

# **ÍNDICE:**

- Objetivos
- Descripción del proyecto
- Plan de proyecto
  - Selección metodología de Ingeniería del Software
  - Planificación del proyecto
- Especificación de requisitos
  - Análisis de negocio
  - Plan de gestión y especificación de requisitos
- Análisis y diseño técnico
  - Arquitectura global
  - Diferentes tipos de diagramas
  - Diseño interfaz de usuario
- Conclusiones del proyecto



#### **OBJETIVOS:**

#### Académicos

- Poner en práctica las competencias adquiridas en los estudios de la Ingeniería Técnica en Informática de Sistemas
- Aplicar, reforzar y asentar esp<mark>ecialme</mark>nte los contenidos de la rama de Ingeniería del Software

### Profesionales

- Desarrollar una aplicación que ayude a una asociación de familiares de enfermos de Alzheimer a gestionar a los usuarios, los recursos y todas las necesidades presentes y futuras de la asociación.



### **DESCRPICIÓN DEL PROYECTO:**

El proyecto define una aplicación como nexo de unión entre familiares, pacientes y equipo de la asociación; sin olvidar la función principal de gestión de recursos terapéuticos para los pacientes, así como el registro de datos de uso para posterior estudio de la evolución de cada paciente.

Mejoras que pretende conseguir el proyecto con la aplicación:

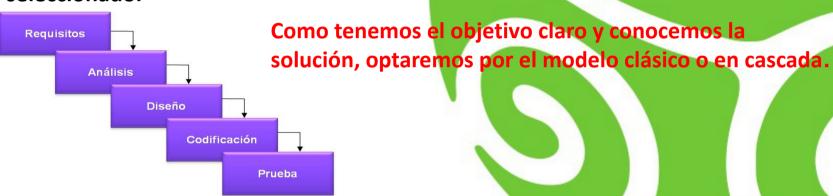
- Facilitar el acceso a los recursos por parte de los usuarios pacientes, de una forma ágil y sencilla.
- Conectar entre sí los diferentes tipos de usuarios y facilitar la captación de datos de uso de los recursos para su posterior análisis.
- Dinamizar y optimizar la aplicación, además de ofrecer información a cerca de todas las actividades a realizar por las diferentes áreas de la asociación.



#### **PLAN DEL PROYECTO**

El proyecto define una aplicación que posibilita la gestión, ayuda y control de enfermos de Alzheimer; facilitándoles herramientas para estimular la memoria a través de recursos y generando datos para que el equipo de la asociación pueda hacer un seguimiento y estudio de la evolución de los pacientes.

- 1. Selección de proceso de Ingeniería de Software
- 2. Selección de la metodología de trabajo dentro del proceso que hemos seleccionado.



3. Realización de la planificación del proyecto, aplicando la metodología seleccionada.



El proyecto se ha dividido en 4 fases para trazar un plan de trabajo, planificado y estimado por tiempos para progresar en el desarrollo del trabajo.

**FASE 1.** Solución y propuesta del proyecto

Tareas PEC1	Comienzo	Finalización
Selección y propuesta del proyecto	16/09/15	23/09/15
Validación del proyecto	24/09/15	24/09/15
Planificación del proyecto	25/09/15	28/09/15
Elaboración de la documentación PEC1	28/09/15	29/09/15
Entrega de la documentación	30/09/15	30/09/15

**FASE 2.** Requisitos y especificaciones

Tareas PEC2	Comienzo	Finalización
Toma de requisitos	30/09/15	05/10/15
Identificación de actores	05/10/15	08/10/15
Identificación de guiones	08/10/15	12/10/15
Análisis de casos de uso	12/10/15	18/10/15
Definición del diagrama E/R	18/10/15	25/10/15
Requisitos funcionales de la interface	25/10/15	03/11/15
Elaboración de la documentación PEC2	05/10/15	03/11/15
Entrega de la documentación	03/11/15	04/11/15

