



**INGENIERÍA INFORMÁTICA
CENTRO POLITÉCNICO SUPERIOR
UNIVERSIDAD DE ZARAGOZA**



Desarrollo de una aplicación para la gestión y dirección de proyectos con tecnología .NET

ANEXOS

Realizado por: JOSÉ ÁNGEL PERIBÁÑEZ GUILLÉN

Director: FERNANDO CORTÉS FRANCO
Responsable del Departamento de Desarrollo de Sistemas
Endalia

Ponente: SANTIAGO VELILLA MARCO
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

**JULIO 2010
TOMO 2/2**


TABLA DE CONTENIDOS

- Estándar de documentación
- Estándar de codificación
- Plan de gestión de configuraciones
- Estudio de mercado
- Modelo de negocio
- Especificación de requisitos
- Análisis
- Diseño
- Implementación
- Plan de pruebas
- Manual de usuario

Sistema para la gestión de proyectos

Estándar de documentación

Versión 1.1 – Fecha: 28/08/2009

	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
03/09/2007	1.0	Estándar de documentación	Alfonso Romay
28/08/2009	1.1	Adecuación de estándar de documentación de Endalia para documentación del PFC	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	4
1.1 PROPÓSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 DEFINICIONES	4
1.4 REFERENCIAS	4
1.5 RESUMEN	4
2. FORMATO DE DOCUMENTACIÓN	5
2.1 SOFTWARE UTILIZADO EN DOCUMENTACIÓN	5
2.2 MAQUETACIÓN	5
2.2.1 FORMATO DE LAS HOJAS	5
2.3 ORDEN DE CONTENIDOS	7
2.3.1 PORTADA:	7
2.3.2 REVISIONES Y COPYRIGHT:	7
2.3.3 ÍNDICE:	7
2.3.4 CONTENIDOS	7
2.4 FUENTES Y ESTILOS	8
2.5 INTERLINEADO Y FORMATO DE PÁRRAFOS	8
2.6 IMÁGENES Y DIAGRAMAS	9
2.7 GUÍA DE ESTILOS	9
3. PLANTILLAS DE DOCUMENTACIÓN	11
3.1 PLANTILLA DE DOCUMENTO	11
3.1.1 HOJA 1. PORTADA	11
3.1.2 HOJA 2. INFORMACIÓN DE DOCUMENTO	12
3.1.3 HOJA 3 Y SIGUIENTES. ÍNDICE	13
3.2 PLANTILLA DE ACTA DE REUNIÓN	14
4. BIBLIOGRAFÍA	15
4.1 REFERENCIAS	15
4.2 REFERENCIAS WEB	15

 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

El propósito del presente estándar es establecer las normas de documentación a seguir durante el desarrollo de la aplicación de dirección y gestión de proyectos.

Este documento recoge las normas de maquetación, así como la metodología de documentación a la hora de exponer los distintos aspectos que pueda abarcar la documentación software, y un conjunto de guías para que la exposición de los distintos aspectos de la documentación sea clara y estén recogidos de una forma homogénea.

1.2 Alcance del documento

El alcance del documento son todos aquellos documentos realizados durante y para el desarrollo de la aplicación de dirección y gestión de proyectos. Por supuesto, este documento ha sido creado de acuerdo a las normas que define.

1.3 Definiciones

- Fuente: Un miembro de una familia de tipo de letra.


1.4 Referencias

En este documento no se realizan referencias a otros documentos del proyecto.

1.5 Resumen

El presente documento es el estándar de documentación de Endalia. Se compone de los siguientes apartados:

- Apartado 1. Muestra el propósito del documento y se define su alcance.
- Apartado 2. Especifica el formato de documentación del proyecto, indicando el orden de sus contenidos. Incluye una guía de estilo, donde se marca una línea de estilo a seguir a la hora de redactar los documentos, y se aconseja sobre el uso de los distintos elementos de formato.
- Apartado 3: Plantillas de los principales documentos a utilizar durante la realización de la documentación del proyecto, para, en caso de ser necesarias, tenerlas a mano y modificarlas de forma rápida.
- Apartado 4. Bibliografía y referencias web utilizadas en la confección de este documento.

	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

2. FORMATO DE DOCUMENTACIÓN

2.1 Software utilizado en documentación

Para la elaboración de la documentación de este proyecto se utiliza el siguiente software:

- Microsoft Office Word Professional Edition 2007 para la edición de texto. Dado que este software no está instalado en todos los equipos de la empresa, el formato de guardado de todos los documentos, deberá ser el compatible con la versión del 2003.
- Microsoft Office Visio Professional Edition 2003 para la elaboración de diagramas.
- Adobe Photoshop CS2 para la edición de imágenes.

2.2 Maquetación

Los documentos deberán seguir las siguientes normas de maquetación:

2.2.1 Formato de las hojas

2.2.1.1 Márgenes

Las hojas utilizadas serán A4 con los márgenes que se encuentran marcados en el diagrama de la hoja que se presenta a continuación:

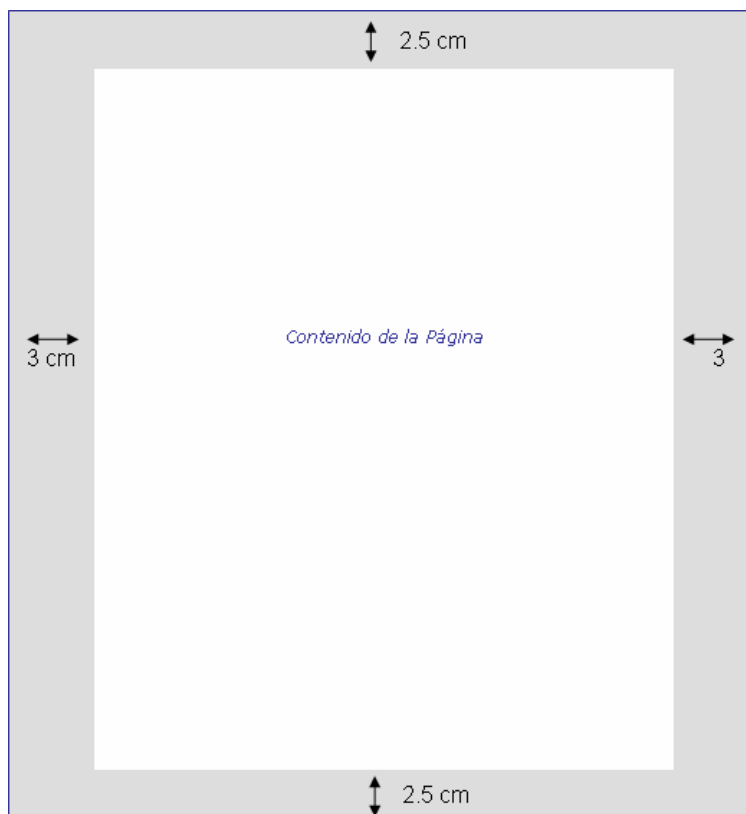



Figura 1. Diseño de los márgenes de las páginas

	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

2.2.1.2 Encabezados y pies de página

Los encabezados y pies de página ocupan respectivamente las medidas indicadas en el siguiente esquema:

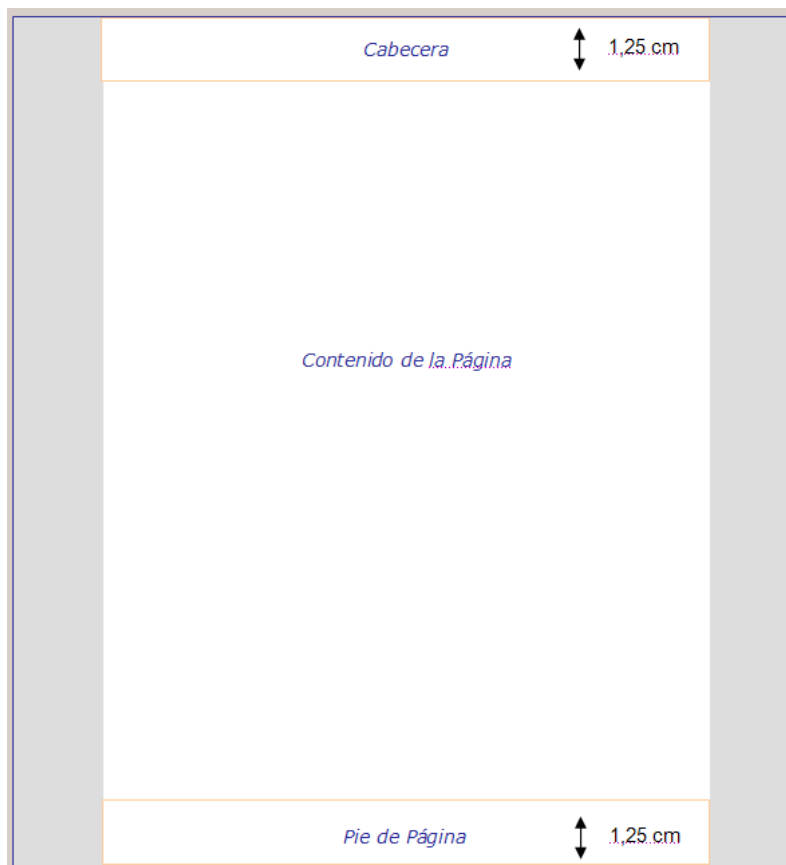



Figura 2. Diseño del encabezado y el pie de página

El logotipo del encabezado está alineado a la izquierda y sus dimensiones son de alto 1,50 cm y de ancho 3 cm, proporcional al tamaño original de la imagen.

El tipo de letra para el encabezado y el cierre de página es Arial a tamaño 8. El encabezado es una tabla de tres columnas. En la primera columna, se encuentra el logo corporativo de la empresa. En la columna central, el nombre del proyecto, título del documento, y nombre del documento. Una fila y tres columnas, en la primera celda se encuentran el nombre del documento. En la columna de la derecha, se indicará la versión del documento, y la fecha en que se modificó por última vez. Las celdas de la columna central serán alineadas a la izquierda, y las de la columna de la derecha, a la derecha.

En el pie de página se indicará a la derecha el número de página con el formato “Página 1 de 10”.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

2.3 Orden de contenidos

Un documento presenta sus contenidos en el siguiente orden:

2.3.1 Portada:

Contiene el nombre del cliente, el título del documento, su versión y la fecha del mismo. Es la única página que no presenta numeración de página, cabecera y pie de página. (Ver *Figura 4*)

2.3.2 Revisiones y copyright:

Incluye información de copyright y de control de distribución y autorización, además del control de versiones del documento. (Ver *Figura 5*)

2.3.3 Índice:

El índice se realiza a partir de una tabla de contenidos lo que significa que se debe de realizar al final del documento o ir actualizándose conforme se va agregando información. El estilo es *Sofisticado* y de tres niveles de profundidad.


El índice se desarrolla a partir de los títulos de las secciones de forma que el título de la sección más general va subrayado y los demás van disminuyendo el tamaño según la profundidad a la que se encuentren. Estos tres niveles de profundidad son correlativos con los estilos de título empleados en los documentos, es decir, que existen tres niveles de índice correspondientes a los diferentes estilos de título de los documentos, desde el de mayor importancia o *Título 1*, hasta el de menor importancia incluido dentro del índice *Título 3*.

Quedan exentos de incluirse en el índice la página de *Portada* y de *Identificación del documento*, así como el índice. No se incluyen puesto que no se consideran contenido del documento en sí, si no que son metadatos del mismo. Para que no queden incluidos dentro del mismo ha sido necesario definir otro estilo de título, del mismo nivel jerárquico que *Título 1*, pero sin numeración ni inclusión en la tabla de contenidos del índice. Este estilo está detallado en la sección de estilos de este mismo documento. (Ver *Figura 6*)

2.3.4 Contenidos

Contenidos propiamente dichos del documento en cuestión: Usan una estructura de información jerarquizada. Esta estructura está formada por cinco niveles de profundidad usando los estilos de título. En cada uno de estos niveles jerárquicos, se muestra la información respectiva. Todo el contenido usa alguno de los estilos descritos en la sección de estilos de este mismo documento. Así, nos podemos encontrar con párrafos, viñetas, imágenes, pies de foto, viñetas numeradas, etc.

Es importante hacer ver que en ningún momento se mezcla semántica con presentación. Es decir. Nunca se ponen tabulados a mano, sangrías, saltos de página ó retornos de carro. Eso sería introducir semántica extra en el contenido de la página. Todos estos detalles de presentación han sido incluidos en los estilos, definidos en la sección de estilos de este documento.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

2.4 Fuentes y estilos

A continuación, se definen los tipos de fuente utilizados para los diferentes formatos de texto del documento:

Título 1: Arial 16, Negrita.

Título 2: Arial 14, Negrita, Cursiva.

Título 3: Arial 13, Negrita.

Título 4: Trebuchet MS 10, Negrita.

Texto normal: Arial 10. Alineación justificada.

Texto en pie de imágenes: Lista numerada, de fuente Arial 10, Cursiva, centrado.

Texto de Código fuente o acciones de línea de comandos: Courier New 10.

2.5 Interlineado y formato de párrafos

El interlineado utilizado en la documentación será sencillo. El texto se justificará por ambos márgenes.


Se empezará página nueva entre apartados de primer nivel (Título 1)

Se evitará dejar títulos de segundo y tercer nivel como última línea de una página, siendo ubicados en la página siguiente.

Todos los títulos tendrán al menos 20 puntos de separación con el texto anterior.

Las relaciones de elementos se separarán mediante un salto de línea y se indicarán con un punto al principio de la línea, sin tabulado. En el caso de subrelaciones se indentarán mediante tabulados sin guión de la siguiente manera:

- Elemento 1
 - Elemento de Nivel 2
 - Elemento de Nivel 3
 - Elemento de Nivel 3
 - Elemento de Nivel 3
 - Elemento de Nivel 2
 - Elemento de Nivel 2
 - Elemento de Nivel 2
 - Elemento de Nivel 2
 - Elemento 2
 - Elemento 3

 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

2.6 Imágenes y diagramas

Las imágenes y diagramas se colocarán centrados y ajustando en lo posible su tamaño a los márgenes habituales de la página, excepto en el caso de que su tamaño sea excesivo para visualizarlos de manera óptima dentro de esos márgenes, en cuyo caso se colocaran en posición apaisada en una página nueva. Todas las imágenes estarán numeradas y se les referenciará en el texto al que acompañan y mediante una definición centrada debajo de las mismas que siempre se colocará orientada verticalmente incluso en imágenes que se hayan ubicado en el documento apaisadas de la siguiente manera (Ver *Figura 3*).



Figura 3. Ejemplo de formato de imagen

2.7 Guía de estilos

Deberá utilizarse un estilo de redacción claro y conciso, y mostrando ejemplos si eso ayuda en la comprensión. Así mismo, la información podrá estar recogida en resúmenes, históricos, gráficos u otros esquemas aclarativos, que aunque repliquen la información existente, ayuden a la comprensión de los contenidos de un documento. Siguiendo esta misma línea, se podrán incluir apéndices en los documentos con los mismos fines.

La redacción de contenidos deberá hacerse de forma sencilla, clara y concisa, sin caer en un lenguaje lleno de tecnicismos. Para ello se permitirá el uso de expresiones coloquiales utilizando comillas.


Así mismo, se permite el uso de la negrita cuando se considere necesario resaltar o dar importancia a algún texto, quedando la cursiva reservada para el marcado de ciertos elementos dentro de la maquetación como el nombrado de imágenes o referencias a documentos.

Es importante reducir el uso de tablas y/o gráficos que introduzcan nuevos estilos que no aparezcan en este documento. De esta forma, se evita caer en el uso de un número grande de estilos, con la consiguiente sensación de caos que esto produce.


Las normas y guías de estilo sobre la colocación de ciertos elementos o colores se realizan con la función de facilitar al lector la búsqueda de los elementos que precise de la mejor manera posible.

A continuación, se describe para que deben utilizarse los principales estilos y aquellas apreciaciones que puedan resultar de interés al usarlos:

- **Títulos:** su funcionalidad es mantener bien acotados los contenidos de la sección y facilitar su localización. Es importante observar que los títulos sirven para dar jerarquía al contenido de los documentos, por lo que es interesante tener la estructura del mismo clara antes de utilizarlos.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

- Enumeraciones (Viñetas): se utilizan para listar elementos, pueden ser con viñetas o numeradas. Normalmente se utilizarán viñetas y solo en caso de necesidad de establecer un orden entre los puntos se recurrirá a las numeradas.
- Marcado: se utiliza el estilo *cursiva* para referirnos a cualquier documento o sección del mismo. Siempre que se haga una referencia a una sección de un documento ajeno al actual deberá citarse también el documento al que se hace referencia.
- Espacios: El espaciado está introducido en los propios estilos. Con esto lo que se quiere decir es que no se deben introducir espacios manualmente, ya sean tabulaciones, saltos de línea, saltos de página, retornos de carro, etc. Por poner un ejemplo, el estilo *Título 1* ya incorpora un salto de página para que cada vez que se usa, se haga en una página nueva.
- Imágenes: Las imágenes deberán llevar el estilo *Imagen centrada*, y llevarán pies explicativos. En tal caso, estas explicaciones deben llevar el estilo *Pie de foto*.

	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

3. PLANTILLAS DE DOCUMENTACIÓN

Las plantillas explicadas en este apartado se encuentran guardadas en formato electrónico y por razones de espacio no se pueden mostrar aquí a tamaño real. El objetivo de este apartado es especificar el formato de los documentos para cualquiera que desarrolle algún texto para el proyecto. Por ello se muestra una captura a tamaño reducido y una explicación de las partes que las componen y la manera de utilizarlas. Como muestra del aspecto final sirve el presente documento.

3.1 *Plantilla de documento*

3.1.1 Hoja 1. Portada

El formato de la portada de todos los documentos es el siguiente.

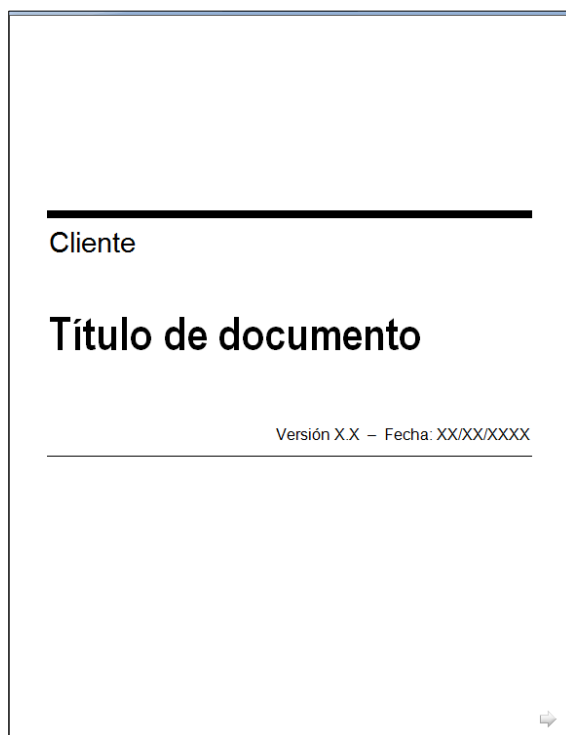



Figura 4. Hoja 1 - Portada

En él se indica el cliente con fuente Arial Narrow 20, título del documento con fuente Arial 34 negrita, y fecha y versión del mismo en fuente Arial 11.

	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

3.1.3 Hoja 3 y siguientes. Índice

El formato del índice se indica en la siguiente imagen. Se genera automáticamente con Word estableciendo el tipo de letra de los apartados de nivel 1 a Trebuchet 12 Negrita.


	Bases de empleo Web	Versión: 2.2
	Estándar de documentación	Fecha: 27/01/2009
	ESTANDARDOCUMENTACION.doc	

TABLA DE CONTENIDOS


1. INTRODUCCIÓN	4
1.1 PROPOSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 DEFINICIONES	4
1.4 REFERENCIAS	4
1.5 RESUMEN	4
2. FORMATO DE DOCUMENTACIÓN	5
2.1 SOFTWARE UTILIZADO EN DOCUMENTACIÓN	5
2.2 MAQUETACIÓN	5
2.2.1 FORMATO DE LAS HOJAS	5
2.3 ORDEN DE CONTENIDOS	7
2.3.1 PORTADA	7
2.3.2 REVISIONES Y COPYRIGHT	7
2.3.3 ÍNDICE	7
2.3.4 CONTENIDOS	7
2.4 FUENTES Y ESTILOS	8
2.5 INTERLINEADO Y FORMATO DE PÁRRAFOS	8
2.6 IMÁGENES Y DIAGRAMAS	9
2.7 GUÍA DE ESTILOS	9
3. PLANTILLAS DE DOCUMENTACIÓN	11
3.1 PLANTILLA DE DOCUMENTO	11
3.1.1 HOJA 1. PORTADA	11
3.1.2 HOJA 2. INFORMACIÓN DE DOCUMENTO	12
3.1.3 HOJA 3 Y SIGUIENTES. ÍNDICE	13
3.2 PLANTILLA DE ACTA DE REUNIÓN	14
4. BIBLIOGRAFÍA	15
4.1 REFERENCIAS	15
4.2 REFERENCIAS WEB	15

CONFIDENCIAL

© 2009 Endalia

Página 3 de 15

Figura 6. Índice

	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

3.2 Plantilla de acta de reunión

El formato de plantilla de acta de reunión es el siguiente. Incluye el nombre del proyecto, la fecha lugar y hora de la reunión, los asistentes, el orden del día y las incidencias y decisiones adoptadas.

Acta de Reunión

PROYECTO: SISTEMA WEB DE GESTIÓN DE PROYECTOS

FECHA: X de X de 2005

LUGAR: formación y Perfeccionamiento

HORA DE INICIO: XXXX

HORA DE FINALIZACIÓN: XXXX


ASISTENTES

Fernando Cortés
Diego Calvo

ORDEN DEL DÍA

INCIDENCIAS Y DECISIONES ADOPTADAS

Figura 7. Acta de Reunión

 endalia®	Sistema para la gestión de proyectos	Versión: 1.1
	Estándar de documentación	Fecha: 28/08/2009
	ESTANDARDOCUMENTACION.doc	

4. BIBLIOGRAFÍA

4.1 Referencias

[HUT] Edward J Huth *Scientific Style and Format: The CBE Manual for Authors, Editors, and Publishers* Cambridge University Press 1994

[EDE] Estándar de documentación de Endalia S.L.

4.2 Referencias Web

[Ref. Web 1] <http://www.wikipedia.org>

[Ref. Web 2] <http://www.monografias.com/trabajos6/dosi/dosi.shtml>

[Ref. Web 3] http://www.buenosaires.gov.ar/dgsinf/estandares/estandar_docu.php


[Ref. Web 4] <http://www.apa.org/journals/webref.html>

[Ref. Web 5] http://www.ucm.es/BUCM/psi/guia_red.htm

Sistema para la gestión de proyectos

Estándar de codificación

Versión 1.2 – Fecha: 18/09/2009

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

REVISIONES

17/03/2005	1.0	Realización de estándar de codificación.	Diego Calvo
21/03/2005	1.1	Revisión del documento de codificación.	Fernando Cortés
18/09/2009	1.2	Actualización del estándar para la gestión y dirección de proyectos	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España




 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	5
1.1 PROPÓSITO DEL DOCUMENTO	5
1.2 ALCANCE DEL DOCUMENTO	5
1.3 ACRÓNIMOS	5
1.4 DEFINICIONES	6
1.5 REFERENCIAS	6
1.6 RESUMEN	6
2. ESTRUCTURA DE LOS FICHEROS	7
2.1 INTRODUCCIÓN	7
2.2 CODIFICACIÓN DE ARCHIVOS DE CÓDIGO FUENTE	7
2.2.1 CONSIDERACIONES GENERALES SOBRE LAS REGIONES	8
2.3 CODIFICACIÓN DE ARCHIVOS DE CLASE Y ACCESO A DATOS .CS	9
2.3.1 REGIÓN DIRECTIVAS USING	9
2.3.2 REGIÓN DECLARACIÓN NAMESPACE Y CLASS.	9
2.3.3 REGIÓN ATRIBUTOS	10
2.3.4 REGIÓN CONSTRUCTORES	10
2.3.5 REGIÓN PROPIEDADES	10
2.3.6 REGIÓN MÉTODOS	10
2.4 CODIFICACIÓN DE ARCHIVOS DE RECURSOS DE INTERNACIONALIZACIÓN	11
3. REGLAS DE CODIFICACIÓN	12
3.1 IDENTACIÓN	12
3.1.1 LONGITUD DE LÍNEA	12
3.1.2 RUPTURA DE LÍNEAS	12
3.2 COMENTARIOS	12
3.2.1 APLICACIÓN DE LOS COMENTARIOS	13
3.2.2 FORMATOS DE IMPLEMENTACIÓN DE COMENTARIOS	14
3.3 DECLARACIONES	15
3.3.1 NÚMERO DE DECLARACIONES POR LÍNEA	15
3.3.2 INICIALIZACIÓN	15
3.3.3 SITUACIÓN	15
3.4 SENTENCIAS	16
3.4.1 SENTENCIAS SIMPLES	16
3.4.2 SENTENCIAS COMPUESTAS	16
3.4.3 SENTENCIAS DE RETORNO	16
3.4.4 SENTENCIAS IF, IF-ELSE, IF ELSE-IF ELSE	17
3.4.5 SENTENCIAS FOR	17
3.4.6 SENTENCIAS WHILE	18
3.4.7 SENTENCIAS DO-WHILE	18
3.4.8 SENTENCIAS SWITCH	18
3.4.9 SENTENCIAS TRY-CATCH	19

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.5	ESPACIOS EN BLANCO	19
3.5.1	LÍNEAS EN BLANCO	19
3.5.2	ESPACIOS EN BLANCO	20
3.6	CONVENCIONES DE NOMBRES	20
3.6.1	CLASES	20
3.6.2	MÉTODOS	21
3.6.3	VARIABLES Y PARÁMETROS	21
3.6.4	CONSTANTES	21
3.7	HÁBITOS DE PROGRAMACIÓN	21
3.7.1	REFERENCIAS A VARIABLES Y MÉTODOS DE CLASE	22
3.7.2	CONSTANTES	22
3.7.3	ASIGNACIONES DE VARIABLES	22
3.7.4	PARÉNTESIS	23
3.7.5	VARIABLES DE RETORNO	23
3.7.6	EXPRESIONES ANTES DE ‘?’ EN EL OPERADOR CONDICIONAL	23
3.7.7	COMENTARIOS ESPECIALES	23
4.	ESTANDAR DE NOMBRADO DE BASE DE DATOS	24
4.1	IDIOMA A UTILIZAR	24
4.2	CONVENCIONES DE NOMBRADO DE TABLAS	24
4.2.1	NOMBRADO DE TABLAS DE ENTIDAD	24
4.2.2	NOMBRADO DE TABLAS DE RELACIÓN	24
4.2.3	NOMBRES DE TABLAS PREDEFINIDOS	25
4.3	CONVENCIONES DE NOMBRADO DE CAMPOS	25
4.3.1	NOMBRADO DE CAMPOS DE TABLAS DE ENTIDAD	25
4.3.2	NOMBRADO DE CAMPOS DE TABLAS DE RELACIÓN	26
5.	BIBLIOGRAFÍA	27
5.1	REFERENCIAS	27
5.2	REFERENCIAS WEB	27

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

El presente documento describe las normas que deben seguirse en el desarrollo de cualquier tipo de elemento de codificación realizado en este proyecto. El objetivo primordial de la confección de un estándar de codificación se basa en homogeneizar el proceso de implementación y codificación, para lograr obtener beneficios en la comprensión del código, en la verificación y validación del mismo, y en posibles modificaciones posteriores.

Existen un gran número de razones por las que definir las convenciones de código es importante para los programadores:

- El 80% del coste del código de un programa se invierte en su mantenimiento.
- Casi ningún software lo mantiene toda su vida el autor original.
- Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
- La distribución de código fuente como producto exige su presentación de manera adecuada.

De este modo, el presente documento pretende ser una colección de reglas que deben aplicarse a todo el código generado, con el propósito de que sea homogéneo. Esta homogeneidad permitirá una comprensión más efectiva del código tanto para su autor como para otros programadores, facilitando su distribución y mantenimiento.

1.2 Alcance del documento

Este documento se ubica dentro de la fase inicial de desarrollo de la aplicación para la gestión y dirección de proyectos, como elemento necesario previo a la realización de cualquier tipo de código fuente del proyecto, y será utilizado como guía durante toda la fase de implementación.


1.3 Acrónimos

BD: Base de datos

HTML: HyperText Markup Language (*Lenguaje de Marcas de Hipertexto*),

ID: Identificador

URL: Uniform resource locator (*Localizador universal de recursos*)

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

1.4 Definiciones

- Pascal-Casing: Notación en la que los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula.
- Camel-Casing: Notación similar a la Pascal-Casing con la excepción que la letra inicial del identificador no debe estar en mayúscula.


1.5 Referencias

En este documento no se realiza ninguna referencia a otros documentos del proyecto:

1.6 Resumen

El presente documento describe las normas que deben seguirse en el desarrollo de cualquier tipo de elemento de codificación realizada en este proyecto

- Apartado 1. Se muestra el propósito del documento y se define su alcance. Se proporciona una lista de acrónimos y definiciones útiles para la comprensión del documento, así como una lista de los documentos del proyecto referenciados y el presente resumen.
- Apartado 2. Se muestra la estructura de codificación de los diferentes tipos de archivos de código fuente desarrollados en el proyecto.
- Apartado 3. Se muestran reglas, recomendaciones y buenas prácticas para el desarrollo del código fuente del proyecto.
- Apartado 4. Se muestran reglas de nombrado en bases de datos.
- Apartado 5. Bibliografía y referencias Web utilizadas en la confección de este documento.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

2. ESTRUCTURA DE LOS FICHEROS


2.1 Introducción

En este apartado se define la estructura y organización de los archivos de código fuente del proyecto, se especifica la codificación para los archivos de clases y acceso a datos (.cs) y para los archivos de recursos de internacionalización (.txt que son posteriormente compilados a .resx)

2.2 Codificación de archivos de código fuente

A continuación se muestra la estructura de codificación de los archivos de código subyacente que será comentada posteriormente. Para una comprensión más sencilla se muestra mediante una tabla, en la columna de la izquierda aparece un índice para facilitar la posterior descripción de la región de la estructura definida en la columna derecha.

1 Directivas using	<code>using System; <directivas using></code>
2 Declaración namespace y class	<code>namespace MyNamespace1 { public class MyClass {</code>
3 Región variables	<code> #region variables #region Variables I18N <declaraciones de variables de internacionalización> #endregion Variables I18N #region Variables globales <declaraciones de variables globales> #endregion Variables globales #region Variables de Sesión <Descripción de las variables de sesión utilizadas> #endregion Variables de Sesión #endregion variables</code>
4 Región I18N	<code> #region LoadI18N <Código LoadI18N> <Código class_Init> #endregion LoadI18N</code>
Cierre llaves de class y namespace	<code> } }</code>

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

2.2.1 Consideraciones generales sobre las regiones

Como se ha visto en el apartado 2.2 y para facilitar la organización y estructuración del código fuente se utilizan las directivas `#region` y `#endregion`. Las regiones no aportan funcionalidad como tal, se utilizan para marcar y agrupar una sección del código. Las regiones que se especifican en el apartado 2.2 son las obligatorias en el caso de que aparezcan los elementos para los que han sido definidas. En caso de que aparezcan elementos no especificados en las regiones del apartado anterior podrán definirse nuevas regiones para especificar la sección de código referida a los eventos y métodos del elemento o control. En cualquier caso, no se permitirán eventos o métodos que no estén incluidos dentro de alguna región.

2.2.1.1 Región Directivas using

En esta región se colocarán, por orden alfabético creciente, las directivas que especifiquen las clases utilizadas en el código fuente definido en la clase actual.

2.2.1.2 Región declaración namespace y class.

En esta región se colocarán las cabeceras que especifican el espacio de nombres en los que se integra el código y el nombre de la clase. Esta última irá precedida por una cabecera en la que se especificarán los siguientes datos:

```

/// <summary>
/// Nombre del fichero: Nombre del fichero
/// Autor           : Nombre del autor
/// Descripción     : Descripción de la funcionalidad y
objetivo del fichero
/// Copyright © 2009, ENDALIA, S.L. Todos los derechos
reservados.
/// </summary>

```

2.2.1.3 Región declaración de variables


Esta región estará integrada por dos subregiones que se especifican a continuación

- Región variables I18N: En esta región aparecen las declaraciones de cadenas de internacionalización que son obtenidas del archivo de recursos y que se utilizan para definir todos los textos que son presentados al usuario.
- Región variables globales: En esta región aparecen las declaraciones de variables globales utilizadas, acompañadas de una descripción de su funcionalidad.

2.2.1.4 Región I18N

En esta región se colocan dos métodos:

- Método `InitializeLabels()` : Realiza la lectura del fichero de recursos en el que se almacenan las cadenas de internacionalización, cargando estas en variables de cadena.
- Método `LoadI18N()`: Que carga en los campos de texto del formulario las variables de cadenas obtenidas en el método `InitializeLabels()`.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

2.3 Codificación de archivos de clase y acceso a datos .cs

A continuación se muestra la estructura de codificación de los archivos de acceso a datos y definición de clase. De la misma manera que en el apartado se muestra mediante una tabla con dos columnas, en la columna de la izquierda aparece un índice para facilitar la posterior descripción de la estructura definida en la columna de la derecha

1 Directivas using	<code>using System; <directivas using></code>
2 Declaración namespace y class	<code>namespace MyNamespace1 { public class MyClass {</code>
3 Región atributos	<code>#region atributos <Declaración atributos> #endregion atributos</code>
4 Región constructores	<code>#region constructores <Métodos constructores clase> #endregion constructores</code>
5 Región propiedades	<code>#region propiedades <declaración propiedades> #endregion propiedades</code>
6 Región Métodos	<code>#region métodos <Código Métodos> #endregion métodos</code>
7 Cierre de llaves de class y namespace	<code> } }</code>


2.3.1 Región Directivas using

En esta región se coloca, por orden alfabético creciente, las directivas que especifiquen las clases utilizadas en el código fuente definido en la clase actual.

2.3.2 Región declaración namespace y class.

En esta región se coloca las cabeceras que especifican el espacio de nombres en los que se integra el código y el nombre de la clase. Esta última irá precedida por una cabecera en la que se especificarán los siguientes datos:

```
/// <summary>
/// Nombre del fichero: Nombre del fichero
/// Autor           : Nombre del autor
/// Descripción      : Descripción de la funcionalidad y
objetivo del fichero
/// Copyright © 2009, ENDALIA, S.L. Todos los derechos
reservados.
/// </summary>
```

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

2.3.3 Región atributos

En esta región se colocará la declaración de los atributos de una clase. Los atributos se nombrarán mediante Pascal-Casing exceptuando la primera letra, que será en minúscula y precedida por un guión bajo de este modo:

```
private int _actionPlanItemID;
```

Asimismo en esta región se declararán las constantes de la clase, que se nombrarán con mayúsculas.

2.3.4 Región constructores

En esta región se colocará la declaración de los métodos constructores de la clase.


2.3.5 Región propiedades

En esta región se coloca la declaración de las propiedades públicas de la clase. Las propiedades se nombran con Pascal-Casing y en el caso de que representen el acceso al valor de un atributo de la clase su nombre es el mismo del atributo sin el guión bajo y con la primera letra en mayúscula, especificando el acceso a los métodos get y set de este modo:

```
public int ActionPlanItemID
{
    get{ return _actionPlanItemID; }
    set{ _actionPlanItemID = value; }
}
```

2.3.6 Región métodos

En esta región se colocará la declaración de los métodos de la clase.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	


2.4 Codificación de archivos de recursos de internacionalización

Los archivos txt a partir de los cuales se generan los archivos de recursos de internacionalización se construirán del siguiente modo:

Para cada una de las secciones del programa que tengan una entidad lo suficientemente importante como para ser diferenciada se colocará un comentario y a continuación la relación de las etiquetas. El nombrado de las etiquetas se hace del siguiente modo:

- La parte inicial del nombre de la etiqueta será la misma que el nombre del archivo .cs en el que se utilizará la etiqueta. A continuación se colocará un guión bajo seguido de un prefijo que indicará la utilización de la etiqueta seguida de un guión bajo siguiendo esta convención:
 - etiqueta o campo de texto: `_lbl_`
 - etiqueta de hyperlink: `_lnk_`
 - texto de botón : `_btn_`
 - Cabecera de columna de DataGridView: `_gridHeader_`
- En el caso de que la etiqueta sea un tooltip se colocará a continuación el prefijo `_Tip_`
- A continuación, se colocará un nombre descriptivo de la función de la etiqueta que utilizará Pascal-Casing. No se especifica una norma rígida, para este nombrado pero a continuación se muestran unos ejemplos que muestran buenas prácticas del mismo.

```
ProjectActionPlan_lbl_TitleTree
ProjectActionPlan_btn_SaveActionPlan
ProjectTask_btnImg_Tip_EditActionPlan
```

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3. REGLAS DE CODIFICACIÓN

3.1 Identación

Dada la actual uniformidad y estandarización de los editores utilizados para el desarrollo de código C# en .net se utilizará el tabulador como unidad de indentación estándar. Los comentarios se indentarán al mismo nivel de indentación que el código que se esté documentando.

3.1.1 Longitud de Línea

Se recomienda no escribir líneas con más de 80 caracteres, ya que no son bien manejadas por muchos terminales y herramientas.

3.1.2 Ruptura de Líneas

Cuando una expresión no entre en una sola línea, se debe romper de acuerdo a estos principios generales:

- Romper después de una coma.
- Romper antes de un operador.
- Preferir las rupturas de alto nivel a las de bajo nivel.
- Alinear la nueva línea con el principio de la expresión al mismo nivel de la línea anterior.

3.2 Comentarios

En C# hay tres formas de escribir comentarios:

La primera consiste en encerrar todo el texto que se desee comentar entre caracteres `/*` y `*/` siguiendo la siguiente sintaxis:


```
/*<texto>*/
```

- Estos comentarios pueden abarcar tantas líneas como sea necesario. No es posible anidar comentarios de este tipo.

En la segunda forma se considera como indicador del comienzo del comentario la pareja de caracteres `//` y como indicador de su final el fin de línea. Por tanto, la sintaxis que siguen estos comentarios es:

```
// <texto>
```

La tercera manera es utilizando el trío de caracteres `///`. Este tipo de comentario tiene la particularidad de ser reconocido y utilizado por las herramientas de generación automática de documentación y será el utilizado para la descripción de métodos y clases, ya que en el caso de los primeros genera la estructura de tags de documentación de nombres, parámetros y valores de retorno utilizados para la documentación automatizada.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.2.1 Aplicación de los comentarios

Como se ha comentado en el punto anterior, el tercer tipo de comentario (el que va precedido de los caracteres ///
se utiliza para crear de manera automática los tags que permiten la generación automática de documentación.


Aparte de este punto los comentarios deberían usarse para una introducción del código y proporcionar información adicional que no está disponible en el propio código. Los comentarios sólo deberían tener información que sea relevante para leer y entender el programa. Por ejemplo, información sobre cómo está construida la clase correspondiente o en qué directorio reside, no debería ser incluida como comentarios.

Las discusiones no triviales o decisiones de diseño no obvias son apropiadas, pero debemos evitar la duplicidad de información que esté presente en el código. Es demasiado fácil que los comentarios redundantes se queden anticuados. En general, debemos evitar cualquier comentario que se pueda quedar anticuado cuando el código evolucione.

La frecuencia en los comentarios algunas veces refleja una pobre calidad de código. Cuando nos sentimos obligados a llenarlo de comentarios, debemos considerar la reescritura del código para hacerlo más claro. Los comentarios no deben encerrarse en grandes cajas dibujadas con asteriscos u otros caracteres. Los comentarios nunca deberían incluir caracteres especiales como saltos de página, etc.

Los siguientes puntos son técnicas de comentarios recomendadas.

- Cuando se modifica el código, se mantienen siempre actualizados los comentarios circundantes.
- Evitar los comentarios recargados, como las líneas enteras de asteriscos. En su lugar se utilizan espacios para separar los comentarios y el código.
- Evitar rodear un bloque de comentarios con un marco tipográfico. Puede resultar agradable, pero es difícil de mantener.
- Antes de la implementación, quitar todos los comentarios temporales o innecesarios, para evitar cualquier confusión en la futura fase de mantenimiento.
- Si se necesita realizar comentarios para explicar una sección de código compleja, examinar el código para decidir si se debería volver a escribir. Siempre que sea posible, no documentar un código malo, volver a escribirlo. Aunque, por regla general, no debe sacrificarse el rendimiento para hacer un código más simple para el usuario, es indispensable un equilibrio entre rendimiento y mantenibilidad.
- Usar frases completas al escribir comentarios. Los comentarios deben aclarar el código, no añadirle ambigüedad.
- Ir comentando al mismo tiempo que se programa, porque probablemente no habrá tiempo de hacerlo más tarde. Por otro lado, aunque se tuviera oportunidad de revisar el código que se ha escrito, lo que parece obvio hoy es posible que seis semanas después no lo sea.
- Evitar comentarios superfluos o inapropiados, como comentarios divertidos al margen.
- Usar los comentarios para explicar el propósito del código como si fueran traducciones interlineales.

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

- Comentar cualquier cosa que no sea legible de forma obvia en el código.
- Para evitar problemas recurrentes, hacer siempre comentarios al depurar errores y solucionar problemas de codificación, especialmente cuando se trabaje en equipo.
- Hacer comentarios en el código que esté formado por bucles o bifurcaciones lógicas. Se trata en estos casos de áreas clave que ayudarán a los lectores del código fuente.
- Realizar los comentarios en un estilo uniforme, respetando una puntuación y estructura coherentes a lo largo de toda la aplicación.
- Separar los comentarios de sus delimitadores mediante espacios. Si se respeta esta norma, los comentarios serán más claros y fáciles de localizar si trabaja sin indicaciones de color.

3.2.2 Formatos de Implementación de Comentarios

Los programas pueden tener cuatro estilos de implementación de comentarios:

- **Bloque de comentarios:** Los bloques de comentarios se usan para proporcionar descripciones de ficheros, métodos, estructuras de datos y algoritmos. Los bloques de comentarios podrían usarse al principio de cada fichero y antes de cada método. También pueden usarse en otros lugares, como dentro de los métodos. Para este tipo de comentario se preferirá la estructura `/* - */`. Un bloque de comentario debería ir precedido por una línea en blanco para configurar un apartado del resto del código:


```
/*
 * Esto es un bloque de comentarios.
 */
```

- **Comentarios de una línea:** Los comentarios cortos pueden aparecer como una sola línea indentada al nivel del código que la sigue. Si un comentario no se puede escribir en una sola línea, debería seguir el formato de los bloques de comentario. Un comentario de una sola línea debería ir precedido de una sola línea en blanco. Para este tipo de comentario se preferirá utilizar los caracteres `//`. A continuación se muestra un ejemplo:

```
if (condition)
{
    // Código de la condición.
    ...
}
```

- **Comentarios finales:** Los comentarios muy cortos pueden aparecer en la misma línea que el código que describen, pero deberían separarse lo suficiente de las sentencias. Si aparece más de un comentario en el mismo trozo de código, deberían estar indentados a la misma altura. Para este tipo de comentario se preferirá utilizar los caracteres `//`. Aquí tenemos un ejemplo utilizando estos caracteres y la estructura `/* - */`:

```
if (a == 2)
{
    return TRUE;                // caso especial
```


	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

```

    }
    else
    {
        return isPrime(a);           /* otro comentario */
    }

```

3.3 Declaraciones

3.3.1 Número de declaraciones por línea

Se recomienda una declaración por línea ya que mejora los comentarios. En otras palabras:

```

int level; // nivel de indentación
int size;  // tamaño

```

Se prefiere sobre:

```

int level, size;

```

No debemos poner diferentes tipos en la misma línea. Por ejemplo:

```

int foo, fooarray[]; //Evitar

```

3.3.2 Inicialización

Debemos intentar inicializar las variables locales donde son declaradas. La única razón para no inicializar una variable donde es declarada es si el valor inicial depende de algún cálculo que tiene que ocurrir antes.

3.3.3 Situación

Ponemos las declaraciones sólo al principio de los bloques. No debemos esperar a declarar variables hasta que son usadas por primera vez; puede confundir al programador y estorbar la portabilidad del código dentro del ámbito.

```

void myMethod()
{
    int int1 = 0;           // comienzo de bloque

    if (condition)
    {
        int int2 = 0;       // comienzo de bloque if
    }
}


```

La única excepción a esta regla son los indexados para los bucles, que en C# pueden ser declarados en la sentencia for:

```

for (int i = 0; i < maxLoops; i++) { ... }

```

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

Debemos evitar las declaraciones locales que oculten las declaraciones de nivel superior. Por ejemplo, no debemos declarar el mismo nombre de variable en un bloque interno:

```
int count;

myMethod() {
    if (condition) {
        int count; // Evitar

    }
}
```

3.4 Sentencias

3.4.1 Sentencias Simples

Cada línea debe contener como máximo una sentencia. Por ejemplo:

```
argv++;      // Correcto
argc++;      // Correcto
argv++; argc--; // Evitar
```

3.4.2 Sentencias Compuestas

Las sentencias compuestas son sentencias que contienen listas de sentencias encerradas entre llaves "{sentencias}". Se ilustrarán ejemplos en las siguientes secciones:


- Las sentencias encerradas deben indentarse uno o más niveles que la sentencia compuesta.
- La llave ('{') de apertura debe empezar una nueva línea a continuación de la línea que empieza la sentencia compuesta; la llave de cierre ('}') debe empezar una nueva línea y estar indentada con el principio de la sentencia compuesta.
- Las llaves se usan alrededor de todas las sentencias, incluso para sentencias simples, cuando éstas forman parte de una estructura de control como una sentencia if-else o for. Esto hace más fácil la adición de sentencias sin introducir errores debido al olvido de las llaves.

Las únicas excepciones a estas reglas serán para los métodos get y set de las clases de acceso a datos descritos en el apartado 2.3.5.

3.4.3 Sentencias de Retorno

Una sentencia de retorno no deberá usar paréntesis a menos que el valor de retorno sea más obvio de esta forma. Por ejemplo:

```
return;
return myDisk.size();
return (size ? size : defaultSize);
```

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.4.4 Sentencias if, if-else, if else-if else

Las sentencias de tipo if-else deberá tener la siguiente forma:

```

if (condition)
{
    statements;
}

if (condition)
{
    statements;
}
else
{
    statements;
}

if (condition)
{
    statements;
}
else if (condition)
{
    statements;
}
else
{
    statements;
}

```

Las sentencias if siempre usan llaves. Debemos evitar el siguiente caso:

```

if (condition) //Evitar, se han omitido las llaves {}!
    statement;

```

3.4.5 Sentencias for

Una sentencia for deberá tener la siguiente forma:

```

for (initialization; condition; update)
{
    statements;
}

```


Una sentencia for vacía, en la cual todo el trabajo se hace en las cláusulas de inicialización, condición y actualización, deberá tener la siguiente forma:

```

for (initialization; condition; update);

```

Cuando usamos el operador como en las cláusulas de inicialización o actualización de una sentencia for, debemos evitar la complejidad de usar más de tres variables. Si es necesario, debemos usar sentencias separadas antes del bucle for, para la cláusula de inicialización, o al final del bucle, para la cláusula de actualización.

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.4.6 Sentencias while

Una sentencia while deberá tener la siguiente forma:

```
while (condition)
{
    statements;
}
```

Una sentencia while vacía deberá tener la siguiente forma:

```
while (condition);
```

3.4.7 Sentencias do-while

Una sentencia do-while deberá tener la siguiente forma:

```
do
{
    statements;
}
while (condition);
```

3.4.8 Sentencias switch

Una sentencia switch deberá tener la siguiente forma:


```
switch (expression)
{
    case constant-expression:
        statement
        break;

    case constant-expression:
        statement
        /* continua sin salto */

    [default:
        statement
        [jump-statement]
}
```

Cada vez que un case, no incluye una sentencia break, debemos añadir un comentario donde normalmente iría la sentencia break. Esto se ve en el ejemplo de código anterior con el comentario “/* continua sin salto */”. En cualquier caso se deberá evitar este tipo de construcción.

Toda sentencia switch deberá incluir un valor default. El break en el case por defecto es redundante, pero evita un error de caída si añadimos después otro case.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.4.9 Sentencias try-catch

Una sentencia try-catch deberá tener la siguiente forma:

```
try
{
    statements;
}
catch (Exception e)
{
    statements;
}
```

Una sentencia try-catch también puede ir seguida de un bloque finally, que se ejecuta sin importar si se ha completado con éxito o no el bloque try.

```
try
{
    statements;
}
catch (Exception e)
{
    statements;
}
finally
{
    statements;
}
```

3.5 Espacios en blanco

3.5.1 Líneas en blanco


Las líneas en blanco mejoran la lectura separando secciones de código que están relacionadas lógicamente.

Siempre se deberán usar dos líneas en blanco en las siguientes circunstancias:

- Entre secciones de un fichero fuente.
- Entre definiciones de clases e interfaces.

Siempre se deberán usar una línea en blanco en las siguientes circunstancias:

- Entre métodos.
- Entre las variables locales de un método y su primera sentencia.
- Antes de un bloque de comentarios o un comentario simple.
- Entre secciones lógicas dentro de un método para mejorar su lectura.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.5.2 Espacios en blanco

Los espacios en blanco deberán usarse en las siguientes circunstancias:

- Una palabra clave seguida por un paréntesis deberían estar separados por un espacio en blanco:

```
while (true)
{
}

```

- No se deberá usar un espacio en blanco entre un nombre de método y su paréntesis de apertura. Esto ayuda a distinguir las palabras clave de las llamadas a métodos.
- Después de las comas en una lista de argumentos debe aparecer un espacio en blanco.
- Todos los operadores binarios excepto "." Deberían estar separados de sus operandos por espacios. Los espacios en blanco nunca deben separar los operadores unarios como incremento ("++"), y decremento ("—") de sus operadores. Por ejemplo:

```
a += c + d;
a = (a + b) / (c * d);
while (d++ = s++)
{
    n++;
}

```

- Las expresiones de una sentencia deberán estar separadas por espacio. Por ejemplo:

```
for (expr1; expr2; expr3)

```


3.6 Convenciones de nombres

3.6.1 Clases

Se deberá usar pascal-casing para el nombrado de clases. Debemos intentar mantener los nombres de clases simples y descriptivos. Debemos usar palabras completas y evitar acrónimos y abreviaturas (a menos que la abreviatura se use muy ampliamente como URL o HTML). Ejemplo:

```
public class HelloWorld
{
}

```

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.6.2 Métodos

Los métodos deberán ser verbos. Se deberá usar pascal-casing para su nombrado. Ejemplo:

```
public class HelloWorld
{
    void SayHello(string name)
    {
        ...
    }
}
```

3.6.3 Variables y parámetros

Se usará camel-casing para el nombrado de variables y parámetros. Los nombres de variables deberán ser cortos y llenos de significado. La elección de una variable debería ser mnemónica es decir, diseñada para indicar al observador casual su utilización. Se deben evitar los nombres de variable de un sólo carácter, excepto para variables temporales. Algunos nombres comunes de este tipo de variables son: i, j, k, m, y n para enteros. Ejemplo:

```
public class HelloWorld
{
    int totalCount = 0;
    void SayHello(string name)
    {
        string fullMessage = "Hello " + name;
    }
}
```


3.6.4 Constantes

Los nombres de variables constantes de clases y las constantes ANSI deberán escribirse todo en mayúsculas con las palabras separadas por subrayados ("_"). (Se deberían evitar las constantes ANSI para facilitar la depuración.) Ejemplo:

```
public static int MIN_WIDTH = 4;
public static int MAX_WIDTH = 999;
public static int GET_THE_CPU = 1;
```

3.7 Hábitos de programación

No debemos hacer públicas ninguna variable de instancia o de clase sin una buena razón. A menudo las variables de instancia no necesitan ser asignadas/consultadas explícitamente. Normalmente, esto sucede como efecto lateral de llamadas a métodos. Un ejemplo apropiado de una variable de instancia pública es el caso en que la clase es esencialmente una estructura de datos, sin comportamiento.

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.7.1 Referencias a variables y métodos de clase

Evitar usar un objeto para acceder a una variable o método de clase (static). Usar el nombre de la clase en su lugar. Por ejemplo:

```
classMethod(); //correcto
AClass.classMethod(); //correcto
anObject.classMethod(); //Evitar
```

3.7.2 Constantes

Las constantes numéricas (literales) no se deben codificar directamente, excepto -1, 0 y 1, que pueden aparecer en un bucle for como contadores.

3.7.3 Asignaciones de variables

Evitar asignar el mismo valor a varias variables en la misma sentencia. Es difícil de leer. Ejemplo:

```
fooBar.fChar = barFoo.lchar = 'c'; // AVOID!
```

No usar el operador de asignación en un lugar donde se pueda confundir con el de igualdad. Ejemplo:

```
if (c++ = d++) { // AVOID!
}
}
```

Se debe escribir:


```
if ((c++ = d++) != 0)
{
}
}
```

No debemos usar asignación embebida como un intento de mejorar el rendimiento en tiempo de ejecución. Ese es el trabajo del compilador. Ejemplo:

```
d = (a = b + c) + r; // AVOID!
```

Debería escribirse como:

```
a = b + c;
d = a + r;
```


	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

3.7.4 Paréntesis

En general es una buena idea usar paréntesis en expresiones que implican distintos operadores para evitar problemas con el orden de precedencia de los operadores. Incluso si parece claro el orden de precedencia de los operadores, podría no ser así para otros. No se debe asumir que otros programadores conozcan el orden de precedencia.

```
if (a == b && c == d) // Evitar
if ((a == b) && (c == d)) // Correcto
```

3.7.5 Variables de retorno

Debemos intentar hacer que la estructura de nuestro programa se corresponda con nuestra intención. Por ejemplo:

```
if ( booleanExpression)
{
    return true;
}
else
{
    return false;
}
```

debería escribirse:

```
return booleanExpression;
```

De forma similar,

```
if (condition)
{
    return x;
}
return y;
```

Debería escribirse como:

```
return (condition ? x : y);
```


3.7.6 Expresiones antes de '?' en el operador condicional

Si una expresión contiene un operador binario antes de ? en el operador ternario ?:, se debe colocar entre paréntesis. Ejemplo:

```
(x >= 0) ? x : -x;
```

3.7.7 Comentarios Especiales

Debemos usar XXX, en un comentario para indicar que algo tiene algún error pero funciona. Usar FIXME para marcar algo que tiene algún error y no funciona.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

4. ESTANDAR DE NOMBRADO DE BASE DE DATOS

4.1 Idioma a utilizar

Para el nombrado de todos los elementos de la base de datos el idioma a utilizar será el **inglés**, salvo que se especifique de manera explícita lo contrario.

4.2 Convenciones de nombrado de tablas

4.2.1 Nombrado de tablas de entidad

El nombrado de las tablas de entidad dentro de la BD seguirá la siguiente codificación:

- Todos los nombres comenzarán por tres letras descriptivas del módulo o ámbito de aplicación de la tabla. La primera de estas tres letras será mayúscula y las otras dos restantes serán minúsculas. Los prefijos generales predefinidos con una descripción de los mismos se encuentran en el apartado 2.3 de este documento. En el caso de que el ámbito de aplicación de la tabla a nombrar difiera de los que aparecen en este apartado se definirá un nuevo prefijo y se añadirá a los existentes.
- En el caso de que la tabla forme parte del desarrollo de una sección en concreto dentro de un módulo, o forme parte de un conjunto de tablas relacionadas entre sí dentro de un ámbito de aplicación de la BD, se colocarán después del prefijo anterior y separadas por un guión bajo '_' tres letras descriptivas de la sección. La primera de estas letras será mayúscula y las otras dos restantes serán minúsculas. Si la tabla no tuviese relación con otras dentro del ámbito definido por el prefijo inicial estas tres letras no son necesarias.
- A continuación y separado por un guión bajo '_' se colocará un nombre descriptivo en plural del contenido de la tabla. Se deberá usar pascal-casing para este nombrado. Se deberá intentar mantener los nombres simples y descriptivos. Se deberán usar palabras completas y evitar acrónimos y abreviaturas (a menos que la abreviatura se use muy ampliamente como ID, URL o HTML, en cuyo caso puede escribirse en mayúsculas)

Ejemplos:


```
Gen_SystemOptions
Org_Eva_Developments
Org_Tra_GenericAreas
Org_Tra_GenericCourses
```

4.2.2 Nombrado de tablas de relación

Las tablas de relación comenzarán con un prefijo 'R_'.

A continuación, y en singular, se colocarán utilizando pascal-casing el nombre descriptivo de las tablas de entidad que relaciona, pudiendo reducirse si es muy largo.

De este modo para relacionar las tablas 'Prj_ActionPlanItems' y 'Orh_Emp_Employees' se creará la tabla 'R_ActionPlanItemEmployee'

	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

4.2.3 Nombres de tablas predefinidos

- R : Relación. Tablas de relación.
- Gen : Generic. Tablas relativas a la organización general de la BD
- Org : Organización y recursos humanos.
- Prj : Projects. Tablas relativas a proyectos
- Pro : Tablas relativas a procesos

4.3 Convenciones de nombrado de campos

Una norma establecida es que no puede haber dos campos dentro de una misma BD con el mismo nombre. Asimismo, el orden de los campos de una tabla debe seguir un orden lógico, (no tiene sentido colocar, <nombre, teléfono, apellidos, fax...>, sino <nombre, apellidos, teléfono, fax...>) Los nombres de los campos, dependiendo de si pertenecen a una tabla de entidad o de relación, se construyen siguiendo las normas de los siguientes apartados:

4.3.1 Nombrado de campos de tablas de entidad

4.3.1.1 Nombrado de claves primarias

Las claves primarias comenzarán con el prefijo 'pk_' y a continuación seguirán las convenciones de nombrado del apartado 3.1.3. Como finalización del nombre tendrán la terminación 'ID'.

4.3.1.2 Nombrado de claves ajenas

Las claves ajenas comenzarán con el prefijo 'fk_' y a continuación seguirán las convenciones de nombrado del apartado 3.1.3. Si la clave ajena apunta a una clave primaria de otra tabla como finalización del nombre tendrá la terminación 'ID'.


4.3.1.3 Nombrado de campos

Todos los campos de una misma tabla tendrán un mismo prefijo que resumirá el nombre descriptivo de la tabla. La primera de estas letras será mayúscula y las restantes serán minúsculas. Por ejemplo, si la tabla es 'Org_Reg_DataBind' todos los campos de la tabla en este punto empezarían por 'Dbn' o 'Dbind'. Otro ejemplo: Si la tabla es 'Prj_Diary_Notes' los campos de la tabla empezarían por 'Diary' o 'Dia'.

A continuación se colocará el nombre del campo. Para nombrar los campos de la tabla se utilizarán nombres descriptivos de su significado. Se deberá usar pascal-casing para este nombrado. Se deberá intentar mantener los nombres simples y descriptivos. Se deberán usar palabras completas y evitar acrónimos y abreviaturas (a menos que la abreviatura se use muy ampliamente como ID, URL o HTML, en cuyo caso puede escribirse en mayúsculas)

Ejemplos:

```
fk_AppEndDate
pk_AppDate
AppRol
AppGender
```

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

4.3.2 Nombrado de campos de tablas de relación

4.3.2.1 Nombrado de Claves ajenas

Las claves ajenas comenzarán por 'fk_', a continuación aparecerán, utilizando pascal-casing, los nombres descriptivos de la tablas de entidad que se relacionan (pudiendo resumirse si son muy largos), y se finalizarán con el nombre descriptivo de la tabla a la que apuntan seguido de ID.

Ejemplo:

En la tabla 'R_ActionPlanItemEmployee' que relaciona las tablas 'Prj_ActionPlanItem' y 'Orh_Emp_Employees' se nombrarán de esta manera las claves ajenas:

```
fk_ActionPlanItemEmployeeID
fk_ActionPlanItemProjectID
```


4.3.2.2 Nombrado de campos

Los campos en las tablas de relación se nombrarán mediante pascal-casing. Comenzarán con los nombres descriptivos de la tablas de entidad que se relacionan (pudiendo resumirse si son muy largos). A continuación se colocará el nombre del campo. Para nombrar los campos de la tabla se utilizarán nombres descriptivos de su significado. Se deberá intentar mantener los nombres simples y descriptivos. Se deberán usar palabras completas y evitar acrónimos y abreviaturas (a menos que la abreviatura se use muy ampliamente como ID, URL o HTML, en cuyo caso puede escribirse en mayúsculas)

Ejemplos:

En la tabla 'R_ProjectEmployee' que relaciona las tablas 'Prj_Projects_And_Tasks' y 'Orh_Per_Employees' los nombres de los campos podrían ser:

```
ProjEmpRemarks
ProjEmpDuration
ProjEmpIsCompulsory
```

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Estándar de codificación	Fecha: 18/09/2009
	ESTANDARCODIFICACION.doc	

5. BIBLIOGRAFÍA

5.1 Referencias

[LOW, 2004] Juval Lowy 2003. *C# Coding Standard*. Idesign Inc 2004.

5.2 Referencias Web

[Ref. Web 1] <http://www.dotnetspider.com/technology/tutorials/BestPractices.aspx>

[Ref. Web 2] <http://msdn.microsoft.com/>


[Ref. Web 3] <http://www.idesign.net/idesign/>

[Ref. Web. 4] http://vyaskn.tripod.com/object_naming.htm

Sistema para la gestión de proyectos

Gestión de configuraciones

Versión 1.2 – Fecha: 04/09/2009

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
04/09/2007	1.0	Documento de gestión de configuraciones de Endalia	Alfonso Hinojar
12/09/2007	1.1	Revisión del documento	Fernando Cortés
04/09/2009	1.2	Adaptación del documento para adaptarlos a las características del proyecto	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

TABLA DE CONTENIDOS

<u>1. INTRODUCCIÓN</u>	4
1.1 PROPÓSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 ACRÓNIMOS	4
1.4 DEFINICIONES	4
1.5 REFERENCIAS	5
1.6 RESUMEN	6
<u>2. ESPECIFICACIONES DE GESTIÓN</u>	7
<u>3. ACTIVIDADES DE GESTIÓN DE CONFIGURACIONES</u>	8
3.1 IDENTIFICACIÓN DE LA CONFIGURACIÓN	8
3.2 GESTIÓN DE LA CONFIGURACIÓN	8
3.3 CONTROL DE CAMBIOS DE LA CONFIGURACIÓN	9
<u>4. HERRAMIENTAS UTILIZADAS</u>	10
<u>5. BIBLIOGRAFÍA</u>	14
5.1 REFERENCIAS	14
5.2 REFERENCIAS WEB	14

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

En este documento, se describe cómo se gestionan las configuraciones de software, para mantener la integridad de los productos desarrollados durante la realización del proyecto.

En el desarrollo de software son habituales los cambios en todos los elementos generados, debido principalmente a modificaciones de especificación de requisitos y fallos de análisis o diseño. Por este motivo es necesario llevar un control y registro de los cambios con el fin de reducir errores, aumentar la calidad y la productividad y evitar los problemas que puede acarrear una incorrecta sincronización en dichos cambios, al afectar a otros elementos del sistema o de las tareas realizadas por otros miembros del equipo de trabajo.

1.2 Alcance del documento

Las actividades de GCS se prolongan a lo largo del ciclo de vida del producto software.

1.3 Acrónimos

CCC: Comité de Control de Cambios.

ECS: Elemento de Configuración de Software.

GC: Gestor de Configuraciones.

GCS: Gestión de Configuraciones Software.


IEEE: *Institute of Electrical and Electronic Engineers* (Instituto de Ingenieros Eléctricos y Electrónicos).

RUP: *Rational Unified Process*.

TFS: Team Foundation Server

1.4 Definiciones


- Artefacto: Cualquier tipo de información creada, producida, cambiada o utilizada por el equipo de desarrollo del proyecto.
- Camel-Casing: Notación similar a Pascal-Casing con la excepción de que la letra inicial del identificador debe ser minúscula.
- Elemento de Configuración de Software: cualquier artefacto sujeto a todas las especificaciones estipuladas en el plan de gestión de configuraciones de software.
- Línea base: Punto de referencia en el proceso de desarrollo que queda marcado por la aprobación de uno o varios elementos de Configuración de Software, mediante una revisión técnica formal.
- Pascal-Casing: Notación en la que un identificador está compuesto por múltiples palabras juntas comenzando cada una de ellas por una letra mayúscula.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

- Revisión: Instancia de un ECS, en un momento dado del proceso de desarrollo, que es almacenada en un repositorio, y que puede ser recuperada en cualquier momento para su uso o modificación.

1.5 Referencias


En este documento no se realizan referencias a otros documentos del proyecto.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

1.6 Resumen

Este documento describe el plan de gestión de configuraciones del sistema de gestión y dirección de proyectos. Se compone de cinco apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Descripción de la organización de la gestión de configuraciones.
- Apartado 3: Especificación de las tareas a realizar para llevar a cabo la gestión de configuraciones.
- Apartado 4: Descripción de las herramientas utilizadas para la gestión de configuraciones.
- Apartado 5: Bibliografía y referencias Web utilizadas para la realización de este documento.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	


2. ESPECIFICACIONES DE GESTIÓN

La gestión de configuraciones de software es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando e informando del estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

El propósito de la Gestion de Configuracion del Software es establecer y mantener la integridad de los productos de software durante su ciclo de vida, para lo que, siguiendo el estándar del IEEE Std 828-1990, se realizan las siguientes:

- **Identificación de la Configuración:** consiste en identificar la estructura del producto, sus componentes y la naturaleza de estos, y en hacerlos únicos y accesibles de alguna forma.
- **Control de Cambios en la Configuración:** consiste en controlar las versiones y entregas de un producto y el control de cambios que se producen en el a lo largo de su ciclo de vida.
- **Generación de Informes de Estado:** consiste en informar acerca del estado de los componentes de un producto y de las solicitudes de cambio, recogiendo estadísticas acerca de la evolución del producto.
- **Auditoria de la Configuración:** consiste en validar la completitud de un producto y la consistencia entre sus componentes, asegurando que el producto es lo que el usuario quiere.

No obstante, debido a las características de este proyecto en particular, se considera que la realización de las dos últimas actividades no aporta beneficios suficientes como para implantar su procedimiento, por lo que se desestima su empleo.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

3. ACTIVIDADES DE GESTIÓN DE CONFIGURACIONES

A continuación se describen los artefactos considerados para almacenar en las bibliotecas junto con sus convenciones de nombrado, así como el modo de gestionar el control de cambios de dichas configuraciones.

3.1 Identificación de la configuración

Para la correcta clasificación de los artefactos, se van a agrupar en distintas líneas base, asignando identificadores apropiados a cada uno de ellos. La definición de las líneas base se va a basar en los flujos de trabajo definidos en la metodología RUP utilizada, lo que lleva a la siguiente taxonomía:

- Planificación del proyecto
- Gestión de configuraciones
- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas
- Entrega del producto


Para nombrar cada uno de los ECS que forman los diferentes artefactos definidos en este documento, se seguirán las siguientes convenciones:

El nombre debe estar escrito en mayúsculas, y resumirá e indicará de modo intuitivo el contenido del artefacto, seguido por una breve descripción del tipo de elemento por el que está formado. Excepto en el caso de artefactos correspondientes a código fuente, se nombrarán según la notación “Pascal-Casing”. En caso de tratarse de archivos binarios y scripts, serán nombrados según la notación “Camel-Casing”. El nombre del artefacto, siempre terminará con un punto, y la extensión del contenido del mismo, siguiendo lo estándares de nombrado de archivos.

3.2 Gestión de la configuración

Para la gestión de configuraciones del presente proyecto, se definen las siguientes bibliotecas:

- Biblioteca de trabajo. Se establece al inicio del proyecto y gestiona el área de trabajo de los ECS que se encuentran activos, esto es sobre los que se están realizando modificaciones.
- Biblioteca maestra. Contiene elementos de configuración finalizados respecto a una iteración concreta del proceso de desarrollo. Sus elementos solamente pueden ser accedidos en modo de lectura una vez han sido establecidos.
- Biblioteca de copias de seguridad. Contiene copias incrementales de las bibliotecas de trabajo y maestra generadas periódicamente.


 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

Para la gestión de las bibliotecas se emplea la herramienta Microsoft Team Foundation Server la cual será explicada en detalle en el apartado 4 de este documento. (Ver apartado 4. Herramientas utilizadas)

3.3 Control de cambios de la configuración

Para la gestión de cambios de la configuración, se van a tener en cuenta dos tipos de modificaciones:

- Control de cambios informal. Cambios realizados en ECS fuera de la política de control de cambios, siendo éstos responsabilidad del desarrollador. Esta ausencia de control está motivada por la necesidad de realizar modificaciones de un modo más dinámico al comienzo del desarrollo de un ECS cuando los cambios son continuos y las notificaciones y controles de corrección podrían saturar el proceso.
- Control de cambios formal. Cambios realizados cuando un ECS es transferido a la biblioteca de trabajo o a la biblioteca maestra, para ello, el responsable de la modificación debe realizar una solicitud de cambio indicando si éste está motivado por la detección y resolución de un defecto, por una variación en los requisitos o por la realización de una mejora, la prioridad de la modificación según la relevancia de esta y la línea base a la que afecta. Una vez realizada la petición, el CCC debe revisar el ECS y aprobarlo si estima que es correcto o rechazarlo en caso contrario devolviéndolo a la versión anterior.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

4. HERRAMIENTAS UTILIZADAS


La herramienta principal utilizada para la gestión de cambios de las configuraciones de software es Microsoft Team Foundation Server (TFS) que es una aplicación que se integra con el entorno de desarrollo Microsoft Visual Studio. Esta herramienta permite llevar a cabo la gestión de código, recopilación de datos, generación de informes y trazado de proyectos en un entorno colaborativo de desarrollo de software.

El TFS proporciona una interfaz completamente integrada con el entorno de desarrollo Visual Studio, utilizado para el desarrollo del presente proyecto, facilitando en gran medida las labores de petición, comprobación y gestión de modificaciones de ECS. Asimismo, proporciona herramientas para gestionar posibles conflictos en ECS modificados por distintos desarrolladores y el acceso que estos deben tener a cada uno de los recursos.

También permite la gestión de diferentes bibliotecas, aunque, por motivos de seguridad, la biblioteca de copias de seguridad debe ser gestionada de un modo alternativo, mediante copias incrementales de ficheros.

A continuación, se describen las principales funcionalidades del TFS empleadas para la gestión de configuraciones:

- Bloquear para modificación. Esta funcionalidad permite el bloqueo tanto exclusivo como no exclusivo de un ECS, indicando la intención por parte del desarrollador de modificarlo. Si lo bloquea en modo exclusivo otros desarrolladores solo podrán acceder a ese archivo en modo lectura.

	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

- Descargar versión de la biblioteca. Esta funcionalidad permite la actualización de ECS modificados por el resto del equipo en la biblioteca local de cada desarrollador.

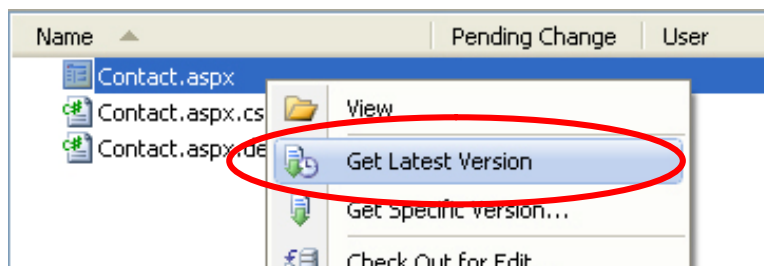


Figura 2. Descarga de versión

- Modificación de ECS. Esta funcionalidad permite la actualización de uno o varios ECS por parte de un desarrollador que los tenía previamente bloqueados o en edición. Es necesario indicar qué se ha realizado en esa modificación, o qué ha motivado su realización:
 - Indicando en modo texto el motivo del cambio en el campo de comentarios, manteniendo un encabezado con un formato específico de cabecera determinado por el departamento de soporte.

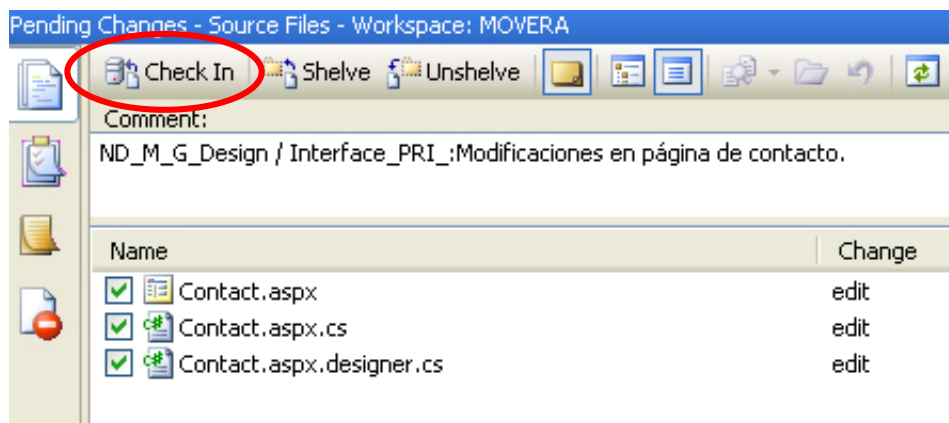



Figura 3. Modificación de ECS

	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

- Resolución de conflicto en ECS modificado por diferentes desarrolladores. Esta funcionalidad permite mezclar el contenido de un ECS modificado simultáneamente por diferentes desarrolladores. Para que surja la necesidad de emplear esta funcionalidad es necesario que los desarrolladores bloqueen dicho artefacto en modo no exclusivo. Este tipo de bloqueo se realiza automáticamente en el momento que se edita un artefacto.

