

## Escuela Superior de Ingeniería

### **PROYECTO FIN DE CARRERA**

**Estudio Dietético Programado 2011**

# Estudio Dietético Programado 2011

Manuel González Pérez, <sup>(1)</sup>Manuel Palomo Duarte,  
<sup>(2)</sup>Iván Ruiz Rube

*Avda Cayetano del Toro 16, Cádiz, España*  
*manuel.gonzalezperez@alum.uca.es*

*<sup>(1)(2)</sup>C/Chile 1, CP 11003, Cádiz, España*

Este proyecto pretende servir de ayuda a todas aquellas empresas dedicadas a la nutrición que necesiten un software que les permita llevar el seguimiento diario de los pacientes. Por otro lado pretende contribuir en el mundo del Software Libre.

**Palabras Clave:** Python, PyQt, EDP11, software libre, gestión de dietas

## 1. Introducción

Este Proyecto Fin de Carrera (PFC) tiene como objetivos: aplicar los conocimientos adquiridos durante la titulación Ingeniería Técnica en Informática de Sistemas, aprender y utilizar nuevas herramientas que enriquecen los conocimientos que no se han podido adquirir durante la titulación y aportar una contribución al mundo del Software Libre y al sector de la nutrición a la que esta orientado dicho proyecto.

El proyecto consiste en la creación de un software que permita gestionar los hábitos alimenticios de los pacientes, así como la posibilidad de gestionar una pequeña o mediana empresa de nutricionistas, como son las franquicias de nutrición más conocidas, albergando el numeroso personal de nutricionistas que tienen al frente.

## **2. Planificación**

La duración del proyecto ha sido de aproximadamente 404 días, comprendido entre el 18 de Mayo del 2011 y el 3 de Diciembre del 2012. Durante este periodo se han desarrollado las siguientes etapas:

1. **Fase de Inicio (18/05/11 - 15/06/11):** Durante esta etapa se plantea la idea y se consulta con los tutores si es adecuada y la disponibilidad.
2. **Especificación de requisitos (16/06/11 – 29/07/11):** La toma de requisitos fue una labor compleja, debido a los constantes cambios y añadidos de funcionalidades.  
  
Finalmente se determinaron con éxito todas las funcionalidades y exigencias para el proyecto.
3. **Análisis (01/08/11 – 12/10/11):** Teniendo los requisitos bien definidos el análisis ha sido más claro aunque también duradero.
4. **Diseño (13/10/11 – 30/01/12):** Así como la implementación, ambas han sido las etapas que más tiempo han consumido. Esta etapa se ha hecho con especial cuidado de realizar un diseño correcto para no interrumpir la implementación por posibles errores.
5. **Implementación (30/01/12 - 12/07/12):** Dicha etapa ha sido la más larga durante el desarrollo, en la que se han implementado todos los requisitos satisfaciendo las necesidades previstas.
6. **Pruebas (13/07/12 – 09/08/12):** Se han probado todas las funcionalidades comprobando su correcto funcionamiento y ejecución, asegurándose que no hay ningún tipo de error.

7. **Fase de Documentación (10/08/12 - 03/12/12):** En esta etapa se redactó este documento y todos los necesarios para el complemento del proyecto.

### **3. Descripción general del proyecto**

Este PFC tiene la condición de Software Libre, por lo que cualquier persona podrá continuarlo en el futuro, ya sea el propio autor o cualquier persona ajena al mismo.

#### **3.1 Interfaces software**

La aplicación utiliza Python[1,2,3] como lenguaje de programación, usando la adaptación de la librería gráfica Qt para Python, llamado PyQt[4]. Se trabaja directamente sobre el sistema operativo.

Para el diseño de la interfaz de usuario se utiliza la herramienta QtDesigner, y para el almacenamiento de los datos se utiliza SQLite3 como Sistema Gestor de Base de Datos.