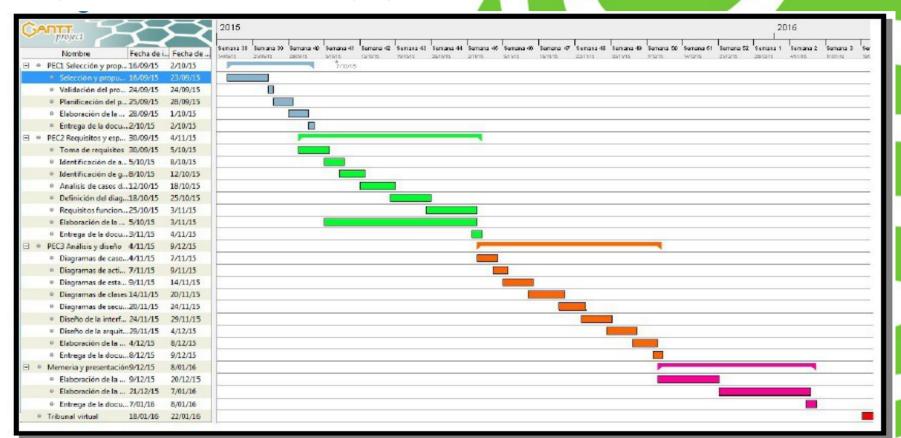
FASE 3. Análisis y diseño del proyecto

Tareas PEC3	Comienzo	Finalización
Diagramas de casos de uso	04/11/15	07/11/15
Diagramas de actividad	07/11/15	09/11/15
Diagramas de estados	09/11/15	14/11/15
Diagramas de clases	14/11/15	20/11/15
Diagramas de secuencia	20/11/15	24/11/15
Diseño de la interfaz de usuario	24/11/15	29/11/15
Diseño de la arquitectura	29/11/15	04/12/15
Elaboración de la documentación PEC3	04/12/15	08/12/15
Entrega de la documentación	08/12/15	09/12/15

**FASE 4.** Memoria y presentación del proyecto

### 4. Planificación del proyecto

Utilizamos diagramas de Gantt para la representación gráfica de la temporización de las tareas de un proyecto.



Hitos de la evaluación continua:

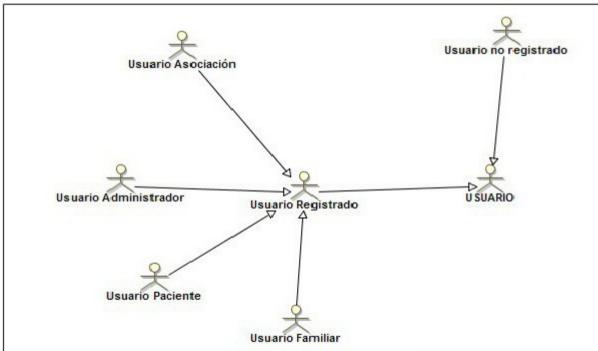
- **PEC1** (30/09/15) : Plan de trabajo
- PEC2 (04/11/15) : Especificación de requisitos
- PEC3 (09/12/15) : Diseño técnico
- Memoria y presentación (08/01/16)



# **ESPECIFICACIÓN DE REQUISITOS**

- Análisis de negocio; debemos comprender bien las necesidades de la Asociación sobre la que vamos a definir una solución para su aplicación web. Identificar carencias y necesidades actuales, definir características claves para dar un servicio que cubran esas carencias y necesidades.
- Plan de gestión y especificación de requisitos; los requisitos expresan las necesidades y restricciones que afectan a nuestra apliación y nos ayudan a delimitar las soluciones mas adecuadas para los problemas, los divideremos en requisitos funcionales y no funcionales. También debemos identificar a los actores que van a hacer uso de nuestra aplicación para ayudarnos a gestionar los casos de uso.