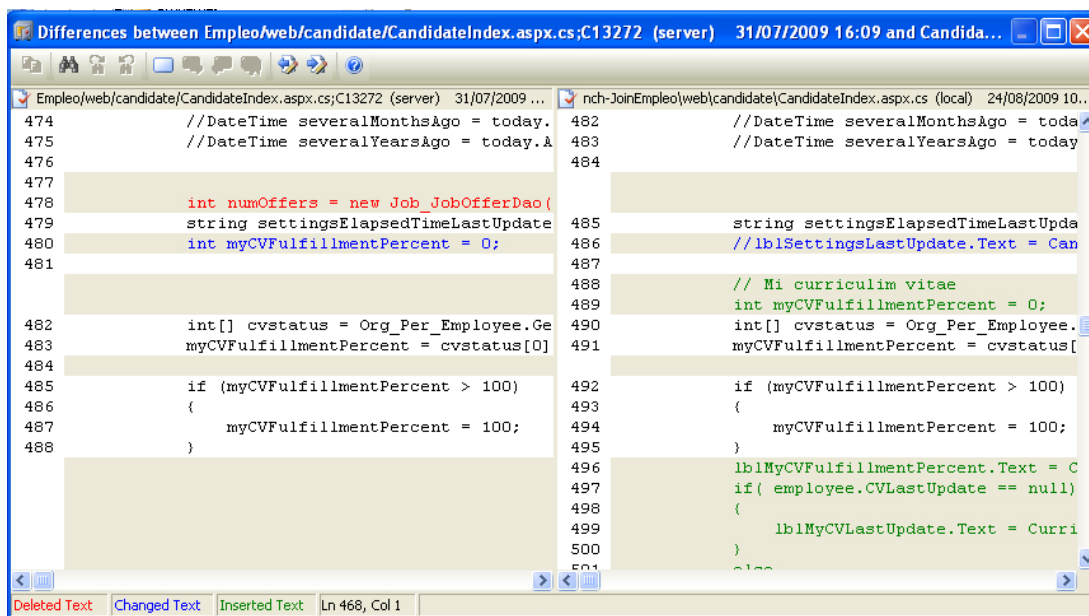



Figura 4. Resolución de conflictos

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Gestión de configuraciones	Fecha: 04/09/2009
	GESTIONCONFIGURACIONES.doc	

5. BIBLIOGRAFÍA

5.1 Referencias

[IGJ, 2000] - I. Jacobson, G. Booch, J. Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. Pearson Education

5.2 Referencias Web

[Ref. Web 1] <http://www.wikipedia.org>


[Ref. Web 2] <http://www.ieee.org>

[Ref. Web 3] <http://www.histaintl.com/soluciones/configuracion/configuracion.php>

Sistema para la gestión de proyectos

Estudio de mercado

Versión 1.0 – Fecha: 25/09/2009

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
21/09/2009	1.0	Versión inicial del documento de estudio de mercado	José Ángel Peribáñez
23/09/2009	1.1	Revisión del documento	Fernando Cortes
25/09/2009	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

TABLA DE CONTENIDOS

<u>1. INTRODUCCIÓN</u>	4
1.1 PROPÓSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 ACRÓNIMOS	4
1.4 DEFINICIONES	4
1.5 REFERENCIAS	5
1.6 RESUMEN	5
<u>2. LA GESTION DE PROYECTOS</u>	6
2.1 INTRODUCCIÓN	6
2.2 DEFINICIÓN DE LA GESTIÓN DE PROYECTOS	6
2.3 PERSPECTIVA HISTÓRICA	6
2.4 OBJETIVOS DE LA GESTIÓN DE PROYECTOS	7
<u>3. SISTEMAS DE GESTIÓN DE PROYECTOS</u>	8
3.1 INTRODUCCIÓN	8
3.2 APLICACIONES ACTUALES	8
3.2.1 MICROSOFT PROJECT	8
3.2.2 GANTT PROJECT	11
3.2.3 OPEN PROJECT	13
<u>4. CONCLUSIONES</u>	15
<u>5. BIBLIOGRAFÍA</u>	16
5.1 REFERENCIAS	16
5.2 REFERENCIAS WEB	16

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

En este documento se realiza el estudio de mercado de este proyecto. Se ha llevado a cabo un estudio del mercado de software centrado en el análisis de aplicación de gestión de proyectos. El propósito de un estudio de mercado es la obtención de la mayor cantidad posible de información relacionada con un producto o servicio, la cual se utiliza para identificar y definir las necesidades, oportunidades y problemas del mercado; para generar, refinar y evaluar las medidas de comercialización y para mejorar la comprensión del proceso y posibilidades del mismo.

El estudio de mercado que se va a llevar a cabo va a tener un carácter genérico, con el objeto de recabar información acerca del contexto actual en el que se va a enmarcar este proyecto.

1.2 Alcance del documento


Este documento constituye la fase inicial de estudio dentro del proyecto de desarrollo de la aplicación de dirección y gestión de proyectos. Las conclusiones obtenidas de este estudio serán aplicadas en el resto de las fases del proyecto. Su alcance comprende todo el periodo temporal del proyecto ya que sus contenidos pueden ser modificados en cualquier momento.

1.3 Acrónimos

- EPM: Enterprise Project Management
- GUI: Graphical User Interface
- IEEE: Institute of electrical and Electronics Engineers
- PMI: Project Management Institute
- PYME: Pequeña y mediana empresa
- SGP: Sistema de gestión de proyectos

1.4 Definiciones

- Interfaz gráfico: Conjunto de imágenes, objetos pictóricos y texto que componen la capa con la que interactúa un usuario con un sistema computacional.
- Sistema de gestión de proyectos: Herramienta que permite la gestión de los diferentes recursos que intervienen en desarrollo de un proyecto.
- Software libre: Software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido sin coste alguno.
- Outsourcing: Proceso en el cual una empresa identifica una porción de su proceso de negocio que podría ser desempeñada más eficientemente y/o mas efectivamente por otra corporación, la cual es contratada para desarrollar esa porción de negocio.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

- Graphical User Interface (GUI): método para facilitar la interacción del usuario con el ordenador a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas...) además de texto.


1.5 Referencias

En este documento no se realizan referencias a otros documentos del proyecto.

1.6 Resumen

Este documento describe el estudio de mercado del proyecto. Se compone de cinco apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Visión global de la técnica de la gestión de proyectos, proporcionando una perspectiva histórica de la misma, y una descripción de sus objetivos principales.
- Apartado 3: Descripción de algunas aplicaciones de gestión de proyectos.
- Apartado 4: Conclusiones obtenidas en la realización del estudio de mercado.
- Apartado 5: Bibliografía y referencias Web utilizadas para la realización de este documento.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

2. LA GESTION DE PROYECTOS

2.1 Introducción

En este apartado se va a realizar un estudio genérico de la gestión de proyectos. Esta visión global nos permitirá definir más adecuadamente cuales son los elementos, técnicas y fases que debemos trasladar del aspecto conceptual de la gestión de proyectos al desarrollo real del SGP.

En este apartado se desarrollan los siguientes temas:

- Subapartado 2.2: Definición de la gestión de proyectos.
- Subapartado 2.3: Perspectiva histórica
- Subapartado 2.4: Objetivos de la gestión de proyectos


2.2 Definición de la gestión de proyectos

El hecho de que el concepto de proyecto se aplique a prácticamente todas las áreas que requieren algún tipo de administración y organización de acciones y recursos, hace que en cada uno de estos campos la definición y aplicación de lo que es un proyecto varíe de manera sustancial, no teniendo excesiva relación entre sí. Como definición genérica del mismo se puede decir que es un conjunto de actividades interdependientes orientadas a un fin específico, con una duración predeterminada. Completar con éxito el proyecto significa cumplir con los objetivos dentro de las especificaciones técnicas, de costo y de plazo de terminación.

Para la definición del concepto de proyecto, de sus fases y de sus áreas de actuación, utilizamos la referencia del PMI, "La gestión de proyectos es la aplicación de conocimiento, habilidades, herramientas y técnicas con el objetivos de cumplir los requerimientos de un proyecto particular. Consta de cinco procesos: iniciación, planificación, ejecución, control y cierre, e integra nueve áreas de estudio: Integración, alcance, tiempo del proyecto, coste, calidad del proyecto, recursos humanos asignados, comunicación del proyecto, gestión de riesgos y abastecimiento del proyecto".

2.3 Perspectiva histórica

La aplicación del concepto de proyecto a prácticamente todas las áreas de actuación humana hace que sea imposible una recopilación exhaustiva de la evolución histórica de la gestión de proyectos. En cada campo que han existido proyectos la historia de la técnica y herramientas de gestión de los mismos ha sido muy diferente y dependiente del contexto histórico y científico del momento. Obviando de nuevo las particularidades y tomando una perspectiva generalista la gestión de proyectos ha existido desde tiempos muy antiguos. Ha estado históricamente relacionada con proyectos de ingeniería de construcción de obras civiles o en campañas militares, donde entran en juego muchos elementos de gestión. Pero es a partir de la II Guerra Mundial cuando el avance de estas técnicas desde el punto de vista profesional han transformado la administración de proyectos en una disciplina de investigación que involucra


 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

actualmente instituciones universitarias, civiles y gubernamentales en todo el mundo.

2.4 Objetivos de la gestión de proyectos

En este apartado se van a detallar los objetivos principales de la gestión de proyectos. Existen múltiples relaciones de estos objetivos pero todas clasifican los siguientes conceptos genéricos:

- Definir el proyecto: La gestión de proyectos tiene que establecer con claridad la naturaleza del proyecto y su ámbito de existencia. Esto puede parecer una obviedad pero es necesario que el proyecto tenga una razón de ser lógica y evidente a partir de la cual se puedan articular todas las acciones necesarias para su desarrollo.
- Optimizar la estructura organizativa del equipo: Es necesario buscar la organización, número, capacidad y ubicación más adecuada de los recursos humanos que van a trabajar en el proyecto.
- Establecer adecuadamente los objetivos y la planificación del Proyecto: La gestión debe definir de manera concisa y sin posibles interpretaciones las metas finales que debe cumplir el proyecto una vez haya sido finalizado, así como las diferentes fases necesarias para alcanzar de manera optima dichas metas.
- Hacer estimaciones razonables de tiempo, costes y recursos: Es necesario evaluar la capacidad de los diferentes recursos, tanto materiales como humanos, que se van emplear, así como medir el esfuerzo necesario para el cumplimiento de las tareas del proyecto. A partir de esta información se debe realizar una planificación lógica y realista de los costes temporales y económicos del mismo.
- Prever los posibles riesgos que puedan surgir: Es necesario un análisis de los posibles errores que pueden aparecer en el proyecto. La gestión de proyectos debe facilitar que se anticipen los problemas más usuales, y se implanten mecanismos para mitigar o evitar los menos previsibles.
- Mantener un control detallado de la evolución del proyecto: Se debe proveer de las herramientas necesarias para poder evaluar en cada momento el cumplimiento de las tareas en el calendario previsto, la situación y evolución de los recursos asignados y de los responsables de los mismos y el cumplimiento de las especificaciones de calidad. Se debe, asimismo, facilitar la comunicación, organización y archivo de la evolución de esta información.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

3. SISTEMAS DE GESTIÓN DE PROYECTOS

3.1 Introducción

En este apartado se analizan las características de diferentes soluciones software que se utilizan como herramienta en la gestión de proyectos.


3.2 Aplicaciones actuales

3.2.1 Microsoft Project

Microsoft Project es un programa de la suite Microsoft Office usado para la gestión de proyectos, es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

Las características proporcionadas por esta aplicación se han convertido en las clásicas que proporciona cualquier gestor de proyectos:

- Definir el proyecto: Crea entidades 'proyecto', sobre las que se trabaja en el programa y a las que se proporciona datos identificativos (nombre, objetivo, fechas, responsabilidades...) Posteriormente cada proyecto puede ser configurado con el resto de opciones.
- Crear una lista de las tareas del proyecto: Dentro de cada proyecto se puede crear una jerarquía de las fases del proyecto. En cada una de estas fases se definen las tareas y subtareas que constituyen el mismo, configuradas con datos identificativos (nombre, objetivos, fechas, responsables,...igual que los proyectos) que permiten diferenciar unas de otras.
- Definir periodos laborales: Permite ajustar el calendario del periodo temporal del proyecto definiendo, las horas laborales para cada día de la semana, así como los días festivos que van a existir, informando cuando se asignen tareas a días u horas festivos.
- Establecer fechas límites y delimitar tareas: Permite establecer días límite en que las tareas o el proyecto mismo deben estar terminados. Provee un sistema de aviso, mediante mensajes en ventana.
- Identificar riesgos del proyecto: Permite definir una tabla de riesgos, clasificada según la gravedad de los mismos, que se asignan a cada una de las tareas del proyecto.
- Agregar documentos al proyecto: Es posible vincular documentos de cualquier tipo como información anexa a cada una de las tareas del proyecto.
- Gestión de personal: Permite crear una relación jerárquica de personal que trabaja en el proyecto y asignarlo a diferentes tareas.


 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

- Agregar recursos y asignarlos a tareas: Se pueden definir recursos físicos y anexarlos como dependientes para cada una de las tareas
- Realizar seguimiento mediante informes: Genera de manera automática informes de diferente tipo con la información del avance y la configuración del proyecto. Provee la posibilidad de crear gráficos de diagrama de Gantt, diagrama de Red, diagrama de Gantt de seguimiento y grafico de recursos y asignación de los mismos a tareas.

Además de estas características que podríamos denominar ‘estándar’ dentro del distinto software de gestión de proyectos, Microsoft Project presenta alguna característica específica:

- Seguimiento del origen de las incidencias: Los controladores de tareas le ayudan a determinar el factor (como dependencia de tareas, limitaciones de calendario, fecha de programación o período vacacional) que marca la fecha de inicio de la tarea, de manera que se puede seguir una cadena de factores hasta llegar a la causa inicial de un retraso determinado.
- Ver la repercusión de un cambio: Resalta automáticamente todos los elementos afectados como resultado del último cambio realizado.
- Experimentación con escenarios hipotéticos: Permite hacer y deshacer cambios en las vistas, los datos y las opciones con Anulación de nivel múltiple. También se pueden deshacer acciones o conjuntos de acciones de macros, de manera que se pueden probar e invertir escenarios hipotéticos con el fin de comprender totalmente las implicaciones de cada opción al realizar cambios de alcance.
- Seguimiento y análisis flexibles de los proyectos: Ofrece el seguimiento de las principales variables exclusivas de su proyecto definiendo campos personalizados basados en fórmulas. Los indicadores gráficos pueden avisarle cuando se cumplen condiciones específicas.
- Guía de proyectos: Un asistente interactivo y pormenorizado que ayuda a definir proyectos, administrar tareas y recursos, realizar el seguimiento del estado y comunicar la información de los proyectos.

En la siguiente figura observamos la aplicación MS Project:

	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

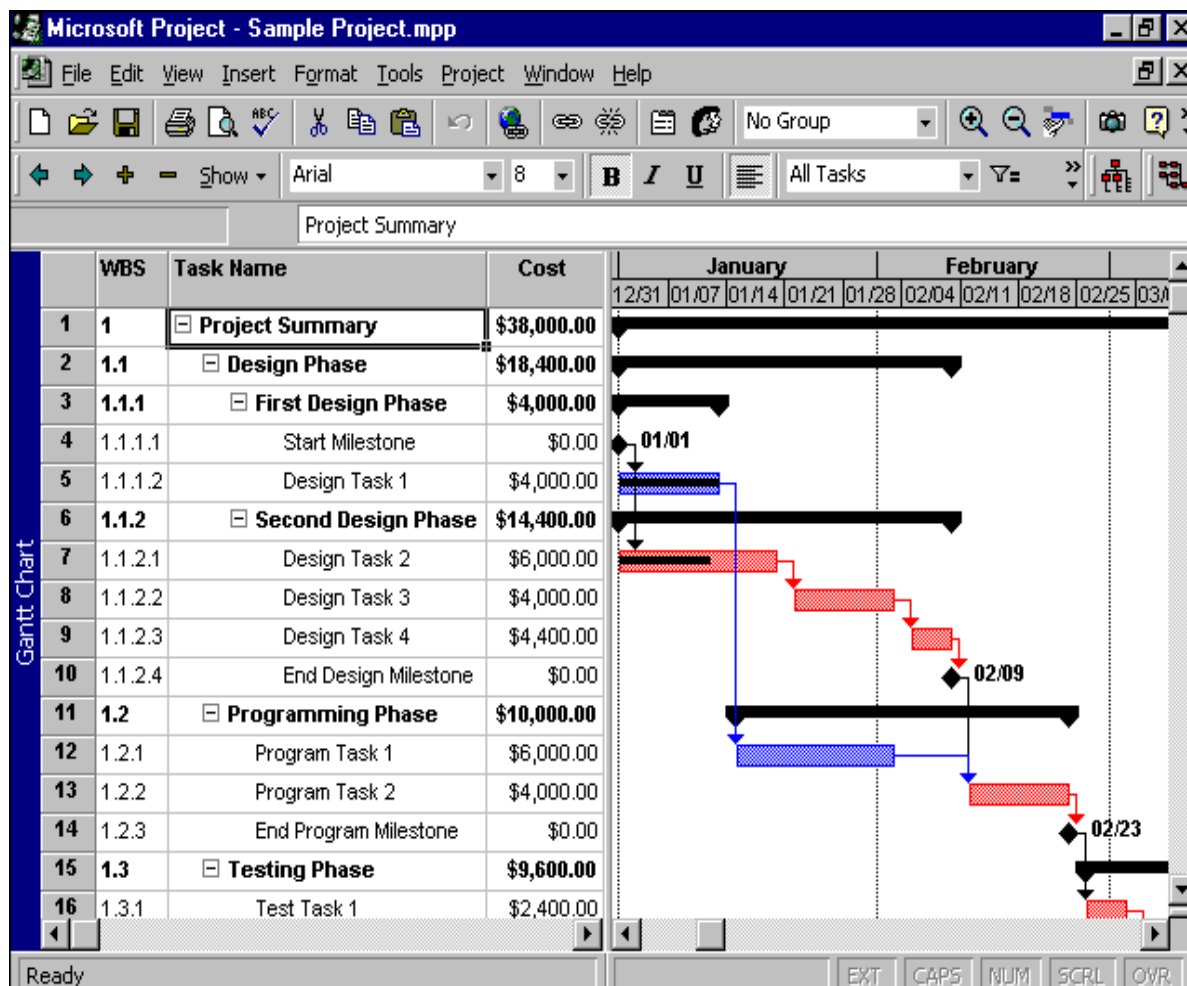



Figura 1. GUI MS Project

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

3.2.2 Gantt Project

GanttProject es software libre usado como herramienta para la gestión de proyectos, que permite realizar diagramas de GANTT a fin de planificar un proyecto y gestionar los recursos, representaciones gráficas sobre dos ejes, que nos permiten planificar todas las tareas y las actividades de un proyecto en el tiempo, facilitando una visualización amena del estado de progreso de cada actividad.

Gantt Project ha sido desarrollado en Java, tiene la capacidad de dividir los proyectos en subtareas, cada una con sus fechas de inicio y previsión de finalización, enlazadas a todas las demás actividades contiguas, con sus respectivas relaciones de dependencia, recursos adyacentes, comentarios y apuntes diversos, etc.

Para las copias de seguridad utiliza el formato XML o JPG, en documentos PDF o páginas HTML.

Las principales características de este software son las siguientes:

- La creación de tareas se puede realizar con un clic derecho desde el panel bajo la pestaña Gantt o desde el menú Tarea y los datos que se pueden incluir en la creación o modificación de las tareas son los que están asociados a las columnas que se pueden visualizar, según la explicación que se realizó sobre la pestaña Gantt.
- En el caso de los recursos es muy similar. La creación se puede realizar con un clic derecho desde el panel bajo la pestaña Recursos o desde el menú Recursos y los datos que se pueden crear o modificar son los que se mencionaron como visibles bajo la pestaña Recursos.
- La creación y configuración de tareas y recursos incluye también aspectos de presentación que pueden brindar información específica de forma ágil. De esta forma se pueden emplear colores para las tareas que deseemos agrupar, resaltar o para darle al diagrama los colores que identifiquen a una institución o temática
- Al final de tareas críticas o de grupos de tareas se suelen tener entregables o se requiere haber cumplido con objetivo principal que puede ser definido como un hito. La creación de hitos, en Gantt Project, se realiza de la misma forma en la que se crean las tareas con la configuración adicional de marcar la casilla de hito
- Los archivos que crea Gantt Project son documentos XML y gracias a la tecnología XSL se realizan transformaciones para obtener páginas HTML o archivos PDF. Cualquier aplicación que desee procesar diagramas de Gantt puede emplear los archivos generados por Gantt Project y elaborar una hoja de estilos XSL para cada tipo de transformación que requiera.
- Gantt Project ofrece la creación de diagramas Pert a partir de diagramas de Gantt que muestran la secuencia de las tareas y discriminan los hitos.
- Cuando una tarea no se cumple de acuerdo a lo planificado es importante analizar el tiempo y los recursos que se emplearon realmente, Gantt Project tiene la funcionalidad de comparar el estado actual de las tareas con estados anteriores, y podemos utilizar esta funcionalidad para ajustar la configuración de tiempo y recursos en el futuro.

A continuación podemos observar la aplicación Gantt Project:

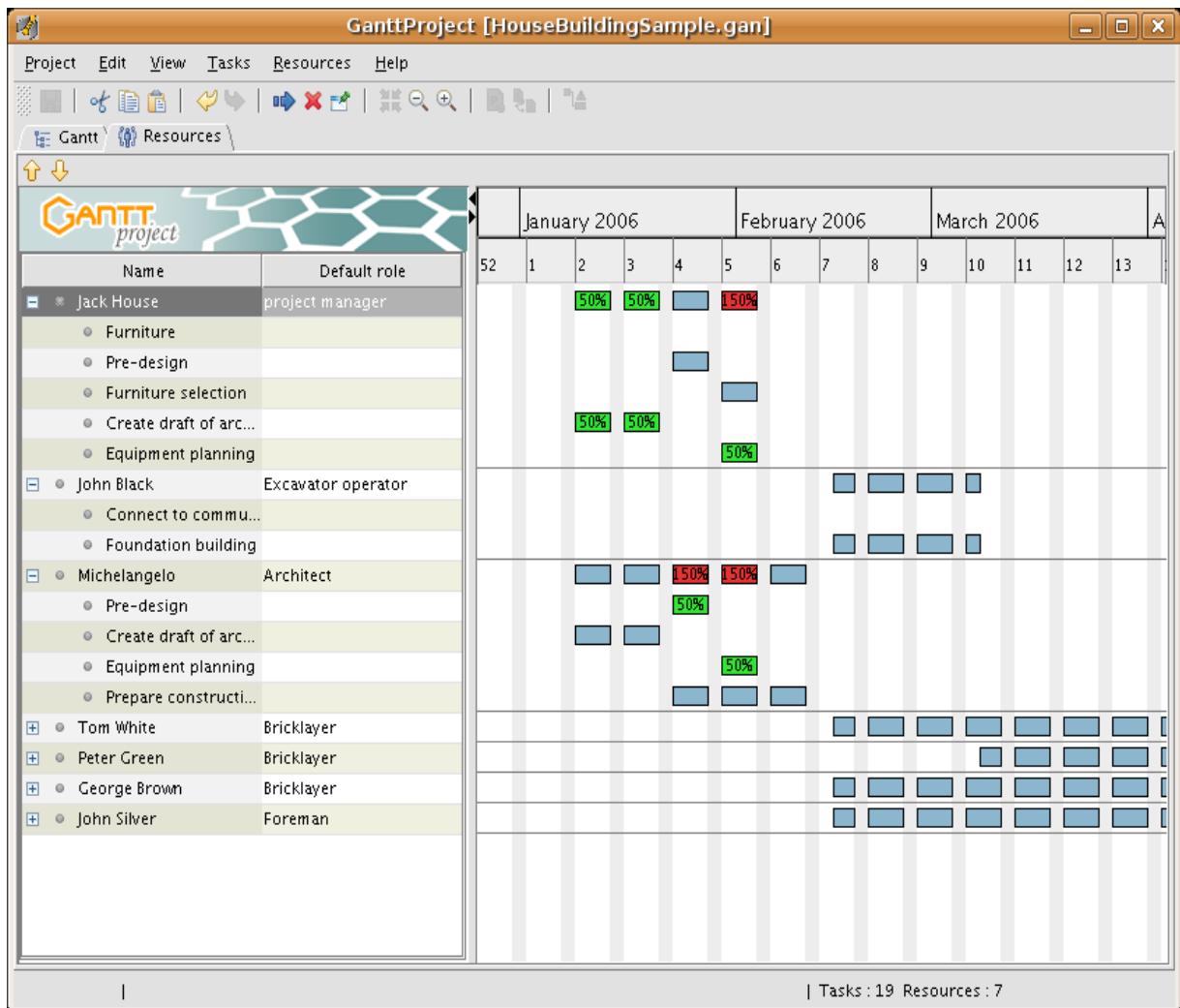



Figura 2. GUI GanttProject

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

3.2.3 Open Project

OpenProject es un software de código abierto, open source, para la gestión y control de proyectos. Supone una alternativa gratuita a Microsoft Project para la realización de gráficos de Gantt y diagramas de red PERT.

Sus funciones principales, se concentran en el diseño de graficas de Gantt, diagramas de redes PERT, y muchas otras gráficas más.

Cabe destacar que tiene dos de las herramientas más comerciales en gestión de proyectos. Por un lado las Gráficas de Gantt, que son una técnica sencilla de control que se puede emplear para preparar la producción programada y la real.

Sin embargo, se considera generalmente que estas gráficas de Gantt son inadecuadas para proyectos sofisticados. Los modelos de redes, como el método PERT, muestran las interrelaciones entre las diversas actividades. La red PERT permite determinar el itinerario crítico de forma que se pueda centrar la atención en este programa, impidiendo un retraso en el tiempo de acabado del proyecto.


OpenProject controla todos los aspectos referentes a la gestión de proyectos, como la planificación y programación, la gestión y asignación de recursos, la simulación de alternativas en procesos críticos, etc. Asimismo, proporciona la funcionalidad necesaria para trabajar con entornos multiproyecto (incluso relacionarlos).

También nos permite hacer un seguimiento de las distintas etapas de un proyecto: presupuesto, definición, estimación, etc. Cuando un proyecto se termina, existe un histórico donde se ven reflejados cada uno de los anteriores puntos.

Presenta funcionalidades avanzadas de gestión de planificación como son:

- Gestión de calendarios de trabajo
- Gestión de líneas base para la replanificación controlada del proyecto
- Función de seguimiento de proyectos mediante dedicación de recursos o simplemente por avance del porcentaje

Para ampliar su funcionalidad, existe una versión adicional de OpenProj, que incluye características adicionales. Esta versión se denomina, Project-ON-Demand, y comparte el mismo entorno de trabajo. Solo incluye módulos adicionales, que en algunos casos pueden resultar muy convenientes para un mejor manejo del proyecto.

	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

A continuación observamos la aplicación Open Project.

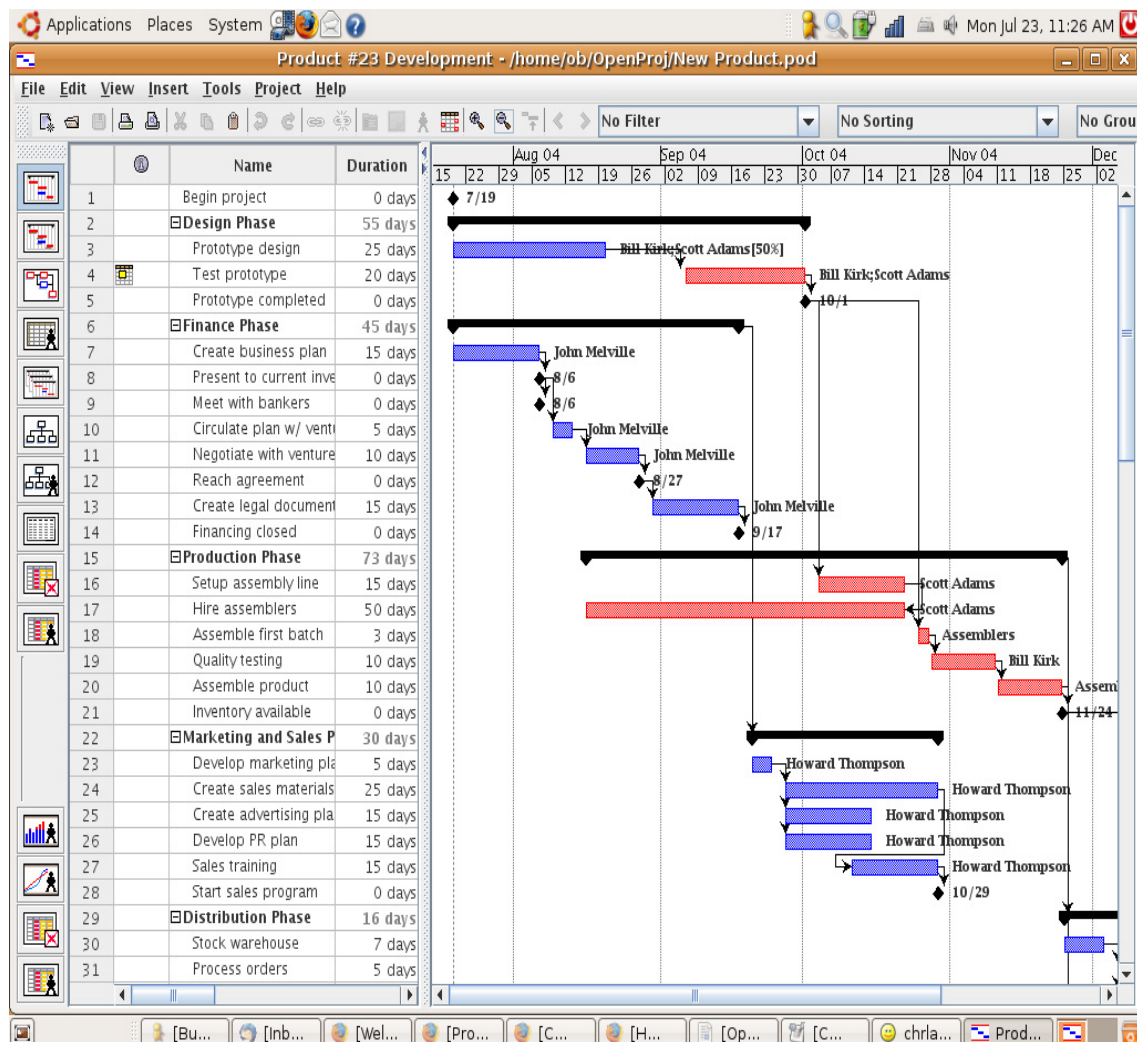



Figura 3. GUI Open Project


 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

4. CONCLUSIONES

El objetivo de este plan de mercado, era analizar el funcionamiento y prestaciones que ofrecen los diferentes SGP, solamente se ha podido analizar en mayor profundidad las aplicaciones que ofrecían una demo o que eran software libre. De todas formas el análisis realizado se considera suficiente para valorar de manera crítica las aplicaciones disponibles y las soluciones que aportan a la gestión de proyectos.

La principal utilidad que podemos destacar de este análisis es la obtención de ideas acerca de las diferentes funcionalidades (interfaz gráfico, utilidades,...) que pueden servir de modelo para determinadas partes del proyecto a desarrollar y que serán definidas en fases más avanzadas de este proyecto.

Los SGP analizados son programas independientes dedicados exclusivamente a la gestión de proyectos, y que ofrecen unas funcionalidades muy similares con pequeñas diferencias en aspectos muy particulares y con una clara orientación a los diagramas de Gantt, sin embargo el contexto de este proyecto dentro de una plataforma de gestión integral hace que su funcionalidad y orientación sean muy dependientes de las necesidades del cliente. Por este motivo el sistema a desarrollar no es una herramienta solo de gestión de proyectos como las analizadas sino que debe presentar facilidades para ser personalizado y adaptado a las necesidades de los futuros usuarios.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.0
	Estudio de mercado	Fecha: 25/09/2009
	ESTUDIOMERCADO.doc	

5. BIBLIOGRAFÍA

5.1 Referencias

[PMI, 2000] A Guide to the Project Management Body of Knowledge (PMBOK Guide) – 2000 Edition. Project Management Institute.

[MORRIS, 1981] Morris, Peter W.G. 1981. Managing Project Interfaces: Key Points for Project Success. In Cleland and King. Project Management Handbook, Second Edition. Englewood Cliffs, N.J: Prentice-Hall

[IEEE, 1993] IEEE Std 729-1993 IEEE Software Engineering Standard 729-1993: Glossary of Software Engineering Terminology. IEEE Computer Society Press, 1993

5.2 Referencias Web

[Ref. Web 1] <http://www.wikipedia.org>

[Ref. Web 2] <http://www.monografias.com/>

[Ref. Web 3] <http://www.getec.etsit.upm.es/docendia/gproyectos/gproyectos.htm>

[Ref. Web 4] <http://www.pmi.org/>

[Ref. Web 5] <http://www.csi.map.es/>


[Ref. Web 6] <http://openproj.org/>

[Ref. Web 7] <http://www.ganttproject.biz/>

Sistema para la gestión de proyectos

Modelo de negocio

Versión 1.2 – Fecha: 05/10/2009

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
28/09/2009	1.0	Versión inicial del documento modelo de negocio.	José Ángel Peribáñez
02/10/2009	1.1	Revisión del documento	Fernando Cortes
05/10/2009	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

TABLA DE CONTENIDOS

<u>1. INTRODUCCIÓN</u>	4
1.1 PROPÓSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 ACRÓNIMOS	4
1.4 DEFINICIONES	4
1.5 REFERENCIAS	4
1.6 RESUMEN	5
<u>2. INTRODUCCIÓN AL MODELO DE NEGOCIO</u>	6
<u>3. APROXIMACIÓN AL MODELO DE NEGOCIO</u>	7
<u>4. MODELO DE CASOS DE USO EN EL NEGOCIO</u>	9
4.1 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	9
4.2 CASO DE USO DEL NEGOCIO “ALTA DE PROYECTOS Y RESPONSABLES”	10
4.3 CASO DE USO DEL NEGOCIO “GESTIÓN DE PROYECTOS”	11
4.4 CASO DE USO DEL NEGOCIO “GESTIÓN DE COLABORADORES EN EL PROYECTO”	12
<u>5. BIBLIOGRAFÍA</u>	13
5.1 REFERENCIAS	13
5.2 REFERENCIAS WEB	13

	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

El presente documento analiza el modelo de negocio en el cual se define el proyecto a desarrollar, una aplicación para la gestión y dirección de proyectos (SGP). La especificación del modelo de negocio, según la metodología RUP de desarrollo software, es previa a la captura y determinación de los requisitos funcionales del sistema. Una explicación más extensa que facilita la comprensión de la naturaleza del modelo de negocio se da en el apartado 2 de este documento.

1.2 Alcance del documento

Este documento constituye la fase inicial de la metodología RUP utilizada en el desarrollo de la aplicación para la gestión y dirección de proyectos. El análisis obtenido en este documento sirve de base para las futuras fases del proyecto. Su alcance abarca el espacio temporal comprendido entre el estudio de mercado, en el cual se obtuvieron conclusiones generales acerca de las necesidades del sistema a desarrollar, y la fase de captura de requisitos funcionales.

1.3 Acrónimos

SGP: Sistema de gestión de proyectos

RUP: Rational Unified Process

UML: Unified Modeling Language


1.4 Definiciones

- UML: Lenguaje unificado de modelado de sistemas de software desarrollado por Rational y de uso muy extendido.
- RUP: Metodología de desarrollo de software, compuesta por un conjunto de fases adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM.

1.5 Referencias

En este documento aparecen referencias a los siguientes documentos del proyecto:


- ESTUDIODEMERCADO: Estudio de mercado en el cual se analiza la situación actual de las técnicas de gestión de recursos humanos, presentado diferentes alternativas existentes.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

1.6 Resumen

En el presente documento se muestra el análisis del modelo de negocio en el cual se define el proyecto a desarrollar (SGP). Se compone de 5 apartados:

- Apartado 1: Se muestra el propósito del documento y se define su alcance. Se proporciona una lista de acrónimos y definiciones útiles para la comprensión del documento, así como una lista de los documentos del proyecto referenciados y el presente resumen.
- Apartado 2. Se explica la técnica de análisis del modelo de negocio y su utilidad dentro de la metodología RUP.
- Apartado 3. Se muestra una aproximación al modelo de negocio de SGP, desde un enfoque a alto nivel que facilite la comprensión de posteriores análisis.
- Apartado 4. Se realiza el análisis del modelo de negocio y su descomposición en casos de uso, los cuales son analizados de manera individual.
- Apartado 5. Bibliografía y referencias Web utilizadas en la confección de este documento.


 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

2. INTRODUCCIÓN AL MODELO DE NEGOCIO

El modelo de negocio es una técnica de análisis, especificada en el RUP, y utilizada para comprender los procesos de negocio de la organización. El modelo de negocio “Describe los procesos de negocio de una organización en términos de casos de uso del negocio y actores del negocio, que corresponden con los procesos del negocio y los clientes respectivamente” [IGJ,2000]

El objetivo de este proceso es la captura de los casos de uso de más alto nivel del sistema. En esta descripción se mostrarán las entidades de negocio que el software debe ser capaz de soportar y gestionar, aunque siempre desde un punto de vista general (un análisis más detallado será abordado en fases posteriores del proyecto). Para ello, se desarrolla el modelo de casos de uso del negocio, el cual presenta el sistema desde la perspectiva de uso, y esquematiza como el sistema proporciona valor a sus usuarios.

En resumen, el modelo de negocio define el contexto del sistema y los casos de uso de alto nivel derivados del mismo, siendo el paso previo a la fase de análisis, en la cual se estudiará en detalle el sistema a desarrollar.

	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

3. APROXIMACIÓN AL MODELO DE NEGOCIO

Previamente al análisis del modelo de negocio mediante diagramas y casos de uso se presenta en este apartado una visión generalista del sistema de gestión de proyectos a implementar. Esta visión aproximada se ubica en el contexto de este análisis en la necesidad, para una mejor comprensión del proyecto, de un enlace entre la perspectiva global de empresa y la visión concreta y específica mediante notación UML que se realiza en los apartados siguientes. Mediante la misma se establecerán los procesos básicos de funcionamiento del proceso de negocio del proyecto, utilizando como herramienta un SGP.

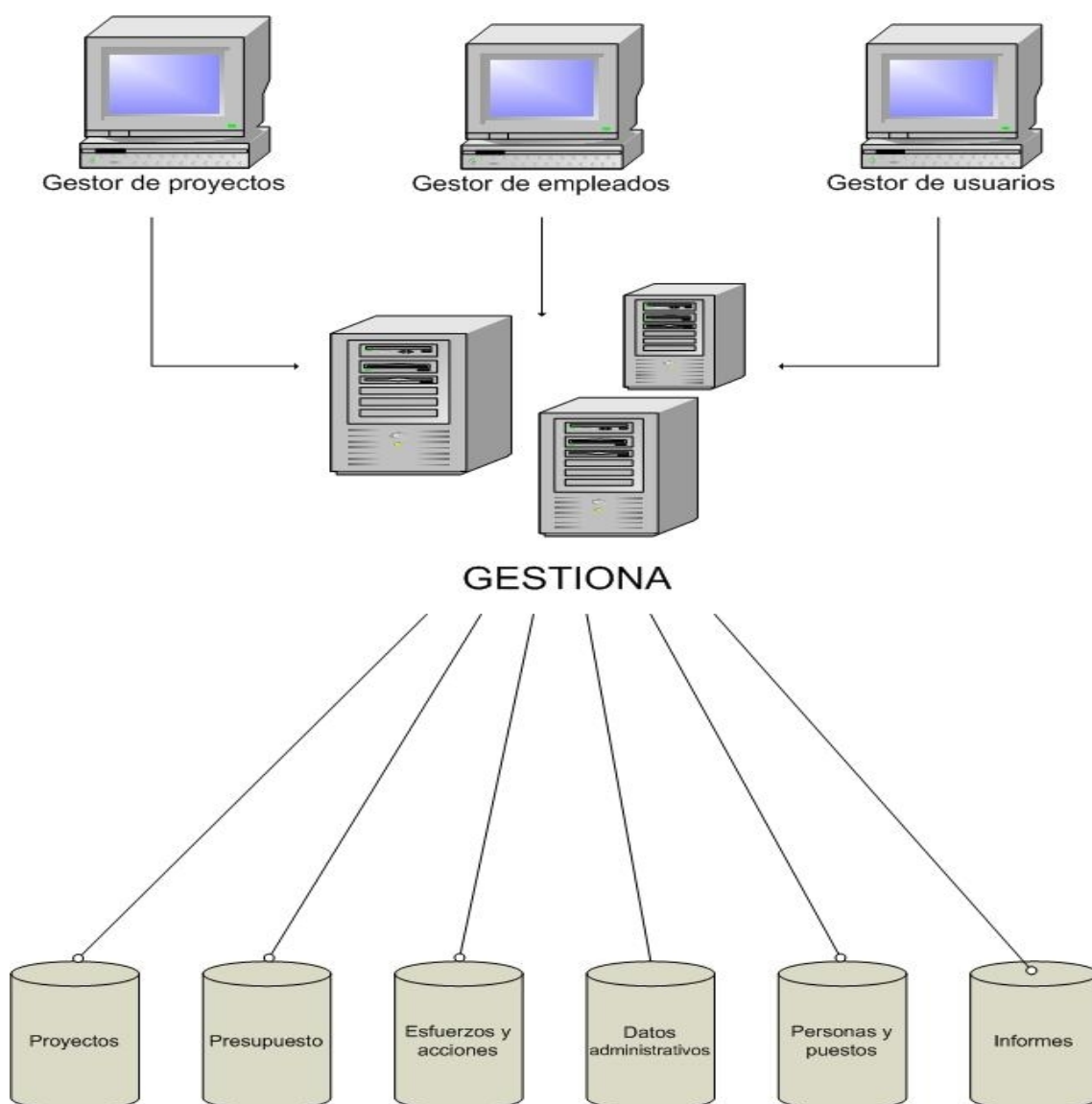




Figura 1. Modelo de negocio

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

En esta visión aproximada se puede ver como el SGP sirve como herramienta para facilitar la gestión y dirección del proyecto, sirviendo como puente entre el gestor del proyecto y toda la información, recursos, responsables... asociadas al mismo. A partir de entonces y en lo referente al contexto de ese proyecto, el responsable del proyecto se encargará mediante el SGP del control y gestión de la organización del proyecto, colaboradores, documentos y el resto de información asociada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

4. MODELO DE CASOS DE USO EN EL NEGOCIO

4.1 Diagrama de casos de uso del negocio

El modelo de casos de uso del negocio describe los procesos de negocio que tiene lugar en el SGP. El modelo de casos de uso vendrá dado por el siguiente diagrama de casos de uso:

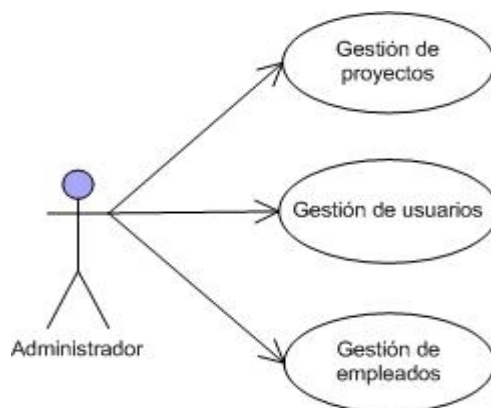



Figura 2. Caso de uso del negocio

La descripción de cada uno de los casos de uso se va a realizar siguiendo un orden lógico en el proceso de gestión del SGP. Para cada uno de ellos se realiza una descripción y se muestra, si procede, el caso de uso con un nivel más de detalle para conocer de manera más específica los procesos que tienen lugar en ese caso de uso

	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

4.2 Caso de uso del negocio “Gestión de proyectos”

A continuación mostramos el caso de uso del negocio ‘Gestión de proyectos’ con un nivel más de detalle.

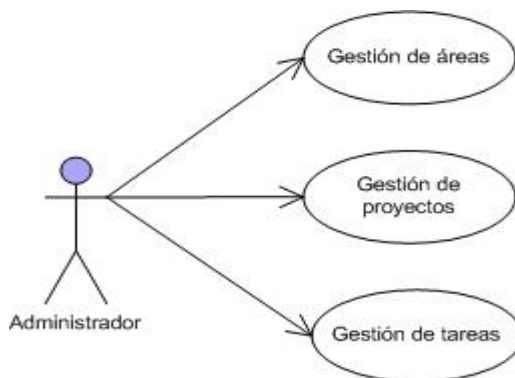



Figura 3. Caso de uso ‘Gestión de proyectos’

Dentro del modelo de negocio de la gestión de proyectos dentro de la organización, el responsable de dirección tiene la función de determinar la estructura jerárquica de proyectos, que abarcan los planes de acción estratégicos de la empresa y las acciones operativas de la misma. Asimismo tendrá la función de determinar un responsable para cada uno de estos proyectos de entre los miembros de la organización.

	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

4.3 Caso de uso del negocio “Gestión de empleados”

A continuación mostramos el caso de uso del negocio ‘Gestión de empleados’ con un nivel más de detalle

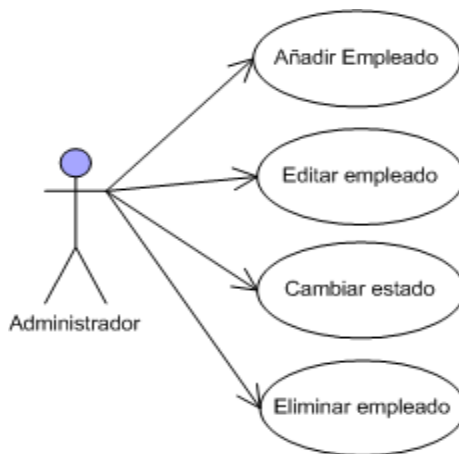



Figura 4. Caso de uso ‘Gestión de empleados’

El administrador tiene la función de configurar los empleados que aparecerán en el sistema y administrar su información y realizar la gestión de los elementos que lo componen, actividades que detallamos a continuación.

- Gestión de información de empleados: Gestionar los datos utilizados para identificar al empleado dentro de la estructura, así como la información que se asocie al mismo.
- Gestión de alta de empleados: Dar de alta nuevos empleados dentro del sistema, especificando sus datos y funciones.

	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

4.4 Caso de uso del negocio “Gestión de usuarios”

A continuación mostramos el caso de uso del negocio ‘Gestión de usuarios’ en un nivel más en detalle.

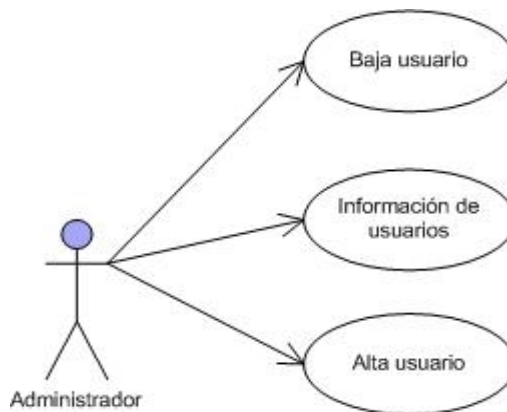



Figura 5. Caso de uso ‘Gestión de usuarios’

El administrador del sistema tiene la función de gestionar todos los usuarios dentro del mismo, estableciendo sus datos y sus privilegios. Administrar todos los datos referidos a los usuarios de la organización y realizar la gestión de los elementos que lo componen, actividades que detallamos a continuación.

- Alta de usuario: El administrador será el encargado de dar de alta nuevos usuarios asociados a empleados dentro de la aplicación, y de establecer sus permisos dentro del sistema
- Gestión de la información de usuarios: El administrador tendrá la función de gestionar la información relativa a los usuarios, principalmente establecer y eliminar permisos en la utilización del sistema.
- Baja de usuarios: El administrador tendrá la función de eliminar usuarios del sistema cuando estos dejen de ser validos o cuando estime oportuno eliminarlos.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Modelo de negocio	Fecha: 05/10/2009
	MODELONEGOCIO.doc	

5. BIBLIOGRAFÍA

5.1 Referencias

[IGJ, 2000] I. Jacobson, G. Booch, J. Rumbaugh 2000. El Proceso de Unificado de Desarrollo de Software. Pearson Education.

[OMG, 2003] Object Management Group, Inc. 2003. OMG Unified Modeling Language Specification


5.2 Referencias Web

[Ref. Web 1] <http://uml.org>

Sistema para la gestión de proyectos

Especificación de requisitos

Versión 1.2 – Fecha: 16/10/2009

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
7/10/2009	1.0	Especificación de requisitos	José Ángel Peribáñez
13/10/2009	1.1	Revisión del documento	Fernando Cortés
16/10/2009	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	4
1.1 PROPÓSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 ACRÓNIMOS	4
1.4 DEFINICIONES	4
1.5 REFERENCIAS	4
1.6 RESUMEN	4
2. DESCRIPCIÓN DEL PROCESO	5
2.1 FUNCIONES PRINCIPALES DEL PRODUCTO	5
2.2 CARACTERÍSTICAS DE LOS USUARIOS	5
3. REQUISITOS FUNCIONALES	6
3.1 GESTIÓN DE USUARIOS	6
3.2 CONFIGURACIÓN DEL SISTEMA	7
3.3 ESTRUCTURACIÓN DE ÁREAS, PROYECTOS Y TAREAS DEL SISTEMA	8
3.4 RESPONSABILIZAR LOS NIVELES DE JERARQUÍA	9
3.5 GENERACIÓN DE INFORMES	10
3.6 CONTROL DE ESFUERZOS Y ACCIONES	11
3.7 ADMINISTRACIÓN DE DOCUMENTOS	12
3.8 CONTROL DE PROYECTOS Y TAREAS	13
3.9 ADMINISTRACIÓN DE RECURSOS	14
3.10 GESTIONAR INFORMACIÓN PROYECTOS Y TAREAS	15
3.11 ASIGNACIÓN DE COLABORADORES	16
3.12 GESTIÓN DE EMPLEADOS	16
4. REQUISITOS NO FUNCIONALES	17
4.1 APLICACIÓN INTUITIVA	17
4.2 INTERACCIÓN A TRAVÉS DE GUI	17
4.3 APLICACIÓN ESCALABLE	18
4.4 POSIBILIDAD DE INTERNACIONALIZACIÓN	18
4.5 RESOLUCIONES DE PANTALLA	19
4.6 MANUAL DE AYUDA	19
5. BIBLIOGRAFÍA	20
5.1 REFERENCIAS	20
5.2 REFERENCIAS WEB	20

	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

En este documento se detallan los aspectos y especificaciones funcionales y técnicas que el sistema a desarrollar debe cumplir, por lo que debe tomarse como referencia para su realización. Contiene una descripción de los requisitos del sistema lo más específica posible para poder cumplirlos adecuadamente.

1.2 Alcance del documento

El alcance del documento, comprende todo el periodo temporal del proyecto, ya que sus contenidos pueden ser modificados en cualquier momento conforme las especificaciones y necesidades del sistema cambien en el tiempo.

1.3 Acrónimos

SGP: Sistema de gestión de proyectos

GUI: Graphical User Interface (Interfaz gráfico de usuario)

ILC: Interfaz de línea de comandos

1.4 Definiciones

- Interfaz gráfico de usuario: Método para facilitar la interacción del usuario con el ordenador a través de la utilización de un conjunto de imágenes y objetos pictóricos, además de texto.


1.5 Referencias

En este documento no se han realizado referencias a otros documentos del proyecto.

1.6 Resumen

Este documento contiene los principales requisitos que debe cumplir el sistema a desarrollar. Se compone de 5 secciones:

- Apartado 1: Muestra el propósito del documento y se define su alcance.
- Apartado 2: Descripción general de la funcionalidad del producto y naturaleza de sus usuarios.
- Apartado 3: Relación de los requisitos funcionales especificando su naturaleza, criticidad, dificultades técnicas y riesgos.
- Apartado 4: Descripción de los requisitos no funcionales del sistema.
- Apartado 5: Bibliografía y referencias Web utilizadas en la confección de este documento.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

2. DESCRIPCIÓN DEL PROCESO


2.1 Funciones principales del producto

EL SGP es una herramienta desarrollada mediante la utilización de tecnología .NET que permite la gestión de los diferentes recursos que intervienen en el desarrollo de un proyecto. Mediante el SGP se realiza la gestión de los diferentes proyectos de la empresa y para cada uno de ellos se integra de manera concurrente las acciones de dirección, definición, y seguimiento de proyectos por parte de los responsables y la actualización y seguimiento de sus propias tareas.

2.2 Características de los usuarios

Los roles de usuarios que pueden interactuar con ese sistema, son los siguientes:


- Administrador: Se encarga de la gestión de la estructura del plan de acciones (áreas, proyectos y tareas), de la configuración del sistema y la gestión de empleados y usuarios.
- Responsable de área: Responsable de los procesos y proyectos involucrados dentro de un área de trabajo
- Responsable de proyecto: Responsable de los procesos y tareas involucrados dentro de un proyecto.
- Responsable de tarea: Responsable de los procesos y tareas involucrados dentro de una tarea
- Colaboradores: Responsable de algún aspecto dentro de una tarea/proyecto, debe trasladar su evolución a los responsables superiores

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

3. REQUISITOS FUNCIONALES


3.1 Gestión de usuarios

Nombre	REQ-F01.- Gestión de usuarios
Descripción	<p>Se debe gestionar el acceso y los permisos de los usuarios dentro del SGP. Cada usuario dispondrá de unas funcionalidades distintas en función de sus permisos. Es necesario especificar:</p> <ul style="list-style-type: none"> • Categoría de usuario y funcionalidades asociadas a cada categoría. • Archivo de la lista de usuarios y seguridad asociada al mismo. • Responsable de la gestión de lista de usuarios.
Criticidad	Alta
Implicaciones Técnicas	<p>Se debe especificar de manera estricta el sistema físico de almacenaje en que se va ubicar la lista de usuarios y los procesos de seguridad para mantener integridad del mismo.</p> <p>Debe especificarse un sistema de privilegios que permita clasificar sin ambigüedades a los usuarios de SGP y a partir de su clasificación controlar las funcionalidades asignadas a cada uno.</p>
Coste y planificación	Medio
Riesgos	<p><u>Altos</u></p> <p>No guardar de manera óptima la lista de usuarios.</p> <p>No especificar de manera clara el sistema de privilegios</p>
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.2 Configuración del sistema

Nombre	REQ-F02.- Configuración del sistema
Descripción	<p>El SGP permitirá la parametrización de distintos conceptos para personalizar el uso de la aplicación. Estos conceptos serán:</p> <ul style="list-style-type: none"> • Tipos de proyecto. • Indicadores económicos. • Indicadores no económicos • Partidas presupuestaría. • Centros de trabajo • Centros de coste • Nacionalidades • Idiomas • Etc.
Criticidad	Alta
Implicaciones Técnicas	Se debe especificar de manera estricta el sistema físico de almacenaje en que se va ubicar la lista de parámetros configurables.
Coste y planificación	Medio
Riesgos	<p><u>Altos</u></p> <p>No guardar de manera óptima la parametrización de los ítems.</p>
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.3 Estructuración de áreas, proyectos y tareas del sistema

Nombre	REQ-F03.- Estructuración de áreas, proyectos y tareas
Descripción	<p>El SGP debe permitir definir la estructura de las diferentes áreas organizativas, de los proyectos a realizar dentro de las mismas y de las tareas que conforman los proyectos. Es necesario definir las informaciones asociadas a los diferentes niveles de esta jerarquía. Las acciones sobre esta estructura en cada nivel estarán permitidas a los usuarios con determinados privilegios.</p> <p>La organización de la estructura jerárquica de planes de acción queda determinada por las siguientes reglas:</p> <ul style="list-style-type: none"> • La entidad de más alto nivel dentro de la estructura de proyectos es el área. Un área puede contener otras áreas por debajo. • La siguiente entidad en importancia en la jerarquía es el proyecto. Un proyecto siempre estará en un estado definido y solamente puede contener tareas como entidades dependientes. • La entidad de más bajo nivel es la tarea. Una tarea siempre será dependiente de un proyecto y podrá contener tareas como entidades dependientes.
Criticidad	Alta
Implicaciones Técnicas	Se debe especificar la estructura lógica en la que se va a almacenar la organización del sistema, la información asociada a la misma y las operaciones que sobre ella van a poder realizar los usuarios autorizados.
Coste y planificación	--
Riesgos	<p><u>Medio</u></p> <p>No definir una estructura óptima, flexible y de manejo eficiente puede llevar a degradar las prestaciones del sistema.</p>
Dependencias	--
Comentarios	La estructuración de proyectos y tareas dependerá en buena medida del GUI a desarrollar y la organización de los procesos de trabajo sobre el proyecto por parte del usuario.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.4 Responsabilizar los niveles de jerarquía

Nombre	REQ-F04.- Determinar responsables para cada nivel de jerarquía
Descripción	Dentro de cada uno de los niveles de la jerarquía del sistema (área, proyecto, tarea) el responsable de cada uno de ellos debe poder asignar un responsable a los niveles inferiores e independientes del que él gestiona. Dependiendo de esta asignación se definirán los recursos y privilegios del responsable asignado.
Criticidad	Alta
Implicaciones Técnicas	Esta asignación será dependiente del sistema de permisos y usuarios definidos en el REQ-01. Se deberá permitir al usuario responsable de un nivel la visualización de los planes de acción dependientes de ese nivel pero no de los superiores, teniendo en cuenta que en la mayoría de los casos habrá usuarios del sistema que sean responsables de un plan de acción aislado de los niveles superiores de la jerarquía (por ejemplo una tarea) con lo que solo deberá poder ver los datos de esa tarea, y no aquellas de las que dependa.
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.5 Generación de informes

Nombre	REQ-F05.- Generar informes y estadísticas
Descripción	<p>El SGP debe facilitar la elaboración de informes y estadísticas que proporcionen información sobre el estado y evolución de los procesos tanto del SGP como de los proyectos gestionados mediante él. Dentro de este concepto de informes incluimos tanto la documentación típica de listados y relaciones de información como gráficos y tablas estadísticas.</p> <p>Es necesario especificar la naturaleza de los informes que se pueden generar dependiendo del rol de los usuarios, así como facilitar la inclusión de futuros nuevos tipos de informes.</p>
Criticidad	Medio
Implicaciones Técnicas	<p>Se debe especificar los tipos y formatos de informes a generar y exportar y los métodos de creación de los mismos (desde filtros, automáticos, definidos por usuario etc....).</p> <p>La creación de estos informes será dependiente de las posibilidades tecnológicas actuales del sistema de desarrollo.</p>
Coste y planificación	--
Riesgos	Medio
Dependencias	Los tipos de informes a generar dependen de la especificación de usuarios comentada en el REQ-01
Comentarios	El formato y naturaleza de los informes será muy dependiente de las futuras personalizaciones para los clientes, con lo que se debe crear una estructura de generación de informes lo más flexible posible.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.6 Control de esfuerzos y acciones

Nombre	REQ-F06.- Control de esfuerzos y acciones
Descripción	El SGP debe permitir controlar los esfuerzos invertidos por parte de los usuarios en las diferentes acciones involucradas en el proyecto. Se debe proveer de un sistema que permita a los responsables observar esta información en los distintos niveles de la jerarquía.
Criticidad	Medio
Implicaciones Técnicas	--
Coste y planificación	Alto
Riesgos	<u>Medio</u> Se debe desarrollar un sistema de filtrado y visualización que permita realizar consultas y resúmenes de esfuerzos de empleados.
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.7 Administración de documentos

Nombre	REQ-F07.- Administración de documentos
Descripción	El SGP debe permitir gestionar y administrar documentos asociados a los diferentes niveles del sistema (proyectos y tareas).
Criticidad	Medio
Implicaciones Técnicas	<p>Se debe especificar el sistema de gestión de documentos y la estructura lógica en que se almacenará, así como la manera de asociar esta información a los diferentes niveles de la jerarquía.</p> <p>Asimismo deberán establecerse las limitaciones de tipo y tamaño de ficheros a almacenar de manera fácilmente parametrizable, ya que previsiblemente esto será variable en diferentes instalaciones en clientes.</p>
Coste y planificación	--
Riesgos	<p><u>Medio</u></p> <p>Deberá definirse una política de actuación respecto a cambios en documentos, permisos de modificación.</p>
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.8 Control de proyectos y tareas

Nombre	REQ-F08.- Control de la evolución de proyectos y tareas
Descripción	El SGP debe servir como herramienta de seguimiento y gestión de la evolución de los proyectos y tareas. A partir de la información de fechas especificada por el REQ-07 y de la información de seguimiento asociada introducida por los usuarios del sistema, el SGP debe proveer métodos que permitan la visualización de la situación histórica y actual de los procesos del sistema.
Criticidad	Media
Implicaciones Técnicas	Deben especificarse las interfaces que trasladen la información de seguimiento al usuario.
Coste y planificación	Alto
Riesgos	<u>Medio</u> Es necesario definir interfaces que aúnen claridad y sencillez de visualización con rigurosidad y alto nivel de información.
Dependencias	Se debe utilizar la información y datos especificados por el REQ-05
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	


3.9 Administración de recursos

Nombre	REQ-F09.- Administración de recursos
Descripción	El SGP debe proveer de métodos que permitan administrar los recursos asignados a las diferentes tareas que componen el proyecto.
Criticidad	Media
Implicaciones Técnicas	Se debe especificar el modo en que se van a representar los recursos asignados así como las operaciones que se van a poder realizar con ellos.
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

3.10 Gestionar información proyectos y tareas

Nombre	REQ-F10.- Gestionar información asociada a proyectos y tareas
Descripción	Se debe permitir definir información genérica asociada a los proyectos y tareas para que el usuario pueda personalizar, explicar y definir de manera clara elementos informativos relacionados con el proyecto o tarea.
Criticidad	Bajo
Implicaciones Técnicas	Será necesario desarrollar o estudiar un sistema que permita editar y guardar la información relacionada con un proyecto utilizando formato.
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--


 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

3.11 Asignación de colaboradores

Nombre	REQ-F11.- Asignación de colaboradores
Descripción	Asociados a un proyecto o tarea se podrán definir colaboradores que serán responsables de alguna actividad.
Criticidad	Media
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	Será necesario desarrollar un sistema que permita interactuar y comunicarse a los usuarios involucrados en los proyectos (responsables y colaboradores) de manera sencilla y eficaz, así como asignación de tareas.

3.12 Gestión de empleados

Nombre	REQ-F12.-Gestión de empleados
Descripción	El SGP permitirá dar de alta de nuevos empleados, cambiar su estado, eliminar empleados
Criticidad	Alta
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

4. REQUISITOS NO FUNCIONALES


Además de los requisitos funcionales descritos en la sección anterior, el sistema debe cumplir los siguientes requisitos no funcionales:

4.1 Aplicación intuitiva

Nombre	REQ-NF01.- Aplicación intuitiva
Descripción	La aplicación deberá ser intuitiva y fácil de usar, incluso por personas poco familiarizadas con las nuevas tecnologías.
Criticidad	Alto
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

4.2 Interacción a través de GUI

Nombre	REQ-NF02.- Interacción con la aplicación a través de GUI
Descripción	La interacción con el SGP será a través de GUI (y no a través de línea de comandos)
Criticidad	Media
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--


 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

4.3 Aplicación escalable

Nombre	REQ-NF03.- Aplicación escalable
Descripción	El SGP será fácilmente escalable. Se podrán añadir nuevos módulos y plug-ins con facilidad.
Criticidad	Media
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

4.4 Posibilidad de internacionalización

Nombre	REQ-NF04.- Posibilidad de internacionalización
Descripción	El SGP debe poder ser fácilmente adaptable a diferentes culturas, tanto en el idioma de los textos como cualquier otro elemento que dependa de ellas, como formatos de fechas, monedas o decimales.
Criticidad	Alto
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--


 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

4.5 Resoluciones de pantalla

Nombre	REQ-NF05.- Resoluciones de pantalla soportadas
Descripción	El SGP deberá ser visualizado correctamente para resoluciones de pantalla superiores o iguales a 1024 x 768.
Criticidad	Alto
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

4.6 Manual de ayuda

Nombre	REQ-NF06.- Manual de ayuda
Descripción	El SGP contará con un manual de ayuda que guíe al usuario acerca de las características básicas del sistema y su modo de uso.
Criticidad	Bajo
Implicaciones Técnicas	--
Coste y planificación	--
Riesgos	--
Dependencias	--
Comentarios	--

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Especificación de requisitos	Fecha: 16/10/2009
	ESPECIFICACIONREQUISITOS.doc	

5. BIBLIOGRAFÍA

5.1 Referencias

[IGJ, 2000] - I. Jacobson, G. Booch, J. Rumbaugh. 2000. El Proceso Unificado de Desarrollo de Software. Pearson Education


5.2 Referencias web

[Ref. Web 1] <http://www.wikipedia.org>

Sistema para la gestión de proyectos

Análisis

Versión 1.3 – Fecha: 25/11/2009

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
23/10/2009	1.0	Documento de análisis	José Ángel Peribáñez
01/11/2009	1.1	Modificación del documento	José Ángel Peribáñez
19/11/2009	1.2	Revisión del documento	Fernando Cortés
25/11/2009	1.3	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España




 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	5
1.1 PROPÓSITO DEL DOCUMENTO	5
1.2 ALCANCE DEL DOCUMENTO	5
1.3 ACRÓNIMOS	5
1.4 DEFINICIONES	5
1.5 REFERENCIAS	5
1.6 RESUMEN	6
2. DESCRIPCIÓN DEL PROCESO	7
3. ANÁLISIS DE LOS CASOS DE USO	8
3.1 INTRODUCCIÓN	8
3.2 ACTORES DE LOS CASOS DE USO	8
3.2.1 ADMINISTRADOR	9
3.2.2 RESPONSABLE DE ÁREA	9
3.2.3 RESPONSABLE DE PROYECTO	9
3.2.4 RESPONSABLE DE TAREA	10
3.2.5 COLABORADOR	10
3.3 CASOS DE USO	10
3.3.1 ESQUEMA GENERAL DE LOS CASOS DE USO	10
3.3.2 CASO DE USO DE ÁREA	12
3.3.3 CASO DE USO DE PROYECTO	19
3.3.4 CASOS DE USO DE TAREA	27
3.3.5 CASOS DE USO DEL SISTEMA	30
3.3.6 CASOS DE USO DE PERSONAS	35
3.3.7 CASOS DE USO DE USUARIOS	41
4. ANÁLISIS DE PAQUETES	48
4.1 INTRODUCCIÓN	48
4.2 IDENTIFICACIÓN DE PAQUETES DE ANÁLISIS	48
4.2.1 PAQUETES DE ANÁLISIS ‘GESTIÓN DE ESTRUCTURA DE PLANES DE ACCIÓN’	48
4.2.2 PAQUETES DE ANÁLISIS ‘GESTIÓN INFORMACIÓN DE ÁREA’	49
4.2.3 PAQUETES DE ANÁLISIS ‘GESTIÓN INFORMACIÓN DE PROYECTO’	50
4.2.4 PAQUETES DE ANÁLISIS ‘GESTIÓN INFORMACIÓN DE TAREA’	50
4.2.5 PAQUETES DE ANÁLISIS ‘GESTIÓN DE INFORMES’	51
4.2.6 PAQUETES DE ANÁLISIS ‘GESTIÓN DE INCIDENCIAS’	51
4.2.7 PAQUETES DE ANÁLISIS ‘ACCESO A REPOSITORIO DE PERSISTENCIA DE DATOS’	51
4.2.8 PAQUETES DE ANÁLISIS ‘INTERNALIZACIÓN DE ENTIDADES VISIBLES’	52
4.2.9 PAQUETES DE ANÁLISIS ‘GESTIÓN DE PERSONAS’	52
4.2.10 PAQUETES DE ANÁLISIS ‘GESTIÓN DE USUARIOS’	53
5. REQUERIMIENTOS ESPECIALES	54

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

5.1	PERSISTENCIA	54
5.2	TOLERANCIA A FALLOS	54
5.3	INTERNACIONALIZACIÓN	54
6.	<u>BIBLIOGRAFÍA</u>	<u>55</u>
6.1	REFERENCIA	55
6.2	REFERENCIAS WEB	55

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

1. INTRODUCCIÓN

1.1 Propósito del documento

El propósito del presente documento es describir la fase de análisis del proyecto de desarrollo de un SGP, según los requisitos del documento de especificación de requisitos.

En el análisis podemos razonar más sobre los aspectos internos, y por tanto resolver aspectos relativos a la interferencia de casos de uso. Se analiza la arquitectura del sistema, definiendo casos de uso, estructura de paquetes y clases identificadas.

1.2 Alcance del documento

El Alcance del documento comprende todas las fases de análisis del proyecto.

1.3 Acrónimos

SGP: Sistema de gestión de proyectos.

UML: Unified Modeling Language.

GUI: Graphic user interface

BD: Base de datos


1.4 Definiciones

- Caso de uso: Técnica para describir la interacción entre un usuario del sistema y éste.
- Diagrama de casos de uso: Notación gráfica que describe un caso de uso.
- Diagrama de actividades: Notación gráfica que representa los flujos de trabajo paso a paso de los componentes en un sistema. Muestra el flujo de control general del sistema.
- Diagrama de secuencia: Notación gráfica que muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo
- UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema desarrollado por Rational de uso extendido.

1.5 Referencias

En este documento se realizan referencias a los siguientes documentos del proyecto:


- ESPECIFICACIONREQUISITOS.doc: Documento en el que se especifican los requisitos del sistema.
- MODELONEGOCIO.doc: Documento en el que se especifica el modelo de casos de uso del negocio

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

1.6 Resumen

El presente documento describe el análisis del proyecto del sistema para la gestión de proyectos. Se compone de los siguientes apartados:

- Apartado 1: Muestra el propósito del documento y se define su alcance.
- Apartado 2: Describe el proceso de análisis llevado a cabo para la realización de este documento.
- Apartado 3: Contiene la descripción de los casos de uso.
- Apartado 4: Identifica las entidades y los paquetes de análisis.
- Apartado 5: Identifica requerimientos especiales.
- Apartado 6: Bibliografía y referencias web utilizadas en la confección de este documento.

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	


2. DESCRIPCIÓN DEL PROCESO

El flujo de trabajo del análisis se compone de distintas actividades, como son: análisis de los casos de uso, análisis de paquetes y análisis de requisitos especiales comunes.

El análisis de casos de uso tiene como objetivo identificar todos los procesos que tiene lugar en el sistema, y los actores que forman parte de los mismos.

En análisis de paquetes pretende identificar los componentes esenciales del sistema, garantizando que mediante los mismos se cumplen los objetivos definidos en el modelo de negocio y en el documento de requisitos, así como en la descripción de los casos de uso, y describir las dependencias entre paquetes, de forma que pueda estimarse el efecto de cambios futuros.

El análisis de los requisitos especiales comunes tiene el objetivo de identificar las peculiaridades del sistema que determinaran restricciones y especificaciones para fases futuras del proyecto.

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3. ANÁLISIS DE LOS CASOS DE USO

3.1 Introducción

En este apartado se va a realizar el análisis de casos de uso del sistema a partir del estudio realizado en el documento de modelo de negocio. En esta fase se identificaran todos los procesos que tienen lugar en el sistema y no solo los de alto nivel como el modelo de negocio.

Para formalizar el detalle de los casos de uso se van utilizar las siguientes técnicas:


- Diagramas de casos de uso: Describen la relación entre los usuarios del sistema y los casos de uso.
- Diagramas de secuencia: Describen las interacciones existentes a lo largo del tiempo.
- Diagrama de actividades: Describe el flujo de trabajo de los casos de uso.

El conjunto de casos de uso obtenidos, van a ser agrupados, según su naturaleza en subgrupos. Dependiendo, sobre qué elemento incide cada una de las funcionalidades representadas por cada caso de uso. Los grupos establecidos son:

- Usuario: Casos de uso relacionados con la gestión y administración de los usuarios del sistema.
- Área: Casos de uso relacionados con la gestión de la información de área.
- Proyecto/tarea: Casos de uso relacionados con la gestión de la información de proyectos y tareas.
- Sistema: Casos de uso relacionados con la administración del sistema
- Empleados: Casos de usos relacionados con le gestión y administración de personas en el sistema.

