

Proyecto Fin de Carrera

Desarrollo de sistema informático para la automatización del proceso de gestión de objetivos y retribución de incentivos.

ANEXOS

Autor
Santiago Pelegrín Costea

Director
Fernando Cortés Franco
Director/CEO (Endalia)

PONENTE
Santiago Velilla Marco
Departamento de Informática e Ingeniería de Sistemas (Universidad de Zaragoza)

Escuela de Ingeniería y Arquitectura
Septiembre 2013

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 ayuda
- Documento de instalación

Sistema de Gestión de Objetivos y Retribución de Incentivos.

Estándar de Documentación

Versión 2.2

Publicado el 16/02/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
03/09/2007	1.0	Redacción del Estándar de documentación de Endalia	Endalia
15/02/2013	2.0	Modificación del documento para adaptarlo a las características del SGORI	Santiago Pelegrín Costea
15/02/2013	2.1	Revisión del documento	Fernando Cortés Franco
16/02/2013	2.2	Corrección del documento	Santiago Pelegrín Costea

Índice

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.	Formato de documentación	6
2.1	Fuentes y estilos.....	6
2.2	Interlineado y formato de párrafos	6
2.3	Imágenes y diagramas	7
3.	Plantillas de documentación	8
3.1	Plantilla de documento.....	8
3.1.1	Hoja 1. Portada.....	8
3.1.2	Hoja 2. Información de copyright	9
3.1.3	Hoja 3. Información de documento.....	9
3.1.4	Hoja 4 y siguientes. Índice.....	10
3.2	Plantilla de acta de reunión	10
4.	Bibliografía	12
4.1	Referencias.....	12
4.2	Referencias Web	12

1. Introducción

1.1 Propósito del documento

En el presente documento se define el estándar de documentación del proyecto de desarrollo del SGORI. Se definen los formatos, diseños, tipología de fuentes y plantillas a utilizar para elaborar esta documentación.

1.2 Alcance del documento

Las especificaciones de este documento alcanzan a toda la documentación generada durante el proyecto.

1.3 Acrónimos

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos

1.4 Definiciones

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

1.5 Referencias

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

1.6 Resumen

El presente documento es el estándar de documentación de Endalia. Se compone de 4 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 especifica el formato de documentación del proyecto.
- Apartado 3. Plantillas de documentación
- Apartado 4. Bibliografía y referencias Web utilizadas en la confección de este documento.

2. Formato de documentación

2.1 Fuentes y estilos

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

- Título 1: Helvética Neue 18, Negrita.
- Título 2: Helvética Neue 14, Negrita.
- Título 3: Helvética Neue 12.
- Texto normal: Helvética Neue 10.
- Texto en pie de imágenes: Helvética Neue 9, Cursiva.
- Texto de Código fuente o acciones de línea de comandos: Courier New 10.

2.2 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.

Entre todos los títulos independientemente del nivel, habrá dos saltos de línea de separación.

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

2.3 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 colocarán 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, de la siguiente manera (Figura 1):



Figura 1. Ejemplo de formato de imagen

3. Plantillas de documentación

Las plantillas explicadas en este apartado se encuentran guardadas en formato electrónico. 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 2).

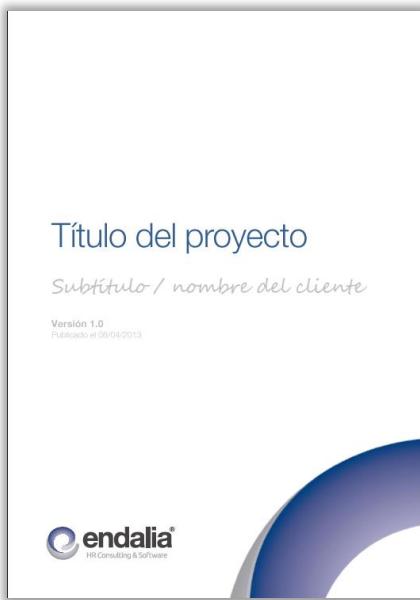


Figura 2. Hoja 1 - Portada

En él se indica el título del documento con fuente Helvética Neue 48 y se incluye información de copyright y de control de distribución y autorización.

Todas las demás páginas del documento poseen una cabecera común, en la que aparece el logotipo de Endalia. El pie de página es también común a todas las páginas (a excepción de la portada), e incluye el título del documento, así como información del número de página.

3.1.2 Hoja 2. Información de copyright

El formato de la segunda hoja de todos los documentos es el siguiente (Figura 3).

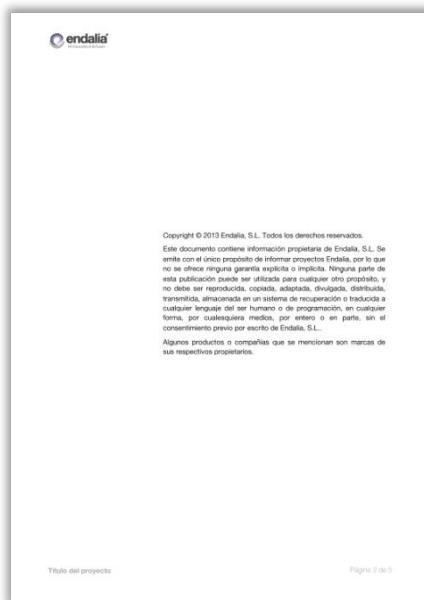


Figura 3. Hoja 2 - Información de copyright

En esta hoja, aparte de los elementos comunes de cabecera y pie de página, aparece información relativa al copyright del documento.

3.1.3 Hoja 3. Histórico de revisiones

El formato de la tercera hoja de todos los documentos es el siguiente (Figura 4).

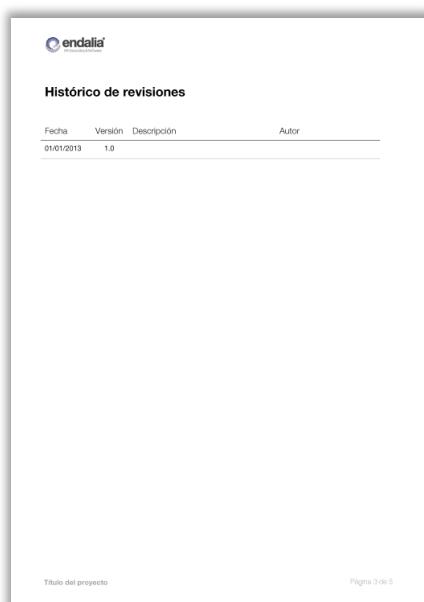
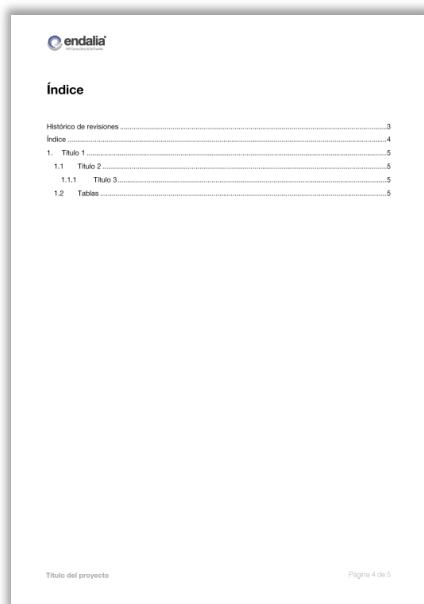


Figura 4. Hoja 3 – Histórico de revisiones

En esta hoja, aparte de los elementos comunes de cabecera y pie de página, aparece un cuadro con información acerca de las sucesivas revisiones realizadas sobre el documento.

3.1.4 Hoja 4 y siguientes. Índice

El formato del índice se indica en la siguiente imagen (Figura 5). Se genera automáticamente con Word, estableciendo el tipo de letra de los apartados de nivel 1 a Helvética Neue 10 Negrita.



Histórico de revisiones	3
Índice	4
1. Título 1	5
1.1. Título 2	5
1.1.1. Título 3	5
1.2. Tablas	5

Figura 5. Índice

3.2 Plantilla de acta de reunión

El formato de plantilla de acta de reunión es el siguiente (Figura 6 y Figura 7). En la primera hoja se incluye la fecha, hora y lugar de la reunión, la persona emisora del documento, la fecha de emisión del mismo, los asistentes, la distribución y el orden del día. En la segunda hoja se incluye el acta de la reunión, donde aparecen, divididos por secciones, los distintos acuerdos alcanzados, la persona responsable de los mismos, la fecha de compromiso y el estado, expresado en cifra porcentual.

 **endalia®**

Nombre de la sesión

Este documento pretende recopilar la información de la reunión de [NOMBRE DE LA SESIÓN] de [CLIENTE] que tendrá lugar el próximo día [FECHA] en [DIRECCIÓN]. A continuación se define el Orden del día. Posteriormente se ampliará el presente documento con el acta de la sesión y los acuerdos alcanzados.

Fecha / hora _____
Lugar _____
Emisor _____
Fecha emisión _____
Asistentes _____
Distribución _____

Orden del día

Informe de sesión de trabajo | [Nombre de la empresa]
Página 1 de 2

Figura 6. Acta de reunión, hoja 1

 **endalia®**

Acta de la reunión

Sección 1

Acuerdo	Responsable	Compromiso	Estado
		01/01/2013	0%

Sección 2

Acuerdo	Responsable	Compromiso	Estado
		01/01/2013	0%

Informe de sesión de trabajo | [Nombre de la empresa]
Página 2 de 2

Figura 7. Acta de reunión, hoja 2

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

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 de Gestión de Objetivos y Retribución de Incentivos.

Estándar de Codificación

Versión 2.2

Publicado el 16/02/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
17/03/2005	1.0	Redacción inicial del documento	Endalia
15/02/2013	2.0	Modificación del documento para adaptarlo a las características del SGORI	Santiago Pelegrín Costea
15/02/2013	2.1	Revisión del documento	Fernando Cortés Franco
16/02/2013	2.2	Corrección del documento	Santiago Pelegrín Costea

Índice

1. Introducción	7
1.1 Propósito del documento	7
1.2 Alcance del documento.....	7
1.3 Acrónimos.....	7
1.4 Definiciones	8
1.5 Referencias.....	8
1.6 Resumen.....	8
2. Estructura de los ficheros	9
2.1 Introducción.....	9
2.2 Codificación de archivos de código fuente.....	9
2.2.1 Consideraciones generales sobre las regiones.....	10
2.2.2 Region directivas <i>using</i>	10
2.2.3 Región declaración <i>namespace</i> y <i>class</i>	10
2.2.4 Región declaración de variables	10
2.3 Codificación de archivos de clase de acceso a datos.....	11
2.3.1 Región directivas <i>using</i>	11
2.3.2 Región declaración <i>namespace</i> y <i>class</i>	11
2.3.3 Región atributos	12
2.3.4 Región constructores	12
2.3.5 Región propiedades	12
2.3.6 Región métodos	12
2.4 Codificación de archivos de recursos de internacionalización.....	13
3. Reglas de codificación.....	14
3.1 Identación	14
3.1.1 Longitud de línea	14
3.1.2 Ruptura de líneas.....	14
3.2 Comentarios	14
3.2.1 Aplicación de los comentarios	15
3.2.2 Formatos de implementación de comentarios.....	16
3.3 Declaraciones	17
3.3.1 Número de declaraciones por línea.....	17
3.3.2 Inicialización	17
3.3.3 Situación.....	17

3.4	Sentencias	18
3.4.1	Sentencias simples.....	18
3.4.2	Sentencias compuestas.....	18
3.4.3	Sentencias de retorno	18
3.4.4	Sentencias <i>if</i> , <i>if-else</i> , <i>if else- if else</i>	18
3.4.5	Sentencias <i>for</i>	19
3.4.6	Sentencias <i>while</i>	19
3.4.7	Sentencias <i>do-while</i>	20
3.4.8	Sentencias <i>switch</i>	20
3.4.9	Sentencias <i>try-catch</i>	20
3.5	Espacios en blanco	21
3.5.1	Líneas en blanco.....	21
3.5.2	Espacios en blanco	21
3.6	Convenciones de nombres.....	22
3.6.1	Clases	22
3.6.2	Métodos.....	22
3.6.3	Variables y parámetros.....	22
3.6.4	Constantes.....	22
3.7	Hábitos de programación.....	23
3.7.1	Referencias a variables y métodos de clase.....	23
3.7.2	Constantes.....	23
3.7.3	Asignaciones de variables.....	23
3.7.4	Paréntesis.....	24
3.7.5	Variables de retorno	24
3.7.6	Expresiones antes de '?' en el operador condicional	24
3.7.7	Comentarios especiales	24
4.	Estándar de nombrado de Base de Datos.....	25
4.1	Idioma a utilizar.....	25
4.2	Convenciones de nombrado de tablas	25
4.2.1	Nombrado de tablas de entidad.....	25
4.2.2	Nombrado de tablas de relación.....	25
4.2.3	Nombres de tablas predefinidos	26
4.3	Convenciones de nombrado de campos.....	26
4.3.1	Nombrado de Campos de Tablas de Entidad.....	26
4.3.2	Nombrado de campos de tablas de relación.....	27

5. Bibliografía	28
5.1 Referencias.....	28
5.2 Referencias Web	28

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 son importantes 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 del Sistema de Gestión de Objetivos y Retribución de Incentivos, 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

- ANSI: American National Standards Institute.
- HTML: Hypertext Markup Language.
- ID: Identificador.
- URL: Uniform Resource Locator.

1.4 Definiciones

- Pascal-Casing: Notación en la que un identificador está compuesto por múltiples palabras juntas comenzando cada una de ellas por una letra mayúscula.
- Camel-Casing: Notación similar a Pascal-Casing con la excepción de que la letra inicial del identificador debe ser minúscula.

1.5 Referencias

En este documento no se realizan referencias 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. 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 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 muestra el estándar de nombrado de los elementos de base de datos.
- Apartado 5: Bibliografía y referencias Web utilizadas en la confección de este documento.

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. Asimismo se especifica la codificación para los archivos de clases y acceso a datos (.cs) y para los archivos de recursos de internacionalización (.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 con dos columnas. En la columna de la izquierda aparece un índice para facilitar la posterior descripción de la región de la estructura definida en la columna de la derecha.

<1> Directivas <i>using</i>	using System; <i><directivas using></i>
<2> Declaración <i>namespace</i> y <i>class</i>	namespace MyNamespace1 { public class MyClass {
<3> Región variables	#region variables #region Variables I18N <i><declaraciones de variables de internacionalización></i> #endregion Variables I18N #region Variables globales <i><declaraciones de variables globales></i> #endregion Variables globales #endregion variables
<4> Región <i>I18N</i>	#region I18N <i>< Código InitializeLabels ></i> <i>< Código LoadI18N ></i> #endregion I18N
	}

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 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.

2.2.2 Region 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.

2.2.3 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 © 2013, Endalia, S.L. Todos los derechos reservados
/// </summary>
```

2.2.4 Región declaración de variables

Esta región estará integrada por cuatro 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.
- 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 éstas en variables de cadena.
 - Método *LoadI18N*: Carga en los campos de texto del formulario las variables de cadenas obtenidas en el método *InitializeLabels*.

2.3 Codificación de archivos de clase de acceso a datos

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 anterior, se muestra mediante una tabla con dos columnas. En la columna de la izquierda aparece un índice para facilitar la posterior descripción de la región de la estructura definida en la columna de la derecha.

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

2.3.1 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.

2.3.2 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 © 2005, Formación y Perfeccionamiento, S.L. Todos los derechos reservados
/// </summary>
```

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 _indicatorID;
```

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 IndicatorID
{
    get{ return _indicatorID; }
    set{ _indicatorID = value; }
}
```

2.3.6 Región métodos

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

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 la siguiente convención:
 - etiqueta o campo de texto: `_lbl_`
 - etiqueta de hyperlink: `_lnk_`
 - texto de botón : `_btn_`
- En el caso de que la etiqueta sea un *tooltip*, se colocará a continuación el prefijo `_toolTip_`
- 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.

```
IndicatorsTargetStruct_lbl_IndicatorName
IndicatorsTargetStruct_btn_OpenIndicatorInfo
IndicatorRemunerationStruct_btn_toolTip_DeleteRemunerationInfo
```

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 identación estándar. Los comentarios se identarán al mismo nivel de identación 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 entra 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 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 etiquetas o tags de documentación de nombres, parámetros y valores de retorno utilizados para la documentación automatizada.

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 las etiquetas 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 sintamos 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, sino 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.
- 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 identada 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 identados 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
}
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 identación
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++) { ... }
```

Debemos evitar las declaraciones locales que ocultan 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 identarse uno o más niveles que la sentencia compuesta.
- La llave ({}) de apertura debe empezar una nueva línea a continuación de la que empieza la sentencia compuesta; la llave de cierre (}) debe empezar una nueva línea y estar identada 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 esta regla 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);
```

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

Las sentencias de tipo *if-else* deberán 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;
```

Aunque es perfectamente válido a nivel de código, puede llevar a confusión y a la introducción de errores.

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.

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
        /* continúa 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 “/* continúa 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*.

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;
}
```

```
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 deberá 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.

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
{
    ...
}
```

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 pública 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.

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 asignaciones embebidas como un intento de mejorar el rendimiento en tiempo de ejecución. Ese es el trabajo del compilador. Ejemplo:

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

Debería escribirse como:

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

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 la etiqueta FIXME para marcar algo que tiene algún error y no funciona.

