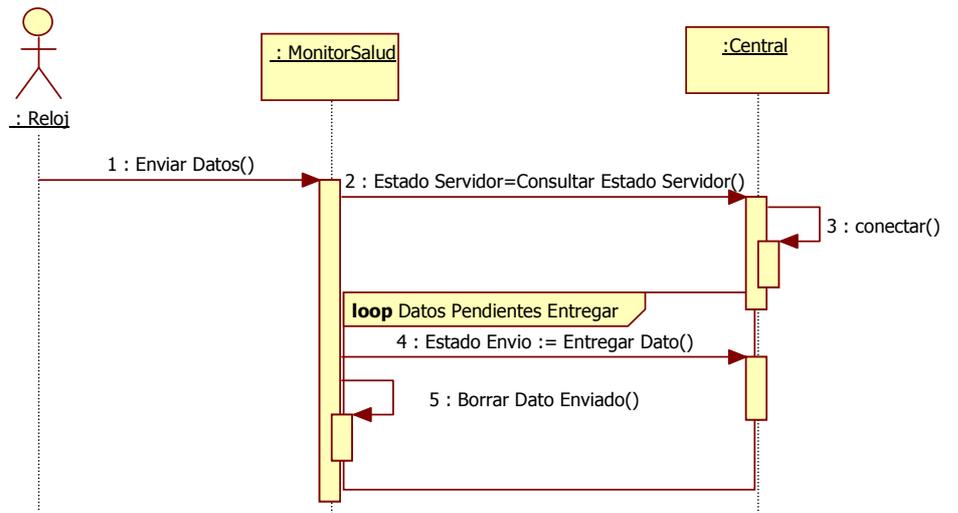


Figura 17. Diagrama de Secuencia UC-0009

- UC-0010: Notificar Situación de Riesgo

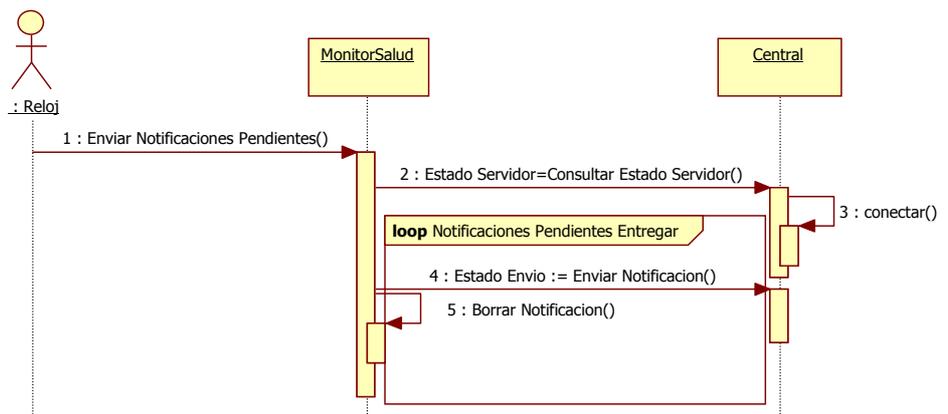


Figura 18. Diagrama de Secuencia UC-0010

2.3. Modelo de Dominio

2.3.1. Diagrama de Clases

El Modelo de Dominio es fruto de la unión del diagrama inicial de clases, los atributos necesarios para cada clase y de las operaciones obtenidas de los diagramas de secuencia.

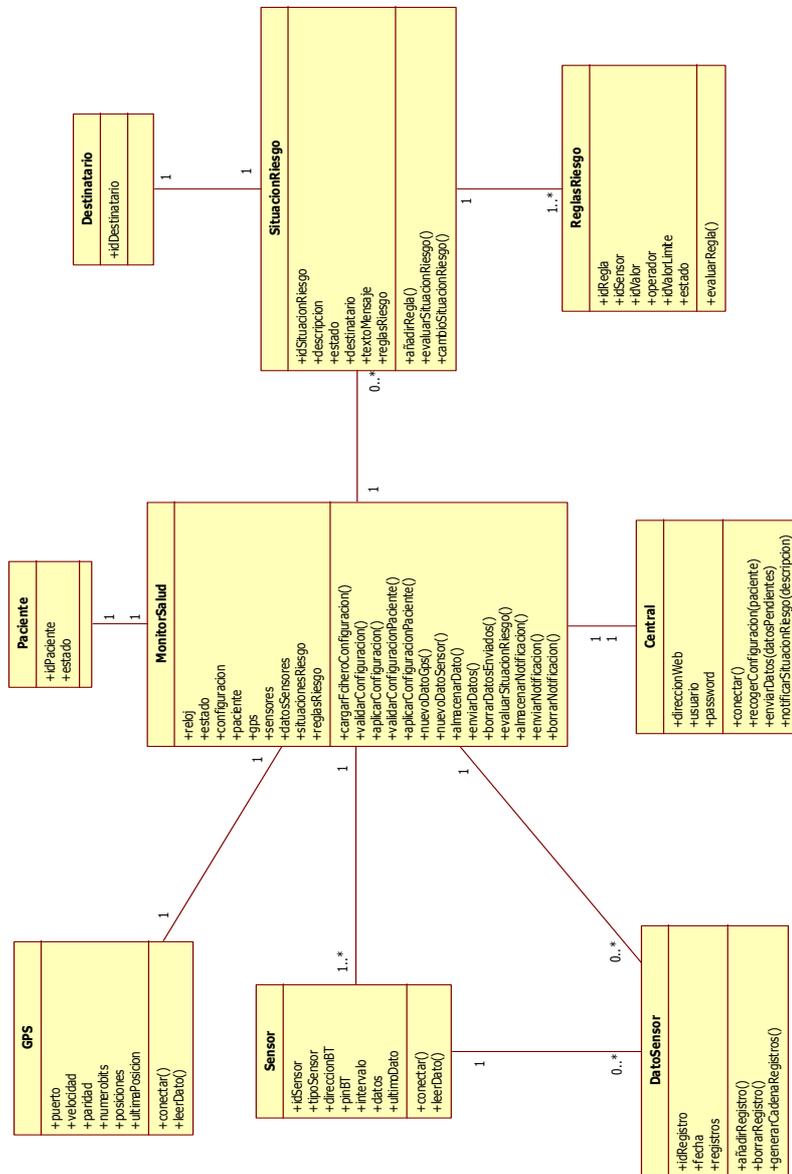


Figura 19. Diagrama de Clases Detallado

DISEÑO

1. Introducción

Exponemos a continuación el documento que muestra el Modelo de Diseño del sistema y que servirá de guía para las fases posteriores del proyecto.

Parte del modelo de análisis planteado en la sección anterior y profundiza en las estructuras y comportamientos descritos para acercarlo a la implementación del sistema.

El presente documento se divide en los siguientes puntos

- Modelo Estructural
- Diseño de la capa Interfaz
- Diseño de la capa Lógica de Negocio
 - Modelo de Interacción
 - Diagrama Detallado de Clases
- Diseño de la capa de Persistencia
- Modelo de Datos

2. Modelo Estructural

2.1. Proyecto Global Monitorización Pacientes

El proyecto global donde se incluye esta aplicación se compone de 3 paquetes, completamente independientes, que se presentan a continuación:

- MonitorizacionPDA
Se trata de la aplicación que se encarga de recoger la información de los sensores, evaluarla y enviarla al servidor central.
- MonitorizacionWS
Se trata de un conjunto de servicios web que permiten recibir los datos de los pacientes y las notificaciones de cambio de estado. También dispone de la información relativa a sensores y pacientes, y de la configuración de estos últimos.
- MonitorizacionPC
Es la aplicación que permite al personal medico disponer en tiempo real del estado de los pacientes.



Figura 1. Estructura del Proyecto Global

En el presente documento solo se detallaran las cuestiones de diseño que tengan que ver con la aplicación para el PDA, por tanto solo se documentara el paquete MonitorizacionPDA.

2.2. Diagrama de Paquetes para Monitorización PDA

A continuación se muestra el diagrama de clases general del paquete MonitorizacionPDA, que debido a su tamaño, se ha optado por dividirlo en más paquetes. Esta división se ha realizado en coherencia con el modelo de diseño por capas, y en concreto el modelo de 3 capas: interfaz, lógica de negocio y persistencia.

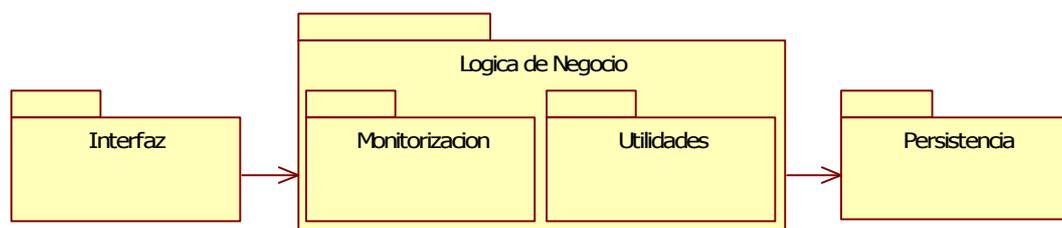


Figura 2. Diagrama de Paquetes General

La principal ventaja de este diseño es que los cambios que se producen en una capa no afectan al resto de las capas, pudiendo evolucionar cada capa independientemente del resto, manteniendo definido el interfaz entre ellas.

En el presente proyecto la distribución de las capas es bastante desigual, ya que se trata de una aplicación con una baja interacción con el usuario, y con un pequeño conjunto de datos, por lo que la gran mayoría de clases diseñadas se quedan en la capa de lógica de negocio.

Explicamos a continuación las capas diseñadas:

1. Capa de Presentación

Conocida como la interfaz de usuario, es la capa que utiliza el usuario para interactuar con la aplicación. Se compone de los formularios que dispone la aplicación para visualizar y configurar la aplicación. En nuestro caso se trata de una capa pasiva, que muestra los cambios que se van produciendo en la capa de negocio.

2. Capa de Negocio

Esta capa soporta toda la lógica de negocio de la aplicación, y es donde se coordinarán todas las lecturas de los sensores y la evaluación de las situaciones de riesgo. Esta capa será la que proporcionará a la capa de presentación el estado de todos los sensores, situaciones de riesgo, reglas de riesgo, y sobre todo el estado del paciente. También interactuará con la capa de datos para el almacenamiento de los datos recogidos por los sensores y los cambios de notificación detectados.

Esta capa se divide en dos partes. La primera es la encargada de la monitorización del paciente, mientras que la segunda dispone de una serie de clases que van a permitir la configuración del PDA y del paciente.

3. Capa de persistencia

Es la capa encargada de interactuar entre la capa de negocio y los diferentes gestores de bases de datos, proporcionando independencia entre aplicación y el gestor de base de datos utilizado.

3. Diseño de Capa de Presentación. Interfaz

La interfaz de usuario es la presentación de la aplicación para el usuario, y es la que se va a encargar de trasladar todo lo que ocurre al sistema. En toda aplicación es importante que esta capa se encuentre bien diseñada, ya que es la que realmente va a informar de lo que ocurre al usuario, tanto del buen funcionamiento del sistema, como de algún fallo de conexión con el servidor o con los sensores.

En nuestro caso, el interfaz tiene una importancia menor, debido a que la mayoría del tiempo la aplicación esta funcionando en modo pasivo, recogiendo la información del paciente y enviándola al servidor. La intervención del personal medico con la aplicación es mínima, prácticamente para la configuración del paciente.

A pesar de eso no hemos descuidado esta, y hemos procurado disponer de un interfaz amable, sencillo y ergonómico, tanto para que el personal médico pueda configurar la aplicación, como el paciente pueda conocer su estado en todo momento.

3.1. Diagrama de Clases

Presentamos a continuación las clases correspondientes a la capa de interfaz. Se tratan del conjunto de clases que nos permiten principalmente dos funciones; por una parte nos permite comprobar el estado de la aplicación; y concretamente el estado del paciente a través de sus situaciones de riesgo, el estado del GPS, y el estado de diferentes sensores. Por otro lado nos permite la configuración de la aplicación en dos sentidos, la configuración general de la aplicación y la configuración de las características asociadas al paciente.

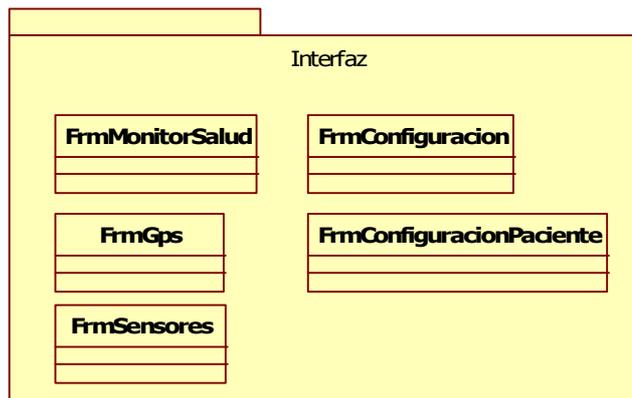


Figura 3. Diagrama de Clases de Interfaz

Podemos distinguir dos grupos de clases; por una parte las encargadas de indicarnos los estados del paciente, GPS y sensores. Y por otra parte, tenemos las clases que nos van a permitir hacer la configuración de la aplicación, y en concreto, la del paciente.