3.2 Actores de los casos de uso

Los actores de los casos de uso, son los usuarios que interaccionan con el sistema. Estos usuarios se identifican a partir del documento de especificación de requisitos, y representan los diferentes roles de los usuarios en el sistema.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.2.4 Responsable de tarea



Figura 4. Responsable de tarea

El responsable de tarea representa el rol de encargado de una tarea dentro de la estructura jerárquica de la organización del SGP. Se encarga de mantener la estructura de tareas que dependen de su tarea si existen y de nombrar responsables de las mismas, asimismo debe asignar colaboradores a su tarea y gestionar toda la información, recursos, evolución, y planificación de la misma.

3.2.5 Colaborador



Figura 5. Colaborador

El colaborador representa el rol de quien tiene asignada alguna función dentro de un proyecto o tarea. Se encarga de actualizar la información de su responsabilidad para que esta sea controlada por los responsables.

3.3 Casos de uso

En este apartado se especifican mediante una descripción textual y los diagramas explicativos comentados los diferentes casos de uso del SGP. Debido a que varias de las funcionalidades asociadas a proyectos y tareas son muy parecidas se detalla un caso de uso que explique esta funcionalidad y se comentaran las variaciones nominales que existan en los casos de uso similares.

3.3.1 Esquema general de los casos de uso

En este apartado se muestra una visión general de los casos de uso que pueden realizar cada uno de los actores del sistema. En el resto de apartados serán analizados en detalle los casos de uso del sistema separados por nivel jerárquico.

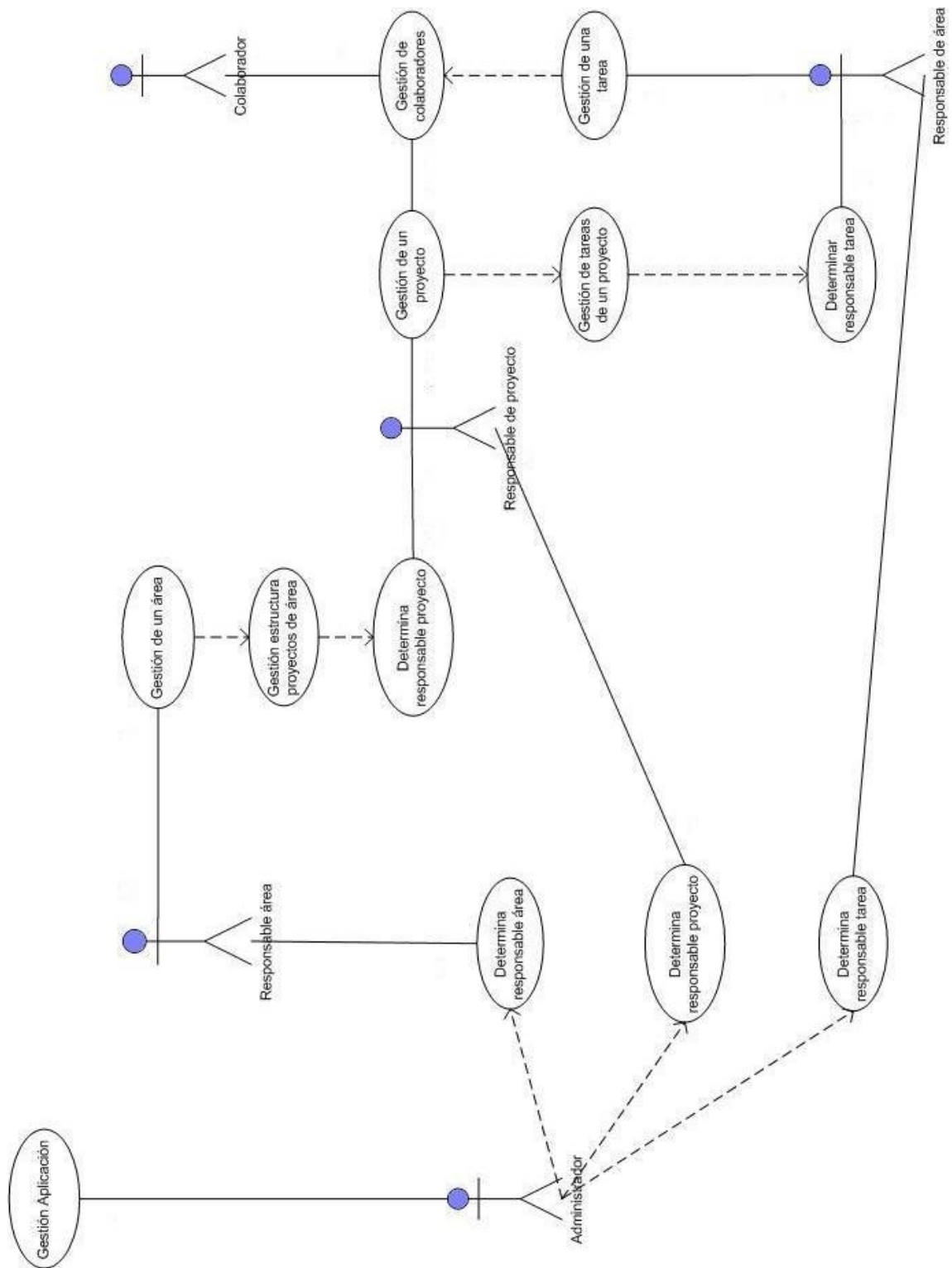



Figura 6.- Esquema general casos de uso

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

En el diagrama se observa como el actor principal es el administrador, que se encarga de la gestión y administración de la aplicación y del nombramiento de las responsabilidades dentro de las áreas, proyectos y tareas.

Para los niveles siguientes en la jerarquía de la estructura de áreas, proyectos y tareas, y consecuentemente para los actores involucrados, la funcionalidad de casos de uso es equivalente. El responsable del nivel más alto se encarga de su gestión, lo que incluye la gestión de la estructura de áreas dependientes de él y la determinación de los responsables de los mismos. En el caso de proyectos y las tareas aparece el actor *colaborador*. Los responsables de cada proyecto o tarea se encargarán de gestionar las relaciones con sus colaboradores en el plan de acción.

3.3.2 Caso de uso de área

Los casos de uso especificados en este apartado son los relacionados con la gestión de las áreas en el SGP. El área es la entidad de mayor nivel dentro de la jerarquía del SGP y de ella dependen proyectos que tengan un campo de acción o fines similares. Toda área tiene asignada un responsable de la misma que se encarga de la administración de la información asociada al área y de la creación y control de los proyectos dependientes de la misma. Este responsable es nombrado por el administrador. Los casos de uso de área identificados en este apartado se muestran a continuación.

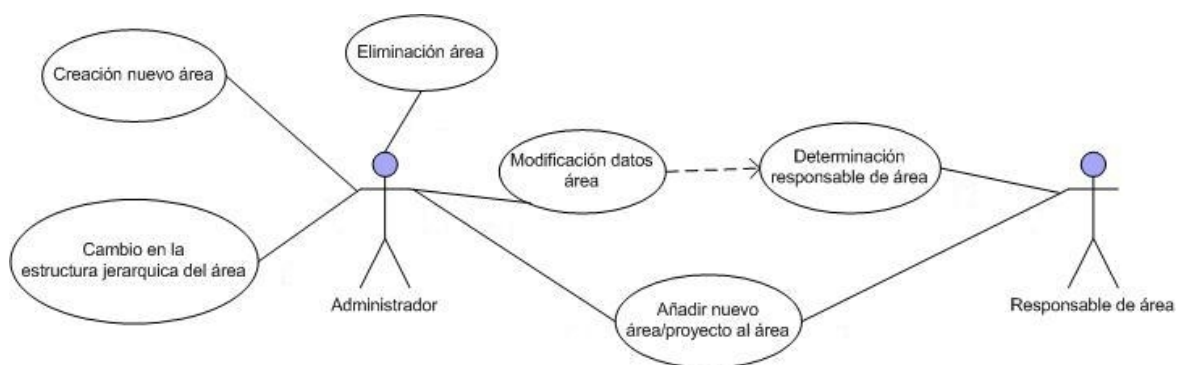



Figura 6. Caso uso general nivel 1 SGP

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.2.1 Caso de uso 'creación de un nuevo área'

En este caso de uso se define la funcionalidad por parte del administrador de crear un área dentro de la estructura de planes de acción mediante el SGP. El administrador interactúa con el GUI de nueva área del SGP y mediante un formulario introduce los datos de entrada de área que se encarga de que los datos introducidos sean correctos. El control de gestión de área se encarga de la creación y almacenaje de la nueva entidad o del aviso en caso de error. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso

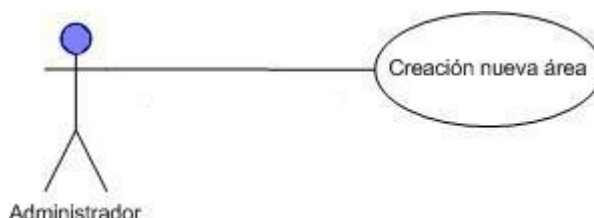


Figura 7. Caso uso 'creación de un nuevo área'

- Diagrama de secuencia

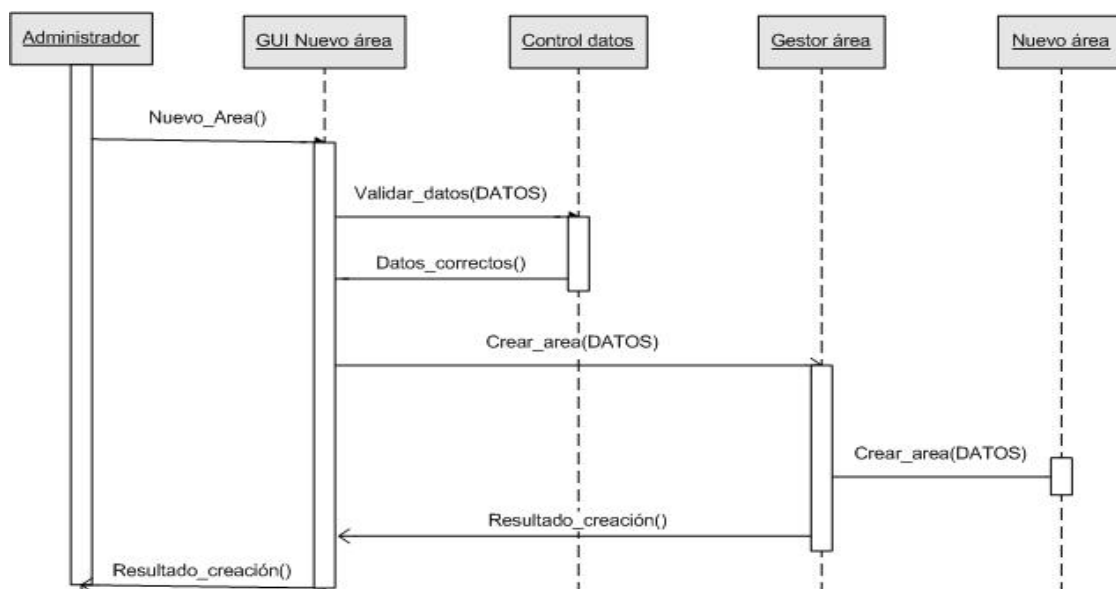



Figura 8. Diagrama de secuencia del caso de uso 'creación de un nuevo área'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividades

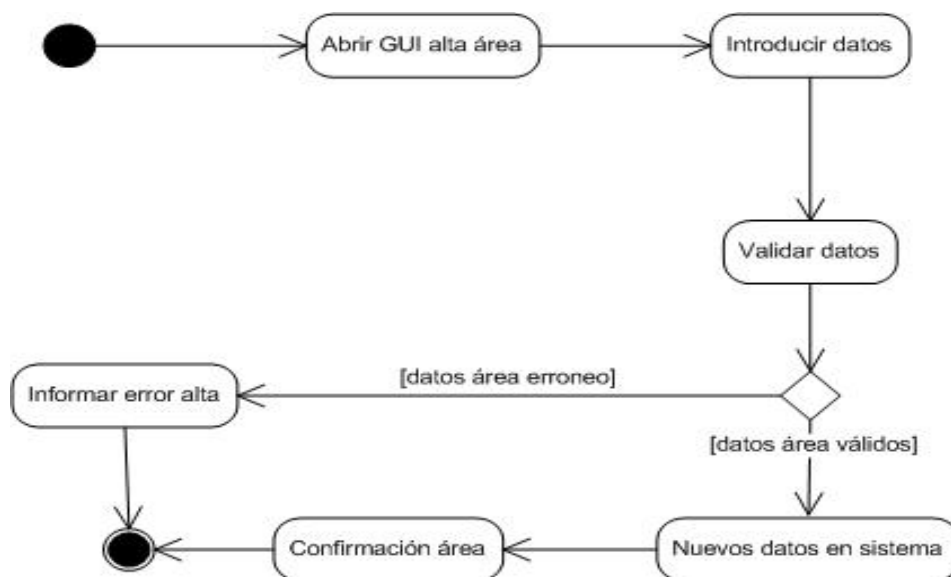


Figura 9. Diagrama de actividades caso de uso 'creación de un nuevo área'


3.3.2.2 Caso de uso 'eliminación de un área'

El caso de uso eliminar área describe la funcionalidad por parte del administrador de eliminar un área de la estructura de áreas del SGP. Asimismo el responsable del área tiene también la posibilidad de eliminarla. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso



Figura 10. Caso de uso 'eliminación de un área'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

– Diagrama de secuencia

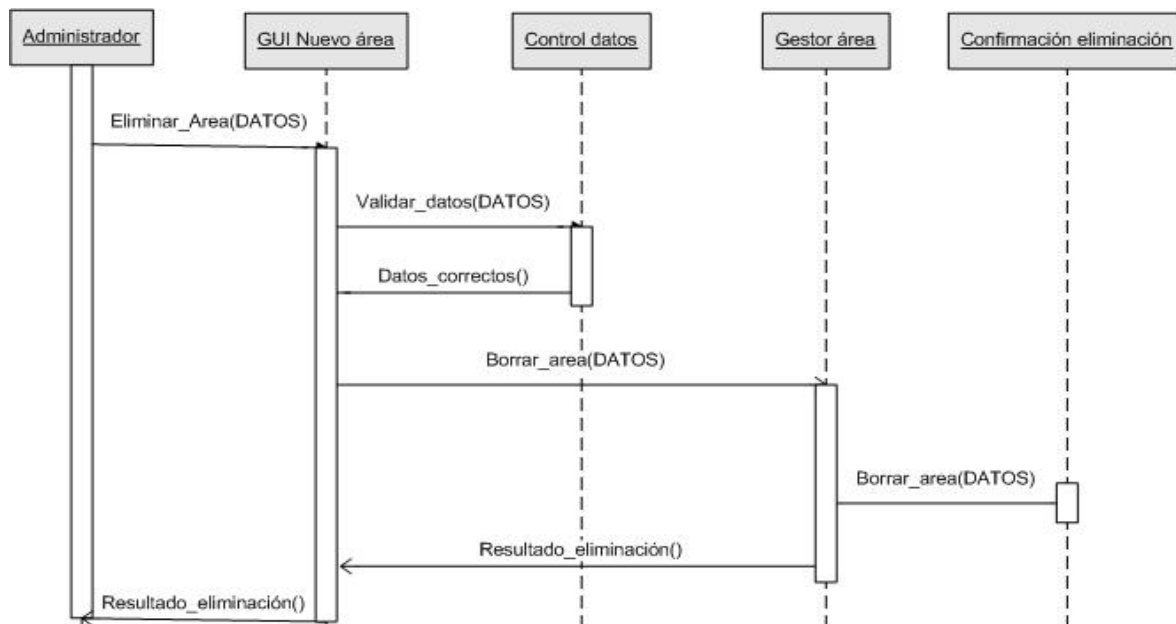


Figura 11. Diagrama de secuencia caso uso 'eliminación de un área'

– Diagrama de actividades

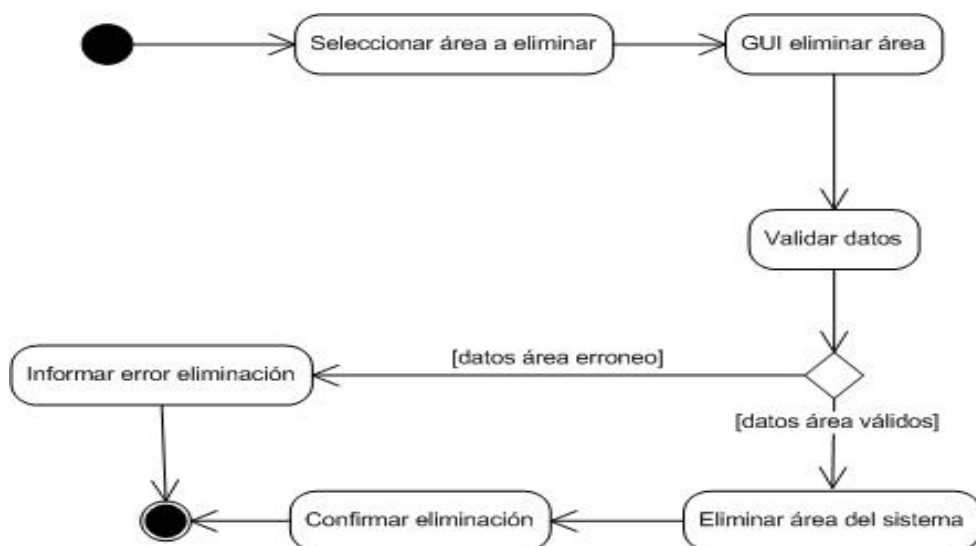



Figura 12. Diagrama de actividades caso uso 'eliminación de un área'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.2.3 Caso de uso 'determinar responsable de área'

El responsable de área es parte de la información a especificar cuando se crea un área, aun así se documenta aquí como un caso de uso aparte por su importancia, y por el hecho de que esta operación es similar a las que se realizan con actores diferentes en los casos de uso referido a proyectos y tareas.

Este caso de uso se describe como el administrador determina un responsable de área que se encargara de la gestión de la misma. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades

- Diagrama de caso de uso



Figura 13. Caso de uso 'determinar responsable de área'

- Diagrama de secuencia

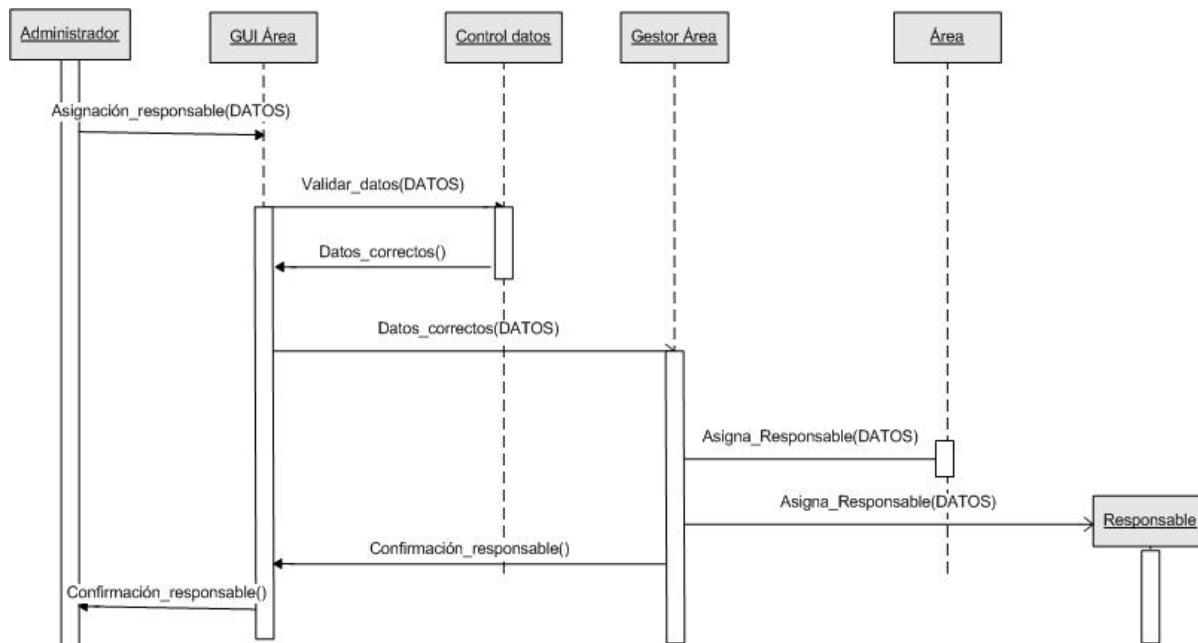



Figura 14. Diagrama de secuencia del caso de uso 'determinar responsable de área'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividad

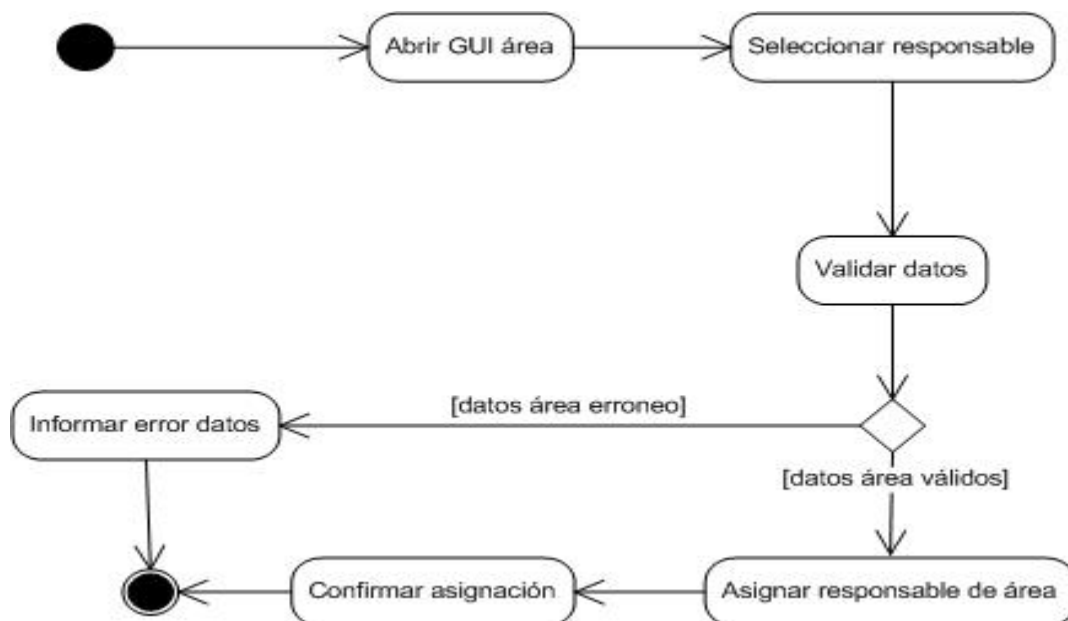


Figura 15. Diagrama de actividad del caso de uso 'determinar responsable de área'

3.3.2.4 Caso de uso 'modificar información de área'

En este caso de uso se describe la acción de modificar mediante el SGP la información (definición, planificación, documentos, información del proceso) guardadas en el SGP y relacionados con el área. Las operaciones de crear, borrar y modificar la información asociada se han descrito mediante una sola operación genérica 'editar' dado que sus operativas son similares. En ella el responsable de área o el administrador interactúa con la GUI de gestión de información asociada al área. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades

- Diagrama de caso de uso

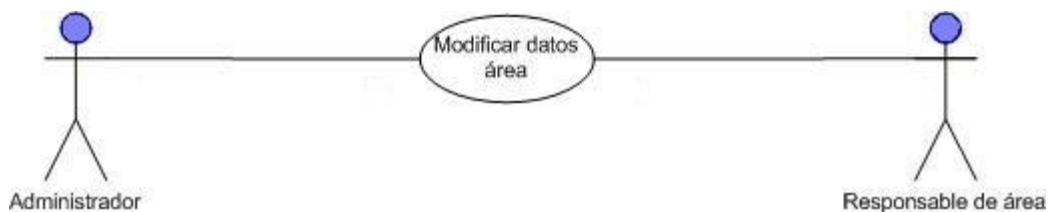


Figura 16. Caso de uso 'modificar información de área'

– Diagrama de secuencia

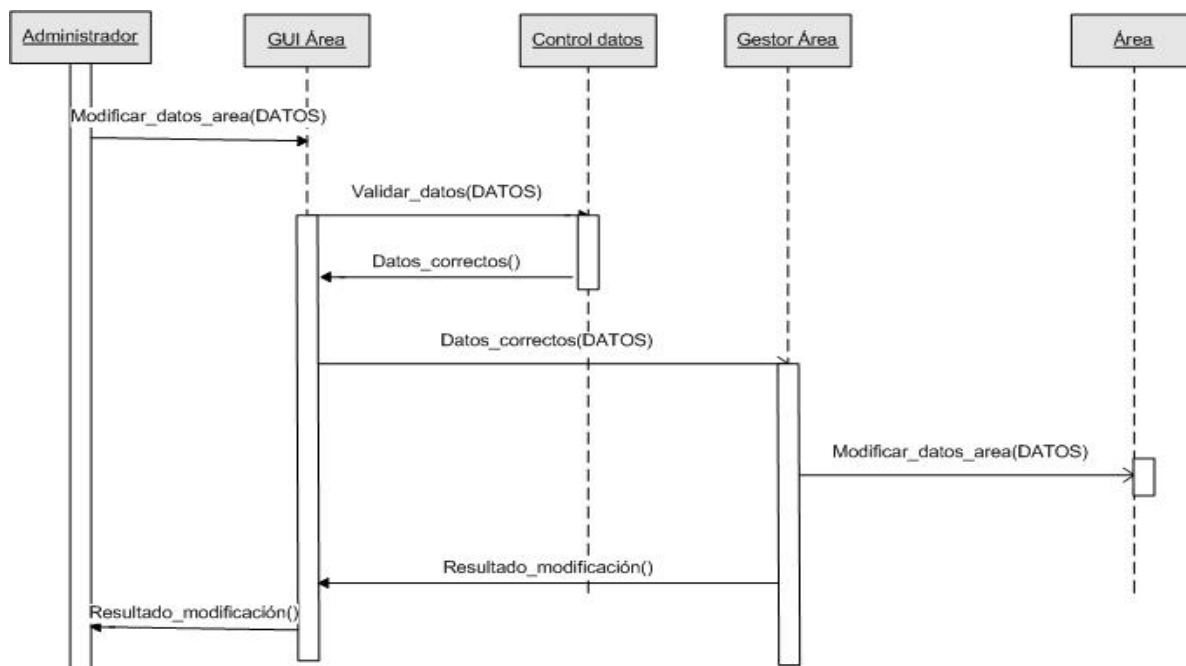


Figura 17. Diagrama de secuencia de caso de uso 'modificar información de área'

– Diagrama de actividades

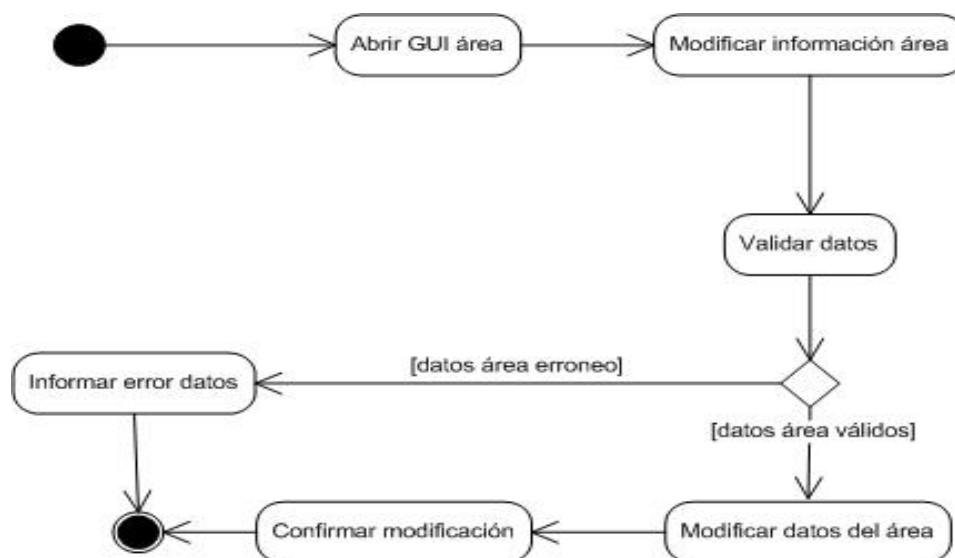



Figura 18. Diagrama de actividades de caso de uso 'modificar información de área'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.3 Caso de uso de proyecto

Los casos de uso especificados en este apartado son los relacionados con la gestión de los proyectos en el SGP. El proyecto es la entidad dependiente del área dentro de la jerarquía del SGP y de él dependen tareas que representan procesos o fases constituyentes o dependientes del proyecto. Cada entidad tiene un usuario responsable que se encarga de la administración de la información asociada, de la creación y control de las tareas dependientes del mismo, de generar informes, administrar los documentos, gestionar fechas de planificación.

Los casos de uso de proyecto identificados en esta apartado se muestran a continuación.

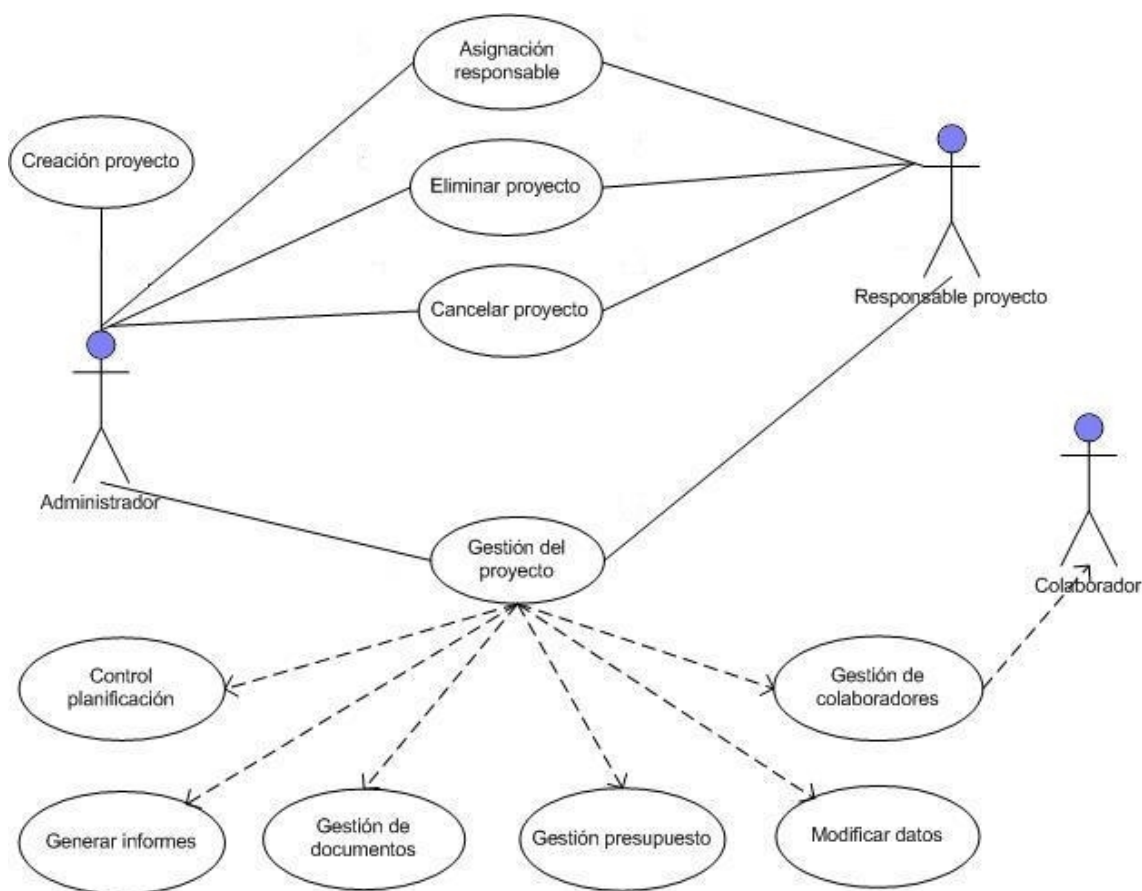



Figura 19. Caso de uso proyecto

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.3.1 Caso de uso ‘Creación de un nuevo proyecto’

En este caso de uso se define la funcionalidad por parte del responsable de área de crear un proyecto dentro de la misma. El proyecto creado pasa a estar integrado dentro de la estructura jerárquica del SGP. Esta operación es similar a la descrita en el apartado 3.3.2.1 ‘Caso de uso creación de una nueva área’ con la variación de que en este caso el actor que crea el nuevo proyecto es el responsable de área. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.1 con los siguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis el apartado comentado.

3.3.3.2 Caso de uso ‘eliminar proyecto’


En este caso de uso se define la funcionalidad por parte del responsable del proyecto o de área de eliminar un proyecto dependiente de la misma. Esta operación es similar a la descrita en el apartado 3.3.2.2 ‘Caso de uso eliminar Área’ con la variación de que en este caso el actor que elimina el proyecto es el responsable de área o el del proyecto. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.2 con los siguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.3.3 Caso de uso ‘cancelar proyecto’

En este caso de uso se define la funcionalidad por parte del responsable de área o proyecto de cancelar un proyecto. Esta operación es similar a la descrita en el apartado anterior 3.3.3.2 ‘Eliminar proyecto’ y por tanto a la descrita en el apartado 3.3.2.2 ‘Caso de uso Eliminar área’ con la variación de que en este caso el actor que cancela el proyecto es el responsable de área o el responsable de proyecto, y de que el proyecto cancelado no es eliminado del SGP como en los casos anteriores, sino simplemente marcado como no activo. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.2 con los consiguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.3.4 Caso de uso ‘determinar responsable de proyecto’

En este caso de uso se define la funcionalidad por parte del responsable de área o administrador de asignar un responsable aun proyecto dependiente de la misma. Esta operación es similar a la descrita en el apartado anterior 3.3.2.4 ‘Determinar responsable de área’ con la variación de que en este caso el actor que nombra el responsable del proyecto es el responsable de área, y de que, lógicamente, el actor determinado por el caso de uso es el

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

responsable de proyecto. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.4 con los siguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.3.5 Caso de uso 'gestionar informes de proyecto'

En este caso de uso se describen las operaciones de generación de informes acerca de la situación y estado actual de un proyecto. El formato definitivo de los informes que se podrán generar con el SGP se especificará en fases más avanzadas de este proyecto, por lo que en esta clasificaremos los informes en tres tipos: 'informes de datos' referido a la documentación típica de listados y relaciones de información, 'estadísticas', referido a gráficos y porcentajes y 'diagramas' referido a representaciones gráficas de procesos y relaciones entre entidades del proyecto y al diagrama de Gantt de un proyecto. Dado que las operaciones a realizar son similares para los tres tipos de informes en los diagramas de este caso de uso se muestra una entidad genérica 'informe' y unas operaciones también generales sobre la misma que se especifican de manera concreta según el tipo de informe. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso

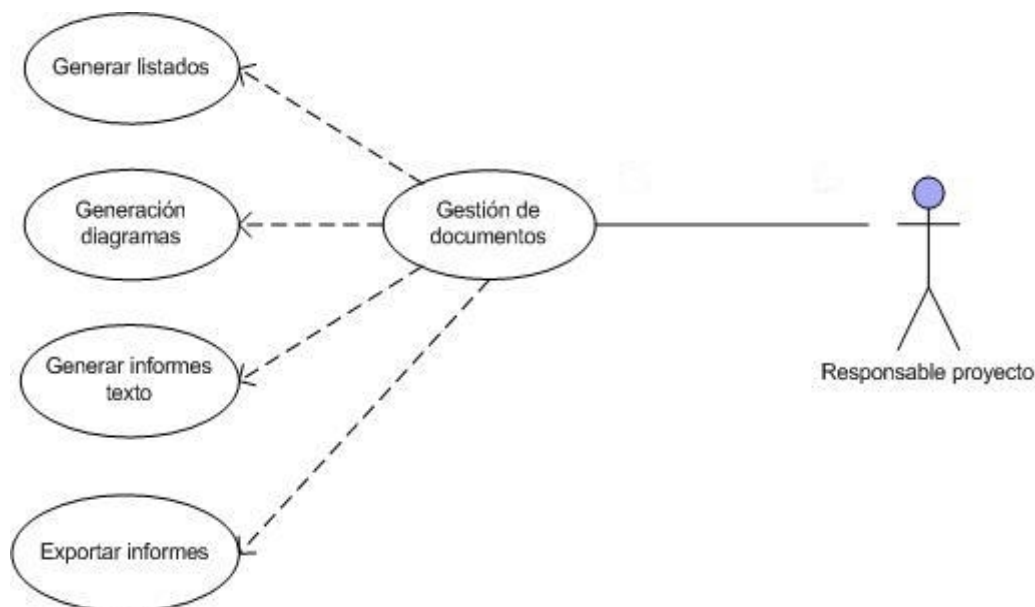


Figura 20. Caso de uso 'gestionar informes de proyecto'

- Diagrama de secuencia 'Generar informe de proyecto'

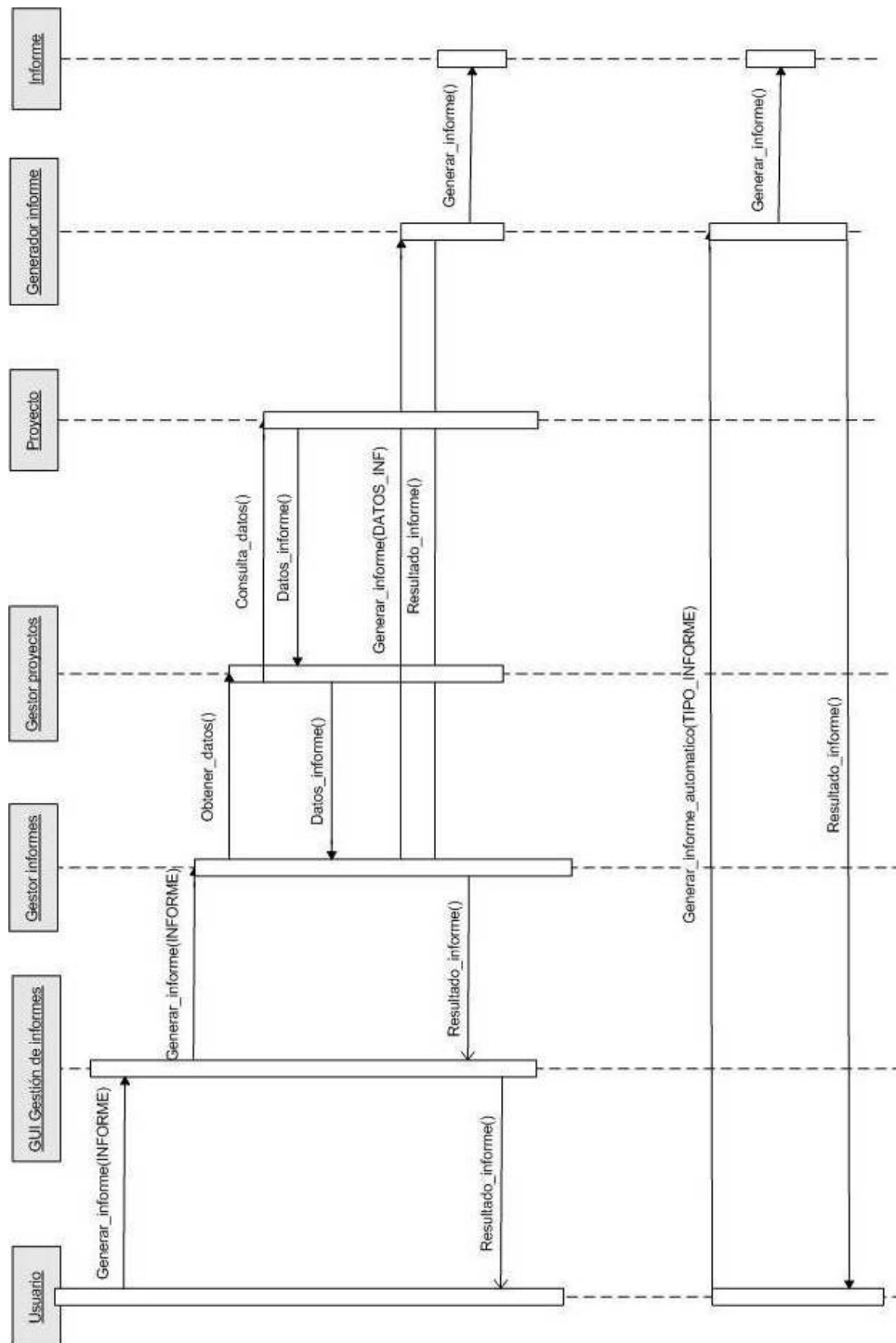



Figura 21. Diagrama de secuencia del caso de uso 'gestionar informes de proyecto'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividades caso de uso ‘gestionar informes de proyecto’

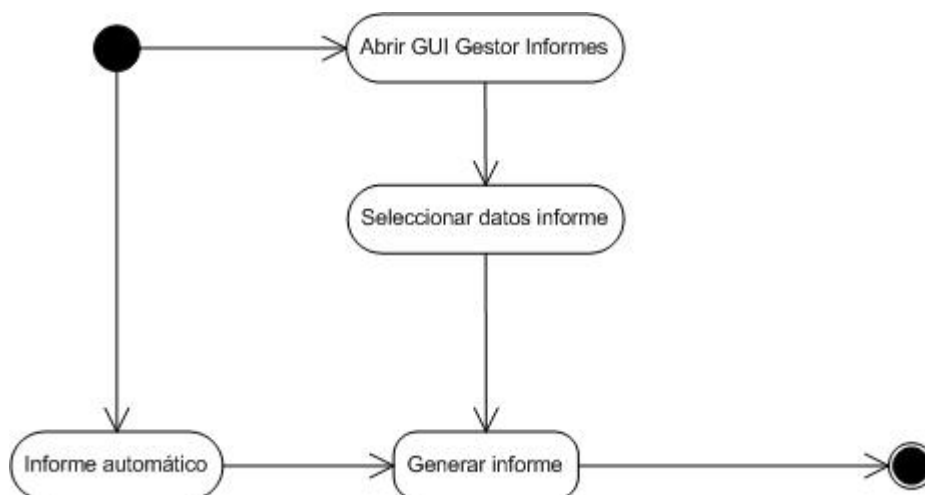


Figura 22. Diagrama de actividades de caso de uso ‘gestionar informes de proyecto’

3.3.3.6 Caso de uso ‘gestionar documentos de proyecto’

En este caso de uso se define la funcionalidad por parte del responsable de proyecto de administrar mediante el SGP los documentos (cualquier tipo de fichero externo al SGP) relacionado con el proyecto. Para facilitar la comprensión en este caso de uso se describen las funcionalidades de añadir, eliminar, modificar información de documentos asociados al proyecto de manera genérica. En ellos el usuario interactúa con el GUI de gestión de documentos y a través de él determina que operación quiere realizar. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y actividades.

- Diagrama de caso de uso

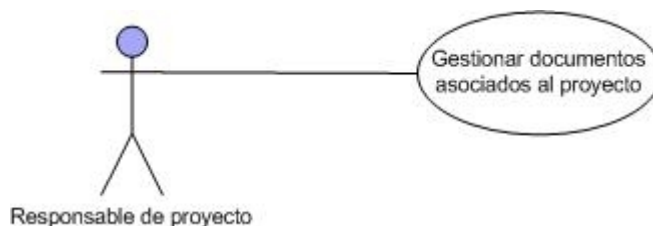



Figura 23. Caso de uso ‘gestionar documentos asociados al proyecto’

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

– Diagrama de secuencia

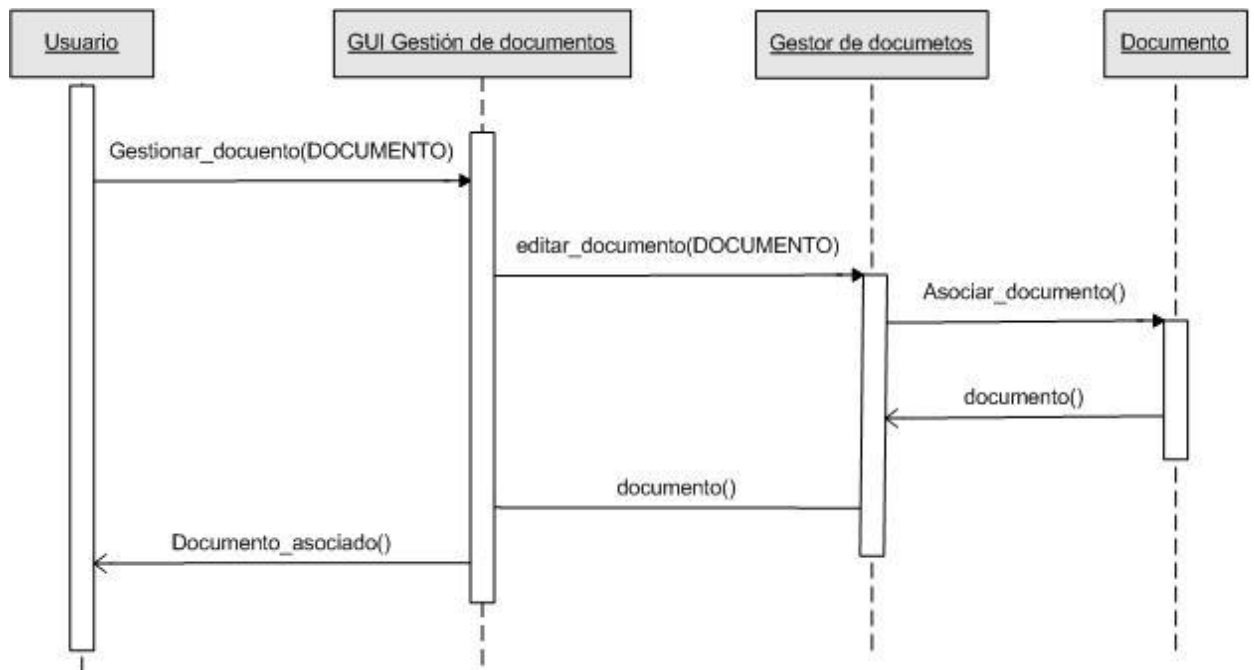



Figura 24. Diagrama de secuencia de caso de uso 'gestionar documentos asociados al proyecto'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividades de caso de uso 'gestionar documentos asociados al proyecto'

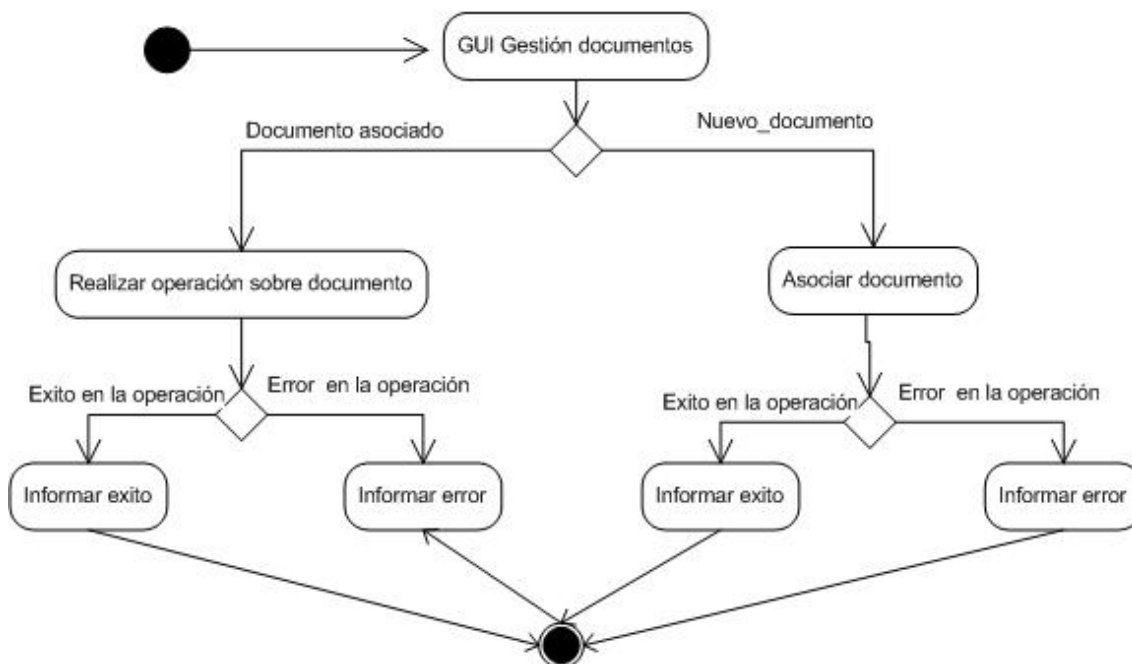



Figura 25. Diagrama de actividades de caso de uso 'gestionar documentos asociados al proyecto'

3.3.3.7 Caso de uso 'gestionar colaboradores del proyecto'

En este caso de uso se describe la funcionalidad por parte del responsable de proyecto de elegir y eliminar de entre los miembros de la organización a los que van a tener el rol de colaboradores de proyecto. El colaborador de proyecto es aquel miembro de la organización que, sin tener una responsabilidad directa dentro del proyecto, desempeña alguna tarea en el mismo, y por tanto es necesario tener controlado quienes han sido designados como colaboradores y aportar funcionalidades dentro del SGP que permitan gestionar y actualizar la información asociada a los mismos. Para la realización de este caso de uso el responsable de proyecto interactúa con el GUI de gestión de colaboradores que presenta a los posibles miembros colaboradores de la organización de una manera estructurada. El control de gestión de colaboradores realizara la operación validando los empleados elegidos, y le informará del resultado. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de caso de uso

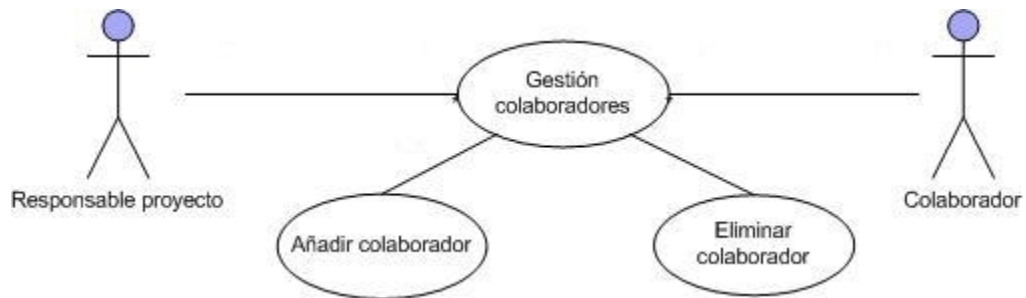


Figura 26. Caso de uso 'gestionar colaboradores del proyecto'

- Diagrama de secuencia de caso de uso 'gestión de colaboradores del proyecto'

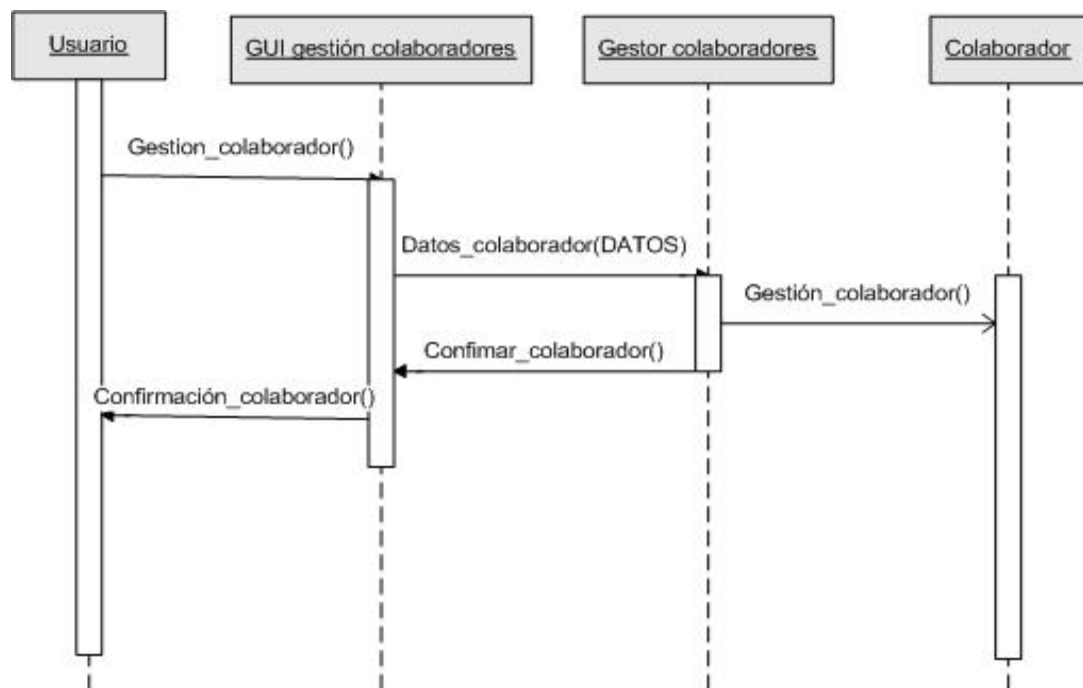


Figura 27. Diagrama de secuencia de caso de uso 'gestionar colaboradores del proyecto'

- Diagrama de actividades caso de uso 'gestión colaboradores del proyecto'

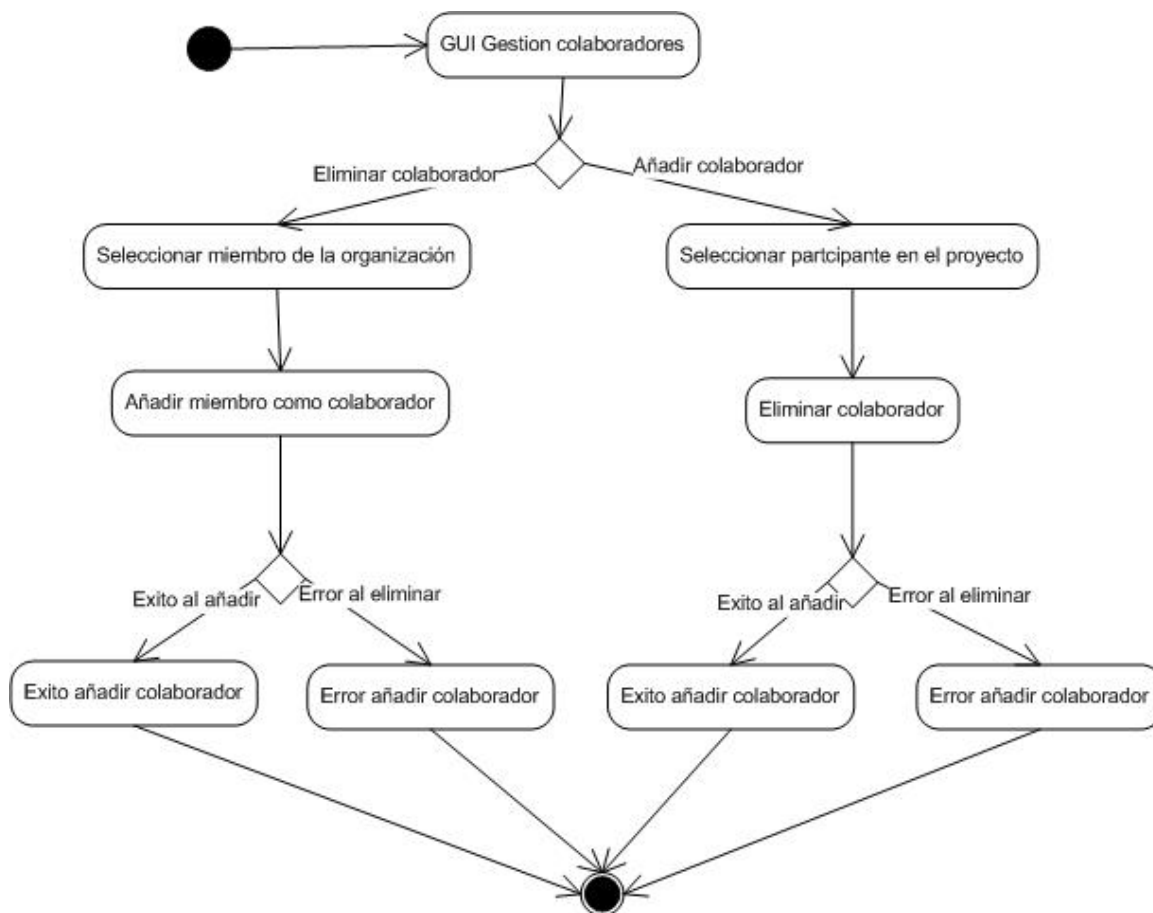



Figura 28. Diagrama de actividades de caso de uso 'gestionar colaboradores del proyecto'

3.3.4 Casos de uso de tarea

Los casos de uso especificados en este apartado son los relacionados con la gestión de las tareas en el SGP. La tarea es la entidad de más bajo nivel dentro de la jerarquía del SGP y representa un proceso o fase constituyente o dependiente de un proyecto. Las tareas presentan una funcionalidad en cuanto a casos de uso muy similar a la de un proyecto, con algunas excepciones. Como se especifica en el documento de requisitos toda tarea tiene un usuario responsable de la misma que se encarga de la administración de la información asociada a la tarea, de gestionar las fechas de la tarea, gestionar los documentos y colaboradores asociados, así como gestionar la organización de tareas predecesoras. Es función del responsable de proyecto la creación y eliminación de tarea y la asignación de sus responsables.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

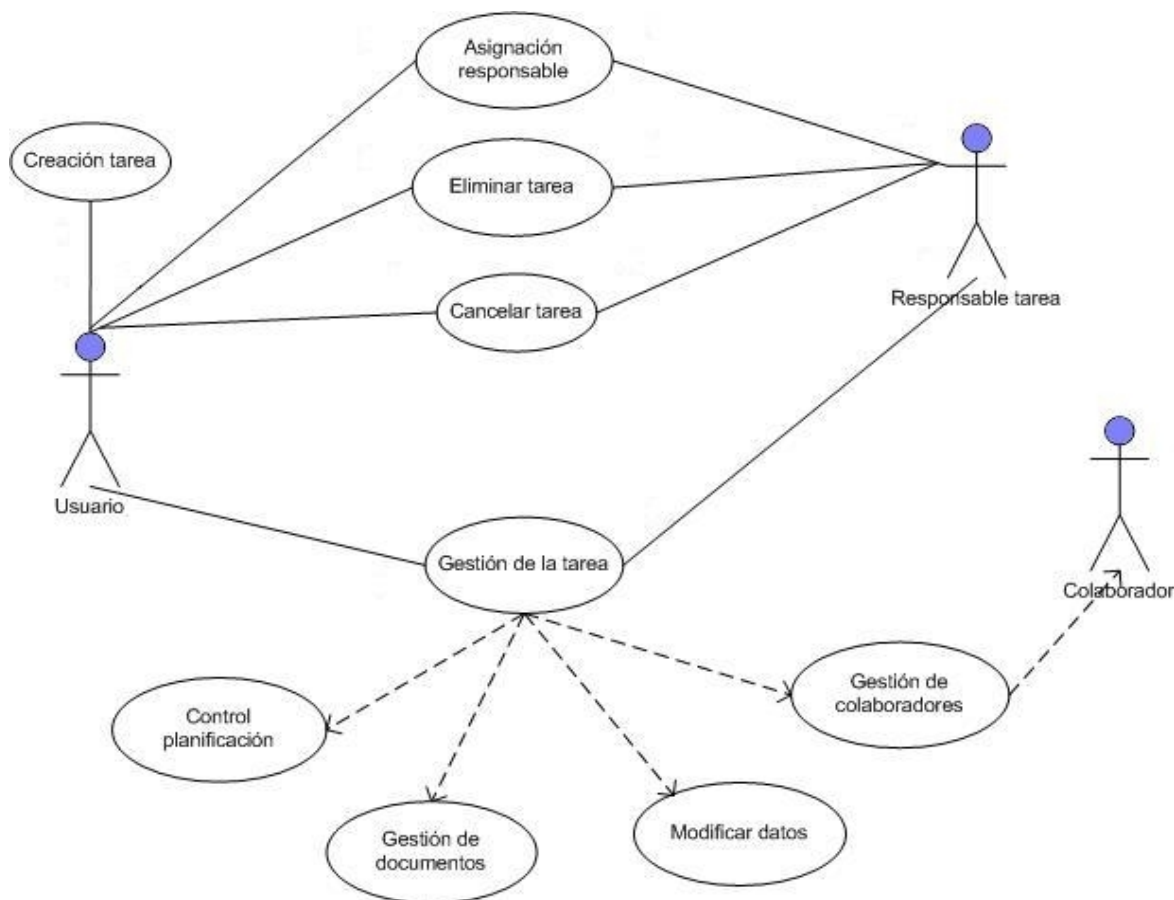



Figura 29. Caso de uso general gestión de tareas

3.3.4.1 Caso de uso 'Creación de una nueva tarea'

En este caso de uso se define la funcionalidad por parte del administrador o del responsable de crear una tarea dentro de la misma. La tarea creada pasa a estar integrada dentro de la estructura jerárquica del SGP. Esta operación es similar a la descrita en el apartado 3.3.2.1 'Caso de uso creación de una nueva área' con la variación de que en este caso el actor que crea la nueva tarea es el administrador o responsable. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.3.1 con los siguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis el apartado comentado.

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.4.2 Caso de uso ‘eliminar tarea’

En este caso de uso se define la funcionalidad por parte del administrador o del responsable de eliminar una tarea dependiente de la misma. La tarea eliminada desaparece de la estructura jerárquica del SGP. Esta operación es similar a la descrita en el apartado 3.3.2.2 ‘Caso de uso eliminar Área’ con la variación de que en este caso el actor que elimina la tarea es el administrador o el responsable de proyecto o de tarea. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.2 con los siguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.4.3 Caso de uso ‘cancelar tarea’


En este caso de uso se define la funcionalidad por parte del administrador, responsable de proyecto o de tarea de cancelar una tarea. Esta operación es similar a la descrita en el apartado anterior 3.3.3.2 ‘Eliminar proyecto’ y por tanto a la descrita en el apartado 3.3.2.2 ‘Caso de uso Eliminar área’ con la variación de que en este caso el actor que cancela la tarea puede ser el responsable de tarea, y de que la tarea cancelada no es eliminada del SGP como en los casos anteriores, sino simplemente marcada como no activa. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.2 con los consiguientes cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.4.4 Caso de uso ‘determinar responsable de tarea’

En este caso de uso se define la funcionalidad por parte del responsable de proyecto o administrador e incluso por otro responsable de tarea si la nueva tarea es dependiente de esta, de asignar un responsable a una tarea dependiente de la misma. Esta operación es similar a la descrita en el apartado anterior 3.3.2.4 ‘Determinar responsable de área’ con la variación de que en este caso el actor que nombra el responsable de la tarea es el responsable de proyecto o de otra tarea, y de que, lógicamente, el actor determinado por el caso de uso es el responsable de tarea. Las operaciones, flujos, controles y entidades de este caso de uso serían similares a los del apartado 3.3.2.4 con los siguiente cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.4.5 Caso de uso ‘gestionar colaboradores de tarea’

En este caso de uso se describe la funcionalidad por parte del responsable de tarea de elegir y eliminar de entre los miembros de la organización a los que van a tener el rol de colaboradores de tarea. Esta operación es similar a la descrita en el apartado 3.3.3.7 ‘gestionar colaboradores del proyecto’ con la variación de que en este caso se establece una responsabilidad con la

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

tarea y el actor encargado de interactuar con el GUI es el responsable de tarea. . Las operaciones, flujos, controles y entidades de esta caso de uso serian similares a los del apartado 3.3.3.7 con los siguiente cambios nominales y por ello no se repiten en este punto, remitiendo su análisis al apartado comentado.

3.3.5 Casos de uso del sistema

3.3.5.1 Caso de uso 'configurar el sistema'

En este apartado se describen los casos de uso relacionados con la configuración del sistema, la cual consiste en la parametrización de distintos conceptos, para personalizar el uso de la aplicación. Dichos conceptos o parámetros serán: centros del proyecto, tipo de proyecto, partida presupuestaria, tipo de alcance, tipo de convocatoria, tipo de notificación, indicadores económicos, centros de coste, centros de trabajo, idioma, nacionalidad.

En la siguiente figura, se observa el diagrama de caso de uso 'configurar el sistema', a nivel 1.

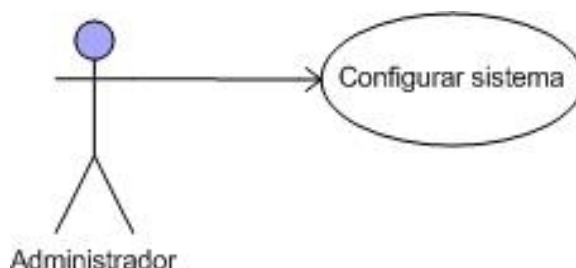


Figura 30. Caso de uso general configuración del sistema

A continuación bajamos un nivel en el anterior diagrama, para descomponerlo en casos de uso más sencillos, que serán analizados individualmente.

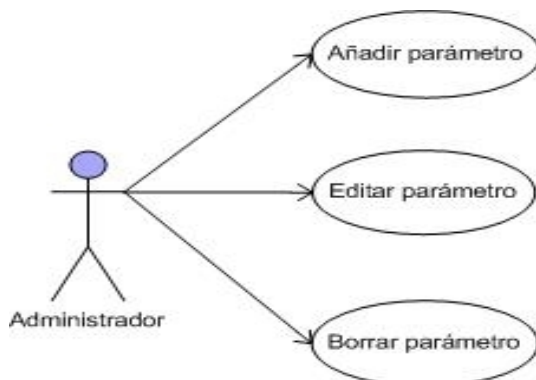



Figura 31. Caso de uso 'configurar el sistema'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.5.2 Caso de uso 'añadir parámetro'

En este caso de uso se define la funcionalidad de añadir un nuevo parámetro por parte de administrador. Este interactúa con la GUI del sistema, introduciendo mediante un formulario los datos del parámetro a añadir. Mediante el control de datos, se asegura que los datos serán validos, pasándoselos al control de gestión de parámetros, que se encarga de la creación y almacenaje del nuevo parámetro en el sistema. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso 'añadir parámetro'

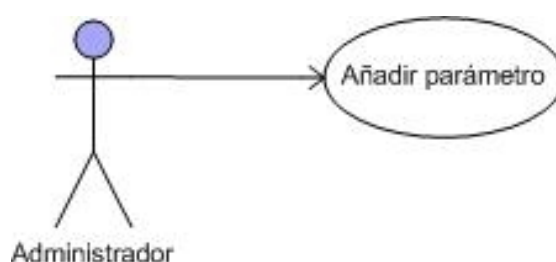


Figura 32. Caso de uso 'añadir parámetro'

- Diagrama de secuencia

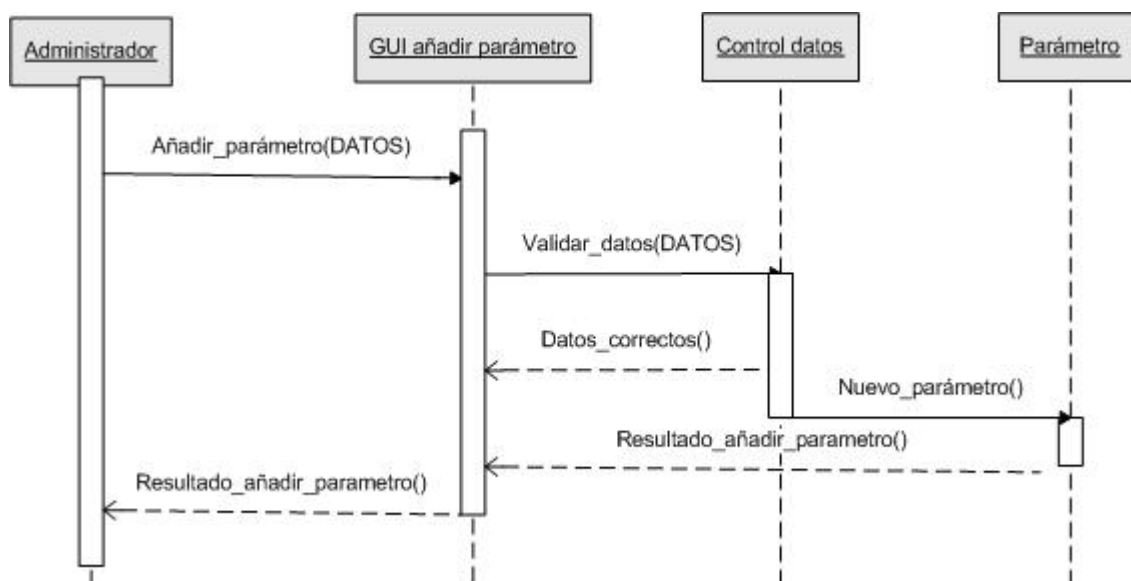



Figura 33. Diagrama de actividades de caso de uso 'añadir parámetro'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividad

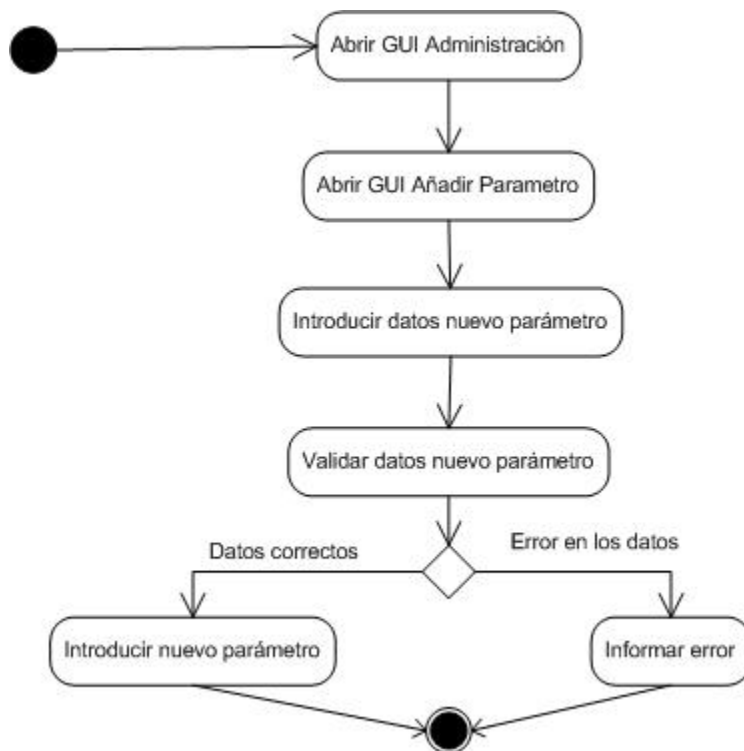


Figura 34. Diagrama de actividades de caso de uso 'añadir parámetro'

3.3.5.3 Caso de uso 'editar parámetro'

En este caso de uso se define la funcionalidad de editar un parámetro ya existente en el sistema, por parte del administrador que interactúa con la GUI del sistema, modificando mediante un formulario los datos del parámetro. Mediante el control de datos, se asegura que los nuevos datos sean válidos, pasándoselos al control de gestión de parámetros, que se encarga de modificar los datos del parámetro. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso

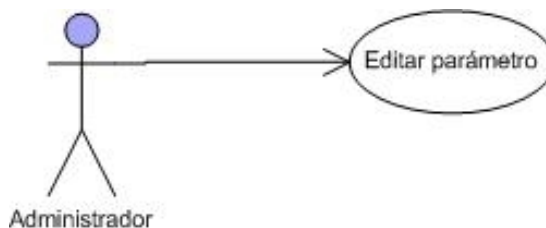



Figura 35. Caso de uso 'editar parámetro'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

– Diagrama de secuencia

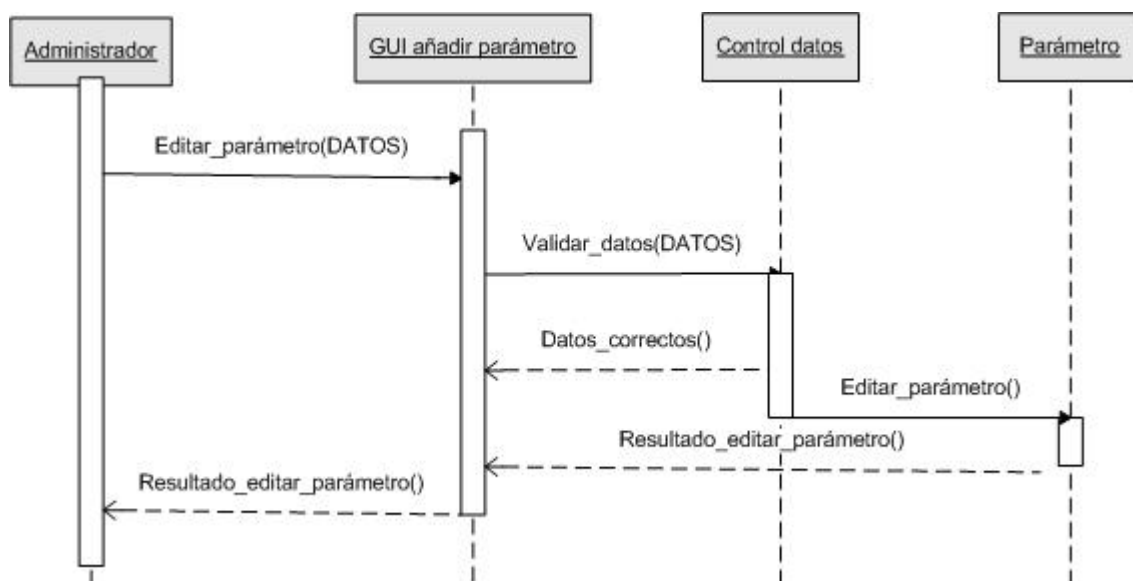


Figura 36. Diagrama de secuencia de caso de uso 'editar parámetro'

– Diagrama de actividad

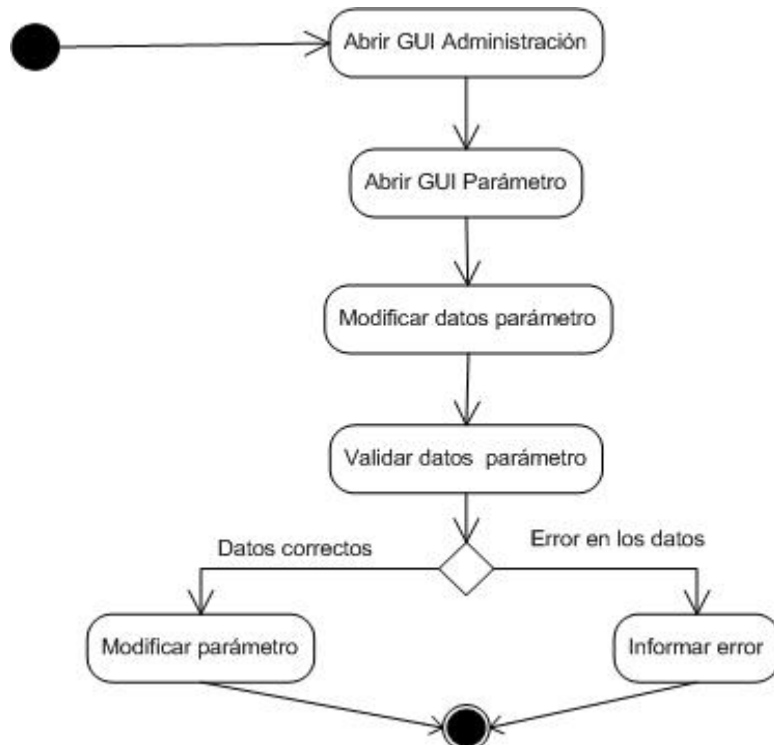



Figura 37. Diagrama de actividad de caso de uso 'editar parámetro'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.5.4 Caso de uso 'eliminar parámetro'

En este caso de uso se define la funcionalidad de eliminar un parámetro del sistema por parte del administrador que interactúa con la GUI del sistema, indicando el parámetro a eliminar. El control de datos comprueba si el parámetro seleccionado es válido, ya que puede haber restricciones que impidan la eliminación de dicho parámetro.