#### **3.2 Interfaces de usuario**

La aplicación esta compuesta por una ventana principal desde la cual se podrá tener acceso a todas las funcionalidades de la misma. Cuenta con una barra de menú agrupando las funcionalidades por categorías, así como botones de acceso directos.

### **4. Análisis**

Se ha optado por una metodología basada en la 'Rational Unified Process', usando la notación 'UML' [11](Lenguaje Unificado de Modelado) para toda la documentación, ya que es la metodología estándar más usada para el análisis, diseño y documentación de sistemas orientados a objetos.

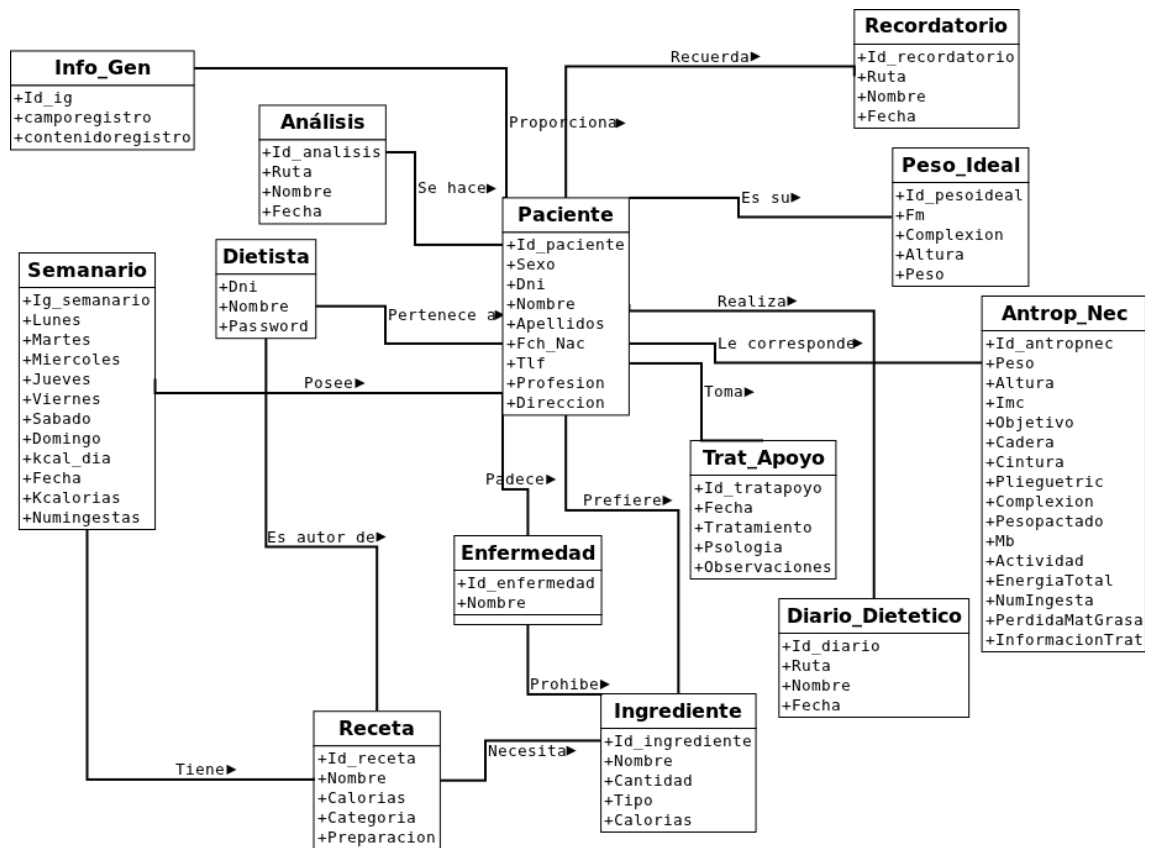
#### **4.1 Requisitos funcionales**

Tras realizar el estudio de las necesidades, a continuación se exponen los requisitos funcionales de la aplicación:

