



**UNIVERSIDADE
DE VIGO**

ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA

Memoria do Traballo de Fin de Grao que presenta

D. Víctor Díaz Seoane

para a obtención do Título de Graduado en Enxeñaría Informática

Ubiko: aplicación web para a localización de pacientes.

MEMORIA



Xuño, 2016

Traballo de Fin de Grao Nº: El 15/16-012

Titor/a: Alma María Gómez Rodríguez

Cotitor: David Ramos Valcárcel

Área de coñecemento: Linguaxes e Sistemas Informáticos

Departamento: Informática

Índice

1.INTRODUCCIÓN	4
1.1.PRESENTACIÓN DEL PROYECTO	4
1.2.IDENTIFICACIÓN DEL PROYECTO	4
1.3.DOCUMENTACIÓN APORTADA	5
2.DESCRIPCIÓN DEL PROYECTO	6
3.TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS	7
4.PLANIFICACIÓN TEMPORAL	8
4.1.METODOLOGÍA DE DESARROLLO	8
4.2.PLANIFICACIÓN INICIAL	8
4.3.DISTRIBUCIÓN REAL DEL TIEMPO	10
4.3.1.Trabajo real realizado	10
4.3.2.Funcionalidades y tareas realizadas en cada <i>Sprint</i>	10
4.3.3.Desvíos	14
5.ESTIMACIÓN DE COSTES	15
5.1.COSTES DE PERSONAL	15
5.2.COSTES DE SOFTWARE	15
5.3.COSTES DE HARDWARE	16
6.PROBLEMAS ENCONTRADOS Y SOLUCIONES	16
6.1.Implementación del Modelo-Vista-Controlador en PHP	16
6.2.Cambio de color de la pestaña actual	16
6.3.Implementación del carrusel	17
7.CONCLUSIONES	17
7.1.Conclusiones sobre las herramientas utilizadas	17
7.2.Conclusiones sobre la planificación y gestión del proyecto	17
7.3.Conclusiones personales	18
8.MEJORAS Y POSIBLES AMPLIACIONES	19
9.BIBLIOGRAFÍA	19

1. INTRODUCCIÓN

1.1.PRESENTACIÓN DEL PROYECTO

Este proyecto pretende resolver el problema asociado a la ubicación en la que se encuentra un paciente en el área de urgencias de una clínica u hospital. Las frecuentes desventajas presentadas en este caso es la pérdida de tiempo por parte del personal sanitario del departamento en el momento de buscar, trasladar, atender, etc. a los múltiples pacientes. Por tanto, a modo de solución, se implementará una aplicación web que permita mantener localizados en todo momento los diferentes pacientes que acuden diariamente a urgencias facilitando así la labor de los sanitarios. De esta manera, el personal sanitario sabrá en cada momento en que sala o consulta se encuentra determinado paciente (Triage, Box, Rayos, etc), únicamente consultando la aplicación.

La aplicación se ha desarrollado únicamente para funcionar correctamente en ordenadores personales con navegadores Google Chrome, Mozilla Firefox o Internet Explorer. Debido al aumento del uso de dispositivos con pantallas “grandes” como es el caso de tablets, las cuales podemos encontrar como herramienta de trabajo en la mayoría de las medianas y grandes empresas, se podría migrar la aplicación con relativa facilidad para asegurar el correcto funcionamiento sobre los citados dispositivos.

1.2.IDENTIFICACIÓN DEL PROYECTO

TÍTULO DEL PROYECTO: UBIKO: SISTEMA WEB PARA LA LOCALIZACIÓN DE PACIENTES.

CÓDIGO: EI 15/16-012.

ALUMNO: VÍCTOR DÍAZ SEOANE.

DNI: 44475785H.

TUTORA: ALMA MARÍA GÓMEZ RODRÍGUEZ.

DNI: 34973525R.

COTUTOR: DAVID RAMOS VALCÁRCEL.

DNI: 44455866V

ÁREA DE CONOCIMIENTO: LENGUAJES Y SISTEMAS INFORMÁTICOS.

DEPARTAMENTO: INFORMÁTICA.