- Diagrama de caso de uso

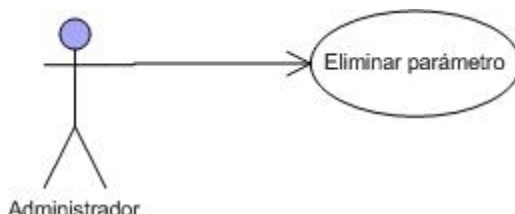


Figura 38. Caso de uso 'eliminar parámetro'

- Diagrama de secuencia

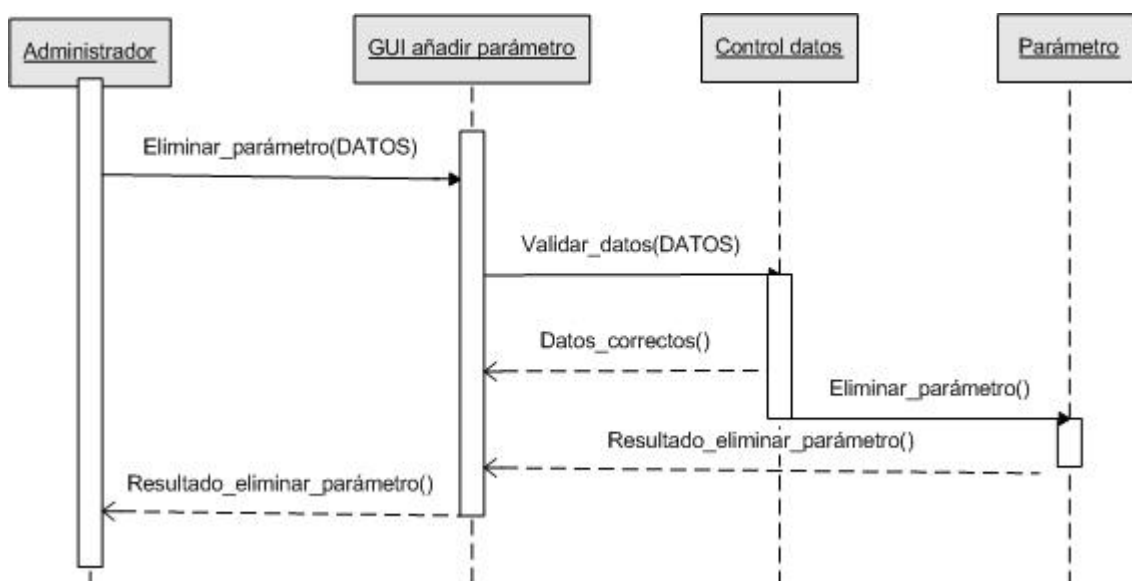



Figura 39. Diagrama de secuencia case de uso 'eliminar parámetro'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividad

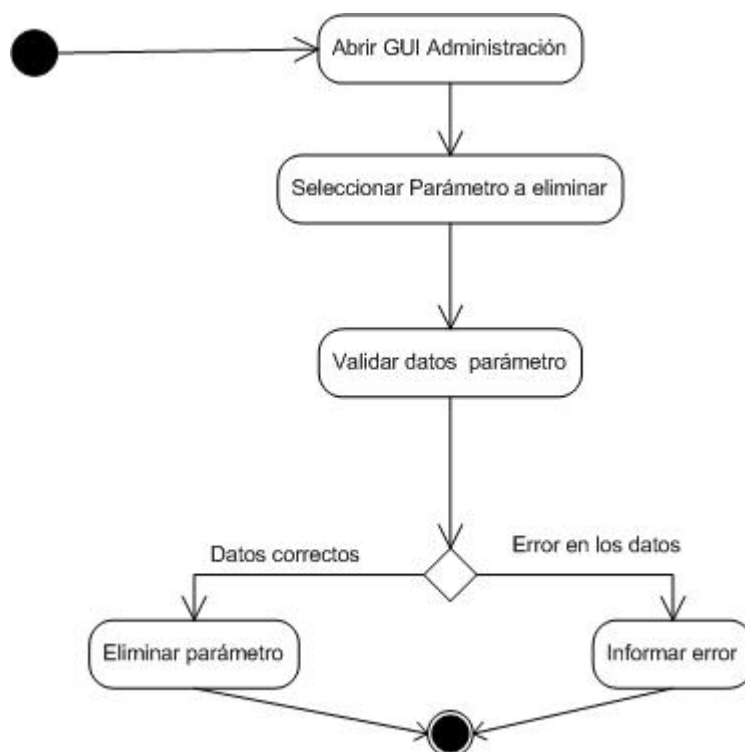


Figura 40. Diagrama de actividades de caso de uso 'eliminar parámetro '

3.3.6 Casos de uso de personas

En este apartado se describen los casos referidos a la gestión de personas por parte del administrador de la aplicación.

- Diagrama de caso de uso

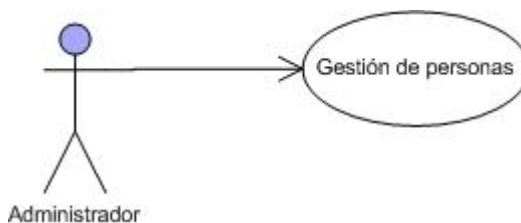



Figura 41. Caso de uso 'gestión de personas'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

En la siguiente figura, se observa el diagrama del caso de uso 'Gestión de personas' con un mayor nivel de detalle. Se muestra que la gestión de personas consta de tres casos de uso diferenciados: añadir empleado, editar datos de empleado, cambiar estado de empleado.

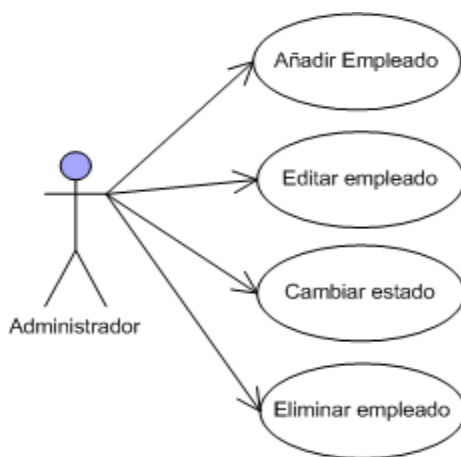


Figura 42. Caso de uso 'gestión de personas'

3.3.6.1 Caso de uso 'añadir empleado'

En este caso de uso se define la funcionalidad de añadir un nuevo empleado al sistema por parte de administrador. Este interactúa con la GUI del sistema, introduciendo mediante un formulario los datos del empleado a añadir. Mediante el control de datos, se asegura que los datos serán validos, pasándoselos al control de gestión de personas, que se encarga de la creación y almacenaje del nuevo empleado en el sistema. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso 'añadir empleado'

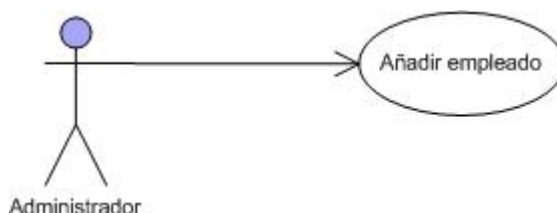


Figura 43. Caso de uso 'añadir empleado'

– Diagrama de secuencia

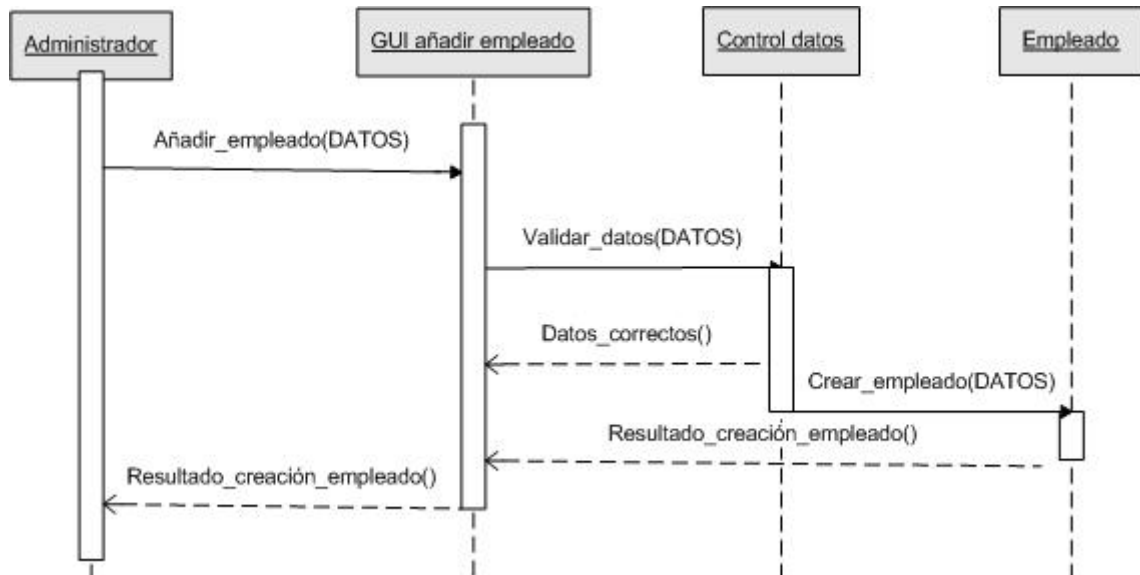


Figura 44. Diagrama de secuencia caso de uso 'añadir empleado'

– Diagrama de actividad

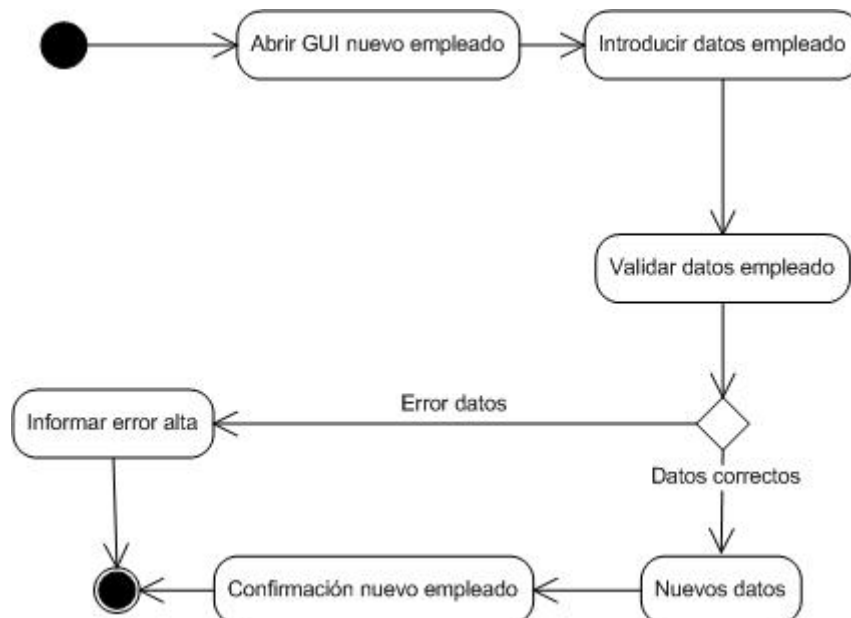



Figura 45. Diagrama de actividades de caso de uso 'añadir empleado'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.6.2 Caso de uso 'editar empleado'

En este caso de uso se define la funcionalidad de editar los datos de un empleado ya existente en el sistema por parte de administrador. Este interactúa con la GUI del sistema, modificando mediante un formulario los datos del empleado. Mediante el control de datos, se asegura que los datos serán válidos, pasándoselos al control de gestión de personas, que se encarga del almacenaje de los nuevos datos del empleado en el sistema. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso 'editar empleado'

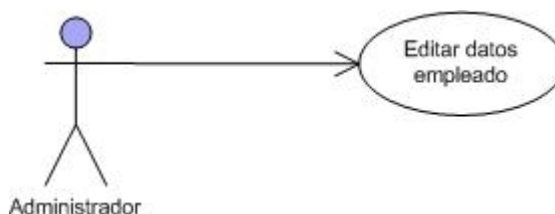


Figura 46. Caso de uso 'editar empleado'

- Diagrama de secuencia

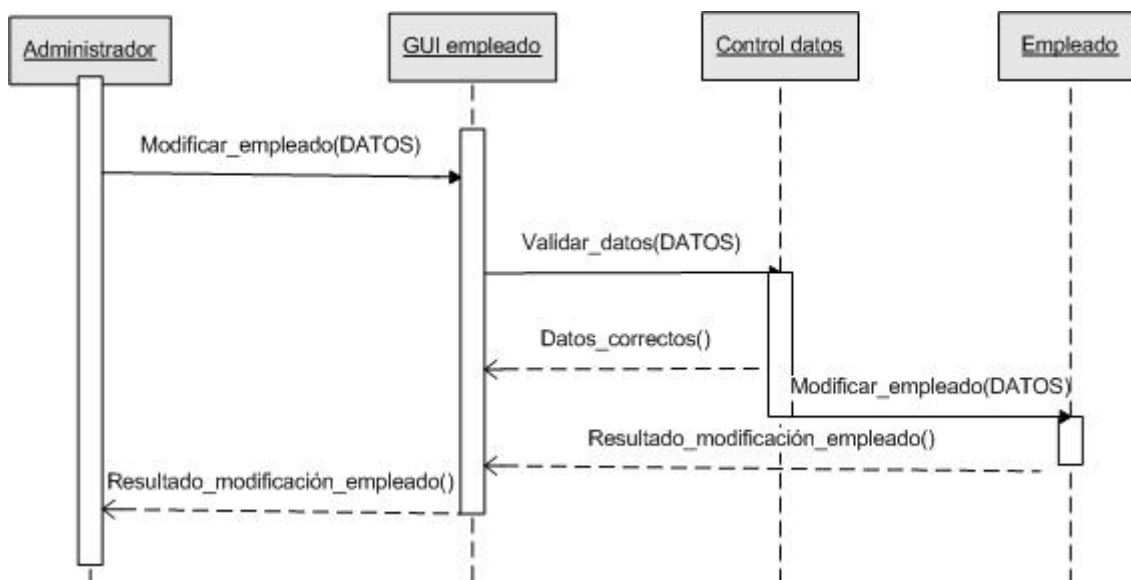


Figura 47. Diagrama de secuencia de caso de uso 'editar empleado'

- Diagrama de actividad

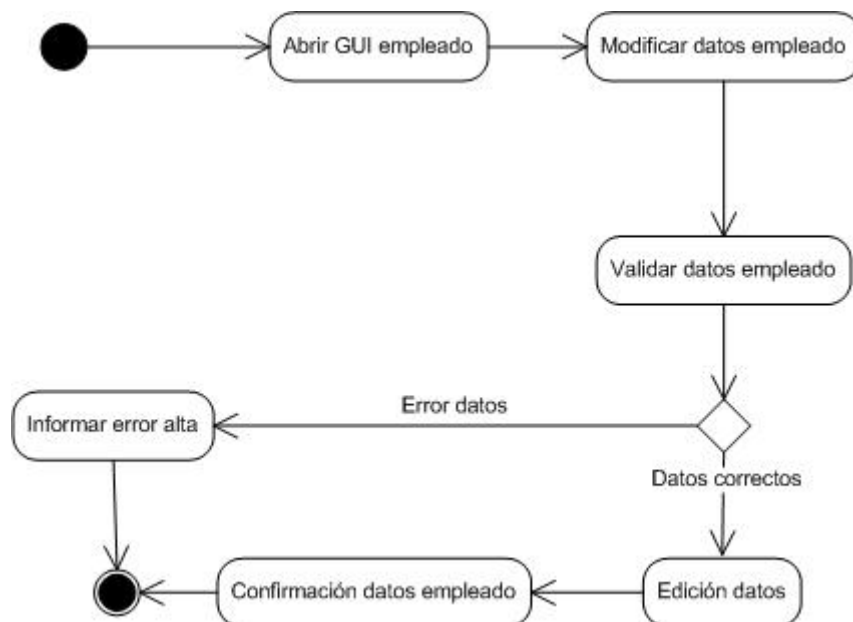


Figura 48. Diagrama de actividades de caso de uso 'editar empleado'

3.3.6.3 Caso de uso 'cambiar estado de empleado'

En este caso de uso se define la funcionalidad de cambiar de estado un empleado en el sistema por parte de administrador. Este interactúa con la GUI del sistema, estableciendo un nuevo estado para el empleado, los distintos estados que puede tener un empleado son actual, externo, histórico y externo histórico. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso 'cambiar estado de empleado'

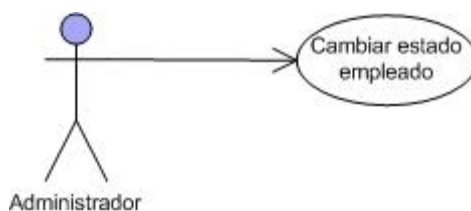


Figura 49. Caso de uso 'cambiar estado de empleado'

– Diagrama de secuencia

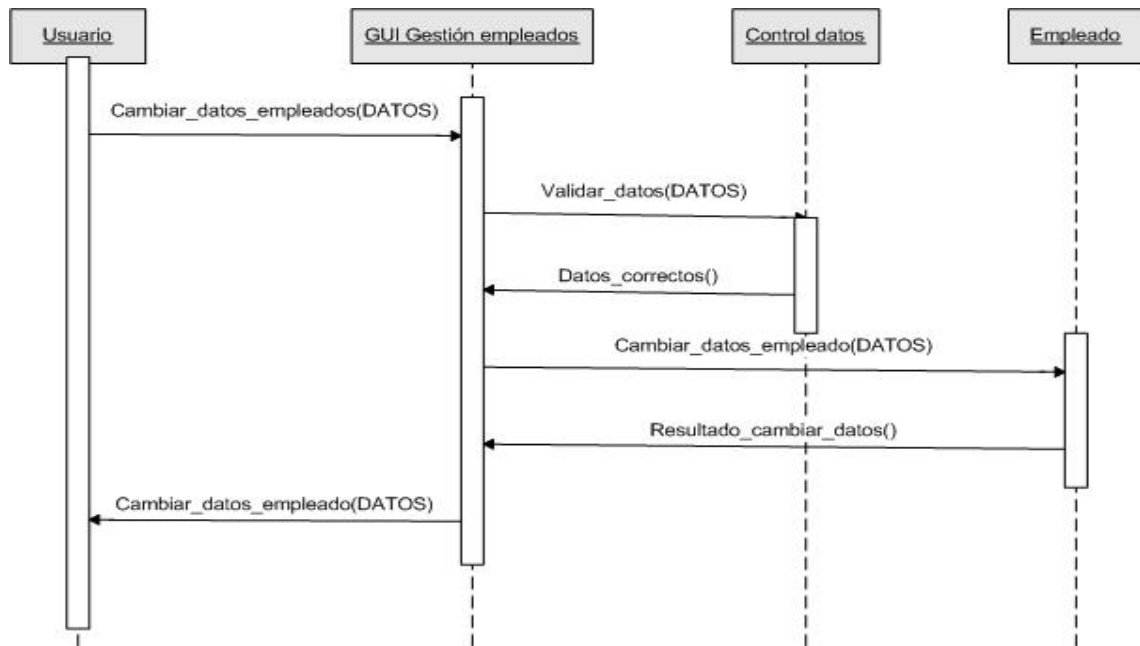


Figura 50. Diagrama de secuencia caso de uso 'cambiar estado de empleado'

– Diagrama de actividad

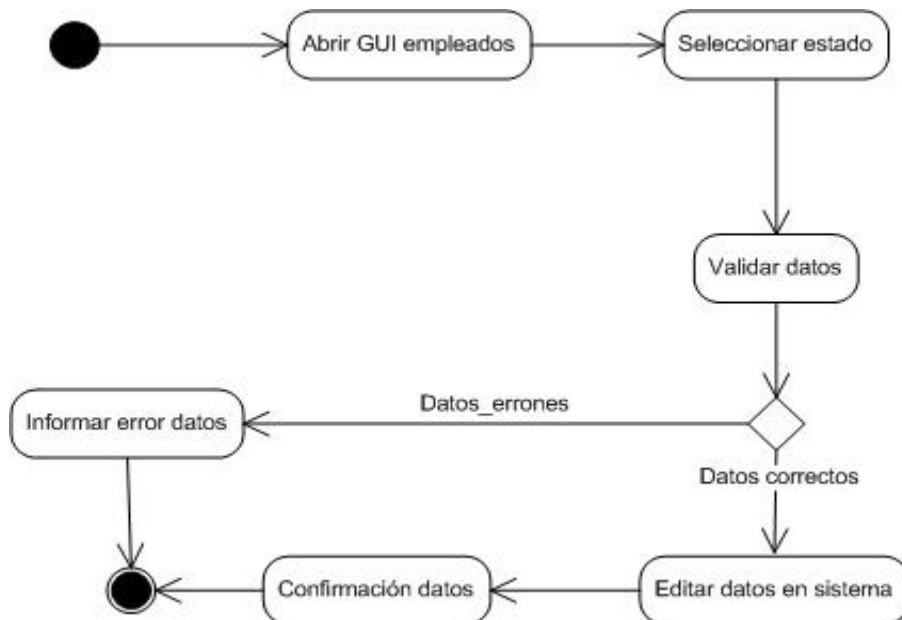



Figura 51. Diagrama de actividades de caso de uso 'cambiar estado de empleado'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.6.4 Caso de uso eliminar un empleado

En este caso de uso se define la funcionalidad de eliminar un empleado del sistema por parte de administrador. Este interactúa con la GUI del sistema, estableciendo eliminado completamente al empleado del sistema. Para eliminar el empleado este debe encontrarse en un estado que no sea actual. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso ‘eliminar un empleado’

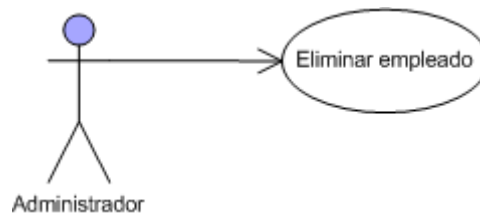


Figura 52. Caso de uso ‘cambiar estado de empleado’

- Diagrama de secuencia

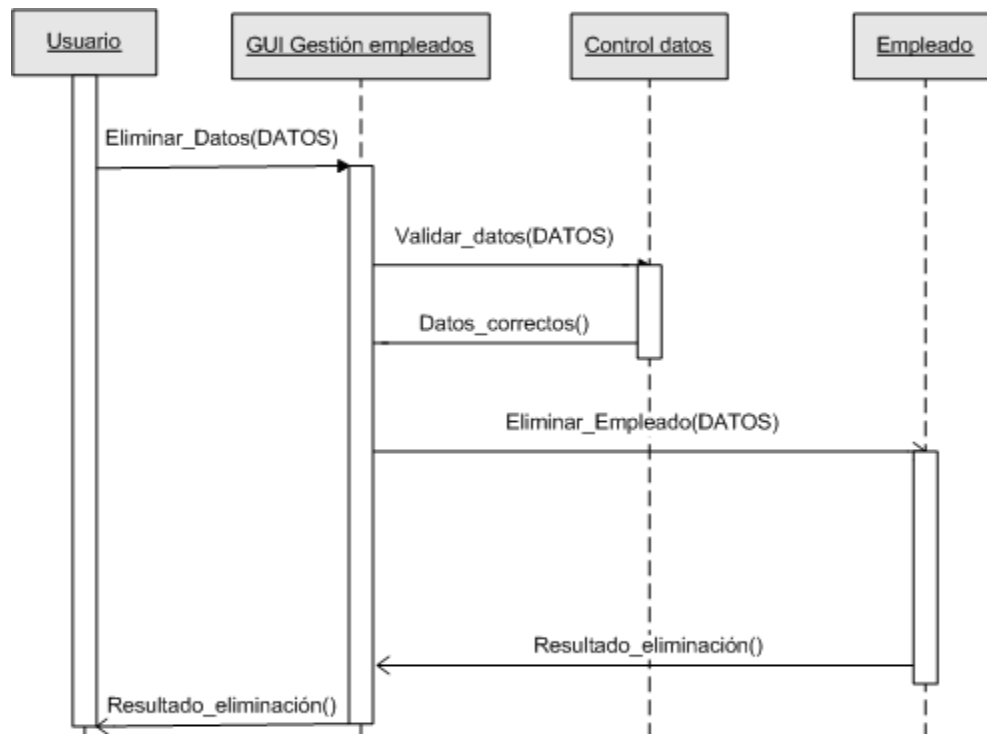



Figura 53. Diagrama de secuencia caso de uso ‘eliminar empleado’

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

- Diagrama de actividad

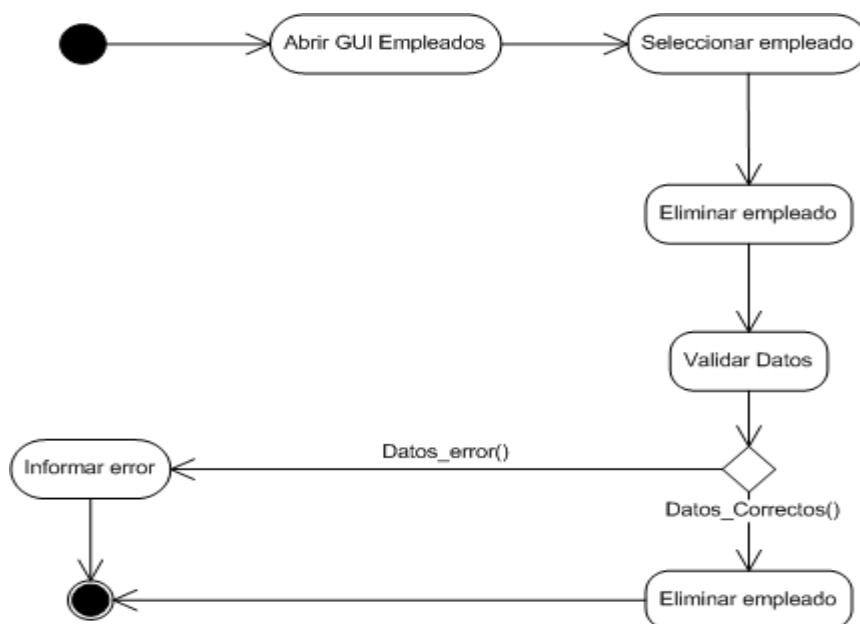


Figura 54. Diagrama de actividades de caso de uso 'cambiar estado de empleado'

3.3.7 Casos de uso de usuarios

En este apartado se describen los casos referidos a la gestión de usuarios por parte del administrador de la aplicación.

- Diagrama de caso de uso

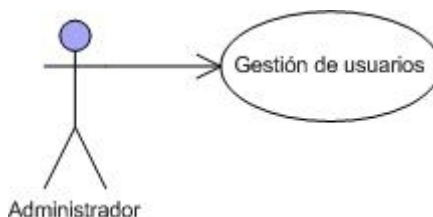



Figura 55. Caso de uso 'gestión de usuarios'

En la siguiente figura, se observa el diagrama del caso de uso 'Gestión de usuarios' con un mayor nivel de detalle. Se muestra que la gestión de usuarios consta de tres casos de uso diferenciados: crear usuario, eliminar usuario y editar datos de usuario.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

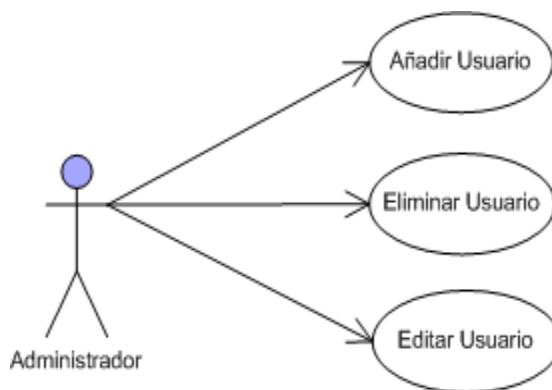


Figura 56. Caso de uso 'gestión de usuarios'

3.3.7.1 Caso de uso 'crear usuario'

En este caso de uso se define la funcionalidad de crear un nuevo usuario por parte del administrador para un empleado ya existente en el sistema. Este interactúa con la GUI del sistema, añadiendo los nuevos datos del usuario. Mediante el control de datos, se asegura que los datos serán válidos, pasándoselos al control de gestión de usuarios, que se encarga del almacenaje de los nuevos datos del usuario en el sistema. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso 'crear usuario'

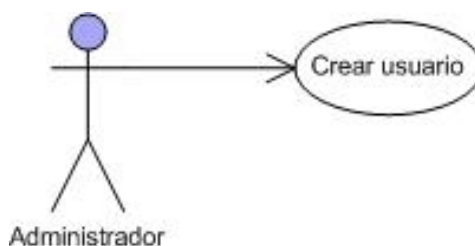


Figura 57. Caso de uso 'crear usuario'

– Diagrama de secuencia

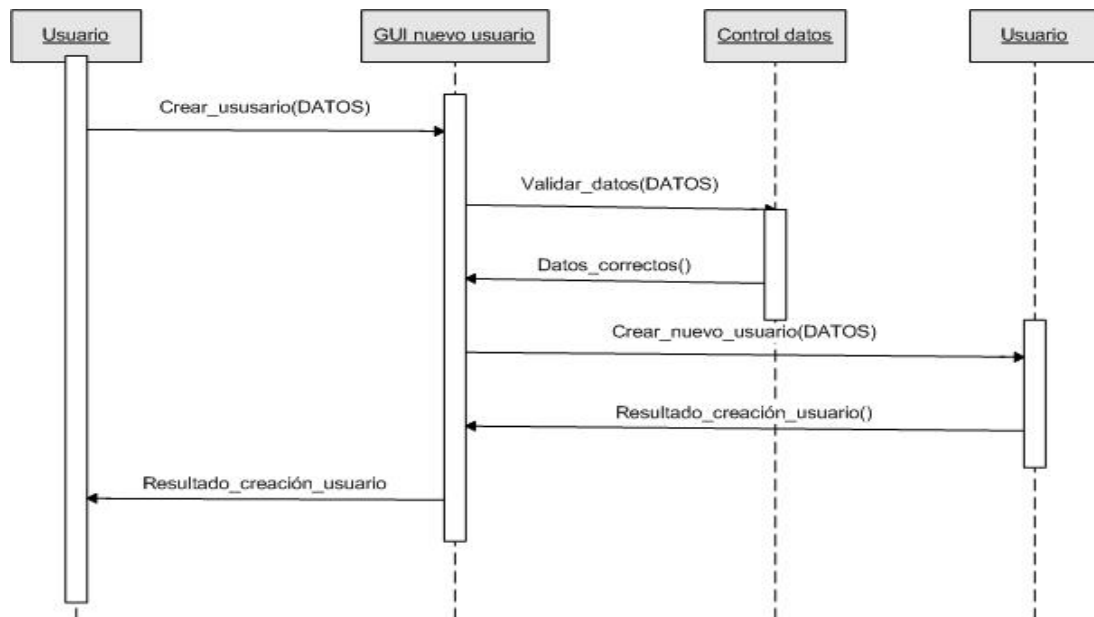


Figura 58. Diagrama de secuencia de caso de uso 'crear usuario'

– Diagrama de actividad

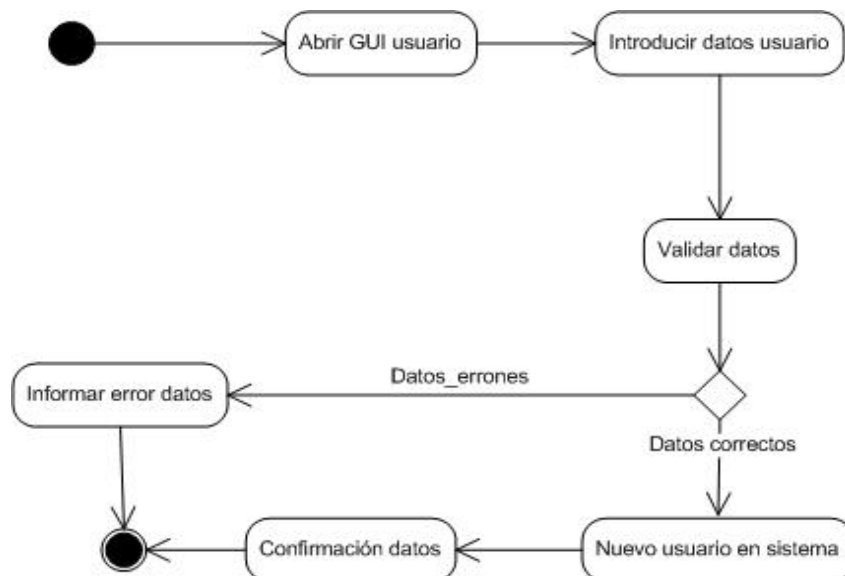



Figura 59. Diagrama de actividad de caso de uso 'crear usuario'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

3.3.7.2 Caso de uso 'eliminar usuario'

En este caso de uso se define la funcionalidad de eliminar un usuario del sistema por parte del administrador. Este interactúa con la GUI del sistema, seleccionando el usuario a eliminar y validando los datos de la eliminación. Mediante el control de datos, se asegura que los datos serán válidos, pasándoselos al control de gestión de usuarios, que se encarga de la eliminación de un usuario. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso

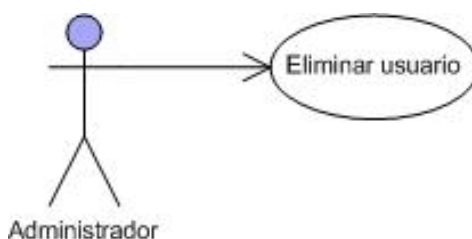


Figura 60. Caso de uso 'eliminar usuario'

- Diagrama de secuencia

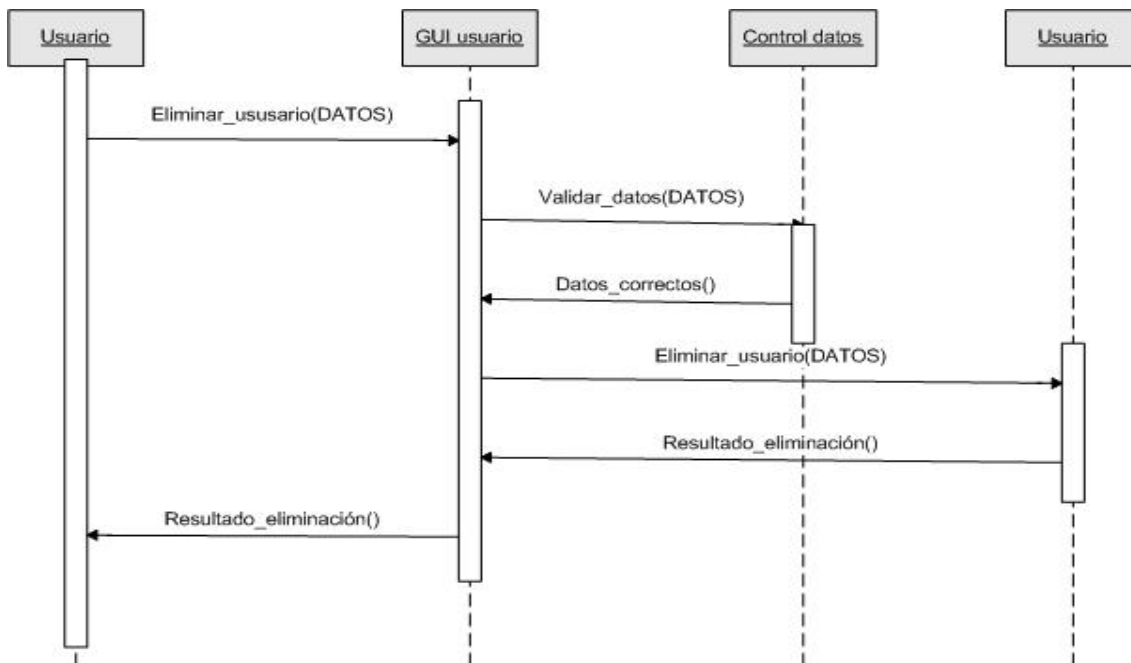


Figura 61. Diagrama de secuencia de caso de uso 'eliminar empleado'

- Diagrama de actividad

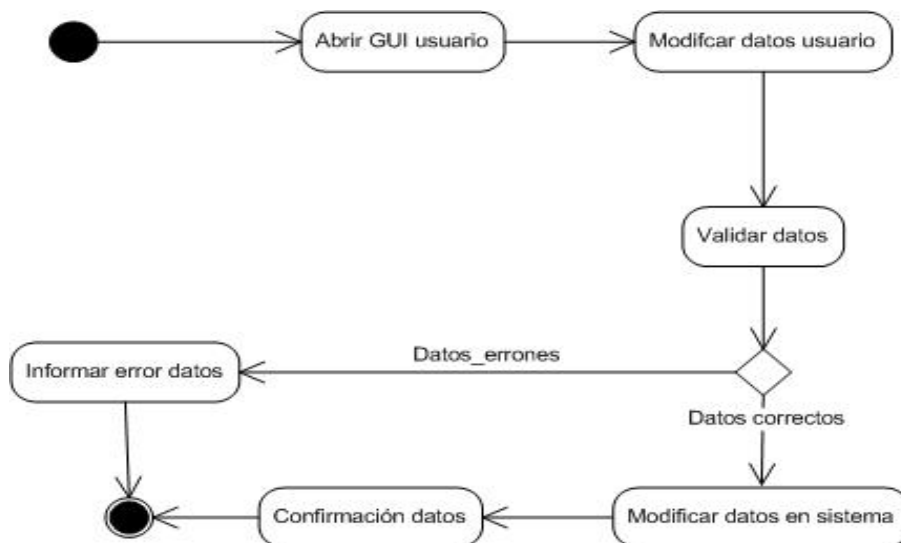


Figura 62. Diagrama de actividad caso de uso 'eliminar empleado'

3.3.7.3 Caso de uso 'editar datos de usuario'

En este caso de uso se define la funcionalidad de editar los datos de un usuario por parte del administrador del sistema. Este interactúa con la GUI del sistema, modificando los datos del usuario. Mediante el control de datos, se asegura que los datos serán válidos, pasándoselos al control de gestión de usuarios, que se encarga del almacenaje de los nuevos datos del usuario en el sistema. La descripción gráfica de este caso de uso viene dada por los siguientes diagramas de caso de uso, de secuencia y de actividades.

- Diagrama de caso de uso

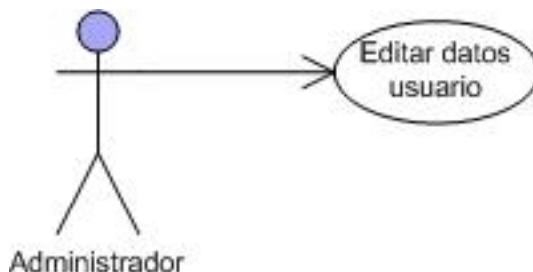


Figura 63. Caso de uso 'editar datos usuario'

– Diagrama de secuencia

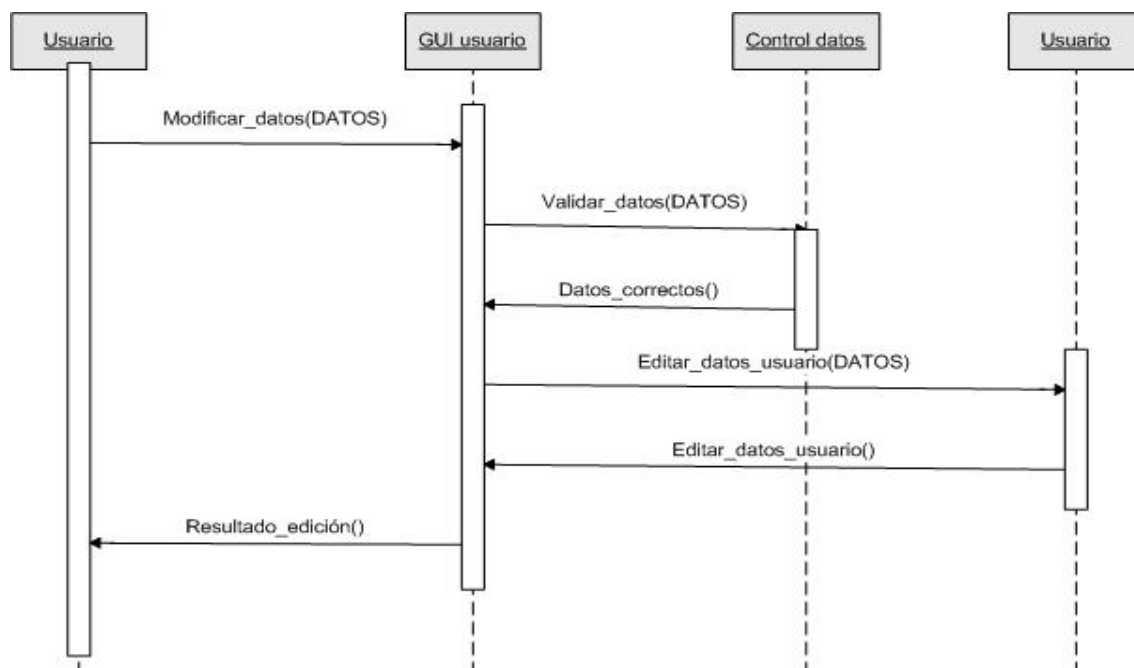


Figura 64. Diagrama de secuencia caso de uso 'editar datos usuario'

– Diagrama de actividad

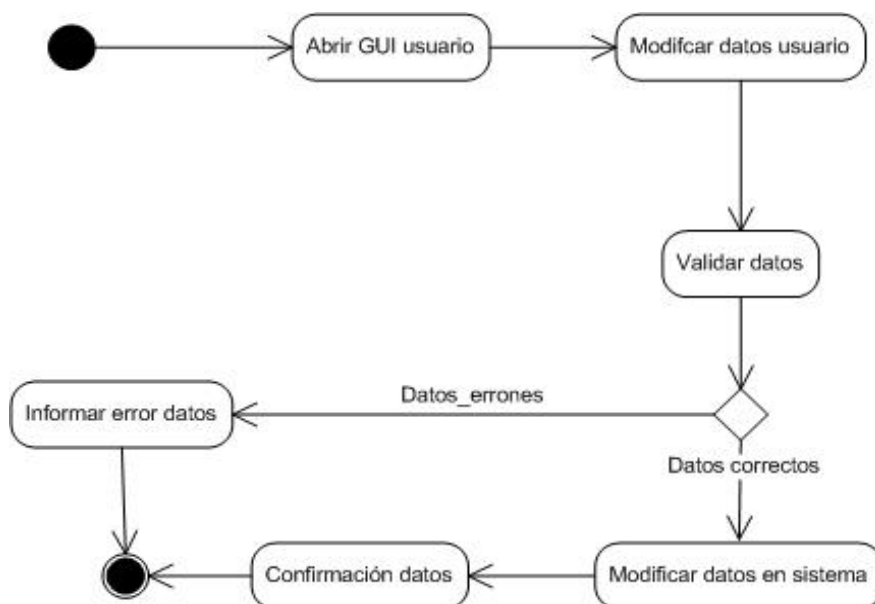



Figura 65. Diagrama de actividad caso de uso 'editar datos usuario'

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

4. ANÁLISIS DE PAQUETES

4.1 Introducción

Tras la identificación de los casos de uso en base a las entidades del sistema realizadas en apartado anterior, la siguiente fase en el proceso de análisis es la identificación de los paquetes del análisis.


Los paquetes del análisis proporcionan un medio para organizar el modelo del sistema en piezas más pequeñas y más manejables que serán la base natural de los futuros módulos de diseño. El principal objetivo en este caso es por lo tanto describir la estructuración de paquetes del sistema y observar las relaciones y/o dependencias existentes entre paquetes de análisis. Una identificación de los paquetes del análisis se hace de manera natural basándose en los requisitos funcionales descritos en el documento de requisitos, en la descripción del modelos de negocio y en los casos de uso detallados en el apartado 3 de este documento.

4.2 Identificación de paquetes de análisis

Una forma directa de identificar paquetes del análisis es asignar la mayor parte de un cierto número de casos de uso a un paquete concreto y después realizar la funcionalidad correspondiente dentro de ese paquete. A continuación en este apartado se irán detallando los paquetes de análisis mediante este proceso.

4.2.1 Paquetes de análisis 'Gestión de estructura de planes de acción'

El paquete de análisis, 'Gestión de estructura de planes de acción', engloba todas las funcionalidades relacionadas con la gestión de la estructura jerárquica de áreas, proyectos y tareas del SGP. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

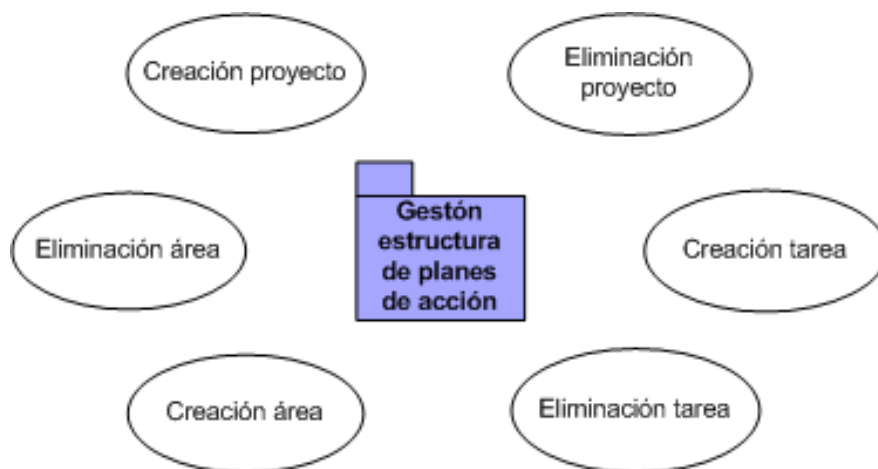



Figura 66. Paquete de análisis 'gestionar estructura de planes de acción'

4.2.2 Paquetes de análisis 'Gestión información de área'

El paquete de análisis, 'Gestión de información de área', engloba todas las funcionalidades relacionadas con la gestión de la información de los datos (identificativos y contenidos en) un área. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.



Figura 67. Paquete de análisis 'gestión información de áreas'

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

4.2.3 Paquetes de análisis ‘Gestión información de proyecto’

El paquete de análisis, ‘Gestión de información de proyecto’, engloba todas las funcionalidades relacionadas con la gestión de la información de los datos (identificativos y contenidos en) un proyecto. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.

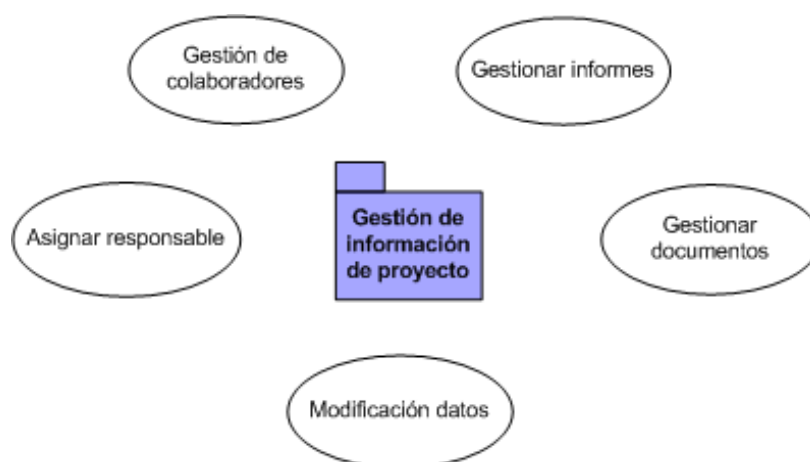



Figura 68. Paquete de análisis ‘gestionar información del proyecto’

4.2.4 Paquetes de análisis ‘Gestión información de tarea’

El paquete de análisis, ‘Gestión de información de tarea’, engloba todas las funcionalidades relacionadas con la gestión de la información de los datos (identificativos y contenidos en) una tarea. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.



Figura 69. Paquete de análisis ‘gestión información de tarea’

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

4.2.5 Paquetes de análisis ‘Gestión de informes’

El paquete de análisis, ‘Gestión de informes’, engloba todas las funcionalidades relacionadas con la gestión de los informes. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.



Figura 70. Paquete de análisis ‘gestionar informes’

4.2.6 Paquetes de análisis ‘Gestión de incidencias’


El paquete de análisis, ‘Gestión de incidencias’, engloba todas las funcionalidades relacionadas con la gestión y generación de ficheros log con el conjunto de incidencias ocurridas a lo largo del tiempo en nuestro sistema. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.



Figura 71. Paquete de análisis ‘gestión de incidencias’

4.2.7 Paquetes de análisis ‘Acceso a repositorio de persistencia de datos’

El paquete de análisis, ‘Acceso a repositorio de persistencia de datos’, engloba todas las funcionalidades relacionadas con la persistencia de aquellas entidades del sistema que necesitan de ello a lo largo de su ciclo de vida. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

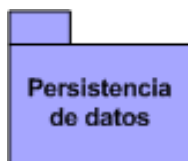


Figura 72. Paquete de análisis 'acceso a repositorio de persistencia de datos'

4.2.8 Paquetes de análisis 'Internalización de entidades visibles'

El paquete de análisis, 'Internalización de entidades visibles' es el responsable de desarrollar todas aquellas funcionalidades necesarias para que nuestro sistema sea internacionable. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.

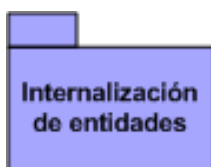



Figura 73. Paquete de análisis 'internalización de entidades visibles'

4.2.9 Paquetes de análisis 'Gestión de personas'

El paquete de análisis, 'Gestión de personas', engloba todas las funcionalidades relacionadas con la gestión de personas en la aplicación. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.

	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

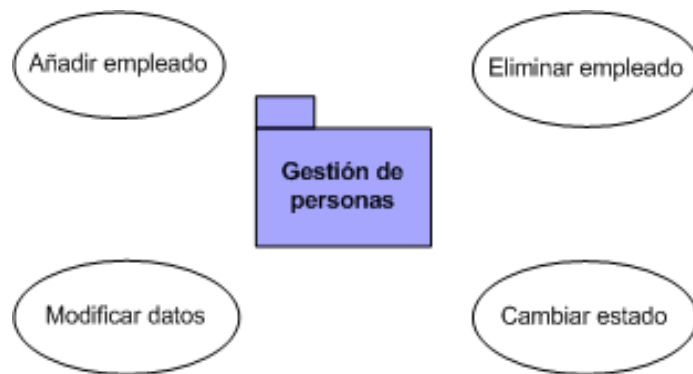


Figura 74. Paquete de análisis 'gestión de personas'

4.2.10 Paquetes de análisis 'Gestión de usuarios'

El paquete de análisis, 'Gestión de usuarios', engloba todas las funcionalidades relacionadas con la gestión de usuarios en la aplicación. La identificación de este paquete de análisis se ha realizado mediante la agrupación de los siguientes casos de uso de la fase anterior de identificación de casos de uso.

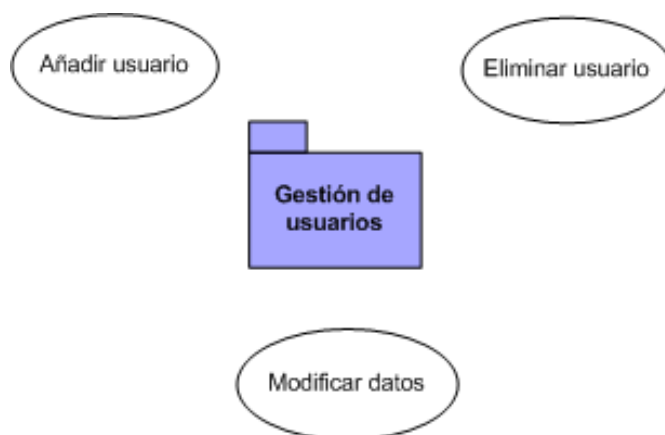



Figura 75. Paquete de análisis 'gestión usuarios'

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

5. REQUERIMIENTOS ESPECIALES

En esta sección se describen requerimientos especiales identificados durante la fase de análisis, y que son importantes para el sistema.

5.1 *Persistencia*

El sistema a desarrollar debe garantizar la persistencia de algunos objetos de los identificados en la fase de análisis. La definición y especificación de las clases que necesitarán de esta propiedad de persistencia del sistema será concretado y definido en la fase del diseño del sistema.


En esta fase de análisis, se ha definido el paquete 'Acceso a repositorio de persistencia de datos', el cual contendrá necesaria para garantizar dicha persistencia.

5.2 *Tolerancia a fallos*

El sistema debe ser capaz de recuperarse de una acción no permitida, y volver a un estado estable y válido. El paquete de análisis 'Gestión de incidencias' será el que contenga la funcionalidad necesaria para que el sistema sea tolerante a fallos.

5.3 *Internacionalización*

El sistema debe permitir la internacionalización del mismo, es decir, debe ser posible modificar fácilmente todos los textos mostrados por el sistema en función del idioma del usuario. El paquete de análisis 'Internacionalización de entidades visibles' será el que contenga la funcionalidad correspondiente a ese punto.

 endalia®	Sistema para gestión de proyectos	Versión: 1.3
	Análisis	Fecha: 25/11/2009
	ANALISIS.doc	

6. BIBLIOGRAFÍA

6.1 Referencia

- [IGJ,2000] I. Jacobson, G. Booch, J. Rumbaugh. 2000. "El proceso unificado de desarrollo de Software". Pearson Education
- [IGJ, 1999] I. Jacobson, G. Booch, J. Rumbaugh. 1999. "El lenguaje unificado de modelado. Manual de referencia". Ed. Addison Wesley.
- [FOW, 1999] Martin Fowler. 1999. "UML Distilled". Addison-Wesley 1999 (2nd ed.).


6.2 Referencias web

- [Ref. Web 1] <http://www.uml.org>
- [Ref. Web 2] <http://www.wikipedia.org>
- [Ref. Web 3] <http://www.rational.com>

Sistema para la gestión de proyectos

Diseño

Versión 1.0 – Fecha: 13/01/2010

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
01/12/2009	1.0	Documento de diseño	José Ángel Peribáñez
08/01/2010	1.1	Revisión del documento	Fernando Cortés
13/01/201	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España




 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	5
1.1 PROPÓSITO DEL DOCUMENTO	5
1.2 ALCANCE DEL DOCUMENTO	5
1.3 ACRÓNIMOS	5
1.4 DEFINICIONES	6
1.5 REFERENCIAS	6
1.6 RESUMEN	6
2. DESCRIPCIÓN DEL PROCESO	8
3. CONSIDERACIONES INICIALES	9
3.1 INTRODUCCIÓN	9
3.2 ESPECIFICACIONES TECNOLÓGICAS	9
3.3 ESPECIFICACIONES DE DISEÑO	9
3.4 PLATAFORMA .NET	9
3.4.1 NET FRAMEWORK	10
3.4.2 COMMON LANGUAGE RUNTIME	11
3.4.3 BIBLIOTECA DE CLASES BASE (BCL)	12
3.5 SQL SERVER	13
3.6 NHIBERNATE	14
4. DISEÑO DE LA ARQUITECTURA	16
4.1 INTRODUCCIÓN	16
4.2 ESTRUCTURA GENERAL DEL SISTEMA	16
4.3 ESTRUCTURA DE CAPAS DEL SISTEMA	16
4.4 ESTRUCTURA DE SUBSISTEMAS	17
4.4.1 SUBSISTEMA DE CONFIGURACIÓN	17
4.4.2 SUBSISTEMA DE GESTIÓN DE PERSONAS	17
4.4.3 SUBSISTEMA DE USUARIOS	18
4.4.4 SUBSISTEMA DE PROYECTOS	18
4.4.5 SUBSISTEMA DE INFORMES	18
4.4.6 SUBSISTEMA DE ACCESO A BASE DE DATOS	18
4.4.7 SUBSISTEMA DE GESTIÓN DE INCIDENCIAS	18
4.4.8 SUBSISTEMA DE INTERNALIZACIÓN DE ENTIDADES VISIBLES	19
5. CLASES DEL SISTEMA	20
5.1 INTRODUCCIÓN	20
5.2 CLASES DE INTERFAZ	20
5.2.1 CLASES DEL SUBSISTEMA DE CONFIGURACIÓN	21
5.2.2 CLASES DEL SUBSISTEMA DE GESTIÓN DE PERSONAS	21
5.2.3 CLASES DEL SUBSISTEMA DE GESTIÓN DE USUARIOS	22

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

5.2.4	CLASES DEL SUBSISTEMA DE GESTIÓN DE PROYECTOS	22
5.2.5	CLASES DE SUBSISTEMA DE GESTIÓN DE INFORMES	23
5.3	CLASES DE ACCESO A DATOS	23
5.3.1	GEN_ADMINISTRATIONITEM.CS:	24
5.3.2	GEN_SYSTEMOPTION:	24
5.3.3	GEN_USERS:	24
5.3.4	ORH_EMP_EMPLOYEE.CS:	25
5.3.5	PRJ_ACTIONPLANITEM:	26
5.3.6	PRJ_ACTIONPLANAREA:	27
5.3.7	PRJ_ACTIONPLANACTIONS:	27
5.3.8	PRJ_ACTIONPLANDOCUMENTS:	28
5.3.9	PRJ_ACTIONPLANWORKTIME:	29
5.3.10	PRJ_ACTIONPLANINDICATORS:	29
5.3.11	PRJ_ACTIONPLANINDICATORSHISTORY:	29
5.3.12	PRJ_ACTIONPLANINDICATORSTARGET:	30
5.3.13	PRJ_ACTIONPLANITEMAPPROVALHISTORY:	30
5.3.14	PRJ_ACTIONPLANITEMINTERNALHISTORY:	30
5.3.15	PRJ_ACTIONPLANITEMPROGRESS:	31
5.3.16	R_ACTIONPLANITEM_EMPLOYEE:	31
5.3.17	R_GEN_USER_PRIVILEGE:	32
5.3.18	R_GEN_USER_SYSTEMOPTION:	32
6.	<u>DISEÑO DE LA BASE DE DATOS</u>	33
6.1	INTRODUCCIÓN	33
6.2	DISEÑO GENERAL DE LA BASE DE DATOS	33
6.2.1	USUARIOS:	33
6.2.2	EMPLEADOS	35
6.2.3	DIAGRAMA DE PLANES ACCIÓN (AREAS, PROYECTOS, TAREAS)	36
6.3	TABLAS DE LA BASE DATOS	37
6.3.1	USUARIOS	37
6.3.2	EMPLEADOS	38
6.3.3	PLANES DE ACCION (AREAS, PROYECTOS, TAREAS)	39
6.3.4	ADMINISTRACIÓN	46
6.3.5	RELACIONES	46
7.	<u>PROTOTIPADO DE LA INTERFAZ</u>	48
7.1	PROTOTIPOS	48
8.	<u>BIBLIOGRAFÍA</u>	52
8.1	REFERENCIAS	52
8.2	REFERENCIAS WEB	52

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

1. INTRODUCCIÓN

1.1 *Propósito del documento*

El objetivo del diseño es obtener, a partir del análisis, un punto de partida para actividades de implementación, capturando los requisitos, interfaces y clases a partir de las especificaciones de requisitos y análisis previos.

1.2 *Alcance del documento*

Este documento comprende la fase de diseño del sistema, y describe los resultados obtenidos durante esta fase, detallando cada uno de los artefactos y actividades que son necesarios realizar en esta fase del desarrollo.

1.3 *Acrónimos*

ACID: Atomicity- Consistency-Isolation-Durability.

ADO: ActiveX Data Objects.

API: Application Programming Interface.

ASP: Active Server Pages.

BCL: Base Class Library.

CIL: Common Intermediate Language.

CLI: Common Language Infrastructure.

CLR: Common Language Runtime.

CLS: Common Language Specification.

DDL: Data Definition Language.

DML: Data Manipulation Language.

ECMA: European Computer Manufacturer Association.

GDI: Graphics Device Interface.

GUI: Graphic User Interface.

IEC: International Electrotechnical Commission.

ISO: International Organization for Standardization.

JIT: Just-in-Time.


LGPL: Lesser GNU Public License.

MSIL: Microsoft Intermediate Language.

SGBD: Sistema Gestor de Bases de Datos.

SGP: Sistema Gestor de proyectos.

SOAP: Simple Object Access Protocol.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

TCP/IP: Transmission Control Protocol / Internet Protocol.

T-SQL: Transact – Structured Query Language.

WSDL: Web Services Descriptor Language.

XML: Extensive Markup Language.

1.4 Definiciones

- Principio ACID: es el conjunto de propiedades de una base de datos que aseguran la realización de transacciones seguras. En concreto, ACID es un acrónimo de *Atomicity, Consistency, Isolation and Durability* (Indivisibilidad, Consistencia, Aislamiento y Durabilidad en castellano).
 - Indivisibilidad: es la propiedad que asegura que todas las tareas incluidas en la transacción han sido realizadas, o bien que ninguna de ellas lo ha sido.
 - Consistencia: es la propiedad que asegura que la base de datos está en un estado coherente antes del comienzo de la transacción, y que queda en otro estado coherente (sea el mismo u otro) después de la finalización de la transacción.
 - Aislamiento: es la propiedad que asegura que una operación externa a la transacción no puede acceder a un estado intermedio de los producidos durante la misma.
 - Durabilidad: es la propiedad que asegura que una vez realizada la transacción con éxito, ésta persistirá y no se podrá deshacer aunque falle el sistema

1.5 Referencias


En este documento se han utilizado las siguientes referencias a otros documentos del proyecto:

- ESTMERCADO: estudio de mercado
- MODELODENEGOCIO: documento de modelo de negocio.
- ESPECIFICACIONREQUISITOS: documento de especificación de requisitos.
- ANALISIS: documento de análisis del sistema.

1.6 Resumen


Este documento describe el proceso de diseño del SGP. Se compone de ocho apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describe el proceso de diseño seguido para la confección de este documento.
- Apartado 3: Se describen las decisiones y restricciones iniciales del diseño, y se comparan y describen diferentes alternativas.
- Apartado 4: Descripción de la arquitectura del sistema tanto a nivel físico como a nivel

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	


de organización del mismo.

- Apartado 5: Detalle de las clases identificadas.
- Apartado 6: Detalle del diseño de la base de datos.
- Apartado 7: Prototipado del interfaz de las diferentes pantallas con las que ha de interactuar el usuario del sistema.
- Apartado 8: Bibliografía y referencias Web utilizadas durante esta fase del proyecto.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

2. DESCRIPCIÓN DEL PROCESO

Partiendo de las entidades, casos de uso y paquetes identificados en la fase de análisis, y de los requisitos del sistema identificados en la especificación de requisitos, se procede a la descripción de la arquitectura del sistema, para su posterior implementación. A partir de los casos de uso, se identifican las necesidades de interacción entre el usuario y el sistema, y se define la estructura del interfaz, así como los prototipos del mismo. A partir de las entidades, se describen las tablas necesarias en base de datos, así como sus clases de acceso a datos.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

3. CONSIDERACIONES INICIALES

3.1 Introducción

En esta sección se describen las primeras decisiones y especificaciones en el diseño del SGP, que sirven como base para el diseño del resto del sistema. Asimismo, se describen las principales características de las distintas tecnologías utilizadas.

3.2 Especificaciones tecnológicas

El proyecto que nos ocupa se lleva a cabo en el marco de la empresa Endalia. Este hecho condiciona la tecnología a utilizar que, evidentemente, debe ser la misma usada en el resto de aplicaciones llevadas a cabo en dicha empresa, con el objeto de hacerlas compatibles y fácilmente integrables.

Es por ello que el SGP se va a desarrollar en la plataforma .NET de Microsoft, usando el lenguaje de programación C# y el gestor de base de datos Microsoft SQL Server.


3.3 Especificaciones de diseño

A partir de los requisitos identificados en el documento de análisis de requisitos, podemos definir como características necesarias del sistema las siguientes:

- Escalabilidad: el sistema debe soportar más carga de trabajo sin necesidad de modificar el software.
- Extensibilidad: el sistema debe soportar la adición de nuevos componentes y funcionalidades sin que ello afecte al resto de componentes.
- Usabilidad: el sistema debe poder ser manejado de forma intuitiva.
- Acceso a base de datos de forma transaccional, cumpliendo el principio ACID.
- Seguridad: El sistema debe ser fiable, tanto a nivel de autenticación, como de autorización, y debe mantener la privacidad de la información confidencial.
- Rendimiento: El sistema debe soportar un incremento en la carga de trabajo sin que ello repercuta notablemente en el usuario.

3.4 Plataforma .NET

Microsoft .NET es, de acuerdo con la definición de Microsoft, una plataforma que comprende servidores, clientes y servicios. Consiste en un conjunto de aplicaciones como Visual Studio .NET, los servicios .NET, etc. Esta plataforma es una implementación basada en estándares abiertos como SOAP, WSDL, C#, etcétera. Desde el punto de vista del programador, el entorno .NET ofrece un solo entorno de desarrollo para todos los lenguajes que soporta (por ejemplo, Visual Basic, C++, C#, Visual J#, Fortran, Cobol...).

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

3.4.1 Net Framework

El “framework” o marco de trabajo constituye la base de la plataforma .Net (Figura 1), y denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.

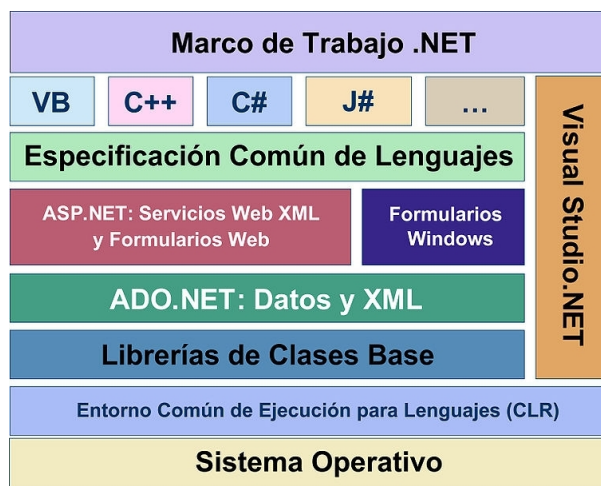



Figura 1: .NET Framework

Bajo el nombre .NET Framework, o Marco de Trabajo .NET, se encuentran reunidas una serie de normas, entre las cuales se encuentran:

- La norma que define las reglas que debe seguir un lenguaje de programación para ser considerado compatible con el marco de trabajo .NET (ECMA-335, ISO/IEC 23271). Por medio de esta norma se garantiza que todos los lenguajes desarrollados para la plataforma ofrezcan al programador un conjunto mínimo de funcionalidad, y compatibilidad con todos los demás lenguajes de la plataforma.
- La norma que define el lenguaje C# (ECMA-334, ISO/IEC 23270). Éste es el lenguaje insignia del marco de trabajo .NET, y pretende reunir las ventajas de lenguajes como C/C++ y Visual Basic en un solo lenguaje.
- La norma que define el conjunto de funciones que debe implementar la librería de clases base (BCL, siglas en inglés) (incluido en ECMA-335, ISO/IEC 23271). Tal vez el más importante de los componentes de la plataforma, esta norma define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma, independientemente del sistema operativo para el cual hayan sido implementadas.

Los principales componentes del marco de trabajo son:

- El conjunto de lenguajes de programación.
- La Biblioteca de Clases Base o BCL.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

- El entrono Común de Ejecución para Lenguajes o CLR.

Debido a la publicación de la norma para la infraestructura común de lenguajes (CLI por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación (C#, Visual Basic, C++, Perl, Python, Fortran...) y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos.

3.4.2 Common Language Runtime


El CLR (Figura 2) es el verdadero núcleo del Framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo.



Figura 2: Common Language Runtime (CLR)

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio (MSIL, Microsoft Intermediate Language), similar al BYTECODE de Java. Para generar dicho código, el compilador se basa en el Common Language Specification (CLS), que determina las reglas necesarias para crear ese código MSIL compatible con el CLR.

Para ejecutarse se necesita un segundo paso: un compilador JIT (Just-in-Time), que es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue con .NET independencia de la plataforma hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. El código ejecutable obtenido se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.


 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

3.4.3 Biblioteca de Clases Base (BCL)

La Biblioteca de Clases Base (Base Class Library, BCL) proporciona las clases básicas predefinidas que manejan la mayoría de las operaciones que se encuentran involucradas en el desarrollo de aplicaciones, incluyendo entre otras:

- Interacción con los dispositivos periféricos.
- Manejo de datos (ADO.NET).
- Administración de memoria.
- Cifrado de datos.
- Transmisión y recepción de datos por distintos medios (XML, TCP/IP).
- Administración de componentes Web que corren tanto en el servidor como en el cliente (ASP.NET).
- Manejo y administración de excepciones.
- Manejo del sistema de ventanas.
- Herramientas de despliegue de gráficos GDI.
- Herramientas de seguridad e integración con la seguridad del sistema operativo.
- Manejo de tipos de datos unificado.
- Interacción con otras aplicaciones.
- Manejo de cadenas de caracteres y expresiones regulares.
- Operaciones aritméticas.
- Manipulación de fechas, zonas horarias y periodos de tiempo.
- Manejo de arreglos de datos y colecciones.
- Manipulación de archivos de imágenes.
- Aleatoriedad.
- Generación de código.
- Manejo de idiomas.
- Auto descripción de código.
- Interacción con el API Win32 o Windows API.
- Compilación de código.

Esta funcionalidad se encuentra organizada por medio de espacios de nombre jerárquicos en lo que se denomina Namespace, como se puede apreciar en la siguiente figura (Figura 3):

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

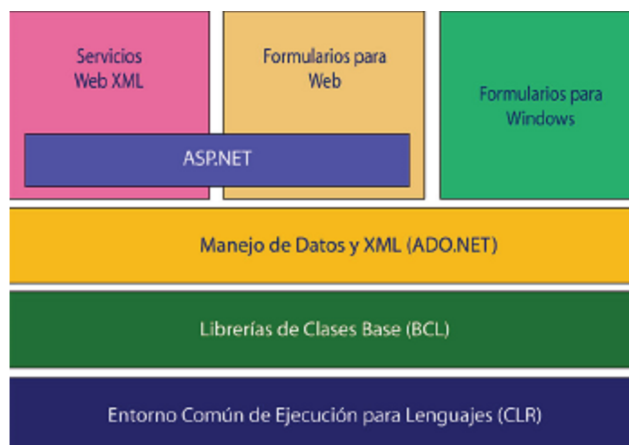



Figura 3: esquema de distribución jerárquica de namespaces

3.5 Sql Server

Es un SGBD, para bases de datos relacionales, desarrollado por Microsoft. Sus principales características son:

- Soporte para transacciones (bajo el principio ACID).
- Escalabilidad.
- Estabilidad.
- Seguridad.
- Soporta procedimientos almacenados.
- Entorno gráfico que permite ejecutar comandos DDL y DML.
- T-SQL como lenguaje de consultas nativo.