1. **Gestión de dietistas:** agregar nuevo, abrir, cerrar y eliminar un perfil de dietista.
2. **Gestión de pacientes:** agregar nuevo, abrir, editar, cerrar, eliminar, añadir y modificar analíticas, tratamiento farmacológico o enfermedades/patologías, , añadir información general, añadir y eliminar diarios dietéticos, añadir y eliminar recordatorios 24h., añadir la frecuencia de ingesta de alimentos, añadir recetas al semanario y ver semanarios existentes.
3. **Gestión de recetas:** añadir, editar y eliminar recetas.

#### **4.2 Modelo conceptual de datos**

A continuación se presenta el diagrama de clases conceptuales en la **Figura 1**.



*Figura 1. Diagrama Conceptual de Datos*

Las restricciones de clave primaria son: (Dietista, d\_dni), (Paciente, p\_id), (Semanario, s\_id), (Enfermedad, ep\_id), (Receta, r\_id), (Ingrediente, i\_id)

## 5. Diseño

Para llevar a cabo el diseño de la aplicación, se ha optado por un patrón arquitectónico de tres capas:

1. **Capa de presentación:** se comunica con el usuario.
2. **Capa de dominio:** implementa las funcionalidades del sistema.
3. **Capa de gestión de datos:** interactúa con la base de datos.

## 5.1 Diseño de la base de datos

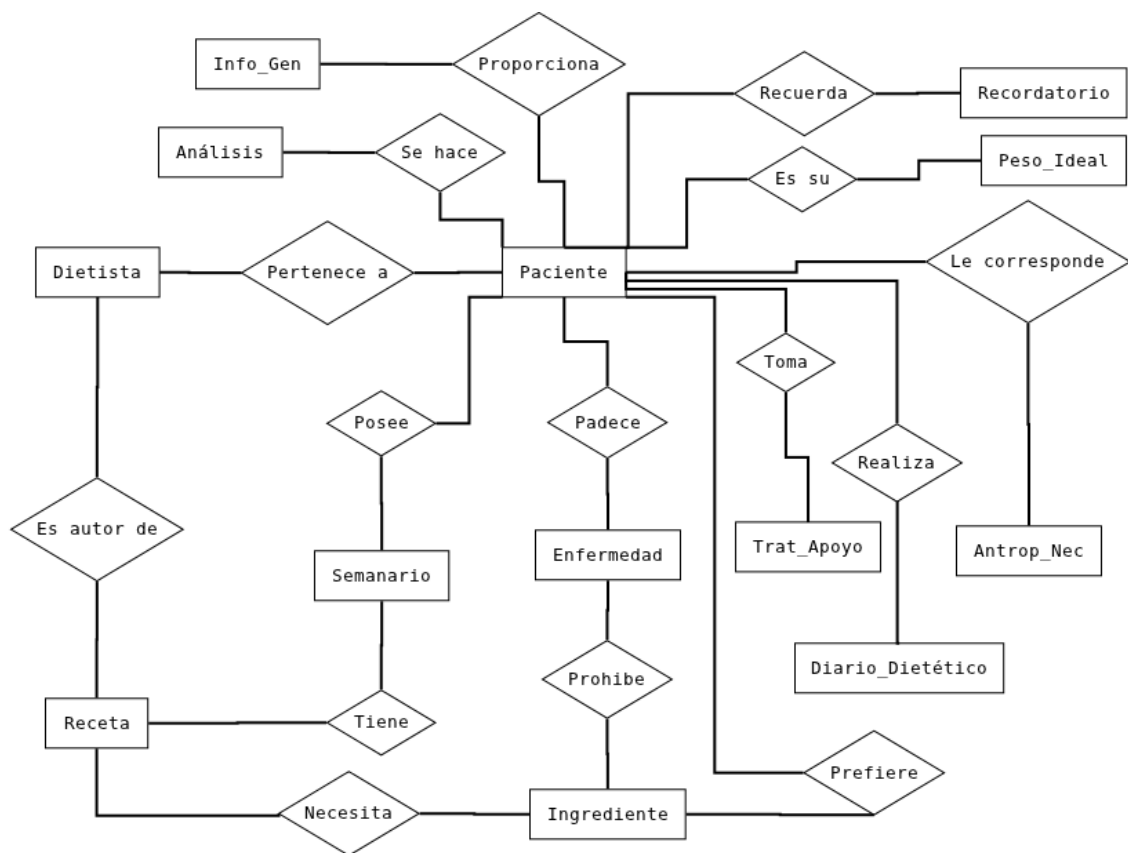
Una vez realizado todo el estudio lógico de la base de datos y llevado a cabo la normalización, las tablas resultantes son las reflejadas en la **Tabla 1**. Se reflejan subrayadas las claves primarias y en cursiva las foráneas.