4. Estándar 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 base de datos 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 base de datos, 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 el 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  
Orh_Tra_GenericAreas  
Obj_Ind_Indicators  
Obj_Ind_IndicatorPrivileges
```

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 alguno de ellos.

De este modo, para relacionar las tablas 'Obj_Ind_Indicators' y 'Orh_Emp_Employees' se creará la tabla 'R_IndicatorEmployee'

4.2.3 Nombres de tablas predefinidos

- R : Relación. Tablas de relación.
- Gen : Generic. Tablas relativas a la organización general de la base de datos.
- Orh : 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 base de datos 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

- Nombrado de claves primarias: Las claves primarias comenzaran 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’.
- Nombrado de claves ajenas: Las claves ajenas comenzaran 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’.
- 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 ‘Orh_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
```

4.3.2 Nombrado de campos de tablas de relación

- 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_IndicatorEmployee’ que relaciona las tablas ‘Obj_Ind_Indicators’ y ‘Orh_Emp_Employees’ se nombrarán de esta manera las claves ajenas:

fk_IndEmpEmployeeID
fk_IndEmpIndicatorID

- Nombrado de campos: Los campos en las tablas de relación se nombrarán mediante Pascal-Casing. Comenzaran 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_IndicatorEmployee’ que relaciona las tablas ‘Obj_Ind_Indicators’ y ‘Orh_Emp_Employees’ los nombres de los campos podrían ser:

IndEmpRemarks
IndEmpPercentWeight
IndEmpIsCompulsory

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 de Gestión de Objetivos y Retribución de Incentivos.

Plan de Gestión de Configuraciones

Versión 2.2

Publicado el 16/02/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
04/09/2007	1.0	Redacción del Plan de Gestión de Configuraciones de Endalia	Endalia
15/02/2013	2.0	Modificación del documento para adaptarlo a las características del SGORI	Santiago Pelegrín Costea
15/02/2013	2.1	Revisión del documento	Fernando Cortés Franco
16/02/2013	2.2	Corrección del documento	Santiago Pelegrín Costea

Índice

1.	Introducción	5
1.1	Propósito del documento	5
1.2	Alcance del documento.....	6
1.3	Acrónimos.....	6
1.4	Definiciones	6
1.5	Referencias.....	7
1.6	Resumen.....	7
2.	Especificaciones de gestión.....	8
3.	Actividades de Gestión de Configuraciones.....	9
3.1	Identificación de la Configuración.....	9
3.2	Gestión de la Configuración.....	9
3.3	Control de cambios de la Configuración.....	10
4.	Herramientas Utilizadas.....	11
5.	Bibliografía	14
5.1	Referencias.....	14
5.2	Referencias Web	14

1. Introducción

1.1 Propósito del documento

En el presente documento se definen las políticas, estándares, procedimientos y actividades asociadas a la gestión de configuraciones de este proyecto.

En el desarrollo de software son habituales los cambios en todos los elementos generados, generalmente debidos a modificaciones de especificación de requisitos y fallos de análisis o diseño. Es preciso 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 a las tareas realizadas por otros miembros del equipo de trabajo (si los hubiera).

En términos generales, la Gestión de Configuración del Software (GCS) se puede definir como una disciplina cuya misión es controlar la evolución de un sistema software.

Según Wayna A. Babich, uno de los personajes que más han publicado sobre este tema: “El arte de coordinar el desarrollo de software para minimizar la confusión, se denomina Gestión de Configuración Software. La GCS es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores”.

El objetivo de la GCS es mantener la integridad de los productos que se obtienen a lo largo del desarrollo de los sistemas de información, garantizando que no se realizan cambios incontrolados y que todos los participantes en el desarrollo del sistema disponen de la versión adecuada de los productos que manejan.

La GCS se realiza durante todas las actividades asociadas al desarrollo del sistema, y continúa registrando los cambios hasta que éste deja de utilizarse.

De la misma forma, la GCS facilita el mantenimiento del sistema, aportando información precisa para valorar el impacto de los cambios solicitados y reduciendo el tiempo de implementación de un cambio, tanto evolutivo como correctivo. Asimismo, permite controlar el sistema como producto global a lo largo de su desarrollo, obtener informes sobre el estado de desarrollo en que se encuentra y reducir el número de errores de adaptación y/o integración del sistema, lo que se traduce en un aumento de calidad del producto, de la satisfacción del cliente y, en consecuencia, de la mejora de la organización.

Los productos registrados en el sistema de gestión de la configuración se encuentran identificados y localizados únicamente, de manera que la información relativa a los productos es de fácil acceso.

Como resumen, se puede afirmar que la GCS es una disciplina de control, dentro del proceso de desarrollo del proyecto.

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 de Software.
- IEEE: Institute of Electrical and Electronic Engineers (Instituto de Ingenieros Eléctricos y Electrónicos).
- PGCS: Plan de Gestión de Configuraciones de Software.
- RUP: Rational Unified Process.
- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos
- TFS: Team Foundation Server

1.4 Definiciones

- Artefacto: Cualquier tipo de información creada, producida, cambiada o utilizada por el equipo de desarrollo del proyecto.
- Pascal-Casing: Notación en la que un identificador está compuesto por múltiples palabras juntas comenzando cada una de ellas por una letra mayúscula.
- 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.
- 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.

1.6 Resumen

Este documento describe el plan de gestión de configuraciones del SGORI. Se compone de cinco apartados:

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

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 Gestión de Configuración 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 actividades:

- Identificación de la Configuración: consiste en identificar la estructura del producto, sus componentes y la naturaleza de éstos, 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 él 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.
- Auditoría 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.

3. Actividades de Gestión de Configuraciones

En esta sección se van a describir 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 el nombrado de los diferentes ECS que constituye cada artefacto que sigue las normas definidas en este PGCS, se debe seguir la convención de un nombre en mayúsculas, resumiendo lo más claramente posible el contenido del artefacto, seguido de una descripción breve del tipo de elemento que constituye, si se estima necesario, excepto en el caso de artefactos correspondientes a código fuente, que se nombrarán siguiendo la notación Pascal-Casing, o archivos binarios y scripts que se nombrarán siguiendo Camel-Casing. En cualquier caso, el nombre del artefacto deberá estar concluido por un punto seguido de una extensión representativa del contenido del mismo siguiendo los 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.

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 mismo documento.

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 ésta y la línea base a la que afecta.

4. Herramientas Utilizadas

Como se ha comentado anteriormente, la herramienta utilizada para la gestión de cambios de las configuraciones de software es Microsoft Team Foundation Server (TFS, aplicación que se integra con el entorno de desarrollo Visual Studio (también de Microsoft), permitiendo realizar 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 utilizado para el desarrollo del presente proyecto, lo que facilita 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 no permite que otros desarrolladores puedan tener acceso al mismo excepto que sea en modo de sólo lectura.

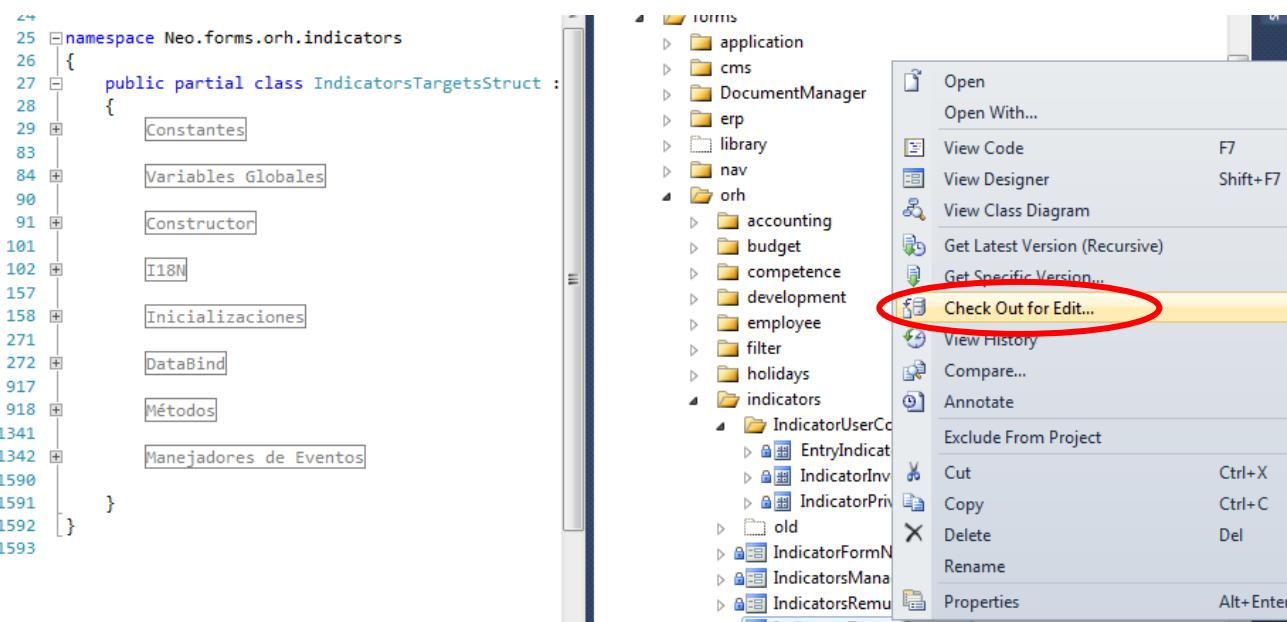


Figura 1: Bloqueo de ECS

- Descargar versión de la biblioteca. Esta funcionalidad permite la actualización de ECS en la biblioteca local de cada desarrollador para obtener la última versión con los últimos elementos modificados por el resto del equipo.

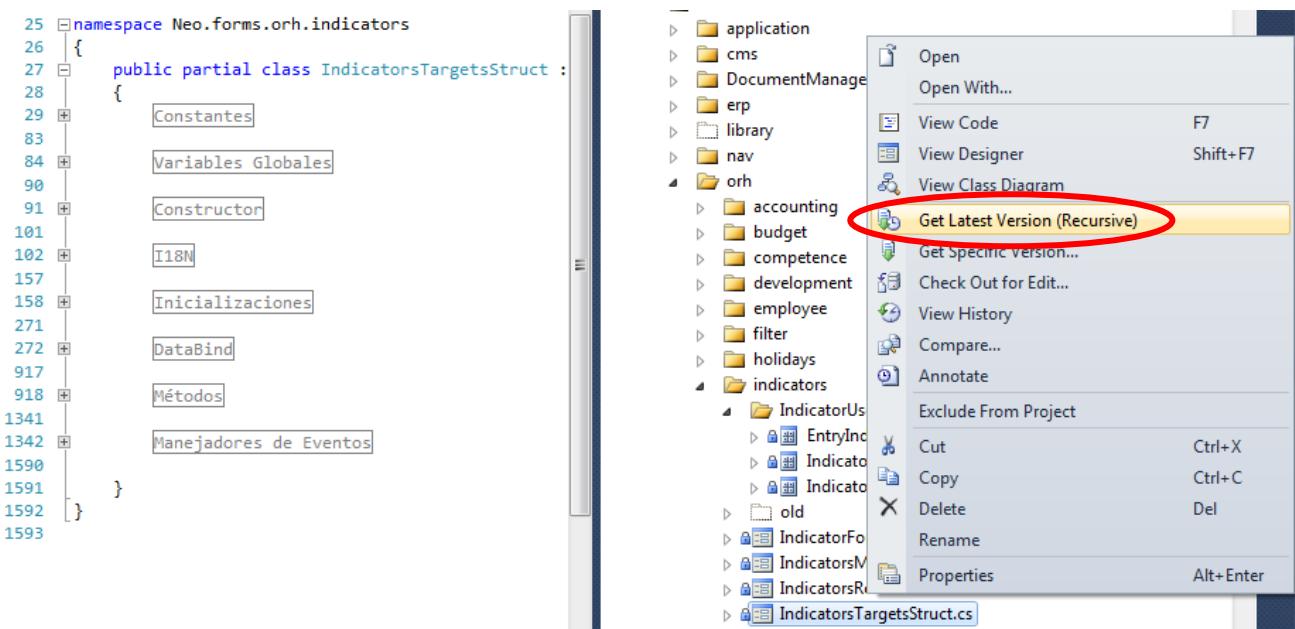


Figura 2: Descarga de versión

- Modificación de ECS. Esta funcionalidad permite la actualización de uno o varios ECS por parte de un desarrollador que los tenía previamente bloqueados, debiendo éste indicar la petición en el formato previamente establecido.

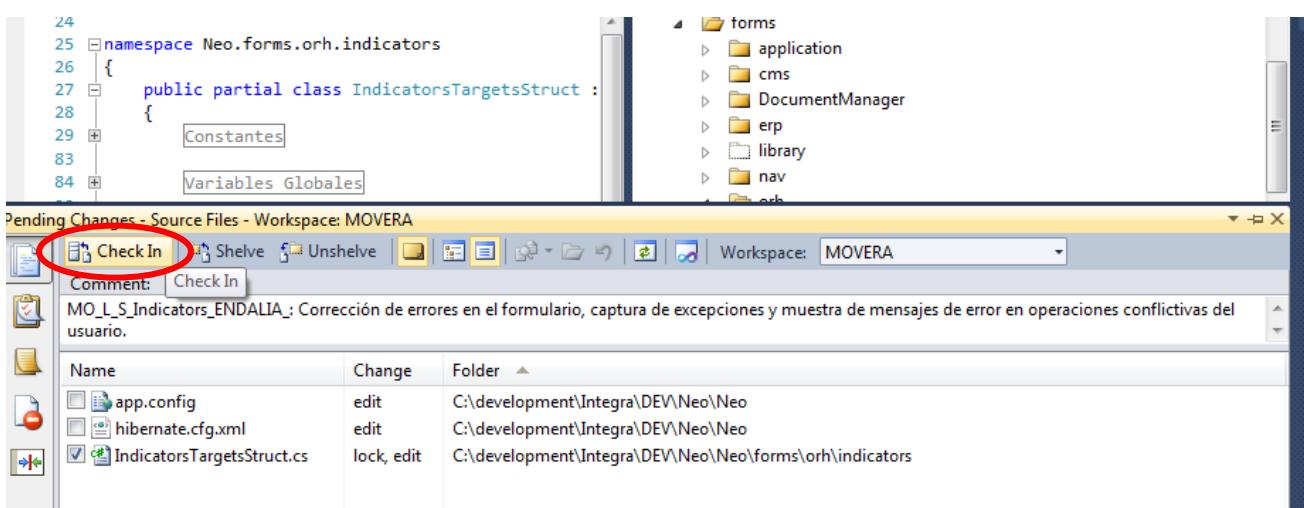
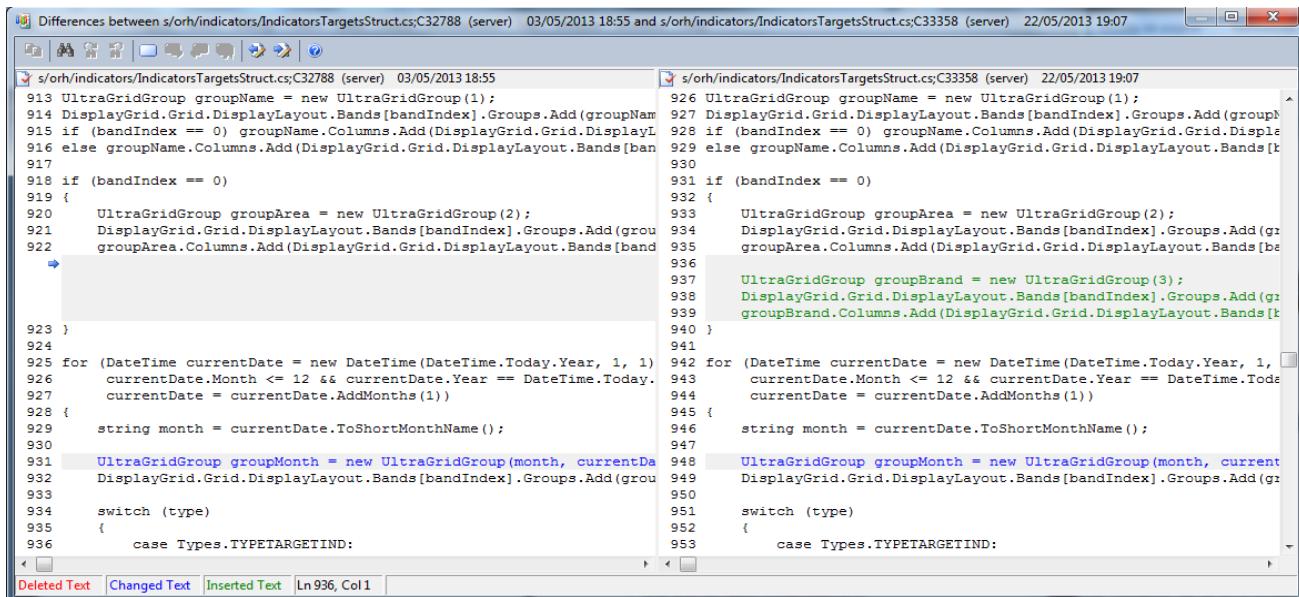


Figura 3: Modificación de ECS

- 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.



```

913 UltraGridGroup groupName = new UltraGridGroup(1);
914 DisplayGrid.Grid.DisplayLayout.Bands[bandIndex].Groups.Add(groupName)
915 if (bandIndex == 0) groupName.Columns.Add(DisplayGrid.Grid.DisplayL
916 else groupName.Columns.Add(DisplayGrid.Grid.DisplayLayout.Bands[ban
917
918 if (bandIndex == 0)
919 {
920     UltraGridGroup groupArea = new UltraGridGroup(2);
921     DisplayGrid.Grid.DisplayLayout.Bands[bandIndex].Groups.Add(grou
922     groupArea.Columns.Add(DisplayGrid.Grid.DisplayLayout.Bands[band
923 }
924
925 for (DateTime currentDate = new DateTime(DateTime.Today.Year, 1, 1)
926     currentDate.Month <= 12 && currentDate.Year == DateTime.Today.
927     currentDate = currentDate.AddMonths(1))
928 {
929     string month = currentDate.ToShortMonthName();
930
931     UltraGridGroup groupMonth = new UltraGridGroup(month, currentDa
932     DisplayGrid.Grid.DisplayLayout.Bands[bandIndex].Groups.Add(grou
933
934     switch (type)
935     {
936         case Types.TYPETARGETIND:
937             UltraGridGroup groupMonth = new UltraGridGroup(month, current
938             DisplayGrid.Grid.DisplayLayout.Bands[bandIndex].Groups.Add(g
939             groupMonth.Columns.Add(DisplayGrid.Grid.DisplayLayout.Bands[band
940 }
941
942 for (DateTime currentDate = new DateTime(DateTime.Today.Year, 1,
943     currentDate.Month <= 12 && currentDate.Year == DateTime.Today.
944     currentDate = currentDate.AddMonths(1))
945 {
946     string month = currentDate.ToShortMonthName();
947
948     UltraGridGroup groupMonth = new UltraGridGroup(month, current
949     DisplayGrid.Grid.DisplayLayout.Bands[bandIndex].Groups.Add(g
950
951     switch (type)
952     {
953         case Types.TYPETARGETIND:

```

Deleted Text | Changed Text | Inserted Text | Ln 936, Col 1

Figura 4: Resolución de conflictos

- Deshacer las últimas modificaciones realizadas, y volver a la versión previa de ECS. Esta funcionalidad permite revertir los últimos cambios realizados por el desarrollador, volviendo así a la versión anterior que fue registrada.