Una base de datos de SQL Server, es una colección de tablas con columnas con tipo definido, más otros objetos como restricciones, vistas, procedimientos almacenados o índices. Puede contener un máximo de 2^{31} objetos. El espacio de almacenamiento está dividido en páginas de 8KB, que es la unidad básica de entrada-salida para una operación de SQL Server. Las filas de cada tabla se almacenan físicamente en fichero o bien en un montículo (*heap*) o bien en un árbol-B. Los índices (que son estructuras para acelerar el acceso a datos en las consultas) definidos son almacenados siempre en árboles-B. Hay dos tipos de índices en SQL Server, índices agregados, en los que se almacena los datos de la fila indexada en las hojas del árbol, e índices no agregados, que en las hojas del árbol-B almacenan una referencia a la hoja del índice agregado correspondiente, o bien una referencia a la página correspondiente. Sólo puede haber un índice agregado por tabla, y éste habitualmente es la clave primaria de ésta.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

3.6 NHibernate

NHibernate es la conversión de Hibernate de lenguaje Java a C# para su integración en la plataforma .NET.

NHibernate es un Framework de persistencia, es decir, provee herramientas que facilitan la tarea de persistir objetos (i.e. almacenar el estado de un objeto con el fin de recuperarlo en el futuro). La motivación principal de NHibernate es abstraer por completo al desarrollador de la base de datos asociada al proyecto en desarrollo, es decir, el desarrollador debe pensar que sólo trabaja con objetos, los cuales puede guardar en una base de datos utilizando métodos de los mismos objetos, pero sin necesidad de escribir sentencias SQL en su código.

Al usar NHibernate para el acceso a datos, el desarrollador se asegura de que su aplicación es “agnóstica” en cuanto al motor de base de datos a utilizar en producción, pues NHibernate soporta los más habituales en el mercado: MySQL, PostgreSQL, Oracle, MS SQL Server, etc. Sólo se necesita cambiar una línea en el fichero de configuración para que podamos utilizar una base de datos distinta.

NHibernate es software libre, distribuido bajo los términos de la LGPL (Licencia Pública General Menor de GNU).


Dada una descripción XML de las entidades y relaciones de la base de datos subyacente, NHibernate genera automáticamente código SQL para cargar y almacenar los objetos.

NHibernate soporta “persistencia transparente”, es decir, las clases objeto no tienen que seguir un modelo de programación restrictivo. Las clases de persistencia no necesitan implementar ninguna interfaz o heredar características de una clase base determinada. Esto hace posible diseñar la lógica de negocio usando objetos .NET planos, y un lenguaje orientado a objetos.

Las características más destacadas de NHibernate son:

- Modelo de programación natural. NHibernate soporta lenguajes orientados a objetos: herencia, polimorfismo, composición y colecciones de .NET, incluyendo colecciones genéricas.
- .NET nativo. La API de NHibernate usa las convenciones y lenguajes de .NET.
- Permite especificar el código SQL que NHibernate debe usar para persistir los objetos. Soporta procedimientos almacenados para Microsoft SQL Server.

La siguiente figura (Figura 4) representa de una manera simple la arquitectura de NHibernate:

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

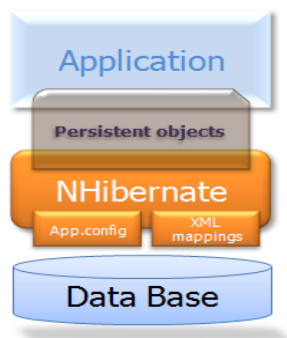


Figura 4: Arquitectura de NHibernate

Podemos observar que la aplicación trabajará con objetos persistentes, pero sin necesidad de comunicarse directamente con la base de datos. En su lugar, la comunicación será con el Framework NHibernate, el cual se compone de una sección de configuración (app.config) y un conjunto de mapeos “objeto-relacionales”. Utilizando estos elementos, NHibernate se comunicará con la base de datos y realizará las acciones requeridas por los objetos persistentes (inserción, actualización, borrado, selección).

En la siguiente figura (Figura 5) se observa una vista más detallada de la arquitectura usada.

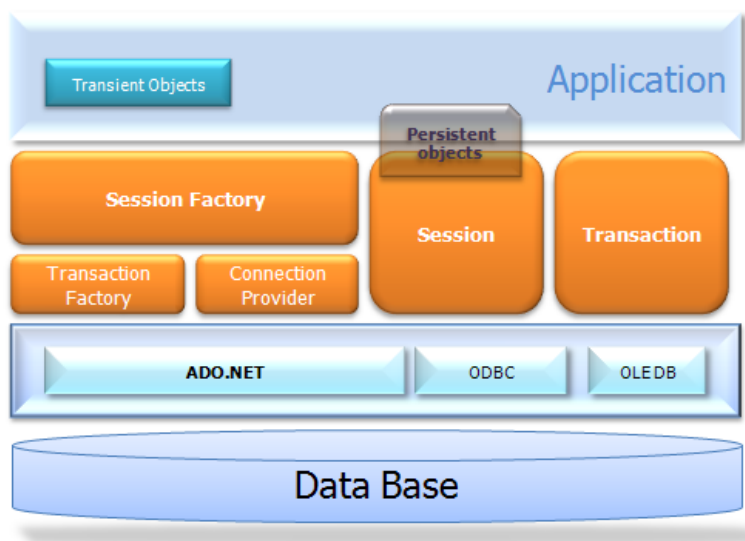



Figura 5: Arquitectura NHibernate detallada

En esta arquitectura, NHibernate provee el control de las transacciones y de las conexiones ADO.NET, que no están expuestas a la aplicación, aunque pueden ser extendidas o implementadas por los desarrolladores.

En el documento de implementación se mostrarán las implicaciones del uso de NHibernate en la codificación propiamente dicha de la aplicación.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

4. DISEÑO DE LA ARQUITECTURA

4.1 Introducción

En esta sección se describe la estructura del sistema, tanto desde el punto de vista físico como lógico, detallando las decisiones tomadas en cada momento y sus motivos o restricciones que llevan a tomar esas decisiones.

4.2 Estructura general del sistema

La estructura general del sistema está compuesta por dos partes claramente diferenciadas:

- La aplicación de Gestión de Proyectos (SGP). Los usuarios acceden al sistema, que estará instalado localmente en su equipo.
- La base de datos. Es el repositorio de datos de la aplicación, implementado desde el sistema gestor de bases de datos SQL Server 2005.

En la siguiente figura (Figura 6) vemos gráficamente la estructura general del sistema:

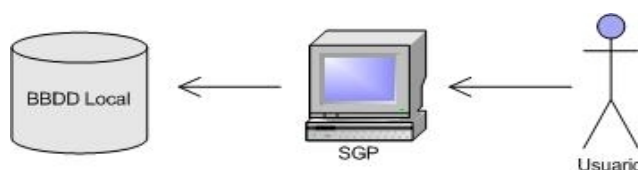



Figura 6: Estructura general del sistema

4.3 Estructura de capas del sistema

Una posible forma de organizar un sistema de un tamaño considerable es agrupando funcionalidades que comparten la misma naturaleza, funcionalidad y estructura en capas. De esta manera, se consigue que cambios en algún componente (por ejemplo, presentación) no afecten al resto de elementos del sistema, una óptima escalabilidad y mejorar el rendimiento.

El sistema está basado en la siguiente arquitectura multicapa (Figura 7):

- Capa de presentación. Es la capa que se encarga de crear el interfaz gráfico y de gestionar las interacciones del usuario con el sistema. Esto se consigue mediante Winforms en C# y controles de usuario.
- Capa de lógica de negocio. Contiene los objetos que representan los datos almacenados en el repositorio de datos, así como la lógica necesaria para procesarlos. En nuestro sistema se corresponde con las clases de acceso a datos.
- Capa de integración. Contiene objetos que automatizan el acceso a datos. Esto se corresponde con los procedimientos almacenados en la base de datos y con NHibernate.
- Capa de datos. Contiene los sistemas de información de la aplicación, habitualmente una base de datos. En nuestro sistema se corresponde con la base de datos y con los

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

archivos de recursos.



Figura 7: Estructura de capas del sistema

4.4 Estructura de subsistemas

Los subsistemas son un medio para organizar el modelo en partes más pequeñas y manejables.

Una de las opciones para la realización de esta actividad se basa en la identificación de subsistemas de diseño a partir de los paquetes definidos en la fase de análisis. La correspondencia no siempre debe ser uno a uno, ya que intervienen ciertos condicionantes que la limitan, pero sí constituye un punto de partida para iniciar la identificación.

A continuación, se presentan los subsistemas de diseño que forman el SGP.

4.4.1 Subsistema de configuración


Subsistema en el cual se desarrollan las funcionalidades identificadas en el paquete 'Configuración', relacionadas con la existencia de la entidad 'parámetro'. Dichas funcionalidades son:

- Creación de nuevos parámetros.
- Edición de parámetros ya existentes.
- Eliminación de parámetros.

4.4.2 Subsistema de gestión de personas

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en el paquete 'Gestión de Personas', relacionadas con la existencia de la entidad 'persona' (también usamos el término de 'empleado' indistintamente para referirnos a la misma entidad). Dichas funcionalidades son:

- Adición de nuevas personas al sistema.
- Edición de datos de personas.
- Cambio de estado de personas.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

4.4.3 Subsistema de usuarios

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en el paquete 'Gestión de Usuarios', relacionadas con la existencia de la entidad 'usuario' (también usamos el término de 'empleado' indistintamente para referirnos a la misma entidad). Dichas funcionalidades son:

- Adición de nuevo usuario asociado a una persona al sistema.
- Edición de datos de usuario.
- Eliminación de usuario.

4.4.4 Subsistema de proyectos

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en los paquetes 'Gestión de información de Área', 'Gestión de información de Proyecto', 'Gestión de información de Tarea' relacionadas con la existencia de la entidad 'usuario' (también usamos el término de 'empleado' indistintamente para referirnos a la misma entidad). Dichas funcionalidades son:

- Adición de nuevos áreas, proyectos y tareas.
- Edición de datos de áreas, proyectos y tareas.
- Distintas acciones sobre áreas, proyectos y tareas (Eliminación cancelación, activación duplicado,...).

4.4.5 Subsistema de Informes

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en el paquete 'Gestión de informes', que son:


- Generación de informes agregados
- Generación de informes directos

4.4.6 Subsistema de acceso a base de datos

En este subsistema se desarrollan las funcionalidades necesarias para gestionar el acceso a base de datos de manera transparente y eficiente.


4.4.7 Subsistema de gestión de incidencias

Subsistema en el cual se desarrolla la funcionalidad identificada en el paquete 'Gestión de incidencias', relacionada con el control de los eventos y errores producidos por el sistema.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

4.4.8 Subsistema de internalización de entidades visibles

En este subsistema se desarrollan las funcionalidades necesarias para que la aplicación pueda adaptar sus contenidos de manera automática a la cultura en que se ejecute.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

5. CLASES DEL SISTEMA

5.1 Introducción

En este apartado se detallan las clases del SGP. Estas clases se dividen en dos grupos:

- Clases de interfaz: son las encargadas de crear la GUI y gestionar las interacciones del usuario con el sistema.
- Clases de acceso a datos: son las encargadas de gestionar la persistencia de los datos del sistema y la interacción con la base de datos.

5.2 Clases de interfaz

En esta sección se van a analizar las clases de interfaz del SGP. En la siguiente figura se observa el esquema general de clases de interfaz del sistema (Figura 8):

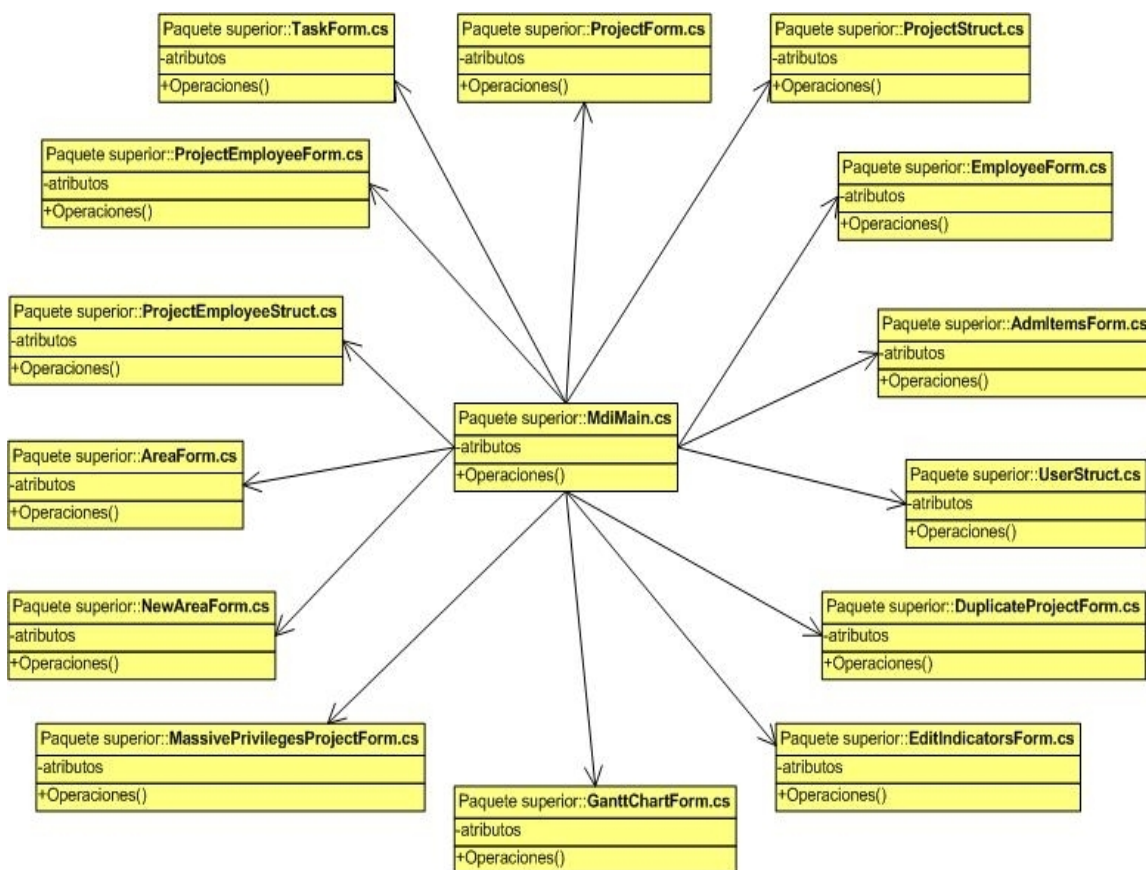



Figura 8: Esquema general de clases del sistema

Como se aprecia en la figura, la clase MDIMain.cs es la que da acceso a los diferentes módulos del sistema.

En los siguientes subapartados se describirán el resto de clases de interfaz del sistema, divididas en función del subsistema al que pertenezcan.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Para referirnos a las clases de interfaz, usaremos también indistintamente el nombre de formulario o WinForm.

5.2.1 Clases del subsistema de configuración

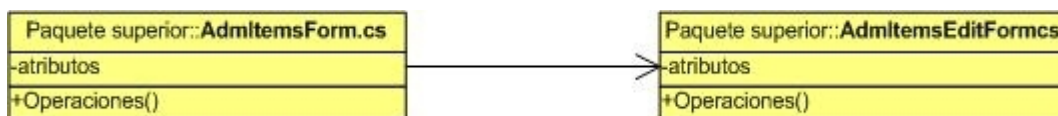


Figura 9: Clases del subsistema de configuración

- AdmItemsForm.cs: formulario que muestra la estructura de parámetros del sistema y permite realizar operaciones de alta, edición y eliminación de parámetros.
- AdmItemsEditForm.cs: formulario que permite la visualización y modificación de los datos de un parámetro. Se usa tanto para el alta de parámetros como para su edición.

5.2.2 Clases del subsistema de gestión de personas

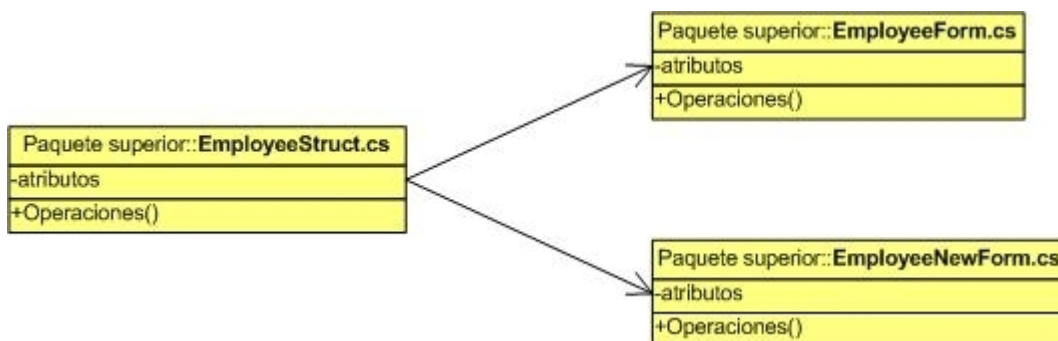



Figura 10: Clases del subsistema de gestión de personas

- EmployeeStruct.cs: formulario que muestra la estructura de empleados de la organización. Permite realizar operaciones de alta, baja, edición y cambio de estado de empleados.
- EmployeeForm.cs: formulario que muestra los datos de un empleado, permitiendo su edición.
- EmployeeNewForm.cs: formulario que permite dar de alta a un empleado.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

5.2.3 Clases del subsistema de gestión de usuarios

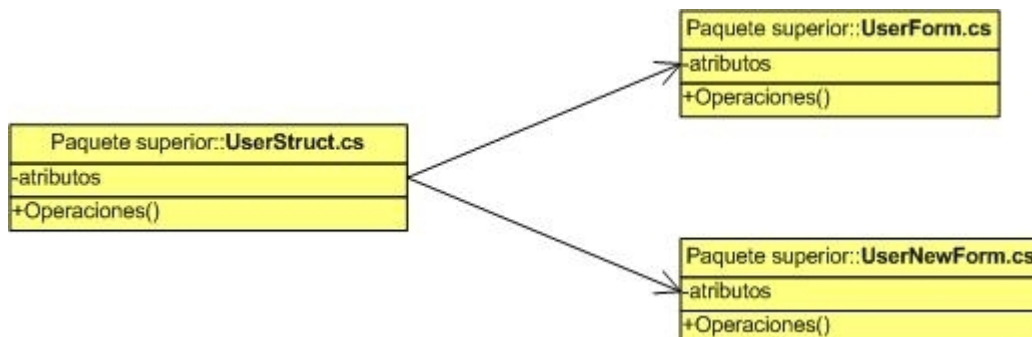


Figura 11: Clases del subsistema de gestión de personas

- UserStruct.cs: formulario que muestra la estructura de usuarios de la organización. Permite realizar operaciones de alta, baja, edición y cambio de estado de usuarios.
- UserForm.cs: formulario que muestra los datos de un usuarios, permitiendo su edición.
- UserNewForm.cs: formulario que permite dar de alta a un usuario.

5.2.4 Clases del subsistema de gestión de proyectos

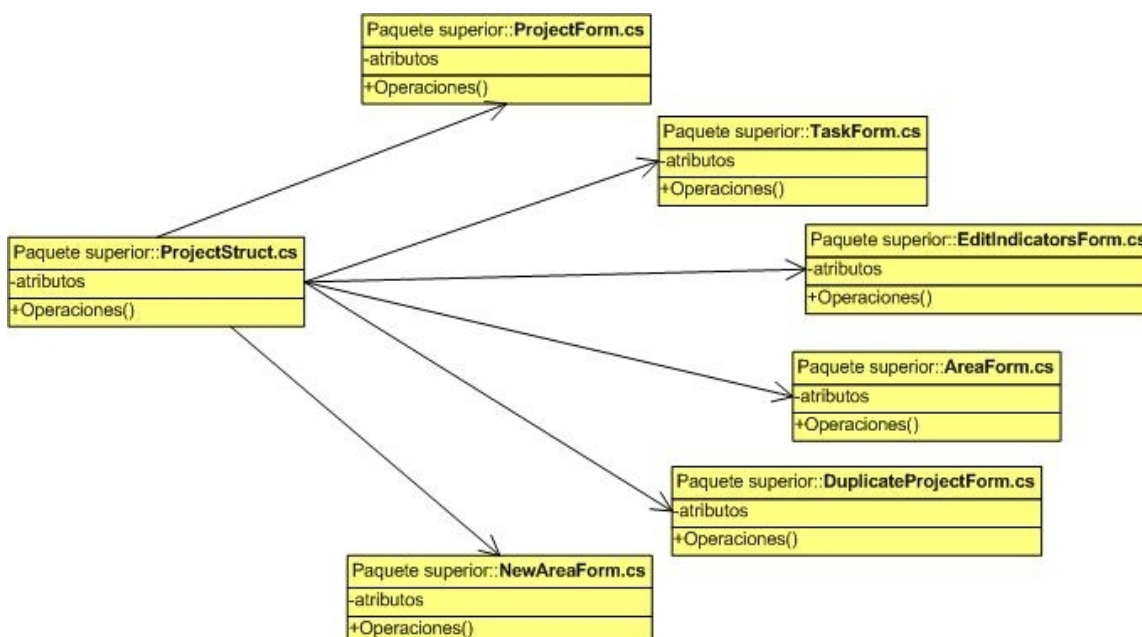



Figura 12: Clases del subsistema de gestión de proyectos

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

- ProjectStrcut.cs: formulario que muestra la estructura de áreas, proyectos, tareas, y un listado con la información de esa estructura en la parte derecha. Además contiene la barra de herramientas con todas las operaciones que se puede ejecutar sobre los ítems. También muestra en la parte derecha los distintos informes directos que genera la aplicación.
- ProjectForm.cs: formulario que contiene toda la información asociada a un proyecto clasificada en distintas pestañas, alguna de la información que contiene es: características, recursos, indicadores económicos, documentos asociados....
- TaskForm.cs: formulario que contiene toda la información asociada a una tarea clasificada en distintas pestañas, alguna de la información que contiene es: recursos, documentos asociados, objetivos...
- AreaForm.cs: formulario que contiene la información asociada a un área, esa información es nombre, responsable y descripción.
- NewAreaForm.cs: formulario que permite dar de alta un nuevo área.
- EditIndicatorsForm.cs: formulario que permite editar los indicadores (económicos, no económicos o de cliente) asociados a un proyecto.
- DuplicateProjectForm.cs: formulario que permite duplicar proyectos y/o tareas y sus tareas descendientes.

5.2.5 Clases de subsistema de gestión de informes

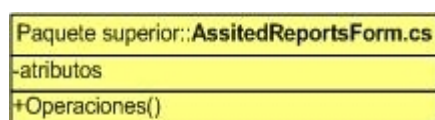



Figura 13: Clases del subsistema de gestión de informes

- AssitedReportsForm.cs: formulario que permite la generación de un informe mediante la selección por parte del usuario de los campos a mostrar mediante unos filtros de información.

5.3 Clases de acceso a datos

En esta sección se van a analizar las clases de acceso a datos del SGP. Se describirán los atributos de las clases de acceso a datos más representativas.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

5.3.1 Gen_AdministrationItem.cs:

Es la clase de acceso a datos que representa a la entidad 'Parámetro'. Sus atributos son:

Atributo	Descripción
_ID	Identificador del parámetro
_name	Nombre del parámetro
_parent	Referencia al parámetro padre
_type	Tipo de parámetro

5.3.2 Gen_SystemOption:


Es la clase de acceso a datos que se usa para cargar distintas opciones del programa, cada opción del sistema da lugar a un comportamiento diferente. Sus atributos son:

Atributo	Descripción
_name	Identificador de la opción del sistema
_description	Descripción de la opción de sistema
_value	Valor

5.3.3 Gen_Users:

Es la clase de acceso a datos que representa a la entidad 'Usuario'. Sus atributos son:

Atributo	Descripción
_ID	Identificador del usuario
_mainJobID	Referencia al nivel laboral al que pertenece
_Language	Referencia al language asociado con el usuario
_Login	Nombre de usuario dentro de la aplicación
_Password	Contraseña del usuario para acceder al sistema
_IsAdmin	Booleano que indica si es administrador
_canViewAllStructure	Booleano que indica si puede ver toda estructura
_JobID	Referencia al puesto laboral al que pertenece
_LastLogin	Fecha último acceso al sistema
_LastUpdate	Fecha de la última modificación de datos
_Inactive	Booleano que indica si es inactivo
_CanBelImpersonated	Booleano que indica si es externo
_FailedLoginAttempts	Numero de intentos fallidos de acceso al sistema


 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

_dateCreated	Fecha de creación del usuario
_dateNotified	Fecha de confirmación de usuario
_InactiveDateStart	Fecha desde la que es usuario inactivo
_IsExternalUser	Booleano que indica si es externo

5.3.4 Orh_Emp_Employee.cs:

Es la clase de acceso a datos que representa a la entidad 'Empleado' (o 'Persona'). Sus atributos son:

Atributo	Descripción
_ID	Identificador del empleado
_workLevel	Referencia al nivel laboral al que pertenece
_type	Referencia a la categoría interna del empleado
_group	Referencia al grupo de cotización del empleado
_contractType	Referencia al tipo de contrato que posee el empleado
_salaryGroup	Referencia a la categoría profesional a la que pertenece el empleado
_costCenter	Referencia al centro de coste del empleado
_workCenter	Referencia al centro de trabajo del empleado
_workSubdivision	Referencia a la subdivisión a la que pertenece el empleado
_nationality	Referencia a la nacionalidad del empleado
_professionalGroup	Referencia al grupo profesional del empleado
_country	País de residencia
_code	Código del empleado
_name	Nombre
_surname	Apellidos
_cardID	Número de DNI
_birthDate	Fecha de nacimiento
_gender	Sexo del empleado
_address	Dirección de la residencia habitual del empleado
_postalCode	Código postal de la residencia habitual del empleado
_city	Ciudad de residencia
_email	Dirección de correo electrónico de uso profesional


 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

_workPhoneNumber	Número del teléfono profesional del empleado
_workMobilePhoneNumber	Número del teléfono móvil de uso profesional
_extension	Extensión del teléfono profesional del empleado
_personalPhoneNumber	Número del teléfono personal del empleado
_personalMobilePhoneNumber	Número del teléfono móvil de uso personal
_personalEmail	Correo electrónico de uso personal
_fax	Fax
_status	Estado del empleado
_remarks	Comentarios acerca del empleado
_photoPath	Ruta de la foto del empleado
_civilStatus	Referencia al estado civil del empleado
_isHandicapped	Indicador de si el empleado posee algún tipo de minusvalía
_handicapPercentage	Porcentaje de minusvalía

5.3.5 Prj_ActionPlanItem:

Es la clase de acceso a datos que representa a la entidad 'Proyecto' y la entidad 'Tarea'. Sus atributos son:

Atributo	Descripción
_ID	Identificador
_ParentNodeID	Referencia al nodo padre en la estructura
_ParentAreaID	Referencia al area a la que pertenece el proyecto
_EmpResponsibleID	Referencia al responsable
_BudgetEntryID	Referencia a la partida presupuestaria
_VersionID	Referencia a la version
_ProjectType	Referencia al tipo de proyecto
_FinancialType	Referencia al tipo de financiación
_InitiativeType	Referencia al tipo de iniciativa
_ScopeType	Referencia al tipo de alcance
_NotificationType	Referencia al tipo de notificación
_ParticipationType	Referencia al tipo de participación
_Center	Referencia al tipo de centro asociado
_code	Código
_name	Nombre
_type	Tipo (P = Proyecto; T = Tarea)

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

_state	Stado
_creationDate	Fecha de creación
_purpose	Propósito
_plannedStart	Fecha inicio planificado
_actualStart	Fecha de inicio
_plannedEnd	Fecha planificada de fin
_actualEnd	Fecha de fin
_deadline	Fecha de compromiso
_progress	Progreso
_notes	Notas
_private	Privacidad
_objectives	Objetivos
_supplier	Proveedores
_outputs	Resultados esperados
_historyRemarks	Antecedentes
_duration	Duración
_plannedDuration	Duración planeada
_executionTime	Tiempo de ejecución
_LastUpdate	Ultima modificación


5.3.6 Prj_ActionPlanArea:

Es la clase de acceso a datos que representa a la entidad 'Área'. Sus atributos son:

Atributo	Descripción
_ID	Identificador del área
_name	Nombre del área
_parentNodeID	Referencia al área a la que pertenece en la estructura
_responsible	Responsable de área
_notes	Descripción
_status	Estado

5.3.7 Prj_ActionPlanActions:

Es la clase de acceso a datos que representa las acciones introducidas en un proyecto o tareas. Sus atributos son:


 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Atributo	Descripción
_ID	Identificador de la acción
_planItemID	Referencia al proyecto/tarea asociado con la acción
_responsableID	Referencia al responsable de la acción
_agreement	Referencia al grupo de cotización del empleado
_deadline	Fecha de compromiso
_hours	Horas invertidas en la acción
_remarks	Descripción
_state	Estado
_fullfilmentRemarks	Comentarios obligatorios
_creationDate	Fecha de creación
_lastUpdateEmployee	Referencia al empleado que ha realizado la última actualización
_lastUpdateDate	Fecha de la última actualización

5.3.8 Prj_ActionPlanDocuments:

Es la clase de acceso a datos que representa los documentos asociados con proyectos o tareas. Sus atributos son:

Atributo	Descripción
_ID	Identificador del documento
_ActionPlanItemID	Referencia al proyecto/tarea asociado con el documento
_typeID	Referencia al tipo de documento
_name	Nombre
_visibility	Visibilidad
_employeeLastUpdate	Referencia al empleado que realizado la ultima modificación
_size	Tamaño
_description	Descripción
_file	Imagen del documento
_isCompressed	Bit que indica si esta comprimido
_isDeleted	Bit que indica si ha sido borrado del sistema
_lastUpdate	Fecha última modificación del documento

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

5.3.9 Prj_ActionPlanWorkTime:

Es la clase de acceso a datos que representa los esfuerzos introducidos en un proyecto/tarea. Sus atributos son:

Atributo	Descripción
_ID	Identificador del esfuerzo
_actionPlanID	Referencia al proyecto/tarea asociado con el esfuerzo
_employeeID	Referencia al empleado responsable del esfuerzo
_date	Fecha de introducción del esfuerzo
_hoursInvested	Horas invertidas
_description	Descripción
_isOvertime	Bit que indica si son horas extra


5.3.10 Prj_ActionPlanIndicators:

Es la clase de acceso a datos que representa los indicadores económicos, no económicos o de cliente asociados con proyectos, para calcular su presupuesto. Sus atributos son:

Atributo	Descripción
_ID	Identificador del identificador
_parentIndicatorID	Referencia al indicador de primer nivel asociado con el indicador
_actionPlanID	Referencia al proyecto asociado con el indicador
_ReferenceIndicatorTypeID	Referencia al tipo de indicador de primer nivel asociado con el indicador
_type	Tipo de indicador
_trend	Tendencia del indicador
_target	Objetivo
_dateReference	Fecha de introducción
_revisionSchedule	Revisión
_remarks	Descripción
_isFixed	Bit que indicar si es un indicador fijo

5.3.11 Prj_ActionPlanIndicatorsHistory:

Es la clase de acceso a datos que representa la evolución de los valores asociados con la clase anterior 'Indicadores'. Sus atributos son:

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Atributo	Descripción
_IdentificadorID	Referencia al indicador
_date	Fecha de introducción
_value	Valor
_employee	Referencia al empleado
_creationDate	Fecha de creación
_remarks	Descripción

5.3.12 Prj_ActionPlanIndicatorsTarget:

Es la clase de acceso a datos que representa los objetivos de los valores asociados con la clase 'Indicadores'. Sus atributos son:

Atributo	Descripción
_IdentificadorID	Referencia al indicador
_date	Fecha de introducción
_value	Valor
_employee	Referencia al empleado
_creationDate	Fecha de creación
_remarks	Descripción


5.3.13 Prj_ActionPlanItemApprovalHistory:

Es la clase de acceso a datos que representa el registro de aprobaciones que tiene un proyecto a lo largo de su desarrollo. Sus atributos son:

Atributo	Descripción
_itemID	Referencia al proyecto asociado
_date	Fecha de la aprobación
_type	Tipo de aprobación
_remarks	Descripción
_employee	Nombre del empleado que ha realizado la aprobación
_result	Resultado

5.3.14 Prj_ActionPlanItemInternalHistory:

Es la clase de acceso a datos que representa el registro de modificaciones que sufre un proyecto/tarea a lo largo de su desarrollo. Sus atributos son:

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Atributo	Descripción
_ID	Identificador
_itemID	Referencia al proyecto/tarea asociado
_section	Sección modificada
_variationType	Tipo de modificación
_date	Fecha de la modificación
_beginDate	Fecha de comienzo de la modificación
_value	Valor de la modificación
_employee	Nombre del empleado que realiza la modificación
_remarks	Descripción

5.3.15 Prj_ActionPlanItemProgress:


Es la clase de acceso a datos que representa el seguimiento del progreso un proyecto/tarea. Sus atributos son:

Atributo	Descripción
_ID	Identificador del ítem de progreso
_actionID	Referencia al proyecto/tarea asociado con el ítem de progreso
_employeeID	Referencia al empleado que introduce el progreso
_date	Fecha de introducción del progreso
_progress	Valor del progreso
_remarks	Descripción del progreso

5.3.16 R_ActionPlanItem_Employee:

Es la clase de acceso a datos que representa la relación de un empleado con un proyecto/tarea. Sus atributos son:

Atributo	Descripción
_itemID	Referencia al proyecto/tarea
_employeeID	Referencia al empleado
_rolID	Referencia al rol del empleado dentro del proyecto / tarea
_privilege	Privilegios
_percent	Porcentaje de desarrollo
_notes	Descripción

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

_hoursPlanned	Horas planadas
_costPerHour	Coste por hora

5.3.17 R_Gen_User_Privilege:


Es la clase de acceso a datos que representa la relación de un usuario con los permisos de sistema asociados. Sus atributos son:

Atributo	Descripción
_userID	Referencia al usuario
_privilegeName	Referencia al permiso
_privilegeValue	Valor

5.3.18 R_Gen_User_SystemOption:

Es la clase de acceso a datos que representa la relación de un usuario con las opciones del sistema. Sus atributos son:

Atributo	Descripción
_userID	Referencia al usuario
_optionName	Referencia a la opción del sistema
_optionValue	Valor

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

6. DISEÑO DE LA BASE DE DATOS

6.1 Introducción


En este apartado se procede a diseñar la base de datos. En primer lugar se muestra la estructura general de la base de datos del SGP. El objetivo es proporcionar una idea global para, a continuación, mostrar con más detalle cada una de las tablas de forma individual.

6.2 Diseño general de la base de datos

Debido al tamaño de la base de datos, se va a separar en subconjuntos de menor tamaño, ya que no sería posible representarlo de otro modo en este formato de documento.

6.2.1 Usuarios:

Contiene la información relacionada con los usuarios del sistema, y sus datos de acceso. Todos los usuarios del sistema tienen una entrada en la tabla Gen_Users, ya que es la que contiene los datos de acceso al mismo. Los candidatos tienen otra entrada en la tabla Orh_Emp_Employee. Los usuarios tienen unos permisos asignados que están en la tabla R_Gen_User_Privileges, y además unas opciones del sistema que se relaciona mediante la tabla R_Gen_User_SystemOption.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

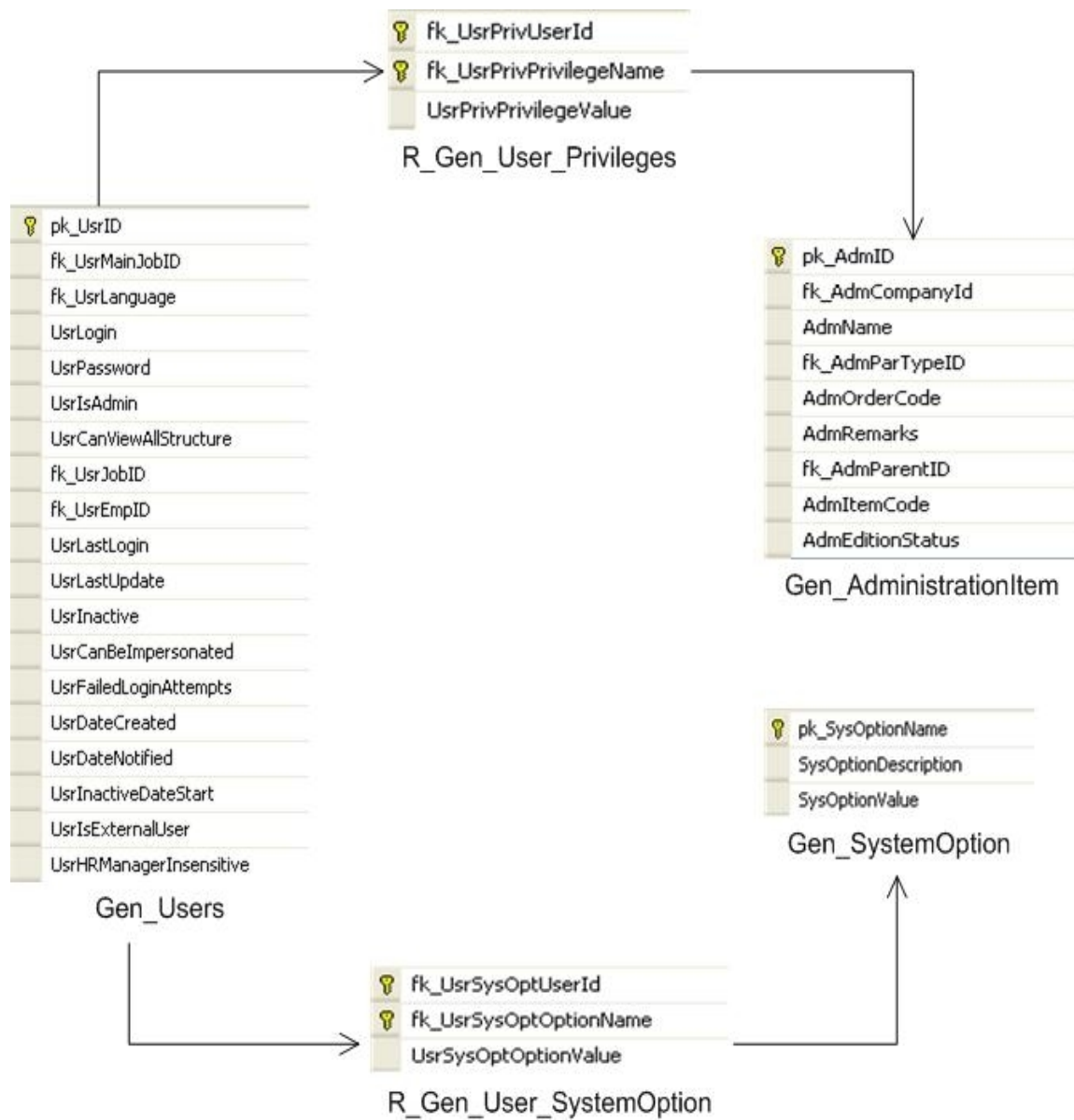



Figura 14:Diagrama de usuarios

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

6.2.2 Empleados

Contiene la información relacionada con los empleados del sistema. Además también reflejamos su relación con la estructura de proyectos.

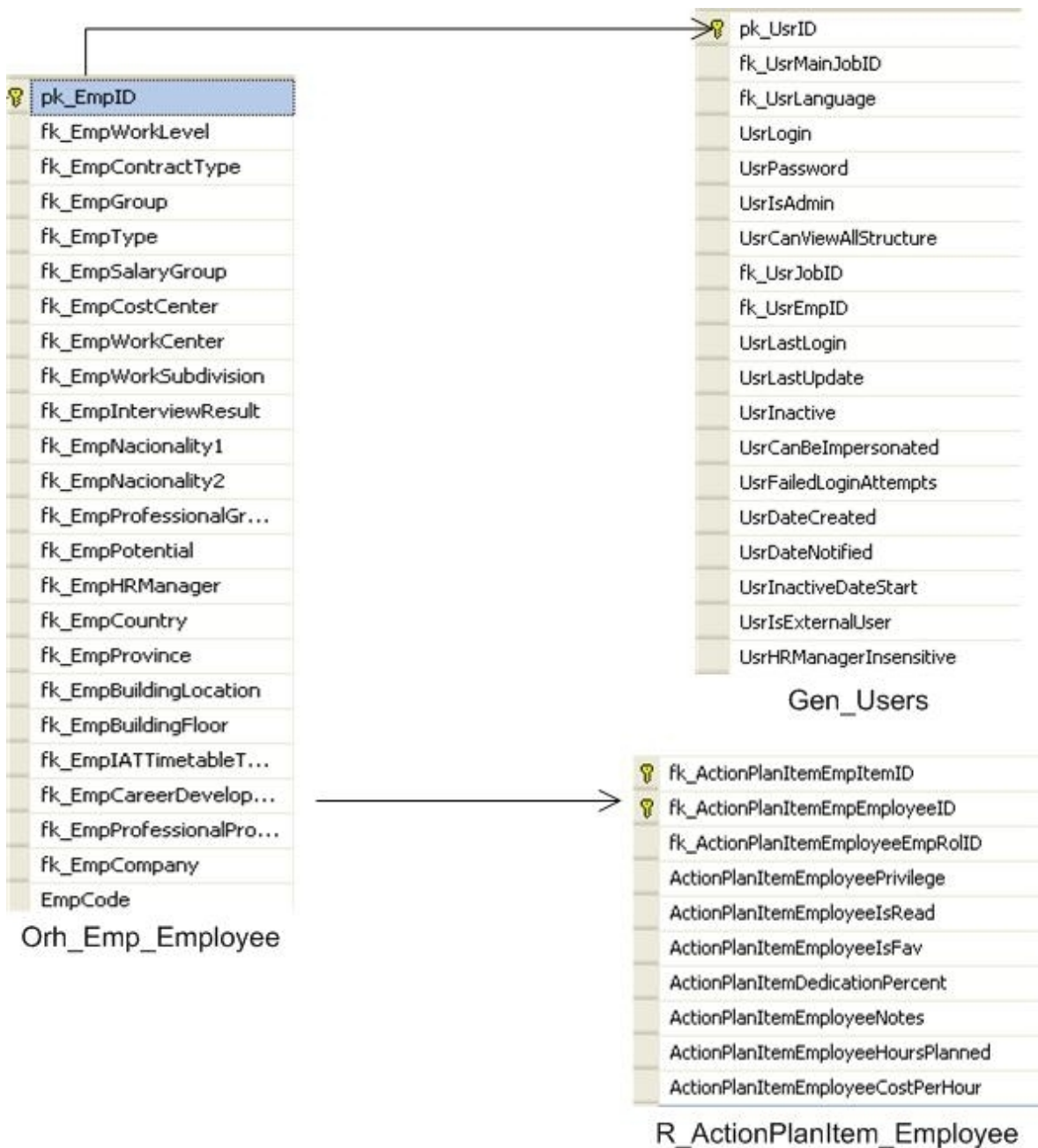



Figura 15: Diagramas de empleados

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

6.2.3 Diagrama de planes acción (areas, proyectos, tareas)

Contiene la información de las evaluaciones de competencias. Como se trata de una parte del proyecto heredada de otro sistema, solo se comentará por encima, por no haberse desarrollado durante la realización de este proyecto.

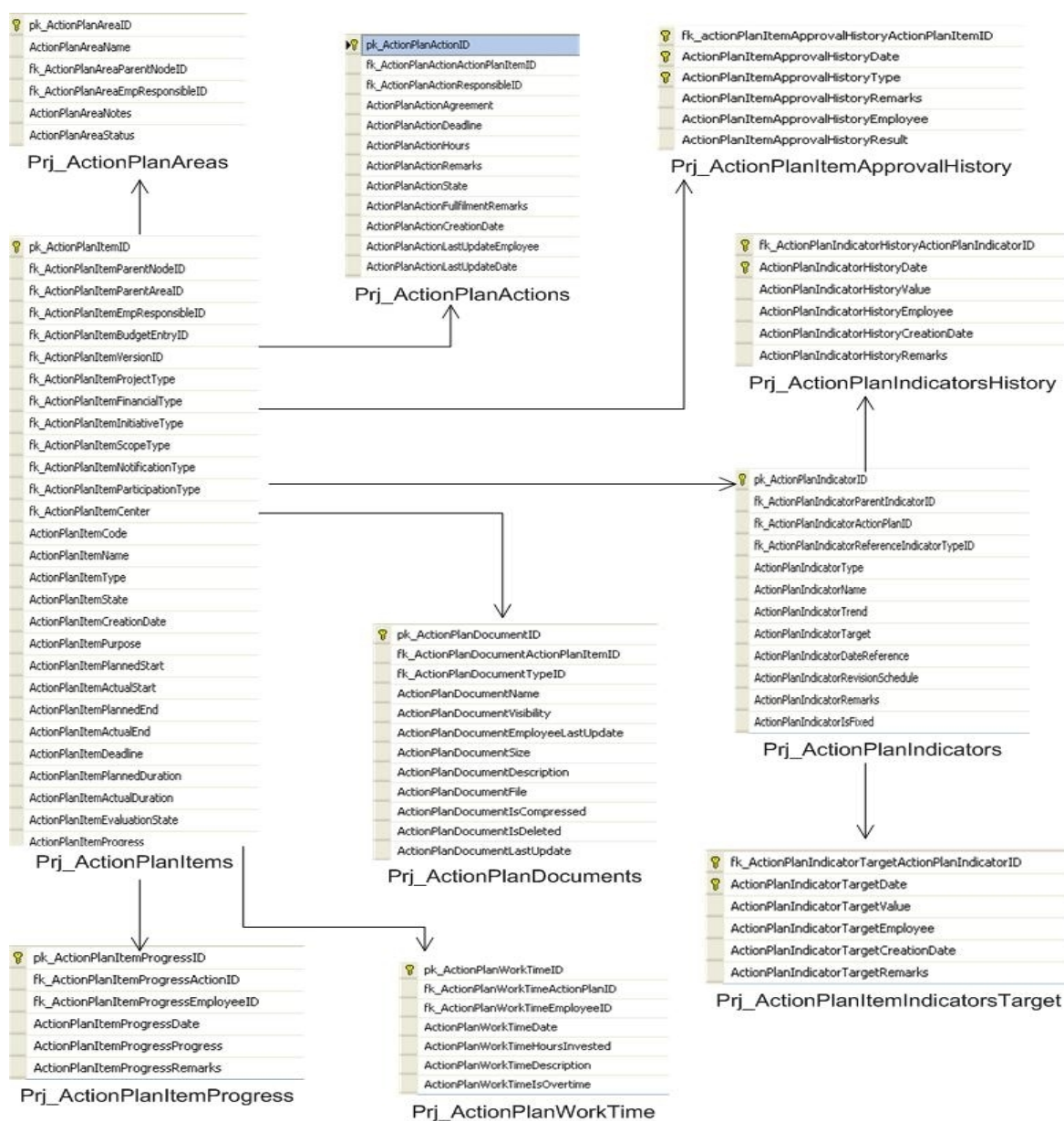



Figura 16:Diagrama de planes de acción

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	


6.3 Tablas de la base datos

A continuación se explican en detalle cada una de las tablas de la base de datos. Se agruparán del mismo modo que se han agrupado en el esquema general, para facilitar su búsqueda.

6.3.1 Usuarios

Gen_User


Campo	Tipo	Nulo	Descripción
pk_UsrID	Int	No	Identificador del usuario. Clave primaria
Fk_usrMainJobID	Int	Si	Referencia al puesto principal asociado con el usuario
Fk_UsrLanguage	nvarchar(20)	Si	Referencia al idioma del usuario
UsrLogin	Nvarchar(70)	No	Nombre de usuario
UsrrPassword	Nvarchar(50)	No	Contraseña del usuario
UsrlsAdmin	Bit	No	Indica si el usuario es administrador del sistema
UsrCanViewAllStructure	Bit	No	Indica si el usuario puede ver toda la estructura del sistema
Fk_UsrJobID	Int	No	Referencia al puesto asociado con el usuario
Fk_UsrEmpID	Int	No	Referencia con el empleado asociado al usuario
UsrLastLogin	smalldatetime	Sí	Fecha en que el usuario accedo por última vez al sistema
UsrLastUpdate	smalldatetime	Sí	Ultima fecha en que el usuario modifiko sus datos
UsrlsInactive	Bit	SI	Indica si es usuario inactivo
UsrFailedLoginAttenpts	Int	Si	Número de intentos fallidos para acceder a la aplicación
UsrDateCreated	Smalldatetime	Si	Fecha de creación del usuario
UsrDateNotified	Smalldatetime	Si	Fecha en que el usuario acepta la clausula de privacidad
UsrlsExternalUSer	Bit	SI	Indica si es usuario externo

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

6.3.2 Empleados

Orh_Emp_Employee

Campo	Tipo	Nulo	Descripción
pk_EmpID	int	No	Clave primaria del empleados
fk_EmpWorkLevel	int	Si	Referencia al nivel de trabajo.
fk_EmpcontractType	int	Sí	Referencia al tipo de contrato
fk_EmpGroup	int	Sí	Referencia a la categoría
fk_EmpType	int	Sí	Referencia al tipo de empleado
fk_EmpSalaryGroup	int	Sí	Referencia al salario
fk_EmpCostCenter	int	Sí	Referencia al centro de coste
fk_EmpWorkCenter	int	Sí	Referencia al centro de trabajo
fk_EmpWorkSubdivision	int	Sí	Referencia a la subdivisión de trabajo
fk_EmpInterviewResult	int	Sí	Referencia a la evaluación de desempeño
fk_EmpNationality1	int	Sí	Referencia a la nacionalidad
Fk_EmpNationality2	int	Sí	Referencia a la segunda nacionalidad
Fk_EmpProfessionalGroup	int	Sí	Referencia al grupo profesional
Fk_EmpPotential	int	Sí	Referencia a la potencialidad del empleado
Fk_EmpHRManager	int	Si	Referencia al manager de recursos humanos
Fk_EmpCountry	int	Sí	Referencia al país de trabajo
Fk_empProvince	int	Sí	Referencia a la provincia de trabajo
Fk_EmpBuildingLocation	int	Sí	Referencia al edificio de trabajo
Fk_EmpCareerDevelopment	Int	Sí	Referencia al desarrollo de carrera
EmpCode	nvarchar(8)	No	Código
EmpName	nvarchar(50)	No	Nombre
EmpSurname	nvarchar(80)	No	Apellido
EmpCardID	nvarchar(30)	Sí	DNI
EmpHealthID	nvarchar(30)	Sí	Nº seguridad social

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

EmpBirthDate	Smalldatetime	Si	Fecha nacimiento
empGender	nvarchar(1)	Si	Sexo
EmpAddress	nvarchar(120)	Si	Dirección
EmpPostalCode	nvarchar(10)	Si	Código postal
EmpCity	nvarchar(40)	Si	Ciudad
EmpRegion	nvarchar(30)	Si	Provincia


6.3.3 Planes de acción (Áreas, Proyectos, Tareas)

Prj_ActionPlanAreas


Campo	Tipo	Nulo	Descripción
Pk_ActionPlanAreaID	int	No	Clave primaria.
ActionPlanAreaName	int	No	Nombre
Fk_ActionPlanAreaParentoNodeID	char(1)	Sí	Referencia área asociadas con el área
Fk_ActionPlanAreaEmpResponsableID	Int	No	Referencia responsable del area
ActionPlanAreaNotes	Nvarchar(2000)	No	Descripción
ActionPlanAreaStatus	nvarchar(1)	No	Stado

Prj_ActionPlanItem

Campo	Tipo	Nulo	Descripción
Pk_ActionPlanItemID	int	No	Clave primaria.
fk_ActionPlanItemParentNodeID	int	Si	Referencia al nodo padre dentro de la estructura
Fk_ActionPlanItemParentAreaID	int	Sí	Referencia al area dentro de la estrutura
Fk_ActionPlanItemResponsableID	int	Sí	Referencia al responsable
Fk_ActionPlanItemBudgetEntry	int	Sí	Referencia a la partida presupuestaria
Fk_ActionPlanItemVersionID	int	Sí	Referencia a la version
Fk_ActionPlanItemProjectType	int	Sí	Referencia al tipo de proyecto
Fk_ActionPlanItemFinacialType	int	Sí	Referencia al tipo de financiación
Fk_ActionPlanItemInitiativeType	int	Sí	Referencia al tipo de iniciativa

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	


Fk_ActionPlanItemScopeType	int	Sí	Referencia al tipo de alcance
Fk_ActionPlanItemNotificationType	int	Sí	Referencia al tipo de notificación
Fk_ActionPlanItemParticipationType	int	Sí	Referencia al tipo de participación
Fk_ActionPlanItemCenter	int	Sí	Referencia al tipo de centro
ActionPlanItemCode	Navarchar(50)	No	Code
ActionPlanItemName	Nvarchar(MAX)	No	Nombre
ActionPlanItemType	Char(1)	No	Tipo (P=Proyecto; T=Tarea)
ActionPlanItemState	Char(1)	Sí	Stado
ActionPlanItemCreationDate	smalldatetime	No	Fecha de cración
ActionPlanItemPurpose	ntext	Sí	Proposito
ActionPlanItemPlannedStart	smalldatetime	Sí	Fecha inicio planificado
ActionPlanItemActualStart	smalldatetime	Sí	Fecha de inicio
ActionPlanItemPlannedEnd	smalldatetime	Sí	Fecha de fin planificada
ActionPlanItemActualEnd	smalldatetime	Sí	Fecha de fin
ActionPlanItemDeadline	smalldatetime	Sí	Fecha de compromiso
ActionPlanItemPlannedDuration	int	Sí	Duración planeada
ActionPlanItemActualDuration	Int	Sí	Duracion actual
ActionPlanItemEvaluationState	Char(1)	Sí	Stado de evaluacion
ActionPlanItemProgress	int	Sí	Progreso

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

ActionPlanItemNotes	ntext	Sí	Notas
ActionPlanItemFirstPlannedStart	smalldatetime	Sí	Primera fecha de inicio planificado
ActionPlanItemFirstPlannedEnd	smalldatetime	Sí	Primera fecha de fin planificado
ActionPlanItemExecutionTime	int	Sí	Numero de ejecuciones
ActionPlanItemExecutionMeasureUnit	Char(1)	Sí	Unidad de ejecución
ActionPlanItemPrivate	bit	Sí	Privacidad
ActionPlanItemLastUpdate	smalldatetime	Sí	Ultima actualización
ActionPlanItemPostMortemAnalysis	ntext	Sí	Notas históricas
ActionPlanItemObjectives	ntext	Sí	Objetivos

Prj_ActionPlanActions

Campo	Tipo	Nulo	Descripción
Pk_ActionPlanActionID	int	No	Clave primaria
fk_ActionPlanActionActionPlanItemID	Int	Si	Referencia al proyecto/tarea
Fk_ActionPlanActionResponsibleID	Int	No	Referencia al responsable
ActionPlanActionAgreement	Nvarchar(MAX)	Si	Compromiso
ActionPlanActionDeadline	Smalldatetime	Si	Fecha de compromiso
ActionPlanActionHours	Float	Si	Horas
ActionPlanActionRemarks	Nvarchar(MAX)	Si	Descripción
ActionPlanActionState	Char(1)	Si	Estado
ActionPlanActionFullfilmentRemarks	Nvarchar(MAX)	Si	Descripción de la acción (campo obligatorio)
ActionPlanActionCreationDate	smalldatetime	Si	Fecha de creación
ActionPlanActionLastUpdate	smalldatetime	Si	Fecha ultima modificación


	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Prj_ActionPlanWorkTime

Campo	Tipo	Nulo	Descripción
Pk_ActionPlanWorkTimeID	int	No	Clave primaria.
fk_ActionPlanWorkTimeActionPlanID	int	No	Referencia al proyecto/tarea.
Fk_ActionPlanWorkTimeEmployeeID	int	No	Referencial el empleado
ActionPlanWorkTimeDate	Smalldatetime	No	Fecha del esfuerzo
ActionPlanWorkTimeHoursInsested	Float	No	Horas invertidas
ActionPlanWorkTimeDescription	ntext	No	Descripción
ActionPlanWorkTimeIsOvertime	bit	Si	Bit que representa horas extra

Prj_ActionPlanDocuments

Campo	Tipo	Nulo	Descripción
Pk_ActionPlanDocumentID	int	No	Clave primaria.
fk_ActionPlanDocumentActionPlanItemID	int	No	Referencia al proyecto/tarea
Fk_ActionPlanDocumentTypeID	int	Sí	Referencia al tipo de documento
ActionPlanDocumentName	Nvarchar(255)	Sí	Nombre
ActionPlanDocumentVisibility	Char(1)	Sí	Visibilidad
ActionPlanDocumentEmployeeLastUpdate	Nvarchar(255)	No	Ultima modificación
ActionPlanDocumentSize	int	No	Tamaño del documento
ActionPlanDocumentDescription	Nvarchar(255)	Sí	Descripción
ActionPlanDocumentFile	image	Sí	Fichero
ActionPlanDocumentIsCompressed	Bit	Sí	Bit que representa si esta comprimido
ActionPlanDocumentIsDeleted	bit	Sí	Bit que representa si esta borrado
ActionPlanDocumentLastUpdate	smalldatetime	Sí	Fecha ultima modificación

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Prj_ActionPlanApprovalHistory


Campo	Tipo	Nulo	Descripción
fk_actionPlanItemApprovalHistoryActionPlanItemID	int	No	Clave primaria.
ActionPlanItemApprovalHistoryDate	datetime	No	Fecha aprobación
ActionPlanItemApprovalHistoryType	char(1)	No	Tipo de aprobación
ActionPlanItemApprovalHistoryRemarks	Nvarchar(MAX)	Si	Descripción
ActionPlanItemApprovalHistoryEmployee	Nvarchar(150)	Si	Empleado que realiza la aprobación
ActionPlanItemApprovalHistoryResult	Char(1)	Si	Resultado de la aprobación

Prj_ActionPlanInternalHistory

Campo	Tipo	Nulo	Descripción
Pk_ActionPlanItemIntHisID	Int	No	Clave primaria.
Fk_ActionPlanItemHisItemID	Int	No	Referencia al proyecto/tarea.
ActionPlanItemIntHisSection	Int	No	Indica si el empleado está suscrito
ActionPlanItemIntHisVariationType	Int	No	Tipo de modificación
ActionPlanItemIntHisDate	Datetime	No	Fecha modificación
ActionPlanItemIntHisBeginDate	Datetime	Si	Fecha inicio modificación
ActionPlanItemIntHisValue	Nvarchar(MAX)	No	Valor modificación
ActionPlanItemIntHisEmployee	Nvarchar(150)	Si	Empleado que realizar la modificación
ActionPlanItemIntHisRemarks	Nvarchar(MAX)	Si	descripción

Prj_ActionPlanIndicators


Campo	Tipo	Nulo	Descripción
pk_ActionPlanIndicatorID	int	No	Clave primaria.
fk_ActionPlanIndicatorParentIn	int	Si	Referencia al indicador.

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

dicatorID			
Fk_ActionPlanIndicatorActionPlanID	Int	Sí	Referencia al proyecto
Fk_ActionPlanIndicatorReferenceIndicatorTypeID	Int	Si	Referencia al tipo de indicador
ActionPlanIndicatorType	Char(1)	Si	Tipo
ActionPlanIndicatorName	Nvarchar(MAX)	Si	Nombre
ActionPlanIndicatorTrend	Char(1)	Si	Tendencia
ActionPlanIndicatorTarget	Float	Si	Objetivo
ActionPlanIndicatorDateReference	smalldatetime	Si	Fecha de referencia
ActionPlanIndicatorRevisionSchedule	Int	Si	Fecha de la revisión
ActionPlanIndicatorRemarks	Nvarchar(MAX)	Si	Descripción
ActionPlanIndicatorsIsFixed	bit	Si	Bit que indica si es fijo


Prj_ActionPlanIndicatorsHistory

Campo	Tipo	Nulo	Descripción
fk_ActionPlanIndicatorHistoryActionPlanIndicatorID	int	No	Clave primaria.
ActionPlanIndicatorHistoryDate	datetime	No	Fecha del próximo valor
ActionPlanIndicatorHistoryValue	float	Sí	Valor
ActionPlanIndicatorHistoryEmployee	Nvarchar(200)	Sí	Empleado que introduce el indicador
ActionPlanIndicatorHistoryCreationDate	smalldatetime	Sí	Fecha de objetivo
ActionPlanIndicatorsHistoryRemarks	Nvarchar(200)	Sí	Descripción

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Prj_ActionPlanIndicatorsTarget

Campo	Tipo	Nulo	Descripción
fk_ActionPlanindicatorTargetActionPlanIndicatorID	int	No	Clave primaria.
ActionPlanIndicatorTargetDate	datetime	No	Fecha del próximo
ActionPlanIndicatorTargetValue	float	Sí	Valor
ActionPlanIndicatorTargetEmployee	Nvarchar(200)	Sí	Empleado que introduce el indicador
ActionPlanIndicatorTargetCratiodate	smalldatetime	Sí	Fecha
ActionPlanIndicatorsTargetRemarks	Nvarchar(200)	Sí	Descripción

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

6.3.4 Administración

Gen_AdministrationItem

Campo	Tipo	Nul o	Descripción
Pk_AdmID	int	No	Clave primaria.
fk_AdmCompanyId	int	Si	Referencia a la compañía
AdmName	Nvarchar(255)	No	nombre
Fk_AdmParTypeID	Nvarchar(2)	No	Referencia al tipo
AdmOrderCode	Int	Si	Código de ordenación
admRemarks	Nvarchar(3000)	Si	Descripción
Fk_AdmParentID	Int	Si	Referencia a la clase general
AdmItemCode	Nvarchar(20)	Si	Código
AdmEditionStatus	int	Si	Estado


6.3.5 Relaciones

R_Gen_User_Privileges

Campo	Tipo	Nulo	Descripción
fk_UsrPrivUserID	int	No	Referencia al usuario
fk_UsrPrivPrivilegeName	Nvarchar(50)	No	Referencia al privilegio
UsrPrivPrivilegeValue	tinyint	Sí	Valor del privilegio


R_Gen_User_SystemOption

Campo	Tipo	Nulo	Descripción
fk_UsrSysOptUserId	int	No	Referencia al usuario
fk_UsrSysOptOptionName	Nvarchar(100)	No	Referencia a la opción del sistema
UsrSysOptOptionValue	Nvarchar(500)	Sí	Valor de la opción del sistema

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

R_ActionPlanItem_Employee

Campo	Tipo	Nulo	Descripción
fk_actionPlanItemEmpltemID	int	No	Referencia al proyecto/tarea
Fk_actionPlanItemEmpemploy eelID	Int	No	Referencia el empleado
Fk_ActionPlanItemEmployeeE prolID	Int	Sí	Referencia al rol del empleado
ActionPlanItemEmployeePrivil ege	Int	Sí	Privilegios
ActionPlanItemEmployeeIsRe ad	Bit	Sí	Bit de solo lectura
ActionPlanEmployeeIsFav	Bit	Sí	Bit de preferencia
ActionPlanItemDedicationPerc ent	Float	Sí	Porcentaje de dedicación
ActionPlanItemEmployeesNot es	Nvarchar(MAX)	Sí	Descripción
ActionPlanItemEmployeeHour sPlanned	float	Si	Horas planeadas
ActionPlanItemEmployeeCost PerHour	Float	Si	Coste por hora

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

7. PROTOTIPADO DE LA INTERFAZ

En este apartado se muestran los prototipos de interfaz de pantalla diseñados para el sistema. No se ha buscado realizar prototipos de aspecto realista, sino la mayor esquematización posible, huyendo de posibles detalles dependientes de implementaciones de controles concretos.

Los objetivos y decisiones tomadas en este punto son:

- La resolución mínima para la que se diseñará la aplicación es 1024x768.
- Primar la claridad y la usabilidad por encima de otros aspectos.
- Los elementos tendrán siempre que sea posible diseño líquido, es decir, adaptarán sus dimensiones dependiendo de la resolución de la pantalla en que se visualicen.

Procederemos a prototipar los tipos de interfaces principales o más representativos, indicando a qué formularios concretos es aplicable cada uno de los esquemas.

7.1 Prototipos

En la siguiente figura se observa el esquema general de todas las interfaces de la aplicación. Podemos observar tres partes diferenciadas (Figura 28):

- Menú del SGP: en la parte superior de la pantalla se ubicará el menú de la aplicación. Este menú será siempre visible, y permitirá la navegación a cualquier sección de la aplicación deseada.
- Franja de pestañas abiertas: bajo el menú principal, aparecerán las pestañas de los formularios abiertos en cada momento.
- En la parte inferior de la pantalla es donde se mostrarán los diferentes formularios de la aplicación. Por tanto, el resto de prototipos de interfaces deberán ocupar el espacio asociado a esta sección.

Este prototipo se corresponde con el formulario MDIMain.cs.

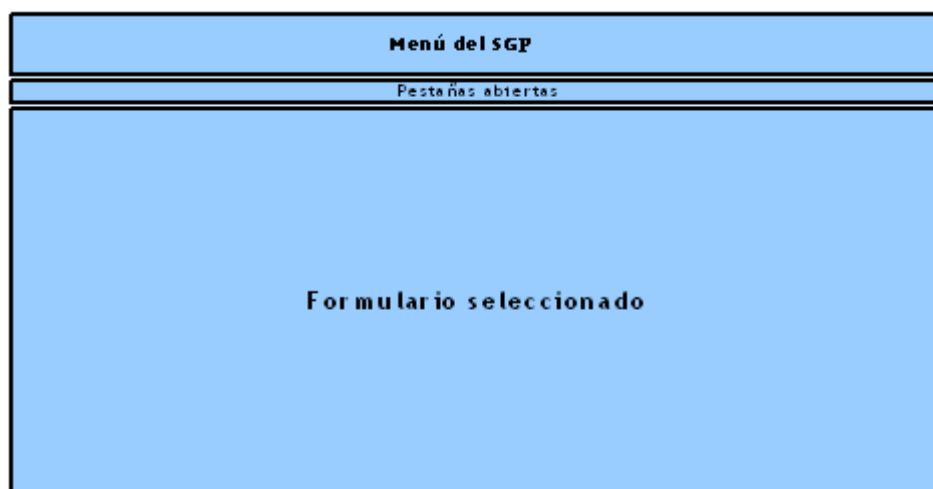



Figura 28: Esquema general de interfaz

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

En la siguiente figura (Figura 29) observamos el siguiente esquema de interfaz existente. Consta de dos partes diferenciadas:

- Una barra de herramientas del formulario, en la parte superior del mismo. En ella se ubicarán las diferentes acciones y utilidades disponibles para el formulario.
- En la parte inferior de la pantalla se mostrará el contenido del formulario.

Este esquema de interfaz se corresponde con los formularios: ProjectForm.cs, TaskForm.cs, UserForm.cs, NewUserForm.cs



Figura 29: Interfaz tipo barra herramientas + contenido.


En la siguiente figura (Figura 30), se observa un nuevo prototipo de interfaz, que consta de tres secciones diferentes:

- Una barra de herramientas del formulario, en la parte superior del mismo.
- Una estructura tipo árbol, en la parte izquierda.
- Una sección de visualización de datos, en la parte derecha del formulario.

Este tipo de esquema se corresponde con los formularios: ProjectStruct.cs, EmployeeStruct.cs y UserStruct.cs.

En función del formulario concreto de los anteriormente mencionados, en la parte izquierda visualizaremos la estructura de parámetros, empleados, competencias o puestos, respectivamente.

En la parte derecha se visualizará la información asociada al elemento del árbol seleccionado en cada momento.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

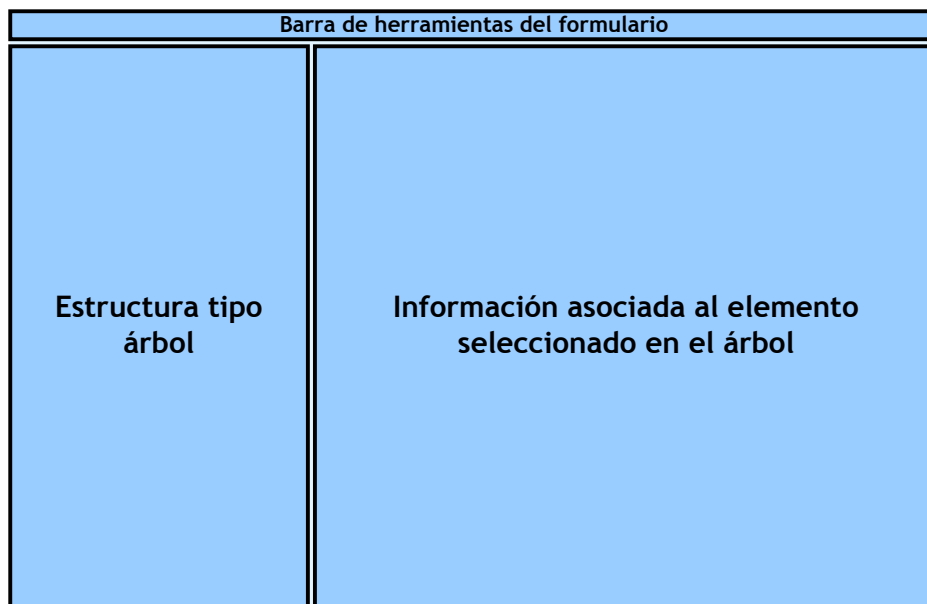



Figura 30: Interfaz tipo estructura

En la siguiente figura (Figura 31) observamos un prototipo de interfaz menos genérico, ya que está asociado a un único formulario: EmployeeForm.cs y NewEmployeeForm. Este formulario muestra la ficha de un empleado. Consta de cinco secciones:

- Barra de herramientas del formulario.
- Información básica. En ella aparecerán datos básicos del empleado, que serán visibles en todo momento, tales como: nombre, apellidos, centro de trabajo o código del empleado.
- Fotografía del empleado.
- Menú de secciones del formulario. En esta franja central de la pantalla se ubicará un menú que permitirá navegar entre las distintas secciones en las que se divide la información almacenada del empleado. Dichas secciones serán: información personal, información profesional, curriculum, perfil de conocimientos /competencias, información salarial e información de costes.
- En la parte inferior de la pantalla se visualizará la información asociada a la sección seleccionada por el usuario.

	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

Barra de herramientas del formulario	
Información básica	Foto
Menú secciones formulario	
Información asociada a la sección seleccionada	


Figura 31: Interfaz de ficha de empleado

En la siguiente figura (Figura 32), se observa el prototipo de interfaz asociado al formulario AreaForm.cs, NewAreaForm.cs, TaskRegisterFastForm.cs, duplicateProjecForm.cs, AdmItemsEditForm.cs, JobForm.cs. Dicho formulario se usar cuando la cantidad de información no es mucha y las operaciones que se pueden realizar son muy básicas:

- El formulario muestra la información básica del elemento seleccionado.
- En la parte inferior se visualizará una serie de botones, normalmente aceptar y cancelar para realizar la operación o cerrar el formulario.

Contenido del formulario
Botones

Figura 32: Interfaz de ficha general

 endalia®	Aplicación para la gestión de proyectos	Versión: 1.2.0
	Diseño	Fecha: 13/01/2010
	DISEÑO.doc	

8. BIBLIOGRAFIA

8.1 Referencias

- [IGJ, 2000] I. Jacobson, G. Booch, J. Rumbaugh. 2000. "El Proceso Unificado de Desarrollo de Software". Pearson Education.
- [IGJ, 1999] I. Jacobson, G. Booch, J. Rumbaugh. 1999. "El lenguaje unificado de modelado. Manual de referencia". Ed. Addison Wesley.
- [RUM, 1991] J. Rumbaugh 1991. "Modelado y Diseño Orientado a Objetos". Ed. Prentice Hall, 1991.
- [HOA 2003] Hoang Lam, Thuan L. Thai. ".NET Framework Essentials, 3rd Edition". O'Reilly 2003.
- [GIT 2003] Art Gittleman. "Computing With C# and the .Net Framework". Jones and Bartlett Publishers 2003.
- [HHBK 2008] P. Henri Kuate, T. Harris, C. Bauer, G. King. "NHibernate in Action". Manning Publications 2008.