**Tabla 1.** *Tablas que componen la base de datos*

Entidad	Atributos
Dietista	<u>Dni</u> , Nombre, Password
Paciente	<u>Id_paciente</u> , Sexo, Dni, Nombre, Apellidos, Fch_Nac, Tlf, Profesion, Direccion, <i>Dni_Diet</i>
Antrop_Nec	<u>Id_antropnec</u> , Peso, Altura, Imc, Objetivo, Cadera, Cintura, Plieguetric, Complexion, Pesopactado, Mb, Actividad, EnergiaTotal, NumIngesta, PerdidaMatGrasa, InformacionTrat, <i>Id_paciente</i>
Analisis	<u>Id_analisis</u> , Ruta, Nombre, Fecha, <i>Id_paciente</i>
Diario_Dietetico	<u>Id_diario</u> , Ruta, Nombre, Fecha, <i>Id_paciente</i>
Recordatorio	<u>Id_recordatorio</u> , Ruta, Nombre, Fecha, <i>Id_paciente</i>
Info_Gen	<u>Id_ig</u> , camporegistro, contenidoregistro, <i>Id_paciente</i>
Trat_Apoyo	<u>Id_tratapoyo</u> , Fecha, Tratamiento, Psologia, Observaciones, <i>Id_paciente</i>
Peso_Ideal	<u>Id_pesoideal</u> , Fm, Complexion, Altura, Peso, <i>Id_paciente</i>
Receta	<u>Id_receta</u> , Nombre, Calorias, Categoria, Preparacion,
Ingrediente	<u>Id_ingrediente</u> , Nombre, Cantidad, Tipo, Calorias
Receta_Ingredientes	<u>Id_recing</u> , NombreRec, NombreIngred, CantIngred, CalIngred, <i>Dni_Diet</i>
Enfermedad	<u>Id_enfermedad</u> , Nombre
Enf_Ingred	<u>Id_enfingred</u> , <i>Id_enfermedad</i> , <i>Id_ingrediente</i>
Enf_Paciente	<u>Id_enfpac</u> , <i>Id_paciente</i> , <i>Id_enfermedad</i>

Semanario	<u>Id_semanario</u> , Lunes, Martes, Miercoles, Jueve Viernes, Sabado, Domingo, kcal_dia, Fecha, Kcalorias, Numingestas, <u>Id_paciente</u>
-----------	---

Para mayor comprensión del diagrama se omiten los atributos proporcionados en la tabla de entidades **Figura 2**.



**Figura 2.** Diagrama Entidad-Relación

## 5.2 Diseño de la capa de presentación

En esta sección se van a mostrar algunas de las ventanas pertenecientes a la aplicación.



