



# **ESCUELA SUPERIOR DE INGENIERÍA**

## **INGENIERÍA INFORMÁTICA**

### **GESTIÓN DE CENTRO DE MEJORA DEL RENDIMIENTO Y LA SALUD**

Jesús Soriano Candón

22 de septiembre de 2015





## INGENIERÍA INFORMÁTICA

### GESTIÓN DE CENTRO DE MEJORA DEL RENDIMIENTO Y LA SALUD

- Departamento: Ingeniería Informática
- Director del proyecto: Lorena Gutiérrez Madroñal
- Autor del proyecto: Jesús Soriano Candón

Cádiz, 22 de septiembre de 2015

Fdo: Jesús Soriano Candón



## ***Agradecimientos***

*Introduzca aquí, si lo desea, los agradecimientos.*

## Resumen

Introduzca aquí un resumen no superior a 500 palabras, que servirá de descripción pública del trabajo realizado.

**Palabras clave:** Lista de palabras clave que reflejen el contenido del trabajo en aras de facilitar su búsqueda en sistemas bibliográficos.

# Índice general

<b>I</b>	<b>Prolegómeno</b>	<b>1</b>
<b>1.</b>	<b>Introducción</b>	<b>5</b>
1.1.	Motivación . . . . .	5
1.2.	Alcance . . . . .	5
1.3.	Glosario de Términos . . . . .	6
1.4.	Organización del documento . . . . .	6
<b>2.</b>	<b>Planificación</b>	<b>7</b>
2.1.	Metodología de desarrollo . . . . .	7
2.1.1.	Primer Ciclo . . . . .	7
2.1.2.	Segundo Ciclo . . . . .	7
2.1.3.	Tercer Ciclo . . . . .	8
2.1.4.	Cuarto Ciclo . . . . .	8
2.1.5.	Quinto Ciclo: Pruebas . . . . .	8
2.2.	Planificación del proyecto . . . . .	8
2.3.	Organización . . . . .	8
2.4.	Costes . . . . .	9
2.5.	Riesgos . . . . .	9
<b>II</b>	<b>Desarrollo</b>	<b>11</b>
<b>3.</b>	<b>Requisitos del Sistema</b>	<b>15</b>
3.1.	Situación actual . . . . .	15
3.1.1.	Procesos de Negocio . . . . .	15
3.1.2.	Entorno Tecnológico . . . . .	15
3.1.3.	Fortalezas y Debilidades . . . . .	15
3.2.	Necesidades de Negocio . . . . .	15
3.2.1.	Objetivos de Negocio . . . . .	16
3.2.2.	Procesos de Negocio . . . . .	16
3.3.	Objetivos del Sistema . . . . .	16
3.4.	Catálogo de Requisitos . . . . .	16
3.4.1.	Requisitos funcionales . . . . .	16
3.4.2.	Requisitos no funcionales . . . . .	16
3.4.3.	Reglas de negocio . . . . .	16
3.4.4.	Requisitos de información . . . . .	16
3.5.	Alternativas de Solución . . . . .	16

3.6. Solución Propuesta . . . . . 17



# Índice de figuras



# Índice de cuadros



Parte I

Prolegómeno



La primera parte de la memoria del PFC debe contener una introducción y una planificación del proyecto.

La introducción es un capítulo que, a modo de resumen, debe contener una breve descripción del contexto de la disciplina en la que el proyecto tiene aplicación y la motivación para su desarrollo, así como del alcance previsto.

El segundo capítulo debe incluir una planificación del proyecto. La planificación deberá ajustarse a las prácticas de ingeniería en general, y de la ingeniería del software en particular. Deberá tener en cuenta los plazos, los entregables (documentos y software), los recursos (humanos y de equipamiento inventariable) y el método de ingeniería de software a emplear.