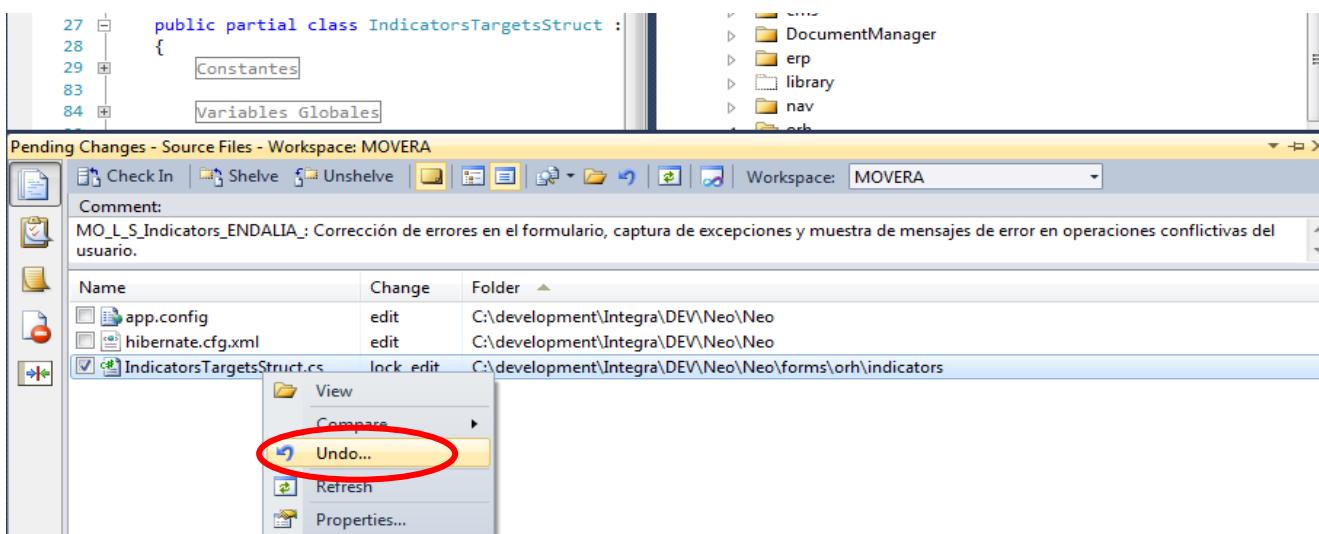


Figura 5: Descartar modificaciones

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 de Gestión de Objetivos y Retribución de Incentivos

Estudio de Mercado

Versión 1.2

Publicado el 04/03/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
22/02/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
25/02/2013	1.1	Revisión del documento	Fernando Cortés Franco
04/03/2013	1.2	Corrección del documento	Santiago Pelegrín Costea

Índice

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 Definiciones	6
1.6 Resumen.....	6
2. Gestión de objetivos y retribución de incentivos.....	7
2.1 Introducción.....	7
2.2 Definición de gestión de objetivos y retribución de incentivos.....	7
2.3 Situación actual de la gestión de objetivos y retribución de incentivos.....	8
2.4 Elementos de la gestión de objetivos y retribución de incentivos.....	8
2.4.1 Indicadores	9
2.4.2 Objetivos.....	9
2.4.3 Elegibilidad	9
2.4.4 Gestión/Administración	9
2.4.5 Condicionantes.....	9
2.4.6 Ponderación.....	9
3. Sistemas de Gestión de Objetivos y Retribución de Incentivos.....	10
3.1 Introducción.....	10
3.2 Sistemas actuales.....	10
3.2.1 Cezanne HR.....	10
3.2.2 IDARVE Software	12
3.2.3 Ceinsa	12
4. Conclusiones.....	15
5. Bibliografía	16
5.1 Referencias.....	16
5.2 Referencias web.....	16

1. Introducción

1.1 Propósito del documento

El objetivo de este documento consiste en estudiar los conceptos que van a ser manejados a lo largo de todo el desarrollo del proyecto, para así comprender mejor las necesidades y características del producto a desarrollar.

Por otra parte, se analizarán algunas de las aplicaciones software más destacadas del mercado, orientadas a la gestión de objetivos y retribución de incentivos en pequeñas y medianas empresas.

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 desarrollo del Sistema de Gestión de Objetivos y Retribución de Incentivos. Las conclusiones obtenidas en este estudio serán aplicadas en el resto de fases del proyecto. El alcance del documento comprende toda la duración del proyecto, ya que su contenido puede ser actualizado en cualquier momento, si se detectan variaciones relevantes en el mercado.

1.3 Acrónimos

- RRHH: Recursos Humanos
- PYME: Pequeña y Mediana Empresa
- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos
- ERP: Enterprise Resource Planning

1.4 Definiciones

- Pequeña y mediana empresa: empresa con características distintivas, con dimensiones con ciertos límites ocupacionales y financieros prefijados por los Estados o Regiones. Son agentes con lógicas, culturas, intereses y espíritu emprendedor específico [Ref. Web 1].
- Outsourcing: proceso en el cual una empresa identifica una porción de su proceso de negocio que podría ser desempeñada más eficientemente y/o más efectivamente por otra corporación, la cual es contratada para desarrollar esa porción de negocio.
- Sistema ERP: Un ERP *Enterprise Resource Planning* (Gestión de Recursos Empresariales) es una aplicación informática que permite relacionar en un solo sistema integrado de datos, todos los procesos del negocio (planificación estratégica, recursos humanos, compras, ventas, marketing, logística, distribución, proyectos, entre otros).

1.5 Definiciones

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

1.6 Resumen

El presente documento es un estudio de mercado que se enmarca dentro de la fase inicial de obtención de información del proyecto de desarrollo de un SGORI. 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 documentos del proyecto referenciados y el presente resumen.
- Apartado 2. Se muestra una visión global de la técnica de gestión de objetivos y de retribución de incentivos, destacando sus principales funciones y elementos, y explicando la situación actual de la misma.
- Apartado 3. Se muestran algunos SGORI y sus funcionalidades principales.
- Apartado 4. Se describen las conclusiones obtenidas durante esta fase del proyecto.
- Apartado 5. Se muestra la bibliografía y las referencias Web utilizadas.

2. Gestión de objetivos y retribución de incentivos

2.1 Introducción

Como se ha indicado en el apartado 1.1 (“Propósito del documento”) del presente texto, el punto de partida para posteriores fases del proyecto (tales como la identificación de requisitos o el análisis del sistema) va a ser el estudio de mercado. Éste nos permitirá tener una visión global del entorno y nos ayudará a la hora de definir qué elementos, fases y técnicas tenemos que trasladar del aspecto conceptual de la gestión de objetivos y retribución de incentivos al desarrollo real del software que nos ocupa.

En este apartado se desarrollan los siguientes temas:

- Subapartado 2.2: Definición de gestión de objetivos y retribución de incentivos.
- Subapartado 2.3: Situación actual de la gestión de objetivos y retribución de incentivos.
- Subapartado 2.4: Elementos de la gestión de objetivos y retribución de incentivos.

2.2 Definición de gestión de objetivos y retribución de incentivos

La gestión de objetivos y retribución de incentivos es un concepto ambiguo desde el punto de vista científico, y por tanto, difícil de definir. No hay un consenso en cuanto a su significado, y existe una cierta confusión en torno a su ámbito de utilidad. Incluso llega a existir una cierta discordancia en cuanto a su nombre, ya que también puede ser referido como retribución variable.

Según la Universidad a distancia de Madrid (UDIMA), la gestión de objetivos y retribución de incentivos “*es una herramienta de dirección cuyo objetivo es impulsar la estrategia de negocio de la compañía*”. Además, se trata de “*un sistema que recompensa resultados, y no sólo esfuerzo, y que integra [...] la aportación de las personas a la consecución de los objetivos de la compañía*”.

La UDIMA, además, enumera algunas las ventajas que aporta la utilización de un SGORI:

- Ventajas para la compañía:
 - Impulsa la consecución de los objetivos de negocio.
 - Contribuye a mejorar la competitividad.
 - Fortalece la visión de conjunto y de equipo.
 - Permite modular los costes al ser un coste variable y delimitado, no consolidable.
- Ventajas para el empleado:
 - Reconoce la importancia que tiene su actuación para el logro de los objetivos.
 - Premia y recompensa la actuación individual y de equipo.
 - Incrementa la satisfacción al vincular de forma tangible la retribución y el logro de objetivos.
 - Permite incrementar la competitividad salarial del empleado.

2.3 Situación actual de la gestión de objetivos y retribución de incentivos

Actualmente puede observarse una clara tendencia por la cual el número de empresas que cuentan con un SGORI va en aumento. Además, compañías que ya cuentan con un SGORI están revisando sus modelos de retribución variable, aumentando el peso de indicadores de grupo o colectivos frente a los individuales, con objeto de impulsar la estrategia de negocio.

Dada la situación actual de crisis económica, los SGORI recobran importancia y protagonismo, primando la productividad, y siendo ampliado el número de colectivos a los que son aplicados.

2.4 Elementos de la gestión de objetivos y retribución de incentivos

De forma general, podemos establecer en la actualidad seis elementos básicos en la gestión de objetivos y retribución de incentivos (Figura 1), presentes en casi todas las organizaciones que incorporan estos sistemas:

- Indicadores.
- Objetivos.
- Elegibilidad.
- Gestión/Administración.
- Condicionantes.
- Ponderación.

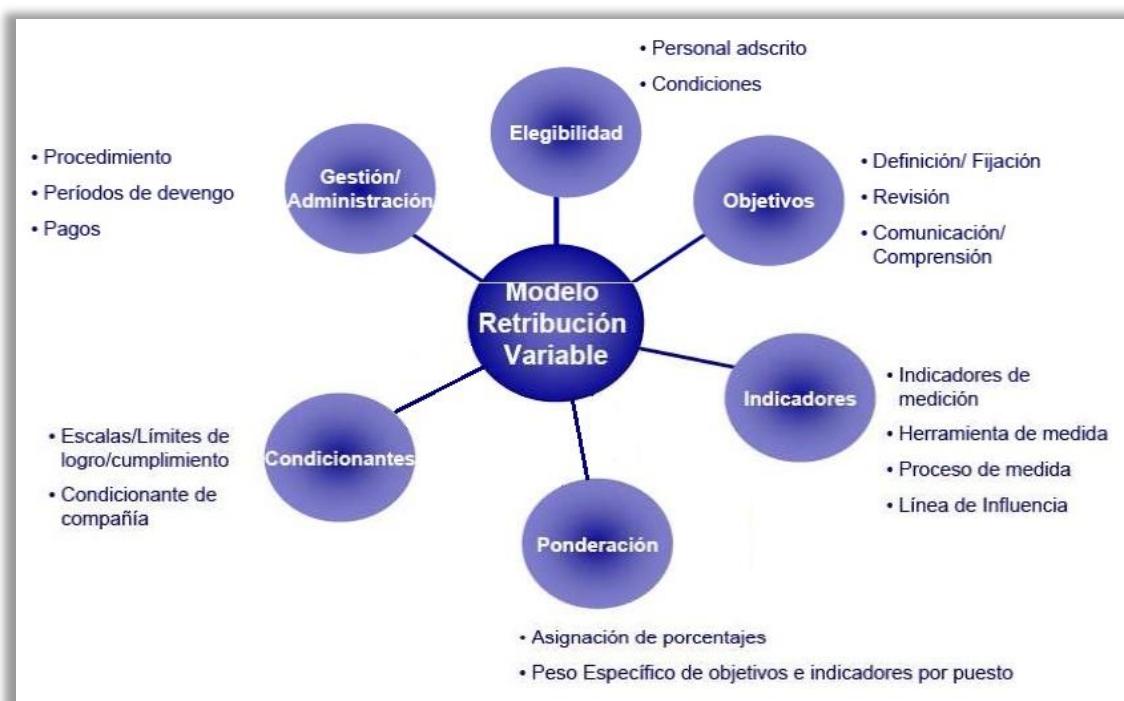


Figura 1. Elementos de un SGORI

2.4.1 Indicadores

Los indicadores constituyen la herramienta básica de medición. Un indicador hace referencia a aquello que va a ser medido, y puede dar información acerca de las unidades y/o los métodos usados para las mediciones, así como el dominio al que son aplicados (pueden influir sobre un grupo o colectivo, o sobre un individuo concreto).

2.4.2 Objetivos

En cualquier SGORI es necesario definir y fijar unos objetivos, asociados a los datos indicadores, que establezcan cuáles van a ser los valores reales mínimos a conseguir, a partir de los cuales se tendrá en cuenta la retribución de incentivos para los miembros de la compañía implicados. Estos objetivos deberán estar en continua revisión, para así evitar valores inalcanzables o poco cercanos a la realidad. Además, los valores objetivos establecidos deberán ser comunicados a las personas implicadas, haciendo especial hincapié en su comprensión, permitiendo que conozcan y estimen su propio esfuerzo a realizar para poder obtener la retribución adicional correspondiente.

2.4.3 Elegibilidad

Hace referencia al personal adscrito a cada indicador y a cada objetivo, así como a las condiciones impuestas que deben ser cumplidas para poder optar a las retribuciones.

2.4.4 Gestión/Administración

Para que un SGORI pueda funcionar tal y como es planteado, es necesaria una buena administración del mismo, sobre todo en los aspectos relacionados con los pagos de las retribuciones y los períodos de devengo (pagos a tiempo, y acorde a lo previamente establecido y posteriormente conseguido, períodos claramente definidos, acotados y no solapados, etc.).

2.4.5 Condicionantes

Este elemento hace referencia a las escalas y los límites de logro y cumplimiento de objetivos, para los cuales los miembros implicados serán retribuidos mediante los incentivos correspondientes, es decir, las condiciones que establece la compañía para que sus personas opten a las retribuciones salariales.

2.4.6 Ponderación

Es posible que el SGORI no afecte de igual modo a todos los miembros implicados de la compañía. Para ello, se asigna un porcentaje o peso específico de indicadores y objetivos por persona y/o puesto, indicando en qué grado cada miembro o grupo es afectado por la gestión de objetivos y retribución de incentivos.

3. Sistemas de Gestión de Objetivos y Retribución de Incentivos

3.1 Introducción

En este apartado se muestran las características de diferentes alternativas comerciales de SGORI, siempre desde un punto de vista generalista. Un estudio más profundo escapa al propósito de esta fase del proyecto.

3.2 Sistemas actuales

3.2.1 Cezanne HR

Cezanne HR es una compañía fundada en el año 2000 que realiza la gestión y ofrece soluciones online de Recursos Humanos, con oficinas en Reino Unido e Italia.

Cuenta con más de 20 años de experiencia en el sector, y fue creada para gestionar y desarrollar la línea de producto *Cezanne OnDemand* tras la venta de *Cezanne Software*, la cual se centraba en medianas y grandes empresas [Ref. Web 2].

Centrándonos en sus productos software de RRHH, *Cezanne HR* ha desarrollado un sistema informatizado para realizar la gestión del desempeño que proporciona las siguientes ventajas [Ref. Web 3]:

- Simplicidad de la gestión del desempeño.
- Posibilidad de diseñar formatos de evaluación personalizados y propios.
- Distribución de formatos de rendimiento y evaluación de forma sencilla.
- Conexión e interacción entre empleados y sus managers.
- Monitorización de los procesos de evaluación en tiempo real.
- Simplicidad en los informes de gestión del desempeño.