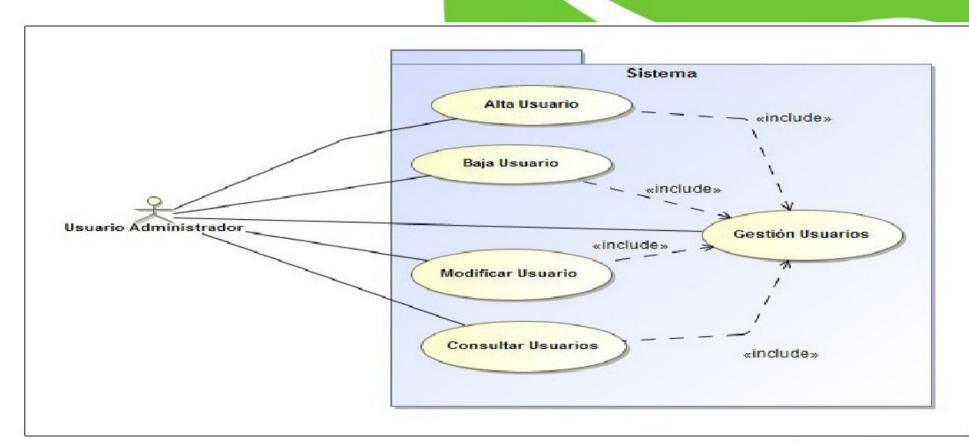




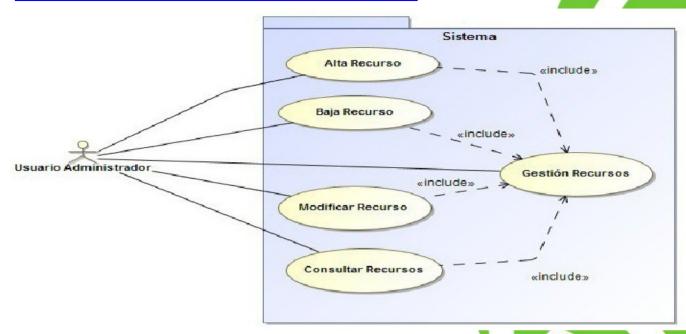
### Análisis de requisitos

La aplicación la vamos a dividir en cuatro subsistemas, otorgando a cada uno de ellos unas funciones diferentes. Los casos de uso se dividirán en las diferentes áreas conformadas por los diferentes subsistemas.

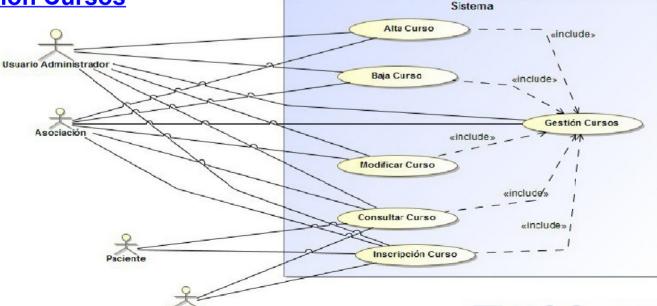
## **Subsistema de Mantenimiento**



# Casos de uso de Gestión Recursos







Realizado por: Julio David Hernández Domenech

9 UOC Universitat Oberta de Catalunya

# Subsistema de Conexión Sistema Usuario Administrador Incio Sesión Cierre Sesión Recordar Contraseña Usuario Registrado Usuario Familiar Cambian contraseña Consultar Datos **Modificar Datos** Subsistema de Análisis Sistema Alta Análisis «include» «indude» Baja Anális is Gestión Analisis Usuario Registrado «include» Consultar Análisis Usuario Asociación

Modificar Análisis

«include»

## **ANÁLISIS Y DISEÑO TÉCNICO**

Para esta aplicación hemos utilizado un patrón de arquitectura de capas, arquitectura cliente-servidor en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño.