8.2 Referencias Web

- [Ref. Web 1] <http://www.wikipedia.org/>
- [Ref. Web 2] <http://www.uml.org/>
- [Ref. Web 3] <http://www.rational.com/>
- [Ref. Web 4] <http://www.hibernate.org/>
- [Ref. Web 5] <http://nhforge.org>

Sistema de gestión de proyectos

Implementación

Versión 1.2 – Fecha: 02/04/2010

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
18/01/2010	1.0	Realización del documento de implementación	José Ángel Peribáñez
17/03/2010	1.1	Revisión del documento	Fernando Cortés
02/04/2010	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

TABLA DE CONTENIDOS


<u>1. INTRODUCCIÓN</u>	5
1.1 PROPÓSITO	5
1.2 ALCANCE	5
1.3 ACRÓNIMOS	5
1.4 DEFINICIONES	5
1.5 REFERENCIAS	6
1.6 RESUMEN	6
<u>2. DESCRIPCIÓN DEL PROCESO</u>	7
<u>3. TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES</u>	8
<u>4. IMPLEMENTACIÓN DE LA INTERNACIONALIZACIÓN</u>	9
4.1 DEFINICIÓN	9
4.2 OBJETIVOS DE LA INTERNACIONALIZACIÓN	9
4.3 ELEMENTOS A INTERNACIONALIZAR	9
4.4 CLAVES DE LA INTERNACIONALIZACIÓN	9
4.5 PROCESO DE IMPLEMENTACIÓN DE LA INTERNACIONALIZACIÓN	10
<u>5. IMPLEMENTACIÓN DE LA GESTIÓN DE INCIDENCIAS</u>	12
<u>6. IMPLEMENTACIÓN DEL ACCESO A BASE DE DATOS</u>	14
6.1 INTRODUCCIÓN	14
6.2 SQL SERVER 2005	14
6.3 NHIBERNATE	14
6.3.1 CONFIGURACIÓN DE NHIBERNATE	14
6.3.2 IMPLEMENTAR EL MODELO DE DOMINIO	16
6.3.3 MAPEAR EL MODELO DE DOMINIO	23
6.3.4 MANIPULACIÓN DE DATOS PERSISTENTES	25
6.3.5 EL GESTOR DE PERSISTENCIA	27
6.3.6 PROCEDIMIENTOS ALMACENADOS	27
<u>7. IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO</u>	33
7.1 INTRODUCCIÓN	33
7.2 ELEMENTOS DE INTERFAZ	33
7.3 PANTALLAS DEL SISTEMA	33
7.3.1 PÁGINA DE INICIO	34
7.3.2 CONFIGURACIÓN DEL SISTEMA	35
7.3.3 GESTIÓN DE USUARIOS	35

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

7.3.4	GESTIÓN DE EMPLEADOS	36
7.3.5	GESTIÓN DE PROYECTOS	37
7.3.6	INFORMES DIRECTOS	42
7.3.7	GENERACIÓN DE INFORMES	43

8. BIBLIOGRAFÍA **46**

8.1	REFERENCIAS	46
8.2	REFERENCIAS WEB	46

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

1. INTRODUCCIÓN

1.1 PROPÓSITO

El presente documento describe la fase de implementación del proyecto de desarrollo del SGP. A partir de las bases obtenidas mediante el análisis y el diseño, se obtienen los archivos de código fuente, librerías y recursos necesarios para la ejecución del sistema.

1.2 ALCANCE

El alcance del documento comprende la fase de implementación del SGP, en la fase final de desarrollo del sistema.

1.3 ACRÓNIMOS

ADO: ActiveX Data Objects.

ANSI: American National Standards Institute.

API: Application Programming Interface.

ASP: Active Server Pages.

CRUD: Create, Read, Update, Delete.

DDL: Data Definition Language.

DML: Data Manipulation Language.

GUI: Graphical User Interface.

SDK: Software Development Kit.

SGBD: Sistema Gestor de Base de Datos.

SGP: Sistema de Gestión de proyectos.

SQL: Structured Query Language.


T-SQL: Transact Structured Query Language.

XML: Extensible Markup Language.

WPF: Windows Presentation Foundation.

1.4 DEFINICIONES

- Archivo de recursos: archivo en el que se almacenan datos que se corresponden con cierta información que maneja el sistema, pero que no dependen específicamente de las clases que contienen lógica.
- Control de versiones: gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.
- SGBD transaccional: SGBD capaz de mantener la integridad de los datos, haciendo que las transacciones no puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, deshace las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

1.5 REFERENCIAS


En este documento se realizan referencias a los siguientes documentos del proyecto:

- DISEÑO.doc: documento que describe la fase de diseño del proyecto.
- ESTANDARCODIFICACION.doc: documento que especifica el estándar de codificación utilizado en el proyecto.

1.6 RESUMEN

Este documento describe el proceso de implementación de SGP. Se compone de los siguientes apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describe el proceso de implementación seguido para la confección de este documento.
- Apartado 3: Se describen las tecnologías, herramientas y lenguajes empleados durante la implementación del sistema.
- Apartado 4: Descripción del proceso seguido para la implementación de la internacionalización de la aplicación.
- Apartado 5: Descripción del proceso seguido para la gestión de incidencias ocurridas en el sistema.
- Apartado 6: Detalle de las consideraciones necesarias para la implementación del acceso a datos.
- Apartado 7: Descripción del proceso seguido para la implementación definitiva del interfaz de usuario.
- Apartado 8: Bibliografía y referencias web utilizadas para la realización de este documento.


 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

2. DESCRIPCIÓN DEL PROCESO

A partir de los subsistemas, clases y estructuras identificados en el diseño del sistema, se comienza el proceso de implementación del mismo. Dicho proceso es un proceso iterativo que, junto con la fase de pruebas, tiene por objeto conseguir el desarrollo de la aplicación final, con la calidad necesaria. Una vez concluida una iteración de la implementación, se pasa a la fase de pruebas, donde se identifican las partes del proceso que es necesario repetir total o parcialmente.

Dentro del proceso de implementación propiamente dicho, podemos identificar tres fases:


- En la primera fase se realiza la implementación de los sistemas genéricos de la aplicación. Éstos son:
 - Sistema de gestión de incidencias: controla las incidencias y eventos del sistema, generando un fichero de log que permite trazar los sucesos ocurridos durante la ejecución del mismo.
 - Sistema de internacionalización: permite la adaptación a diferentes culturas sin necesidad de cambios estructurales.
- La segunda fase consiste en la implementación de la base de datos descrita en el documento de diseño.
- La tercera fase consiste en la implementación de los diferentes módulos y subsistemas de la aplicación identificados en la fase de diseño.

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

3. TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES

Para el desarrollo del presente proyecto se han utilizado las siguientes tecnologías, lenguajes y herramientas:

- Microsoft .NET Framework. Plataforma de desarrollo descrita en el apartado 3.4 del Documento de Diseño.
- Microsoft SQL Server 2005. SGBD utilizado para la gestión de la base de datos del sistema y descrito en el apartado 3.5 del Documento de Diseño.
- C#. Lenguaje de programación de propósito general nativo de la plataforma .NET.
- Microsoft Visual Studio 2008. Entorno de programación y depuración de código de la plataforma .NET.
- Microsoft SQL Management Studio 2005. Herramienta gráfica que permite realizar tareas de mantenimiento de la base de datos sobre SQL Server 2005.
- Microsoft Team Foundation Server 2008. Herramienta que permite, entre otras funcionalidades, la gestión del control de versiones de un proyecto sobre Visual Studio 2008.
- NHibernate: herramienta de mapeo objeto-relacional para la plataforma .NET.
- T-SQL. Lenguaje de acceso a datos basado en SQL. Está descrito en el apartado 6.3.6.1 del presente documento.
- Infragistics NetAdvantage 8.3. Librería para .NET que contiene controles para diversos entornos de desarrollo dentro de dicha plataforma, como ASP.NET, Winforms o WPF.
- ILOG Diagram for .NET 1.6. Suite de herramientas gráficas para la creación de diagramas para la plataforma .NET.
- Log4Net. Herramienta para ayudar en la generación de ficheros de registro.
- Microsoft Excel para la exportación de los informes a formato .xls, y para la importación de datos de forma masiva desde plantillas prefijadas.

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

4. IMPLEMENTACIÓN DE LA INTERNACIONALIZACIÓN

4.1 DEFINICIÓN

Uno de los requisitos iniciales planteados para el SGP es la internacionalización del sistema. La internacionalización se define como el “proceso de diseñar una aplicación que pueda ser adaptada a distintos idiomas y culturas sin necesidad de efectuar cambios estructurales en la misma”. En este apartado se describe la solución adoptada en el sistema para realizar la internacionalización del mismo.

4.2 OBJETIVOS DE LA INTERNACIONALIZACIÓN

Un sistema internacionalizado persigue las siguientes características:

- El sistema debe poder ser ejecutado en cualquier lugar del mundo.
- El texto mostrado por el sistema debe estar en el idioma del usuario final.
- El texto mostrado por el sistema no debe estar codificado dentro del programa, sino que debe ser almacenado de forma externa y ser recuperado de forma dinámica en tiempo de ejecución.
- Otros aspectos culturales como números, fechas u horas deben aparecer en el formato e idioma del usuario.

4.3 ELEMENTOS A INTERNACIONALIZAR


Los elementos susceptibles de ser internacionalizados en el SGP son:

- Textos.
- Números: pueden variar por el carácter delimitador de decimales, o el separador de miles, por ejemplo.
- Fechas y horas: pueden variar de muchas formas, por ejemplo, el orden en que se indican los días y los meses de una fecha.

4.4 CLAVES DE LA INTERNACIONALIZACIÓN

A continuación se presentan un conjunto de reglas que es necesario cumplir para desarrollar de manera óptima el proceso de internacionalización de una aplicación .NET.

- Identificar los elementos dependientes de la cultura. Los mensajes de texto son los componentes que de manera más obvia varían con la cultura, ya que deben ser traducidos al idioma del usuario final. Sin embargo, hay otros tipos de datos que podrían variar con la región o el idioma.
- Aislar el texto traducible. El texto traducido debe agruparse y aislarse en archivos de recursos. El texto traducido incluye mensajes de estado, mensajes de error, etiquetas

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

de componentes GUI, etc. La codificación siguiente, en la que los mensajes se asignan de manera implícita en el código fuente, debe evitarse totalmente:

```
String buttonLabel = "Aceptar";
```

```
btnOk.Text = buttonLabel;
```

- Formatear números y monedas. Si la aplicación muestra números y monedas, será necesario darles el formato adecuado. Evitar:

```
Double amount = 25.0;
```

```
TextboxAmount.Text = amount.ToString + "€";
```

- Formatear fechas y horas. El formato de las fechas y horas varía según la cultura. Evitar:

```
DateTime date = "10/05/2005";
```

```
TextboxDate.Text = date.ToString();
```

- Reservar espacio suficiente en el GUI. El texto de las etiquetas varía de longitud según el idioma en que esté codificado, siendo por ejemplo los mensajes en inglés generalmente más cortos que en otros idiomas.

4.5 PROCESO DE IMPLEMENTACIÓN DE LA INTERNACIONALIZACIÓN

El proceso de internacionalización de aplicaciones .NET es el siguiente:


- Almacenar las etiquetas en un archivo de texto que llamaremos archivo de origen de datos o de recursos de cadena (este punto puede variar, pudiéndose guardar también en otros formatos como archivos .resx). El formato de estos archivos es el siguiente:
 - Una declaración de cadena por línea en la que se declara una etiqueta de internacionalización y su valor correspondiente dentro de la referencia cultural que define al archivo separados por un signo '='. Por ejemplo:

```
ActionPlanList_btn_Ok=Aceptar
```

- Comentarios precedidos por un punto y coma ';' al principio de la línea:

```
;esto es un comentario dentro del archivo de datos
```

- Convertir el archivo de origen de datos en un formato compatible con los ensamblados satélites (.resources). Para realizar este paso se utiliza la herramienta resgen. Esta herramienta convierte archivos .txt y archivos .resx (formato de recursos basado en XML en archivos .resources binarios de Common Language Runtime, que se pueden incrustar en un archivo ejecutable binario de motor de tiempo de ejecución o compilar en ensamblados satélite.
- Compilar el ensamblado satélite con los archivos de recursos. A continuación, para generar el ensamblado satélite es necesario utilizar otra herramienta del SDK denominada al.exe (Assembly Linker) la cual crea dicho ensamblado y a su vez embebe el recurso correspondiente.
- Declarar en el código fuente las cadenas de internacionalización. En el código fuente

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

de la aplicación en que vayamos a utilizar los recursos hay que crear un campo de tipo string con un modificador de acceso private para cada uno de los datos que serán cargados desde el ensamblado satélite. La codificación de estas variables se describe en el documento 'Estándar de codificación'.

```
private string CompetenceList_lbl_btnOk
```

- Cargar las cadenas de internacionalización con los valores del archivo de recursos. Para ello, en el constructor de cualquiera de nuestros formularios, llamaremos a los métodos:

```
InitializeLabels();
LoadI18N();
```

El método InitializeLabels realiza la lectura del fichero de recursos en el que se almacenan las cadenas de internacionalización, cargando éstas en variables de cadena, de la siguiente forma:

```
private ResourceManager rm;

private void InitializeLabels()
{
    Assembly a = Assembly.Load("labels");
    rm = new ResourceManager("labels", a);


    if (rm != null)
    {
        JobForm_lbl_neoTxtName =
rm.GetString("JobForm_lbl_neoTxtName");
        JobForm_lbl_neoTxtCode =
rm.GetString("JobForm_lbl_neoTxtCode");
    }
}
```

Para poder trabajar con la clase ResourceManager es necesario declarar el espacio de nombre System.Resources.

Una vez inicializado el objeto 'rm', se cargan los valores del archivo de recursos a los campos. Para ello se utiliza el método GetString del objeto 'rm', el cual recibe como parámetro el nombre del campo que queremos invocar del archivo de recursos que nos provee el ensamblado satélite.

El método LoadI18N carga en los campos de texto del formulario las variables de cadenas obtenidas en el método InitializeLabels, de la siguiente forma:

```
neoTxtName.Label = JobForm_lbl_neoTxtName_Label;
neoTxtCode.Label = JobForm_lbl_neoTxtCode_Label;
```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

5. IMPLEMENTACIÓN DE LA GESTIÓN DE INCIDENCIAS

El sistema de gestión de incidencias, o Log de la aplicación, se encarga de registrar secuencialmente los eventos del sistema en un fichero, guardando información acerca del tipo de evento, de cuándo ocurrió y qué lo causó. Dicho fichero puede posteriormente ser consultado, auditado y analizado para conocer el uso, funcionamiento y posibles problemas del sistema.

La siguiente información es registrada siempre, independientemente del resto del contenido del registro de log:

- Fecha y hora de registro del evento.
- Sistema gestor del log (clase que provoca la llamada).
- Tag con el tipo de mensaje según la siguiente clasificación:
 - DEBUG: mensaje informativo para depuración de la ejecución del sistema.
 - INFO: mensaje informativo sobre algún suceso acaecido en el sistema.
 - WARM: mensaje de aviso sobre intento de alguna ejecución peligrosa para el sistema.
 - ERROR: mensaje acerca de una ejecución fallida pero controlada del sistema.
 - FATAL: mensaje de interrupción total de la ejecución del sistema.

La siguiente información variará dependiendo del tipo de registro:

- Mensaje de error generado en caso de error de ejecución (por ejemplo en caso de fallo de ejecución de una instrucción SQL escribiría el mensaje devuelto por SQL Server).
- Cualquier mensaje informativo que necesite ser registrado por el sistema en un momento dado.

Para registrar en el fichero de Log las interacciones del usuario con la aplicación, se emplea la librería log4net, pero es necesario que en cada evento de la aplicación se escriba explícitamente en él, por ejemplo:


```
Log.Write("EmployeeForm.toolbar1_ToolClick: Tool= " + e.Tool.Key,
Log.DEBUG);
```

A continuación, mostramos un ejemplo del contenido del fichero de log:

```
08/03/2010 [11:07:18] ERROR OSUser: japeribanez; AppUser: DEMO ;
Gen_AdministrationItem.Name Asignación de NULL a campo no anulable

08/03/2010 [11:07:19] DEBUG OSUser: japeribanez; AppUser: DEMO ;
ProjectForm.ProjectForm projID=8

08/03/2010 [11:07:19] DEBUG OSUser: japeribanez; AppUser: DEMO ;
```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	


ProjectStruct.InitializeLabels

08/03/2010 [11:07:19] DEBUG OSUser: luralde; AppUser: DEMO ; ProjectStruct.LoadI18N

08/03/2010 [11:07:19] DEBUG OSUser: japeribanez; AppUser: DEMO ;
Gen_AdministrationItem.ListByTypesSimple

08/03/2010 [11:07:19] DEBUG OSUser: japeribanez; AppUser: DEMO ;
ProjectStruct.InitializeHeader

08/03/2010 [11:07:19] DEBUG OSUser: japeribanez; AppUser: DEMO ;
ProjectForm.headerInfo_Modified

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

6. IMPLEMENTACIÓN DEL ACCESO A BASE DE DATOS

6.1 INTRODUCCIÓN

Como se ha especificado en el documento de diseño, el Sistema Gestor de Base de Datos utilizado ha sido Microsoft SQL Server 2005, y se ha utilizado NHibernate como mapeador objeto-relacional.

En este apartado se van a describir las principales características e implicaciones del uso conjunto de ambos sistemas en la implementación propiamente dicha del SGP.

6.2 SQL SERVER 2005

Es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje SQL, con las siguientes características principales:


- Soporte de transacciones.
- Gran estabilidad.
- Seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos.
- No es multiplataforma, ya que sólo está disponible en sistema Operativos de Microsoft.

6.3 NHIBERNATE

NHibernate es la conversión a .NET de la popular librería “Hibernate” de Java. NHibernate pretende ser una completa solución al problema de manejar la persistencia de datos al trabajar con bases de datos relacionales junto con un modelo de dominio de clases basado en el paradigma orientado a objetos. NHibernate asume el trabajo de mediar entre la aplicación y la base de datos, liberando al desarrollador de esta tarea.

6.3.1 CONFIGURACIÓN DE NHIBERNATE

Como ya se ha comentado, NHibernate facilita la interacción entre objetos y tablas en un modelo de datos relacional. Si bien el desarrollador no interactúa directamente con la base de datos, NHibernate sí lo hace, por lo que es necesario indicar en un archivo de configuración

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

cuál será la base de datos a utilizar y cómo conectarse con dicha base de datos.

Para configurar la comunicación entre NHibernate y la base de datos, se han usado los ficheros "app.config" y "hibernate.cfg.xml".

A continuación, se muestra parte del contenido del fichero "app.config":

```
<?xml version="1.0"?>
<configuration>
  <configSections>
    <section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
    <section name="NHibernate.Burrow"
type="NHibernate.Burrow.Configuration.NHibernateBurrowCfgSection,
NHibernate.Burrow" />
  </configSections>
  <NHibernate.Burrow>
    <persistenceUnits>
      <add name="PersistenceUnit" nh-config-file="hibernate.cfg.xml"
/>>
    </persistenceUnits>
  </NHibernate.Burrow>
  <connectionStrings>
    <add name="endalia_rrhh"
connectionString="server=.\SQLEXPRESS;database=endaliaRRHH;Integrated
Security=SSPI" />
  </connectionStrings>


  (...)

</configuration>
```

El contenido del fichero "hibernate.cfg.xml" es el siguiente:

```
<?xml version="1.0" encoding="utf-8" ?>
<hibernate-configuration xmlns="urn:hibernate-configuration-2.2">
  <session-factory>
    <property
name="dialect">NHibernate.Dialect.MsSql2005Dialect</property>
    <property
name="connection.provider">NHibernate.Connection.DriverConnectionProvi
der</property>
    <property
name="connection.driver_class">NHibernate.Driver.SqlClientDriver</prop
erty>
    <property name="connection.connection_string_name">endalia
rrhh</property>
    <property name="show_sql">>false</property>
    <property name="adonet.batch_size">10</property>
    <mapping assembly="IntegraDBAccess" />
  </session-factory>
</hibernate-configuration>
```

Las propiedades que define el fichero "hibernate.cfg.xml" son, entre otras:

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

- El nombre de la clase .NET que implementa el “IConnectionsProvider” para NHibernate. En este caso, se usa la clase por defecto.
- El nombre de la clase .NET que implementa el “Dialect” que habilita ciertas características dependientes de la plataforma. A pesar del esfuerzo de estandarización de ANSI, SQL se implementa de modo diferente por parte de distintos fabricantes de bases de datos. Es por ello que hay que indicar el “dialecto” que se va a utilizar. NHibernate ofrece soporte para las bases de datos SQL más populares, y nuevos dialectos pueden ser definidos en caso de necesidad fácilmente.
- El nombre de la clase .NET que implementa el “Driver” ADO.NET.
- El nombre de la cadena de conexión usada para acceder a la base de datos. La cadena de conexión en sí se encuentra en el fichero “app.config”, bajo dicho nombre (“endalia rrhh”).

6.3.2 IMPLEMENTAR EL MODELO DE DOMINIO

Una de las características de NHibernate es que permite persistir los cambios automáticamente, de modo transparente al modelo de dominio. Existe una completa separación entre las clases del modelo y la lógica que permite su persistencia, donde las clases que van a ser persistidas son totalmente independientes del mecanismo de persistencia. Además:


- NHibernate no necesita que las clases a persistir implementen ninguna interfaz ni hereden características de ninguna clase base determinada.
- Las clases a persistir pueden ser reutilizadas fuera del contexto de la persistencia, en el nivel de presentación al usuario, por ejemplo.
- En un sistema con persistencia transparente, los objetos no se ven afectados por el almacenamiento de datos subyacente. Lo relativo a la persistencia está externalizado a un interfaz de gestión de persistencia que, en el caso de NHibernate, está formado por los interfaces: ISession y IQuery.

Como ejemplo, vamos a mostrar el fichero de clase “Prj_ActionPlanDocument”, que almacena los documentos asociados a proyectos o tareas.

El fichero de clase “Prj_ActionPlanDocument.cs” es el siguiente:

```
using integradbaccess.Core;
using integradbaccess.util;
using System;
using System.Collections.Generic;

namespace integradbaccess.Domain
```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```


{
    /// <summary>
    /// Clase de acceso a datos mapeada para NHibernate de la tabla
    Prj_ActionPlanDocuments
    /// </summary>
    [Serializable]
    public class Prj_ActionPlanDocument : DomainObject<int>
    {

        #region Constantes

        /// <summary>
        /// Nombres de las propiedades de la clase
        Prj_ActionPlanDocument
        /// </summary>
        public struct Properties
        {
            public const string ID = "ID";
            public const string Name = "Name";
            public const string Visibility = "Visibility";
            public const string EmployeeLastUpdate =
"EmployeeLastUpdate";
            public const string Size = "Size";
            public const string Description = "Description";
            public const string File = "File";
            public const string IsCompressed = "IsCompressed";
            public const string IsDeleted = "IsDeleted";
            public const string LastUpdate = "LastUpdate";
            public const string ActionPlanItems = "ActionPlanItems";
            public const string Type = "Type";
        }

        /// <summary>
        /// Restricciones de las propiedades de la clase
        Prj_ActionPlanDocument
        /// </summary>
        public struct Constraints
        {
            public const int Name_MAXLENGTH = 255;
            public const int EmployeeLastUpdate_MAXLENGTH = 255;

```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

    }

    public struct Constants
    {
        public struct Visibility
        {
            public const char PUBLIC = 'P';
            public const char PRIVATE = 'V';
        }
    }

    #endregion Constantes

    #region Atributos


    private string _name;
    private char? _visibility;
    private string _employeeLastUpdate;
    private int _size;
    private string _description;
    private byte[] _file;
    private bool? _isCompressed;
    private bool? _isDeleted;
    private DateTime? _lastUpdate;
    private Prj_ActionPlanItem _actionPlanItems;
    private Gen_AdministrationItem _type;

    #endregion Atributos

    #region Constructores

    /// <summary>
    /// Constructor de Prj_ActionPlanDocument
    /// </summary>
    public Prj_ActionPlanDocument()
    {
    }

```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

    /// <summary>
    /// Constructor de Prj_ActionPlanDocument con ID
    /// </summary>
    public Prj_ActionPlanDocument(int id)
    {
        base.ID = id;
    }


#endregion Constructores

#region Propiedades

    public virtual string shortName
    {
        get
        {
            if (Name.Length > 100)
            { return Name.Substring(0, 100) + "..."; }
            else
            { return Name; }
        }
    }

    /// <summary>
    /// Nombre del fichero
    /// </summary>
    public virtual string Name
    {
        get { return _name; }
        set
        {
            if ((value != null) && (value.Length >
Constraints.Name_MAXLENGTH))
            {
                string errorStr = "Prj_ActionPlanDocument.Name MAX
LEN( " + Constraints.Name_MAXLENGTH + ") Excedida: " + value.Length +

```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

"; Valor: " + value;

        Log.Write(errorStr, Log.ERROR);
    }
    else
    {
        _name = value;
    }
}


}

/// <summary>
/// Visibilidad del documento. T=Público N=Privado. Si está a
null se toma como privado
/// </summary>
public virtual char? Visibility
{
    get { return _visibility; }
    set { _visibility = value; }
}

public virtual string EmployeeLastUpdate
{
    get { return _employeeLastUpdate; }
    set
    {
        if (value == null)
        {
            string errorStr =
"Prj_ActionPlanDocument.EmployeeLastUpdate Asignación de NULL a campo
no anulable";

            Log.Write(errorStr, Log.ERROR);
        }
        else if (value.Length >
Constraints.EmployeeLastUpdate_MAXLENGTH)
        {
            string errorStr =
"Prj_ActionPlanDocument.EmployeeLastUpdate MAX LEN( " +
Constraints.EmployeeLastUpdate_MAXLENGTH + ") Excedida: " +
value.Length + "; Valor: " + value;

```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

        Log.Write(errorStr, Log.ERROR);
    }
    else
    {
        _employeeLastUpdate = value;
    }
}
}

```

```

/// <summary>
/// Tamaño del fichero en KiloBytes
/// </summary>
public virtual int Size
{
    get { return _size; }
    set { _size = value; }
}

```

```


/// <summary>
/// Descripción del documento
/// </summary>
public virtual string Description
{
    get { return _description; }
    set { _description = value; }
}

```

```

/// <summary>
/// El fichero almacenado
/// </summary>
public virtual byte[] File
{
    get { return _file; }
    set { _file = value; }
}

```

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

}

```

/// <summary>
/// Marca si el fichero está o no comprimido
/// </summary>

```

```

public virtual bool? IsCompressed
{
    get { return _isCompressed; }
    set { _isCompressed = value; }
}

```

```

/// <summary>
/// Marca si el fichero con FileID ha sido borrado
/// </summary>

```

```

public virtual bool? IsDeleted
{
    get { return _isDeleted; }
    set { _isDeleted = value; }
}

```

```

/// <summary>
/// Ultima fecha de modificación del documento
/// </summary>

```

```

public virtual DateTime? LastUpdate
{
    get { return _lastUpdate; }
    set { _lastUpdate = value; }
}

```

```


/// <summary>
/// Obtiene un Prj_ActionPlanItem a través de la relación
FK_Prj_ActionPlanDocuments_Prj_ActionPlanItems
/// </summary>

```

```

public virtual Prj_ActionPlanItem ActionPlanItems
{
    get { return _actionPlanItems; }
}

```


	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

        set { _actionPlanItems = value; }
    }

    /// <summary>
    /// Obtiene un Gen_AdministrationItem a través de la relación
    FK_Prj_ActionPlanDocuments_Gen_AdministrationItems
    /// </summary>
    public virtual Gen_AdministrationItem Type
    {
        get { return _type; }
        set { _type = value; }
    }

#endregion Propiedades

    /// <summary>
    /// Obtiene el código hash de un Prj_ActionPlanDocument
    /// </summary>
    public override int GetHashCode()
    {
        return ID.GetHashCode();
    }
}
}

```


6.3.3 MAPEAR EL MODELO DE DOMINIO

El fichero de mapeo "Prj_ActionPlanDocuments.hbm.xml" es el siguiente:

```

<?xml version="1.0" encoding="utf-8" ?>
<hibernate-mapping xmlns="urn:nhibernate-mapping-2.2"
assembly="IntegraDBAccess" namespace="integradbaccess.Domain">
    <class name="Prj_ActionPlanDocument" table="Prj_ActionPlanDocuments"
dynamic-update="true" >
        <id name="ID" type="int" column="pk_ActionPlanDocumentID">

```

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

    <generator class="identity" />
</id>

    <property name="Name" column="ActionPlanDocumentName" not-
null="false" />

    <property name="Visibility" column="ActionPlanDocumentVisibility"
not-null="false" />

    <property name="EmployeeLastUpdate"
column="ActionPlanDocumentEmployeeLastUpdate" not-null="true" />

    <property name="Size" column="ActionPlanDocumentSize" not-
null="true" />

    <property name="Description"
column="ActionPlanDocumentDescription" not-null="false" />

    <property name="File" column="ActionPlanDocumentFile" not-
null="false" />

    <property name="IsCompressed"
column="ActionPlanDocumentIsCompressed" not-null="false" />

    <property name="IsDeleted" column="ActionPlanDocumentIsDeleted"
not-null="false" />

    <property name="LastUpdate" column="ActionPlanDocumentLastUpdate"
not-null="false" />


    <many-to-one name="ActionPlanItems" class="Prj_ActionPlanItem" >
        <column name="fk_ActionPlanDocumentActionPlanItemID" />
    </many-to-one>

    <many-to-one name="Type" class="Gen_AdministrationItem" >
        <column name="fk_ActionPlanDocumentTypeID" />
    </many-to-one>
</class>
</hibernate-mapping>

```

Como se puede observar, un fichero de mapeo XML consta de diferentes secciones:

- Los “mapeos” son declarados dentro de un elemento “<hibernate-mapping>”.
- La clase Prj_ActionPlanDocuments es mapeada a la clase del mismo nombre. Cada fila de dicha tabla representa una instancia de tipo Prj_ActionPlanDocuments.
- Cada registro de la tabla Prj_ActionPlanDocuments tendrá un valor de clave primaria que coincida con la identidad del objeto cuya instancia estará en memoria. El elemento de mapeo <id> se usa para definir los detalles de la identidad de objetos.
- La propiedad LastUPdate, por ejemplo, es mapeada a la columna de base de datos ActionPlanDocumentLastUpdate. NHibernate usará reflexión .NET para descubrir el tipo de dicha propiedad y deducir cómo mapearla con la columna SQL asociada, asumiendo que tienen tipos compatibles.
- Se usan asociaciones para unir una categoría con otra. Aquí, aparecen asociaciones de tipo “many-to-one”, es decir “muchos-a-uno”. En la base de datos, la tabla

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

ActionPlanDocuments contiene una columna llamada fk_ActionPlanDocumentActionPlanItemID, que es una clave ajena a la tabla Prj_ActionPlanItems. Dicha asociación es mapeada en el fichero xml del siguiente modo:

```
<many-to-one name="ActionPlanItems" class="Prj_ActionPlanItem" >
  <column name="fk_ActionPlanDocumentActionPlanItemID" />
</many-to-one>
```

Por convenio, los ficheros de mapeo NHibernate XML son nombrados con la extensión “.hbm.xml”. Otra convención es tener un fichero de mapeo por clase. Y se considera una buena práctica de programación ubicar el fichero de clase y el fichero de mapeo en el mismo directorio.

6.3.4 MANIPULACIÓN DE DATOS PERSISTENTES

Dado que NHibernate es un mecanismo de persistencia transparente, las clases no son “conscientes” de su propia capacidad de persistencia. Es, por tanto, posible, escribir la lógica de la aplicación sin tener en cuenta si los objetos que se manejan representan objetos en estado persistente o en un estado temporal que existe sólo en memoria.

Sin embargo, en cualquier aplicación con un “estado persistente”, la aplicación tendrá que interactuar con el nivel de persistencia cada vez que necesite transmitir el estado mantenido en memoria a la base de datos (o viceversa). Para hacer esto, se pueden utilizar el “persistence manager” de NHibernate o las “query interfaces”.

En la siguiente figura (Figura 1) podemos ver los tres estados que define NHibernate en el ciclo de vida de la persistencia de los objetos (o “Persistence lifecycle” en inglés). Estos tres estados son “transient” (pasajero, o fugaz, en español) “persistent” (persistente) y “detached” (separado, distante).

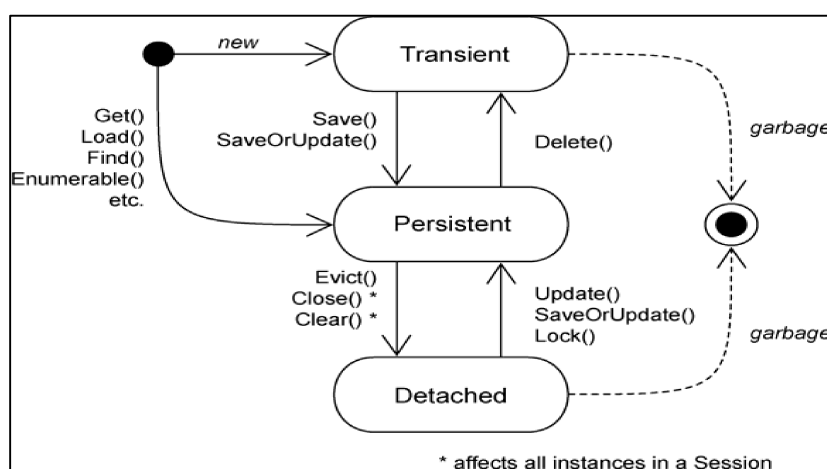



Figura 1: Ciclo de vida de la persistencia de un objeto en NHibernate

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

6.3.4.1 TRANSIENT OBJECTS

En NHibernate, los objetos instanciados usando el operador “new” no son inmediatamente persistentes. Su estado es “transient” (o pasajero), lo que significa que no están asociados con ninguna fila de base de datos, y por tanto son como cualquier otro objeto de la aplicación .NET. Más específicamente, su estado se pierde tan pronto como dejan de ser referenciados por cualquier otro objeto de la aplicación, momento en el cual pasan a estar disponibles para el recolector de basura (garbage collector).

Un objeto puede pasar de “transient” a “persistent” de dos modos:

- Salvándolo mediante la instrucción “Save()” del gestor de persistencia (persistence manager en inglés).
- Creando una referencia a dicho objeto desde otra instancia ya persistente.

6.3.4.2 PERSISTENT OBJECTS

Una instancia persistente es cualquier instancia con una “identidad de base de datos”. Esto significa que una instancia persistente tiene un valor de clave primaria idéntica a su identificador de base de datos.


Las instancias persistentes pueden ser objetos instanciados por la aplicación y hechos persistentes después a través del método Save() o del gestor de persistencia (ISession). Pueden ser también objetos que se han convertido en persistentes al crearse una referencia a ellos desde otro objeto persistente ya existente y previamente asociado al gestor de persistencia.

De forma alternativa, una instancia persistente puede ser una instancia obtenida de base de datos mediante la ejecución de una consulta, mediante la búsqueda a través de su identificador, o a través de la navegación del “grafo” de objetos, comenzando de otra instancia persistente.

En otras palabras, las instancias persistentes están siempre asociadas con una ISession y son transaccionales.

Las instancias persistentes participan en las transacciones (su estado es sincronizado con la base de datos al final de la transacción). Cuando una transacción finaliza, el estado mantenido en memoria es propagado a la base de datos mediante la ejecución de una sentencia SQL: INSERT, UPDATE o DELETE. Este procedimiento puede ocurrir también en otros momentos de tiempo.

Finalmente, una instancia persistente puede convertirse en “transient” a través de la instrucción Delete() del gestor de persistencia. Dicha instrucción implica el borrado de la fila de la tabla de base de datos asociada.

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

6.3.4.3 DETACHED OBJECTS

Cuando una transacción se completa y los datos son escritos en la base de datos, las instancias persistentes asociadas con el gestor de persistencia todavía existen en memoria. Si la transacción se completó correctamente, el estado de estas instancias habrá sido sincronizado con el de la base de datos. Estas instancias pierden su asociación con el gestor de persistencia cuando se cierra la `ISession`. Como ya no están asociadas al gestor de persistencia, nos referimos a este tipo de objetos como “detached” (separado o distante en español). Las instancias “detached” no tiene por qué mantener su sincronización con el estado de base de datos, ya no están bajo el control de `NHibernate`. Sin embargo, todavía contienen datos persistentes. Es posible, y común, retener una referencia en la aplicación y actualizar un objeto “detached” fuera de una transacción, y por tanto sin que `NHibernate` tenga control de los cambios. Afortunadamente, `NHibernate` permite usar dichas instancias en una nueva transacción, volviendo a asociarlas con el gestor de persistencia. Después de la “reasociación”, dichas instancias son consideradas persistentes de nuevo.

6.3.5 EL GESTOR DE PERSISTENCIA

Cualquier herramienta de persistencia transparente como `NHibernate` incluye alguna forma de API de gestión de persistencia que proporciona servicios para:

- Operaciones básicas CRUD (Create, Read, Update, Delete).
- Ejecución de consultas.
- Control de transacciones.

El gestor de persistencia puede estar manejado a través de diferentes interfaces. En el caso de `NHibernate`, dichas interfaces pueden ser: `ISession`, `IQuery`, `ICriteria` e `ITransaction`.


La interfaz central entre la aplicación y `NHibernate` es “`Session`”; es el punto de partida para todas las operaciones citadas anteriormente. En general, al referirnos a `NHibernate`, los conceptos de gestor de transacciones y `Session` (o sesión en español) son intercambiables.

Al comienzo de una unidad de trabajo, un thread obtiene una instancia de `ISession` de la aplicación `ISessionFactory`. Tras abrir una sesión nueva, ésta puede ser usada para cargar y salvar objetos.

6.3.6 PROCEDIMIENTOS ALMACENADOS

Un procedimiento almacenado (Stored Procedure) de Microsoft SQL SERVER, es un conjunto de instrucciones T-SQL que residen físicamente en la base de datos. Las principales características de los procedimientos almacenados son:

- Ejecución directamente en el motor de base de datos, el cual, suele estar en un servidor separado, por lo que evita en gran medida el tráfico de datos (la consulta en sí) entre el SGBD y las aplicaciones cliente.
- Posibilidad de almacenamiento del plan de ejecución por parte del SGBD, lo que repercute positivamente en la eficiencia de la consulta.

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

- Encapsulan la lógica de negocio, de tal modo que permiten que varios programas cliente puedan usarla, reduciendo el coste de mantenimiento de los mismos.
- Versatilidad. Al estar desarrollados en T-SQL, permiten realizar funciones del mismo nivel de complejidad que las que se puede realizar en código de aplicación.

NHibernate permite ejecutar procedimientos almacenados mediante IQuery. Para ello:

- En la clase de acceso a datos se ejecutará el procedimientos almacenado del siguiente modo:

```
public List<Prj_ActionPlanItem> ListTasksByProyectAllDescendants(int
projectID, bool retrieveAll)
{
    try
    {
        IQuery namedQuery = new
BurrowFramework().GetSession().GetNamedQuery("ListTasksByProyectAllDes
cendants");


        namedQuery.SetInt32("projectID", projectID);
        namedQuery.SetBoolean("retrieveAll",
retrieveAll);

        Log.Write("Prj_ActionPlanItemDao.ListTasksByProyectAllDescendants.
ProjectID: " + projectID.ToString(), Log.DEBUG);

        return namedQuery.List<Prj_ActionPlanItem>()
as List<Prj_ActionPlanItem>;
    }
    catch (Exception ex)
    {
        string strError =
"Prj_ActionPlanItemDao.ListTasksByProyectAllDescendants projectID=" +
projectID + " " + ex.ToString();
        Log.Write(strError, Log.ERROR);
        return new List<Prj_ActionPlanItem>();
    }
}
```

- En el fichero de mapeo xml, deberá estar mapeado el procedimiento almacenado a invocar:

```
<sql-query name="ListTasksByProyectAllDescendants">
```

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

    <return class="Prj_ActionPlanItem"/>
    exec          I_Prj_ActionPlanItem_ListTaskByProjectAllDescendants
:projectID, :retrieveAll
</sql-query>

```

Donde el procedimiento I_Prj_ActionPlanItem_ListTaskByProjectAllDescendants es el siguiente:


```

set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go
PROCEDURE [dbo].[I_Prj_ActionPlanItem_ListTaskByProjectAllDescendants]
(
    @projectID int,
    @retrieveAll bit
)
AS
DECLARE @state as char
SELECT @state=ActionPlanItemState
FROM Prj_ActionPlanItems
WHERE (pk_ActionPlanItemID=@projectID)

;WITH TempItems(ItemID)
AS
(
    --devolvemos solo los que tengan el mismo estado que el
    proyecto o sea una tarea cancelada
    SELECT pk_ActionPlanItemID
    FROM Prj_ActionPlanItems
    WHERE (fk_ActionPlanItemParentNodeID = @projectID)
        AND
        (
            ( (@retrieveAll=0) AND (( ActionPlanItemState= @state )
OR (ActionPlanItemState= 'X'))))
            OR
            ( (@retrieveAll=1) )
        )

    UNION ALL

```

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

--devolvemos solo los que tengan el mismo estado que el proyecto o sea una tarea cancelada

```

SELECT I.pk_ActionPlanItemID
FROM Prj_ActionPlanItems I
JOIN TempItems T ON I.fk_ActionPlanItemParentNodeID = T.ItemID
AND
(
(
(@retrieveAll=0) AND ((ActionPlanItemState= @state) OR
(ActionPlanItemState= 'X')) --devolvemos solo los que tengan el mismo
estado que el proyecto
OR
( (@retrieveAll=1) )
)
)

SELECT *
FROM Prj_ActionPlanItems
WHERE pk_ActionPlanItemID IN (SELECT ItemID FROM TempItems)

```

Otro modo de ejecutar procedimientos almacenados es a través del ConfigurationManager, sin necesidad de mapear el procedimiento en el fichero de mapeo. La ejecución del procedimiento almacenado se realiza, en este caso, directamente desde la clase de acceso a datos, del siguiente modo:

```


/// <summary>
public DataTable
GetPrj_ActionPlanDocuments_ListAllDocumentsByProject(int projectID,
bool canSeeAllDocs)
{
    try
    {
        DataSet dsData = SqlHelper.ExecuteDataset(

System.Configuration.ConfigurationManager.AppSettings["conn"],
        "I_Prj_ActionPlanDocuments_ListAllDocumentsByProject",
projectID, canSeeAllDocs);

        if (dsData.Tables[0] != null)
        {

Log.Write("Projects.ProjectSectionDocuments.GetPrj_ActionPlanDocuments
_ListAllDocumentsByProject ", Log.DEBUG);

```


	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```

        return dsData.Tables[0];
    }
    else
    {
        return null;
    }
}

catch (SqlException e)
{
    Log.Write("Projects.ProjectSectionDocuments.GetPrj_ActionPlanDocuments
_ListAllDocumentsByProject" + e.ToString(), Log.ERROR);
    return (null);
}

```

Donde el procedimiento I_Prj_ActionPlanDocuments:ListAllDocumentsByPorject es el siguiente:


```

set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go

ALTER PROCEDURE
[dbo].[I_Prj_ActionPlanDocuments_ListAllDocumentsByProject]
    @idProject INT, @viewAllDocuments BIT
AS
DECLARE @status CHAR(1)
SELECT @status = ActionPlanItemState
FROM dbo.Prj_ActionPlanItems
WHERE pk_ActionPlanItemID = @idProject

;WITH TempActionPlanItem(ActionPlanItemID)
AS
(
    SELECT @idProject --Caso trivial (inicio de la recursividad)
    UNION ALL
        --Regla para iniciar la recursividad

```

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