# Capítulo 1

## Introducción

### 1.1. Motivación

*CoreSport*<sup>1</sup> es un centro de mejora del rendimiento y la salud que ofrece, entre otros servicios, clases dirigidas de entrenamiento funcional, TRX, nutrición, fisioterapia... y otros tantos que se irán viendo a lo largo de la memoria del presente Proyecto Fin de Carrera (PFC en adelante). Este proyecto consiste en el desarrollo de una aplicación web que permita la gestión de las actividades del centro, así como de los usuarios del mismo. En concreto, se desarrollará la página web de la empresa, con acceso a área de clientes, donde cada usuario tendrá acceso a la gestión de actividades, así como los administradores a una gestión más amplia sobre actividades y usuarios.

Actualmente no existe en dicha organización ningún proceso telemático para realizar este tipo de gestión, por lo que todos los datos de actividades y citas quedan registrados en papel. Esto produce mayor esfuerzo para la gestión y el mantenimiento de la información, así como trabajo extra en la comunicación del usuario con el centro para la gestión de sus actividades o citas, ya sea vía telefónica o personalmente en el mismo centro.

Por lo tanto, con el desarrollo del sistema, se ofrecerá una herramienta para ambas partes, administradores y usuarios, que mejorará y facilitará la forma que hasta ahora han tenido para comunicarse y gestionar sus peticiones.

### 1.2. Alcance

La aplicación resultante se utilizará vía online por los administradores y usuarios del centro de salud y rendimiento *CoreSport*, situado en Chiclana de la Frontera. Por lo que será accesible desde cualquier dispositivo con acceso a Internet.

No obstante, aunque el proyecto se centre en los requisitos de esta empresa, se tendrá en cuenta la posibilidad de que otros centros similares hagan uso del software. Por tanto, aspectos claves como las actividades ofrecidas, interfaz de usuario o logotipo de la empresa serán fácilmente adaptables a nuevos posibles centros interesados en el uso de la aplicación.

---

<sup>1</sup>Para más información acerca de CoreSport, visita su página web [www.coresport.es](http://www.coresport.es)

### 1.3. Glosario de Términos

- Entrenamiento funcional: Este tipo de entrenamiento se centra en sesiones cortas, dinámicas, efectivas y entretenidas. Entre otras propiedades, podemos destacar la mejora de movilidad general, tanto articular como muscular, el gran gasto calórico que conlleva o la mejora de habilidades motrices: agilidad, coordinación y equilibrio.
- TRX (Entrenamiento en suspensión): Se considera *entrenamiento en suspensión* a los ejercicios funcionales que se desarrollan a través de un arnés sujeto por un punto de anclaje, ajustable no elástico fabricado de distintos materiales que permite realizar un entrenamiento completo para todo el cuerpo utilizando el propio peso corporal y la resistencia a la gravedad.
- Bulgarian Bag (Saco Búlgaro): equipamiento de ejercicio en forma de luna creciente usado en entrenamiento de fuerza, pliometría, entrenamiento con pesas, ejercicio aeróbico, y fitness en general.

### 1.4. Organización del documento

El presente documento se divide en tres partes bien diferenciadas:

- **Prolegómeno:** Esta parte contiene una introducción al proyecto, en la cual se explica al lector en qué consistirá este de una forma general junto al contexto donde será usado, además de la planificación del mismo.
- La segunda sería la parte de **desarrollo**, donde se especifican los requisitos, análisis, diseño, construcción y pruebas del sistema. Es decir, se explica en detalle el proceso de desarrollo del proyecto, desde su planteamiento hasta las pruebas realizadas una vez finalizado, incluyendo toda la ingeniería del software. Es la parte más técnica de la documentación.
- **Epílogo:** Es la última parte del documento, donde encontraremos principalmente el manual de usuario, bibliografía e información sobre la licencia de la documentación y el software.
- **Software:** El producto final se divide en dos partes:
  - La **página web** pública de la organización con la que se trabaja, de acceso libre.
  - La **aplicación web** mediante la cual los administradores y usuarios tendrán la opción de realizar su gestión. Esta se podrá acceder desde la web pública, con la diferencia que se necesitará llevar a cabo un registro para su uso. Sería la parte principal del proyecto.