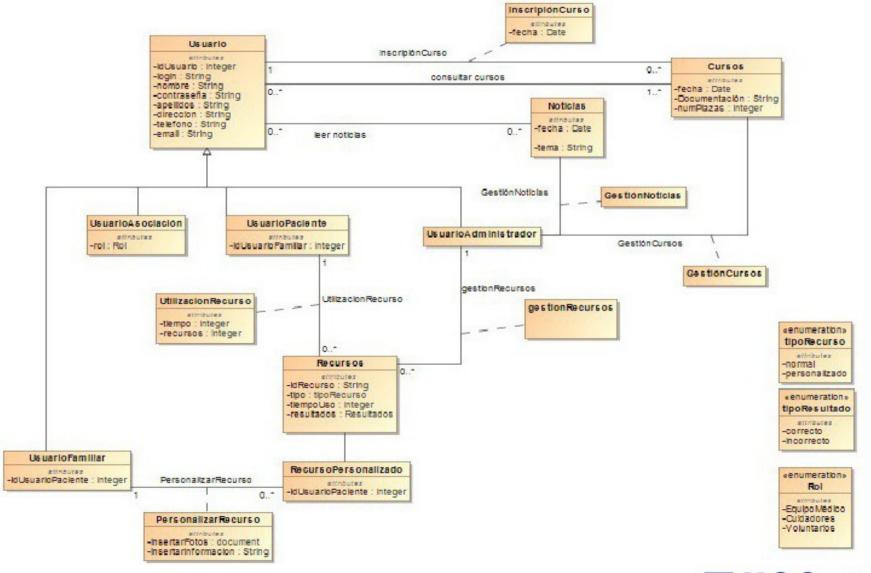
## Tarjetas CRC (Clase, responsabilidad y colaboración)

Classe	Usuario
Descripción	Define las propiedades comunes de todos los usuarios de la aplicación
Superclases	-
Subclases	UsuarioAsociación, UsuarioFamiliar, UsuarioPaciente y UsuarioAdministrador
Responsabilidades	Modela un objeto Usuario con sus datos básicos
Constructores	Usuario (idUsuario: Integer, login: String, nombre: String, apellidos: String, contraseña: String, dirección: String, teléfono: String, email: String)
Atributos	- idUsuario: Integer - login: String - nombre: String - apellidos: String - contraseña: String - dirección: String - teléfono: String - email: String
Métodos	+ getIdUsuario(): Integer + getUogin(): String + getMombre(): String + setNombre(nombre: String) + getAplelidos(): String + setAplelidos(): String + setAplelidos(): Paring + setContraseña(): String + setContraseña(): String + setContraseña(): String + setDirección(dirección: String) + setDirección(dirección: String) + setTeléfono(teléfono: String) + setTeléfono(teléfono: String) + getTelefono(teléfono: String) + setTelefono(teléfono: String) + setEmail(): String
Colaboraciones	-

Clase	RecursoPersonalizado
Descripción	Hereda de Recursos añadiendo otros datos. Esta clase contiene todos los atributos y métodos propios de un RecursoPersonalizado
Superclases	Recursos
Subclases	-
Responsabilidades	Modela un objeto RecursoPersonalizado con sus datos básicos
Constructores	- idRecurso: String - tipo: tipoRecurso - tiempoUso: Integer - resultados: Resultados - idUsuarioPaciente: Integer
Atributos	- idUsuarioPaciente: Integer
Métodos	+ getIdRecurso(): String + getTipo(): TipoRecurso + setTipo(TipoRecurso: String) + getTiempoUso(): Integer + getResultados(): Resultados + setResultados(Resultados: String) + getIdUsuarioPaciente(): Integer
Colaboraciones	-

