

## **Propuesta Proyecto Final de Carrera 2010/2011**

## **PFC-1**

Los abajo firmantes solicitamos la aprobación de la propuesta del proyecto fin de carrera que se describe en la memoria adjunta y que lleva por título:

*Desarrollo de un Sistema de Gestión Online de Consultas Médicas Particulares*

Así mismo, proponemos como tutores del citado a proyecto a:

Tutor: Borja Rubio Reyes

Las Palmas de Gran Canaria, a 17 de Junio de 2011

Fdo.: *Borja Rubio Reyes*

Ingeniero Informática. Profesor de la ULPGC  
del área de conocimiento de Ciencias de la Computación  
e Inteligencia Artificial.

Fdo.: *Pablo Eduardo Ojeda Vasco*

Alumno de la Facultad Informática U.L.P.G.C.

**SR. PRESIDENTE DE LA COMISIÓN DE PROYECTOS FIN DE CARRERA DE LA ESCUELA DE INFORMÁTICA DE LA ULPGC**

**ANEXO PFC-1**

**Memoria de Propuesta de Proyecto Fin de Carrera de la Escuela de Informática de la ULPGC**

Título: *Desarrollo de un Sistema de Gestión Online de Consultas Médicas Particulares*

Tipo de proyecto: *Individual*

Personas que realizan la propuesta:

- *Borja Rubio Reyes, DNI: 78496422-K, Profesor de departamento de Informática y Sistemas de la ULPGC*
- *Pablo Eduardo Ojeda Vasco, DNI: 12387725-V, Alumno Facultad Informática ULPGC*

**Descripción General del Proyecto**

El dominio de la aplicación es el de las consultas médicas privadas, donde normalmente existen **dos actores principales**: el médico y el paciente.

En el sector de la sanidad privada, la gestión de centros médicos (desde el seguimiento de un paciente hasta la organización interna del mismo) se ha desarrollado **tradicionalmente en papel** (información analógica). Sin embargo, con el avance tecnológico y con la aparición de las nuevas tecnologías de información y comunicación (TICs), se crean nuevos modelos de gestión y administración de los centros, pasando todos los flujos de datos a versión digital. Es decir, la información comienza a gestionarse usando software específico, pero **centralizado en una localización física concreta**. A pesar de todo, sigue habiendo muchos especialistas que no han dado el paso y continúan utilizando métodos arcaicos.

Por otro lado, **el proceso de concertar una cita** sigue siendo el mismo que antaño: un paciente debe presentarse en la consulta del médico físicamente y solicitar cita ó bien debe realizar una llamada telefónica para ser atendido por un administrativo.

Por tanto, con el objetivo de poder gestionar digitalmente y desde internet todos los aspectos referentes al flujo de información de una consulta médica, se requiere desarrollar una aplicación web desde la que se gestionen un gran número de consultas privadas, con todos los datos referentes a médicos (precios, horarios de disponibilidad, etc.), a pacientes (ficha médica, diagnósticos, pruebas, exploraciones, etc.), los pagos y además, permita que sean los propios especialistas los que muestren a los pacientes de que horas libres disponen, para que sean estos últimos los encargados de asignarse, desde cualquier ubicación en la que exista un computador con internet, la cita que más les convenga.

Para el análisis de requisitos se realizarán entrevistas con médicos que trabajen en consultas médicas particulares. Finalmente se realizará una validación del prototipo final en una situación de aplicación real en algún centro médico.

## Objetivos

Se propone diseñar e implementar una aplicación SaaS (Software as a Service), que lo que pretende es ofertar un **software como servicio** a todos aquellos especialistas sanitarios interesados.