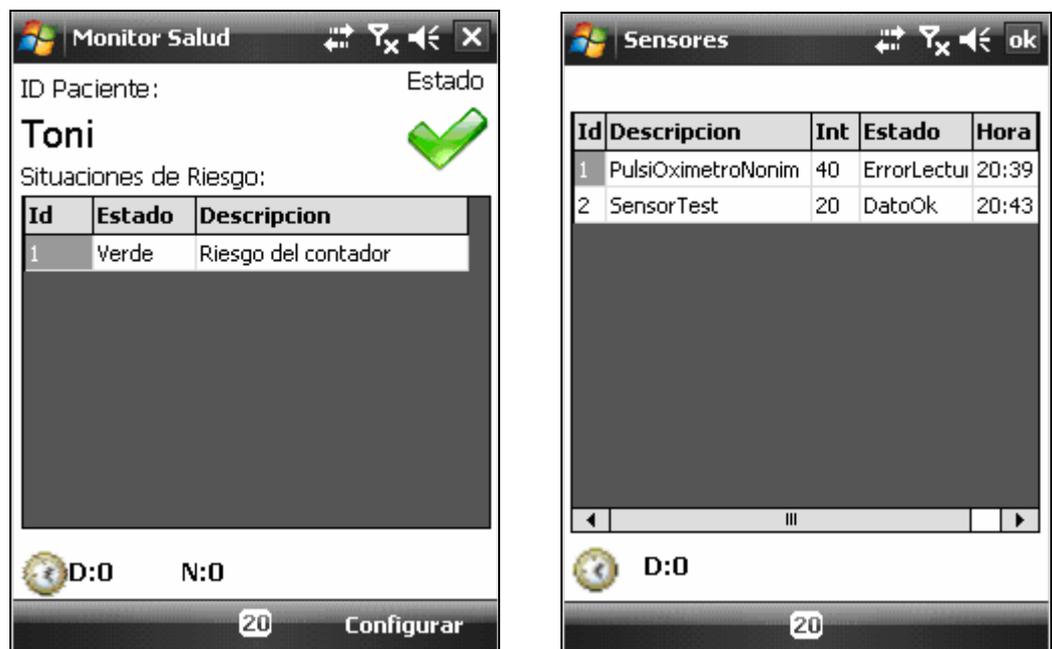


Figura 4. Formulario FrmMonitorSalud y FrmSensores

4. Diseño de la capa de Lógica de Negocio

Esta es la capa más compleja de las tres que tenemos, ya que es la encargada de tener toda la funcionalidad del proyecto, que resumimos en:

- Lectura de datos
- Evaluación
- Envío datos y notificaciones.

También es la capa que se nos va a permitir configurar la aplicación, y sobre todo configurar los sensores correspondientes al paciente.

Presentamos a continuación un resumen el Diagrama de Clases que concentra toda la funcionalidad de la lógica de negocio, y donde hemos excluido determinadas clases de soporte con el objeto de no complicar en exceso el diagrama como excepciones y enumeraciones.

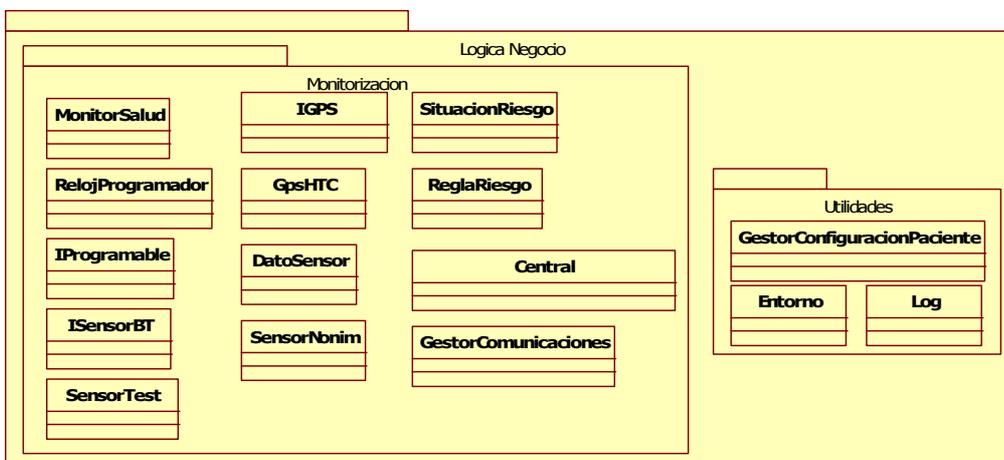


Figura 5. Diagrama de Paquetes Lógica de Negocio

Explicamos a continuación las nuevas clases que aparecen respecto al modelo de dominio que vimos en el documento de análisis.

- Abstracción de Hardware

La aplicación se ha desarrollado para que sea instalado en diferentes dispositivos PDA y que pueda utilizar diferentes dispositivos hardware. En este sentido se han desarrollado las clases abstractas como ISensorBT e IGps, las cuales son implementadas por las clases concretas GpsHTC y las clases correspondientes a cada uno de los sensores; en el ejemplo Sensor Nonim y SensorTest.

- Temporizar las lecturas de datos y envíos a la central

Para ello se ha aplicado el patrón observador a través de dos clases, Reloj Programador e IProgramable. El primero será el encargado de notificar a los observadores el paso del tiempo, mientras que la clase IProgramable será el observador que cuando alcance un número de

segundos ordenará ejecutar la función correspondiente. Las clases que requieran ser temporizadas, solo deberán implementar esta clase, y registrarse en la clase sujeto, que en este caso es RelojProgramador.

- Caché de Datos

Se contempla que la comunicación con la central no siempre será posible. Para cubrir esta falta de conexión disponemos una nueva clase llamada GestorComunicaciones, el cual se encargara de almacenar los datos y notificaciones en caso de ser necesario, para que sean enviados cuando tengamos comunicación con la central.

- Central

Esta clase se trata de una clase Proxy que nos permitirá comunicarnos con la aplicación web que se alojara en un servidor web remoto. Básicamente dispone de los métodos que vamos a utilizar en el servidor web remoto.

- Emulación de Sensores

Debido a la falta de sensores con los que probar la aplicación, se ha incluido la clase SensorTest que trata de emular un dispositivo que proporciona un entero a través de un contador incremental.

- Configuración de Paciente

Esta clase representa una utilidad para la personalización de la aplicación, con las necesidades de un nuevo paciente.

- Entorno

Se trata de una clase que dispone de la configuración general de la aplicación, y dispone de los métodos necesarios para la carga de esta configuración desde los ficheros de configuración.

- Traza de Aplicación

La aplicación de monitorización es una aplicación compleja, donde se desencadenan varios hilos de ejecución. Esta clase proporciona métodos para trazar las acciones de la aplicación y volcarlas en un fichero.

4.1. Diagramas de Secuencia

Para realizar los diagramas de secuencia entre las clases definidas, se ha tomado como base los diagramas de secuencia de la fase de análisis, y se han tenido en cuenta las nuevas clases que han surgido en el modelo de diseño.

Hemos centrado el documento en las funcionalidades más interesantes de la aplicación, y que cubren el mayor número de casos de uso.