## Capítulo 2

# Planificación

### 2.1. Metodología de desarrollo

Previamente al desarrollo de la aplicación web, se ha llevado a cabo el de la web pública de la empresa. Para ello, se ha realizado un diseño inicial de forma orientativa para su posterior desarrollo y pruebas de funcionamiento en diferentes dispositivos. Una vez finalizada, y al poder ser usada independientemente, se ha puesto en producción mientras se implementaba el área de cliente, la parte principal en la que se centrará el proyecto.

Para el desarrollo de la mencionada aplicación web se ha usado un modelo incremental e iterativo. Primeramente se realizó un análisis de la aplicación en general y herramientas a utilizar para su desarrollo. A partir de ahí, se inicia el proceso de implementación del producto dividido en varios ciclos, en los cuales se han ido añadiendo distintas funciones a la aplicación, obteniendo así una versión más completa al final de cada una de las fases. En cada una de ellas se analiza, diseña, implementa y prueba estas funcionalidades a añadir.

A continuación, se describirá las tareas realizadas en cada uno de estos ciclos:

#### 2.1.1. Primer Ciclo

Primero de todo, se ha estructurado la implementación del proyecto en distintos módulos. Una vez hecho esto, se ha configurado el servidor de aplicaciones y definido los distintos tipos de usuarios del sistema, implementando seguidamente el registro e identificación de usuarios, con manejo de sesiones y de errores.

A su vez, se ha realizado el diseño general de la interfaz del software, que se ha aplicado a las pantallas de esta fase del desarrollo, como registro de usuario, inicio de sesión o la página de inicio una vez realizado el login, teniendo en cuenta que existen diferentes vistas de la interfaz dependiendo del tipo de usuario.

Esta será multilenguaje, dando opción al usuario a elegir su lenguaje por defecto seleccionándolo en el menú desplegable para tal efecto.

#### 2.1.2. Segundo Ciclo

En este segundo ciclo se ha implementado todo lo relacionado con la gestión de usuarios: vista y edición del perfil de los usuarios registrados, gestión de los mismos por parte de administradores,

gestión de administradores por el super-administrador, comunicación entre todo tipo de usuario, histórico de acciones en el sistema, etc.

### **2.1.3. Tercer Ciclo**

Este ciclo comprende la principal funcionalidad del sistema, la gestión de actividades y citas. Se ha desarrollado la creación de actividades, grupos, citas, etc., junto a consulta de los mismos, edición, altas, bajas... Siempre desde una interfaz intuitiva que incluye un calendario donde ver toda la actividad del usuario.

Es la parte más compleja e interesante del producto, ya que cumple el principal requisito funcional por el cual se ha realizado este proyecto.

### **2.1.4. Cuarto Ciclo**

En este último ciclo de implementación se han añadido las funcionalidades restantes para la finalización del producto, como notificaciones, contacto o contenido de la página de inicio, por ejemplo.

### **2.1.5. Quinto Ciclo: Pruebas**

Finalmente, se han realizado las pruebas pertinentes del software y se ha procedido a subsanar los errores encontrados y llevar a cabo pequeñas mejoras. Aclarar que en cada ciclo se han realizado pruebas manuales de la parte correspondiente, por lo que esta fase final de pruebas se ha realizado sin mucha incidencia.

## **2.2. Planificación del proyecto**

*Estimación temporal y definición del calendario básico (hitos principales e iteraciones). Desarrollo de la planificación detallada, utilizando un diagrama de Gantt. Los diagramas de Gantt que se vean correctamente (girados y divididos si hace falta).*

*Se debe incluir una comparación cuantitativa del tiempo y el esfuerzo realmente invertido frente al estimado y planificado. Estos datos pueden recogerse del sistema de gestión de tareas empleado para el seguimiento del proyecto.*

## **2.3. Organización**

Respecto al desarrollo de la aplicación, he sido el único desarrollador, a la vez de testeador. La tutora del proyecto ha sido Lorena Gutiérrez Madroñal, guiándome en su proceso y asegurándose que cumplía los requisitos suficientes para que sea un proyecto completo.