1.3.DOCUMENTACIÓN APORTADA

La documentación entregada para este Trabajo Fin de Grado se divide en 3 partes:

- **Memoria del proyecto:** Documento aquí presente que detallará todo el proceso de desarrollo de la aplicación, incluyendo las desviaciones o problemas surgidos durante la elaboración del mismo.
- **Manual Técnico:** Documento en el que se aporta toda la documentación respectiva a las diferentes fases del proyecto, entre las que podemos encontrar los Modelos de Casos de Uso o el diagrama de diseño referente a la base de datos. Esta documentación será de gran utilidad para posteriores modificaciones realizadas por personas ajenas al proyecto.
- **Manual de Usuario:** Documentación que explica de forma sencilla las diferentes funcionalidades de la aplicación desarrollada con el objetivo de que cualquier persona pueda utilizarla.

Además de la documentación impresa, se entrega un CD que contiene:

- **Documentación en formato digital:** Documentación previamente descrita en formato PDF y DOC.
- **Código desarrollado:** Código fuente del proyecto Ubiko, tanto las clases .php, .js, imágenes etc... como los scripts necesarios para la configuración de la base de datos.

2. DESCRIPCIÓN DEL PROYECTO

La aplicación web Ubiko es un sistema para localizar de forma inequívoca a los diferentes pacientes que acuden al área de urgencias de un centro médico desde que entran hasta que salen. Para ello, el paciente será introducido en la aplicación inmediatamente cuando llegue al área de urgencias, quedando registrados todos los datos necesarios, así como la fecha y la hora.

Una vez registrado el paciente, el sistema mostrará su localización actual. En el momento en que dicho paciente sea atendido, se facilitarán los mecanismos para que su situación se actualice en el sistema mostrando siempre la hora y el tiempo que ha pasado desde una localización a la siguiente.

En caso de que el paciente se encuentre en BOX, el sistema ofrecerá un listado con las camas disponibles resaltando la cama ocupada por el paciente en cuestión.

A mayores, el sistema ofrece la opción de comprobar las estadísticas del hospital: número de pacientes por ubicación y número de pacientes con estados finales, es decir, dados de alta, ingresados o fallecidos.

El objetivo principal es doble, por un lado mantener localizado a cada paciente en todo momento. Y por otro, consecuencia del primer objetivo, reducir el tiempo que emplea el personal sanitario (médicos, enfermeras y celadores) en la búsqueda de los citados pacientes para su traslado, realización de pruebas etc...

Para satisfacer estos requisitos, se ha implementado una aplicación que satisface los siguientes requisitos:

- El sanitario debe poder dar de alta un nuevo paciente
- El personal médico debe ser capaz de modificar la ubicación del paciente seleccionado.
- El personal debe poder consultar la información de los pacientes que sean necesarios mediante una barra de búsqueda.
- El sistema ofrecerá la opción de visualizar las estadísticas del centro hospitalario.
- El sistema ofrecerá la opción de login completa.
- El sistema deberá liberar una cama ocupada cuando su paciente pase a tener un estado final.
- El sistema no debe permitir la inserción de pacientes repetidos.
- El sistema no debe permitir la inserción de ubicaciones erróneas.
- Las estadísticas deben ser claras para cualquier usuario.

3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS

A continuación se enumeran las tecnologías y herramientas empleadas para la realización del proyecto, así como una pequeña justificación de porque son las elegidas. En caso de haberse planteado la utilización de más de una tecnología para una misma funcionalidad, se darán los motivos por los cuales se escogió una u otra.