**Estudio Dietético Programado 2011**

Archivo Paciente Recetas Ayuda

años      Peso Actual:     Objetivo:

Datos Personales    Estudio Antropométrico    Historia Dietética    Registro Dietético

Sexo:

Nombre:       Apellidos:

Fch. Nacimiento:       Edad:  años

D.N.I.:       Dirección:

Teléfono:       Profesión:

**Figura 3.** Menú principal de la aplicación

**Abrir Perfil Dietista**

Dietista	D.N.I.
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

**Figura 4.** Ventana Abrir Perfil Dietista

Estudio Dietético Programado 2011 -

Archivo Paciente Recetas Ayuda

años

Peso Actual: Objetivo: !!

Datos Personales Estudio Antropométrico Historia Dietética Registro Dietético

Nº Ingestas: 6 Ver Recetas

Kcal. / %		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
104.021	Desayuno							
312.063	Media Mañana							
624.126	Almuerzo							
312.063	Merienda							
624.126	Cena							
104.021	Tentempié							

Kcal./día:

Recetas Calorías

- ▶ Desayuno
- Media Mañana
- ▶ Almuerzo
- Merienda
- ▶ Cena
- ▶ Tentempié

**Figura 5. Ventana Registro Dietético**

Nuevo Paciente

Sexo: F

Nombre:  Apellidos:

Fch. Nacimiento: 03/10/1950 Edad:

D.N.I.:  Teléfono:

Profesión:

Dirección:

Cancel OK

**Figura 6. Formulario Nuevo Paciente**



*Figura 7. Ventana Elección Modificar Receta*

## 6. Implementación

Para la implementación del código fuente se ha optado por la siguiente distribución de directorios:

1. **usr/share/edp11**: directorio principal donde se encuentran los archivos principales “.py” de la aplicación. Cada archivo representa una clase implicada en la aplicación.
2. **edp11/PY\_UIs**: directorio en el que se encuentran todas las transformaciones de los archivos “.ui” en “.py” de la interfaz gráfica, necesarios para la ejecución de la aplicación. Cada uno de ellos representa una pantalla o formulario de la aplicación.
3. **edp11/Docu**: directorio en el que se encuentran las plantillas “.odt” necesarias para la generación de los archivos PDFs de la aplicación, así como PDFs

pertenecientes a la aplicación siendo estos documentos del dietista que debe proporcionar.

## **6.1 Herramientas usadas**

Para poder llevar a cabo toda la implementación de la aplicación se han utilizado las siguiente herramientas:

1. Emacs: Editor de texto muy potente, usado para generar los archivos “.py” (Python[1,2,3])
2. Conjunto de librerías para las funcionalidades necesarias: PyQt[4] (adaptación de Qt para Python), Relatorio[9] (para generar informes), Poppler[10] para generar archivos PDFs desde la aplicación.
3. Qt4 Designer : diseñador de interfaces gráficas para Qt. Con esta herramienta generamos los archivos “.ui”, que representan las pantalla y formularios de la aplicación.
4. Git : usado para el control de versiones.

Todas las herramientas usadas se encuentran bajo la licencia GNU General Public License v.3.

## **6.2. Problemas encontrados durante la implementación**

A la hora de implementar la aplicación surgieron diversos problemas, causados principalmente por el desconocimiento de las herramientas usadas, así como del lenguaje de programación Python(absolutamente algo nuevo para mí). En el caso de Python, existe numerosa cantidad de documentación, tanto en su web oficial como en foros y blogs, por lo que acostumbrarse fue cuestión rápida.

En el caso de PyQt, el aprendizaje resultó más difícil, debido a su escasa documentación. Hubo que recurrir a la documentación de Qt así como a foros para solventar dudas y problemas.

Hubo que declarar nuevos tipos de elementos para un comportamiento de los objetos acorde con lo requerido.

Otro caso es la biblioteca Relatorio, que para operar con ella había que determinar como formatear los datos de entrada, lo cual fue laborioso.

Sin hablar de la biblioteca Poppler, para la que la utilización en Python debía seguir una forma, y no fue encontrada en ninguna documentación ni foro ni blog, finalmente tomando como referencia la poca documentación oficial, y haciendo un estudio lógico de la utilización se llegó con éxito.