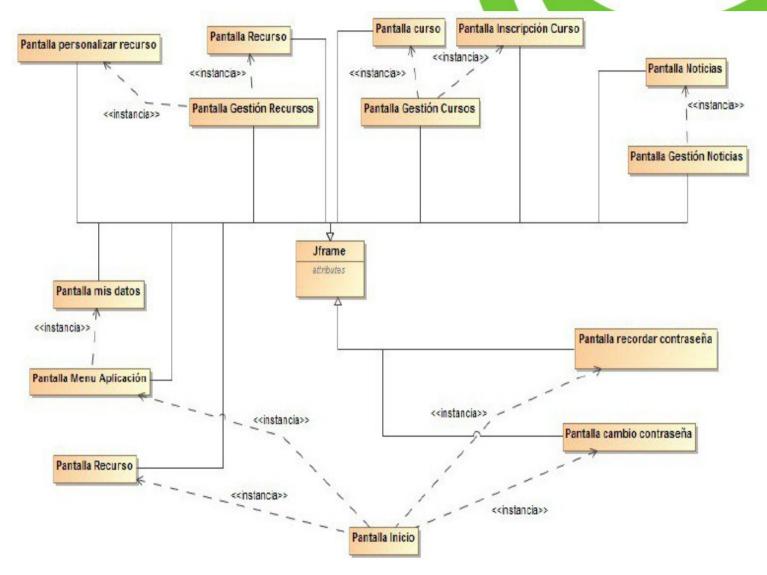
### **ANÁLISIS Y DISEÑO TÉCNICO**

Diagramas de clases (tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre los objetos)



## **ANÁLISIS Y DISEÑO TÉCNICO**

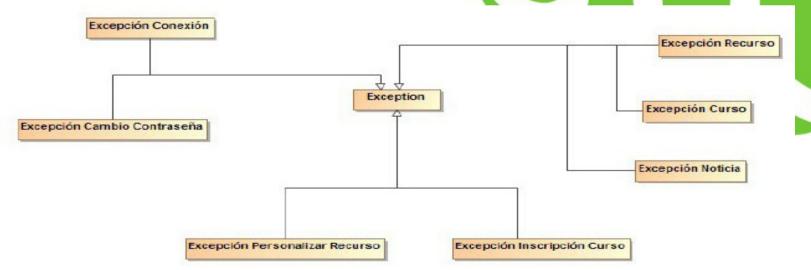
Diagramas de clases frontera (representan las distintas pantallas empleadas por el usuario para interactuar con el sistema)



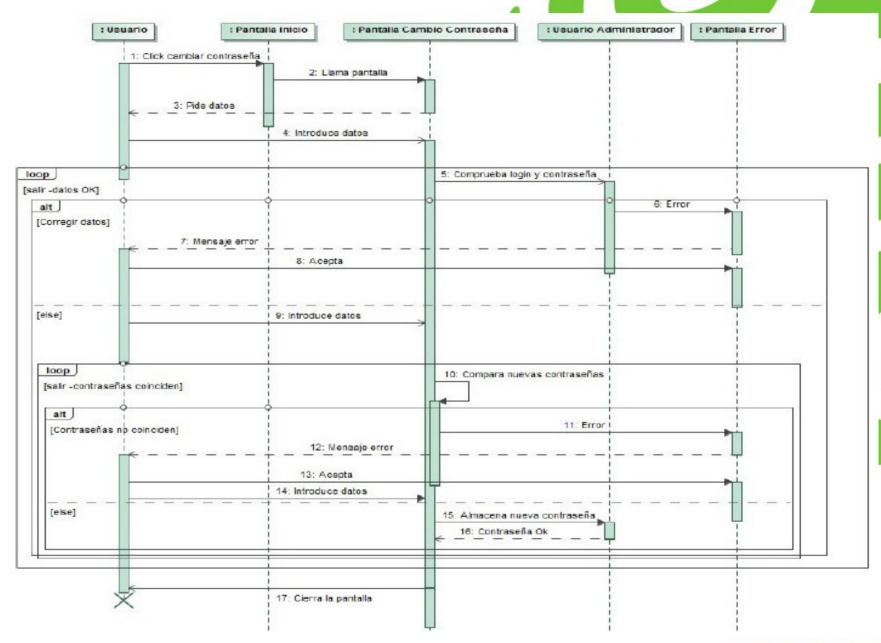
Diagramas de estados (Con estos diagramas podemos reflejar el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación en respuesta a un evento)



Diagramas de excepciones (Sirven para crear nuestros propios tipos de excepciones que permitan tratar situaciones especificas en una aplicación)