- **Adobe Photoshop CC:** Adobe Photoshop CC es un programa de edición de imágenes 2D que nos dota de las herramientas necesarias para crear o modificar las imágenes e iconos utilizados en la aplicación. Otras herramientas similares serían Gimp 2 o Adobe Fireworks. Se optó por esta tecnología debido a la experiencia previa en su manejo.
- **Visual Paradigm v12.2:** Visual Paradigm es una herramienta de modelado UML (Unified Modeling Language) utilizado para la elaboración de los diagramas incluidos en el Manual Técnico presente en esta documentación.
- **MySql Workbench 6.3 CE:** herramienta empleada para el modelado de la base de datos. Con este software se permite la creación de tablas y atributos de forma sencilla, así como la generación del código SQL resultante del diseño, listo para importarse en la base de datos final.
- **Sublime Text 3:** Sublime es un editor de texto muy extendido en el ámbito de la programación debido a las diferentes funcionalidades disponibles, por ejemplo, mecanismo de doble pantalla para contrastar ficheros, puntero múltiple para escribir en multilíneas, plugins gratuitos, entre ellos, plugins de apoyo a la programación PHP tan utilizada durante la implementación de este proyecto. Otras herramientas similares son, por ejemplo, Adobe Dreamweaver. Se ha optado por el uso de Sublime debido a su carácter gratuito y su funcionalidad simple.
- **WampServer64:** herramienta para el desarrollo de páginas web. Dota al sistema de un manejador de base de datos local, así como un servidor para poder ejecutar código PHP de forma local. Otras herramientas muy similares serían, por ejemplo, Xamp Server. Se optó por el servidor Wamp debido a la experiencia previa en el uso de esta herramienta.
- **GitHub:** es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. De esta manera se aseguró el salvaguardado en la nube de todo el proyecto, incluida su documentación. GitHub nos permite instalar la herramienta en el escritorio de forma que la actualización del repositorio es muy sencilla.
- **Adobe Fireworks CC:** herramienta de adobe empleada para la extracción de diferentes imágenes de las capturas propuestas para la maquetación.

- **Google Chrome/ Mozilla Firefox/ Internet Explorer 11**: navegadores actuales usados para la visualización del código desarrollado.
- **Microsoft Office Word 2013**: herramienta empleada para la realización de la documentación.

4. PLANIFICACIÓN TEMPORAL

En este apartado se detallan la metodología de desarrollo y la distribución temporal del proyecto. Esta división temporal expone tanto la planificación inicial del proyecto, como la distribución real por *Sprints*.

4.1.METODOLOGÍA DE DESARROLLO

Para la realización de este proyecto, dada la naturaleza cambiante de los requisitos y la poca experiencia en el desarrollo de este tipo de aplicaciones, se decide aplicar una metodología de desarrollo ágil iterativa e incremental, inspirada en desarrollos tipo *SCRUM*.

Este tipo de desarrollo se caracteriza por ejecutar el proyecto en bloques temporales de corta duración y tamaño fijo, en los que se espera obtener un resultado completo (*Sprint*). Hay que poner de manifiesto y dejar lo suficientemente claro que la metodología utilizada NO es *SCRUM*, sino que se toma como referencia la filosofía iterativa e incremental que tiene *SCRUM*.

Cada una de las diferentes fases del proyecto cumplirá:

- En cada uno de los *Sprints* se realizarán las fases de Análisis, Diseño, Codificación y Pruebas.
- Al inicio del *Sprint* se decidirá un objetivo que se pueda lograr en el transcurso de este *Sprint* y se dividirá en tareas para realizarlo.
- Al finalizar cada *Sprint* se evaluará si se ha cumplido el objetivo y se reflexionará sobre lo que realizamos en el *Sprint* anterior para ver qué es lo que ha salido mal para mejorarlo en los *Sprints* posteriores.
- De forma paralela a los *Sprints* se realizará la documentación referente al proyecto.

4.2.PLANIFICACIÓN INICIAL

El proyecto se dividió inicialmente en cuatro *sprints*, cada uno de ellos de 3 semanas de duración (60 horas/*sprint*).

El primer *sprint* se correspondía con la gestión de pacientes de la clínica, en él se definirá y codificará todo lo necesario para la inclusión de pacientes y usuarios en la base de datos, así como modelar las diferentes ventanas de las que constará la aplicación. Se realizarían también las pruebas pertinentes a esta parte.

En el segundo *sprint* se gestionaría todo lo relacionado a los diferentes servicios que ofrece la aplicación, modelando las ventanas de BOX, Rayos etc. Así mismo se realizarían las pruebas necesarias que garantizaran el correcto funcionamiento de esta parte.