Por tanto, se desarrollará una aplicación web, en la que se abordaran los siguientes módulos:

- **Gestión de médicos.** *Funcionalidades que pueden desarrollar los distintos especialistas para gestionar su consulta de la forma más eficiente posible. Tienen que ver principalmente con su calendario, sus plantillas, sus pacientes y con diversos datos de configuración.*
- **Gestión de pacientes.** *Funcionalidades que pueden desarrollar los usuarios con el rol de paciente. Tienen que ver principalmente con sus citas, sus médicos, con diversos datos de configuración y con la información de sus fichas médicas.*
- **Gestión de fichas médicas.** *Ficha médica de los pacientes, con sus pruebas médicas, diagnósticos, tratamientos, informes, información asociada, etcétera.*
- **Panel de administrador.** *El sistema será gestionado y mantenido por un administrador, el cuál podrá realizar diversas tareas.*
- **Gestión de actividades generales.** *Una serie de funcionalidades que podrán realizar usuarios sin necesidad de estar identificados, tales como darse de alta en la aplicación, ver las condiciones generales, un tour de la aplicación, buscar si un médico forma parte del sistema, etcétera.*

## Metodología

El paradigma del ciclo de vida del software del que se hará uso es el del **desarrollo incremental**, en el cuál se tienen claros los requisitos de usuario y los requisitos de software, pero las fases de diseño detallado, transferencia, operación y mantenimiento se dividen en unidades más manejables. Así, se desarrollan múltiples versiones, cada una de ellas con mayor funcionalidad y capacidad que las anteriores.

El motivo de la elección es que se tienen claros los requisitos desde el comienzo, que es apropiado para proyectos de larga duración con equipos de desarrollo pequeños como es el caso. Se implementaran en primer lugar ciertas funcionalidades críticas y en las sucesivas iteraciones se añadirán nuevas funcionalidades.

Se utilizará un **método de desarrollo** actual, el **orientado a objetos** (entidades que encapsulan datos y procesos). El software se organiza a partir de los elementos que existen en el dominio del problema. La orientación a objetos permite por su propia naturaleza la consecución de diseños modulares efectivos.

En **el análisis**, los requisitos del software se modelan en términos de clases, objetos, atributos y operaciones como componentes principales.

En **el diseño**, se crea un modelo del mundo real que puede realizarse en software, modularizando conjuntamente la información y el procesamiento.

También se realizará el **diseño de la interfaz**, teniendo en cuenta la interfaz entre el hombre y el ordenador, y entre los módulos del software. Y por supuesto el **diseño de componentes** para convertir el diseño de datos, interfaces y arquitectura en una representación intermedia que se pueda transformar fácilmente en código fuente. Dicho código será generado en la fase de **implementación**.

Además se realizarán todo tipo de **pruebas**. Pruebas de unidad, de integración, de validación y del sistema. Por último, considerará realizar **mantenimiento** durante un breve periodo de tiempo.

## Medios necesarios para la elaboración del Proyecto

Para la realización del proyecto debemos disponer de un equipamiento informático básico: computador, impresora, etc... Además, debemos tener acceso a otras herramientas básicas e imprescindibles para la consecución del proyecto: editor de texto, acceso a internet, etc...

Será necesario un servidor de aplicaciones web, en el que correrán un servidor Apache, servidor de bases de datos y framework rails para ruby.

Para la realización del software se va a hacer uso de "ruby on rails". Se intentará emplear en la medida de lo posible software libre.

## Plan de trabajo

### Fase 1. Análisis

- ➡ Actividad 1.1.- Documentación y herramientas
  - Realización encuestas y entrevistas
  - Adquisición de información
  - Estudio de proyectos similares
  - Estudio de software de escritorio similar
  - Búsqueda en internet de software similar
  - Generación de documentación sobre herramientas
- ➡ Actividad 1.2.- Análisis de requisitos de usuario
  - Elaboración de la lista de características
  - Análisis de viabilidad de requerimientos de usuario
  - Generación de documentación de análisis de requisitos de usuario
- ➡ Actividad 1.3.- Análisis de requisitos de software
  - Análisis de requerimientos de software
  - Análisis de viabilidad de requerimientos de software

- Generación documentación análisis de requisitos de software
- ➡ Actividad 1.4.- Verificación de requisitos