```


SELECT P.pk_ActionPlanItemID
FROM Prj_ActionPlanItems P
JOIN TempActionPlanItem T ON T.ActionPlanItemID =
P.fk_ActionPlanItemParentNodeID
AND ActionPlanItemState = @status -- Comprobación de
finalización
)
SELECT pk_ActionPlanDocumentID,
ActionPlanDocumentName,
fk_ActionPlanDocumentTypeID,
Convert(varchar,ActionPlanDocumentLastUpdate,103),
ActionPlanDocumentEmployeeLastUpdate,
ActionPlanDocumentDescription,
ActionPlanDocumentSize,
ActionPlanDocumentVisibility
FROM Prj_ActionPlanDocuments
INNER JOIN TempActionPlanItem ON ActionPlanItemID =
fk_ActionPlanDocumentActionPlanItemID
WHERE ((@viewAllDocuments=1) OR (ActionPlanDocumentVisibility='P'))

```

6.3.6.1 TRANSACT-SQL

Como se acaba de mencionar, estos procedimientos almacenados se escriben en T-SQL (también llamado Transact-SQL). T-SQL es una extensión de Microsoft y Sybase del lenguaje SQL, soportando posibilidades como:

- Todas las instrucciones de acceso a base de datos de SQL (SELECT, JOIN, etc.).
- Instrucciones de control de flujo (IF, ELSE, WHILE, LOOP, etc.).
- Variables locales y por parámetro.
- Autentificación integrada con Microsoft-Windows.
- Instrucciones de procesamiento de cadenas, fechas, funciones matemáticas, etc.

 endalia®	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

7. IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO

7.1 INTRODUCCIÓN

El diseño definitivo de interfaz viene definido por los esquemas de diseño de pantalla especificados en el documento de diseño. Por otra parte, se ha tenido en cuenta el diseño de interfaz de aplicaciones de éxito desarrolladas en la empresa. Una de las máximas que ha guiado el diseño de la interfaz (y en general, de todo el sistema) es que la aplicación fuera: “útil, sencilla y agradable para el usuario”.

A continuación, se especifica la manera en que se han utilizado los diferentes elementos que componen el interfaz finalmente diseñado, y se muestran capturas de pantalla del mismo.


7.2 ELEMENTOS DE INTERFAZ

Los elementos de interfaz empleados han sido:

- **Iconos e imágenes:** La utilización de iconos e imágenes, no es sólo algo meramente estético. Iconos de acceso adecuadamente escogidos, facilitan a los usuarios encontrar las diferentes secciones del sistema con mayor rapidez. Todos los accesos mediante iconos, irán acompañados de texto explicativo y “Tooltips” (texto explicativo que se muestra al colocar el ratón sobre el elemento que lo contiene), en caso de ser posible, para facilitar al máximo la interpretación de estos iconos por parte del usuario.
- **Tablas:** Las tablas se utilizan para mostrar listados de datos. El formato de las tablas deberá coincidir con el estándar de Endalia. El texto de la cabecera destaca sobre el resto haciéndolo en negrita, y las filas alternas deben tener colores de fondo alternos para facilitar la legibilidad de las tablas.
- **Árboles:** Las estructuras de selección de tipo árbol estarán ajustadas con el estándar definido por Endalia.
- **Editores de fechas:** Se han utilizado, siempre que ha sido posible, editores de fechas que permiten una navegación gráfica a través de diferentes fechas mediante la visualización de un calendario.
- **Colores y fuentes:** Como se explica en el documento de diseño, los colores a utilizar son los primarios y secundarios definidos para la bolsa de empleo, y las fuentes y estilos de tipo Arial.
- **Pestañas:** Se ha utilizado la navegación y estructuración de una ventana mediante pestañas siempre que ha sido necesario mostrar al usuario un número extenso de datos o campos que podían agruparse entre sí en diferentes bloques por algún concepto, pero que compartían un origen común

7.3 PANTALLAS DEL SISTEMA

A continuación, se presenta el diseño definitivo de las principales interfaces del sistema:

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

7.3.1 PÁGINA DE INICIO

Esta es la página de inicio del sistema, donde pueden verse un seguimiento de los proyectos activos y de las propuestas, los proyectos o tareas que no cumplen las fechas de compromiso, un control contable de los proyectos y una información de las personas con acceso a la aplicación.

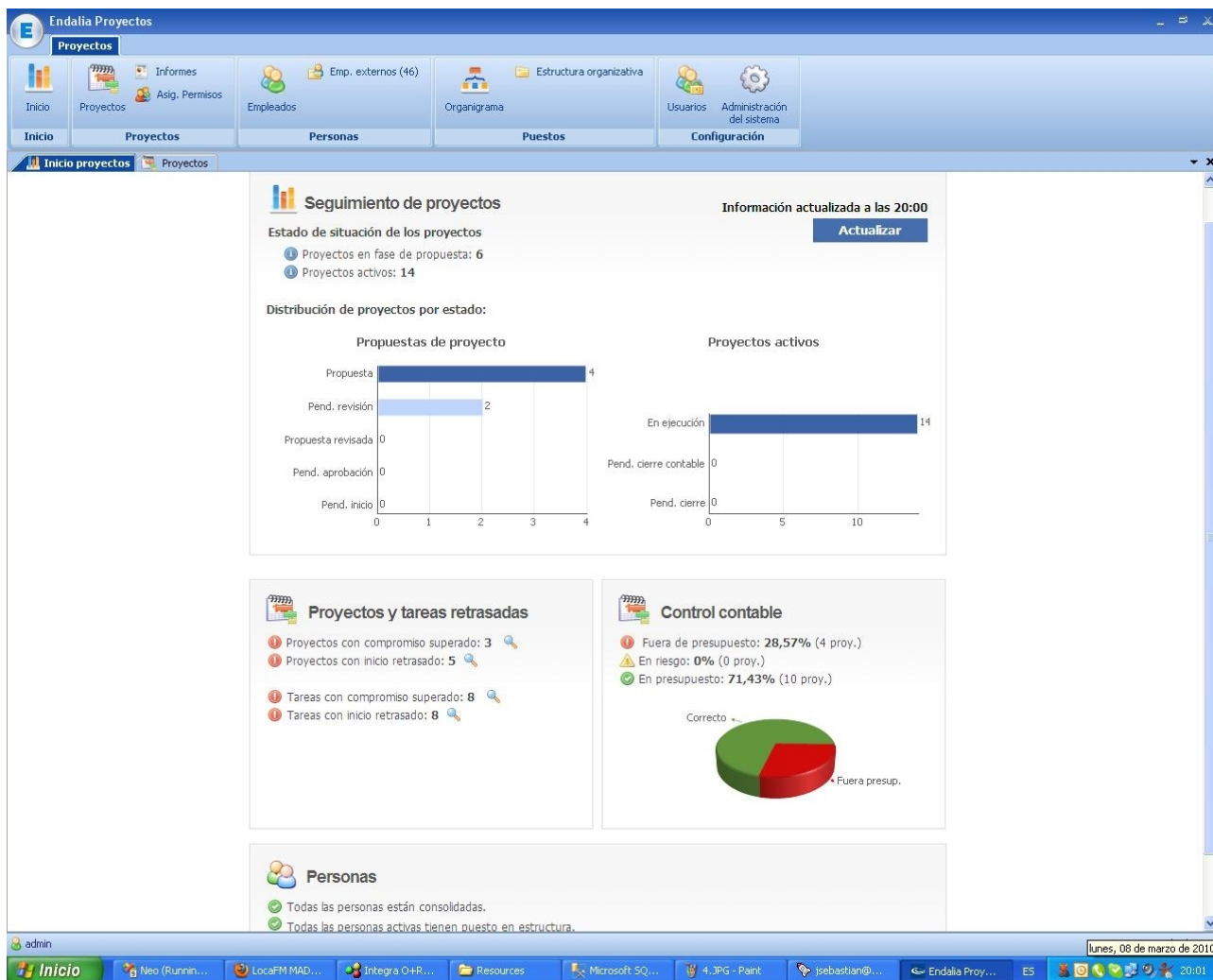



Figura 1. Página de inicio

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

7.3.2 CONFIGURACIÓN DEL SISTEMA

A continuación se muestra la pantalla de configuración del sistema, en la parte izquierda se muestran todos Campos parametrizables y en vista derecha un listado de los mismos.

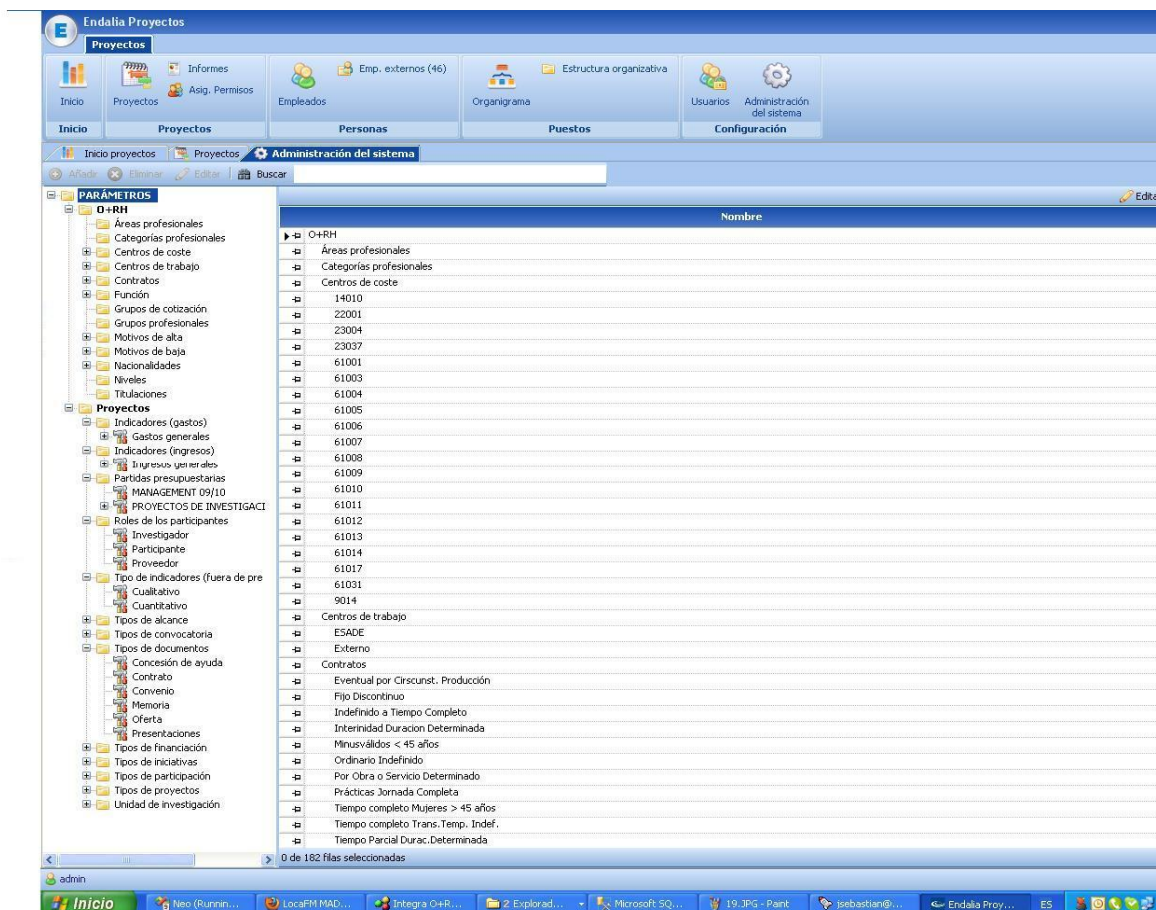

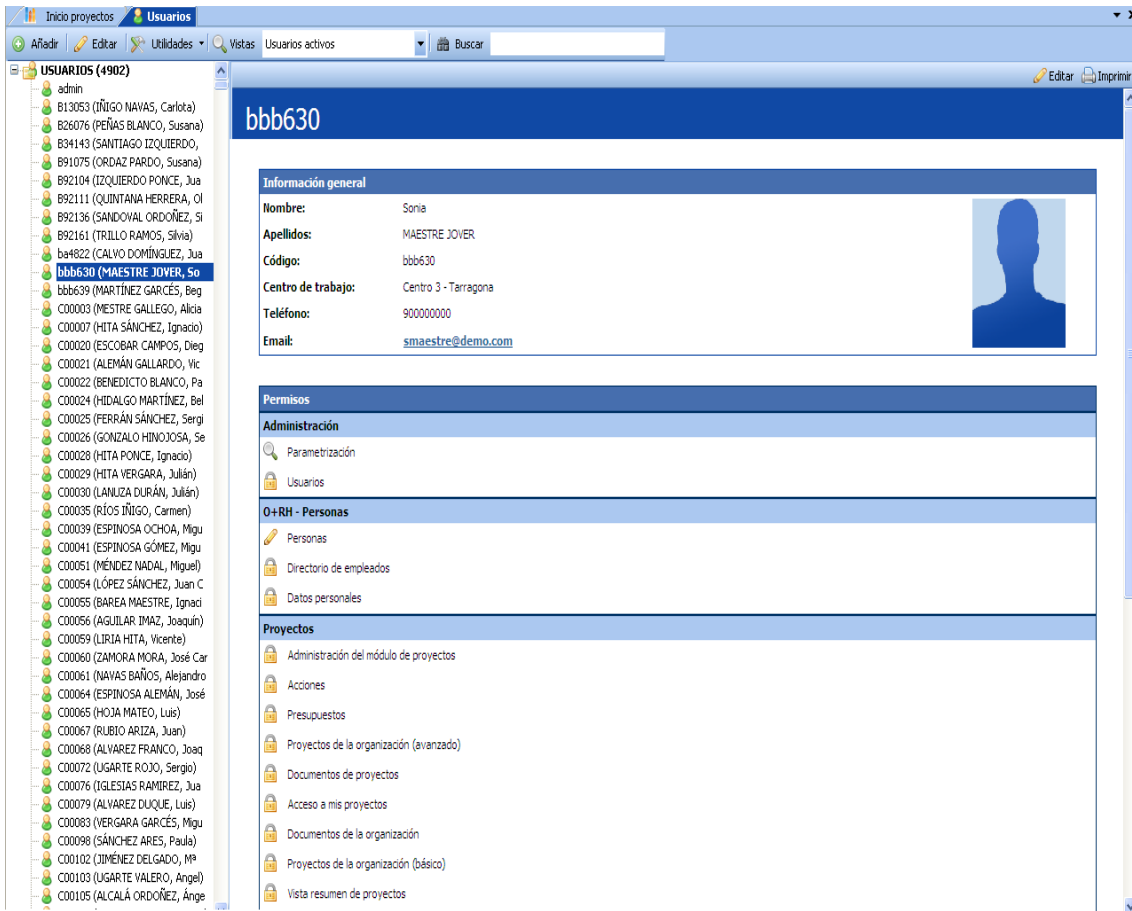


Figura 2. Panel configuración del sistema

7.3.3 GESTIÓN DE USUARIOS

A continuación se muestra la pantalla de gestión de usuarios, en la parte izquierda se muestran los usuarios y en parte derecha un informe HTML con los permisos de casa usuario y su información básica, y haciendo doble click en algún usuario se muestra su ficha donde editar la información.

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	



The screenshot displays the 'Usuarios' (Users) management interface. On the left, a list of users is shown, with 'bbb630 (MAESTRE JOVER, So)' selected. The right panel provides a detailed view of this user, including their general information, permissions, and associated projects.

Información general

Nombre:	Sonia
Apellidos:	MAESTRE JOVER
Código:	bbb630
Centro de trabajo:	Centro 3 - Tarragona
Teléfono:	900000000
Email:	smaestre@demo.com

Permisos

Administración

- Parametrización
- Usuarios

O+RH - Personas

- Personas
- Directorio de empleados
- Datos personales


Proyectos

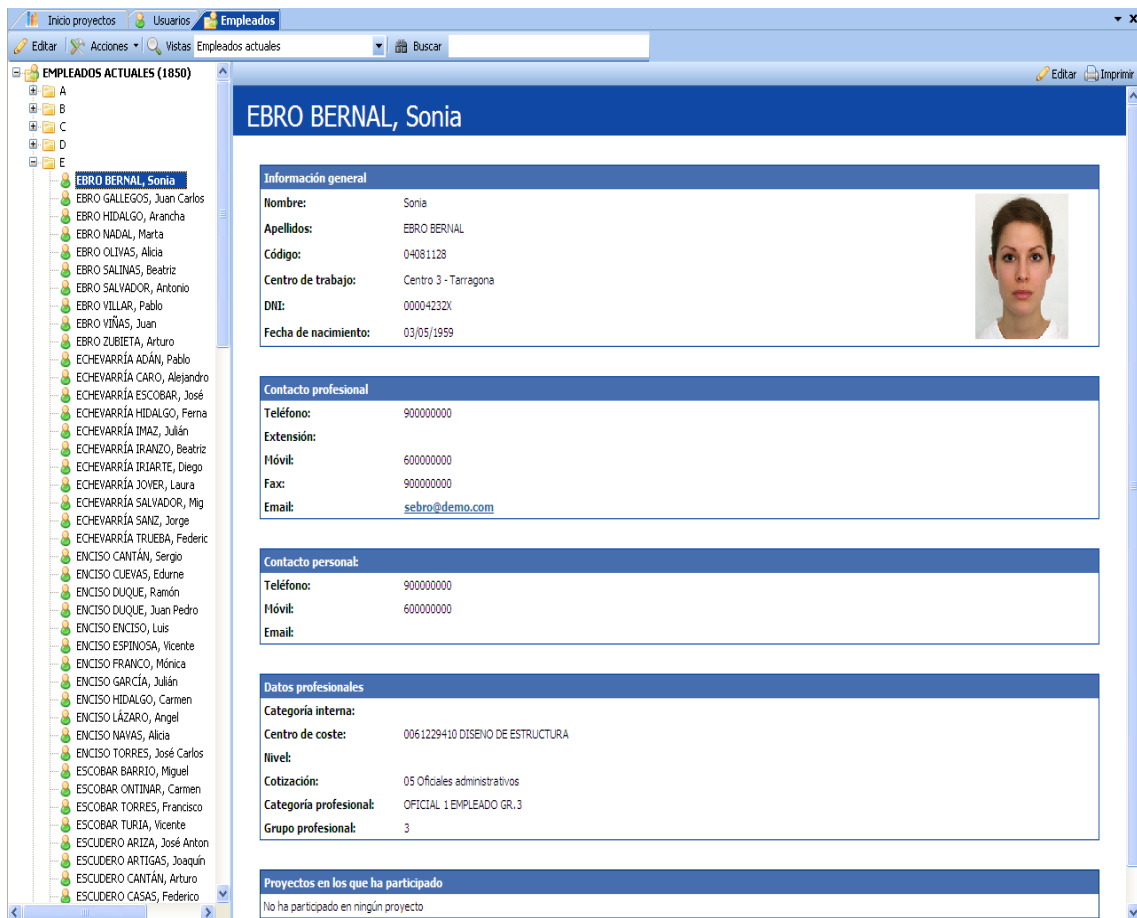
- Administración del módulo de proyectos
- Acciones
- Presupuestos
- Proyectos de la organización (avanzado)
- Documentos de proyectos
- Acceso a mis proyectos
- Documentos de la organización
- Proyectos de la organización (básico)
- Vista resumen de proyectos

Figura 3. Panel gestión de usuarios

7.3.4 GESTIÓN DE EMPLEADOS

A continuación se muestra la pantalla de gestión de empleados, en la parte izquierda se muestran los empleados y en parte derecha un informe HTML con los datos de cada empleado y los proyectos en los que participa, haciendo doble click en algún empleado se muestra su ficha donde editar la información.

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	



EMPLEADOS ACTUALES (1850)

- A
- B
- C
- D
- E
 - EBRO BERNAL, Sonia**
 - EBRO GALLEGOS, Juan Carlos
 - EBRO HIDALGO, Arancha
 - EBRO NADAL, Marta
 - EBRO OLIVAS, Alicia
 - EBRO SALINAS, Beatriz
 - EBRO SALVADOR, Antonio
 - EBRO VILLAR, Pablo
 - EBRO VIÑAS, Juan
 - EBRO ZUBIETA, Arturo
 - ECHEVARRÍA ADÁN, Pablo
 - ECHEVARRÍA CARO, Alejandro
 - ECHEVARRÍA ESCOBAR, José
 - ECHEVARRÍA HIDALGO, Ferna
 - ECHEVARRÍA IMAZ, Julián
 - ECHEVARRÍA IRANZO, Beatriz
 - ECHEVARRÍA IRIARTE, Diego
 - ECHEVARRÍA JOVER, Laura
 - ECHEVARRÍA SALVADOR, Mig
 - ECHEVARRÍA SANZ, Jorge
 - ECHEVARRÍA TRUEBA, Federic
 - ENCISO CANTÁN, Sergio
 - ENCISO CUEVAS, Ederne
 - ENCISO DUQUE, Ramón
 - ENCISO DUQUE, Juan Pedro
 - ENCISO ENCISO, Luis
 - ENCISO ESPINOSA, Vicente
 - ENCISO FRANCO, Mónica
 - ENCISO GARCÍA, Julián
 - ENCISO HIDALGO, Carmen
 - ENCISO LÁZARO, Angel
 - ENCISO NAVAS, Alicia
 - ENCISO TORRES, José Carlos
 - ESCOBAR BARRIO, Miguel
 - ESCOBAR ONTINAR, Carmen
 - ESCOBAR TORRES, Francisco
 - ESCOBAR TURIA, Vicente
 - ESCUDERO ARIZA, José Anton
 - ESCUDERO ARTIGAS, Joaquín
 - ESCUDERO CANTÁN, Arturo
 - ESCUDERO CASAS, Federico

EBRO BERNAL, Sonia

Información general

Nombre: Sonia
 Apellidos: EBRO BERNAL
 Código: 04081128
 Centro de trabajo: Centro 3 - Tarragona
 DNI: 00004232X
 Fecha de nacimiento: 03/05/1959

Contacto profesional

Teléfono: 900000000
 Extensión:
 Móvil: 600000000
 Fax: 900000000
 Email: sebro@demo.com

Contacto personal

Teléfono: 900000000
 Móvil: 600000000
 Email:

Datos profesionales

Categoría interna:
 Centro de coste: 0061229410 DISEÑO DE ESTRUCTURA
 Nivel:
 Cotización: 05 Oficiales administrativos
 Categoría profesional: OFICIAL 1 EMPLEADO GR.3
 Grupo profesional: 3


Proyectos en los que ha participado

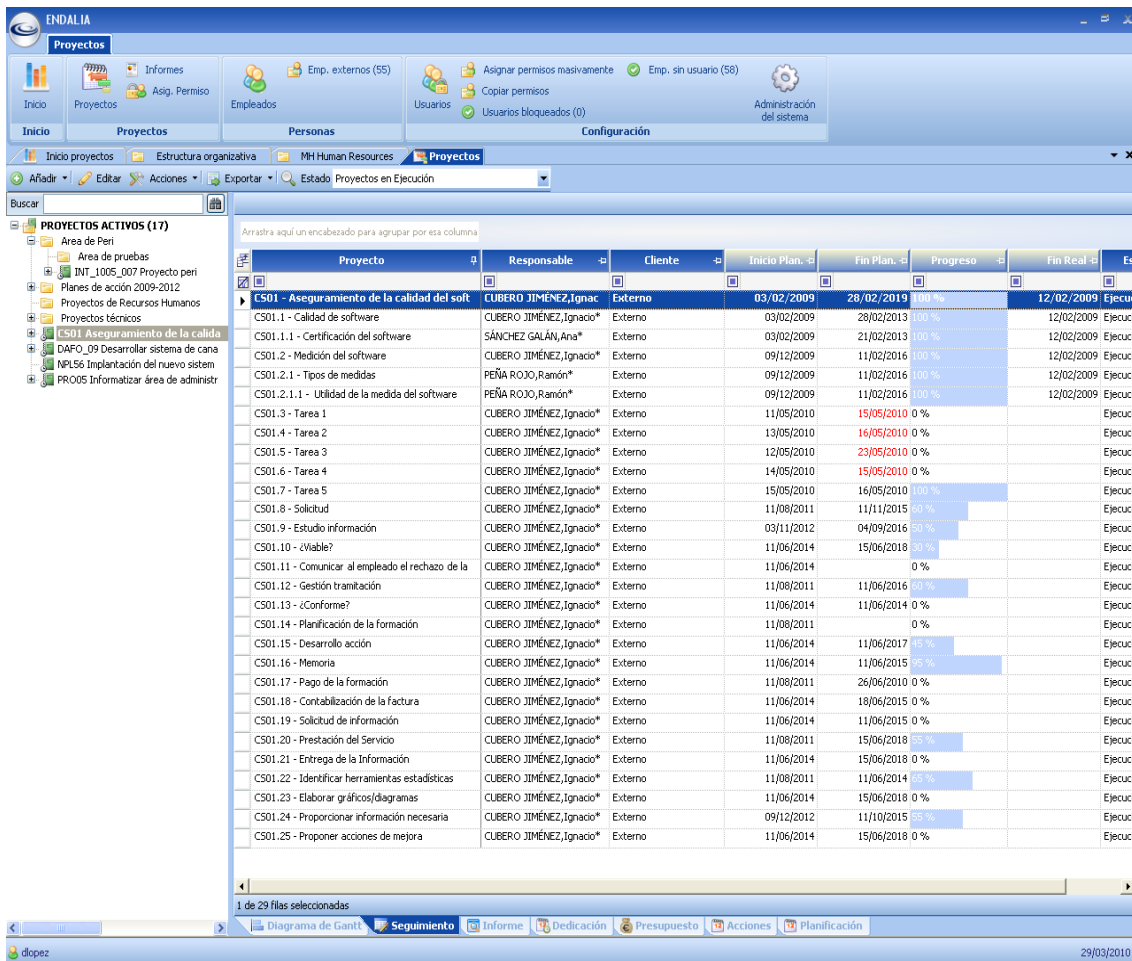
No ha participado en ningún proyecto

Figura 4. Panel gestión de empleados

7.3.5 GESTIÓN DE PROYECTOS

A continuación se muestran las pantallas más significativas que aparecen en la gestión de proyectos.

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	



PROYECTOS ACTIVOS (17)

- Area de Peri
 - INT_1005_007 Proyecto peri
 - Planes de acción 2009-2012
 - Proyectos de Recursos Humanos
- Proyectos técnicos
 - CS01 - Aseguramiento de la calidad**
 - DAFO_09 Desarrollar sistema de cana
 - NPL56 Implantación del nuevo sistem
 - PRO05 Informatizar área de administr


Proyecto	Responsable	Cliente	Inicio Plan.	Fin Plan.	Progreso	Fin Real	Estado
CS01 - Aseguramiento de la calidad del soft	CUBERO JIMÉNEZ, Ignacio*	Externo	03/02/2009	28/02/2019	100 %	12/02/2009	Ejecuci
CS01.1 - Calidad de software	CUBERO JIMÉNEZ, Ignacio*	Externo	03/02/2009	28/02/2013	100 %	12/02/2009	Ejecuci
CS01.1.1 - Certificación del software	SÁNCHEZ GALÁN, Ana*	Externo	03/02/2009	24/02/2013	100 %	12/02/2009	Ejecuci
CS01.2 - Medición del software	CUBERO JIMÉNEZ, Ignacio*	Externo	09/12/2009	11/02/2016	100 %	12/02/2009	Ejecuci
CS01.2.1 - Tipos de medidas	PEÑA ROJO, Ramón*	Externo	09/12/2009	11/02/2016	100 %	12/02/2009	Ejecuci
CS01.2.1.1 - Utilidad de la medida del software	PEÑA ROJO, Ramón*	Externo	09/12/2009	11/02/2016	100 %	12/02/2009	Ejecuci
CS01.3 - Tarea 1	CUBERO JIMÉNEZ, Ignacio*	Externo	11/05/2010	15/05/2010	0 %		Ejecuci
CS01.4 - Tarea 2	CUBERO JIMÉNEZ, Ignacio*	Externo	13/05/2010	16/05/2010	0 %		Ejecuci
CS01.5 - Tarea 3	CUBERO JIMÉNEZ, Ignacio*	Externo	12/05/2010	23/05/2010	0 %		Ejecuci
CS01.6 - Tarea 4	CUBERO JIMÉNEZ, Ignacio*	Externo	14/05/2010	15/05/2010	0 %		Ejecuci
CS01.7 - Tarea 5	CUBERO JIMÉNEZ, Ignacio*	Externo	15/05/2010	16/05/2010	100 %		Ejecuci
CS01.8 - Solicitud	CUBERO JIMÉNEZ, Ignacio*	Externo	11/08/2011	11/11/2015	0 %		Ejecuci
CS01.9 - Estudio información	CUBERO JIMÉNEZ, Ignacio*	Externo	03/11/2012	04/09/2016	50 %		Ejecuci
CS01.10 - ¿Viable?	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	15/06/2018	33 %		Ejecuci
CS01.11 - Comunicar al empleado el rechazo de la	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014		0 %		Ejecuci
CS01.12 - Gestión tramitación	CUBERO JIMÉNEZ, Ignacio*	Externo	11/08/2011	11/06/2016	0 %		Ejecuci
CS01.13 - ¿Conforme?	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	11/06/2014	0 %		Ejecuci
CS01.14 - Planificación de la formación	CUBERO JIMÉNEZ, Ignacio*	Externo	11/08/2011		0 %		Ejecuci
CS01.15 - Desarrollo acción	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	11/06/2017	45 %		Ejecuci
CS01.16 - Memoria	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	11/06/2015	0 %		Ejecuci
CS01.17 - Pago de la formación	CUBERO JIMÉNEZ, Ignacio*	Externo	11/08/2011	26/06/2010	0 %		Ejecuci
CS01.18 - Contabilización de la factura	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	18/06/2015	0 %		Ejecuci
CS01.19 - Solicitud de información	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	11/06/2015	0 %		Ejecuci
CS01.20 - Prestación del Servicio	CUBERO JIMÉNEZ, Ignacio*	Externo	11/08/2011	15/06/2018	55 %		Ejecuci
CS01.21 - Entrega de la Información	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	15/06/2018	0 %		Ejecuci
CS01.22 - Identificar herramientas estadísticas	CUBERO JIMÉNEZ, Ignacio*	Externo	11/08/2011	11/06/2014	0 %		Ejecuci
CS01.23 - Elaborar gráficos/diagramas	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	15/06/2018	0 %		Ejecuci
CS01.24 - Proporcionar información necesaria	CUBERO JIMÉNEZ, Ignacio*	Externo	09/12/2012	11/10/2015	55 %		Ejecuci
CS01.25 - Proponer acciones de mejora	CUBERO JIMÉNEZ, Ignacio*	Externo	11/06/2014	15/06/2018	0 %		Ejecuci

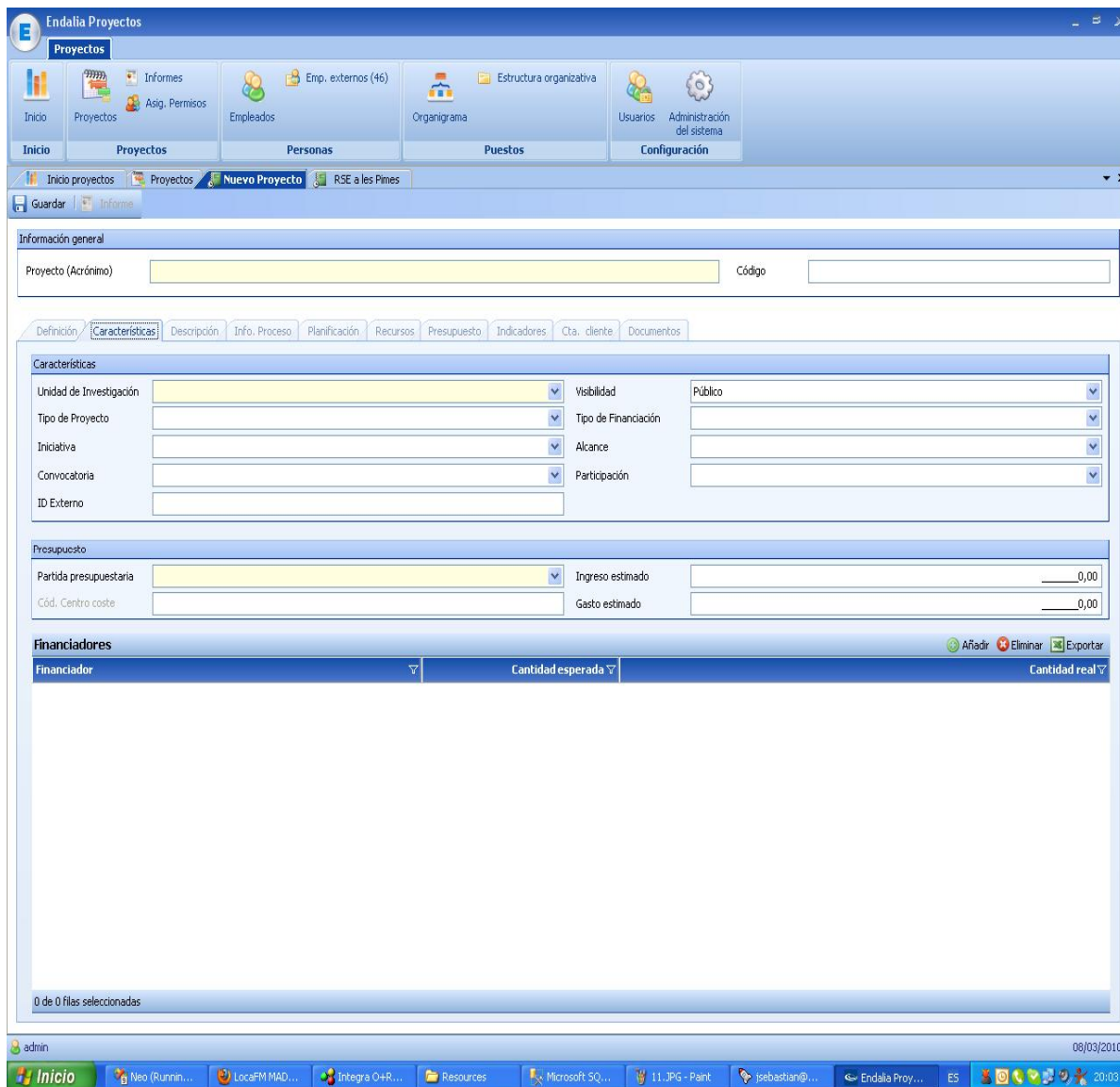
1 de 29 filas seleccionadas

Diagrama de Gantt Seguimiento Informe Dedicación Presupuesto Acciones Planificación

29/03/2010

Figura 5. Pantalla estructura gestión de proyectos

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	




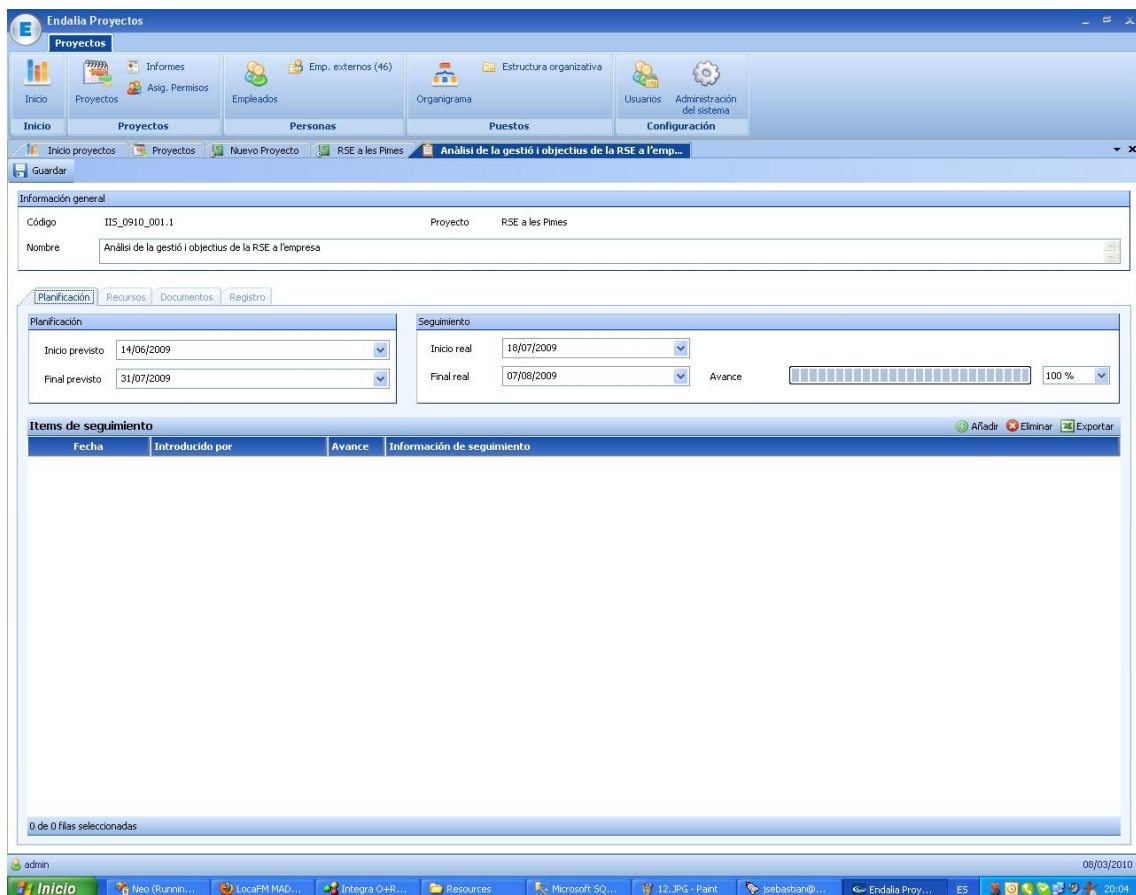
The screenshot shows the 'Endalia Proyectos' web application interface. The top navigation bar includes links for 'Inicio', 'Proyectos', 'Personas', 'Puestos', and 'Configuración'. The main content area is titled 'Nuevo Proyecto' and contains several sections:

- Información general:** Fields for 'Proyecto (Acronimo)' and 'Codigo'.
- Definición:** A tabbed interface with 'Características' selected. It includes fields for 'Unidad de Investigación', 'Tipo de Proyecto', 'Iniciativa', 'Convocatoria', 'ID Externo', 'Visibilidad', 'Tipo de Financiación', 'Alcance', and 'Participación'.
- Presupuesto:** Fields for 'Partida presupuestaria', 'Cód. Centro coste', 'Ingreso estimado', and 'Gasto estimado'.
- Financiadores:** A table with columns for 'Financiador', 'Cantidad esperada', and 'Cantidad real'. It includes buttons for 'Añadir', 'Eliminar', and 'Exportar'.

The bottom status bar shows the user 'admin' and the date '08/03/2010'. The Windows taskbar at the very bottom displays various open applications and the system clock at 20:03.

Figura 6. Ficha de proyecto

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	



Endalia Proyectos

Proyectos

Inicio | Proyectos | Informes | Asig. Permisos | Empleados | Emp. externos (46) | Estructura organizativa | Organigrama | Usuarios | Administración del sistema | Configuración

Inicio proyectos | Proyectos | Nuevo Proyecto | RSE a les Pimes | **Análisis de la gestión i objetivos de la RSE a l'emp...**

Guardar

Información general

Código: IIS_0910_001.1 Proyecto: RSE a les Pimes

Nombre: Análisis de la gestión i objetivos de la RSE a l'emp...

Planificación | Recursos | Documentos | Registro

Planificación

Inicio previsto: 14/06/2009

Final previsto: 31/07/2009

Seguimiento

Inicio real: 18/07/2009

Final real: 07/08/2009

Avance: 100 %

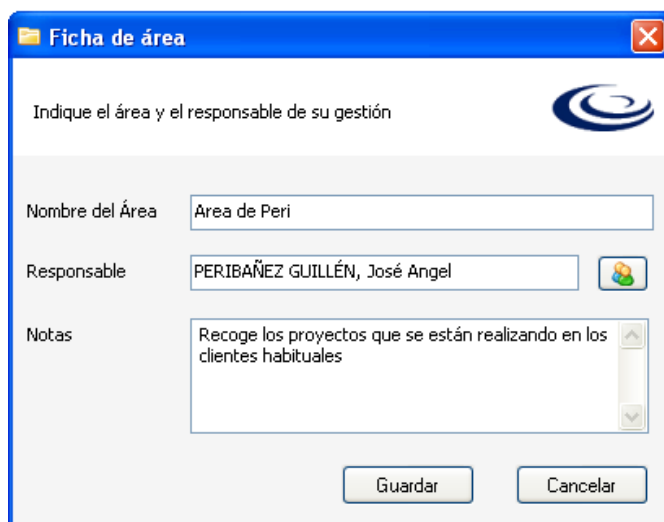
Items de seguimiento

Añadir Eliminar Exportar

Fecha	Introducido por	Avance	Información de seguimiento
0 de 0 filas seleccionadas			

08/03/2010

Figura 7. Ficha de tarea



Ficha de área

Indique el área y el responsable de su gestión


Nombre del Área: Area de Peri

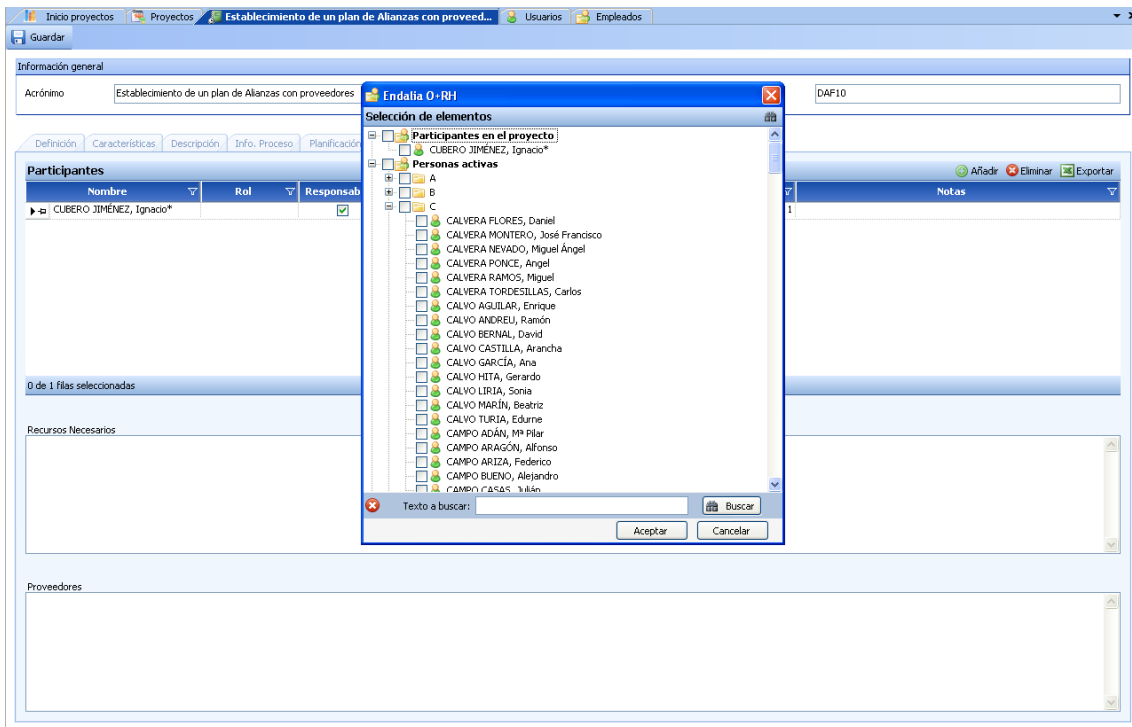
Responsable: PERIBÁÑEZ GUILLÉN, José Angel

Notas: Recoge los proyectos que se están realizando en los clientes habituales

Guardar **Cancelar**

Figura 8. Ficha de área

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	



Información general

Acronimo: Establecimiento de un plan de Alianzas con proveedores

Participantes

Nombre	Rol	Responsab
CUBERO JIMÉNEZ, Ignacio*		<input checked="" type="checkbox"/>

0 de 1 filas seleccionadas

Recursos Necesarios

Proveedores

Selección de elementos

Participantes en el proyecto

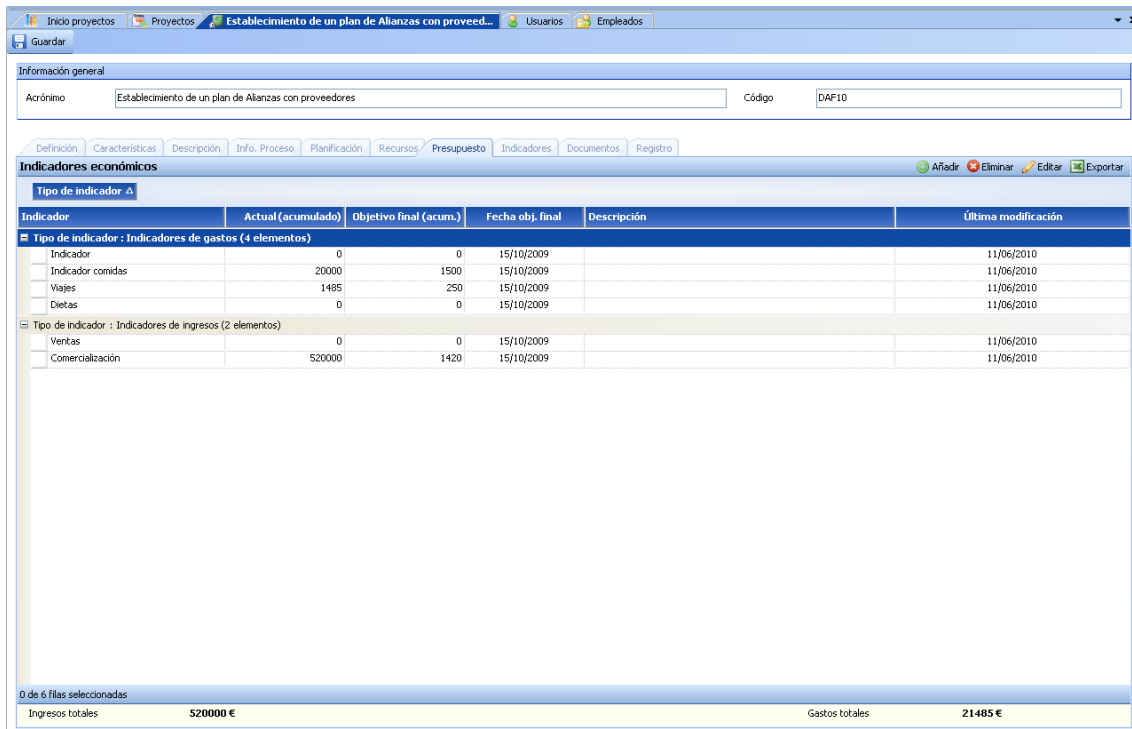
Personas activas

- CUBERO JIMÉNEZ, Ignacio*
- CALVERA FLORES, Daniel
- CALVERA MONTERO, José Francisco
- CALVERA NEVADO, Miguel Ángel
- CALVERA PONCE, Ángel
- CALVERA RAMOS, Miguel
- CALVERA TORDESILLAS, Carlos
- CALVO AGUILAR, Enrique
- CALVO ANDREU, Ramón
- CALVO BERNAL, David
- CALVO CASTILLA, Arancha
- CALVO GARCÍA, Ana
- CALVO HITIA, Gerardo
- CALVO LIRIA, Sonia
- CALVO MARÍN, Beatriz
- CALVO TURIA, Ederne
- CAMPO ADÁN, Mª Pilar
- CAMPO ARAUCO, Alfonso
- CAMPO ARIZA, Federico
- CAMPO BUENO, Alejandro
- CAMPO CASAS, Julián

Texto a buscar:

Buscar Aceptar Cancelar

Figura 9. Ficha de proyecto (detalle añadir nuevo empleado al proyecto)



Información general

Acronimo: Establecimiento de un plan de Alianzas con proveedores

Código: DAF10

Indicadores económicos

Tipo de indicador:


Indicador	Actual (acumulado)	Objetivo final (acum.)	Fecha obj. final	Descripción	Última modificación
Tipo de indicador : Indicadores de gastos (4 elementos)					
Indicador	0	0	15/10/2009		11/06/2010
Indicador comidas	20000	1500	15/10/2009		11/06/2010
Viajes	1485	250	15/10/2009		11/06/2010
Dietas	0	0	15/10/2009		11/06/2010
Tipo de indicador : Indicadores de ingresos (2 elementos)					
Ventas	0	0	15/10/2009		11/06/2010
Comercialización	520000	1420	15/10/2009		11/06/2010

0 de 6 filas seleccionadas

Ingresos totales: 520000 €

Gastos totales: 21485 €

Figura 10. Ficha de proyecto (detalle indicadores económicos)

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

7.3.6 INFORMES DIRECTOS

A continuación se muestran unos ejemplos de los informes HTML que genera la aplicación de forma directa al seleccionar unas de las distintas pestañas en la parte inferior de la ventana. En la primera imagen se muestra un ejemplo del informe del proyecto, y en la imagen siguiente un informe de presupuesto.

Aseguramiento de la calidad del software

Externo

PLANIFICACIÓN

- **INICIO PLANIFICADO:** 03/02/2009
- **FIN PLANIFICADO:** 28/02/2019

PARTICIPANTES

Responsable
CUBERO JIMÉNEZ, Ignacio*

Participante
BENITO ARES, Miguel Ángel
CUBERO JIMÉNEZ, Ignacio*
PEÑA ROJO, Ramón*
SÁNCHEZ GALÁN, Ana*

PRESUPUESTO

Resumen del presupuesto

	Planificado	Real
Ingresos	0,0 €	2.600,0 €
Gastos	0,0 €	1.200,0 €
Margen	● 0,0 €	📈 1.400,0 €

Figura 11. Informe html datos del proyecto

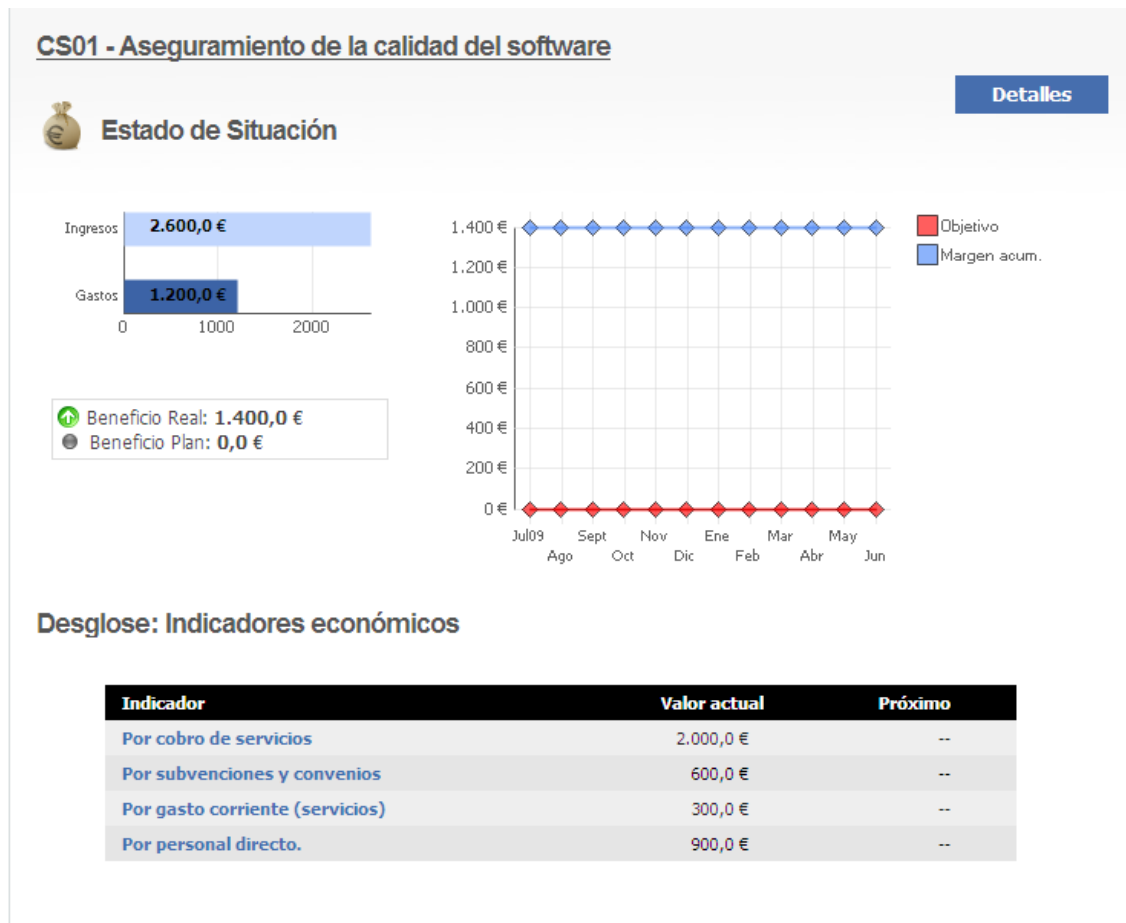



Figura 12. Informe html presupuesto del proyecto

7.3.7 GENERACIÓN DE INFORMES

A continuación se muestra la pantalla principal de la generación de informes complejos y personalizados, en la parte izquierda están los distintos informes a generar y en la parte derecha la selección de información y filtros que mediante una navegación por pestañas generan el informe final.

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

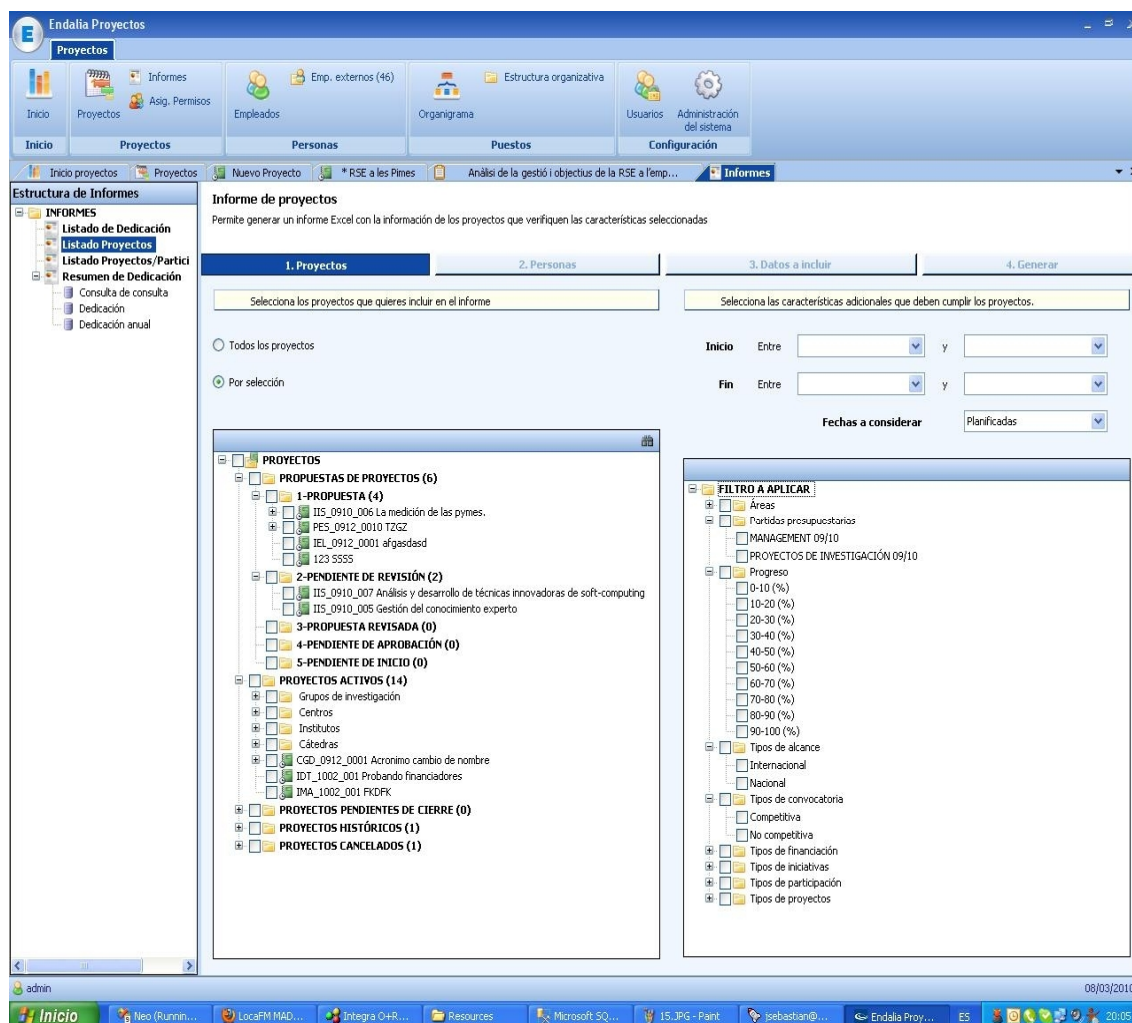



Figura 13. Pantalla generación de informes

7.3.8 DIAGRAMA GANTT

A continuación se muestran un ejemplo del diagrama de Gantt generado por la aplicación, desde el que se puede llevar a cabo la planificación de las tareas del proyecto. Para acceder al diagrama de forma directa debemos seleccionar un proyecto en la estructura organizativa, seleccionar la pestaña 'Diagrama de Gantt'.

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

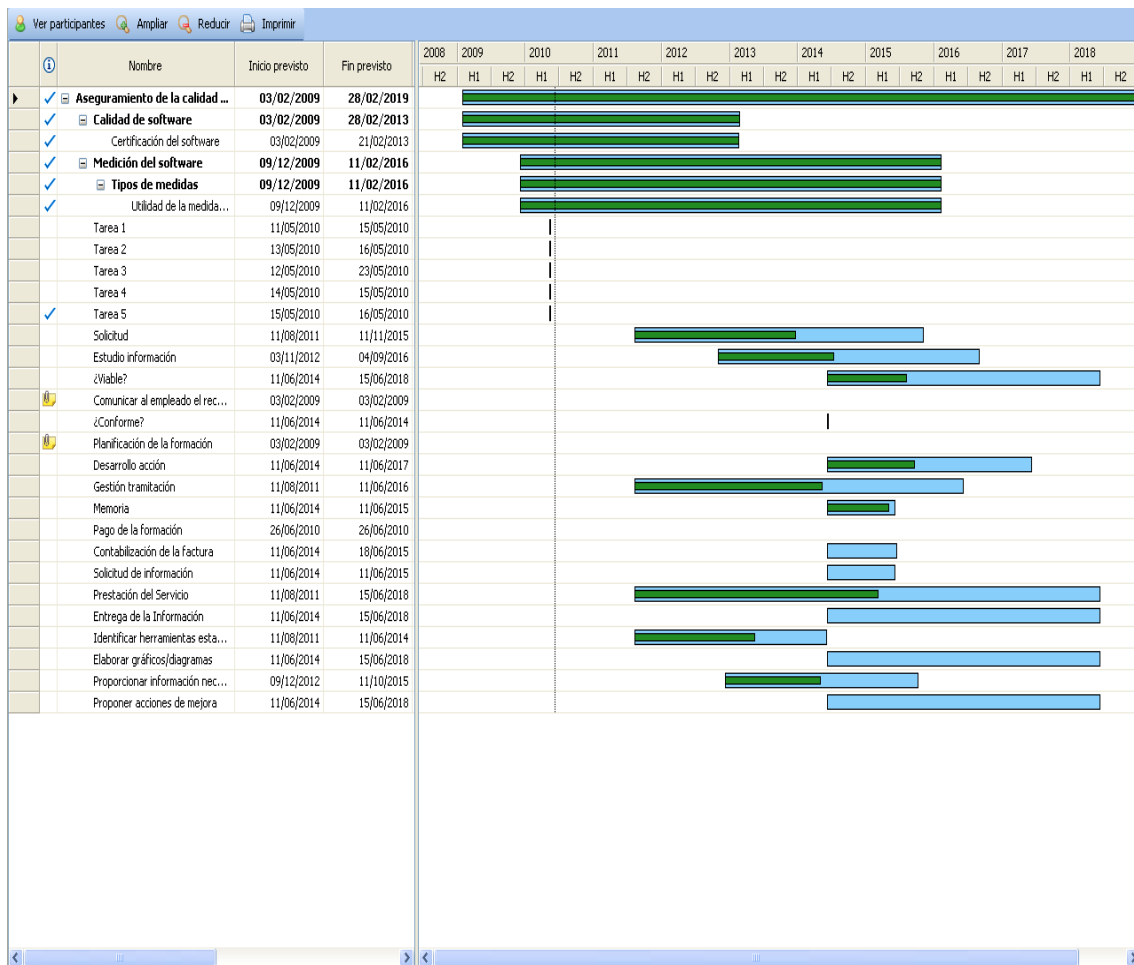



Figura 14: Diagrama de gantt

	Sistema de gestión de proyectos	Versión: 1.2
	Implementación	Fecha: 02/04/2010
	IMPLEMENTACION.doc	

8. BIBLIOGRAFÍA

8.1 REFERENCIAS

- [IGJ, 2000] I. Jacobson, G. Booch, J. Rambaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. Pearson Education
- [HHBK 2008] P. Henri Kuate, T. Harris, C. Bauer, G. King. "NHibernate in Action". Manning Publications 2008.


8.2 REFERENCIAS WEB

- [Ref. Web 1] <http://www.uml.org>
- [Ref. Web 2] <http://www.wikipedia.org>
- [Ref. Web 3] <http://nhforge.org/>
- [Ref. Web 4] <http://www.hibernate.org/>

Sistema para la gestión de proyectos

Plan de Pruebas

Versión 1.2 – Fecha: 03/05/2010

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
17/03/2010	1.0	Redacción inicial del documento	José Ángel Peribáñez
01/05/2010	1.1	Revisión del documento	Fernando Cortés
03/05/2010	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España



 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	4
1.1 PROPÓSITO DEL DOCUMENTO	4
1.2 ALCANCE DEL DOCUMENTO	4
1.3 ACRÓNIMOS	4
1.4 DEFINICIONES	4
1.5 REFERENCIAS	5
1.6 RESUMEN	5
2. DESCRIPCIÓN DEL PROCESO	6
3. PRUEBAS UNITARIAS Y DE INTEGRACIÓN	7
3.1 PRUEBAS DE CONFIGURACIÓN DEL SISTEMA	7
3.2 PRUEBAS DE GESTIÓN DE ÁREAS	8
3.3 PRUEBAS DE GESTIÓN DE PROYECTOS	10
3.4 PRUEBAS DE GESTIÓN DE TAREAS	13
3.5 PRUEBAS DE GESTIÓN DE PERSONAS	16
3.6 PRUEBAS DE GESTIÓN DE USUARIOS	17
3.7 PRUEBAS DE GESTIÓN DE INFORMES	18
3.8 PRUEBAS DE GESTIÓN DE INCIDENCIAS	20
3.9 PRUEBAS DE INTERNALIZACIÓN	20
4. PRUEBAS DE SISTEMA Y DE ACEPTACIÓN	21
4.1 PRUEBAS DE OPERACIÓN	21
4.2 PRUEBAS DE FACILIDAD DE USO	21
5. BIBLIOGRAFÍA	22
5.1 REFERENCIAS	22
4.3 REFERENCIAS WEB	22

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

1. INTRODUCCIÓN

1.1 PROPÓSITO DEL DOCUMENTO

El presente documento describe la fase de pruebas del SGP. Esta fase se entrelaza con la de implementación del sistema, y la complementa. Dicha implementación debe cumplir los requisitos planteados en el comienzo del desarrollo del sistema, así como operar de una forma correcta. El objetivo de esta fase se centra en certificar y asegurar la calidad del software.

1.2 ALCANCE DEL DOCUMENTO


En este documento se describen los resultados obtenidos durante la fase de pruebas del SGP.

1.3 ACRÓNIMOS

- PCS: Prueba de Configuración del Sistema.
- PGA: Prueba de Gestión de Áreas.
- PGP: Prueba de Gestión de Proyectos.
- PGT: Prueba de Gestión de Tareas.
- PGU: Prueba de Gestión de usuarios.
- PGPE: Prueba de Gestión de Personas.
- PGI: Prueba de Generación de Informes.
- PISIS: Prueba de Internacionalización del Sistema.
- SGP: Sistema de Gestión de proyectos.

1.4 DEFINICIONES

- Caso de prueba: conjunto de condiciones o variables bajo las cuales se determina si el requisito de una aplicación es parcial o completamente satisfactorio
- Prueba de caja negra: verificación del correcto cumplimiento de una funcionalidad especificada de un sistema.
- Prueba de integración: verificación de la correcta interconexión entre diferentes componentes de un sistema.
- Prueba de sistema: verificación global del funcionamiento correcto de un sistema.
- Prueba unitaria: verificación del correcto funcionamiento de un componente individual de un sistema.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

1.5 REFERENCIAS


En este documento se realizan referencias a los siguientes documentos del proyecto:

- ESPECIFICACIONREQUISITOS.doc: documento en el que se especifican los requisitos del sistema.
- ANALISIS.doc: documento de análisis del sistema.
- DISEÑO.doc: documento de diseño del sistema.
- IMPLEMENTACIÓN.doc: documento de implementación del sistema.

1.6 RESUMEN

Este documento describe el proceso de pruebas del SGP. Se compone de cinco apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describe el proceso de diseño seguido para la confección de este documento.
- Apartado 3: Se describen las pruebas unitarias y de integración realizadas.
- Apartado 4: Se describen las pruebas de sistema y de aceptación realizadas.
- Apartado 5: Bibliografía y referencias web utilizadas para la realización de este documento.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

2. DESCRIPCIÓN DEL PROCESO

El plan de pruebas tiene como objetivo verificar el resultado de la implementación, comprobando el sistema en sus construcciones internas, intermedias y en cuanto a su funcionalidad. Asimismo, desde esta fase se persigue obtener retroalimentación para la fase de implementación, con objeto de depurar el resultado de ésta.

En el caso del SGP, el proceso de pruebas se ha dividido en dos bloques principales:

- Pruebas unitarias y de integración.
- Pruebas de sistema y de aceptación.

La decisión de agrupar las pruebas unitarias y de integración viene dada por la naturaleza iterativa del proceso de desarrollo de la aplicación. La implementación de los diferentes módulos individuales está muy condicionada por la integración entre los mismos (no siendo posible, por ejemplo, la verificación del correcto funcionamiento del subsistema de gestión de informes si no se desarrolla el subsistema de gestión de proyectos). Debido a ello, y con el objetivo de evitar en lo posible el desarrollo de módulos auxiliares de prueba, se han combinado las pruebas de los componentes y módulos individuales con las pruebas de interconexión entre los mismos.


Dichas pruebas se han diseñado agrupándolas en los diferentes subsistemas identificados en las fases de análisis y diseño. Se ha hecho especial hincapié en la comprobación del correcto funcionamiento de las actividades marcadas por los requisitos del sistema, especificadas en el documento de especificación de requisitos, y por los casos de uso definidos en el documento de análisis del sistema.

Cada prueba estará diseñada como una prueba de “caja negra”, es decir, una prueba del comportamiento observable externamente del sistema. En cada prueba se deberá especificar el identificador de la misma, una breve descripción, el método de realización de la prueba, los diferentes casos especiales a comprobar, el criterio de aceptación de la prueba y el resultado de la misma.

En las pruebas de sistema y de aceptación resulta, por lo general, más complicado especificar los contenidos de las pruebas. Algunos de los conceptos a probar, como usabilidad u operación, resultan difícilmente definibles de manera cuantitativa, por lo que para esos casos se ha optado por realizar una descripción de los objetivos de dichas pruebas.

Por último, aunque es deseable que las pruebas sean llevadas a cabo por un equipo diferente al que ha desarrollado el sistema, debido al carácter académico de este proyecto, el proyectando debe ser el responsable del proceso, y por ello ha desempeñado esta tarea.

Asimismo, y en lo referente a pruebas de sistema y aceptación, el director del presente proyecto, Fernando Cortés Franco, y otros miembros de la organización en la que se ha desarrollado este proyecto, toman parte en las pruebas como evaluadores de conceptos tales como cumplimiento de objetivos, especificaciones y usabilidad.


	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

3. PRUEBAS UNITARIAS Y DE INTEGRACIÓN

3.1 PRUEBAS DE CONFIGURACIÓN DEL SISTEMA

Identificador	PCS – 01
Descripción	Añadir parámetro.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Configuración' del menú principal. 2. Pulsar el botón 'Añadir'. 3. Rellenar los datos del nuevo parámetro. 4. Pulsar el botón 'Guardar'.
Casos especiales a comprobar	<ul style="list-style-type: none"> • No se introduce ningún dato en el formulario de alta de parámetro (paso 3). Resultado esperado: el sistema advierte al usuario de que debe rellenar el formulario.
Criterio de aceptación	El nuevo parámetro es almacenado en el sistema, y visible tanto en el árbol izquierdo como en el listado derecho.
Resultado de la prueba	Aceptada.


Identificador	PCS – 02
Descripción	Eliminar parámetro.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Configuración' del menú principal. 2. Seleccionar un parámetro en el árbol de estructura, que no esté siendo usado en ningún otro lugar de la aplicación. 3. Pulsar el botón 'Eliminar'. 4. En el diálogo de confirmación de borrado, pulsar el botón 'Sí'. 5. Seleccionar un parámetro en el árbol de estructura que esté siendo usado en la aplicación (por ejemplo: Nacionalidad – Española). 6. Pulsar el botón 'Eliminar'.
Criterio de aceptación	<p>Si el parámetro no está siendo usado en la aplicación, será eliminado del sistema.</p> <p>Si el parámetro está siendo usado, se advertirá al usuario de la imposibilidad de eliminar dicho parámetro, y no será eliminado.</p>
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PCS – 03
Descripción	Modificar parámetro.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Configuración' del menú principal. 2. Seleccionar un parámetro en el árbol de estructura. 3. Pulsar el botón 'Editar'. 4. Modificar los datos del parámetro. 5. Pulsar el botón 'Guardar'
Casos especiales a comprobar	<ul style="list-style-type: none"> • No se introduce ningún dato en el formulario de edición del parámetro (paso 4). Resultado esperado: el sistema advierte al usuario de que debe rellenar el formulario.
Criterio de aceptación	El sistema almacena correctamente los nuevos datos del parámetro.
Resultado de la prueba	Aceptada.


3.2 PRUEBAS DE GESTIÓN DE ÁREAS

Identificador	PGA – 01
Descripción	Añadir nuevo área al sistema.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Proyectos' en el menú principal 2. Seleccionar la opción 'Añadir área' dentro del botón 'Nuevo' del menú principal de proyectos. 3. Introducir datos del área. 4. Pulsar botón 'Aceptar'.
Casos especiales a comprobar	<ul style="list-style-type: none"> • Pulsar 'Aceptar' o tecla 'enter' sin introducir ningún dato. Resultado esperado: el sistema avisa mediante un mensaje de la necesidad de escribir un nombre. • Añadir un área con un nombre ya existente en el sistema. Resultado esperado: el sistema avisa mediante un mensaje que el nombre ya está siendo un usado por otra área.
Criterio de aceptación	El sistema crea correctamente la nueva área.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGA- 02
Descripción	Editar datos de área.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Proyectos' en el menú principal 2. Hacer doble click en un área o seleccionar la opción 'editar' dentro del menú 'Acciones' con un área seleccionada 3. Modificar datos del área 4. Pulsar botón 'Aceptar'
Casos especiales a comprobar	<ul style="list-style-type: none"> • Pulsar 'Aceptar' con algún dato obligatorio borrado. Resultado esperado: el sistema avisa mediante un mensaje de la necesidad de escribir ese campo. • Modificar el nombre de área por el de algún área ya existente en el sistema y pulsar aceptar. Resultado esperado: el sistema avisa mediante un mensaje que el nombre ya está siendo un usado por otra área.
Criterio de aceptación	Modificar correctamente los datos del área.
Resultado de la prueba	Aceptada.


Identificador	PGA- 03
Descripción	Eliminar área.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Proyectos' en el menú principal 2. Seleccionar un área 3. Seleccionar la opción 'Eliminar' dentro del menú 'Acciones'
Casos especiales a comprobar	<ul style="list-style-type: none"> • Seleccionar un área que contiene proyectos. Resultado esperado: el sistema avisa mediante un mensaje que el área no está vacía para ser eliminada.
Criterio de aceptación	Se elimina correctamente el área del sistema.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

3.3 PRUEBAS DE GESTIÓN DE PROYECTOS

Identificador	PGP – 01
Descripción	Añadir nuevo proyecto al sistema.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Proyectos' en el menú principal. 2. Seleccionar la opción 'Añadir'. 3. Seleccionar la opción 'Nuevo proyecto' dentro del menú 'Añadir'. 4. Rellenar los datos en la ficha de proyecto 5. Pulsar botón 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • El proyecto aparece en la estructura jerárquica. • Abrimos la ficha del proyecto y comprobamos que los datos que hemos introducido aparecen correctamente.
Resultado de la prueba	Aceptada.


Identificador	PGP – 02
Descripción	Editar los datos del proyecto.
Método	<ol style="list-style-type: none"> 1. Seleccionar un proyecto de la estructura jerárquica. 2. En el botón 'Acciones' seleccionar la opción 'Editar' o hacer doble click sobre el nodo del proyecto en la estructura. 3. En la ficha del proyecto modificados todos los datos posibles. 4. Pulsamos el botón 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • En la pantalla derecha, listado de proyectos aparece la información actualizada. • Abrimos la ficha de proyecto y la información coincide con la que hemos actualizado. • Cambiamos de vista varias veces y volvemos a abrir la ficha del proyecto para comprobar que la información sigue siendo la misma.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGP – 03
Descripción	Cancelar un proyecto.
Método	<ol style="list-style-type: none"> 1. Seleccionamos un proyecto en la estructura jerárquica. 2. Elegimos la opción 'Cancelar' dentro del menú de 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • El proyecto no debe estar previamente en estado cancelado. • El proyecto cambia de estado, desaparece de su vista actual y aparece en la vista de proyectos cancelados.
Resultado de la prueba	Aceptada.

Identificador	PGP – 04
Descripción	Activar un proyecto.
Método	<ol style="list-style-type: none"> 1. Seleccionamos un proyecto cancelado de la estructura jerárquica. 2. Seleccionamos la opción 'Activar' dentro del menú de 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • La opción activar solo se puede seleccionar si el proyecto seleccionado esta cancelado. • Una vez que el proyecto ha sido activado, aparece como el resto de proyectos en la estructura jerárquica de la derecha y en el listado de la izquierda.
Resultado de la prueba	Aceptada.


Identificador	PGP – 05
Descripción	Eliminar un proyecto.
Método	<ol style="list-style-type: none"> 1. Seleccionar un proyecto de la estructura jerárquica. 2. En el botón 'Acciones' seleccionar la opción 'Eliminar'.
Criterio de aceptación	<ul style="list-style-type: none"> • El proyecto solo puede ser eliminar si no tiene acciones o esfuerzos introducidos. • Una vez eliminado el proyecto ya no aparece en la estructura jerárquica de la izquierda ni en el listado de la derecha.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGP – 06
Descripción	Cambiar estado de un proyecto.
Método	<ol style="list-style-type: none"> 1. Seleccionar un proyecto de la estructura jerárquica. 2. En el botón 'Acciones' seleccionar una de las opciones de cambiar de estado, a un estado previo o posterior.
Criterio de aceptación	<ul style="list-style-type: none"> • Los cambios de estado son dependientes del estado del proyecto. • El proyecto ha cambiado a su nuevo estado.
Resultado de la prueba	Aceptada.

Identificador	PGP – 07
Descripción	Mover un proyecto dentro de la estructura jerárquica
Método	<ol style="list-style-type: none"> 1. Seleccionar un proyecto de la estructura jerárquica. 2. Arrastramos el proyecto con el ratón hasta su nueva posición en la estructura. 3. Selecciona la nueva ubicación dentro de las posibilidades.
Criterio de aceptación	<ul style="list-style-type: none"> • Un proyecto no se puede desplazar dentro de una tarea o dentro de otro proyecto. • El proyecto se desplazará a otra posición dentro de un área o a otra área. • El proyecto aparecerá en su nueva ubicación en la estructura jerárquica de la parte izquierda.
Resultado de la prueba	Aceptada.


Identificador	PGP – 08
Descripción	Duplicar proyecto
Método	<ol style="list-style-type: none"> 1. Seleccionar un proyecto de la estructura jerárquica. 2. Seleccionar la opción 'Duplicar elemento' dentro del menú 'Acciones'. 3. Seleccionar las opciones del menú de duplicado y seleccionar la opción 'Duplicar'.
Criterio de aceptación	<ul style="list-style-type: none"> • Junto al proyecto original aparece el nuevo proyecto con el nuevo código elegido para el nuevo proyecto y respetando la estructura del original mismo.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

3.4 PRUEBAS DE GESTIÓN DE TAREAS

Identificador	PGT – 01
Descripción	Añadir nueva tarea en un proyecto.
Método	<ol style="list-style-type: none"> 1. Seleccionamos un proyecto o tarea en la estructura. 2. Seleccionar la opción 'Nuevo Tarea' dentro del menú 'Añadir'. 3. Rellenar los datos en la ficha de la tarea 4. Pulsar botón 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • La tarea aparece en la estructura debajo del proyecto/tarea seleccionado. • La tarea tiene un código correlativo a su proyecto/tarea padre • Abrimos la ficha de la tarea y comprobamos que los datos que hemos introducido aparecen correctamente.
Resultado de la prueba	Aceptada.


Identificador	PGT – 02
Descripción	Editar los datos de tarea.
Método	<ol style="list-style-type: none"> 1. Seleccionar una tarea de la estructura jerárquica. 2. En el botón 'Acciones' seleccionar la opción 'Editar' o hacer doble click sobre el nodo de la tarea. 3. En la ficha de la tarea modificados todos los datos posibles. 4. Pulsamos el botón 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • En la pantalla derecha, listado de proyectos aparece la información de la tarea actualizada. • Abrimos la ficha de la tarea y la información coincide con la que hemos actualizado. • Cambiamos de vista varias veces y volvemos a abrir la ficha de la tarea para comprobar que la información sigue siendo la misma.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGT – 03
Descripción	Cancelar una tarea.
Método	<ol style="list-style-type: none"> 1. Seleccionamos una tarea en la estructura jerárquica. 2. Elegimos la opción 'Cancelar' dentro del menú de 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • La tarea debe estar en ejecución y activa para poder ser cancelada. • La tarea aparece en la estructura jerárquica con un nuevo icono y letra en color rojo para identificarla fácilmente como tarea cancelada.
Resultado de la prueba	Aceptada.


Identificador	PGT – 04
Descripción	Activar una tarea.
Método	<ol style="list-style-type: none"> 1. Seleccionamos una tarea cancelada en la estructura jerárquica. 2. Seleccionamos la opción 'Activar' dentro del menú de 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • La opción activar solo se puede seleccionar si la tarea esta cancelada. • Una vez que la tarea ha sido activada, aparece como el resto de tareas en la estructura jerárquica de la derecha y en el listado de la izquierda.
Resultado de la prueba	Aceptada.

Identificador	PGT – 05
Descripción	Eliminar una tarea.
Método	<ol style="list-style-type: none"> 1. Seleccionar una tarea de la estructura jerárquica. 2. En el botón 'Acciones' seleccionar la opción 'Eliminar'.
Criterio de aceptación	<ul style="list-style-type: none"> • La tarea solo puede ser eliminar si no tiene acciones o esfuerzos introducidos. • Una vez eliminada la tarea ya no aparece en la estructura jerárquica de la izquierda ni en el listado de la derecha. • El código de las tareas por detrás de la tarea eliminada se actualizan siguiendo el orden correlativo.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGT – 08
Descripción	Duplicar tarea
Método	<ol style="list-style-type: none"> 1. Seleccionar una tarea de la estructura jerárquica. 2. Seleccionar la opción 'Duplicar elemento' dentro del menú 'Acciones'. 3. Seleccionar las opciones del menú de duplicado y seleccionar la opción 'Duplicar'.
Criterio de aceptación	<ul style="list-style-type: none"> • La tarea duplicada aparece justo detrás de la tarea original, con un código correlativo a la tarea original.
Resultado de la prueba	Aceptada.

Identificador	PGT – 07
Descripción	Mover una tarea dentro de la estructura jerárquica
Método	<ol style="list-style-type: none"> 1. Seleccionar una tarea de la estructura jerárquica. 2. Arrastramos la tarea con el ratón hasta su nueva posición en la estructura. 3. Selecciona la nueva ubicación dentro las posibilidades.
Criterio de aceptación	<ul style="list-style-type: none"> • Una tarea solo se puede mover dependiendo de proyectos o tareas. • La tarea aparece en su nueva ubicación y el código de esta tarea y sus tareas por debajo y detrás actualizan su código correlativamente.
Resultado de la prueba	Aceptada.


	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

3.5 PRUEBAS DE GESTIÓN DE PERSONAS

Identificador	PGPE – 01
Descripción	Añadir nueva persona.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Empleados' dentro del menú de la aplicación. 2. Seleccionar la opción 'Nuevo empleado' dentro del menú 'Añadir'. 3. Rellenar todos los datos de la ficha del empleado 4. Seleccionar la opción 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • La nueva persona añadida aparece en el árbol de la izquierda dentro de la carpeta de la inicial de su apellido. • El sistema avisa si se introducen datos incorrectos como DNI, o si falta algún dato obligatorio por introducir.
Resultado de la prueba	Aceptada.

Identificador	PGPE – 02
Descripción	Editar datos persona.
Método	<ol style="list-style-type: none"> 1. Seleccionar una persona en la estructura de la parte izquierda. 2. Seleccionar la opción 'Editar' dentro del menú acciones o hacer doble click sobre el empleado seleccionado. 3. Modificar todos los datos posibles de la ficha de la persona 4. Seleccionar la opción 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • Los datos aparecen en la vista izquierda y en pantalla derecha. • Si abrimos la ficha del empleado aparecen los datos modificados.
Resultado de la prueba	Aceptada.

Identificador	PGPE – 03
Descripción	Cambiar estado de una persona.
Método	<ol style="list-style-type: none"> 1. Seleccionar una persona en la estructura de la parte izquierda. 2. Seleccionar la opción 'Cambiar de estado' dentro del menú 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • La persona no aparece en la vista actual. • Al cambiar a la vista correspondiente al nuevo estado del empleado, aparece en la estructura de parte izquierda dentro de la carpeta de la inicial de su apellido.
Resultado de la prueba	Aceptada.


	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGPE – 04
Descripción	Eliminar una persona.
Método	<ol style="list-style-type: none"> 1. Vamos a la vista de empleados históricos 2. Seleccionar una persona en la estructura de la parte izquierda. 3. Seleccionar la opción 'Eliminar' dentro del menú 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • La persona no aparece en la vista actual. • La persona ha desaparecido de todas secciones donde aparecía.
Resultado de la prueba	Aceptada.

3.6 PRUEBAS DE GESTIÓN DE USUARIOS

Identificador	PGU – 01
Descripción	Añadir nuevo usuario.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Usuarios' dentro del menú de la aplicación. 2. Seleccionar la opción 'Nuevo usuario' dentro del menú 'Añadir'. 3. Rellenar todos los datos de la ficha del usuario 4. Seleccionar la opción 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario creado debe estar asociado a un empleado. • El nuevo usuario añadido aparece en el árbol de la izquierda y en la lista de vista derecha.
Resultado de la prueba	Aceptada.


Identificador	PGU – 02
Descripción	Editar datos usuario.
Método	<ol style="list-style-type: none"> 1. Seleccionar un usuario en la estructura de la parte izquierda. 2. Seleccionar la opción 'Editar' dentro del menú acciones o hacer doble click sobre el usuario seleccionado. 3. Modificar todos los datos posibles de la ficha del usuario 4. Seleccionar la opción 'Guardar'
Criterio de aceptación	<ul style="list-style-type: none"> • Los datos aparecen en la vista izquierda y en pantalla derecha. • Si abrimos la ficha del usuario aparecen los datos modificados.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGU – 03
Descripción	Eliminar usuario.
Método	<ol style="list-style-type: none"> 1. Seleccionar un usuario en la estructura de la parte izquierda. 2. Seleccionar la opción 'Eliminar' dentro del menú 'Acciones'.
Criterio de aceptación	<ul style="list-style-type: none"> • El usuario no aparece en la vista actual. • Si intentamos acceder al sistema con ese usuario no podemos.