Ejemplos del sistema de gestión del desempeño desarrollado por *Cezanne HR* pueden verse en las figuras 2 y 3:

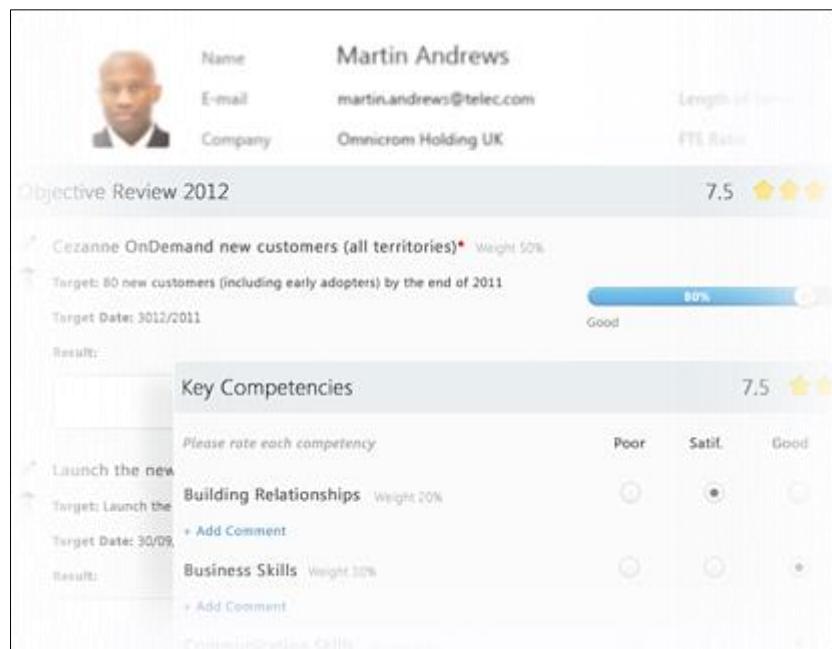


Figura 2. Ejemplo del sistema de gestión del desempeño de Cezanne HR

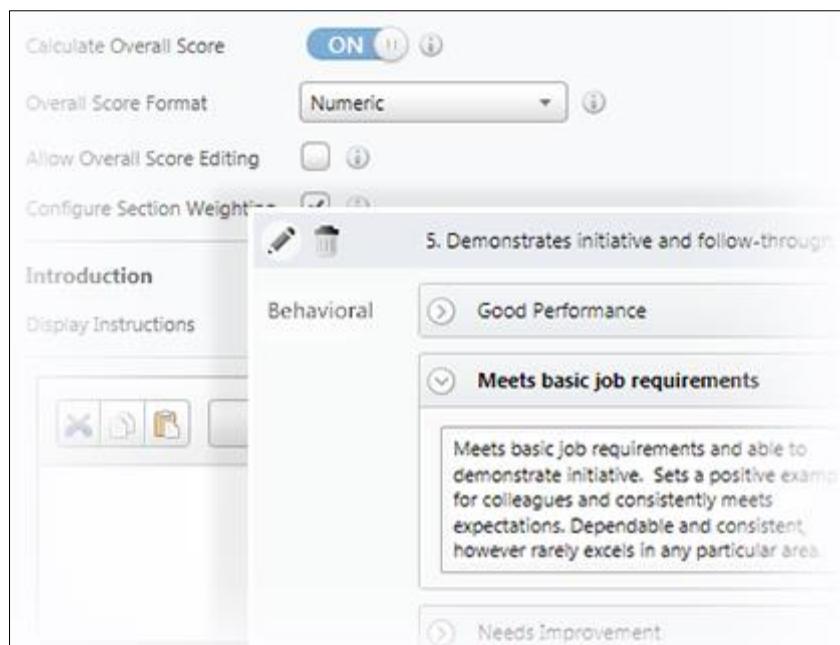


Figura 3. Ejemplo del sistema de gestión del desempeño de Cezanne HR

3.2.2 *IDARVE Software*

Según su página web [Ref. Web 4], *IDARVE* es un paquete informático que permite calcular la cuantía de retribución variable assignable a cada trabajador con derecho a percibirla, para un período de tiempo determinado. Es un software autoejecutable que no requiere ninguna configuración informática específica.

IDARVE software funciona a partir de la cuantía total que se asigna para la retribución variable en un período determinado, la cual se distribuye en función de tres variables esenciales:

- El desempeño de las unidades a las que pertenece el trabajador.
- El desempeño individual del trabajador.
- El salario del trabajador.

En cualquier caso, el programa permite la parametrización de las variables de manera que se adapte a las preferencias específicas de cada empresa. Además, permite realizar la asignación de la retribución variable a partir de una cuantía determinada (método de Totales) o en función del rendimiento general, tomando la cuantía introducida como el máximo a obtener si el rendimiento de unidades y personal fuese siempre óptimo (método de Máximos).

Lamentablemente, la página web de *IDARVE* no ofrece imágenes de ejemplos de la aplicación en funcionamiento. Sin embargo, sí que ofrece un manual de uso, donde se incluyen algunos conceptos básicos a tener en consideración antes de utilizar el software, además de una breve descripción del menú principal de la aplicación [Ref. Web 5].

3.2.3 *Ceinsa*

Ceinsa [Ref. Web 6] es una consultoría de RRHH con sede en Barcelona y 20 años de experiencia en los ámbitos de gestión de compensación y de talento. Forman parte del grupo *RH Asesores* [Ref. Web 7] y se dedican al desarrollo e implantación de soluciones tecnológicas propias para la gestión del capital humano.

Su software, *CeinsaEXPERT*, es fácilmente integrable con cualquier software de Nóminas y ERP del mercado. Además, incluye diversos modelos predefinidos y suministra herramientas de generación que permiten al usuario crear sus propios documentos. Las áreas de gestión de *CeinsaEXPERT* son:

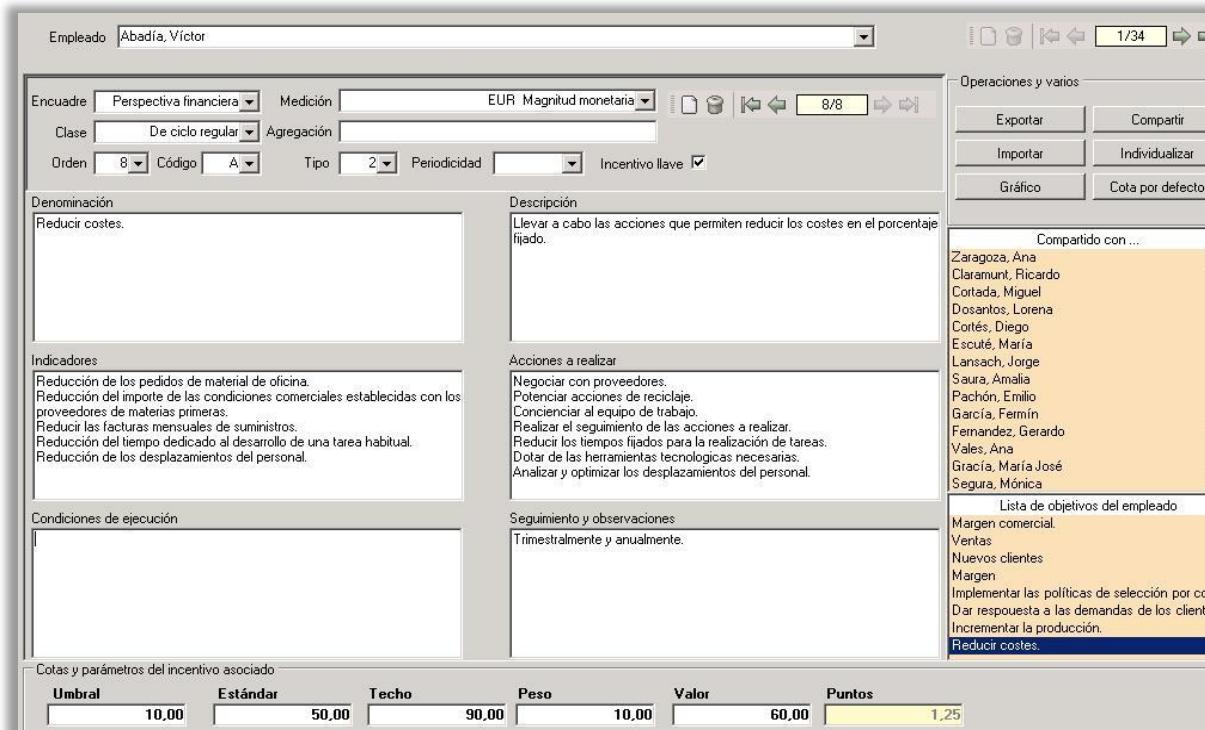
- Gestión de Talento.
- Gestión de la Compensación.

Centrándonos en los módulos que ofrece el software de *Ceinsa*, el módulo de Gestión de Objetivos e Incentivos ofrece [Ref. Web 8]:

- Soporte para vinculación de los objetivos operativos con el plan estratégico de la empresa.
- Soporte para la definición, descripción y comunicación de objetivos.
- Gestión de objetivos individuales, compartidos y colectivos.
- Gestión de curvas de incentivo y su representación gráfica.
- Gestión de matrices de incentivo.
- Cálculo de incentivos ligados a objetivos, con posible integración de resultados de empresa, de equipo e individuales.
- Permite incorporar múltiples modelos de retribución variable (vinculados o no a objetivos), creados y documentados por *Ceinsa*.

- Puede también personalizarse para gestionar sistemas de retribución variable ya existentes en la compañía.
- Biblioteca con modelos estándar de objetivos por áreas profesionales.

A continuación, puede verse un ejemplo de la gestión interna de los objetivos mediante el software *CeinsaEXPERT* (Figura 4) y un ejemplo de representación gráfica de la consecución de objetivos, obtenida por el programa de *Ceinsa* (Figura 5).



The screenshot shows the CeinsaEXPERT software interface for managing objectives. The top bar displays the employee name 'Abadía, Víctor'. The main window is divided into several sections:

- Encuadre:** Perspectiva financiera, Medición: EUR Magnitud monetaria, Clase: De ciclo regular, Agregación: Orden: 8, Código: A, Tipo: 2, Periodicidad: 1, Incentivo llave: checked.
- Denominación:** Reducir costes.
- Descripción:** Llevar a cabo las acciones que permiten reducir los costes en el porcentaje fijado.
- Indicadores:** Reducción de los pedidos de material de oficina, Reducción del importe de las condiciones comerciales establecidas con los proveedores de materias primas, Reducir las facturas mensuales de suministros, Reducción del tiempo dedicado al desarrollo de una tarea habitual, Reducción de los desplazamientos del personal.
- Acciones a realizar:** Negociar con proveedores, Potenciar acciones de reciclaje, Concienciar al equipo de trabajo, Realizar el seguimiento de las acciones a realizar, Reducir los tiempos fijados para la realización de tareas, Dolar de las herramientas tecnológicas necesarias, Analizar y optimizar los desplazamientos del personal.
- Condiciones de ejecución:**
- Seguimiento y observaciones:** Trimestralmente y anualmente.
- Operaciones y varios:** Exportar, Compartir, Importar, Individualizar, Gráfico, Cota por defecto.
- Compartido con ...:** Zaragoza, Ana; Claramunt, Ricardo; Cortada, Miguel; Dosantos, Lorena; Cortés, Diego; Escuté, María; Lansach, Jorge; Saura, Amalia; Pachón, Emilio; García, Fermín; Fernandez, Gerardo; Vales, Ana; Gracia, María José; Segura, Mónica.
- Lista de objetivos del empleado:** Margen comercial, Ventas, Nuevos clientes, Margen, Implementar las políticas de selección por co, Dar respuesta a las demandas de los client, Incrementar la producción, Reducir costes.
- Cotas y parámetros del incentivo asociado:**

Umbral	Estandar	Techo	Peso	Valor	Puntos
10,00	50,00	90,00	10,00	60,00	1,25

Figura 4. Gestión interna de los objetivos mediante *CeinsaEXPERT*

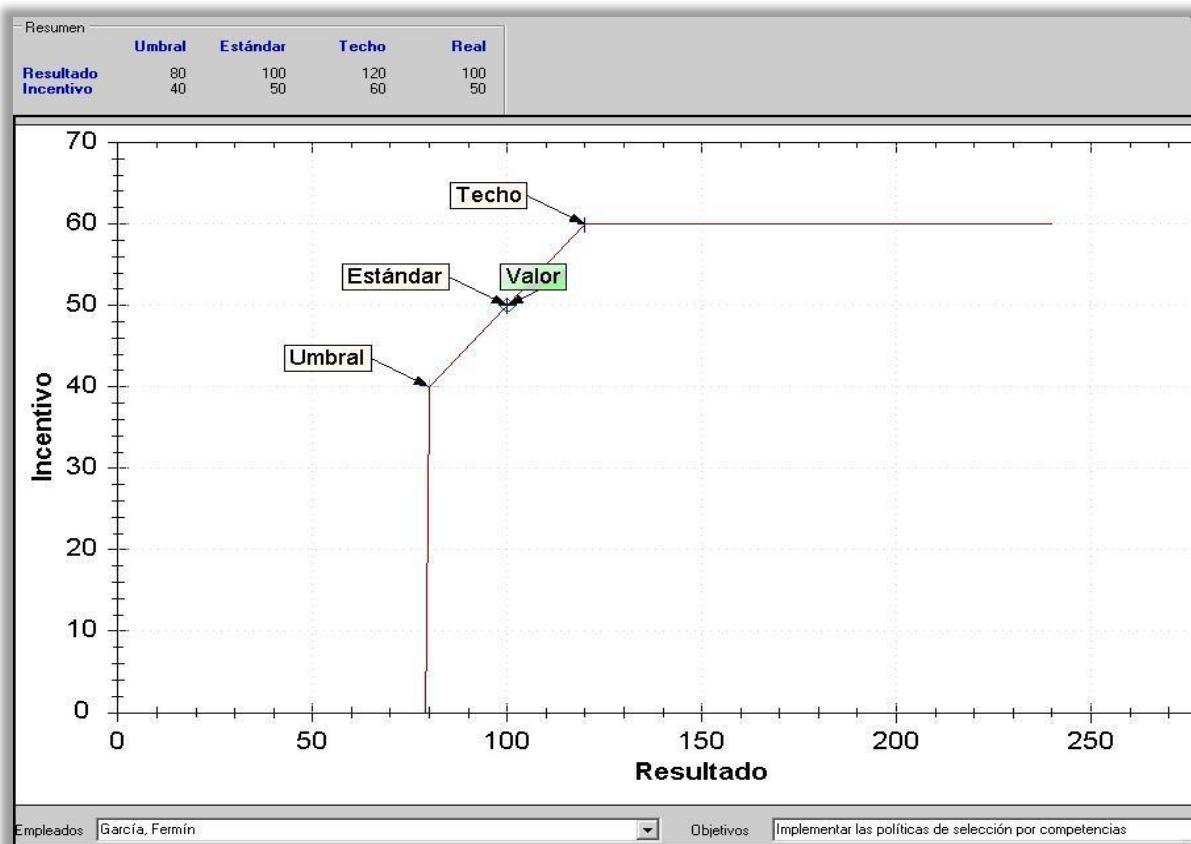


Figura 5. Representación gráfica de la consecución de objetivos mediante CeinsaEXPERT

4. Conclusiones

El presente estudio de mercado nos ha permitido comprender mucho mejor el problema al que nos enfrentamos.

Gracias al análisis plasmado en el segundo apartado de este documento, hemos podido constatar que, en la actualidad, la retribución de incentivos es considerada un concepto clave, constituyendo un elemento estratégico y una ventaja competitiva de la propia organización.

El análisis de las funciones básicas que todo sistema de gestión de objetivos y de retribución de incentivos debe desempeñar, nos servirá como una guía de los elementos que nuestro sistema debe poder gestionar.

El tercer punto de este estudio, en el que se muestran distintas alternativas de aplicaciones comerciales, nos ha servido para obtener ideas acerca de diferentes funcionalidades y características (interfaz gráfico, utilidades, etc.) que pueden servir de modelo para determinadas partes del proyecto a desarrollar, y que serán definidas en fases más avanzadas de este proyecto.

5. Bibliografía

5.1 Referencias

[SOMI, 1997] Solé, F. y Mirabet, M. (1997). "Guía para la formación en la empresa". Madrid: Civitas.

5.2 Referencias web

- [Ref. Web 1] <http://www.slideshare.net/RGblogger/retribucion-variable>
- [Ref. Web 2] <http://cezannehr.com/es/compania/>
- [Ref. Web 3] <http://cezannehr.com/es/software-rrhh/gestion-del-desempeno/>
- [Ref. Web 4] <http://www.idarve.com/que-es-el-idarve/>
- [Ref. Web 5] <http://www.idarve.com/category/manual-de-uso/>
- [Ref. Web 6] <http://www.ceinsa.com/index.php/perfilempresa.html>
- [Ref. Web 7] <http://www.gruporh.com/>
- [Ref. Web 8] <http://www.ceinsa.com/index.php/ceinsaexpert/introduccion/modceinsexp/m3.html>

Sistema de Gestión de Objetivos y Retribución de Incentivos

Modelo de Negocio

Versión 1.3

Publicado el 04/03/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
20/02/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
22/02/2013	1.1	Modificación del documento	Santiago Pelegrín Costea
25/02/2013	1.2	Revisión del documento	Fernando Cortés Franco
04/03/2013	1.3	Corrección del documento	Santiago Pelegrín Costea

Índice

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.	Introducción al modelo de negocio.....	7
3.	Aproximación al modelo de negocio.....	8
4.	Modelo de casos de uso del negocio.....	10
4.1	Diagrama de casos de uso del negocio.....	10
4.2	Caso de uso del negocio “Gestión de indicadores”	11
4.3	Caso de uso del negocio “Gestión de objetivos y datos reales”	12
4.4	Caso de uso del negocio “Gestión de remuneraciones e incentivos”	12
5.	Bibliografía	13
5.1	Referencias.....	13
5.2	Referencias web.....	13

1. Introducción

1.1 Propósito del documento

En el presente escrito se muestra el análisis del modelo de negocio en el cual se enmarca el proyecto a desarrollar, el Sistema de Gestión de Objetivos y Retribución de Incentivos (SGORI). La especificación del modelo de negocio, según la metodología RUP de desarrollo de 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 del Sistema de Gestión de Objetivos y Retribución de Incentivos. 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

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.
- UML: Unified Modeling Language.
- RUP: Rational Unified Process.

1.4 Definiciones

- UML: Lenguaje de modelado de sistemas de software desarrollado por *Rational* y de uso muy extendido [Ref. Web 1].
- 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 [Ref. Web 2].

1.5 Referencias

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

- ESTMER: Estudio de mercado en el cual se analiza la situación actual de las técnicas de gestión de objetivos y de retribución de incentivos, presentando diferentes alternativas existentes.

1.6 Resumen

En el presente documento se muestra el análisis del modelo de negocio en el cual se define el proyecto a desarrollar (SGORI). 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 del SGORI, 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 el desarrollo de esta fase del proyecto.