El tercer *sprint* se dedicaría al módulo de movimiento de pacientes entre servicios, es decir, la reubicación de cada paciente dependiendo en qué departamento de urgencias se encuentre (si el paciente pasa de sala de espera a consulta por ejemplo). Para finalizar el *sprint* se realizarían las pruebas pertinentes que garantizarían su funcionamiento.

Por último el cuarto *sprint* sería dedicado a la elaboración de todas las pruebas necesarias para asegurar el correcto funcionamiento de la aplicación. De ser detectado algún error, sería modificado el código necesario para solventarlo.

Dedicación semanal prevista (en horas/semana): 25	
Fase	Estimación temporal (en semanas)
• Sprint 1: Gestión de pacientes	3
• Sprint 2: Gestión de servicios para pacientes	3
• Sprint 3: Movimientos de pacientes entre servicios	3
• Sprint 4: Pruebas y modificaciones finales	3
Documentación del proyecto	3
TOTAL PROYECTO	15

En el siguiente diagrama de Gantt se puede ver más claramente la planificación inicial del proyecto.

Nombre	Fecha de ini...	Fecha de fin
• Gestion de Pacientes	1/02/16	21/02/16
• Gestión de Servicios para Pa...	22/02/16	13/03/16
• Movimientos de Pacientes e...	14/03/16	3/04/16
• Pruebas y Modificaciones Fi...	4/04/16	24/04/16
• Documentación del proyecto	1/02/16	25/05/16

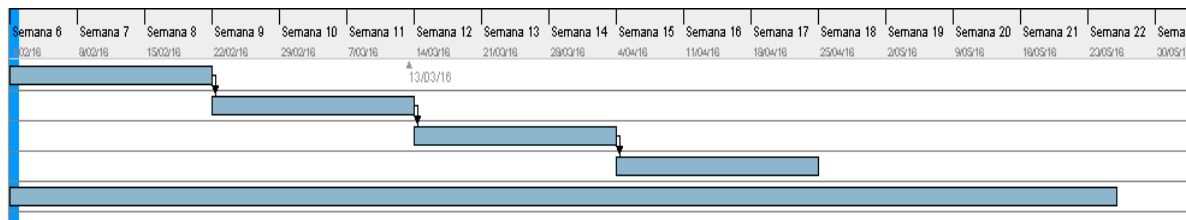


Gráfico 1: Diagrama de Gantt de la aplicación.

4.3.DISTRIBUCIÓN REAL DEL TIEMPO

A pesar de la planificación, por problemas de compatibilidad entre estudios, vida laboral y el proyecto, fue difícil mantener un ritmo constante de trabajo.

A continuación se expone cual fue la distribución real del tiempo.

4.3.1. Trabajo real realizado

En la siguiente tabla se exponen cual fue la duración real en horas de cada uno de los *Sprints* desarrollados, así como la duración real del proyecto completo.

Fase	Estimación temporal (en horas)
• Sprint 1: Gestión de pacientes	80h
• Sprint 2: Gestión de servicios para pacientes	100h
• Sprint 3: Movimientos de pacientes entre servicios	60h
• Sprint 4: Pruebas y modificaciones finales	40h
Documentación del proyecto	40h
TOTAL PROYECTO	320h

4.3.2. Funcionalidades y tareas realizadas en cada *Sprint*

Como método de seguimiento, se establecen una serie de pautas para definir las características de cada uno de los *Sprints*. Para cada *Sprint* se define:

- Objetivo principal.
- Tareas realizadas.
- Resultados obtenidos.
- Duración final del *sprint*.

Sprint 1: Gestión de pacientes

Objetivo principal: maquetación de la ventana de 'logIn', 'Admisión', 'Seguimiento' y 'BOX'.

Tareas a realizar:

- Diseño de la base de datos.
- Implementación de la inserción del paciente en el sistema.
- Maquetación de la ventana de 'LogIn'.
- Maquetación de la ventana de 'Admisión'.
- Maquetación de la ventana de 'Seguimiento'.
- Maquetación de la ventana de 'BOX'.
- Implementación del acceso del usuario al sistema.
- Pruebas.