## **7. Pruebas**

Para comprobar el buen funcionamiento de la aplicación, se han realizado pruebas de caja negra tanto a los módulos u objetos independientes como a los subsistemas que componen la aplicación. Del mismo modo se han realizado pruebas de caja blanca, recorriendo todas las posibles opciones para ver que todo funciona correctamente.

Igualmente, para comprobar que la interfaz tiene un comportamiento adecuado, se ha comprobado los siguientes aspectos: limitar el tipo de campo, limitar la longitud de los datos introducidos y comprobar la existencia de campos vacíos no permitidos.

Todas estas pruebas se han realizado tanto en la fase de desarrollo como una vez terminada la aplicación por completo.

Las pruebas se realizaron sobre el sistema operativo Ubuntu 11.04 y Ubuntu 12.04 Linux, en sus distribuciones Ubuntu 10.10, sin ningún tipo de problema.

## 8. Conclusiones

Previa realización del proyecto se propusieron los objetivos de aprender sobre todo, no sólo en conocimientos, sino a nivel personal, aprender a afrontar y solucionar retos mayores, demostrando la capacidad de ser autosuficientes y valernos por si mismos.

Dada la oportunidad del proyecto, uno de los objetivos principales fue también ayudar; aplicar los conocimiento y capacidades para facilitar el trabajo de otro a través de la informática. Cuestión principal de estudiar esta carrera.

Pese a que durante la titulación se han adquirido multitud de conocimientos relevantes de la informática, no se ha puesto a prueba la situación de valernos por nosotros mismos ante proyectos completos y reales.

Es por ello que para el proyecto se ha empleado el lenguaje de programación *Python*, la biblioteca gráfica *Qt*[5], la biblioteca *Relatorio*, *Git* para el repositorio, así como el IDE generador de interfaces gráficas *Qt4 Designer*, nada de esto visto con anterioridad, ni en la titulación.

Otras herramientas como *LATEX*, para la generación de la memoria y los manuales; *DIA* para la realización de todos los diagramas de la parte de análisis y diseño de la memoria, así como el programa de edición de imágenes *Gimp*, con el que se editaron las capturas de pantalla contenidas en la memoria.

En un futuro, la aplicación se pretende ampliar, añadiendo estadísticas de seguimiento al paciente, paquetes de idiomas, un sistema de colores en base a las preferencias, así como una posterior integración con sistemas móviles y web.

## 9. Bibliografía

1. Página oficial de *Python* . <http://www.python.org>
2. Tutorial de *Python* “*Python para todos*”. <http://mundogeek.net/tutorial-python/>
3. Pilgrim, Mark. *Dive into Python*. Apress, 2004.

4. Página oficial de la biblioteca *Qt* para *Python*.  
<http://www.riverbankcomputing.co.uk/software/pyqt/intro>
5. Página oficial del proyecto *Qt*. <http://qt-project.org/doc/qt-4.8/>
6. Summerfield. *Rapid gui programming with python and qt*. Prentice hall, 2007.
7. Guía de creación de paquetes *Python* en *Ubuntu*.  
<https://wiki.ubuntu.com/PackagingGuide/Python>
8. Página oficial de *GIT*. <http://git-scm.com/>
9. Página oficial de la biblioteca *Relatorio* para *Python*.  
<http://relatorio.openhex.org/wiki/QuickExample>
10. Página oficial de *Poppler* para *Qt* y *Python*. <https://code.google.com/p/python-poppler-qt4/>
11. Larman, Craig. *Applying UML and Patterns*, 3a Edición. Prentice Hall, 2004.

## **10. Agradecimientos**

Agradecimientos a mis familiares por su apoyo, a mi novia por su paciencia y aguante, y a mi director de proyecto Manuel Palomo y codirector Ivan Ruiz, por su dedicación, aceptación e interés que han puesto.