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 desarrollará el modelo de casos de uso del negocio, el cual presenta el sistema desde la perspectiva de su uso, y esquematiza cómo 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.

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 general del Sistema de Gestión de Objetivos y Retribución de Incentivos, para facilitar una mejor comprensión del mismo (Figura 1).



Figura 1. Visión aproximada del modelo de negocio.

En esta figura se puede ver cómo el SGORI sirve como herramienta para facilitar la gestión de los objetivos y de la retribución variable de incentivos que la componen.

Podemos distinguir tres tipos de gestores, desde el punto de vista de sus funciones y de la información que precisan manejar:

- Gestor de indicadores:
 - Sus funciones se basan en la gestión de la información de los indicadores que integran el sistema.
 - Maneja los datos de los indicadores, su información asociada, sus características, las personas relacionadas y los privilegios sobre los mismos.
- Gestor de objetivos y datos reales:
 - Sus funciones principales comprenden la gestión de los valores, tanto objetivos como reales conseguidos, asociados a los indicadores.
 - Maneja datos referentes a los objetivos establecidos para los indicadores, así como los valores reales que posteriormente son logrados y que servirán para calcular la remuneración final conseguida.
- Gestor de remuneraciones e incentivos:
 - Sus funciones se basan en la gestión de las remuneraciones o retribuciones asociadas a los indicadores y objetivos, así como de los incentivos a percibir por las personas.
 - Maneja datos referentes a las retribuciones que serán otorgadas a aquellas personas que consigan, para un indicador determinado, alcanzar el objetivo o uno de los objetivos previamente establecidos.

El SGORI será la herramienta que usen para el desempeño de sus funciones.

4. Modelo de casos de uso del negocio

4.1 Diagrama de casos de uso del negocio

Como se ha comentado previamente, el modelo de casos de uso del negocio describe los procesos de negocio que tienen lugar en el SGORI. El modelo de casos de uso vendrá dado por el siguiente diagrama de casos de uso (Figura 2):

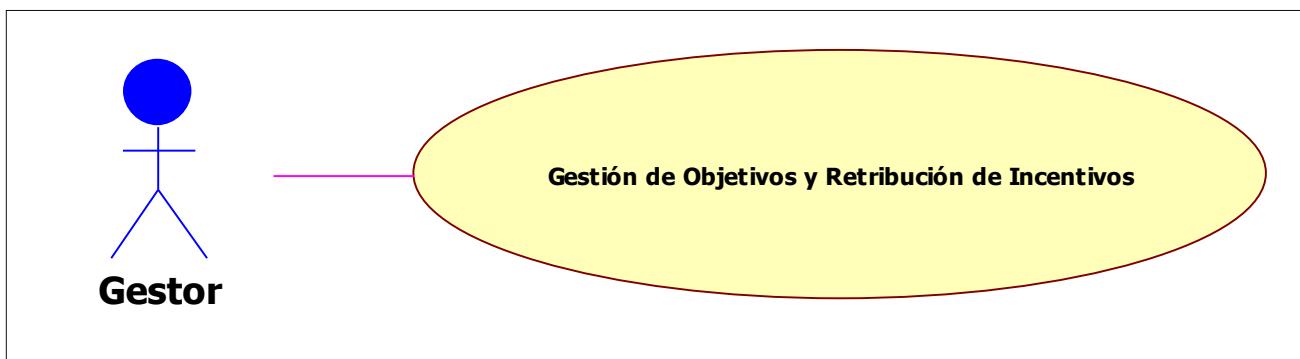


Figura 2: Modelo de casos de uso del negocio (nivel 0)

A continuación, mostramos el mismo diagrama general del sistema, con un nivel más de detalle (Figura 3):

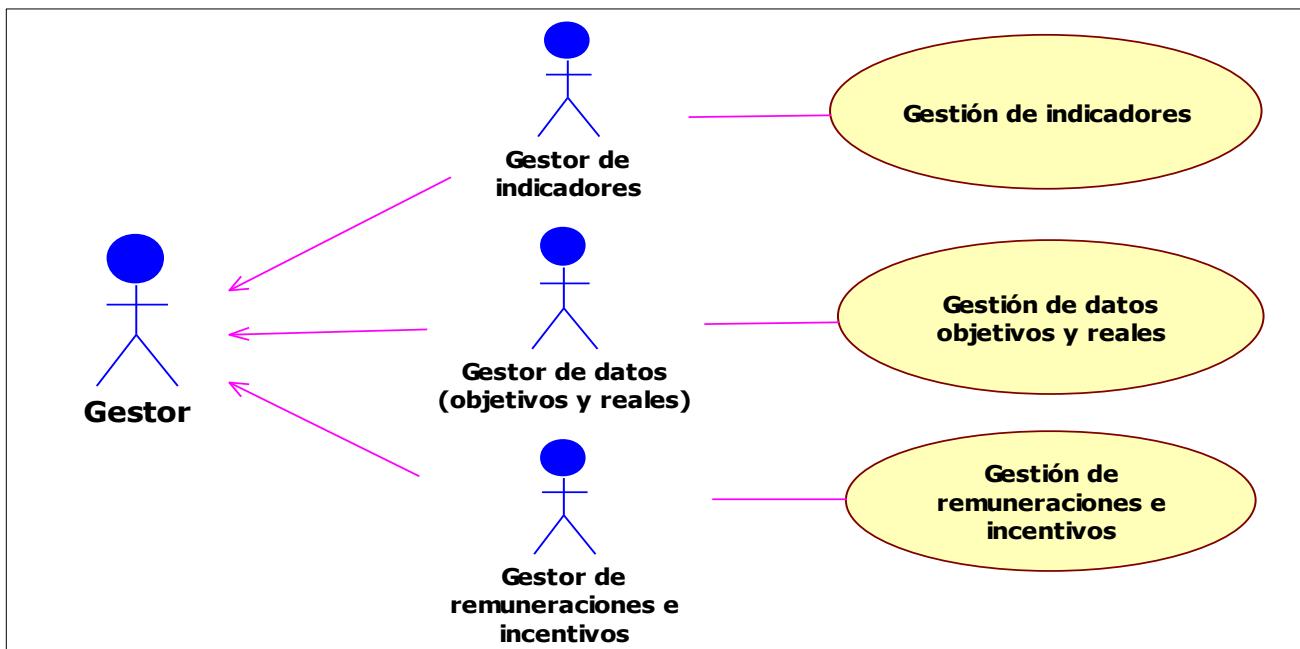


Figura 3: Modelo de casos de uso del negocio (nivel 1)

La anterior figura muestra la subdivisión del sistema en tres procesos diferenciados, como ya se ha comentado previamente. En las subsiguientes secciones se procederá al análisis individualizado de los casos de uso de cada uno de estos tres subprocessos. Cuando la complejidad lo requiera, se procederá a explosionar el diagrama de caso de uso, para mostrar niveles inferiores (y por tanto más detallados) del mismo.

4.2 Caso de uso del negocio “Gestión de indicadores”

A continuación se muestra el caso de uso del negocio “Gestión de indicadores” (Figura 4):

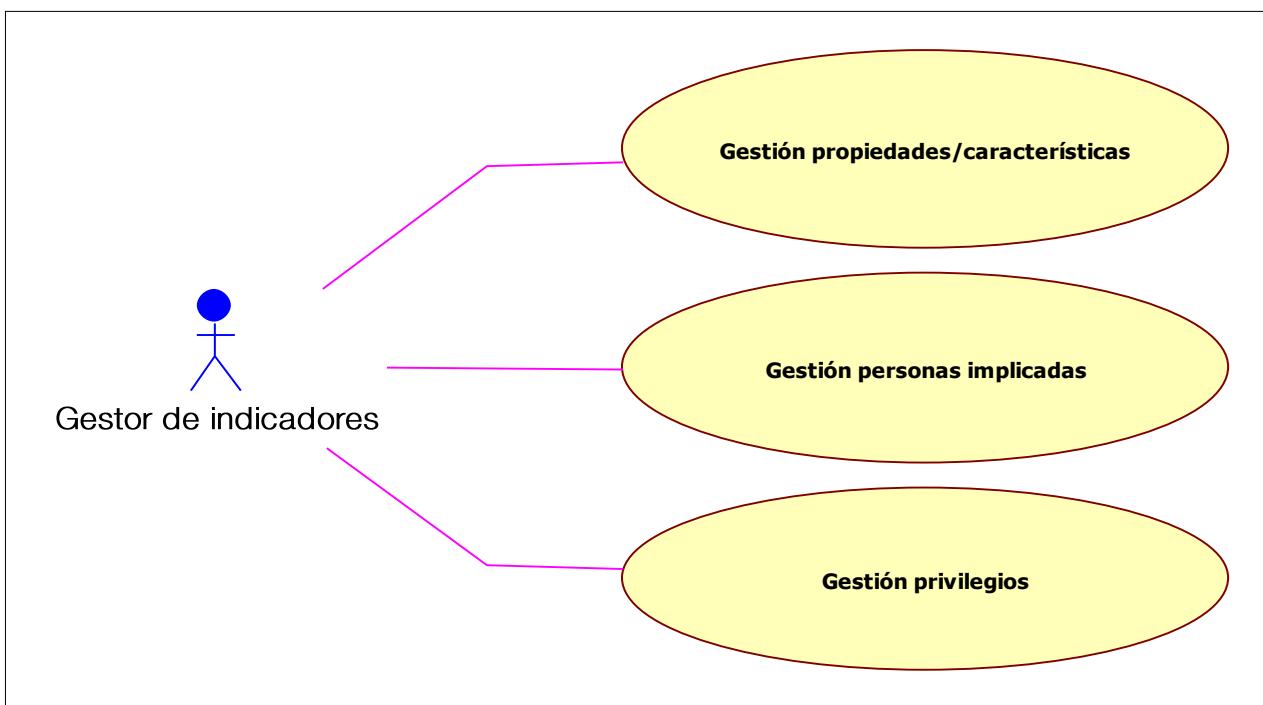


Figura 4: Caso de uso de “Gestión de indicadores”

Como se muestra en la figura previa, el gestor de indicadores es el encargado de gestionar los datos de los indicadores relevantes para la compañía. Dicha gestión de datos se puede dividir asimismo en tres subsistemas diferenciados:

- Gestión de las propiedades y características de los indicadores: comprende datos tales como el nombre, código, persona responsable, tipo, comportamiento (colectivo o individual), área o grupo al que pertenece el indicador, etc.
- Gestión de las personas implicadas: relaciona los indicadores con las personas de la organización, haciendo que se vean afectadas por los mismos.
- Gestión de los privilegios: hace referencia a las personas de la organización a las que les son asignados permisos específicos para la gestión de los datos de objetivo y valores reales asociados a los indicadores.

4.3 Caso de uso del negocio “Gestión de objetivos y datos reales”

A continuación se muestra el caso de uso de “Gestión de objetivos y datos reales” (Figura 5):

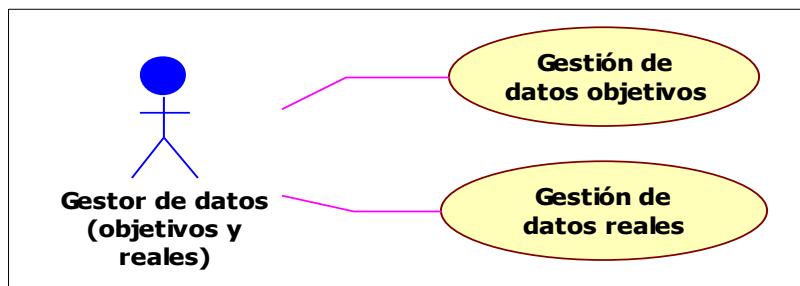


Figura 5: Caso de uso de “Gestión de objetivos y datos reales”

Como se indicó en el apartado anterior, el gestor de indicadores podrá asociar privilegios específicos a las personas de la organización para que sean capaces de gestionar los datos objetivo y reales de los indicadores. La gestión llevada a cabo por esta persona comprende:

- Determinación de los objetivos: identifica y gestiona los objetivos de cada indicador, haciendo lo posible para establecerlos en valores realmente alcanzables y comprensibles por las personas afectadas, las cuales, además, deberán estar informadas de dichos datos.
- Gestión de los valores reales: recopila y gestiona los datos reales conseguidos por los miembros de la organización para cada indicador, haciendo posible el posterior cálculo de la remuneración.

4.4 Caso de uso del negocio “Gestión de remuneraciones e incentivos”

A continuación se muestra el caso de uso de “Gestión de remuneraciones e incentivos” (Figura 6):

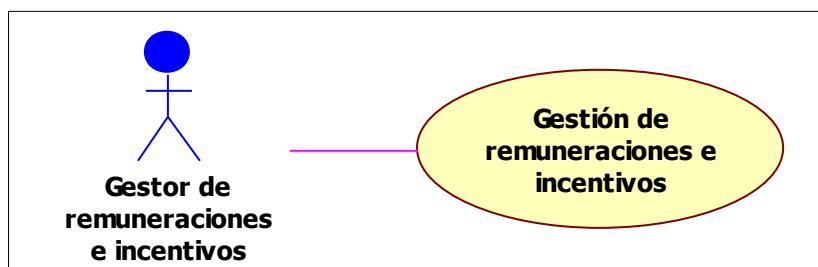


Figura 6: Caso de uso de “Gestión de remuneraciones e incentivos”

El gestor de remuneraciones e incentivos se encarga de establecer valores de retribución asociados a los indicadores, así como de gestionar los incentivos que los miembros de la organización implicados en el sistema percibirán por el cumplimiento de los objetivos que fueron previamente establecidos.

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.uml.org>

[Ref. Web 2] <http://www-01.ibm.com/software/rational/rup/>

Sistema de Gestión de Objetivos y Retribución de Incentivos

Especificación de Requisitos

Versión 1.4

Publicado el 04/03/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
25/02/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
27/02/2013	1.1	Modificación del documento	Santiago Pelegrín Costea
01/03/2013	1.2	Modificación del documento	Santiago Pelegrín Costea
02/03/2013	1.3	Revisión del documento	Fernando Cortés Franco
04/03/2013	1.4	Corrección del documento	Santiago Pelegrín Costea

Índice

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 general	7
2.1 Funciones generales del sistema	7
2.2 Características de los usuarios	7
3. Requisitos del sistema.....	8
3.1 Introducción.....	8
3.2 Requisitos funcionales.....	8
3.3 Requisitos no funcionales	13
4. Bibliografía	15
4.1 Referencias.....	15

1. Introducción

1.1 Propósito del documento

El presente documento pretende identificar y describir los requisitos del sistema a desarrollar. Por requisitos nos referimos a características que deben incluirse en el nuevo sistema.

Una buena especificación de requisitos debe ser [IEEE SRS]:

- Correcta
- Inequívoca
- Completa
- Consistente
- Comprobable
- Modificable
- Identifiable

La especificación de requisitos guiará el desarrollo del resto de fases del proyecto.

1.2 Alcance del documento

El alcance de este documento comprende todo el período temporal del proyecto, ya que su contenido puede ser modificado en cualquier momento, a medida que las especificaciones y necesidades del SGORI cambien.

1.3 Acrónimos

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.
- GUI: Graphical User Interface (Interfaz Gráfico de Usuario).
- DPT: Descripción de Puesto de Trabajo.
- RF: Requisito funcional.
- RNF: Requisito no funcional.

1.4 Definiciones

- Graphical User Interface (GUI): método para facilitar la interacción del usuario con la aplicación software a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas...) además de texto.
- Indicador: herramienta básica del SGORI que permite medir la consecución o evolución de los objetivos de una organización basados principalmente en los aspectos clave del negocio. Pueden influir y ser aplicados sobre un grupo o colectivo, o sobre individuos en concreto.
- Barrera de entrada: indicador que establece el límite a partir del cual se tiene derecho al devengo del incentivo por la superación de los objetivos asignados a los indicadores.

1.5 Referencias

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

- MODNEG: Documento de descripción del modelo de negocio.

1.6 Resumen

El presente documento pretende reflejar el análisis sistemático de los requisitos principales que debe cumplir el sistema a desarrollar. Se compone de 4 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 muestra una descripción general de las funciones del producto y de la naturaleza de sus usuarios.
- Apartado 3: Se muestra la relación de los requisitos del sistema a desarrollar, su descripción y su nivel de criticidad, diferenciando los requisitos funcionales de los no funcionales.
- Apartado 4: Bibliografía utilizada a lo largo de esta fase del proyecto.

2. Descripción general

2.1 Funciones generales del sistema

Como ya se describió en el documento del Modelo de Negocio (MODNEG), el SGORI es una herramienta orientada a facilitar la gestión de los objetivos de una organización y de la retribución de incentivos que la integran. Mediante el uso del SGORI se llevará a cabo la administración de indicadores y la gestión de la retribución variable.

2.2 Características de los usuarios

En los procesos que incluye la gestión de objetivos y de retribuciones en el SGORI, se pueden distinguir los siguientes roles de usuarios:

- Gestor de indicadores: se encarga de la gestión de la información de los indicadores que integran el sistema (información asociada, características, personas relacionadas y privilegios sobre los mismos).
- Gestor de objetivos y datos reales: gestiona los valores, tanto objetivos como reales conseguidos, asociados a los indicadores.
- Gestor de compensaciones e incentivos: sus funciones se basan en la gestión de las compensaciones o retribuciones asociadas a los indicadores, así como de los incentivos a recibir por las personas implicadas.

El SGORI deberá asistir en el correcto desempeño de las tareas asociadas a cada uno de estos usuarios.

La aplicación estará destinada a pequeñas y medianas empresas de cualquier sector de actividad, por lo que será necesario tener en cuenta que los usuarios de la aplicación no tienen por qué estar familiarizados con el uso de las nuevas tecnologías en la gestión de su organización.