Resultado de la prueba	Aceptada.


3.7 PRUEBAS DE GESTIÓN DE INFORMES

Identificador	PGINF - 01
Descripción	Generar informe de proyectos.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Informes' del menú principal de la aplicación. 2. Seleccionar la opción 'Informe proyectos' en la estructura de informes de parte izquierda. 3. Seleccionar los proyectos que aparecen el informe. 4. Seleccionar las características por los que queremos filtrar los proyectos. 5. Seleccionar los datos a mostrar en el informe
Criterio de aceptación	El informe seleccionado es visualizado en pantalla.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

Identificador	PGINF - 02
Descripción	Generar informe participantes en proyectos.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Informes' del menú principal de la aplicación. 2. Seleccionar la opción 'Informe participantes' en la estructura de informes de parte izquierda. 3. Seleccionar los proyectos que aparecen el informe. 4. Seleccionar las personas que queremos que aparezcan en el informe. 5. Seleccionar las características por los que queremos filtrar los proyectos. 6. Seleccionar las características por las que queremos filtrar las participantes. 7. Seleccionar los datos a mostrar en el informe.
Criterio de aceptación	El informe seleccionado es visualizado en pantalla.
Resultado de la prueba	Aceptada.

Identificador	PGINF - 02
Descripción	Generar informe de esfuerzos.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Informes' del menú principal de la aplicación. 2. Seleccionar la opción 'Informe esfuerzos' en la estructura de informes de parte izquierda. 3. Seleccionar los proyectos que aparecen el informe. 4. Seleccionar las características por los que queremos filtrar los proyectos. 5. Seleccionar los datos a mostrar en el informe.
Criterio de aceptación	El informe seleccionado es visualizado en pantalla.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	


Identificador	PGINF - 02
Descripción	Generar informe de acciones.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción 'Informes' del menú principal de la aplicación. 2. Seleccionar la opción 'Informe acciones' en la estructura de informes de parte izquierda. 3. Seleccionar los proyectos que aparecen el informe. 4. Seleccionar las características por los que queremos filtrar los proyectos.
Criterio de aceptación	El informe seleccionado es visualizado en pantalla.
Resultado de la prueba	Aceptada.

3.8 PRUEBAS DE GESTIÓN DE INCIDENCIAS

Identificador	PGINC - 01
Descripción	Comprobar escritura de log.
Método	<ul style="list-style-type: none"> • Abrir el directorio donde se almacena el registro de log (indicado en el fichero de configuración). • Abrir el documento más reciente almacenado.
Criterio de aceptación	<ul style="list-style-type: none"> • El directorio contiene ficheros de log. • El fichero de log abierto contiene información asociada a las últimas operaciones realizadas en la aplicación.
Resultado de la prueba	Aceptada.

3.9 PRUEBAS DE INTERNALIZACIÓN

Identificador	PGINT - 01
Descripción	Cambiar el idioma de la aplicación
Método	<ul style="list-style-type: none"> • Modificamos el fichero app.config para generar el fichero de etiquetas de la aplicación en otro idioma (p.e inglés). • Lanzamos la aplicación.
Criterio de aceptación	<ul style="list-style-type: none"> • Todos los menús, mensajes y etiquetas de la aplicación aparecen en el nuevo idioma.
Resultado de la prueba	Aceptada.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	


4. PRUEBAS DE SISTEMA Y DE ACEPTACIÓN

4.1 PRUEBAS DE OPERACIÓN

En estas pruebas se ha determinado el cumplimiento de los procedimientos de inicio y fin de trabajo y el cumplimiento de las especificaciones y requisitos de la metodología de gestión de que se establece con la utilización del SGP desarrollado.

4.2 PRUEBAS DE FACILIDAD DE USO

En estas pruebas se ha comprobado la adaptabilidad del sistema a las necesidades de los usuarios y al método habitual de trabajo con una herramienta informática común. Se ha verificado asimismo que las interfaces gráficas que componen el sistema, y con las que interactúa el usuario, son adecuadas, sencillas y agradables visualmente.

	Sistema para la gestión de proyectos	Versión: 1.2
	Plan de Pruebas	Fecha: 03/05/2010
	PRUEBAS.doc	

5. BIBLIOGRAFÍA

5.1 REFERENCIAS

[DUS 2002] Elfriede Dustin 2002. *Effective Software Testing*. Ed. Addison Wesley.


4.3 REFERENCIAS WEB

[Ref. Web 1] <http://www.wikipedia.org>

Sistema para la gestión de proyectos

Manual de Usuario

Versión 1.2 – Fecha: 21/05/2010

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

REVISIONES

Fecha	Versión	Descripción	Autor
20/05/2010	1.0	Redacción inicial del documento	José Ángel Peribáñez
21/05/2010	1.1	Revisión del documento	Fernando Cortés
21/05/2010	1.2	Corrección del documento	José Ángel Peribáñez

Copyright © 2009, ENDALIA, S.L. Todos los derechos reservados.

Este documento contiene información propietaria de ENDALIA, S.L. Se emite con el único propósito de informar proyectos Integra, por lo que no se ofrece ninguna garantía explícita o implícita. Ninguna parte de esta publicación puede ser utilizada para cualquier otro propósito, y no debe ser reproducida, copiada, adaptada, divulgada, distribuida, transmitida, almacenada en un sistema de recuperación o traducida a cualquier lenguaje del ser humano o de programación, en cualquier forma, por cualesquiera medios, por entero o en parte, sin el consentimiento previo por escrito de FP.

Algunos productos o compañías que se mencionan son marcas de sus respectivos propietarios.

ENDALIA, S.L. • Plaza Roma F-1 7ºE 50010, Zaragoza • España




 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	5
1.1 PROPÓSITO DEL DOCUMENTO	5
1.2 ALCANCE DEL DOCUMENTO	5
1.3 ACRÓNIMOS	5
1.4 DEFINICIONES	5
1.5 REFERENCIAS	5
1.6 RESUMEN	5
2. INICIO	7
2.1 PANTALLA DE INICIO	7
3. CONFIGURACIÓN INICIAL	9
3.1 CONFIGURACIÓN	9
4. CONTROLES GENÉRICOS DE LA APLICACIÓN	11
4.1 BARRA DE HERRAMIENTAS	11
4.2 USO DE TABLAS	11
5. PROYECTOS	16
5.1 PANTALLA PRINCIPAL	16
5.1.1 DEFINICIÓN	15
5.1.2 PLANIFICACIÓN	20
5.1.3 RECURSOS	20
5.1.4 DOCUMENTOS	20
5.1.5 INDICADORES ECONÓMICOS	21
5.2 VISTAS DERECHAS (INFORMES HTML)	21
5.2.1 ACCIONES	21
5.2.2 DEDICACIÓN	21
5.2.3 PRESUPUESTO	22
5.2.4 INFORME	23
6. PERSONAS	24
6.1 PANTALLA PRINCIPAL	25
6.2 FICHA DEL EMPLEADO	28

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

6.2.1	DATOS PERSONALES	28
6.2.2	FICHEROS	29
7.	USUARIOS	30
7.1	PANTALLA PRINCIPAL	30
8.	GENERACIÓN DE INFORMES	33
8.1	PANTALLA PRINCIPAL	33
8.1.1	PROYECTOS	34
8.1.2	PERSONAS	34
8.1.3	DATOS A INCLUIR	35
8.1.4	GENERAR INFORME	36
9.	BIBLIOGRAFÍA	37
9.1	REFERENCIAS	37
9.2	REFERENCIAS WEB	37

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

1. INTRODUCCIÓN

1.1 PROPÓSITO DEL DOCUMENTO

La filosofía de Endalia para la documentación de ayuda al usuario, se basa en la elaboración de guías rápidas a modo de capturas de pantalla y breves descripciones de la funcionalidad más relevante de dicha pantalla. Por tanto, este documento no pretende ser una descripción exhaustiva de todas las funcionalidades del SGP, sino una ayuda para que un usuario que comienza a utilizar la aplicación disponga de una introducción rápida y sencilla a las funcionalidades que ésta proporciona.

1.2 ALCANCE DEL DOCUMENTO

Este documento pretende ayudar al usuario final de la aplicación. Por tanto, su alcance comprende la fase final del proyecto, una vez que el producto sea distribuido.

1.3 ACRÓNIMOS

- SGP: Sistema de Gestión de la Organización y de Recursos Humanos.

1.4 DEFINICIONES

No se considera necesario definir ningún concepto de los manejados a lo largo de este documento.

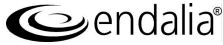
1.5 REFERENCIAS

En este documento no se han realizado referencias a otros documentos del proyecto.


1.6 RESUMEN

El presente documento es una guía básica de ayuda para los usuarios de la aplicación del SGP. Está compuesto de 9 apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describe la pantalla inicial de la aplicación.

 endalia®	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

- Apartado 3: Se describen aspectos relacionados con la configuración inicial del sistema.
- Apartado 4: Se describen los controles genéricos de la aplicación (barra de herramientas y tablas), ilustrando las posibilidades que ofrecen.
- Apartado 5: Este apartado se centra en la descripción del funcionamiento de la aplicación.
- Apartado 6: Este apartado se centra en la descripción del funcionamiento del módulo de personas
- Apartado 7: Este apartado se centra en la descripción del funcionamiento del módulo de usuarios.
- Apartado 8: Este apartado se centra en la descripción de la generación de informes.
- Apartado 9: Bibliografía.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

2. INICIO

2.1 PANTALLA DE INICIO

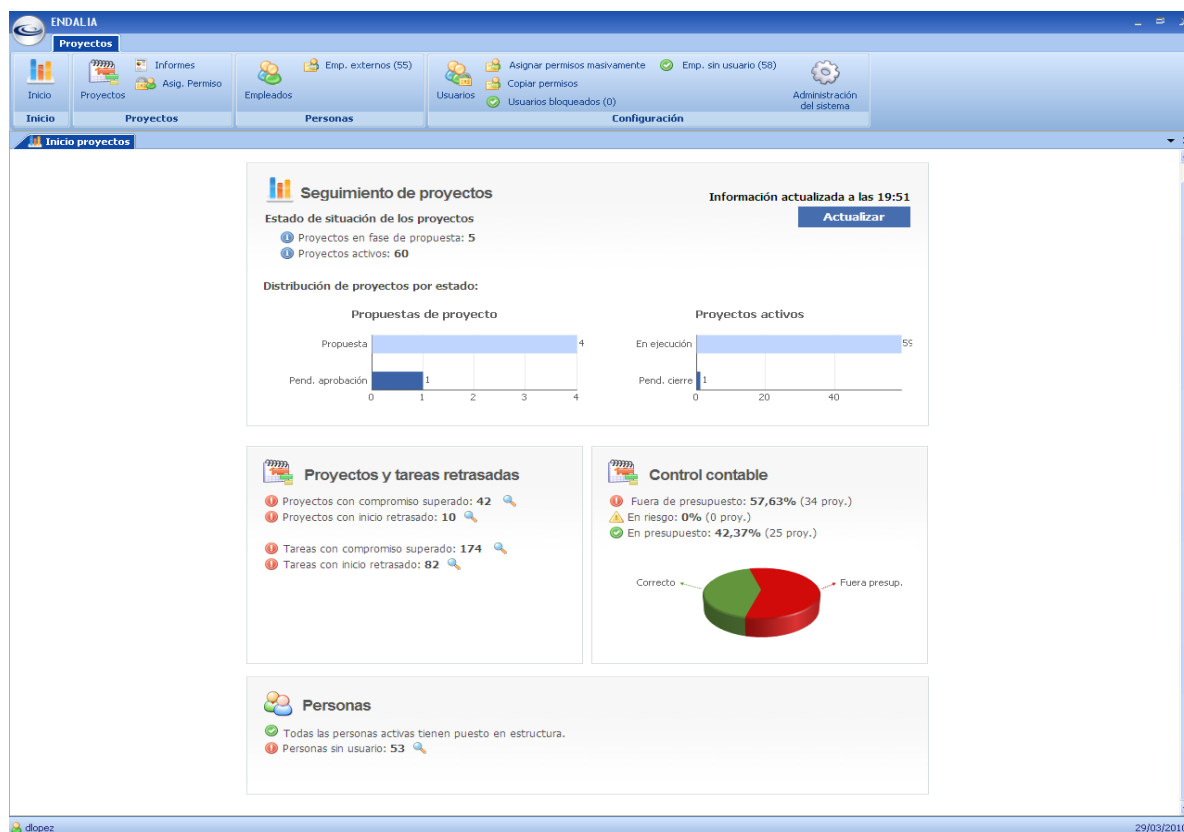


Figura 1: Página de inicio (asistente)

En esta página inicial del programa (Figura 1), se informa al usuario de la aplicación sobre el estado de situación de las proyectos, tareas y personas que hay registradas en el sistema.


- **Proyectos:**

Informa de los proyectos que hay en las distintas fases de vida de los mismos.

Informa de la situación económica de los proyectos, mediante un gráfico circular que refleja la situación económica de los proyectos en ejecución. Además informa de los proyectos y tareas que cuya fecha de compromiso o inicio se ha superado, para completar los datos más fácilmente, el sistema permite acceder a la ficha del proyecto o tarea directamente desde esta página de inicio.

- **Personas:**

Informa acerca de las personas registradas en el sistema sin usuario definido y pendientes de consolidar.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Para acceder a la ficha de un proyecto ó tarea que no cumple con la planificación, o para acceder a la ficha de un persona sin consolidar o sin usuario, tan solo hay que clicar en la lupa junto al texto y seleccionar la/s fichas que queremos abrir de la lista de proyectos/tareas o personas. (Figura 2)

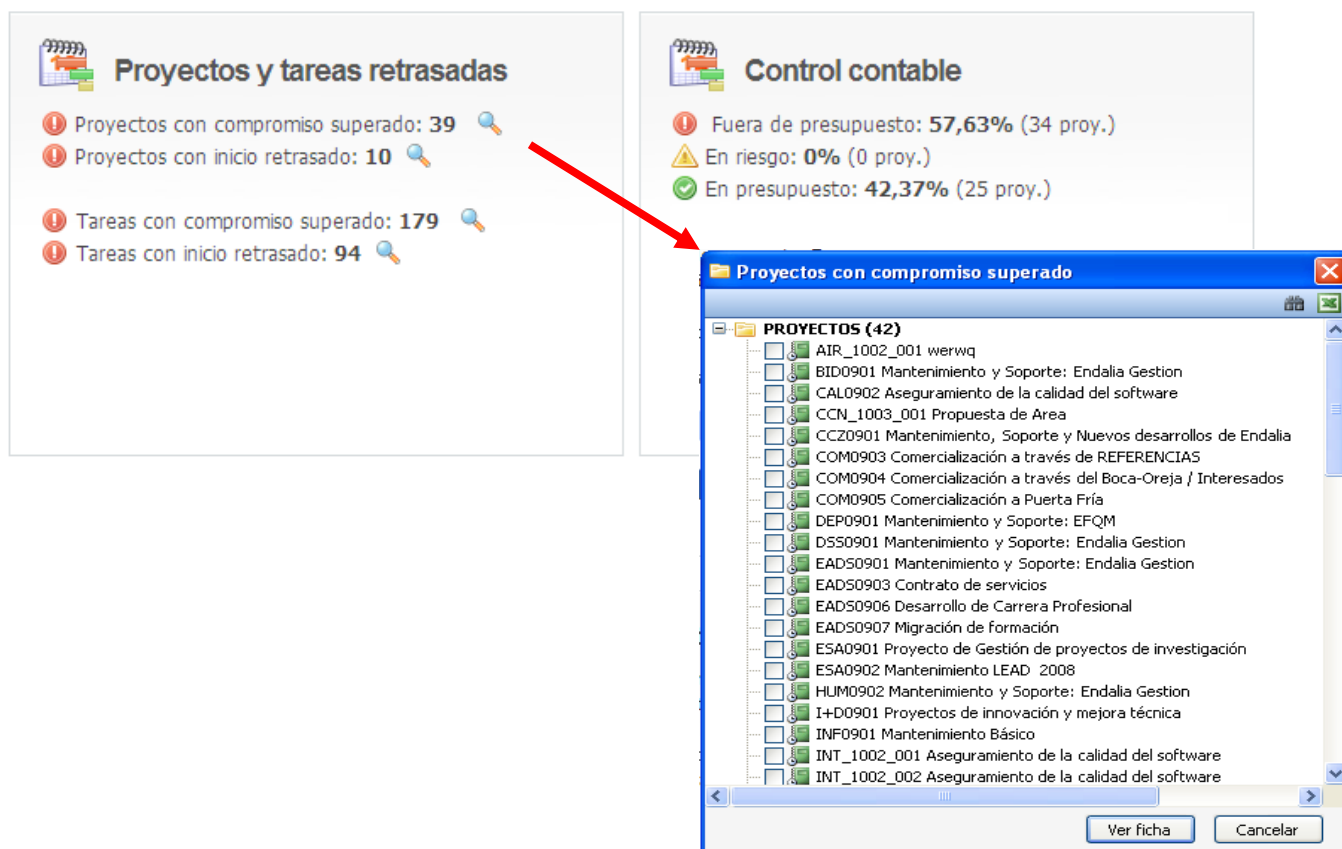
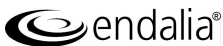


Figura 2: Detalle de página de inicio

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

3. CONFIGURACIÓN INICIAL

3.1 CONFIGURACIÓN


El sistema permite la parametrización de distintos conceptos para personalizar el uso de la aplicación (Figura 3).

Estos conceptos son:

- Tipo de proyecto
- Partida presupuestaria
- Convocatoria
- Alcance
- Notificación
- Participación
- Área
- Indicadores económicos
- Indicadores no económicos
- Centro de coste
- Centro de trabajo
- Lenguaje

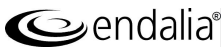


Figura 3: Configuración del sistema

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Para entrar en la configuración de la aplicación, lo haremos haciendo clic en el botón “*Configurar*” de la barra superior de herramientas.

- Ya en el apartado de administración, podremos seleccionar que tipo de parámetro queremos añadir o eliminar. Seleccionamos uno y veremos desplegados los diferentes ítems ya insertados.
- A partir de aquí, podemos “*Añadir*” un elemento, o seleccionarlo y presionar el botón “*Eliminar*” para eliminarlo. Si el parámetro que pretendemos eliminar está siendo utilizado, no se permitirá su eliminación.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

4. CONTROLES GENÉRICOS DE LA APLICACIÓN

4.1 BARRA DE HERRAMIENTAS



Figura 4: Barra de herramientas

Desde la barra de herramientas (Figura 4) podremos acceder a los diversos módulos de la aplicación:

- **Inicio:** Ventana inicial de la aplicación con resumen de información
- **Proyectos:** Módulo para la gestión de proyectos, que además contiene un submódulo para la gestión de informes.
- **Personas:** Módulo para la gestión de personas.
- **Configuración:** Módulo para la administración de las características del sistema y gestión de usuarios de la aplicación.

IMPORTANTE: Si desea agrandar el espacio de trabajo de la aplicación, puede ocultar esta barra de herramientas, haciendo doble clic sobre la pestaña “Navegación”. Para volver a mostrar la barra de herramientas, basta con hacer doble clic de nuevo en la misma pestaña.

4.2 USO DE TABLAS

Las tablas que aparecen en la aplicación son configurables de manera que podemos ordenar, fijar columnas o filas para permitir una comparación entre campos. En el caso de los informes, además de esto, también es posible filtrar los resultados mostrados.

- **Ordenación por campos:** Para ordenar por un campo determinado (columnas), haga clic en el campo que quiera para que aparezca el símbolo de ordenación (caracterizado por una flecha, que estará orientada arriba o abajo según esté ordenado de forma ascendente o descendente respectivamente. Ejemplo (Figura 5):



Figura 5: Ordenación por campos

- **Selección de campos a visualizar:** En algunas de las tablas se permite seleccionar qué campos se desea mostrar en las columnas. Para ello se debe pulsar el botón de la cabecera que se encuentra más a la izquierda (Figura 6) y seleccionar / deseleccionar en la pantalla que aparece los campos a visualizar.

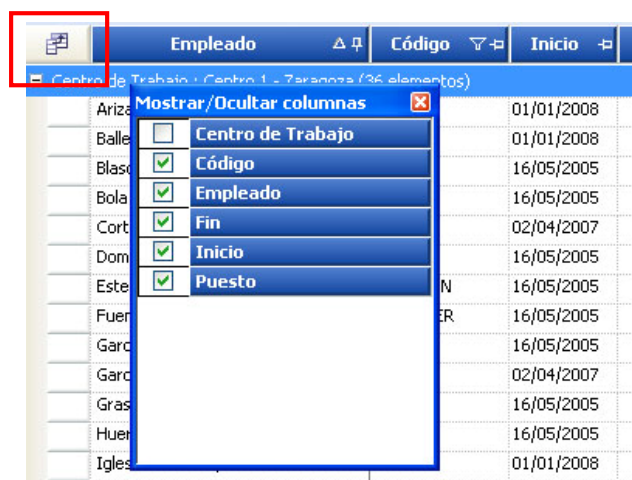


Figura 6: selector de campos

- **Fijado de columnas y filas:** En las columnas y filas, existe un símbolo de chincheta. Ésta por defecto está sin pinchar, de forma que el movimiento por la tabla es libre, pero si queremos dejar una fila o una columna fija sólo tendremos que hacer clic sobre ésta y la fila o la columna seleccionada quedará bloqueada en la parte superior de la tabla si es una fila o en el lateral izquierdo si se trata de una columna.

Ejemplo:







Empleado	Código	Email Prof.	Centro de Trabajo	Segurid
 Fuentes Alba, Miguel	 002	 mfuente@endalia.com	 Centro 1 - Zaragoza	 54587685

Figura 7: Fijado de columnas y filas

En este ejemplo (Figura 7) hemos bloqueado la fila correspondiente al empleado Miguel Fuentes Alba, de manera que aunque movamos los resultados con la barra de desplazamiento (izquierda) la fila correspondiente no se moverá, de forma que es más fácil comparar una fila con otra.

Este funcionamiento es exactamente igual para tratar con columnas, únicamente se diferencia en que en este caso, se quedarían fijadas a la izquierda de la tabla.

- **Agrupación:** Es posible agrupar los resultados de cualquier tabla por determinados campos (esta agrupación únicamente se podrá realizar en algunos determinados campos). Para esto, sólo tendremos que arrastrar el campo por el cual vamos a agrupar a la zona de agrupación (Figura 8).

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Arrastra aquí un encabezado para agrupar por esa columna

Figura 8: Zona de agrupación

Ejemplo:

Responsable ▾						
Código ▾	Nombre ▾	Fin Prev. ▾	Compromiso ▾	Progreso (%) ▾	Dedicación ▾	Presupuesto ▾
+ Responsable : (1 elemento)						
+ Responsable :	Cimorra Velilla, José Carlos (1 elemento)					
+ Responsable :	Cortés Franco, Fernando (10 elementos)					
+ Responsable :	Giménez Jiménez, Pablo (2 elementos)					
+ Responsable :	Gomez Ramo, Isabel (1 elemento)					
+ Responsable :	Guillén Martínez, José Luis (5 elementos)					
+ Responsable :	López Rodríguez, David (24 elementos)					
+ Responsable :	Mayayo Ortigosa, Adrián (1 elemento)					
+ Responsable :	Minguez Barroso, Cristina (1 elemento)					
+ Responsable :	Peciña Galar, Cristina (1 elemento)					
+ Responsable :	Romay Carracedo, Alfonso (9 elementos)					
+ Responsable :	Solera Manzano, Javier (1 elemento)					
+ Responsable :	Uralde Aznar, Laura (1 elemento)					
+ Responsable :	Wroblewski, Grzegorz (1 elemento)					

Figura 9: Agrupación por Responsable

- **Selección múltiple:** Podemos seleccionar varias filas de dos formas diferentes:
 - La primera es utilizar el ratón para hacer clic en la zona izquierda de la fila y arrastrar hacia arriba o abajo con el botón del ratón pulsado.
 - La segunda manera requiere del uso del teclado. Como en casi cualquier aplicación Windows, pulsando la tecla “Control” y manteniéndola podremos ir seleccionando los elementos requeridos. De forma análoga, pulsando “Shift”, podremos marcar una selección continua, seleccionado desde el primer elemento hasta el último (un clic al principio y otro al final).
- **Filtrado:** En los informes, además de lo anteriormente citado, también es posible filtrar los resultados por uno o varios campos. Para realizar esto, buscaremos la línea de filtrado (Figura 10), está situada entre los nombres de los campos (Empleado, Código, Email Prof, etc.) y los datos de la tabla y tiene un color amarillo para una mejor localización y no confundirlo con datos.







Empleado ▾	Código ▴ ▾	Email Prof. ▾	Centro de Trab. ▾
			

Figura 10: Línea de filtrado

Aquí podemos ver los elementos disponibles para el filtrado (no incluimos numeración en las siguientes figuras de este punto por mejorar la legibilidad):

 Uno que tacha un embudo a la izquierda borrará todos los filtros que tengamos creados en el momento.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	



Este es el lugar donde escribiremos el filtro, o elegiremos de la lista desplegable.




En este desplegable, aparecerá tanto cada uno de los distintos elementos que son mostrados en la columna como elementos genéricos referidos a “Todos”, “Vacíos” y “No vacíos”.



Este icono determinará que comparación se hace entre los resultados y el campo de filtro propiamente dicho. Las opciones disponibles son:

- Es igual a
- Distinto a
- Es menor que
- Es menor o igual que
- Es mayor que
- Es mayor o igual que
- Es como
- Coincide con expresión regular
- Comienza con
- Contiene
- Termina en
- No empieza en
- No contiene
- No termina en
- No coincide con
- No es como

Según la opción elegida, la comparación será de una manera u otra. Por ejemplo, “*Es igual a*” comparará letra a letra y solo mostrará campos exactamente iguales al filtro, sin embargo, “*Contiene*”, mostrará los elementos de la tabla que contengan (en la columna filtrada) en cualquier posición del campo el filtro fijado.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	



Cuando coloque el cursor sobre un campo de filtro, aparecerá este icono con el que podrá eliminar el filtro establecido para esa columna en concreto.

- **Doble clic:** Haciendo doble clic sobre las filas de las tablas, la aplicación mostrará la ficha correspondiente al elemento seleccionado. Esta opción aparecerá en las tablas en las que aparezca información relacionada con el empleado.
- **Exportación a Excel:** En todas las tablas aparecerá un botón en la esquina superior derecha que permitirá la exportación de su contenido a Excel (Microsoft Office). Pulsando este botón el sistema mostrará una ventana en la que se solicitará la ruta de su equipo en el que desea guardar el archivo que se generará para Excel (extensión.xls) y en la que podrá definir un nombre para ese archivo. Una vez guardado en el equipo el archivo, podremos ejecutarlo para ver su contenido.

5. PROYECTOS

5.1 PANTALLA PRINCIPAL

A este módulo se accede mediante el botón de “Proyectos” colocado en la barra superior de herramientas, accediendo al listado de proyectos activos de la organización (Figura 11), donde se nos permitirá ver, editar e imprimir su información.

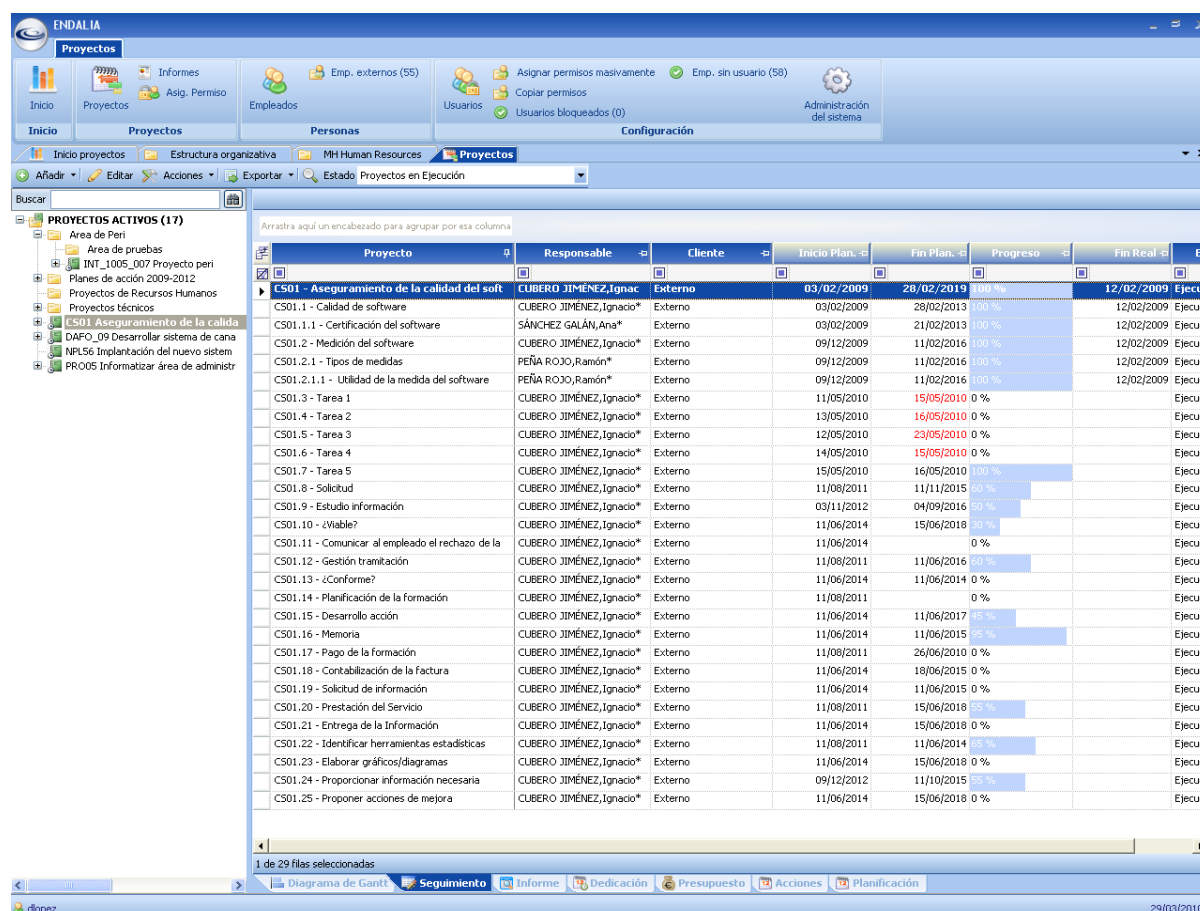



Figura 11: Proyectos en estado activo

En la sección izquierda tendremos una estructura de carpetas en las que se agrupan los proyectos, ordenados por área.

A la derecha se muestran los elementos dependientes del ítem seleccionado en el árbol. Esto es, por defecto está seleccionado en la estructura “Proyectos activos” con lo que aparecen todos los proyectos activos, pero por ejemplo, si seleccionamos un área, solo aparecerán los proyectos activos que pertenezcan a esa área. También están a nuestra disposición cinco vistas (opción “Estado” en la parte superior) diferentes de esta pantalla derecha. Éstas son “Propuesta de proyecto”, “Proyectos activos”, “Proyectos pendientes de cierre”, “Proyectos históricos” y “Proyectos cancelados”.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

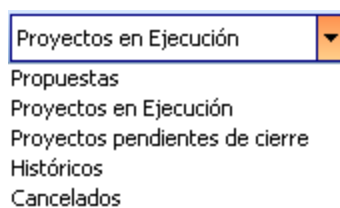


Figura 12: Vistas en listado

Podemos añadir un proyecto, área o tarea haciendo clic en el botón “Añadir”, y seleccionado la opción dentro del menú desplegable, la distintas opciones de añadir dependen del ítem seleccionado en la estructura, así con un área seleccionada se podrá añadir un área o proyecto, mientras que con un proyecto seleccionado solo se podrán añadir tareas.

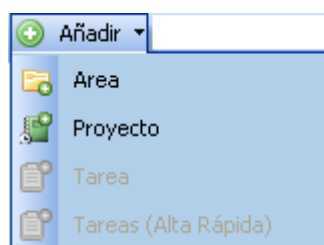


Figura 13: Ventana para añadir áreas, proyectos o tareas

Haciendo doble clic en un área de la estructura accedemos a su ficha.

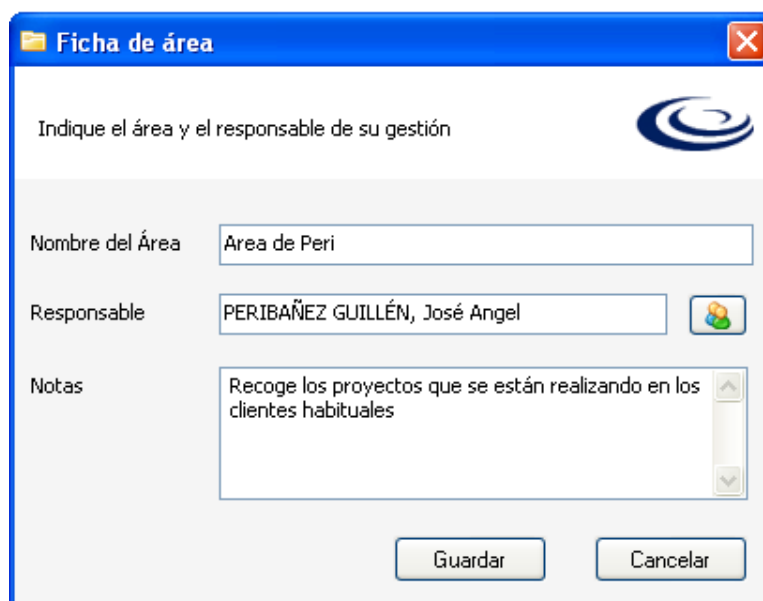



Figura 14: Ficha de áreas

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Dependiendo de la selección se abrirá una ficha donde rellenar los datos para dar de alta el nuevo área, proyecto o tarea. Seleccionado un proyecto aparecerá la siguiente ficha. (Figura 15).

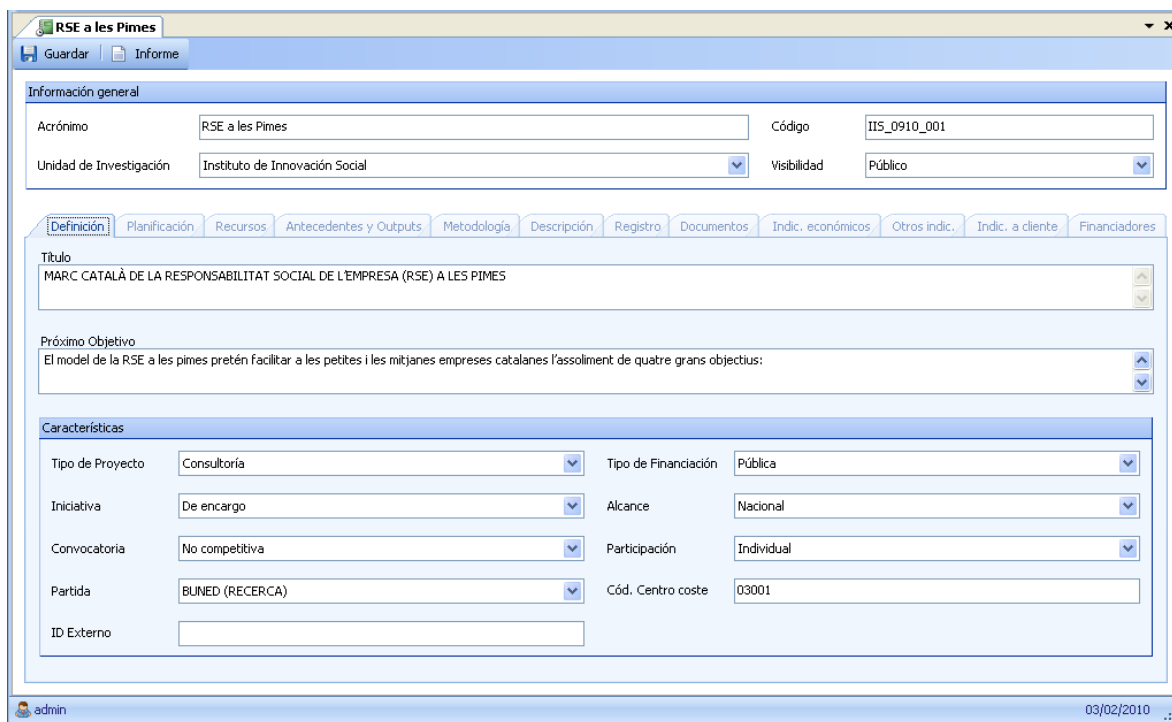


Figura 15: Ficha de proyecto

Además, al seleccionar un ítem en el árbol, se habilitarán las distintas opciones el menú “Acciones” en la zona superior (Figura 16).

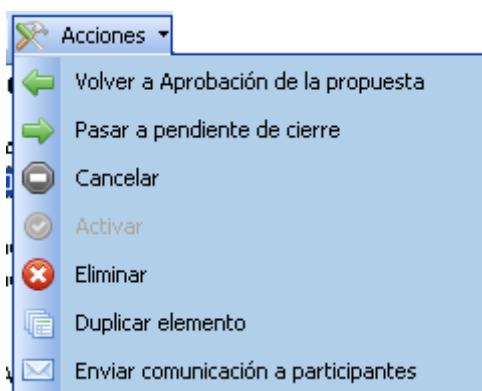



Figura 16: Menú Acciones

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Desde el menú de acciones independientemente de la vista en la que nos encontremos, podremos:

- Cambiar de estado un proyecto.
- Cancelar o activar un proyecto o tarea.
- Eliminar un área, proyecto o tarea.
- Duplicar un proyecto o tarea con sus dependientes.
- Enviar un correo a los participantes en proyectos.

Haciendo doble clic en un proyecto de la estructura o seleccionando un proyecto y clicando la opción 'editar' en la barra de herramienta accedemos a su ficha, similar a la ficha que es necesario rellenar para dar de alta un nuevo proyecto.

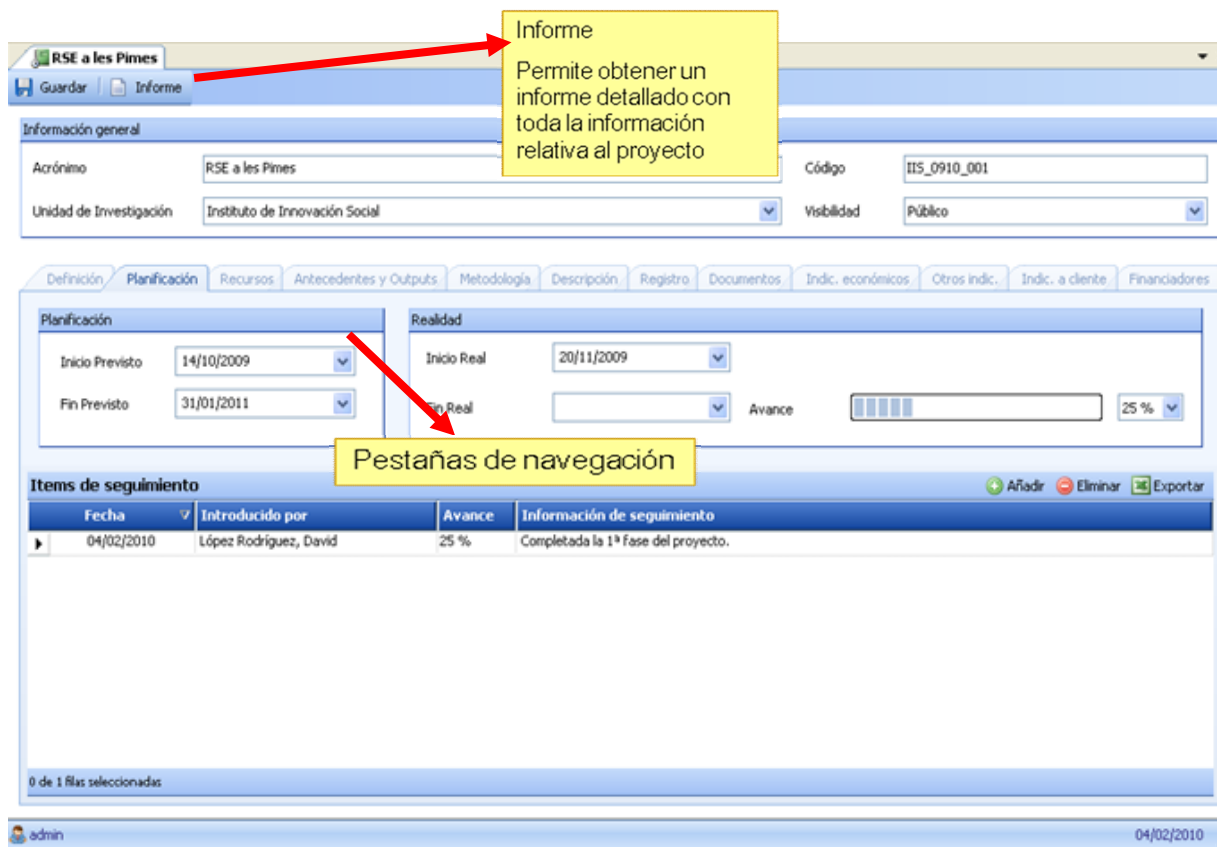



Figura 17: Ficha de un proyecto

En ésta, podremos editar datos de varios ámbitos referentes al proyecto, véase, definición, planificación, características, antecedentes y salidas, documentos, indicadores económicos.

A continuación se detallan los datos almacenados en las principales pestañas de la ficha.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

5.1.1 DEFINICIÓN

En la pestaña “Definición”, se almacena la siguiente información:

- **Características:**
 - Tipo de proyecto
 - Partida presupuestaria
 - Convocatoria
 - Alcance
 - Notificación
 - Área
- **Objetivos**
- **Título largo**

5.1.2 PLANIFICACIÓN

En la pestaña “Planificación”, se almacena la siguiente información:

- **Planificación:**
 - Fecha planificada de inicio
 - Fecha planificada de fin
 - Fecha real de inicio
 - Fecha real de fin
 - Progreso
- **Items de seguimiento**

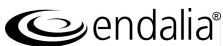
5.1.3 RECURSOS

En la pestaña “Recursos”, se almacena la siguiente información:

- **Recursos necesarios**
- **Proveedores**
- **Participantes en el proyecto**

5.1.4 DOCUMENTOS

En la pestaña “Documentos”, se almacenas todos los documentos adjuntos al proyecto durante su desarrollo.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

5.1.5 INDICADORES ECONÓMICOS

En la pestaña “Indicadores Económicos”, se almacenan todos los indicadores económicos de ingresos que gastos que se configuran el presupuesto del proyecto.

Indicadores económicos					
Tipo de indicador 					
Indicador	Actual (acumulado)	Objetivo final (acum.)	Fecha obj. final	Descripción	
[-] Tipo de indicador : Gastos genéricos (2 elementos)					
<input type="checkbox"/> Gastos de personal	244,43	4800	31/12/2009	Actualizado: 29/10/2009 23:00:05	
<input type="checkbox"/> Gestor de viaje	0	200	31/12/2009		
[-] Tipo de indicador : Ingresos genéricos (2 elementos)					
<input type="checkbox"/> Ingreso mantenimiento anual	4833	4833	31/12/2009		
<input type="checkbox"/> Ingreso nuevos desarrollos	0	0	31/12/2009		

Figura 18: Pestaña “Indicadores económicos” de la ficha de proyecto

5.2 VISTAS DERECHAS (INFORMES HTML Y GANTT)

En la pantalla derecha de la aplicación además del seguimiento de proyectos, se muestran unos informes html que en forma de resumen muestran información más destacada de áreas, proyectos y tareas:



Figura 19: Vistas en pantalla derecha

5.2.1 ACCIONES

Este informe html muestra el número de acciones invertidas en un área, proyecto o tarea, con un desglose por persona.

5.2.2 DEDICACIÓN

Este informe HTML muestra las horas planificadas e invertidas en un área, proyecto o tarea, con un grafica de evolución durante el último año y un desglose por persona que participa en el proyecto.



Figura 20: Informe de dedicación

5.2.3 PRESUPUESTO

Este informe HTML muestra un resumen económico de la evolución de los indicadores asociados al proyecto o a los proyectos de un área, con un desglose por indicador y una gráfica de evolución del margen de beneficio real y planeado.

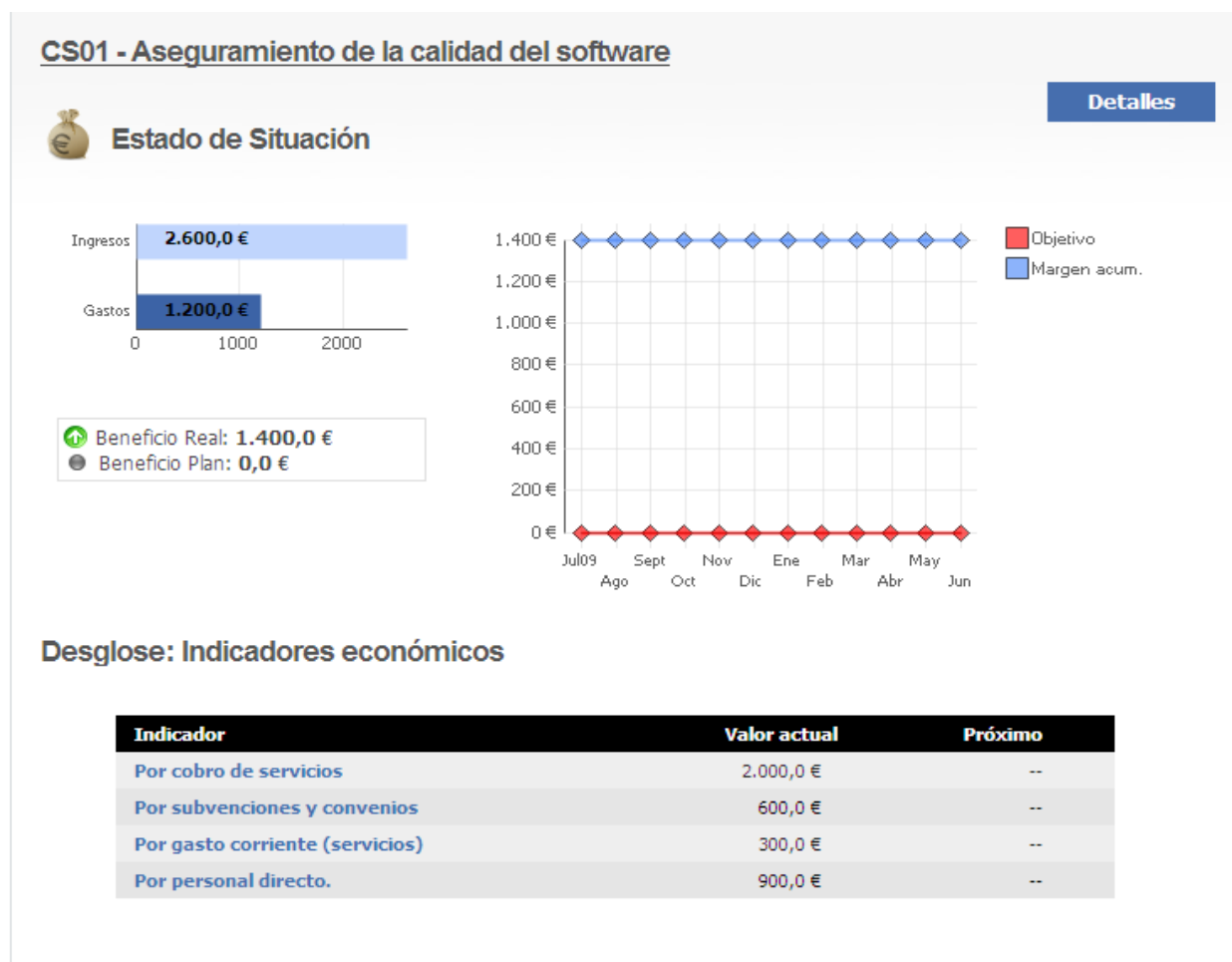


Figura 21: Informe de presupuesto

5.2.4 INFORME

Este informe html muestra un resumen de la principal información de proyecto, para consultar con un simple vistazo.

5.2.5 DIAGRAMA DE GANTT

Seleccionando un proyecto en la estructura de la izquierda, nos aparece una nueva vista derecha junto a los informes html, para visualizar su diagrama de Gantt (Figura 22):

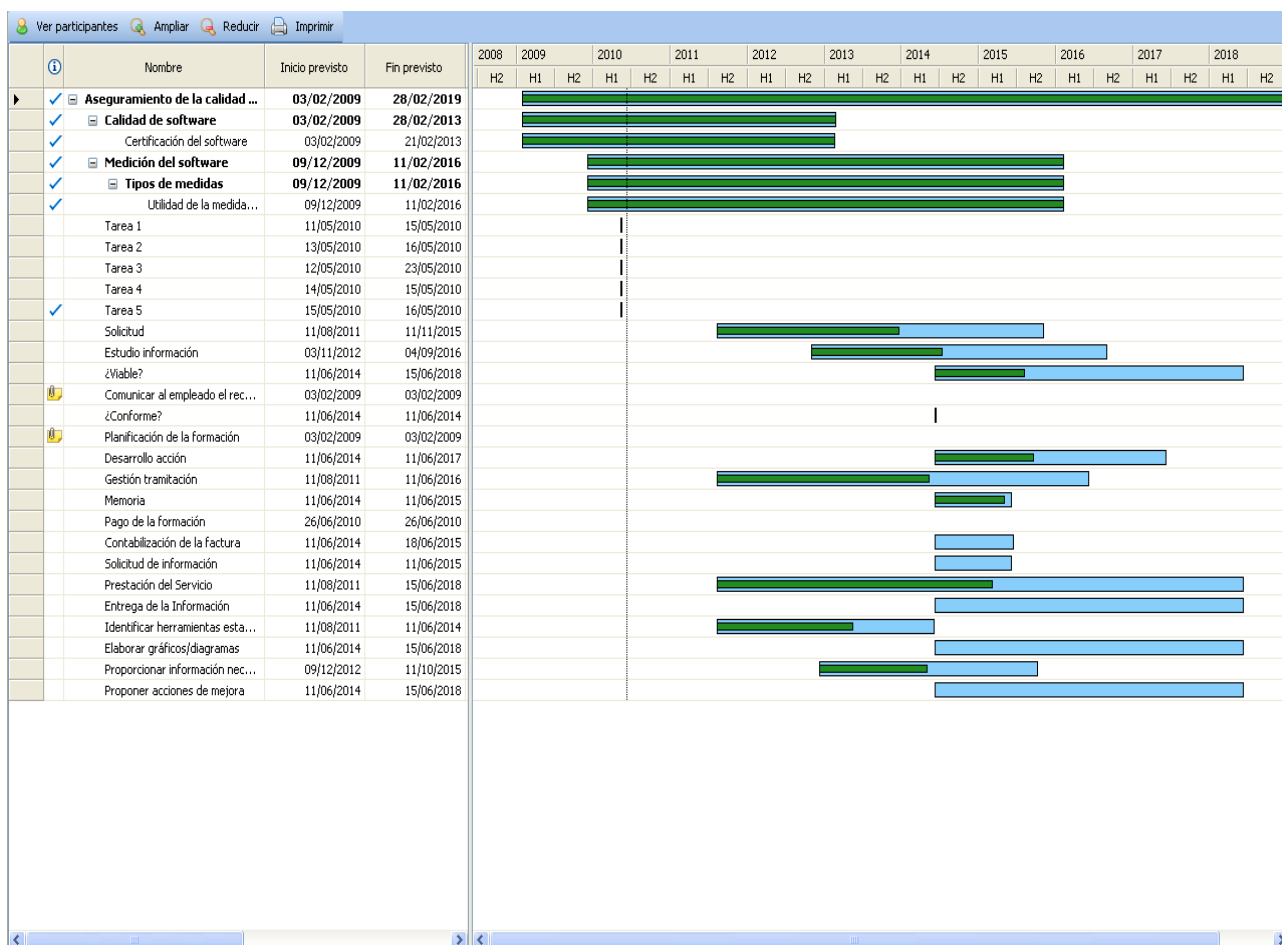
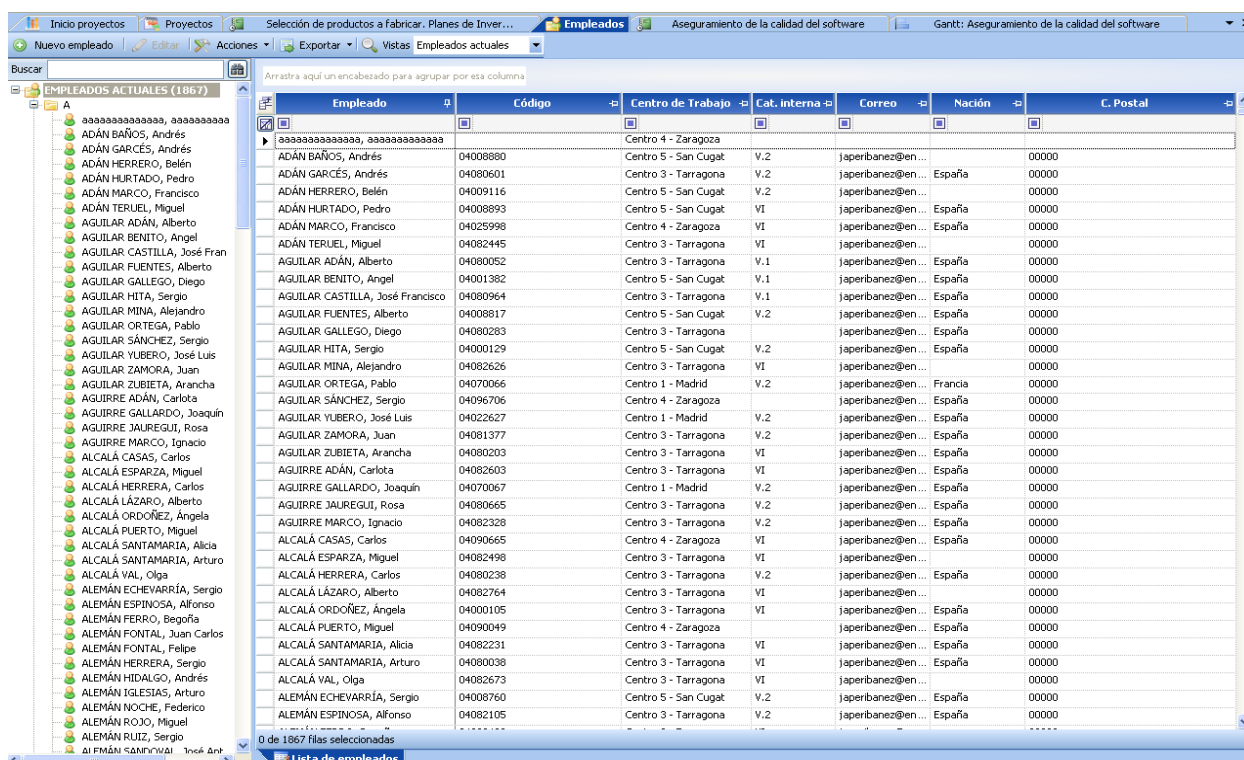


Figura 22: Diagrama de Gantt

6. PERSONAS

6.1 PANTALLA PRINCIPAL

A este módulo se accede mediante el botón de “Personas” colocado en la barra superior de herramientas, accediendo al listado de empleados actuales de la organización (Figura 23), donde se nos permitirá ver, editar e imprimir su información.



Empleado	Código	Centro de Trabajo	Cat. interna	Correo	Nación	C. Postal
ADÁN BAÑOS, Andrés	04008880	Centro 4 - Zaragoza		japeribanez@en...		00000
ADÁN GARCÉS, Andrés	04080601	Centro 3 - Tarragona	V.2	japeribanez@en...	España	00000
ADÁN HERRERO, Belén	04009116	Centro 5 - San Cugat	V.2	japeribanez@en...		00000
ADÁN HURTADO, Pedro	04008893	Centro 5 - San Cugat	VI	japeribanez@en...	España	00000
ADÁN MARCO, Francisco	04025998	Centro 4 - Zaragoza	VI	japeribanez@en...	España	00000
AGUILAR ADÁN, Alberto	04082445	Centro 3 - Tarragona	VI	japeribanez@en...		00000
AGUILAR BENITO, Angel	04080052	Centro 3 - Tarragona	V.1	japeribanez@en...	España	00000
AGUILAR CASTILLA, José Fran	04001382	Centro 5 - San Cugat	V.1	japeribanez@en...	España	00000
AGUILAR FUENTES, Alberto	04080964	Centro 3 - Tarragona	V.1	japeribanez@en...	España	00000
AGUILAR GALLEGU, Diego	04008817	Centro 5 - San Cugat	V.2	japeribanez@en...	España	00000
AGUILAR HITA, Sergio	04080283	Centro 3 - Tarragona		japeribanez@en...	España	00000
AGUILAR MINA, Alejandro	04000129	Centro 5 - San Cugat	V.2	japeribanez@en...	España	00000
AGUILAR ORTEGA, Pablo	04082626	Centro 3 - Tarragona	VI	japeribanez@en...		00000
AGUILAR YUBERO, José Luis	04070066	Centro 1 - Madrid	V.2	japeribanez@en...	Francia	00000
AGUIRRE ADÁN, Arancha	04096706	Centro 4 - Zaragoza		japeribanez@en...	España	00000
AGUIRRE GALLARDO, Joaquín	04022627	Centro 1 - Madrid	V.2	japeribanez@en...	España	00000
AGUIRRE JAUREGUI, Rosa	04081377	Centro 3 - Tarragona	V.2	japeribanez@en...	España	00000
ALCALÁ CASAS, Carlos	04080203	Centro 3 - Tarragona	VI	japeribanez@en...	España	00000
ALCALÁ ESPARZA, Miguel	04082603	Centro 3 - Tarragona	VI	japeribanez@en...	España	00000
ALCALÁ HERRERA, Carlos	04070067	Centro 1 - Madrid	V.2	japeribanez@en...	España	00000
ALCALÁ LÁZARO, Alberto	04080665	Centro 3 - Tarragona	V.2	japeribanez@en...	España	00000
ALCALÁ ORDOÑEZ, Ángela	04082328	Centro 3 - Tarragona	V.2	japeribanez@en...	España	00000
ALCALÁ PUERTO, Miguel	04090665	Centro 4 - Zaragoza	VI	japeribanez@en...	España	00000
ALCALÁ SANTAMARÍA, Alicia	04082498	Centro 3 - Tarragona	VI	japeribanez@en...	España	00000
ALCALÁ VAL, Olga	04080238	Centro 3 - Tarragona	V.2	japeribanez@en...	España	00000
ALEMÁN ECHEVARRÍA, Sergio	04082764	Centro 3 - Tarragona	VI	japeribanez@en...		00000
ALEMÁN ESPINOSA, Alfonso	04000105	Centro 3 - Tarragona	VI	japeribanez@en...	España	00000
ALEMÁN FERRER, Begoña	04090049	Centro 4 - Zaragoza		japeribanez@en...	España	00000
ALEMÁN FONTAL, Juan Carlos	04082231	Centro 3 - Tarragona	VI	japeribanez@en...	España	00000
ALEMÁN HERRERA, Sergio	04080038	Centro 3 - Tarragona	VI	japeribanez@en...	España	00000
ALEMÁN HIDALGO, Andrés	04082673	Centro 3 - Tarragona	VI	japeribanez@en...		00000
ALEMÁN IGLESIAS, Arturo	04008760	Centro 5 - San Cugat	V.2	japeribanez@en...	España	00000
ALEMÁN NOCHE, Federico	04082105	Centro 3 - Tarragona	V.2	japeribanez@en...	España	00000
ALEMÁN ROJO, Miguel						
ALEMÁN RUIZ, Sergio						
ALEMÁN SANTAMARÍA, José Ant						

Figura 23: Empleados actuales

En la sección izquierda tendremos una estructura de carpetas en las que se agrupan los empleados, ordenados por apellido.

A la derecha se muestran los elementos dependientes del ítem seleccionado en el árbol. Esto es, por defecto está seleccionado en la estructura “Empleados actuales” con lo que aparecen todos los empleados actuales, pero por ejemplo, si seleccionamos la carpeta “G”, solo aparecerán los empleados actuales cuyo apellido empiece por “G”. También están a nuestra disposición dos vistas (opción “Vistas” en la parte superior) diferentes de esta pantalla derecha. Éstas son “Empleados actuales” y “Empleados históricos” (Figura 24) y muestran diferentes datos de cada empleado.

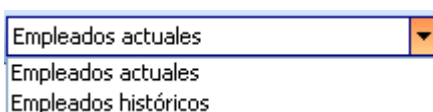



Figura 24: Vistas

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Podemos añadir un empleado activo haciendo clic en el botón “*Añadir persona*”, aparecerá una ventana (Figura 25) para rellenar tres datos mínimos en el alta de un nuevo empleado (Nombre, Apellidos y E-mail).

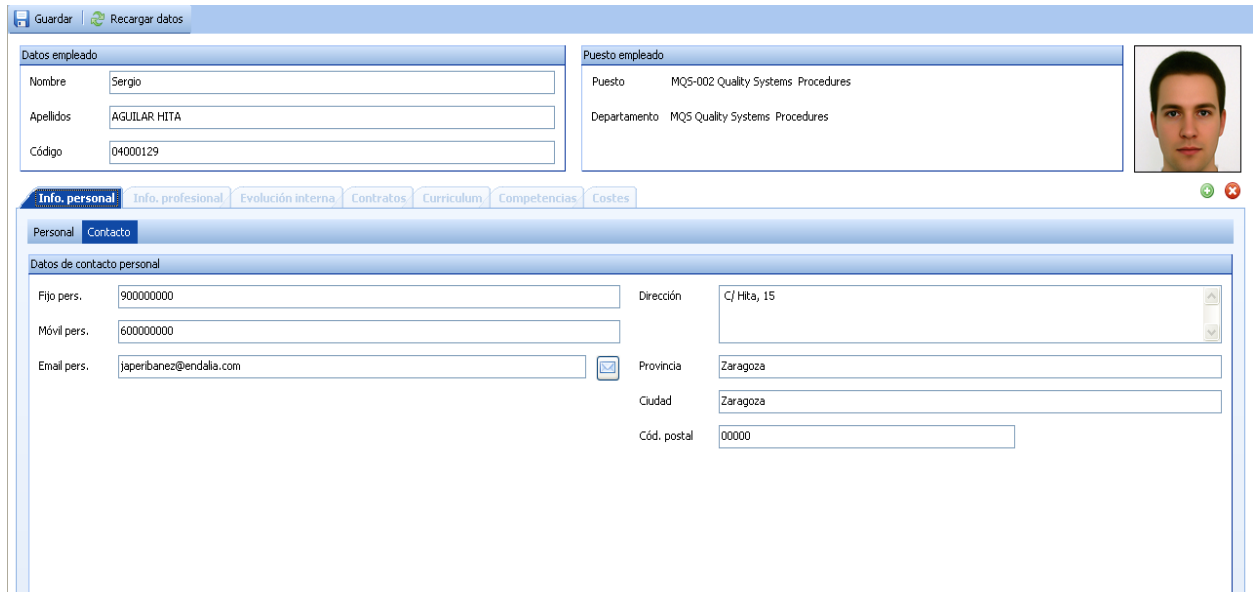



Figura 25: Ventana para añadir personas

Una vez rellenos estos datos, podemos guardar directamente el empleado y seguir con otras tareas, o abrir directamente su ficha y rellenar toda su información.

Seleccionando un empleado en el árbol, aparecerá su ficha en formato HTML en la parte derecha (Figura 26).

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	



The screenshot shows a web application interface for managing employees. On the left is a tree view titled 'Empleados actuales (44)' with folders A through Z. Folder 'B' is expanded, showing three employees: 'Ariza Aguilar, Daniel*', 'Ballester Belmonte, Ariadna' (selected), and 'Blasco Gutiérrez, David'. The main area displays the profile of 'Ballester Belmonte, Ariadna'. The profile includes a header with the name, a photo, and three sections: 'Información general', 'Contacto profesional', and 'Contacto personal'. The 'Información general' section contains fields for Name, Surnames, Code, Work Center, DNI, and Birth Date. The 'Contacto profesional' section contains fields for Phone, Extension, Mobile, Fax, and Email. The 'Contacto personal' section contains fields for Phone, Mobile, and Email. The 'Datos profesionales' section contains a field for Internal Category.

Información general	
Nombre:	Ariadna
Apellidos:	Ballester Belmonte
Código:	022
Centro de trabajo:	Centro 1
DNI:	XD362738237F
Fecha de nacimiento:	05/08/1975

Contacto profesional	
Teléfono:	9300000000
Extensión:	001
Móvil:	
Fax:	
Email:	aballester0@miempresa.com.tk

Contacto personal:	
Teléfono:	910000000
Móvil:	555555555
Email:	aballester0@freemail.tk

Datos profesionales	
Categoría interna:	Titulado superior

Figura 26: Ficha de empleado

Además, al seleccionar un empleado en el árbol, se habilitará el menú “Acciones” en la zona superior (Figura 27).

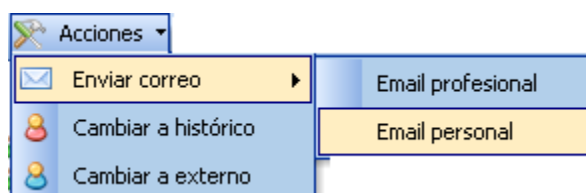



Figura 27: Menú Acciones

Desde el menú de acciones de la estructura de “Empleados actuales”, podremos:

- Enviar un correo electrónico a su dirección personal o profesional (siempre que la tenga configurada).
- Cambiar el estado del empleado activo a histórico (por lo que desaparecerá el empleado de la estructura que estamos visualizando y su ficha se transfiere al listado de empleados históricos).

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

En caso de que nos encontráramos visualizando la estructura de “*Empleados históricos*”, el funcionamiento es análogo al de “*Empleados actuales*”, pero además podremos realizar las siguientes acciones:

- Cambiar el estado del empleado histórico a activo (por lo que desaparecerá el empleado de la estructura que estamos visualizando y su ficha se transfiere de nuevo al listado de empleados actuales).

6.2 FICHA DEL EMPLEADO

Al hacer doble clic sobre una persona, o hacer clic en “*Editar*” cuando tenemos un empleado seleccionado, aparecerá su ficha (Figura 25). La ficha del empleado es similar a la ficha que debemos rellenar al dar de alta al mismo.

En ésta, podremos editar datos referentes a la información personal y profesional al empleado además de unos datos generales que son:

- Nombre
- Apellidos
- Centro de trabajo
- Código


En cada ficha personal hay una pestaña en común, llamada “*Ficheros*”, donde se almacenan de forma organizada los archivos informáticos referentes a cada apartado. Estos archivos se guardan dentro de la base de datos de la aplicación, por lo que puede almacenar toda su información de personas en esta herramienta.

A continuación se detallan los datos almacenados en cada pestaña.

6.2.1 DATOS PERSONALES

En la pestaña “*Personal*”, se almacena la siguiente información:

- **Datos personales:**
 - DNI
 - N° Seguridad Social
 - Estado Civil
 - Fecha de nacimiento

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

- Nacionalidad
- Segunda Nacionalidad
- **Datos de contacto personal**
 - Teléfono Fijo
 - Teléfono Móvil
 - Email personal
 - Dirección
 - Código Postal
 - Ciudad
 - Provincia

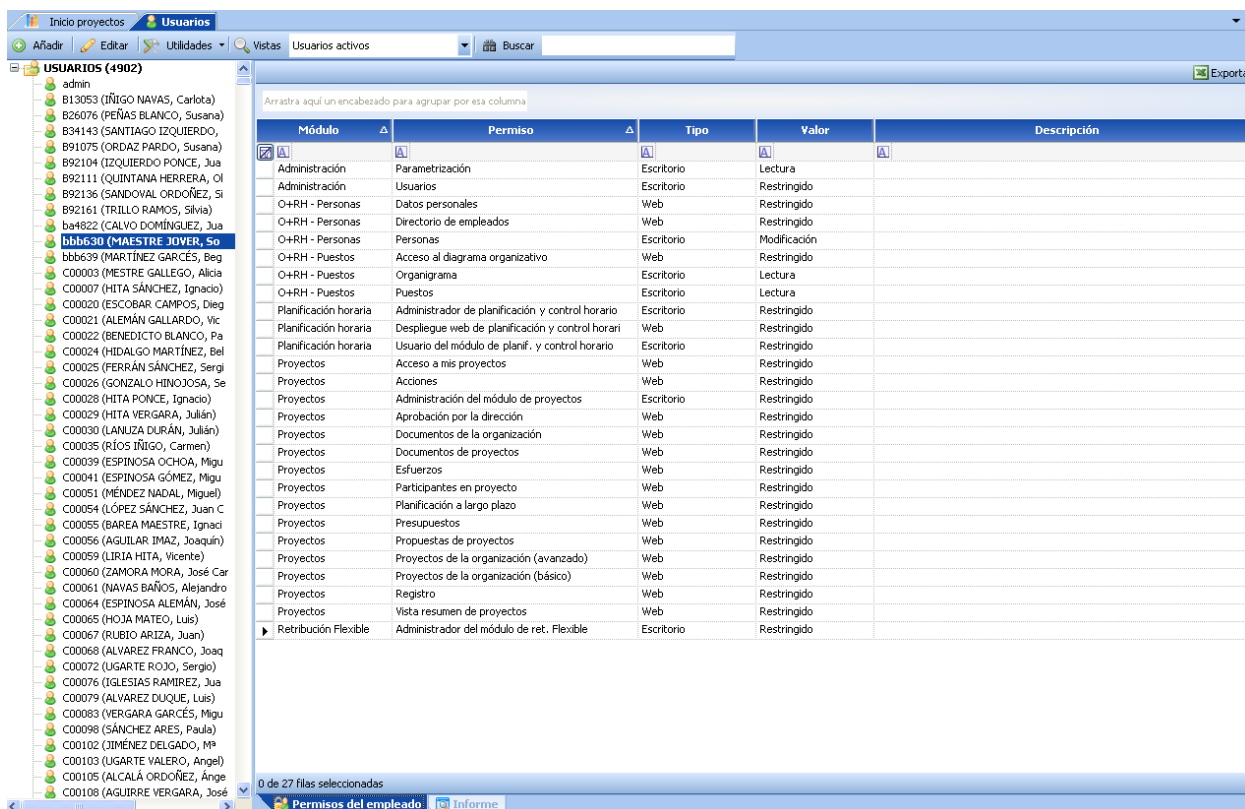
6.2.2 FICHEROS

En esta pestaña se adjuntan ficheros relacionados con el empleado.

7. USUARIOS

7.1 PANTALLA PRINCIPAL

A este módulo se accede mediante el botón de “Usuarios” colocado en la barra superior de herramientas, accediendo al listado de usuarios activos de la organización (Figura 28), donde se nos permitirá ver, editar e imprimir su información.



Módulo	Permiso	Tipo	Valor	Descripción
Administración	Parametrización	Escritorio	Lectura	
Administración	Usuarios	Escritorio	Restringido	
O+RH - Personas	Datos personales	Web	Restringido	
O+RH - Personas	Directorio de empleados	Web	Restringido	
O+RH - Personas	Personas	Escritorio	Modificación	
O+RH - Puestos	Acceso al diagrama organizativo	Web	Restringido	
O+RH - Puestos	Organigrama	Escritorio	Lectura	
O+RH - Puestos	Puestos	Escritorio	Lectura	
Planificación horaria	Administrador de planificación y control horario	Escritorio	Restringido	
Planificación horaria	Despliegue web de planificación y control horario	Web	Restringido	
Planificación horaria	Usuario del módulo de planificación y control horario	Escritorio	Restringido	
Proyectos	Acceso a mis proyectos	Web	Restringido	
Proyectos	Acciones	Web	Restringido	
Proyectos	Administración del módulo de proyectos	Escritorio	Restringido	
Proyectos	Aprobación por la dirección	Web	Restringido	
Proyectos	Documentos de la organización	Web	Restringido	
Proyectos	Documentos de proyectos	Web	Restringido	
Proyectos	Esfuerzos	Web	Restringido	
Proyectos	Participantes en proyecto	Web	Restringido	
Proyectos	Planificación a largo plazo	Web	Restringido	
Proyectos	Presupuestos	Web	Restringido	
Proyectos	Propuestas de proyectos	Web	Restringido	
Proyectos	Proyectos de la organización (avanzado)	Web	Restringido	
Proyectos	Proyectos de la organización (básico)	Web	Restringido	
Proyectos	Registro	Web	Restringido	
Proyectos	Vista resumen de proyectos	Web	Restringido	
Reintegración Flexible	Administrador del módulo de reintegración flexible	Escritorio	Restringido	

Figura 28: Usuarios activos

En la sección izquierda tendremos una estructura de lista en forma de árbol donde se muestran los usuarios ordenados alfabéticamente.

A la derecha se muestran los elementos dependientes del ítem seleccionado en el árbol. Esto es, por defecto está seleccionado en la estructura “Usuarios activos” con lo que aparecen todos los usuarios activos. También están a nuestra disposición dos vistas (opción “Vistas” en la parte superior) diferentes de esta pantalla derecha. Éstas son “Usuarios activos” y “Usuarios históricos” (Figura 29) y muestran diferentes datos de cada usuario.

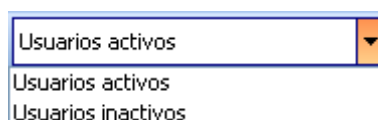



Figura 29: Vistas

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

Podemos añadir un usuario activo haciendo clic en el botón “Añadir usuario”, aparecerá una ficha (Figura 30) para rellenar los datos mínimos en el alta de un nuevo usuario.

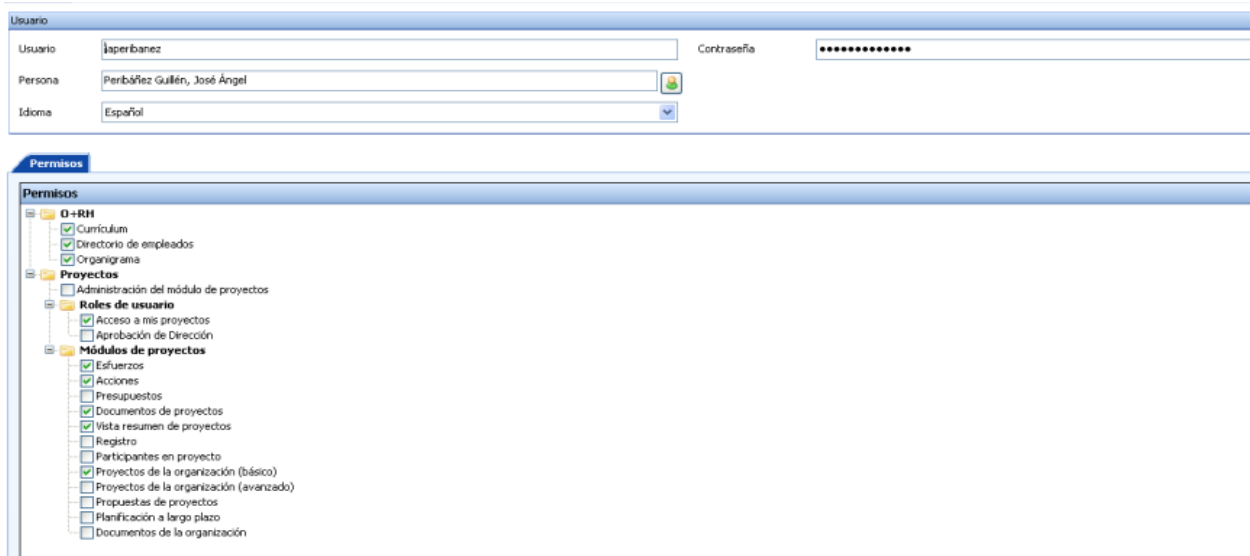


Figura 30: Ficha de usuarios

Seleccionando un usuario en la estructura de árbol podemos ver su ficha en formato HTML en pantalla derecha.

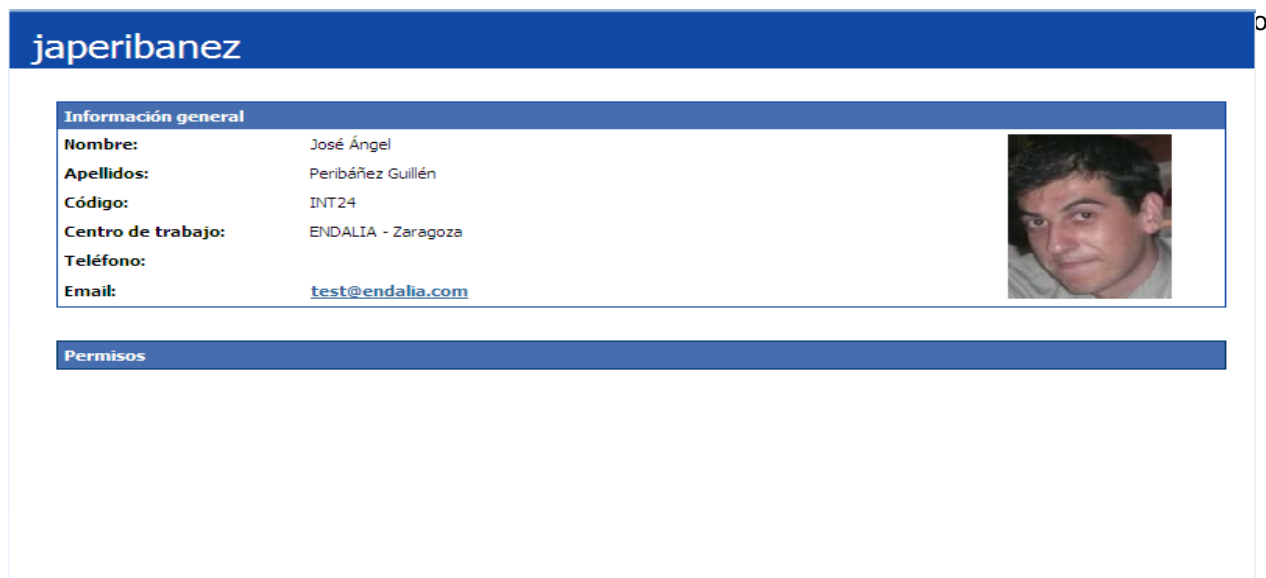



Figura 31: Ficha de usuario

Además, al seleccionar un usuario en el árbol, se habilitará el menú “Utilidades” en la zona superior (Figura 32).

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

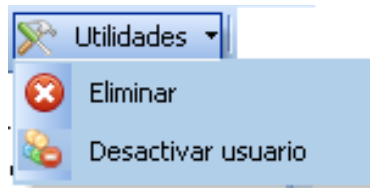



Figura 33: Menú Acciones

Desde el menú de “utilidades” de la estructura de “*Usuarios activos*”, podremos:

- Eliminar completamente un usuario del sistema.
- Desactivar temporalmente un usuario de la aplicación.

En caso de que nos encontráramos visualizando la estructura de “*Usuarios inactivos*”, el funcionamiento es análogo al de “*Usuarios activos*”, pero además podremos realizar las siguientes acciones:

- Activar un usuario

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

8. GENERACIÓN DE INFORMES

8.1 PANTALLA PRINCIPAL

A esta funcionalidad se accede mediante el botón de “*Informes*” colocado en la barra superior de herramientas, dentro del módulo de “Proyectos”.

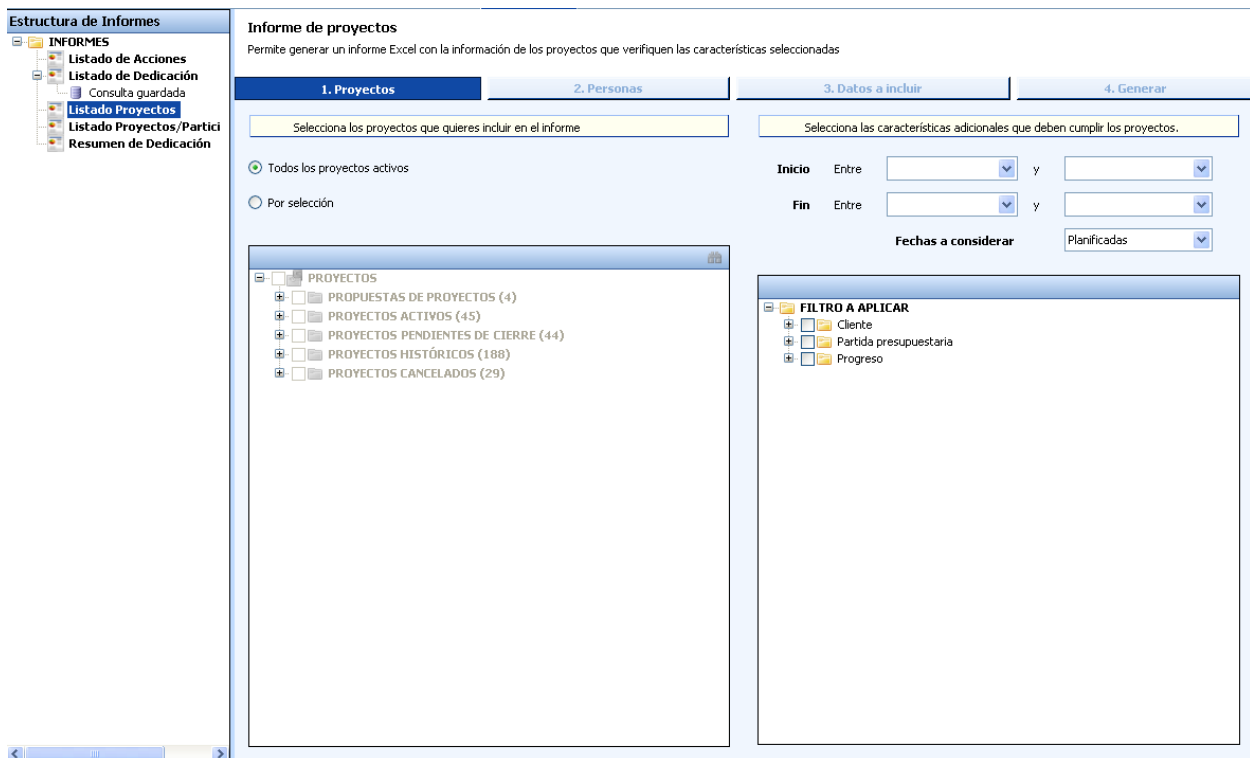


Figura 34: Pantalla inicial generación informes

Esta pantalla nos muestra en la parte izquierda una estructura con los distintos tipos de informes que se pueden generar y con los informes guardados por el usuario y que se pueden generar de forma automática.

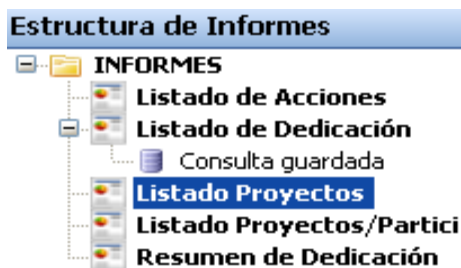



Figura 35: Estructura de informes

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

En pantalla derecha encontramos un asistente que mediante cuatro sencillos pasos guía al usuario para la generación del informe.

Informe de proyectos

Permite generar un informe Excel con la información de los proyectos que verifiquen las características seleccionadas

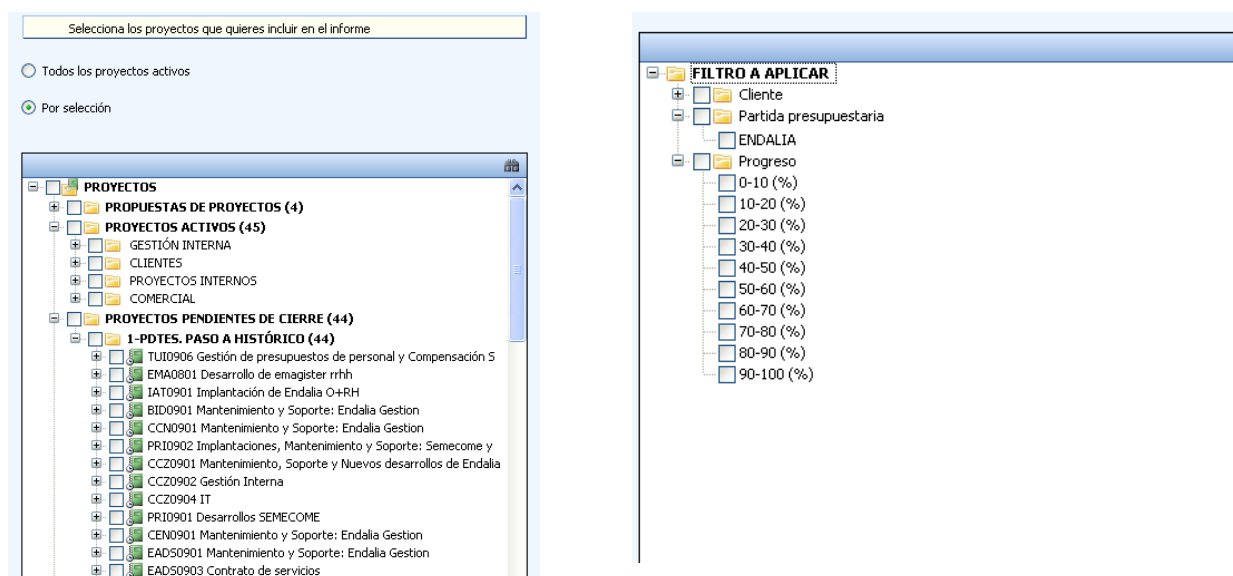
1. Proyectos	2. Personas	3. Datos a incluir	4. Generar
--------------	-------------	--------------------	------------

Figura 36: Pestañas del asistente

A continuación se detallan las características de cada etapa del asistente:

8.1.1 PROYECTOS

En este paso se seleccionan los proyectos de los que se quiere generar los informes y las características por las que filtrar esos proyectos.

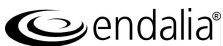


The interface is titled 'Selecciona los proyectos que quieres incluir en el informe'. It has two radio buttons: 'Todos los proyectos activos' (unselected) and 'Por selección' (selected). Below is a tree view of projects. The 'PROYECTOS' folder is expanded, showing 'PROPUESTAS DE PROYECTOS (4)', 'PROYECTOS ACTIVOS (45)', and 'PROYECTOS PENDIENTES DE CIERRE (44)'. Under 'PROYECTOS ACTIVOS', there are sub-folders for 'GESTIÓN INTERNA', 'CLIENTES', 'PROYECTOS INTERNOS', and 'COMERCIAL'. The '1-PDTES. PASO A HISTÓRICO (44)' folder is also expanded, showing a list of projects with checkboxes. To the right, the 'FILTRO A APLICAR' panel shows a tree view of filters: 'Cliente', 'Partida presupuestaria' (with 'ENDALIA' selected), and 'Progreso' (with a list of percentage ranges from 0-10% to 90-100%).

Figura 37: Filtro de selección de proyectos (parte drch.) filtro de características

8.1.2 PERSONAS

En este paso se seleccionan las personas que participan en proyecto y de las que se quiere generar los informes y las características por las que filtrar esas personas.

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

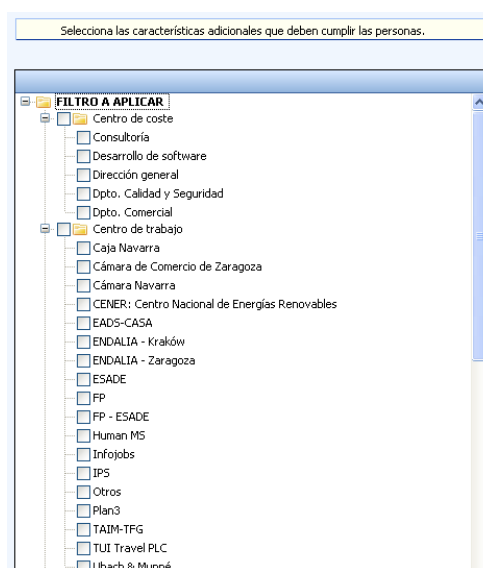


Figura 38: Filtro de características de personas

8.1.3 DATOS A INCLUIR

En este paso se seleccionan los datos que aparecerán en el informe, los datos dependen del tipo de informe seleccionado.

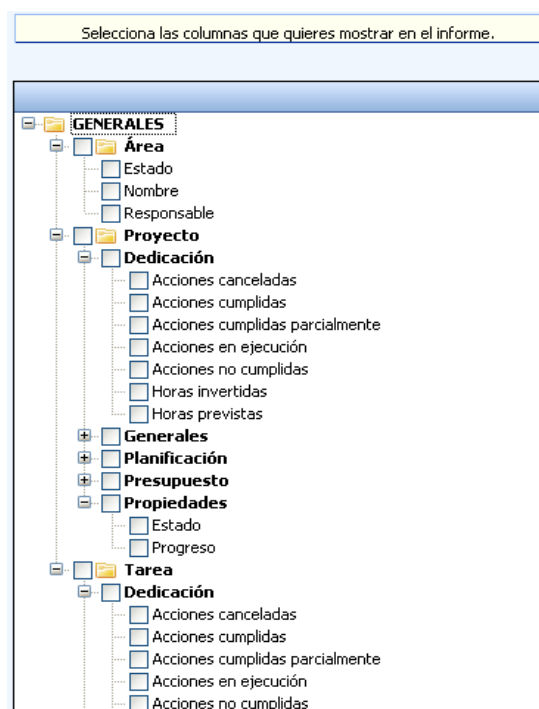



Figura 39: Selección de datos


	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

8.1.4 GENERAR INFORME

En este paso se genera el informe seleccionado por el usuario, el informe aparece en forma de lista en un grid que ofrece la posibilidad de guardar o exportarlo a Excel.

Informe Guardar consulta Exportar								
Área ▾								
Resp. Área	Estado Área	Proyecto	Cod. Proyecto	Resp. Proyecto	Estado Proyecto	Ini. plan. Project	Fin plan	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	=	=	
+ Área : BIDEAN (6 elementos)								
+ Área : CÁMARA DE COMERCIO DE NAVARRA (8 elementos)								
+ Área : CÁMARA DE COMERCIO DE ZARAGOZA (21 elementos)								
+ Área : CENER (6 elementos)								
+ Área : COMERCIAL (148 elementos)								
+ Área : EADS - AIRBUS MILITARY (33 elementos)								
+ Área : EADS - AIRBUS OPERATIONS (75 elementos)								
+ Área : EADS - DEFENCE & SECURITY (31 elementos)								
+ Área : EADS - SPACE (14 elementos)								
+ Área : EMAGISTER (15 elementos)								
+ Área : ESADE (22 elementos)								
+ Área : GESTIÓN INTERNA (28 elementos)								
+ Área : HUMAN (8 elementos)								
+ Área : IAT (6 elementos)								
+ Área : INFOJOBS (36 elementos)								
+ Área : LAIE (7 elementos)								
+ Área : PRI (77 elementos)								
+ Área : PROYECTOS INTERNOS (36 elementos)								
+ Área : SMC (10 elementos)								
+ Área : TUI (189 elementos)								
+ Área : UBACH (10 elementos)								

Figura 40: Informe

	Sistema para la gestión de proyectos	Versión: 1.2
	Manual de Usuario	Fecha: 21/05/2010
	MANUALUSUARIO.doc	

9. BIBLIOGRAFÍA

9.1 *REFERENCIAS*

En este documento no se realizan referencias a otros documentos.

9.2 *REFERENCIAS WEB*

En este documento no se realizan referencias a recursos disponibles en la Web.