- UC-0001: Configurar Aplicación

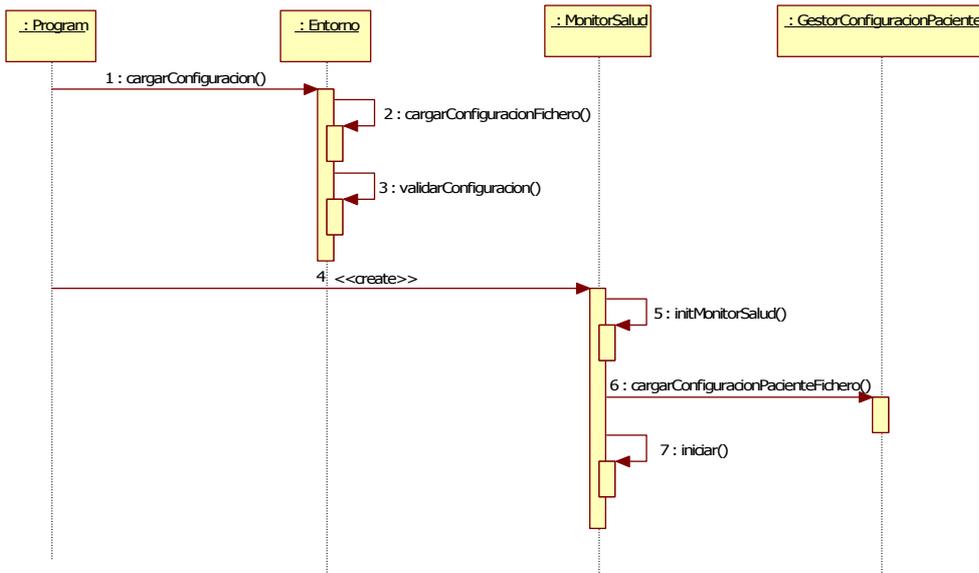


Figura 6. Diagrama Secuencia UC-0001

- UC-0002: Configuración de Paciente

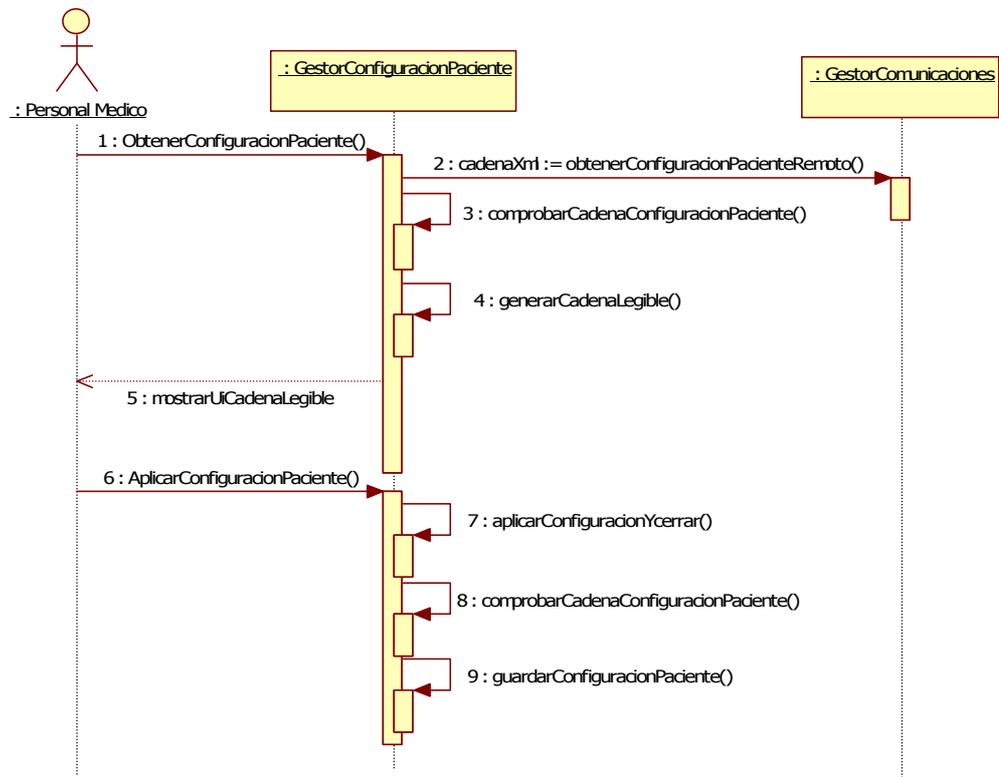


Figura 7. Diagrama Secuencia UC-0002

- UC-0006, UC-0008: Lectura, Evaluación

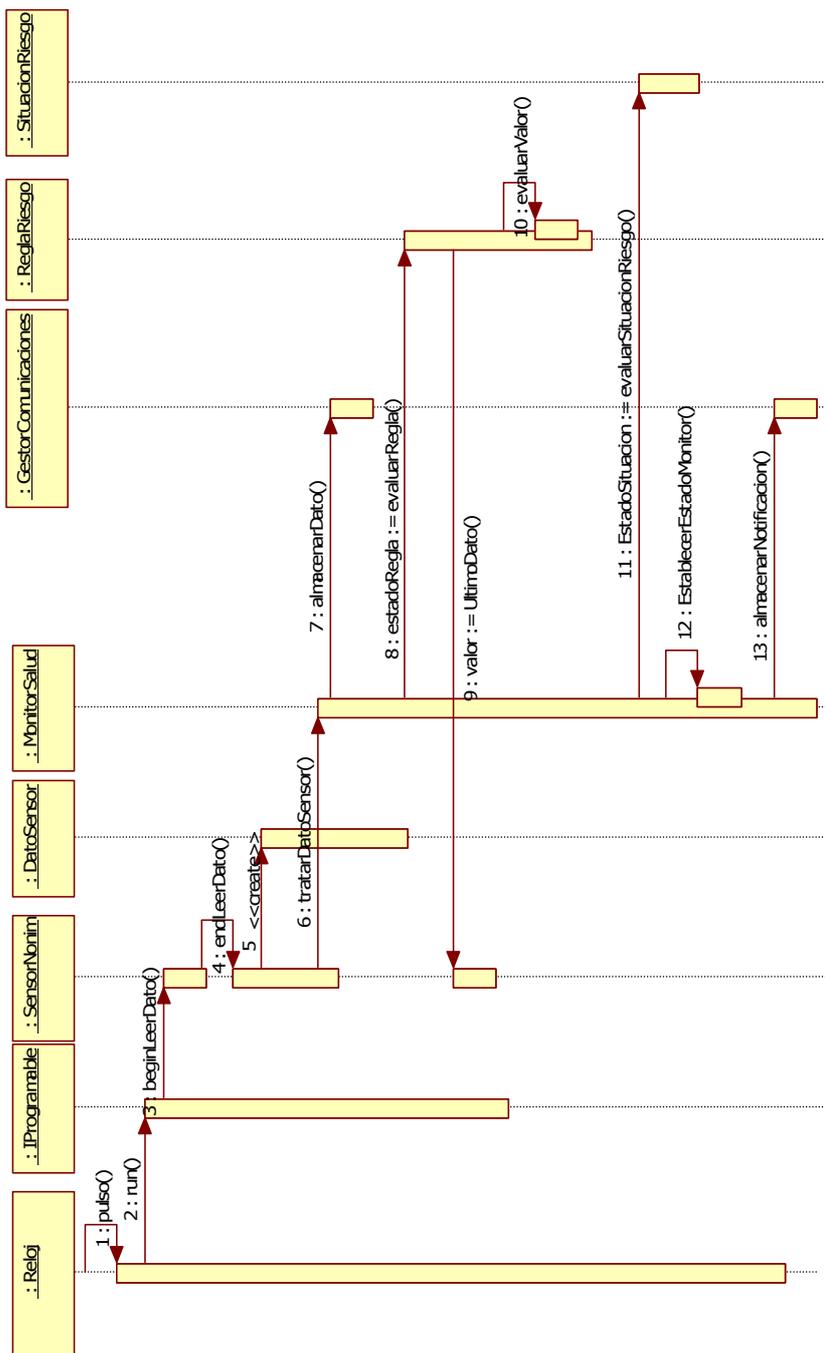


Figura 8. Diagrama Secuencia UC-0006 y UC-0008

- UC-0009: Enviar Datos

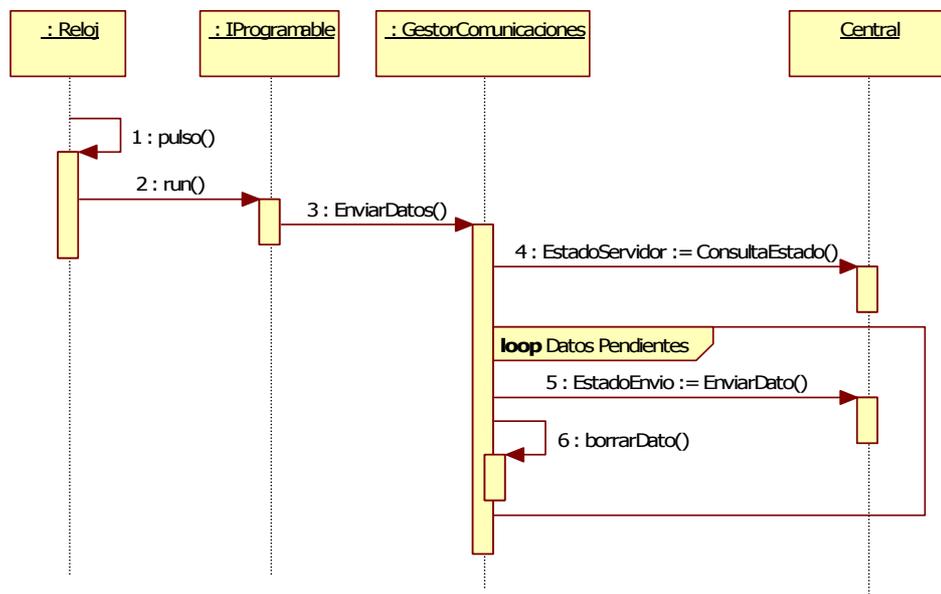


Figura 9. Diagrama Secuencia UC-0009

- UC-0010: Enviar Notificaciones

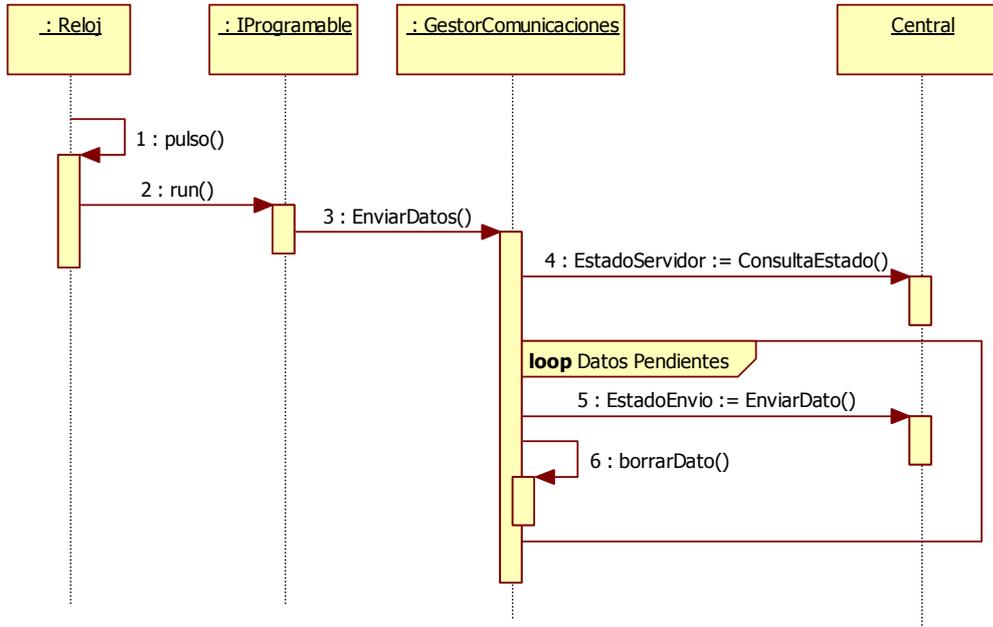


Figura 10. Diagrama Secuencia UC-0010

4.2. Diagrama de Clases Detallado

En función de los diagramas de secuencia planteados en el punto anterior, que contemplan los diagramas de secuencia del modelo de análisis, y que cumplen con los requisitos planteados en los casos de uso; exponemos a continuación el diagrama de clases más detallado de la capa de Lógica de Negocio, y en concreto del paquete Monitorización.

5. Diseño de la Capa de Persistencia

El objetivo de esta capa es aislar el desarrollo de la aplicación del sistema gestor de base de datos que se pretenda utilizar. Tengamos en cuenta que un objetivo del diseño es hacer la aplicación independiente del hardware y del soporte a utilizar, y por tanto debemos aislarla también del soporte de datos a utilizar.

Para ello nos basamos en la idea de DataSets Tipados, clase que dispone la plataforma que empleamos .Net Framework. El Dataset representa a una memoria caché de datos, con estructuras análogas a las de una base de datos, como tablas, columnas, relaciones y restricciones. Sin embargo, aunque se puede utilizar un objeto DataSet como una base de datos (y su comportamiento es muy similar), es importante recordar que los objetos DataSet no interactúan directamente con bases de datos ni con otros datos de origen. Esto permite al programador trabajar con un modelo de programación que siempre es coherente, independientemente de dónde resida el origen de datos.

Presentamos a continuación el diagrama de clases asociado a la capa de persistencia, y en el siguiente apartado abordaremos el diccionario de datos, donde podremos comprobar la estructura de las tablas utilizadas, así como los ficheros de configuración tanto del PDA, como del paciente.

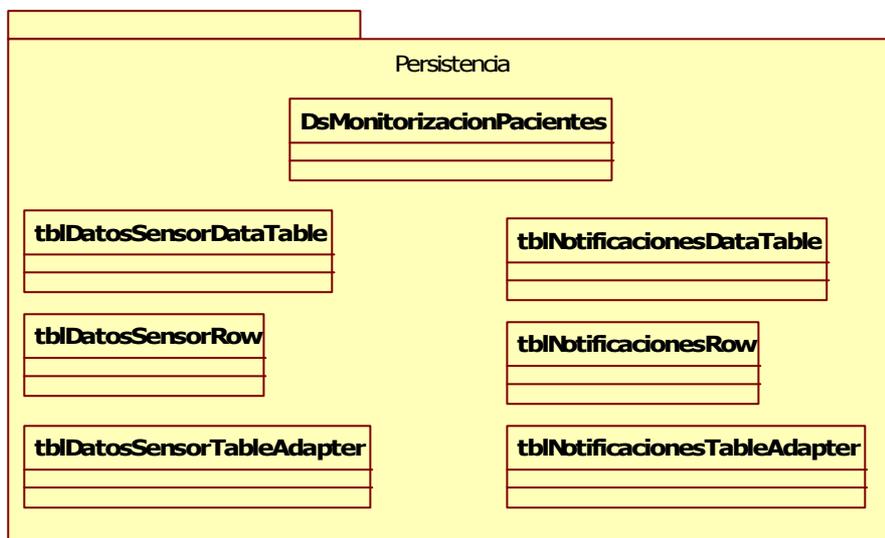


Figura 12. Diagrama de Clases de Persistencia

6. Modelo de Datos

En este documento vamos a contemplar 2 tipos de estructuras de datos, con el objetivo de cubrir dos aspectos bien diferenciados en el proyecto.

- Persistencia de Datos y Notificaciones
- Ficheros de Configuración Paciente y Pda

Ambas estructuras de datos tienen objetivos bien diferentes y por tanto sus características difieren mucho entre sí. Por tanto hemos optado por dos tipos diferentes de soluciones para cada uno.

6.1. Datos de Sensores y Notificaciones de Situaciones de Riesgo

En este punto tenemos dos tipos de datos con objetivos bien diferenciados, pero que se encuentran agrupados debido a que su tratamiento debe ser similar.

Los Datos de Sensores se trata de los datos que se recogen de la lectura de los diferentes sensores, y que posteriormente se deben enviar al servidor para ser interpretados por el personal médico. Se trata de datos importantes que transportan los valores biométricos de los pacientes. Una de las características principales del dato es su abstracción, ya que tengamos en cuenta que este tipo de dato sirve para todos los tipos de sensores del sistema, y por tanto para toda la variedad de información que estos deben proporcionar.

Por otra parte, tenemos las Notificaciones de Situaciones de Riesgo, que son los cambios de estado que sufre un paciente a raíz de la evaluación de reglas de riesgo con los datos de sensor recogidos. Estas notificaciones son muy importantes, ya que en ellas se envían los cambios de estado del paciente.

Debido a la importancia de estos datos, la persistencia de éstos es una parte fundamental del proyecto, ya que al tratarse de una monitorización remota y móvil de un paciente, se puede dar el caso de que el dispositivo PDA, no siempre tenga la cobertura adecuada para transmitir los datos, sobre todo, las notificaciones. Por tanto es imprescindible que los datos y notificaciones queden bien almacenados en una base de datos, para su posterior envío.

Existen multitud de métodos para cumplir con esta persistencia, pero hemos escogido la utilización de la base de datos SQL Server Mobile. Esta base de datos está especialmente ideada para dispositivos móviles, optimizando el uso de la batería, y manteniendo gran parte de la funcionalidad habitual de una base de datos. Además la integración con la tecnología Microsoft .Net elegida, de la misma forma que la integración con los Dataset.

Diccionario de Datos

- Tabla tblDatosSensor

Nombre	Tipo de Datos	Longitud	Permitir Valores Nulos	Único	Clave Principal	Notas
idDatoSensor	bigint	8	No	No	Si	Autoincremental
fecha	datetime	8	No	No	No	
paciente	nvarchar	50	No	No	No	
sensor	nvarchar	50	No	No	No	
cadenaXml	nvarchar	1000	No	No	No	
estadoDatoSensor	nvarchar	10	No	No	No	Pendiente o Enviado

- Tabla tblNotificaciones

Nombre	Tipo de Datos	Longitud	Permitir Valores Nulos	único	Clave Principal	Notas
idNotificacion	bigint	8	No	No	Si	Autoincremental
fecha	datetime	8	No	No	No	
paciente	nvarchar	50	No	No	No	
destinatario	nvarchar	50	No	No	No	
descripcion	nvarchar	100	No	No	No	
textoMensaje	nvarchar	200	No	No	No	
cadenaXml	nvarchar	4000	No	No	No	
estadoSituacion	nvarchar	10	No	No	No	Verde, Amarillo o Rojo
estadoPaciente	nvarchar	10	No	No	No	Verde, Amarillo o Rojo
estadoNotificacion	nvarchar	10	No	No	No	Pendiente o Enviado

6.2. Ficheros Configuración Pda y Paciente

Se trata de los datos que van a almacenar la información relativa a la configuración de la aplicación, y a la configuración de los pacientes. Los datos de configuración van a estar alojados en ficheros simples, debido ya que el uso que se da de ellas no suele ser excesivo, y habitualmente de sólo lectura. De todas formas, también tenemos que tener en cuenta que la configuración del paciente, va a ser recogida de forma remota y va a ser entregada a través de un Web Service.

Finalmente nos hemos decantado por ficheros con estructura XML, por ser una lenguaje estándar, que permite introducir control de errores y que no tiene ningún problema para ser enviado de forma remota. Además la integración de XML con la plataforma Microsoft .Net es total, y Visual Studio proporciona herramientas para la manipulación de este tipo de ficheros.

La definición de estos ficheros la hemos realizado a través de sus esquemas XML, con ficheros XSD. El contenido de estos ficheros se muestra a continuación.