En cuanto al cliente, los tres socios de *CoreSport* con los que he mantenido la comunicación durante el proceso de desarrollo, y posterior a este, han sido Ángel Soriano, Cristina Saucedo y Raúl Coca. Ellos serán los administradores del sistema, con opción de añadir algunos de los trabajadores restantes en el centro, como monitores de las clases, responsables de servicios externos o recepcionista.

El resto de usuarios del software serán los clientes de la empresa, que serán los usuarios finales del producto mediante previo registro.

El hardware utilizado para el desarrollo ha sido el propio ordenador portátil del alumno, un MacBook Pro, sirviendo así de entorno de programación y pruebas mediante el uso del siguiente software:

- **OS X Yosemite** (versión 10.10.3) como sistema operativo.
- **Glassfish** como servidor de aplicaciones local.
- **NetBeans** 8.0.2 como entorno de desarrollo.
- **pgAdmin3** como gestor y administrador de bases de datos, usando **PostgreSQL** como sistema de gestión.
- **Subversion** como sistema de control de versiones.
- **TeXShop** como editor de textos para la documentación en **LaTeX**.

## 2.4. Costes

**TODO:** Esta sección se realizará una vez finalizado el proyecto, para comprobar el tiempo empleado y hacer el coste acorde al mismo.

*Estudio y presupuesto de los costes de los recursos (humanos y materiales) descritos anteriormente, necesarios para el proyecto.*

*Para el cálculo de costes de personal pueden consultarse las tablas salariales de la UCA para el personal técnico de apoyo contratado laboral [?], o bien otras más ajustadas a la realidad. El cálculo del coste del personal del proyecto debe hacerse en personas-mes, y luego hacer la correspondencia al coste monetario.*

## 2.5. Riesgos

Primeramente, tendremos en cuenta el riesgo del hardware. No sabemos cuál va a ser la vida de nuestro equipo en el que estamos desarrollando una aplicación, por lo que siempre debemos tener una copia de nuestro código para evitar posibles pérdidas de datos, ya sea por avería o rotura. Para ello, se ha utilizado *Subversion*, un sistema de control de versiones que puede ser usado desde nuestro entorno de programación *Netbeans*, brindando una herramienta esencial y sencilla de usar. Tendremos así nuestra implementación en un lugar seguro, además de beneficiarnos de las opciones que un control de versiones ofrece.

Uno de los posibles riesgos que no podremos controlar es el cambio de requisitos durante el desarrollo del proyecto. Es común que los clientes cambien o añadan funcionalidades al sistema cuando van viendo algunos resultados, probando prototipos o simplemente puntos que se les haya pasado anteriormente.

Estos cambios conllevarán un tiempo extra en el desarrollo del producto. De aquí la importancia de la fase de elicitación de requisitos y la comunicación con el cliente, para tratar de obtener unos requisitos bien definidos desde el principio y evitar así posibles cambios o nuevas funciones

a añadir.

Otro riesgo potencial en la implementación del software es la actualización de las versiones de la tecnología que estemos usando. Es decir, cuando usamos librerías, frameworks, APIs, etc., estamos expuestos a que estos se actualicen, o incluso dejen de tener soporte. Al ser, de nuevo, un factor externo al desarrollo, no se podrán tomar grandes medidas de prevención, pero sí se puede mejorar con una buena decisión respecto qué tecnologías usar en nuestro desarrollo.

En el desarrollo del software, la inmensa mayoría de las veces existirán diversas formas de conseguir un objetivo. De aquí que tengamos la opción de decidir qué lenguaje de programación, librerías, frameworks, APIs... usar. Aún así, si nos vemos en el hecho de tener que realizar cambios en alguna de ellos durante o después del desarrollo, el impacto no debería ser muy grande con una implementación limpia y bien construida.

Un ejemplo clásico de riesgo podría ser la estimación del tiempo de desarrollo del proyecto. Es común que este sea menor al tiempo real de desarrollo y, como consecuencia, existe un incremento de recursos: el tiempo de entrega del proyecto, gastos extras -ya sea para el cliente o el equipo de desarrollo- o pérdida de beneficios para negocios que no obtienen su software a tiempo. En este caso, un retraso en la estimación del tiempo no traería grandes consecuencias, los usuarios y administradores del centro seguirán llevando la misma rutina de gestión hasta que el producto esté listo para pasar a fase de producción.