3. Requisitos del sistema

3.1 Introducción

En esta sección se van a detallar los diferentes requisitos que el sistema a desarrollar debe satisfacer.

Los requisitos van a estar divididos en dos grupos:

- Requisitos funcionales. Describe las funcionalidades del sistema, referidas a:
 - Funciones de actualización de datos.
 - Funciones de consulta.
 - Informes proporcionados.
 - Datos manejados.
 - Interacción con otros sistemas.
- Requisitos no funcionales. Describen las facilidades que debe proporcionar el sistema, referidas a:
 - Rendimiento.
 - Volumen y tamaño de datos.
 - Frecuencia de tratamiento.
 - Requisitos de seguridad:
 - Control de accesos.
 - Procedimientos de copias de respaldo y recuperación.
 - Integridad de la información.
 - Requisitos especiales de comunicaciones.
 - Requisitos organizacionales:
 - Directrices y técnicas de gestión.
 - Requisitos generales a cumplir en cuanto a necesidades futuras de información.
 - Tendencias de evolución de la organización.

3.2 Requisitos funcionales

Nombre	RF - 01: Gestión de indicadores y barreras de entrada
Descripción	El SGORI permitirá crear nuevos indicadores, cambiar su estado a barreras de entrada y eliminarlos definitivamente del sistema
Nivel de criticidad	Alto
Dependencias	-

Nombre	RF - 02: Gestión de la información asociada a los indicadores
Descripción	<p>El SGORI permitirá almacenar y modificar la información asociada a cada indicador.</p> <p>Esa información incluirá:</p> <ul style="list-style-type: none"> • Nombre • Código • Persona responsable • Área en la que se ubica • Tipo (indicador o barrera de entrada) • Referencia de pago (por cumplimiento de objetivo o por unidad) • Comportamiento (individual o colectivo)
Nivel de criticidad	Alto
Dependencias	La gestión de la información de los indicadores dependerá de la gestión de indicadores (RF - 01).
Nombre	RF - 03: Gestión de las personas afectadas por los indicadores
Descripción	El SGORI deberá almacenar información acerca de los miembros de la organización implicados en cada indicador. El sistema permitirá añadir y eliminar personas implicadas, así como asignar una ponderación porcentual que indique el peso que el indicador supone sobre cada persona.
Nivel de criticidad	Alto
Dependencias	La gestión de las personas afectadas dependerá de la gestión de indicadores (RF - 01).
Nombre	RF - 04: Gestión de privilegios sobre los indicadores
Descripción	El SGORI permitirá almacenar información acerca de los miembros de la organización con privilegios sobre los indicadores, y que serán capaces de gestionar los valores objetivo y reales conseguidos en cada indicador por las personas implicadas.
Nivel de criticidad	Alto
Dependencias	La gestión de los privilegios sobre los indicadores dependerá de la gestión de indicadores (RF - 01).

Nombre	RF - 05: Gestión de indicadores afectados por barreras de entrada
Descripción	El SGORI permitirá almacenar información acerca de los indicadores de tipo barrera de entrada que afectan a otros indicadores. El sistema deberá permitir gestionar una lista de barreras de entrada que afectan a un indicador, ofreciendo la posibilidad de añadir o eliminar elementos de la misma.
Nivel de criticidad	Alto
Dependencias	La gestión de los indicadores afectados por barreras de entrada dependerá de la gestión de indicadores (RF - 01).
Nombre	RF - 06: Gestión de valores reales y objetivo asociados a indicadores
Descripción	El SGORI permitirá gestionar, únicamente a aquellas personas que posean suficientes privilegios, los valores objetivo y reales conseguidos por los implicados en cada indicador. El sistema deberá permitir introducir nuevos valores, así como modificar y eliminar los existentes.
Nivel de criticidad	Alto
Dependencias	La gestión de valores reales y objetivo está relacionada con la gestión de indicadores (RF - 01), la gestión de privilegios sobre los indicadores (RF - 04) y la gestión de personas afectadas por los datos de los indicadores (RF - 03).
Nombre	RF - 07: Listado de datos de indicadores.
Descripción	El SGORI deberá mostrar una lista que incluya la información asociada a los indicadores, como el nombre, el código, el área al que pertenecen, el tipo y los objetivos y valores reales de cada persona implicada. El sistema permitirá el filtrado de los datos contenidos en la lista, pudiendo cambiar la vista mostrada según el tipo (indicadores o barreras de entrada), el período anual en el que son establecidos los objetivos y valores reales y el área al que pertenecen los indicadores.
Nivel de criticidad	Medio
Dependencias	El listado de indicadores dependerá de la gestión de indicadores (RF - 01), la gestión de información de los indicadores (RF - 02) y la gestión de valores reales y objetivo asociados a los indicadores (RF - 06).

Nombre	RF - 08: Gestión de retribuciones
Descripción	El SGORI permitirá gestionar retribuciones, en valor monetario, ofreciendo la posibilidad de introducir, modificar y eliminar las retribuciones ya existentes. Cada retribución gestionada por el sistema estará asociada a un indicador y a un valor escalado o intervalo de consecución. Las retribuciones, además, irán asociadas a un período de tiempo determinado, permitiendo establecer distintas retribuciones para un mismo indicador en distintos períodos.
Nivel de criticidad	Alto
Dependencias	La gestión de retribuciones dependerá de la gestión de indicadores (RF - 01).
Nombre	RF - 09: Gestión de penalizaciones
Descripción	El SGORI permitirá gestionar penalizaciones en valor porcentual, ofreciendo la posibilidad de introducir, modificar y eliminar las penalizaciones ya existentes. Cada penalización gestionada por el sistema estará asociada a un indicador de tipo barrera de entrada y a un período de tiempo determinado, permitiendo establecer distintas penalizaciones para una misma barrera de entrada en distintos períodos.
Nivel de criticidad	Alto
Dependencias	La gestión de penalizaciones está relacionada con la gestión de indicadores (RF - 01).
Nombre	RF - 10: Cálculo de incentivos
Descripción	El SGORI realizará el cálculo automático de los incentivos a recibir por cada miembro de la organización implicado en algún indicador. El sistema permitirá visualizar los incentivos correspondientes a cada persona de un modo general, desglosado por meses, o de un modo detallado, en el que se mostrará la procedencia de cada parte del incentivo, así como las penalizaciones aplicadas a cada una de dichas partes.
Nivel de criticidad	Alto
Dependencias	El cálculo de incentivos está relacionado con los siguientes requisitos funcionales: RF - 01, RF - 02, RF - 03, RF - 05, RF - 06, RF - 08, RF - 09.

Nombre	RF - 11: Revisión y cierre de incentivos
Descripción	El sistema permitirá, a los usuarios correspondientes, revisar los incentivos a recibir por los miembros de la organización, así como cerrar los mismos para fijarlos y establecerlos como revisados y correctos, de modo que si se realizan modificaciones en los datos asociados a las retribuciones de un miembro de la organización, los incentivos de éste marcados como 'cerrados' y que hayan sido revisados y dados por correctos no se verán afectados ni serán modificados.
Nivel de criticidad	Alto
Dependencias	La revisión de incentivos está relacionada con el cálculo de incentivos (RF – 10).

3.3 Requisitos no funcionales

Nombre	RNF - 01: Aplicación intuitiva
Descripción	La aplicación deberá ser intuitiva y fácil de usar, incluso por personas poco familiarizadas con las nuevas tecnologías.
Nivel de criticidad	Alto
Dependencias	-

Nombre	RNF - 02: Interacción con la aplicación a través de GUI
Descripción	La interacción del usuario con el SGORI será a través de GUI (y no a través de línea de comandos).
Nivel de criticidad	Medio
Dependencias	-

Nombre	RNF - 03: Interfaz de usuario consistente
Descripción	El SGORI debe mantener una GUI consistente con la interfaz preexistente, es decir, debe respetar el estilo de los elementos como listas, botones, imágenes, combinaciones de colores, etc., así como la ubicación habitual de los mismos.
Nivel de criticidad	Alto
Dependencias	La interfaz de usuario consistente está relacionada con la interacción a través de GUI (RNF - 02).

Nombre	RNF - 04: Aplicación escalable
Descripción	El SGORI será fácilmente escalable. Será posible añadir nuevos módulos, funcionalidades, elementos y <i>plug-ins</i> con facilidad.
Nivel de criticidad	Medio
Dependencias	-

Nombre	RNF - 05: Posibilidad de internacionalización
Descripción	El SGORI debe poder ser fácilmente a diferentes culturas, tanto en el idioma de los textos como en cualquier otro elemento que dependa de ellas, como formato de fechas, monedas o decimales.
Nivel de criticidad	Alto
Dependencias	-
Nombre	RNF - 06: Tolerancia a fallos
Descripción	El SGORI debe poder ser tolerante a fallos, esto es, debe poder detectar y recuperarse de eventos no controlados, así como proporcionar la información necesaria para su identificación.
Nivel de criticidad	Alto
Dependencias	-
Nombre	RNF - 07: Resoluciones de pantalla soportadas
Descripción	El SGORI deberá ser visualizado correctamente en pantallas con resoluciones iguales o superiores a 1024x768.
Nivel de criticidad	Alto
Dependencias	-
Nombre	RNF - 08: Manual de ayuda
Descripción	El SGORI contará con un manual de ayuda que guíe al usuario acerca de las características básicas del sistema y su modo de uso.
Nivel de criticidad	Bajo
Dependencias	-

4. Bibliografía

4.1 Referencias

[IEEE SRS] IEEE Recommended Practice for Software Requirements Specifications, Std-830-1998.

4.2 Referencias web

[Ref. Web 1] pdfs.wke.es/4/8/0/0/pd0000014800.pdf

Sistema de Gestión de Objetivos y Retribución de Incentivos

Análisis

Versión 1.3

Publicado el 08/04/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
11/03/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
18/03/2013	1.1	Modificaciones en el documento	Santiago Pelegrín Costea
05/04/2013	1.2	Revisión del documento	Fernando Cortés Franco
08/04/2013	1.3	Corrección del documento	Santiago Pelegrín Costea

Índice

1. Introducción	6
1.1 Propósito del documento	6
1.2 Alcance del documento	6
1.3 Acrónimos	6
1.4 Definiciones.....	6
1.5 Referencias	7
1.6 Resumen	7
2. Descripción del proceso.....	8
3. Análisis de los casos de uso	9
3.1 Introducción	9
3.2 Actores de los casos de uso.....	9
3.3 Casos de uso	10
3.3.1 Esquema general de los casos de uso.....	10
3.3.2 Gestión de indicadores.....	11
3.3.2.1 Añadir indicador.....	11
3.3.2.2 Eliminar indicador	13
3.3.2.3 Editar información del indicador.....	14
3.3.2.4 Gestión de personas implicadas en el indicador	15
3.3.2.4.1 Añadir persona implicada.....	15
3.3.2.4.2 Eliminar persona implicada	16
3.3.2.4.3 Editar peso del indicador sobre una persona implicada.....	17
3.3.2.5 Gestión de personas con privilegios sobre el indicador	19
3.3.2.5.1 Añadir persona con privilegios	19
3.3.2.5.2 Eliminar persona con privilegios.....	19
3.3.2.6 Gestión de barreras de entrada que afectan al indicador.....	20
3.3.2.6.1 Añadir barrera de entrada que afecte al indicador.....	20
3.3.2.6.2 Eliminar barrera de entrada que afecta al indicador	20
3.3.3 Gestión de datos	21
3.3.3.1 Editar datos objetivos	21
3.3.3.2 Editar datos reales.....	22
3.3.4 Gestión de remuneraciones	23
3.3.4.1 Añadir remuneración	23
3.3.4.2 Eliminar remuneración.....	24

3.3.4.3	Editar remuneración	26
3.3.4.3.1	Editar indicador asociado.....	26
3.3.4.3.2	Editar escalado.....	27
3.3.4.3.3	Editar retribución económica o penalización porcentual.....	28
3.3.4.4	Gestión de incentivos	28
3.3.4.4.1	Calcular incentivos	29
3.3.4.4.2	Validar incentivos.....	30
4.	Estructura estática	:Error! Marcador no definido.
4.1	Introducción	31
4.2	Identificación de paquetes de análisis.....	31
4.2.1	Gestión de indicadores.....	31
4.2.2	Gestión de datos	32
4.2.3	Gestión de remuneraciones e incentivos	32
4.2.4	Gestión de incidencias	33
4.2.5	Acceso a repositorio de persistencia de datos.....	33
4.2.6	Internacionalización de entidades visibles.....	33
5.	Requerimientos especiales	34
5.1	Persistencia.....	34
5.2	Tolerancia a fallos	34
5.3	Internacionalización	34
6.	Bibliografía	35
6.1	Referencias	35
6.2	Referencias web.....	35

1. Introducción

1.1 Propósito del documento

En el presente documento se muestra la fase de análisis del proyecto de desarrollo del SGORI. A partir de los requisitos descritos en el documento de especificación de requisitos y del estudio previo del sistema descrito en el documento de modelo de negocio, se realiza un análisis de la arquitectura del sistema, definiendo casos de uso, estructura de paquetes y requisitos especiales identificados.

1.2 Alcance del documento

Este documento pretende describir los resultados obtenidos en la fase de análisis del sistema. A lo largo del mismo, se describirán cada uno de los artefactos y actividades a realizar en esta fase del desarrollo, según la metodología RUP utilizada.

1.3 Acrónimos

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.
- RUP: Rational Unified Process.
- GUI: Graphic User Interface.
- BD: Base de Datos.
- GI: Gestor de Indicadores.
- GD: Gestor de Datos (objetivos y/o reales).
- GR: Gestor de Remuneraciones.

1.4 Definiciones

- Artefacto: cualquier tipo de información creada, producida, cambiada o utilizada por el equipo de desarrollo del proyecto.
- UML: lenguaje de modelado de sistemas de software desarrollado por *Rational* y de uso extendido.
- Indicador: herramienta básica del SGORI que permite medir la consecución o evolución de los objetivos de una organización basados principalmente en los aspectos clave del negocio. Pueden influir y ser aplicados sobre un grupo o colectivo, o sobre individuos en concreto.
- Barrera de entrada: indicador que establece el límite a partir del cual se tiene derecho al devengo del incentivo por la superación de los objetivos asignados a los indicadores.
- Caso de uso: especificación de las secuencias de acciones que pueden ser efectuadas por un sistema, subsistema o clase por interacción con actores externos.

- Diagrama de casos de uso: notación gráfica que describe un caso de uso.
- Diagrama de secuencia: notación gráfica que muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo.
- Diagrama de actividades: notación gráfica que representa los flujos de trabajo de los componentes en un sistema, paso a paso. Muestra el flujo de control general del sistema.

1.5 Referencias

En este documento se mencionan los siguientes documentos del proyecto:

- MODNEG: documento en el que se especifica el modelo de casos de uso del negocio.
- ESPREQ: documento en el que se especifican los requisitos del sistema.

1.6 Resumen

Este documento describe el análisis del SGORI. Se compone de seis apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describe el proceso de análisis seguido para la confección de este documento.
- Apartado 3: Descripción de los casos de uso.
- Apartado 4: Identificación de paquetes de análisis.
- Apartado 5: Identificación de los requisitos especiales del sistema.
- Apartado 6: Bibliografía y referencias Web utilizadas para la realización de este documento.

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 tienen lugar en el sistema, y los actores que forman parte de los mismos.

El 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 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 determinarán restricciones y especificaciones para fases futuras del proyecto.

3. Análisis de los casos de uso

3.1 Introducción

En este apartado se va a realizar el análisis de los casos de uso del sistema a partir del estudio realizado en el documento de modelo de negocio (MODNEG). En esta fase se identificarán todos los procesos que tienen lugar en el sistema.

El análisis de un caso de uso se realiza con el objetivo de profundizar en la funcionalidad desarrollada por el mismo, ya que en fases anteriores, la descripción de cada uno de ellos es realizada a muy alto nivel de detalle.

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

- Diagramas de caso de uso: describen la relación entre los usuarios del sistema y los casos de uso.
- Diagramas de secuencia: muestran la vista dinámica del caso de uso. Es decir, describen las interacciones existentes a lo largo del tiempo.
- Diagramas de actividades: describen el flujo de trabajo del caso de uso.

3.2 Actores de los casos de uso

Los actores que formarán parte de nuestro modelo de casos de uso se especifican a partir de las funciones de alto nivel del SGORI comentadas en el documento de modelo de negocio (MODNEG) y de las especificaciones del sistema del documento de especificación de requisitos (ESPREQ). El análisis conjunto de dichos elementos determina los siguientes actores del SGORI:

- Gestor de indicadores (GI):
 - Sus funciones se basan en la gestión de la información de los indicadores que integran el sistema.
 - Maneja los datos de los indicadores, su información asociada, sus características, las personas relacionadas y los privilegios sobre los mismos.
- Gestor de datos objetivos y reales (GD):
 - Sus funciones principales comprenden la gestión de los valores, tanto objetivos como reales conseguidos, asociados a los indicadores.
 - Maneja datos referentes a los objetivos establecidos para los indicadores, así como los valores reales que posteriormente son logrados y que servirán para calcular la remuneración final conseguida.
- Gestor de remuneraciones (GR):
 - Sus funciones se basan en la gestión de las remuneraciones asociadas a los indicadores y los incentivos a percibir por los miembros de la organización implicados en el sistema de indicadores.
 - Maneja datos referentes a las retribuciones que serán otorgadas a aquellas personas que consigan, para un indicador determinado, alcanzar el objetivo o uno de los objetivos previamente establecidos.

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 SGORI. Inicialmente, se presenta el esquema general de los casos de uso, donde se engloban los casos de uso en funcionalidades generales, y se puede tener una visión global del funcionamiento que debe tener el sistema, desde el punto de vista del usuario. Después se pasa a describir en detalle los casos de uso correspondientes a dichas funcionalidades generales.