- Fichero Configuración PDA

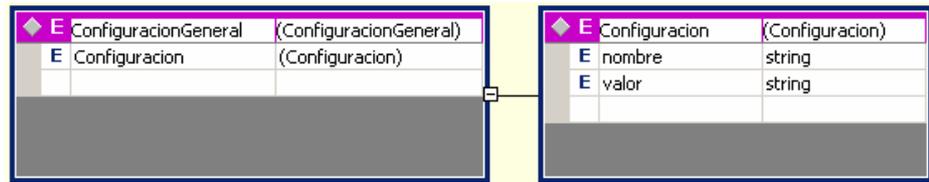


Figura 13. Xml Configuración PDA

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="xsdConfiguracionPda"
targetNamespace="http://tempuri.org/XMLSchemal.xsd" elementFormDefault="qualified"
xmlns="http://tempuri.org/XMLSchemal.xsd"
xmlns:mstns="http://tempuri.org/XMLSchemal.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ConfiguracionGeneral" nillable="false">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Configuracion">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre" type="xs:string" nillable="false" />
              <xs:element name="valor" type="xs:string" nillable="false" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

- Fichero Configuración Paciente

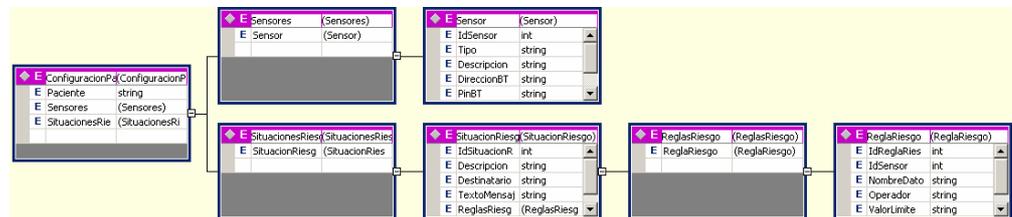


Figura 14. XML Configuración Paciente

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="ConfiguracionPaciente"
targetNamespace="http://tempuri.org/ConfiguracionPaciente.xsd"
elementFormDefault="qualified" xmlns="http://tempuri.org/ConfiguracionPaciente.xsd"
xmlns:mstns="http://tempuri.org/ConfiguracionPaciente.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ConfiguracionPaciente">
    <xs:complexType>
```


6.3 Estados de una Situación de Riesgo

Una situación de riesgo se puede encontrar en un estado concreto en función de si sus reglas de riesgo están activadas o no. Por tanto, una situación de riesgo se puede encontrar en los siguientes estados:

- Verde
Se trata de la situación inicial. Una situación de riesgo se encuentra en este estado, cuando ninguna de las reglas asociadas se encuentra activada.
- Amarillo
Una situación de riesgo cambia a este estado, cuando se activa como mínimo una de sus reglas de riesgo; pero al menos una de sus reglas se encuentra desactivada.
- Rojo
Cuando todas las reglas de riesgo asociadas se encuentran activadas.

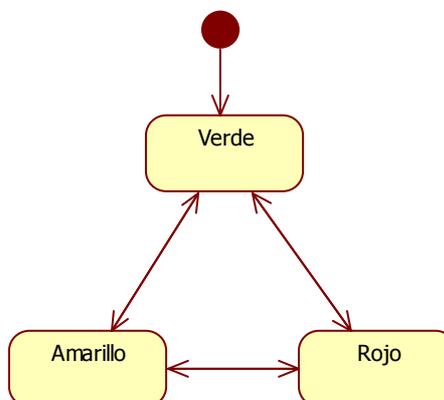


Figura 15. Diagrama de Estados de una Situación de Riesgo

6.4 Estados de un Paciente

De la misma forma que una situación de riesgo, un paciente se puede encontrar en un estado concreto en función del estado de las situaciones de riesgo configuradas. Por tanto, un paciente se puede encontrar en los siguientes estados:

- Verde
Se trata de la situación inicial. Un paciente se encuentra en este estado, si todas las situaciones de riesgo están en estado verde.
- Amarillo
Un paciente cambia a este estado, si al menos una de las situaciones de riesgo está en estado amarillo.
- Rojo
Un paciente cambia a este estado, si al menos una de las situaciones de riesgo está en estado rojo.

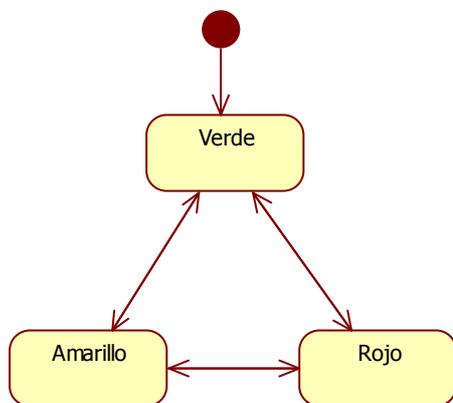


Figura 15. Diagrama de Estados de un Paciente

IMPLEMENTACION

1. Introducción

Este documento muestra el Modelo de Implementación del sistema. Muestra cómo se traduce el modelo de diseño en los distintos componentes ejecutables de la aplicación a desarrollar.

Durante este capítulo detallaremos del Modelo de Implementación del proyecto Monitorización PDA, aunque a continuación presentamos el despliegue de los componentes desarrollados a través de los diferentes elementos hardware.



Figura 1. Diagrama de Despliegue

2. Requisitos Software

La implantación de la aplicación en el PDA requiere del siguiente software:

- Compact Net Framework v2.0

Es el framework de la arquitectura Microsoft .Net que dispone de las clases básicas para el desarrollo de la aplicación.

- Librerías 32feet.Net v2.2

Se tratan de librerías que permiten el manejo del hardware necesario para realizar conexiones Bluetooth a través de código.

- ScreenToggle v1.3

Se trata de una utilidad que permite apagar la pantalla del PDA sin que éste pase a estado stand by, con el objetivo de ahorrar energía.

3. Optimización Uso de Batería

La aplicación de monitorización desarrollada en este PFC, es una aplicación que por sus características hace un uso intensivo de la batería. El PDA es un dispositivo de uso general, que puede tener otras tantas aplicaciones funcionando. Por tanto es necesario configurar el sistema operativo del PDA para optimizar el uso de la batería, y se recomienda no utilizar el PDA para otras aplicaciones que no sean necesarias para la monitorización del paciente.

Debido a la necesidad de que el PDA esté encendido para el correcto funcionamiento de la aplicación, hemos incluido una funcionalidad para el apagado de la pantalla, la cual nos permite

ahorrar batería en esta situación. Así de esta forma, la aplicación ScreenToggle nos permite el apagado de la pantalla de forma manual, únicamente teniendo que pulsar un botón. Para restablecer el PDA, sólo será necesario utilizar el botón de encendido.

4. Ficheros de Configuración

Los ficheros de configuración de la aplicación y que disponen de las opciones de configuración como cadenas de conexión y modos de funcionamiento de la aplicación, se encuentra almacenadas en la propia carpeta de instalación de la aplicación.

Mostramos a continuación varios ejemplos de estos ficheros

4.1. Fichero de Configuración General

Este fichero se carga en el inicio de la aplicación, y nos permite la configuración general del dispositivo, que en este caso esta orientado a la configuración del GPS y del Servidor Central

```
<?xml versión="1.0" encoding="utf-16"?>
<ConfiguracionGeneral>
  <Configuracion>
    <Nombre>MonitorInicioAutomatico</Nombre>
    <Valor>>true</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsActivado</Nombre>
    <Valor>>true</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsPuerto</Nombre>
    <Valor>COM4</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsVelocidad</Nombre>
    <Valor>4800</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsParidad</Nombre>
    <Valor>None</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsNumBits</Nombre>
    <Valor>8</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsBitsParada</Nombre>
    <Valor>One</Valor>
  </Configuracion>
  <Configuracion>
    <Nombre>GpsIntervalo</Nombre>
    <Valor>120</Valor>
  </Configuracion>
</ConfiguracionGeneral>
```

Aplicación para la monitorización remota de pacientes

```
<Configuracion>
  <Nombre>ComunicacionesWebPasarela</Nombre>
<Valor>http://localhost/MonitorizacionPacientes_/Monitorizacion.aspx</Valor>
</Configuracion>
<Configuracion>
  <Nombre>ComunicacionesIntervaloEnvioPasarela</Nombre>
  <Valor>300</Valor>
</Configuracion>
</ConfiguracionGeneral>
```

4.2. Ficheros de Configuración de Paciente

Los ficheros de configuración de paciente tienen como objetivo la configuración del paciente en el PDA, de tal forma que la aplicación quede configurada con el paciente, los sensores necesarios para su monitorización, y las situaciones de riesgo y reglas de riesgo configuradas para monitorizar su dolencia.

```
<?xml version="1.0" encoding="utf-16"?>
<ConfiguracionPaciente>
  <Paciente>Maria</Paciente>
  <Sensores>
    <Sensor>
      <IdSensor>1</IdSensor>
      <Tipo>SensorNonim</Tipo>
      <Descripcion>PulsiOximetroNonim</Descripcion>
      <DireccionBT>00A09617D76D</DireccionBT>
      <PinBT>503208</PinBT>
      <Intervalo>40</Intervalo>
    </Sensor>
    <Sensor>
      <IdSensor>2</IdSensor>
      <Tipo>SensorTest</Tipo>
      <Descripcion>SensorTest</Descripcion>
      <DireccionBT />
      <PinBT />
      <Intervalo>20</Intervalo>
    </Sensor>
  </Sensores>
  <SituacionesRiesgo>
    <SituacionRiesgo>
      <IdSituacionRiesgo>1</IdSituacionRiesgo>
      <Descripcion>Riesgo del contador</Descripcion>
      <Destinatario>Medico</Destinatario>
      <TextoMensaje>Valor incorrecto del contador</TextoMensaje>
      <ReglasRiesgo>
        <ReglaRiesgo>
          <IdReglaRiesgo>1</IdReglaRiesgo>
          <IdSensor>2</IdSensor>
          <NombreDato>ContadorTest</NombreDato>
          <Operador>Mayor</Operador>
          <ValorLimite>1</ValorLimite>
```