Una vez el producto esté en producción, su rendimiento dependerá de un servidor contratado, por lo que es un riesgo externo a tener en cuenta. Si el servidor bajo el que la aplicación esté funcionando falla, no se tendrá acceso a la misma. Este es un riesgo que no podemos controlar, y se confía en que la empresa encargada del mismo tome medidas de seguridad suficientes para que, en el caso de fallo, el servicio no sufra caída alguna.

# Parte II

## Desarrollo









## Capítulo 3

# Requisitos del Sistema

En esta sección se detalla la situación actual de la organización y las necesidades de la misma, que originan el desarrollo o mejora de un sistema informático. Luego se presentan los objetivos y el catálogo de requisitos del nuevo sistema. Finalmente se describen las diferentes alternativas tecnológicas y el análisis de la brecha entre los requisitos planteados y la solución base seleccionada, si aplica.

### 3.1. Situación actual

Esta sección debe contener información sobre la situación actual de la organización para la que se va a desarrollar el sistema software.

#### 3.1.1. Procesos de Negocio

Esta sección debe contener información sobre los modelos de procesos de negocio actuales, que suelen ser la base de los modelos de procesos de negocio a implantar.

#### 3.1.2. Entorno Tecnológico

Esta sección debe contener información general sobre el entorno tecnológico en la organización del cliente antes del comienzo del desarrollo del sistema software, incluyendo hardware, redes, software, etc.

#### 3.1.3. Fortalezas y Debilidades

Esta sección debe contener información sobre los aspectos positivos y negativos del negocio actual de la organización para la que se va a desarrollar el sistema software.

### 3.2. Necesidades de Negocio

Esta sección debe contener información sobre los objetivos de negocio de clientes y usuarios, incluyendo los modelos de procesos de negocio a implantar.

### **3.2.1. Objetivos de Negocio**

Esta sección debe contener los objetivos de negocio que se esperan alcanzar cuando el sistema software a desarrollar esté en producción.

### **3.2.2. Procesos de Negocio**

Esta sección, debe contener los modelos de procesos de negocio a implantar, que normalmente son los modelos de procesos de negocio actuales con ciertas mejoras.

## **3.3. Objetivos del Sistema**

Esta sección debe contener la especificación de los objetivos o requisitos generales del sistema.

## **3.4. Catálogo de Requisitos**

Esta sección debe contener la descripción del conjunto de requisitos específicos del sistema a desarrollar para satisfacer las necesidades de negocio del cliente.

### **3.4.1. Requisitos funcionales**

Descripción completa de la funcionalidad que ofrece el sistema.

### **3.4.2. Requisitos no funcionales**

Descripción de otros requisitos (relacionados con la calidad del software) que el sistema deberá satisfacer: portabilidad, seguridad, estándares de obligado cumplimiento, accesibilidad, usabilidad, etc.

### **3.4.3. Reglas de negocio**

En el desarrollo del sistema, hay que tener en cuenta las denominadas reglas de negocio, es decir, el conjunto de restricciones, normas o políticas de la organización que deben ser respetadas por el sistema, las cuales suelen ser cambiantes.

### **3.4.4. Requisitos de información**

En esta sección se describen los requisitos de gestión de información (datos) que el sistema debe gestionar.

## **3.5. Alternativas de Solución**

En esta sección, se debe ofrecer un estudio del arte de las diferentes alternativas tecnológicas que permitan satisfacer los requerimientos del sistema, para luego seleccionar (si procede) la herramienta o conjunto de herramientas que utilizaremos como base para el software a desarrollar.

### **3.6. Solución Propuesta**

Si se ha optado por utilizar un software de base, debemos identificar y medir las diferencias entre lo que proporciona este software y los requisitos definidos para el proyecto.

El resultado de este análisis permitirá identificar cuáles de éstos requisitos ya están solventados total o parcialmente por el sistema base y cuales tendremos que diseñar e implementar la propuesta de solución.