Resultados:

- Ventanas finales de la aplicación.
- Autenticación correcta de un usuario en el sistema.
- Inserción correcta de un paciente en base de datos.
- Inserción correcta de primeras ubicaciones de un paciente.
- Visualización de la ventana 'Seguimiento' (sin finalizar la maquetación).
- Pruebas y detección de funcionamiento incorrecto.

Duración total: 80 horas.

Sprint 2: Gestión de servicios para pacientes

Objetivo principal: Finalizar maquetaciones, inserción del carrusel y métodos de arrastrado.

Tareas a realizar:

- Maquetación de la ventana de 'Seguimiento'.
- Maquetación de la ventana de 'BOX'.
- Implementación de la búsqueda de un paciente.
- Creación del carrusel.
- Implementación de los métodos drag and drop (arrastrado y soltado) de las ubicaciones.
- Pruebas.

Resultados:

- Ventanas finalizadas de la aplicación.
- Búsqueda correcta de un paciente en el sistema y cargado de los correspondientes datos.
- Inserción correcta de un paciente en base de datos.
- Inserción correcta las ubicaciones en el sistema.
- Pruebas y detección de funcionamiento incorrecto.

Duración total: 100 horas.

Sprint 3: Movimiento de pacientes entre servicios

Objetivo principal: inserción correcta de cualquier tipo de ubicación.

Tareas a realizar:

- Lograr la inserción de las ubicaciones.
- Lograr el pintado de las ubicaciones.
- Insertar paciente en una cama.
- Inserción de estados finales.
- Pruebas.

Resultados:

- Ventana 'BOX' finalizada.
- Inserción correcta de ubicaciones en un paciente.
- Visualización correcta de ubicaciones registradas.
- Tiempos parciales y totales calculados y visualizados para cada ubicación.
- Inserción correcta de ubicaciones finales.
- Liberación de cama.
- Modificación del estado del paciente en base de datos.
- Pruebas y detección de funcionamiento incorrecto.

Duración total: 60 horas.

Sprint 4: Pruebas y modificaciones finales

Objetivo principal: realizar pruebas y finalizar la ventana 'Estadísticas'.

Tareas a realizar:

- Maquetación de la pestaña 'Estadísticas'.
- Funcionalidad de la pestaña 'Estadísticas'.
- Pruebas.

Resultados:

- Ventana de estadísticas finalizada.
- Visualización correcta de gráficos.
- Modificación de inserción de estado final.
- Inclusión de mensajes de aviso.
- Modificación inclusión de ubicaciones repetidas consecutivas.
- Modificación del clonado de las ubicaciones para su arrastre.

Duración total: 40 horas.

Documentación del proyecto

Objetivo principal: documentar íntegramente el proyecto y corregir los últimos errores.

Tareas a realizar:

- Manual de usuario.
- Manual técnico.
- Memoria.
- Corregir errores.

Resultados:

- Aplicación plenamente funcional.
- Documentación finalizada.

Duración total: 40 horas.

4.3.3. Desvíos

En este apartado se definen los diversos problemas encontrados causantes del incumplimiento de la estimación inicial.

Implementación del Modelo-Vista-Controlador en PHP

Debido a la poca experiencia en programación en PHP y el deseo de realizar la funcionalidad de la aplicación siguiendo una arquitectura Modelo-Vista-Controlador, el *sprint 1* tomó más tiempo de lo esperado. Finalmente se consiguió el objetivo y la aplicación esta implementada siguiendo la arquitectura definida anteriormente.

Cambio de color de la pestaña actual

En un principio se pretendía modificar el color de la pestaña en la que el usuario se encontrase mediante código javascript. Debido a que cada arrastre de una ubicación genera un recargado de página, se optó por modificar el CSS mediante una bandera en PHP.

Error en el guardado de ubicaciones

Se detectó un error grave en el guardado de ubicaciones cuyo resultado era el cálculo erróneo de los tiempos parciales por ubicación. La resolución de este problema tomó más horas de lo esperado puesto que hubo que rehacer código de guardado.