```
</ReglaRiesgo>
<ReglaRiesgo>
  <IdReglaRiesgo>2</IdReglaRiesgo>
  <IdSensor>2</IdSensor>
  <NombreDato>ContadorTest</NombreDato>
  <Operador>Mayor</Operador>
  <ValorLimite>2</ValorLimite>
</ReglaRiesgo>
<ReglaRiesgo>
  <IdReglaRiesgo>3</IdReglaRiesgo>
  <IdSensor>2</IdSensor>
  <NombreDato>ContadorTest</NombreDato>
  <Operador>Mayor</Operador>
  <ValorLimite>3</ValorLimite>
</ReglaRiesgo>
</ReglasRiesgo>
</SituacionRiesgo>
</SituacionesRiesgo>
</ConfiguracionPaciente>
```


PRUEBAS

1. Introducción

Este documento recoge los casos de prueba diseñados para detectar posibles errores en la aplicación desarrollada en el proyecto, por tanto quedan excluidas de las pruebas las aplicaciones que no intervienen en la PDA,

Todas las pruebas que presentamos en el documento son de caja negra, diseñadas para encontrar errores funcionales en el proyecto, y por tanto errores sobre los casos de uso planteados en el análisis. Se realizaran pruebas sobre cada caso de uso, comprobando que la aplicación cumple con los requisitos especificados y que funciona de la manera esperada.

Las pruebas de caja blanca no se han incluido debido a su complejidad. Estas fueron realizadas en la etapa de programación de cada clase del proyecto, aprovechando la funcionalidad de Visual Studio 2005 para comprobar el buen funcionamiento de cada método del proyecto.

2. Casos de Prueba

Organizaremos los casos de prueba en función de cada uno de los casos de uso. Utilizaremos la siguiente plantilla para definir la prueba

Prueba	Tipo	Resultado	Valoración
Descripción detallada de la prueba a realizar	Funcional (F), Interfaz(I)	Resultado observado de la prueba	Correcto (OK), Incorrecto(NOK)

- UC-0001: Configurar Aplicación

Este caso de uso representa la personalización de la aplicación con configuración propia de la PDA, independientemente de la configuración del paciente; y que sustituirá a la configuración por defecto de la aplicación.

Prueba	Tipo	Resultado	Valoración
Iniciar Aplicación configuración por defecto	F	La aplicación se inicia correctamente, con el reloj funcionando automáticamente	OK
Iniciar Aplicación configuración personalizada con el reloj apagado	F	La aplicación se inicia correctamente, y en la prueba actual, el reloj se encuentra detenido.	OK
Forzar error en opción configuración	F	La aplicación se inicia correctamente, pero en el Log se indica que existe una opción errónea e indica cual es	OK

- UC-0002: Configurar Paciente

Este caso de uso representa la personalización de la aplicación para cada uno de los pacientes de la aplicación, proporcionando configuración sobre sensores e intervalos de lectura; además de situaciones de riesgo con sus reglas correspondientes.

Prueba	Tipo	Resultado	Valoración
Comprobar configuración actual	F,I	La aplicación muestra en una cuadro de texto el identificador del paciente configurado actualmente, junto con los sensores configurados con su información, y las situaciones de riesgo configuradas con sus reglas de riesgo	OK
Solicitar configuración al servidor	F,I	La aplicación muestra en un cuadro de texto el identificador del paciente solicitado, junto con los sensores configurados con su información, y las situaciones de riesgo configuradas con sus reglas de riesgo.	OK
Solicitar configuración al servidor con nombre erróneo	F,I	La aplicación nos indica que el servidor no dispone de la configuración para el paciente solicitado.	OK
Apagar el servidor	F,I	La aplicación nos indica que no puede acceder al servidor	OK
Configuración paciente errónea	F,I	La aplicación indica que la configuración no es correcta, y la descarta para su uso.	OK
Aplicar configuración	F,I	La aplicación indica que la configuración es correcta, posteriormente que se ha aplicado correctamente, y finalmente indica que la aplicación se tiene que reiniciar. Al concluir con el reinicio comprobamos que el nuevo paciente esta correctamente configurado en la aplicación.	OK
Aplicar diferentes configuraciones de sensores	F,I	Al reiniciar la aplicación comprobamos que las configuraciones de los sensores están correctamente aplicadas.	OK
Aplicar diferentes configuraciones de situaciones de riesgo	F,I	Al reiniciar la aplicación comprobamos que las configuraciones de las situaciones de riesgo están correctamente aplicadas	OK
Aplicar diferentes configuraciones de reglas de riesgo	F,I	Al reiniciar la configuración comprobamos que las configuraciones de las reglas están correctamente aplicadas	OK

- UC-0003: Consultar Estado Paciente

Este caso de uso representa el interés por parte del personal medico de conocer el estado del paciente a través del estado de las situaciones de riesgo que tienen configuradas.

Prueba	Tipo	Resultado	Valoración
Forzar cambio estado del paciente a Amarillo, a través del sensor SensorTest	F,I	Esperamos a que se cumpla la condición de la regla de riesgo. En el formulario de estado del paciente, el icono de estado pasa a Amarillo, y podemos observar en los	OK

Aplicación para la monitorización remota de pacientes

		contadores de Datos y Notificaciones, que ambos se han incrementado en 1; en el primero por el dato generado, y en el segundo por la notificación generada por el cambio de situación	
Forzar cambio estado del paciente a Rojo, a través del sensor SensorTest	F,I	En el formulario de estado del paciente, el icono de estado pasa a Amarillo, y podemos observar en los contadores de Datos y Notificaciones, que ambos se han incrementado en 1; en el primero por el dato generado, y en el segundo por la notificación generada por el cambio de situación	OK
Forzar cambio estado del paciente a Verde, a través del sensor SensorTest	F,I	En el formulario de estado del paciente, el icono de estado pasa a Verde, y podemos observar en los contadores de Datos y Notificaciones, que ambos se han incrementado en 1; en el primero por el dato generado, y en el segundo por la notificación generada por el cambio de situación	OK

- UC-0004: Consultar Estado Sensor

Este caso de uso representa el interés en conocer el buen funcionamiento de los sensores asociados al paciente, conociendo su estado en todo momento.

Prueba	Tipo	Resultado	Valoración
Ir a pantalla de sensores	I	La aplicación muestra una pantalla con todos los sensores configurados, indicando el identificador de sensor, el nombre, el estado de la última lectura y la hora de la última lectura.	OK
Forzar cambio estado sensor	F,I	Esperamos a que se cumpla el intervalo de lectura de uno de los sensores. La aplicación muestra como el sensor cambia a un estado de Leyendo, y cuando termina la lectura el estado pasa a un estado de DatoOK.	OK
Cortar la comunicación en mitad lectura	F,I	Esperamos a que se cumpla el intervalo de lectura de uno de los sensores. La aplicación muestra como el sensor cambia a un estado de Leyendo; pero en este caso el estado se mantiene durante un intervalo mayor de tiempo, y al final cambia a un estado de ErrorLectura	OK

- UC-0005: Consultar Estado GPS

Este caso de uso representa el interés en conocer el buen estado de funcionamiento del GPS, conociendo su estado en todo momento y conociendo la última posición recogida.

Prueba	Tipo	Resultado	Valoración
Ir a pantalla de gps	I	La aplicación muestra la pantalla con la ultima posición capturada por el GPS, e indica el estado actual de este; en el momento de la prueba el sensor se encuentra en estado Iniciado	OK
Cambio estado gps	F,I	Indicamos al GPS que comience la lectura. La aplicación muestra como el estado del GPS cambia a un estado de Iniciado. Pasados aproximadamente dos minutos, el estado del GPS pasa a Detenido y en la pantalla de datos aparece el mensaje TimeOut Alcanzado.	OK
Cambio de posición de paciente	F,I	Para esta prueba nos salimos al exterior, y esperamos a la búsqueda de dos posiciones. En la primera nos muestra nuestra posición actual, tras la cual decidimos movernos. En la segunda, el GPS muestra otra posición diferente a la primera. Cuando finalizan ambas lecturas, el estado del GPS pasa a Detenido.	OK
Cortar la comunicación en mitad lectura	F,I	Nos es imposible realizar esta prueba, ya que el GPS se encuentra integrado en el dispositivo PDA.	No Realizada

- UC-0006: Lectura de Datos de Sensor

Este caso de uso representa la lectura por parte del sistema de los datos de los diferentes sensores y su almacenamiento local.

Prueba	Tipo	Resultado	Valoración
Cambio estado de sensor	F,I	Usamos para esta prueba el sensor PulsioximetroNonim. Esperamos a que se cumpla el intervalo de lectura del sensor y que comience la lectura. Comprobamos que el sensor cambia al estado Leyendo.	OK
Lectura de sensor	F,I	Usamos para esta prueba el sensor SensorTest. Esperamos a que se cumpla el intervalo de lectura del sensor y a que el dato sea leído. Comprobamos como el sensor cambia de estado a Leyendo, y como después cambia a estado DatoOK. También podemos comprobar como el contador de	OK

		datos se incrementa en uno.	
Lectura de sensor con características Bluetooth	F,I	Usamos para esta prueba el sensor PulsioximetroNonim. Esperamos a que se cumpla el intervalo de lectura del sensor y que comience la lectura. Comprobamos que el sensor cambia al estado Leyendo, y pasados unos segundos, como cambia al estado DatoOK. También podemos comprobar como el contador de datos se incrementa en uno.	OK
Cortar la comunicación en mitad lectura en sensores Bluetooth.	F,I	Usamos para esta prueba el sensor PulsioximetroNonim. Esperamos a que se cumpla el intervalo de lectura del sensor y que comience la lectura. Comprobamos que el sensor cambia al estado Leyendo. En ese momento apagamos el sensor. Pasados unos segundos, la aplicación nos muestra como el sensor ha pasado a estado DatoNoOK. Comprobamos también que el contador de Datos no se ha incrementado.	No Realizada

- UC-0007: Lectura de Posición de GPS

Este caso de uso representa la lectura por parte del sistema de la posición del sensor GPS y su almacenamiento local.

Prueba	Tipo	Resultado	Valoración
Lectura Programada GPS en exterior	F,I	Esperamos a que se cumpla el intervalo de lectura del GPS. Comprobamos que el icono de lectura del GPS se ha activado. Pasados aproximadamente unos 30 segundos el icono desaparece. Comprobamos que el contador de datos se ha incrementado en uno.	
Lectura Programada GPS en interior	F,I	Esperamos a que se cumpla el intervalo de lectura del GPS. Comprobamos que el icono de lectura del GPS se ha activado. Pasados aproximadamente 2 minutos, el icono desaparece. Comprobamos que el contador de datos no se ha incrementado	OK
Cambio de posición de paciente	F,I	Para esta prueba nos salimos al exterior, y esperamos a la búsqueda de dos posiciones. Comprobamos que el icono se enciende en cada una de las búsquedas, y apreciamos que la segunda búsqueda es más rápida que la primera. Posteriormente, y ya en el servidor central, comprobamos las dos lecturas.	OK

Cortar la comunicación en mitad lectura	F,I	Nos es imposible realizar esta prueba, ya que el GPS se encuentra integrado en el dispositivo PDA.	No Realizada
---	-----	--	--------------

- UC-0008: Detectar Situaciones de Riesgo

Este caso de uso represente la evaluación de los datos recogidos en el sistema con el fin de identificar cambios en las situaciones de riesgo y por tanto cambios de estado del paciente.

Para la realización de estas pruebas hemos utilizado el SensorTest, el cual realiza un ciclo de un entero de 1 a 5. Las reglas configuradas que cuando el contador se encuentre en...

- 1,2: El paciente se encuentra en estado Verde.
- 3,4: El paciente se encuentra en estado Amarillo.
- 5: El paciente se encuentra en estado Rojo.

Prueba	Tipo	Resultado	Valoración
Cambio de Estado de Paciente a Amarillo	F,I	Comprobamos que el paciente se encuentra en estado Verde, y esperamos a que el contador del sensor SensorTest alcance el numero 3. Comprobamos que el contador de Datos se incrementa en uno, y a continuación el icono de estado pasa al correspondiente del estado Amarillo del paciente. En ese momento comprobamos que el contador de Notificaciones se ha incrementado también en uno.	OK
Cambio de Estado de Paciente a Rojo	F,I	Comprobamos que el paciente se encuentra en estado Amarillo, y esperamos a que el contador del sensor SensorTest alcance el numero 5. Comprobamos que el contador de Datos se incrementa en uno, y a continuación el icono de estado pasa al correspondiente del estado Rojo del paciente. En ese momento comprobamos que el contador de Notificaciones se ha incrementado también en uno.	OK
Cambio de Estado de Paciente a Verde	F,I	Comprobamos que el paciente se encuentra en estado Rojo, y esperamos a que el contador del sensor SensorTest alcance el numero 1. Comprobamos que el contador de Datos se incrementa en uno, y a continuación el icono de estado pasa al correspondiente del estado Verde del paciente. En ese momento comprobamos que el contador de Notificaciones se ha incrementado también en uno.	OK

Aplicación para la monitorización remota de pacientes

- UC-0009: Enviar Datos al Servidor Central

Esta caso de uso representa el envío de los datos recogidos en el PDA para que sean interpretados por el personal medico.

Prueba	Tipo	Resultado	Valoración