Debido a que varias de las funcionalidades son muy parecidas, en algunos casos se detallará un caso de uso que explique la funcionalidad, y se comentarán las variaciones nominales que existan en los casos de uso similares.

3.3.1 Esquema general de los casos de uso

Para comenzar, y con el fin de establecer un marco general que facilite la comprensión de los casos de uso analizados posteriormente, se muestra un esquema general de los casos de uso del sistema (Figura 1). El esquema general de los casos de uso presenta las acciones principales que pueden realizar los actores del sistema.

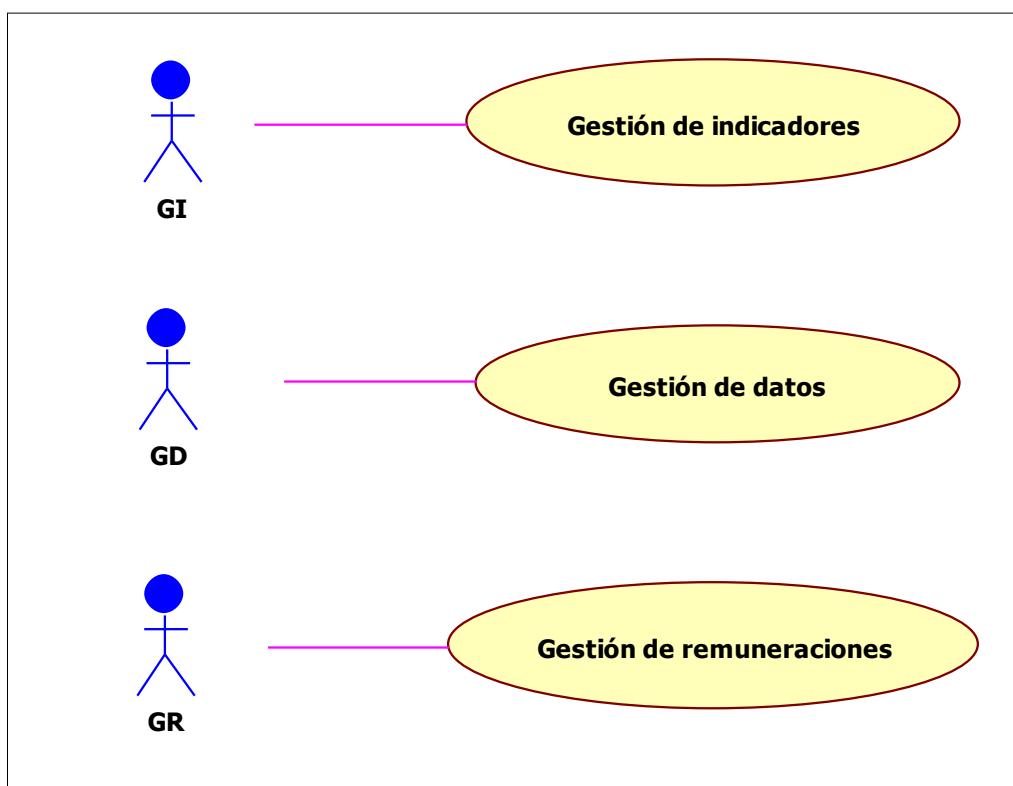


Figura 1: Esquema general de los casos de uso

3.3.2 Gestión de indicadores

En este apartado se describen los casos de uso referidos a la gestión de los indicadores del SGORI por parte del gestor de indicadores (GI). En la siguiente figura (Figura 2) se observa el diagrama del caso de uso ‘Gestión de indicadores’ con un mayor nivel de detalle:

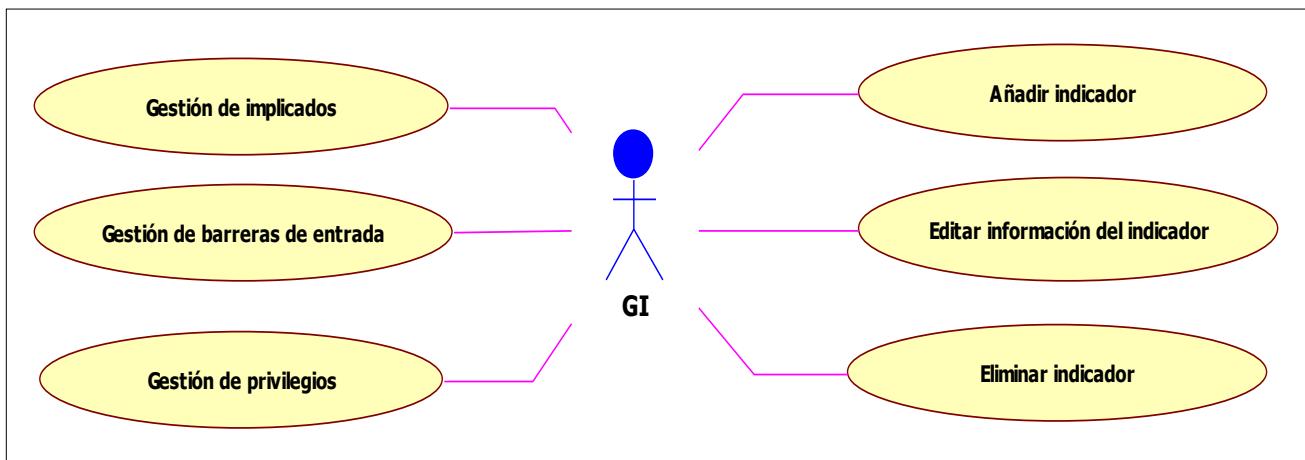


Figura 2: Caso de uso de la gestión de indicadores.

Como indica la figura 2, la gestión de indicadores consta de seis casos de uso diferenciados:

- Añadir indicador.
- Eliminar indicador.
- Editar información del indicador.
- Gestión de personas implicadas en el indicador.
- Gestión de personas con privilegios sobre el indicador.
- Gestión de barreras de entrada que afectan al indicador.

En los siguientes subapartados se analizará individualmente cada uno de estos casos de uso.

3.3.2.1 Añadir indicador

En este caso de uso se define la funcionalidad de añadir un nuevo indicador al sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, introduciendo los datos requeridos del nuevo indicador. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de indicadores, que será el encargado de la creación y almacenamiento del nuevo indicador, así como de informar al gestor de indicadores del resultado del proceso.

- Diagrama de secuencia (Figura 3):

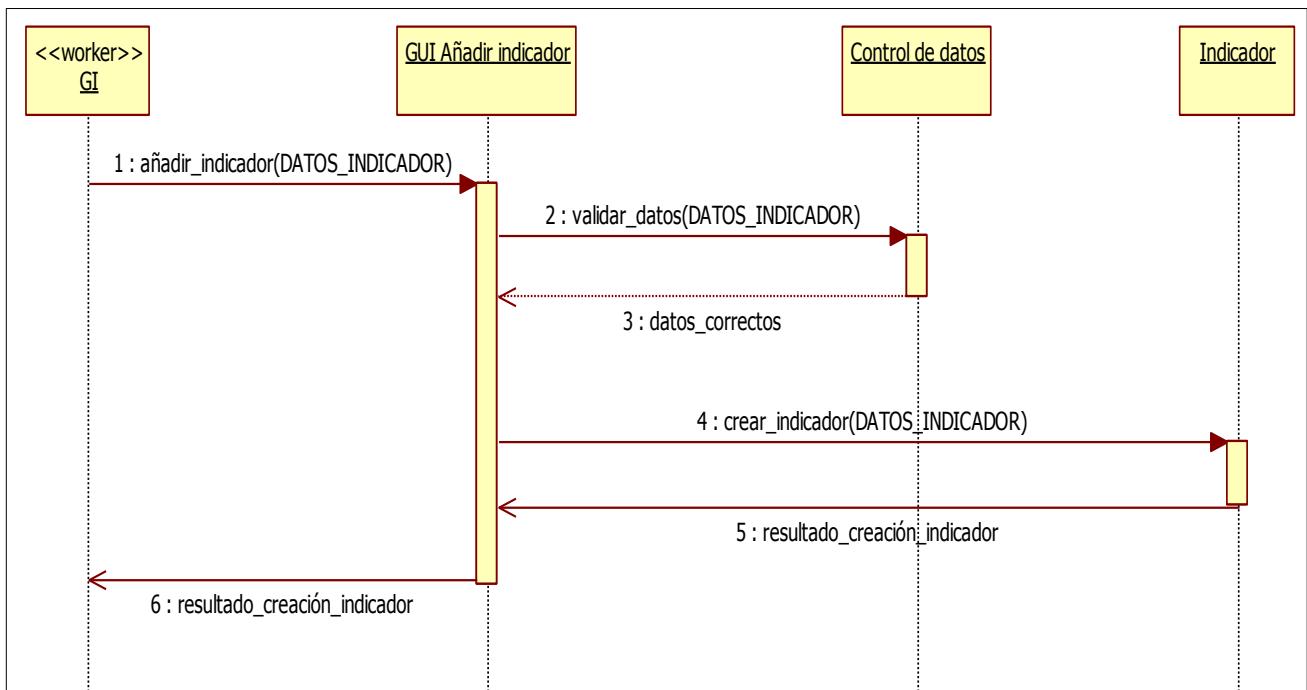


Figura 3: Diagrama de secuencia del caso de uso ‘Añadir indicador’

- Diagrama de actividades (Figura 4):

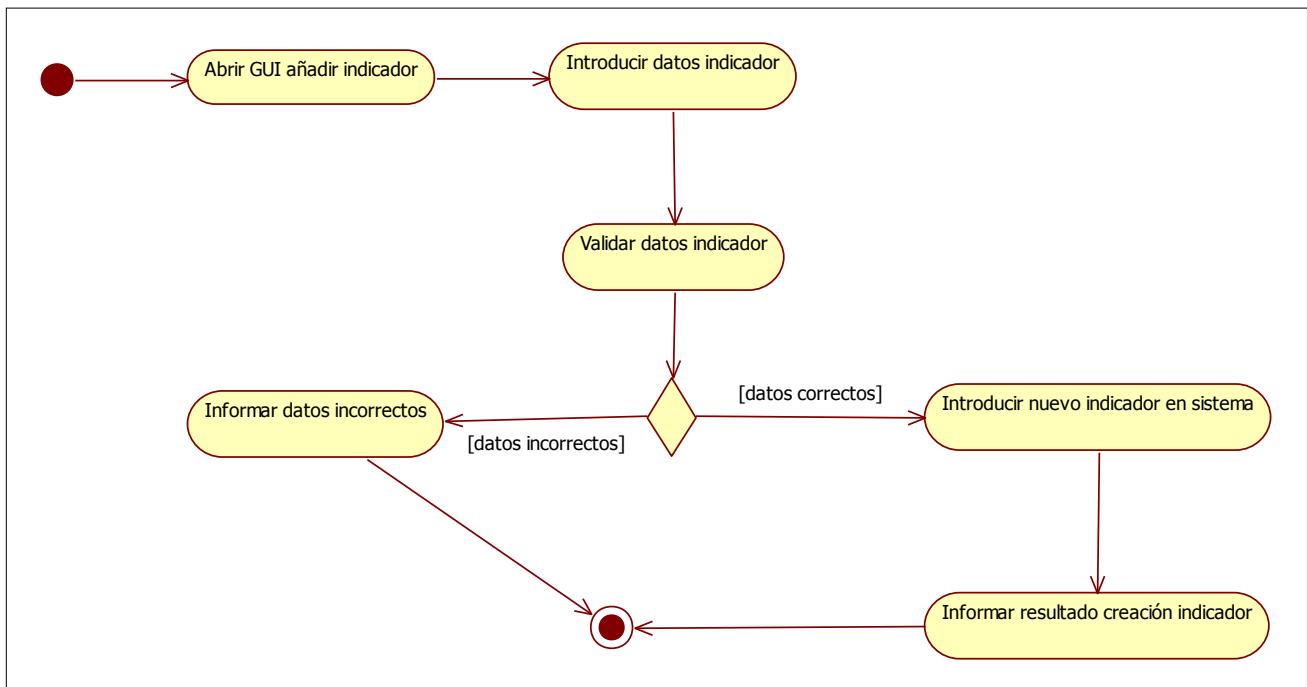


Figura 4: Diagrama de actividades del caso de uso ‘Añadir indicador’

3.3.2.2 Eliminar indicador

En este caso de uso se define la funcionalidad de eliminar un indicador definitivamente del sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, indicando el indicador que quiere eliminar. El control de datos comprueba si los datos son válidos. Si lo son, se procederá a la eliminación del indicador del sistema, así como de informar al gestor de indicadores del resultado del proceso.

- Diagrama de secuencia (Figura 5):

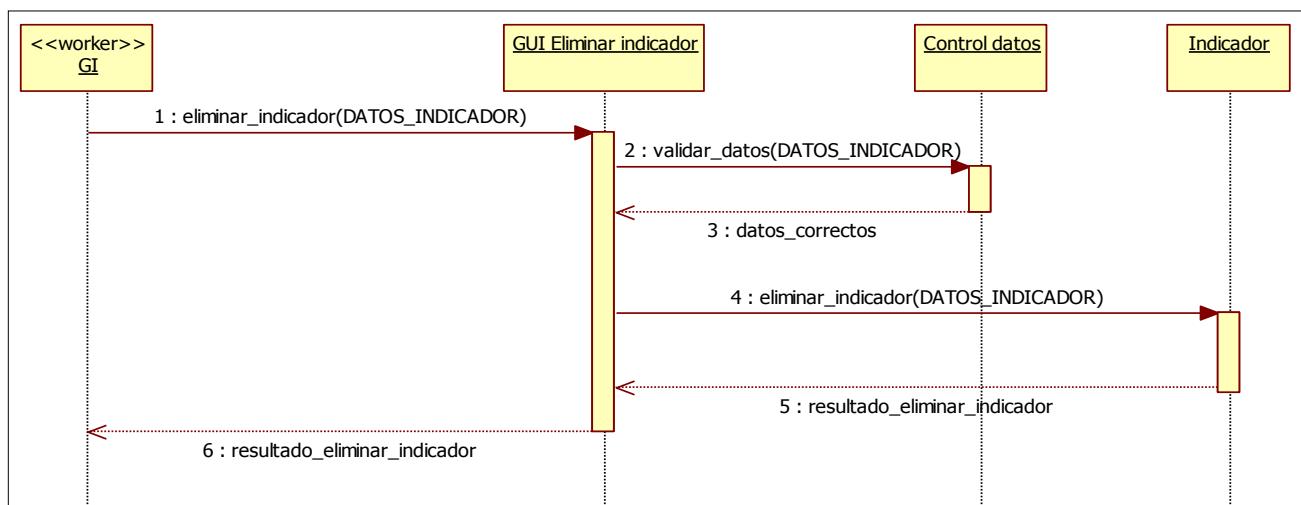


Figura 5: Diagrama de secuencia del caso de uso ‘Eliminar indicador’

- Diagrama de actividades (Figura 6):

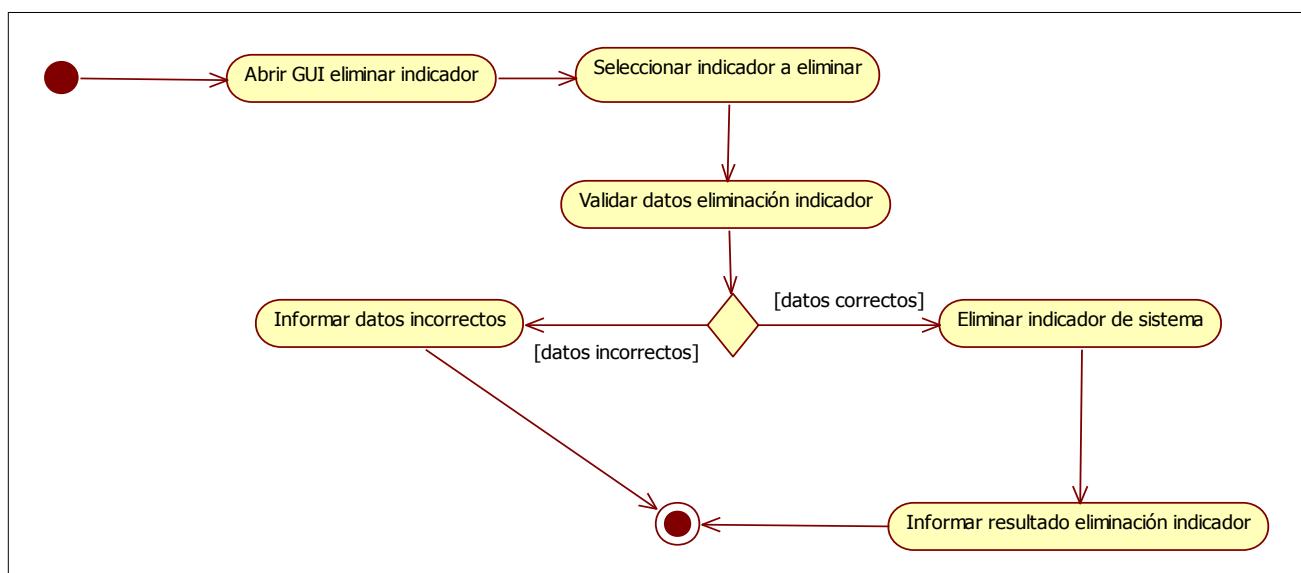


Figura 6: Diagrama de actividades del caso de uso ‘Eliminar indicador’

3.3.2.3 Editar información del indicador

En este caso de uso se define la funcionalidad de editar los datos de un indicador ya existente en el sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, modificando mediante un formulario los datos del indicador. Mediante el control de datos, se asegura que los datos modificados sean válidos, pasándoselos al control de gestión de indicadores, que se encarga de modificar los datos del indicador y de informar posteriormente al gestor de indicadores del resultado del proceso.