Diagramas de secuencias (tienen como objetivo mostrar la interacción entre los diversos objetos para conseguir satisfacer los requisitos establecidos)

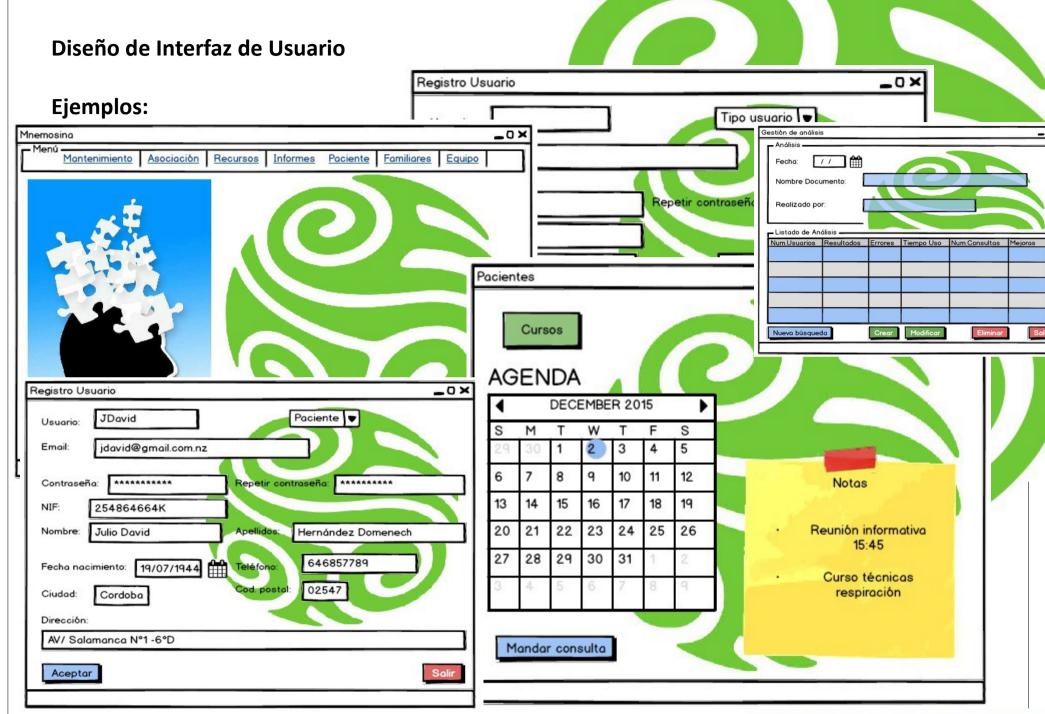


#### Diseño de Interfaz de Usuario

- Un diseño cuidadoso de la interfaz de usuario es parte fundamental del proceso de diseño general del software.
- Las interfaces gráficas deben permitir a los diferentes usuarios la interacción con la aplicación, cada usuario dispondrá de su interfaz, con características especificas en función del tipo de actor que sea.

• Seguiremos unos principios generales para desarrollar nuestras interfaces:







#### CONCLUSIONES

- Se han desarrollado los conocimientos adquiridos en Ingeniería Técnica de Informática de Sistemas
- Se han realizado las fases iniciales de ejecución de un proyecto, mediante el uso de un proceso de desarrollo y una metodología de Ingeniería del Software usuados en los ambientes profesionales reales, con un resultado satisfactorio.
- Se ha creado la semilla de un proyecto que podría continuar como una iniciativa solidaria.
- Nivel de satisfacción alto por los objetivos logrados, cosa que me ayuda para tener mayor confianza para afrontar nuevos retos tecnológicos.
- Me quedo con las ganas de poher en marcha la aplicación de manera efectiva y real, creo que un futuro si llego a disponer de más tiempo libre......
- Ha sido un gran reto para mí, ya que he tenido poco tiempo libre para poderle dedicar y he tenido que perder sueño y momentos familiares por ello; pero ha valido la pena de sobra y se lo dedico a todos los mios por su apoyo.