Forzar envío de datos al servidor	F,I	Comprobamos que tenemos datos en la aplicación a través del contador de Datos del formulario principal. Utilizamos la opción de Enviar Datos del menú. Comprobamos que el icono de acceso al servidor se activa durante el envío de datos, y cuando este acaba, comprobamos que el contador de Datos se encuentra en cero.	OK
Envío de datos programado al servidor	F,I	Comprobamos que tenemos datos en la aplicación a través del contador de Datos del formulario principal. Esperamos a que se cumpla el intervalo Comprobamos que el icono de acceso al servidor se activa durante el envío de datos, y cuando este acaba, comprobamos que el contador de Datos se encuentra en cero.	OK
Envío de datos automático al servidor, tras generar una notificación	F,I	Comprobamos que tenemos datos en la aplicación a través del contador de Datos del formulario principal. Simulamos el cambio en el estado del paciente para el envío de notificaciones y datos. Comprobamos que el icono de acceso al servidor se activa durante el envío de datos, y cuando este acaba, comprobamos que el contador de Datos se encuentra en cero.	OK
Quitar del alcance Wifi PDA	F,I	Comprobamos que tenemos datos en la aplicación a través del contador de Datos del formulario principal. Utilizamos la opción de Enviar Datos del menú. Comprobamos que el icono de acceso al servidor se activa durante el envío de datos, pero pronto termina y el icono desaparece. Comprobamos que el contador de Datos no ha cambiado.	OK
envío de datos con el servidor central desconectado	F,I	Comprobamos que tenemos datos en la aplicación a través del contador de Datos del formulario principal. Utilizamos la opción de Enviar Datos del menú. Comprobamos que el icono de acceso al servidor se activa durante el envío de datos, pero pronto termina y el icono desaparece. Comprobamos que el contador de Datos no ha cambiado.	OK

- UC-0010: Notificar Situación de Riesgo

Este caso de uso representa el envío de los cambios de estado de las situaciones de riesgo, y por tanto la notificación de alerta al personal medico de los cambios de estado del paciente.

Prueba	Tipo	Resultado	Valoración
Forzar envío de notificaciones al servidor central	F,I	Comprobamos que tenemos notificaciones en la aplicación a través del contador de Notificaciones del formulario principal. Utilizamos la opción de Enviar Datos del menú. Comprobamos que el icono de acceso al servidor se activa durante el envío de notificaciones, y cuando este acaba, comprobamos que el contador de Notificaciones se encuentra en cero.	OK
Envío Programado al servidor	F,I	Comprobamos que tenemos notificaciones en la aplicación a través del contador de Notificaciones del formulario principal. Esperamos a que se cumpla el intervalo. Comprobamos que el icono de acceso al servidor se activa durante el envío de notificaciones, y cuando este acaba, comprobamos que el contador de Notificaciones se encuentra en cero.	OK
Envío automático al servidor, tras generar una notificación	F,I	Comprobamos que tenemos notificaciones en la aplicación a través del contador de Notificaciones del formulario principal. Simulamos el cambio en el estado del paciente para el envío de notificaciones y datos. Comprobamos que el icono de acceso al servidor se activa durante el envío de notificaciones, y cuando este acaba, comprobamos que el contador de Notificaciones se encuentra en cero.	OK
Quitar del alcance Wifi PDA	F,I	Comprobamos que tenemos notificaciones en la aplicación a través del contador de Notificaciones del formulario principal. Utilizamos la opción de Enviar Datos del menú. Comprobamos que el icono de acceso al servidor se activa durante el envío de notificaciones, pero pronto termina y el icono desaparece. Comprobamos que el contador de Notificaciones no ha cambiado.	OK
Desconectar el servidor	F,I	Comprobamos que tenemos notificaciones en la aplicación a través del contador de Notificaciones del formulario principal. Utilizamos la opción de Enviar Datos del menú. Comprobamos que el icono de acceso al servidor se activa durante el envío	OK

		de notificaciones, pero pronto termina y el icono desaparece. Comprobamos que el contador de Notificaciones no ha cambiado.	
--	--	---	--

3. Duración de Batería

En la sección anterior hemos comprobado el buen funcionamiento de los casos de uso, obtenidos a partir de los requisitos funcionales recogidos en la fase de análisis.

Nos ha parecido interesante introducir unas pruebas y comprobar algunos de los requisitos orientados a la optimización de la batería. Por tanto hemos definido una serie de pruebas, para comprobar el buen funcionamiento de la aplicación durante el desarrollo de un día habitual para cualquier paciente.

Sirva como orientación a las pruebas, que el intervalo de lectura del pulsioxímetro se ha fijado en 2 minutos, y el intervalo del GPS se ha fijado en 10 minutos.

Hemos contemplado los siguientes perfiles de uso:

- Modo Diurno

Se trata del uso normal de la aplicación. La particularidad de este perfil es la movilidad del paciente, y la disponibilidad de redes donde enviar los datos, siempre dentro del complejo residencial, o en sus alrededores. Tiene el problema de que el GPS se encuentra sin cobertura y agota el tiempo necesario para la lectura de la posición, aproximadamente 2 minutos.

El resultado es que el dispositivo PDA se mantiene funcionando durante 10 horas, y aparentemente gasta la mayor parte de la energía en detectar la localización mediante GPS, y en el mantenimiento de la red Wifi.

- Modo Nocturno

Se trata del uso de la aplicación mientras el paciente esta dormido. Se trata de un perfil de funcionamiento, que en el caso habitual, el dispositivo PDA debiera estar cargándose para su uso durante el modo diurno. Por tanto no aporta nada al estudio del consumo, pero por la disponibilidad de redes, este perfil es similar al modo diurno.

- Modo Travesía

Se trata del uso de la aplicación fuera del recinto del complejo residencial, de tal forma que el GPS se encuentra recogiendo la localización del paciente, mientras que el Bluetooth que recoge la información de los sensores. La red Wifi esta en modo descubrimiento, pero no en estado activo, debido a que el PDA se encuentra fuera del alcance de esta.

La duración de las baterías en este modo ha sido sorprendente y ha alcanzado 16h. En este modo el GPS tarda menos de 10 segundos, en vez de 2 minutos, en localizar al paciente, ya que este se encuentra a cielo abierto. Esto provoca un ahorro de baterías asombroso que permite el funcionamiento continuo del dispositivo; pero por el contrario, la red Wifi no esta disponible para el envío de información.

MANUAL DE USUARIO

1. Introducción

Este documento muestra el manejo de la aplicación PDA, mostrando todos los pasos necesarios para realizar cada una de las tareas desarrolladas. Organizaremos el manual en función del tipo de usuario que tienen que utilizarlo.

De la misma forma, se incluirá un punto que explique el proceso de instalación de la aplicación en cualquier dispositivo que cumpla con los requisitos hardware de la aplicación.

2. Manual de Usuario

2.1. Inicio de la aplicación

Para ejecutar la aplicación, utilizaremos el icono que tenemos en Inicio\Programas\Monitorizacion Remota, el cual nos dará paso a la pantalla principal.



Figura 1. Pantalla Programas.

La pantalla inicial se trata de la Pantalla de Paciente, y se trata de la pantalla principal de la aplicación. Realizamos a continuación una explicación de la distribución de esta pantalla, ya que todas las pantallas seguirán el mismo patrón.

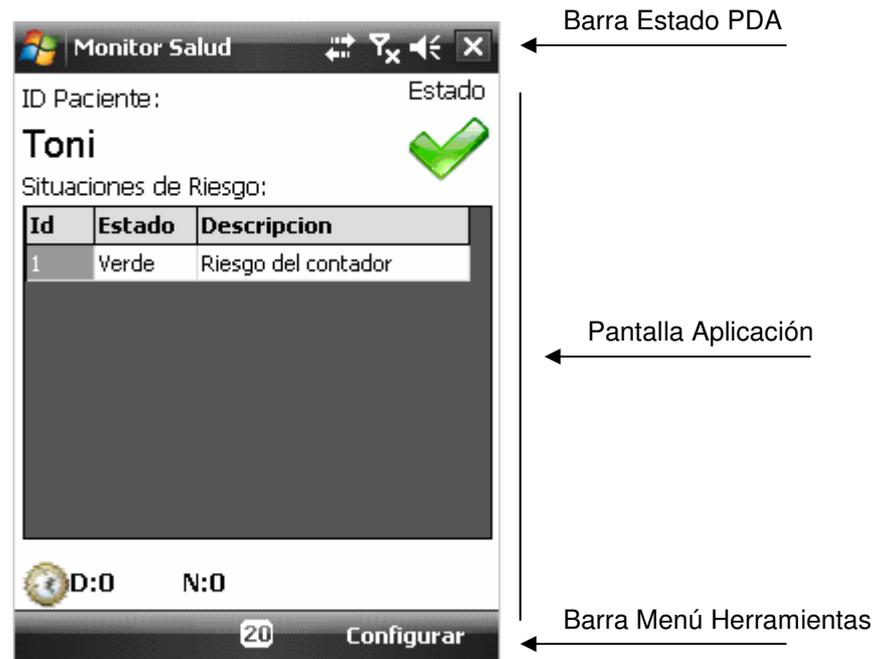


Figura 2. Pantalla Paciente

La distribución de la pantalla es la siguiente:

- Barra Estado PDA

Se trata de una barra propia del Sistema Operativo, y que nos da información relevante sobre el estado del PDA. Son importantes los avisos de disponibilidad de redes y los avisos de falta de batería.



Figura 3. Barra estado PDA.

- Pantalla de Aplicación

Se trata de la pantalla funcional de la aplicación y donde se encontraran toda la información necesaria para el buen uso de la aplicación.

- Barra de Menú Herramientas

Se trata de la barra inferior, donde dependiendo de las pantallas dispondremos diferentes opciones de menú, para realizar diferentes acciones sobre la aplicación.

2.2 Pantalla Estado Paciente

Se trata de la pantalla principal de la aplicación, y por tanto la primera pantalla que aparece tras arrancar el PDA. Nos muestra toda la información relativa al paciente configurado, como su estado de salud y por tanto el estado de sus reglas.



Figura 4. Pantalla Estado Paciente

Es importante destacar el icono de estado que disponemos en la parte superior derecha, y el cual nos permite de una forma grafica y muy rápida conocer el estado del paciente.

Este icono puede tener 3 gráficos, representando cada uno de los estados del paciente, los cuales explicamos a continuación:

- Estado Verde
Se trata del estado que indica que el paciente se encuentra Sano, debido a que ninguna de las reglas que forman las diferentes situaciones de riesgo, se encuentran afectadas.
- Estado Amarillo
Se trata del estado que indica que el paciente tiene alguna regla en mal estado, pero que la situación de riesgo todavía no se ha alcanzado.
- Estado Rojo
Se trata del estado que indica que para una situación de riesgo, todas sus reglas de riesgo están afectadas, y por tanto el paciente debe acudir a un centro medico.



Figura 5. Estados del Paciente

A continuación tenemos una tabla con las situaciones de riesgo configuradas en el paciente junto a una ligera descripción de estas y el estado en que se encuentran. De la misma forma que el paciente, una situación de riesgo puede encontrarse en los estados Verde, Amarillo y Rojo, indicando estos estados la misma situación

Al tratarse de la pantalla principal, también se han incluido un conjunto de iconos y contadores que nos muestran el funcionamiento de cada una de los componentes de la aplicación, indicándonos que esta haciendo la aplicación en segundo plano. Exponemos, de izquierda a derecha, la explicación de cada uno de los iconos y contadores.



Figura 6. Iconos de Estado Aplicación

- **Icono Reloj**
Indica si el reloj que programa las lecturas y entrega de datos esta funcionando.
- **Contador Datos**
Indica el número de datos que están almacenados en la cola de envío, y por tanto pendientes de enviar al servidor.
- **Contador Notificaciones**
Indica el numero de notificaciones que están almacenadas en la cola de envío, y por tanto que están pendientes de enviar al servidor.
- **Icono Comunicación con Servidor**
Indica si la aplicación esta conectada al servidor central, y se encuentra enviando, o datos, o notificaciones.
- **Icono Comunicación con GPS**
Indica si la aplicación esta conectada con el GPS, intentando obtener la posición del paciente.
- **Icono Comunicación con GPS**
Indica si la aplicación esta conectada con algún sensor, intentando obtener sus datos.

Por ultimo tenemos la barra de menús, que nos permite navegar entre las otras pantallas de la aplicación.

2.3. Pantalla Configuración General

Se trata de la pantalla que nos permite realizar parte de la configuración general de la aplicación. Es importante destacar la ventana de Log, la cual nos permite conocer que esta ocurriendo en todo momento con la aplicación.



Figura 7. Pantalla Configuración General

A partir de la siguiente pantalla se pueden realizar las siguientes tareas:

- Iniciar/Detener la aplicación.
Utilizando el botón junto a la etiqueta Reloj, realmente lo que estamos haciendo es detener el reloj, y por tanto la lectura y envío programados de datos. Es útil para configurar la aplicación, impidiendo que esta siga recogiendo datos del paciente que tenga configurado.
- Truncar Tablas
Este botón permite borrar la cola de envío de datos y notificaciones, dejando ambas con cero elementos.
- Enviar Datos
Este botón se utilizar para enviar los datos y notificaciones que se encuentren en la cola de envío al servidor configurado, de tal forma que la cola quede vacía después.
- Cerrar Aplicación
Esta utilidad es útil para realizar un cierre de la aplicación, asegurando que se hace un cierre seguro del Log y de las colas de envío de datos y notificaciones.

Por ultimo tenemos el Log. Esta parte del formulario nos permite conocer todos lo que ocurre en la aplicación, desde los sensores agregados, pasando por el estado de cada lectura, hasta conocer que ocurre con los datos enviados al servidor.

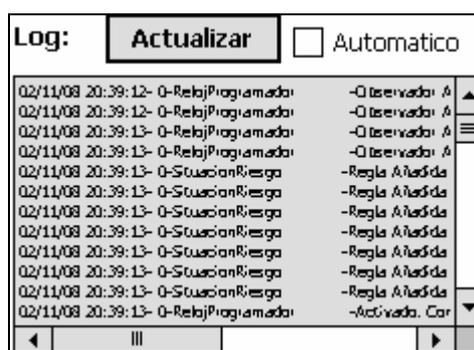


Figura 8. Log de la aplicación.

Para poder visualizar el Log, tenemos dos controles. Por una parte, tenemos el botón de Actualizar, el cual nos permite actualizar la pantalla con los últimos mensajes recogidos por el sistema. Por otra parte, tenemos un check denominado Automático, que si activamos veremos como se actualiza la pantalla automáticamente.

2.4. Pantalla Estado Sensores

Esta pantalla nos permite conocer el estado de los sensores conectados con la aplicación, de tal forma que podemos conocer los datos básicos de estos. Por tanto los datos que disponemos de los sensores son los siguientes:

- Id: Es el identificador del sensor.
- Descripción: Breve indicación sobre el sensor.
- Int: Abreviatura de intervalo. Nos indica el número de segundos que hay entre dos lecturas consecutivas.