Error en el cálculo de pacientes por ubicación.

Se detectó un error en el cálculo del número de pacientes por ubicación necesario para el pintado de los gráficos de la pestaña estadísticas. Su resolución tomó más tiempo del estimado debido a que hubo que rehacer toda la sentencia SQL.

Retraso del plazo de finalización del proyecto por motivos ajenos al proyecto.

En la planificación inicial, el final del proyecto estaba marcado para mediados del mes de mayo de 2016. En cambio por problemas ajenos al desarrollo del proyecto el final de éste se retrasa hasta mediados de junio del mismo año.

Entre las causas de este retraso podemos encontrar:

- Incorporación al mundo laboral del único desarrollador del proyecto.

- Realización de cursos extraacadémicos por parte del único desarrollador del proyecto.

5. ESTIMACIÓN DE COSTES

Los costes estimados se detallan a continuación, diferenciando costes de personal, costes de Software y costes de Hardware.

Esta estimación se realiza teniendo en cuenta las horas dedicadas al proyecto y que es realizada por un único desarrollador.

Recurso	Coste
Costes de Personal	6.400 €
Costes de Software	131.88 €
Costes de Hardware	53.33 €
Total	6585.21 €

5.1.COSTES DE PERSONAL

Teniendo en cuenta las horas realizadas finalmente y que el proyecto fue realizado por un único desarrollador el coste de personal se expone en la siguiente tabla.

Persona	Horas	€/hora	TOTAL
Ingeniero de Software	320	20	6.400€
Total			6.400€

5.2.COSTES DE SOFTWARE

Los costes de software se calculan según los meses durante los cuales se desarrolló el proyecto. El precio de las licencias puede variar durante el transcurso del mismo, por lo que los precios que aquí constan son los vigentes en el momento de redactar este documento. El período de desarrollo del proyecto fueron 4 meses.

Software	€/mes	TOTAL
Adobe Creative Cloud Fotografía	19.66 €	78.64 €
Visual Paradigm Standard	13.31 €	53.24€
Total		131.88€

5.3.COSTES DE HARDWARE

Los costes de hardware se calculan en relación a los meses durante los cuales se desarrolló el proyecto. Dado que el hardware se podrá seguir amortizando una vez terminada la aplicación, se tomará como coste el porcentaje del valor durante los meses que dura el proyecto.

Hardware	Precio	Tiempo/Amortización	Precio/mes	TOTAL
Ordenador Portátil	800 €	5 años	13.33 €	53.33 €
Total				53.33 €

6. PROBLEMAS ENCONTRADOS Y SOLUCIONES

En este apartado se exponen los problemas más graves (que más tiempo de implementación supusieron) así como su solución.

6.1.Implementación del Modelo-Vista-Controlador en PHP

Problema: inexperiencia en la implementación de un Modelo-Vista-Controlador y desconocimiento del lenguaje PHP.

Desde un primer momento se intentó organizar el código lo mejor posible de forma que se facilitase la tarea de otro programador en caso de continuar con alguna evolución del proyecto. Por ello se pensó en el desarrollo de un Modelo-Vista-Controlador. La principal dificultad vino dada por el total desconocimiento de implementación en código PHP.

Solución: visualización y lectura de tutoriales online, los cuales sirvieron de iniciación en el lenguaje PHP y de esa forma se fue evolucionando y dividiendo el código adecuadamente.

6.2.Cambio de color de la pestaña actual

Problema: se pretendía modificar el archivo CSS mediante código Javascript pero no era posible debido al recargado de la página.

Cada vez que la página en cuestión se recargaba, el archivo CSS maestro se cargaba y predominaba sobre el código Javascript introducido, el cual se ejecutaba 'onClick'.

Solución: Después de analizar la situación se optó por crear una bandera mediante PHP la cual determina en que pestaña se encuentra el usuario y modifica el código CSS.

6.3.Implementación del carrusel

Problema: desconocimiento de cómo implementar un carrusel dinámico mediante código Javascript. En un primer momento se optó por la búsqueda de código similar en internet pero su funcionamiento no fue satisfactorio.