## Fase 2. Diseño

- ➡ Actividad 2.1.- Diseño del contenido
  - Diseño del contenido general
  - Diseño del contenido de los médicos
  - Diseño del contenido de los pacientes
  - Diseño del contenido de las fichas médicas
  - Diseño del contenido del administrador
  - Generación documentación del contenido
- ➡ Actividad 2.2.- Diseño de la Arquitectura del sistema
  - Diseño de la arquitectura del sistema
  - Generación documentación de la arquitectura del sistema
- ➡ Actividad 2.3.- Diseño de navegación
  - Diseño de navegación
  - Generación documentación de Diseño de navegación
- ➡ Actividad 2.4.- Diseño de componentes
  - Diseño de módulo de gestión de médicos
  - Diseño de módulo de gestión de pacientes
  - Diseño de módulo de gestión de administrador
  - Diseño de módulo de gestión de actividades generales
  - Diseño de módulo de gestión de fichas médicas
  - Generación documentación de Diseño de componentes
- ➡ Actividad 2.5.- Diseño de interfaz de usuario
  - Diseño de interfaz del médico
  - Diseño de interfaz del paciente
  - Diseño de interfaz del administrador
  - Diseño de interfaz de actividades generales
  - Diseño de interfaz de fichas médicas
  - Generación documentación de Diseño de interfaz de usuario
- ➡ Actividad 2.6.- Diseño de la estética (gráfico)
  - Diseño gráfico "welcome"
  - Diseño gráfico médico
  - Diseño gráfico paciente
  - Diseño gráfico administrador
  - diseño gráfico fichas médicas
  - Generación documentación de Diseño gráfico

## Fase 3. Implementación y pruebas de código

- ➡ Actividad 3.1.- Implementación de módulos de bases de datos
  - Implementación de base de datos
  - Generación documentación de Implementación de base de datos
- ➡ Actividad 3.2.- Implementación de módulos de aplicación
  - Implementación de módulo de gestión de médicos
  - Implementación de modulo de gestión de pacientes
  - Implementación de módulo de gestión de administrador
  - Implementación de módulo de gestión de fichas médicas

- Implementación de módulo de gestión de actividades generales
  - Generación documentación de implementación de módulos de aplicación local
- ➡ Actividad 3.3.- Diseño gráfico
- Maquetación
  - Generación documentación de Diseño gráfico

#### Fase 4. Validación y Verificación

- ➡ Actividad 4.1.- Tests de validación
- Definición de los test de validación
  - Aplicación de los test de validación
  - Análisis de resultados de los test de validación
  - Generación documentación test de validación
- ➡ Actividad 4.2.- Validación de usuario final
- Entrevista para Validación del cliente
  - Fase de test de Validación del cliente
  - Análisis de resultados de los test de Validación del cliente
  - Generación documentación de Validación de usuario final.

#### Fase 5. Instalación y pruebas en el servidor

- ➡ Actividad 5.1.- Integración en el servidor
- ➡ Actividad 5.2.- Pruebas de rendimiento, recuperación, etc.

Temporización del Proyecto de Fin de Carrera									
Fases/Actividades	Meses								horas
	1	2	3	4	5	6	7	8	
Fase 1. Análisis	■	■	■	■	■	■	■	■	270
1.1.- Documentación y herramientas	■	■	■						80
1.2.- Análisis de requisitos de usuario		■	■						70
1.3.- Análisis de requisitos de software			■						80
1.4.- Verificación de requisitos			■						40
Fase 2. Diseño	■	■	■	■	■	■	■	■	210
2.1.- Diseño de contenido			■						30
2.2.- Diseño de la arquitectura			■						20

Temporización del Proyecto de Fin de Carrera																			
Fases/Actividades	Meses																		Horas
	1		2		3		4		5		6		7		8				
2.3.- Diseño de componentes																		80	
2.4.- Diseño de navegación																		10	
2.5.- Diseño de interfaces																		20	
2.6.- Diseño gráfico																		50	
Fase 3. Implementación																		340	
3.1.- Implementación de módulos de B. de datos																		20	
3.2.- Implementación de módulos de aplicación																		240	
3.3.- Maquetación																		80	
Fase 4. Pruebas, Validación y Verificación																		150	
4.1.- Tests de validación																		120	
4.2.- Validación de usuario final																		30	
Fase 5. Instalación y pruebas en el servidor																		25	
5.1.- Integración en el servidor																		5	
5.2.-Pruebas de rendimiento, etc.																		20	
Total de horas																995			