- Estado: Se trata del ultimo estado en el que se encuentra el sensor, los cuales pueden ser Inicial, Leyendo, DatoOK o Error Lectura.
- Hora: La hora de la ultima lectura.



Figura 9. Pantalla Estado Sensores

2.5. Pantalla Estado GPS

Esta pantalla permite conocer el estado de la comunicación con el GPS, y los últimos datos recogidos por este. Por tanto los datos que disponemos del GPS son Estado, Altitud, Longitud, Latitud y Hora.

Además disponemos de un campo adicional denominado Cadena GPS. Este campo permite que personal especializado pueda saber que datos se están recogiendo del GPS. A dicho campo le acompaña el botón Actualizar, el cual deberemos utilizar para conocer el contenido de la cadena.

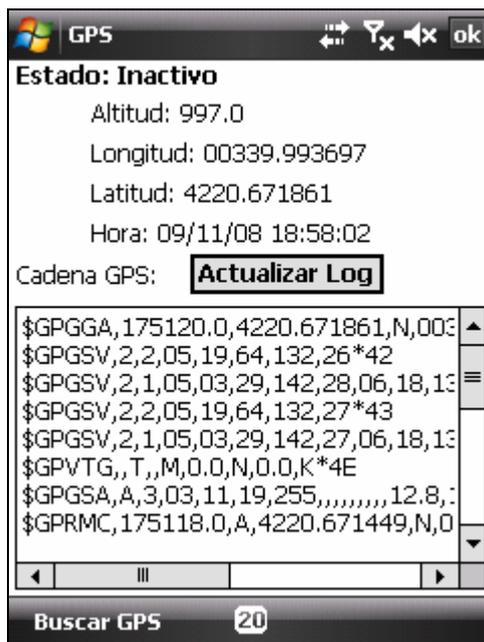


Figura 10. Pantalla Estado GPS

2.6. Pantalla Configuración Paciente

Se trata de la pantalla que permite configurar un paciente en la aplicación. Recordemos que los perfiles de los pacientes se encuentran almacenados en el servidor central; y por tanto los pasos para integrar un perfil en la aplicación serán los siguientes:

- Recoger el perfil del servidor central
- Comprobar el perfil
- Aplicar el perfil

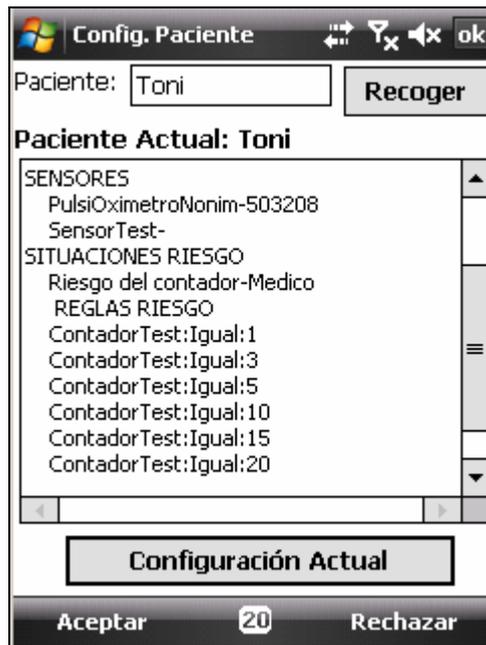


Figura 11. Pantalla Configurar Paciente

Describimos a continuación los pasos necesarios para configurar un perfil en la aplicación.

1. **Recoger el Perfil.**
Indicamos en caja de texto superior el paciente para el cual queremos recoger el perfil del servidor, y pulsamos el botón recoger. Si el servidor central esta disponible el perfil se descargara en el PDA y se mostrara por pantalla; En el caso de que el servidor central no este disponible, se mostrara un mensaje por pantalla indicando la situación.
2. **Comprobar el Perfil**
Se muestra un resumen del perfil por pantalla, de tal forma que el personal medico puede comprobar que la configuración de paciente es correcta, y determinar los sensores que se van a utilizar con el paciente.
3. **Aplicar el perfil**
En la barra de menú de herramientas, tenemos dos opciones: Aceptar y Rechazar. Para aplicar el perfil utilizamos el botón Aceptar, de tal forma que tras el reinicio de la aplicación, la nueva configuración queda aplicada.

También disponemos de un botón denominado “Configuración Actual”, el cual nos permite conocer el perfil aplicado en la actualidad en la aplicación.

CONCLUSIONES

Consideramos que la aplicación desarrollada cumple las expectativas que nos habíamos marcado. Se trata de una herramienta que permite dar un paso adicional en TeleAsistencia, y que permite una monitorización activa de los pacientes, reduciendo los tiempos de respuesta frente a incidencias; de tal forma que la posibilidad de éxito en una intervención medica ante situaciones de emergencia sea mayor.

Este tipo de TeleAsistencia facilita el cumplimiento de otros de los objetivos de la aplicación, en relación con la mejora de la Calidad de Vida del paciente; ello es posible gracias a la monitorización mediante sensores inalámbricos, la integración de diferentes subsistemas y el comportamiento activo del dispositivo móvil y portátil (PDA), que permite gestionar servicios y comunicaciones, proporcionando al paciente una gran libertad de movimientos.

Desde el punto de vista técnico, consideramos que las pruebas realizadas son satisfactorias, ya que el presente proyecto ha conseguido integrar dispositivos estándar, para dar una solución real a un problema creciente, tanto en número de personas como en costes de asistencia; y así cumplir con uno de nuestros principales objetivos: dar una solución al problema con un coste reducido.

Como posible mejora, consideramos que en posteriores versiones del desarrollo debemos introducir mecanismos con el objetivo de optimizar el uso de la batería, ya que aunque se ha hecho un esfuerzo en este sentido, el resultado no ha sido tan satisfactorio como esperábamos.

Pensamos que el avance tecnológico es constante, y cada día se encuentran nuevos dispositivos que encajan mejor en el proyecto, ya sea porque su precio se ha reducido, porque el coste energético de comunicaciones con los dispositivos es menor, o simplemente, porque el tamaño de estos se ha reducido. No obstante, todavía queda mucho por recorrer, sobre todo en el campo de los sensores inalámbricos, ya que todavía no se dispone de una variedad de dispositivos, ó al menos dispositivos que no utilicen un protocolo de comunicación propietario.

A pesar de algunos inconvenientes encontrados en la implementación (en relación sobre todo con las dificultades para identificar características de pasarelas residenciales), consideramos adecuada la orientación del proyecto AIVI como marco general propuesto para las aplicaciones desarrolladas. Cabe esperar que futuras adaptaciones de la aplicación desarrollada serán utilizadas por pacientes en complejos residenciales, escenarios abiertos o cualquier tipo de persona que se encuentre con algún tipo de grado de dependencia. Por consiguiente, esta aplicación proporciona un apoyo activo al personal medico, permitiendo que un solo responsable medico pueda asistir a un mayor numero de pacientes.

Aunque queda mucho camino por recorrer, un objetivo prioritario debe ser alcanzar un sistema medico de asistencia mas informatizado, que facilite una mayor disponibilidad de la información espacio-temporal sobre el estado de los pacientes. De este modo, se pretende reducir los tiempos de respuesta y facilitar una rápida asistencia que pueda llegar a porcentajes crecientes de la población, reduciendo los costes de una monitorización en entidades asistenciales y mejorando la calidad de vida de los ciudadanos.

BIBLIOGRAFIA

Fuentes bibliográficas consultadas:

- ❖ Grady Booch, Ivar Jacobson, James Rumbaugh, “El Lenguaje Unificado de Modelado”, Addison Wesley, 1999
- ❖ Patrick Cauldwell, Rajesh Charla, Vivek Chopra , “Servicios Web XML”, Anaya Multimedia, 2001
- ❖ Ivar Jacobson, Grady Booch, James Rumbaugh, “El Proceso Unificado de Desarrollo de Software”, Addison Wesley, 1999
- ❖ Craig Larman, “UML y patrones”, Prentice Hall, 1999
- ❖ Benoit Marchal, “XML con Ejemplos”, Prentice Hall, 2001
- ❖ Joseph Schmuller, “Aprendiendo UML en 24 Horas”, Prentice Hall, 2001

Referencias web:

Libros

- José Antonio González Seco, “El lenguaje de programación C#”, <http://www.josanguapo.com/>, visitado el 15/03/2008
- Guillermo Som, Unai Zorrilla, Jorge Serrano, “Microsoft Visual Studio 2005”, <http://www.elguille.info/>, visitado el 15/03/2008

Protocolo Comunicación GPS

- <http://www.nmea.org>
- <http://lists.freebsd.org/pipermail/freebsd-mobile/>

Dispositivos Móviles y Programación

- <http://www.todopocketpc.com/>
- <http://www.pdaexpertos.com/>
- <http://blogs.msdn.com/windowsmobile/default.aspx>
- <http://blogs.msdn.com/windowsmobile>

Visual Studio y Lenguaje C#

- <http://www.c-sharpcorner.com>
- <http://www.csharp-station.com>
- <http://www.codehound.com/csharp>
- <http://www.csharpindex.com>
- <http://www.developersdex.com/csharp>

Consumo Energía en Dispositivos Móviles

- <http://community.opennetcf.com>
- <http://www.eggheadcafe.com>

Varios

- <http://www.dotnetwire.com>
- <http://msdnfan.blogspot.com/>
- <http://code.msdn.microsoft.com/>

APENDICES

Apéndice A. Microsoft .Net y Net Compact Framework

1. Introducción.

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones. Basado en esta plataforma, Microsoft desarrolla una estrategia horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems. Debido a las ventajas que la disponibilidad de una plataforma de este tipo puede darle a las empresas de tecnología y al público en general, muchas otras empresas e instituciones se han unido a Microsoft en el desarrollo y fortalecimiento de la plataforma .NET, ya sea por medio de la implementación de la plataforma para otros sistemas operativos aparte de Windows (Proyecto Mono de Ximian/Novell para Linux/MacOS X/BSD/Solaris), el desarrollo de lenguajes de programación adicionales para la plataforma (ANSI C de la Universidad de Princeton, NetCOBOL de Fujitsu, Delphi de Borland, entre otros) o la creación de bloques adicionales para la plataforma (como controles, componentes y bibliotecas de clases adicionales); siendo algunas de ellas software libre, distribuibles ciertas bajo la licencia GPL.

Con esta plataforma Microsoft incursiona de lleno en el campo de los Servicios Web y establece el XML como norma en el transporte de información en sus productos y lo promociona como tal en los sistemas desarrollados utilizando sus herramientas.

.NET intenta ofrecer una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones - o como la misma plataforma las denomina, soluciones - permitiendo a su vez una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.



Figura 1. Resumen Arquitectura .Net

2. .NET Framework

El framework o marco de trabajo, constituye la base de la plataforma .NET y denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.

Bajo el nombre .NET Framework se encuentran reunidas una serie de normas impulsadas por varias compañías además de Microsoft (como Hewlett-Packard , Intel, IBM, Fujitsu Software, Plum Hall, la Universidad de Monash e ISE), entre las cuales se encuentran:

La norma que define las reglas que debe seguir un lenguaje de programación para ser considerado compatible con el marco de trabajo .NET (ECMA-335, ISO/IEC 23271). Por medio de esta norma se garantiza que todos los lenguajes desarrollados para la plataforma ofrezcan al programador un conjunto mínimo de funcionalidad, y compatibilidad con todos los demás lenguajes de la plataforma.

La norma que define el lenguaje C# (ECMA-334, ISO/IEC 23270). Este es el lenguaje insignia del marco de trabajo .NET, y pretende reunir las ventajas de lenguajes como C/C++ y Visual Basic en un solo lenguaje.

La norma que define el conjunto de funciones que debe implementar la librería de clases base (BCL por sus siglas en inglés) (incluido en ECMA-335, ISO/IEC 23271). Tal vez el más importante de los componentes de la plataforma, esta norma define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Ximian pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del software libre o código abierto.

Los principales componentes de .NET Framework son:

- El conjunto de lenguajes de programación
- La Biblioteca de Clases Base o BCL
- El Entorno Común de Ejecución para Lenguajes o CLR

Debido a la publicación de la norma para la infraestructura común de lenguajes (CLI por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje. Algunos de los lenguajes desarrollados para .NET Framework son: C#, Visual Basic, Delphi (Object Pascal), C++, J#, Perl, Python, Fortran y Cobol.NET.

3. Objetivos del .NET Framework

El .NET Framework fue diseñado para cumplir varios objetivos:

- Interoperabilidad.

Un objetivo muy importante es proporcionar la posibilidad de interacción entre nuevas y viejas aplicaciones. Para ello, el .NET Framework proporciona mecanismos que permiten el acceso a funcionalidad que está implementada en programas que se ejecutan fuera del entorno de .NET. Por ejemplo, el acceso a componentes COM mediante el EnterpriseServices.

- Motor común de ejecución.

Los programas escritos bajo .NET se compilan a un lenguaje intermedio conocido como Common Intermediate Language o CIL. Microsoft proporciona su propia implementación de

CIL, el conocido como Microsoft Intermediate Language, o MSIL. En la implementación de Microsoft este lenguaje intermedio no es interpretado, ya que se usa una técnica de compilación en tiempo de ejecución (JIT, just-in-time compilation) que traduce el programa a código nativo. La combinación de todos estos conceptos se denomina Common Language Infrastructure (CLI). La implementación de Microsoft del CLI se conoce como el Common Language Runtime (CLR).

- Independencia de lenguaje. El .NET Framework emplea el Common Type System, o CTS. La especificación del CTS define todos los posibles tipos de datos y estructuras de programación soportados por el CLR y cómo pueden o no pueden interactuar con cada uno. Esta característica permite que el .NET Framework soporte el desarrollo de varios lenguajes de programación distintos.

- Base Class Library.

La Base Class Library (BCL) es una biblioteca de tipos disponibles para todos los lenguajes que usan el .NET Framework. La BCL proporciona clases que encapsulan varias funciones comunes, como funciones de lectura y escritura de ficheros, funciones gráficas, funciones de interacción con bases de datos y funciones de manipulación de documentos XML.

- Ayuda a la instalación.

La instalación de software debe ser cuidadosamente gestionado para asegurarse que no interfiere con otro software previamente instalado. El .NET Framework incluye características y herramientas que ayudan a lograr mantener la estabilidad.

- Seguridad.

.NET permite al código ser ejecutado en diferentes niveles de seguridad mediante el uso de “cajas de arena”.

Hay que remarcar que, a pesar que el objetivo principal de .NET Framework sea la independencia de plataforma, Microsoft únicamente ha implementado versiones completas del .NET Framework en sus propios sistemas operativos. Actualmente se está trabajando en implementaciones de .NET Framework en otros sistemas operativos.