Solución: Se analizó el código de un carrusel maestro y se modificó según las especificaciones del requisito.

7. CONCLUSIONES

En este apartado se exponen las conclusiones extraídas después de la realización del proyecto. Se dividen en conclusiones sobre las herramientas utilizadas, conclusiones sobre la planificación y conclusiones personales.

7.1.Conclusiones sobre las herramientas utilizadas

En un primer momento se comenzó la implementación utilizando el software 'Adobe Dreamweaver' y 'Adobe Fireworks'. Poco después se tomó la decisión de utilizar 'Sublime Text' y 'Adobe Photoshop'. Está fue una elección clave que hizo posible la finalización del proyecto debido a la experiencia ya adquirida en el manejo del editor fotográfico y a los muchas y útiles funcionalidades de 'Sublime'.

En cuanto al IDE 'Wamp Server', facilitó enormemente el manejo de la base de datos local desde el principio, evitando retrasos en este aspecto.

En caso de volver a desarrollar una aplicación similar a esta, volvería a escoger estas herramientas.

7.2.Conclusiones sobre la planificación y gestión del proyecto

Desde el inicio de este proyecto se tenía claro el modelo de desarrollo que mejor se ajustaba al mismo y sus características, era un desarrollo iterativo e incremental en el que por periodos cortos de tiempo se realizara una serie de tareas.

Aun así se ha demostrado lo complicado que es cumplir una planificación con una fecha de inicio y fin establecidos de antemano, ya que siempre suceden imprevistos que afectan al correcto desarrollo del proyecto.

Aunque sucedieron imprevistos por los que la fecha de finalización del proyecto se ha visto afectada, la estructura de *Sprints* marcada en el inicio del proyecto se ha llevado de una forma más o menos acertada cumpliendo con los objetivos en el tiempo estimado del proyecto.

7.3.Conclusiones personales

Desde que se me presentó la oportunidad de desarrollar esta aplicación el sentimiento más arraigado era el de euforia. Jamás se me había planteado una oportunidad de trabajar en una idea con tanta utilidad y tan inmediata. En un primer momento la implementación y el desconocimiento del lenguaje PHP y Javascript dificultó en gran medida el avance general pero una vez visualizados múltiples tutoriales al respecto, no hubo más problemas y poco a poco la aplicación fue tomando forma.

Desde un punto de vista completamente personal, jamás participe en el desarrollo de una aplicación que pudiese ayudar tanto al usuario que la utiliza y que podría además, salvar vidas. Es por tanto para mí un auténtico orgullo haber participado en este proyecto.

8. MEJORAS Y POSIBLES AMPLIACIONES

Debido al acotado tiempo de desarrollo permitido para la implementación de un Proyecto Fin de Grado y sobre todo al destinatario de esta aplicación, las mejoras son muy variadas.

- Funcionamiento total en un dispositivo tipo 'Tablet'.
- Inclusión de múltiples avisos.
- Conexión con una base de datos remota.
- Inclusión de seguridad.
- Inclusión de múltiples funcionalidades como por ejemplo:
 - Visualización diferente de aquellas ubicaciones que no se puedan introducir.
 - Modificación de la información mostrada en 'Estadísticas'.
 - Modificación de ubicaciones ya introducidas.
 - Eliminación de ubicaciones.
 - Administración total del sistema.

9. BIBLIOGRAFÍA

Libros de referencia:

- Juan Diego Gauchat. El gran libro de HTML5, CSS3 y Javascript.
- Álvarez García, Alonso, de las Heras del Dedo, Rafael (2012). *Métodos Ágiles y Scrum (Manuales Imprescindibles)*. Anaya
- Somerville, Ian (2005). *Ingeniería del Software*. Ed. Pearson Addison-Wesley

Documentación técnica:

- Api de JQuery:
< <https://api.jquery.com/> >
- Foro StackOverflow
< es.stackoverflow.com/ >
- Portal de tutoriales y foros W3Schools
< www.w3schools.com/ >
- Curso sobre implementación en PHP
< <http://juandarodriguez.es/cursosf14/unidad2.html> >