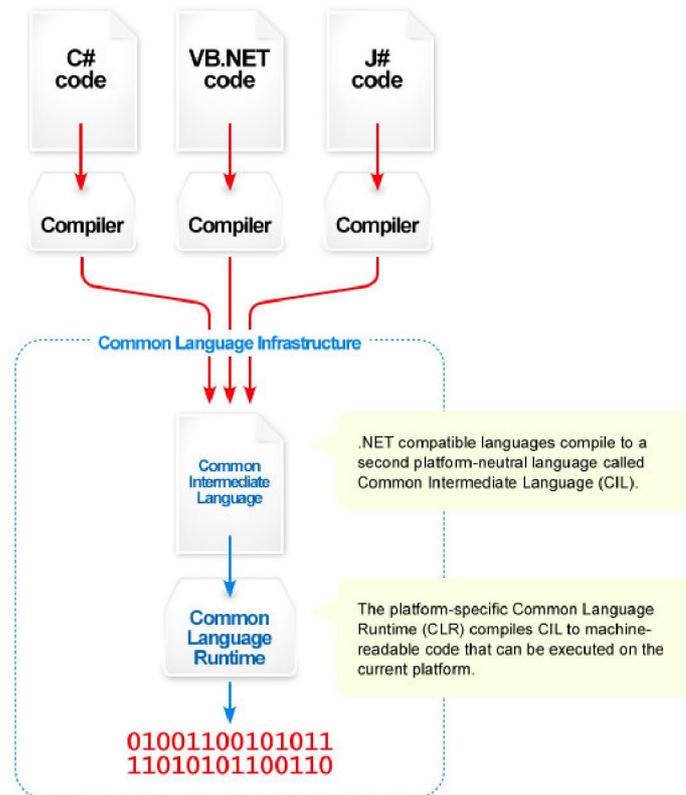


Figura 2. Grafico resumen Compilación

4. Common Language Runtime (CLR)

El CLR es la implementación de Microsoft del CLI (Common Language Infrastructure).

El CLR es el verdadero núcleo del Framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003).

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio (MSIL, Microsoft Intermediate Lenguaje), similar al BYTECODE de Java. Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear ese código MSIL compatible con el CLR.

Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue con .NET independencia de la plataforma hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos, el código ejecutable obtenido, se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.

5. Base Class Library (BCL).

La Librería de Clase Base (BCL) es una librería incluida en el .NET Framework formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR y a las funcionalidades más frecuentemente usadas a la hora de escribir programas. Además, a partir de estas clases prefabricadas el programador puede crear nuevas clases que mediante herencia extiendan su funcionalidad y se integren a la perfección con el resto de clases de la BCL. Por ejemplo, implementando ciertos interfaces podemos crear nuevos tipos de colecciones que serán tratadas exactamente igual que cualquiera de las colecciones incluidas en la BCL.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. A través de las clases suministradas en ella es posible desarrollar cualquier tipo de aplicación, desde las tradicionales aplicaciones de ventanas, consola o servicio de Windows NT hasta los novedosos servicios Web y páginas ASP.NET. Es tal la riqueza de servicios que ofrece que puede crearse lenguajes que carezcan de librería de clases propia y sólo usen la BCL, como C#.

Dado la amplitud de la BCL, ha sido necesario organizar las clases en ella incluida en espacios de nombres que agrupen clases con funcionalidades similares. Por ejemplo, los espacios de nombres más usados son:

Espacio de nombres	Utilidad de los tipos de datos que contiene
System	Tipos muy frecuentemente usados, como los tipos básicos, tablas, excepciones, fechas, números aleatorios, recolector de basura, entrada/salida en consola, etc.
System.Collections	Colecciones de datos de uso común como pilas, colas, listas, diccionarios, etc.
System.Data	Manipulación de bases de datos. Forman la denominada arquitectura ADO.NET.
System.IO	Manipulación de ficheros y otros flujos de datos.
System.Net	Realización de comunicaciones en red.
System.Reflection	Acceso a los metadatos que acompañan a los módulos de código.
System.Runtime.Remoting	Acceso a objetos remotos.
System.Security	Acceso a la política de seguridad en que se basa el CLR.
System.Threading	Manipulación de hilos.
System.Web.UI.WebControls	Creación de interfaces de usuario basadas en ventanas para aplicaciones Web.
System.Windows.Forms	Creación de interfaces de usuario basadas en ventanas para aplicaciones estándar.
System.XML	Acceso a datos en formato XML.

6. .NET Compact Framework

El .NET Compact Framework es una versión "reducida" del .NET Framework y se utiliza en los Windows Mobile, o en los equipos que utilicen el Windows CE o el Windows CE .NET. Para utilizarlo, es necesario el Visual Studio .NET y el SDE (Smart Device Extensions), aunque a partir de Visual Studio 2005, el SDE se encuentra integrado en el IDE; ya que el SDE es el que permite crear en VS .NET proyectos para los Windows Mobile, tanto para VB .NET como para C#.

.NET Compact Framework es un entorno independiente del hardware para ejecutar programas en dispositivos como asistentes digitales personales (PDA), teléfonos móviles y descodificadores. Se ejecuta en el sistema operativo Microsoft Windows CE y se basa en una reconstrucción de Common Language Runtime (CLR) diseñada para funcionar eficazmente cuando se ejecutan programas en dispositivos con limitaciones de recursos. .NET Compact Framework lleva el código administrado y los servicios Web XML a los dispositivos y proporciona ventajas como la seguridad de tipos, la recolección de elementos no utilizados, el control de excepciones y la seguridad. .NET Compact Framework es un subconjunto de la biblioteca de clases .NET Framework y también contiene clases diseñadas expresamente para dispositivos con limitaciones de recursos. Hereda la arquitectura .NET Framework completa de Common Language Runtime y la ejecución de código administrado.

.NET Compact Framework hereda la arquitectura .NET Framework completa de Common Language Runtime para ejecutar código administrado. Proporciona interoperabilidad con el sistema operativo Windows CE de un dispositivo para tener acceso a funciones nativas e integrar los componentes nativos favoritos en una aplicación. Puede ejecutar aplicaciones nativas y administradas de manera simultánea. El host del dominio de aplicación, que también es una aplicación nativa, inicia una instancia del Common Language Runtime para ejecutar el código administrado.

.NET Compact Framework utiliza el sistema operativo Windows CE para la funcionalidad central y para diversas características específicas de dispositivos. Varios tipos y ensamblados, como los de los formularios Windows Forms, gráficos, dibujos y servicios Web, se han recompilado para que se ejecuten eficazmente en los dispositivos, en lugar de copiarse de .NET Framework completo.

En la ilustración siguiente se resume la arquitectura de la plataforma .NET Compact Framework.

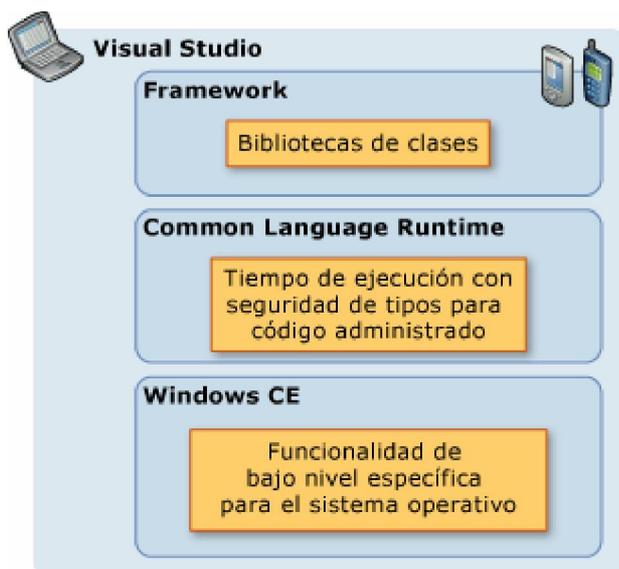


Figura 3. Plataforma .NET Compact Framework

Con respecto a los servicios web, hay que destacar que el .NET Compact Framework permite a las aplicaciones invocar métodos de servicios web, pero no crearlos ni alojarlos en el dispositivo portátil.

7. Desarrollo bajo .NET: Visual Studio.

Visual Studio 2005 es el entorno de programación con el que se ha desarrollado todo el proyecto. Visual Studio .NET es un IDE desarrollado por Microsoft a partir de 2002. Es para el sistema operativo Microsoft Windows y la última versión en funcionamiento del IDE, Visual Studio .NET soporta los nuevos lenguajes .NET: C#, Visual Basic .NET y Managed C++, además de C++. Visual Studio .NET puede utilizarse para construir aplicaciones dirigidas a Windows (utilizando Windows Forms), Web (usando ASP.NET y Servicios Web) y dispositivos portátiles (utilizando .NET Compact Framework).

La característica más notable del IDE es su soporte de los nuevos lenguajes .NET. Los programas desarrollados en esos lenguajes no se compilan a código máquina ejecutable (como por ejemplo hace C++) sino que son compilados a algo llamado CIL. Cuando los programas ejecutan la aplicación CIL, ésta es compilada en ese momento al código de máquina apropiado para la plataforma en la que se está ejecutando. Mediante este método, Microsoft espera poder soportar varias implementaciones de sus sistemas operativos Windows (como Windows CE). Los programas compilados a CIL pueden ejecutarse sólo en plataformas que tengan una implementación de .NET framework. Es posible ejecutar programas CIL en Linux o en Mac OS X utilizando algunas implementaciones .NET que no pertenecen a Microsoft, como Mono y DotGNU.

8. C#.

C# es el lenguaje de programación empleado para elaborar el cliente PDA del proyecto, se trata de un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (más notablemente de Delphi y Java). C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

Aunque C# forma parte de la plataforma .NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Aunque aún no existen, es posible implementar compiladores que no generen programas para dicha plataforma, sino para una plataforma diferente como Win32 o UNIX.

9. ASP.NET

ASP.NET es el lenguaje de programación empleado para elaborar el servidor web del proyecto. ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

Apéndice B. Servicios Web XML

1. Introducción.

Un servicio Web (en inglés Web service) es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Los servicios Web XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Los protocolos que soportan los servicios web se comunican normalmente por el puerto 80, y basándose en HTTP, métodos GET y PUT. Esto hace que podamos acceder a ellos al igual que lo hacemos en una página web. La diferencia entre una página web y un servicio Web, es que la página la visita cualquier individuo interesado, mientras que el servicio sólo lo visitan programas que lo requieren.

2. Los protocolos.

Hay un convenio generalizado que nos da a entender que los Servicios Web se invocan en Internet por medio de protocolos estándar basados en XML. Hoy en día hay dos grandes tendencias: XML-RPC y SOAP. A la hora de programar un servicio web, hay que decidir qué protocolo usar, porque un protocolo es incompatible con el otro. De modo que si programamos nuestro servicio web con XML-RPC, no podremos invocarlo desde un lenguaje de programación que trabaje con SOAP, como por ejemplo .Net de Microsoft.

Tanto SOAP (Protocolo de acceso a objetos simple, Simple Object Access Protocol) como XML-RPC son lenguajes de mensajería basada en XML, estandarizados por el consorcio W3C, que especifican todas las reglas necesarias para ubicar servicios Web XML, integrarlos en aplicaciones y establecer la comunicación entre ellos.

Brevemente, la diferencia entre SOAP y XML-RPC es su complejidad. XML-RPC está diseñado para ser sencillo. SOAP por el contrario está creado con idea de dar un soporte completo y minucioso de todo tipo de servicios web. La curva de aprendizaje de XML-RPC es muy suave, por lo que un programador novato en este campo, puede conseguir resultados satisfactorios con poco esfuerzo. Con SOAP no pasa esto, pero a cambio, dispones de más potencia. Por ejemplo, con XML-RPC no puedes elegir el conjunto de caracteres a utilizar en tus Servicios Web. En SOAP puedes elegir entre US-ASCII, UTF-8 y UTF-16.

SOAP, a diferencia de XML-RPC, incluye una infraestructura a su alrededor. No es un mero protocolo de comunicación entre ordenadores, sino que además se rodea de términos como WSDL y UDDI.

3. SOAP

SOAP es un protocolo elaborado para facilitar la llamada remota de funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web.

El protocolo SOAP tiene diversas ventajas sobre otras maneras de llamar funciones de manera remota como DCOM, CORBA o directamente en TCP/IP:

- Es sencillo de implementar, probar y usar.
- Es un estándar de la industria, creado por un consorcio del cual Microsoft forma parte, adoptado por W3C (<http://www.w3.org/TR/SOAP/>) y por varias otras empresas.
- Utiliza los mismos estándares de la Web para casi todo: la comunicación se hace mediante HTTP con paquetes virtualmente idénticos; los protocolos de autenticación y encriptación son los mismos; el mantenimiento de estado se hace de la misma forma; se implementa normalmente por el propio servidor Web.
- Atraviesa "firewalls" y routers, que "piensan" que es una comunicación HTTP.
- Tanto los datos como las funciones se describen en XML, lo que permite que el protocolo no sólo sea más fácil de utilizar sino que también sea muy sólido.
- Es independiente del sistema operativo y procesador.
- Se puede utilizar tanto de forma anónima como con autenticación (nombre/clave).

Las solicitudes SOAP se pueden hacer en tres estándares: GET, POST y SOAP. Los estándares GET y POST son idénticos a las solicitudes hechas por navegadores de Internet. SOAP es un estándar similar a POST, pero las solicitudes se hacen en XML y permiten recursos más sofisticados, como pasar estructuras y arrays.

Independientemente de cómo se haga la solicitud, las respuestas siempre son en XML. XML describe perfectamente los datos en tiempo de ejecución y evita los problemas ocasionados por cambios inadvertidos en las funciones, ya que los objetos llamados tienen la posibilidad de validar siempre los argumentos de las funciones, haciendo que el protocolo sea muy sólido.

Así mismo, SOAP define un estándar llamado WSDL, que describe perfectamente los objetos y métodos disponibles a través de páginas XML accesibles por la Web. La idea es la siguiente: quien publica un servicio, crea también estas páginas. Quien quiera llamar el servicio, puede utilizar estas páginas como "documentación" de la llamada y también utilizarlas antes de llamar las funciones para verificar si cambió algo.

4.

WSDL

WSDL son las siglas de Web Services Description Language, se trata del mecanismo que permite describir la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. Así, WSDL se usa a menudo en combinación con SOAP y XML Schema. Un programa cliente que se conecta a un servicio web puede leer el WSDL para determinar que funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de XML Schema. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

5. UDDI.

UDDI son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration. El registro en el catálogo se hace en XML. UDDI es una iniciativa industrial abierta (sufragada por la OASIS) entroncada en el contexto de los servicios Web. UDDI es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.