- Diagrama de secuencia (Figura 7):

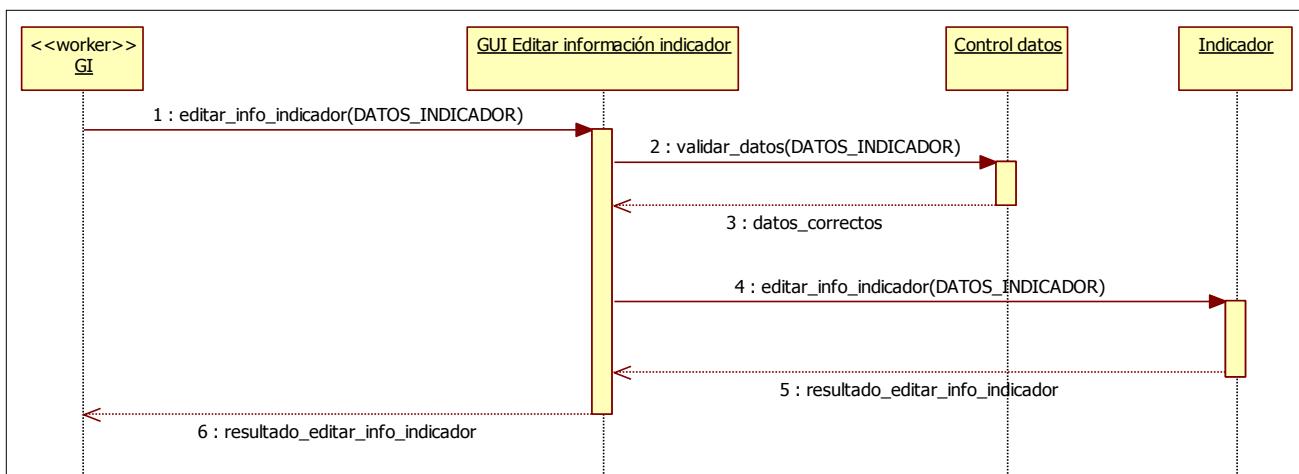


Figura 7: Diagrama de secuencia del caso de uso ‘Editar información del indicador’

- Diagrama de actividades (Figura 8):

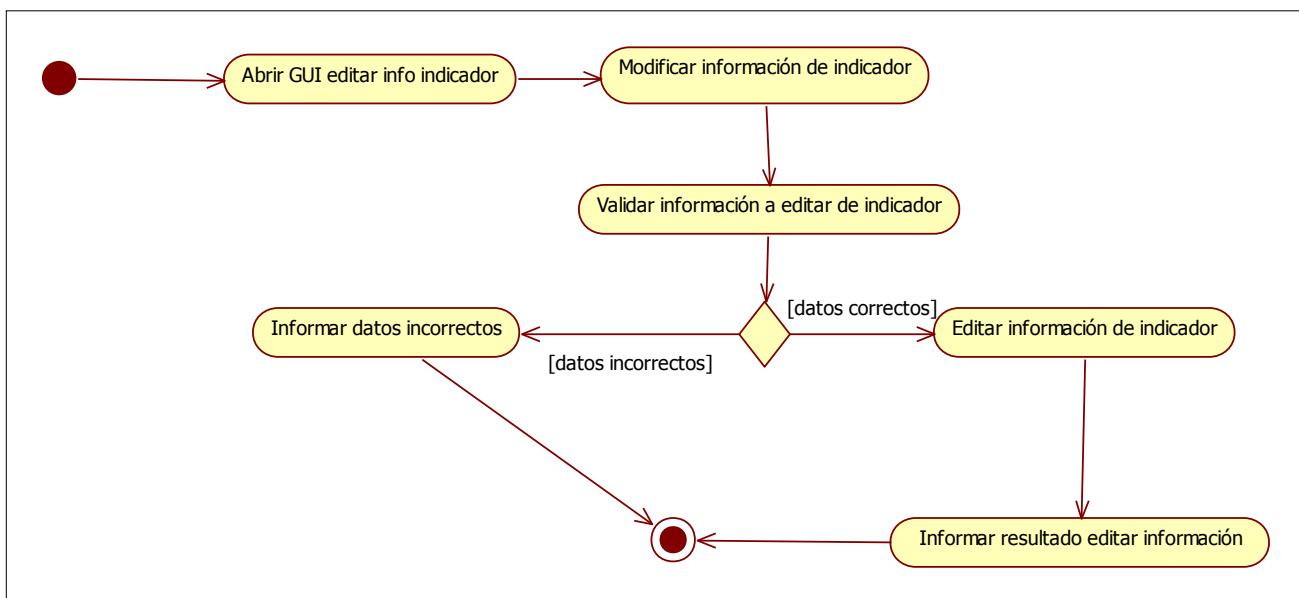


Figura 8: Diagrama de actividades del caso de uso ‘Editar información del indicador’

3.3.2.4 Gestión de personas implicadas en el indicador

En este apartado se describen los casos de uso referidos a la gestión de las personas implicadas en los indicadores del SGORI por parte del gestor de indicadores (GI). En la siguiente figura (Figura 9) se observa el diagrama del caso de uso ‘Gestión de personas implicadas’ con un mayor nivel de detalle:

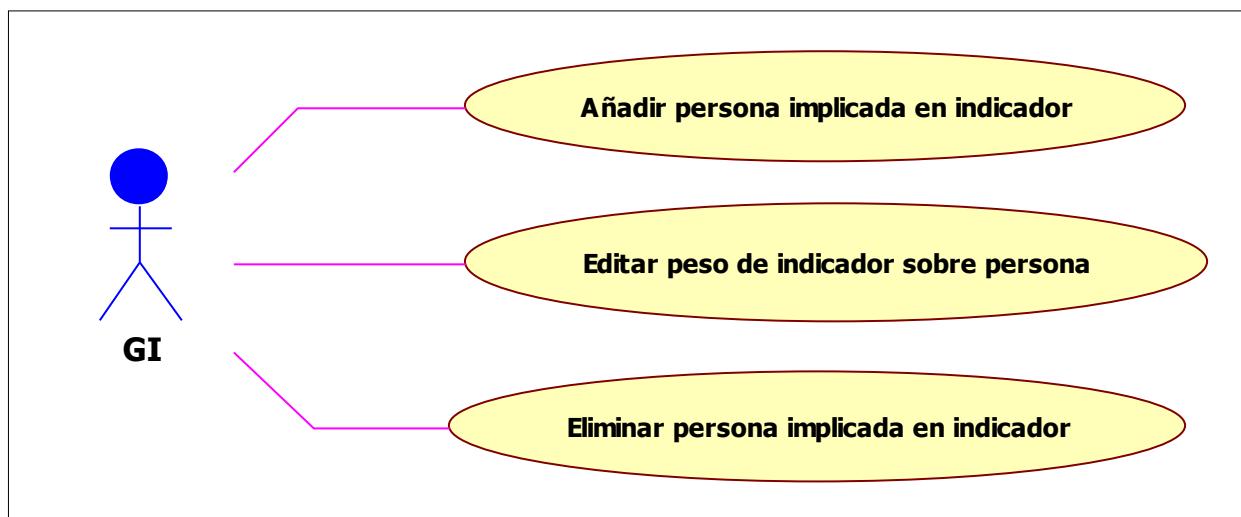


Figura 9: Caso de uso de la gestión de personas implicadas

Como indica la figura 9, la gestión de personas implicadas consta de tres casos de uso diferenciados:

- Añadir persona implicada.
- Eliminar persona implicada.
- Editar peso del indicador sobre una persona implicada.

3.3.2.4.1 Añadir persona implicada

En este caso de uso se define la funcionalidad de añadir una nueva persona implicada a un indicador existente en el sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, introduciendo los datos del indicador y de la persona que desea que esté implicada en el mismo. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de indicadores, que será el encargado del almacenamiento de la persona en la lista de implicados por el indicador, así como de informar al gestor de indicadores del resultado del proceso.

- Diagrama de secuencia (Figura 10):

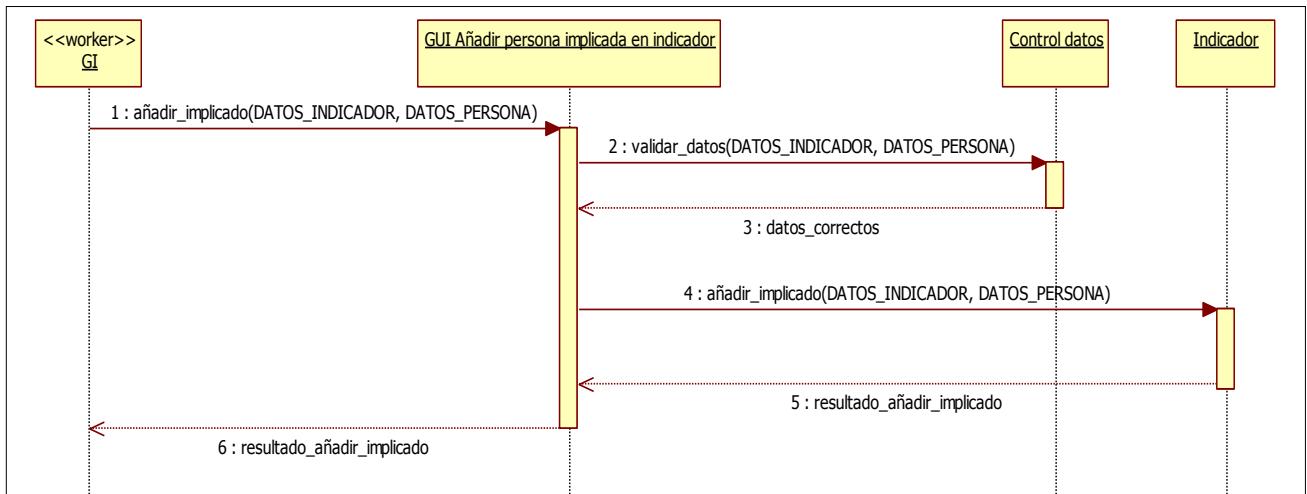


Figura 10: Diagrama de secuencia del caso de uso ‘Añadir persona implicada’

- Diagrama de actividades (Figura 11):

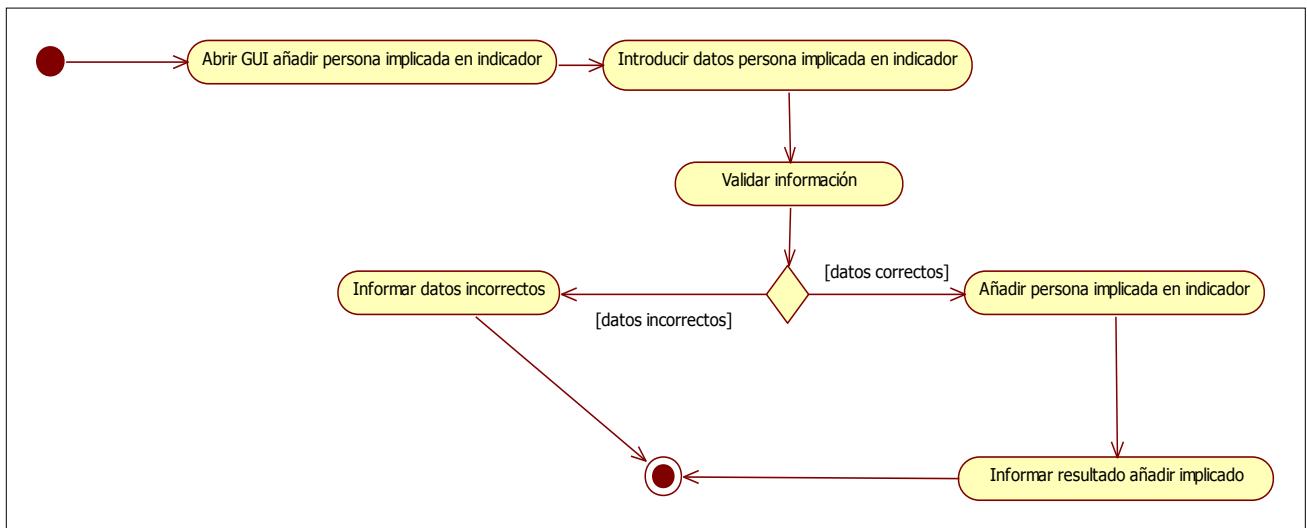


Figura 11: Diagrama de actividades del caso de uso ‘Añadir persona implicada’

3.3.2.4.2 Eliminar persona implicada

En este caso de uso se define la funcionalidad de eliminar una persona definitivamente de la lista de implicados por el indicador por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, indicando los datos del indicador y de la persona que quiere que deje de estar implicada en el mismo. El control de datos comprueba si los datos son válidos. Si lo son, se procederá a la eliminación de la persona de la lista de implicados por el indicador, así como de informar al gestor de indicadores del resultado del proceso.

- Diagrama de secuencia (Figura 12):

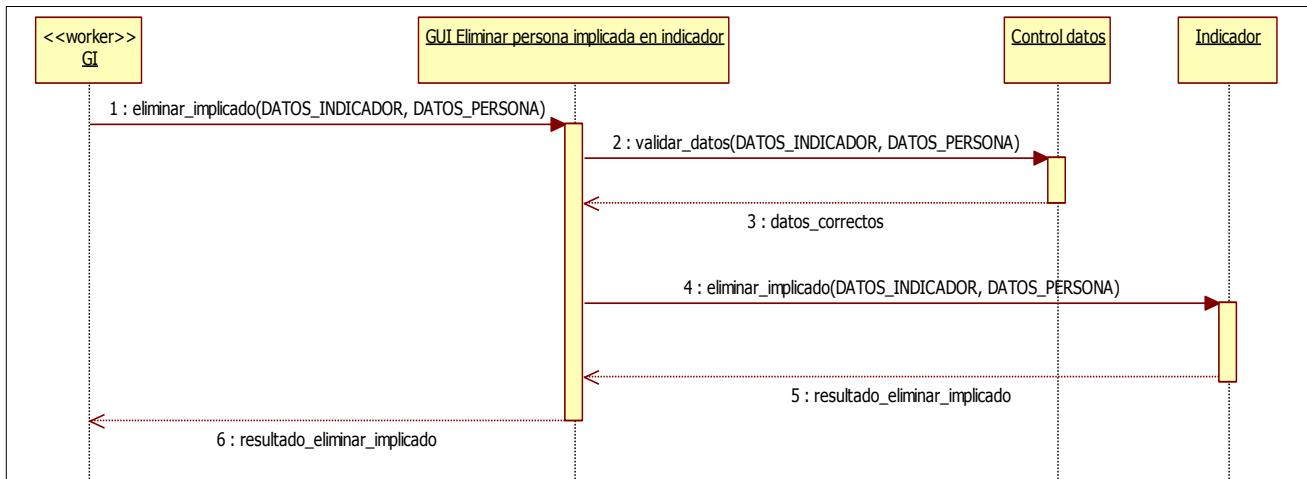


Figura 12: Diagrama de secuencia del caso de uso ‘Eliminar persona implicada’

- Diagrama de actividades (Figura 13):

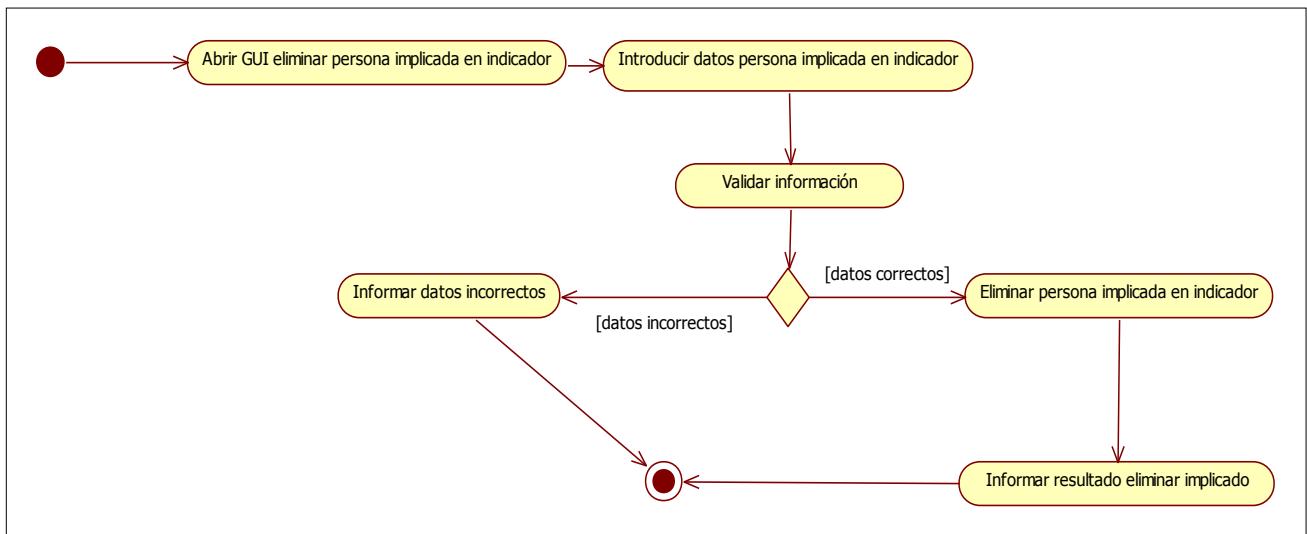


Figura 13: Diagrama de actividades del caso de uso ‘Eliminar persona implicada’

3.3.2.4.3 Editar peso del indicador sobre una persona implicada

En este caso de uso se define la funcionalidad editar la información referente a la ponderación o peso que supone el indicador para una persona implicada en el mismo por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, modificando mediante un formulario el dato de ponderación del indicador sobre una persona. Mediante el control de datos, se asegura que los datos modificados sean válidos, pasándoselos al control de gestión de indicadores, que se encarga de modificar los datos correspondientes y de informar posteriormente al gestor de indicadores del resultado del proceso.

- Diagrama de secuencia (Figura 14):

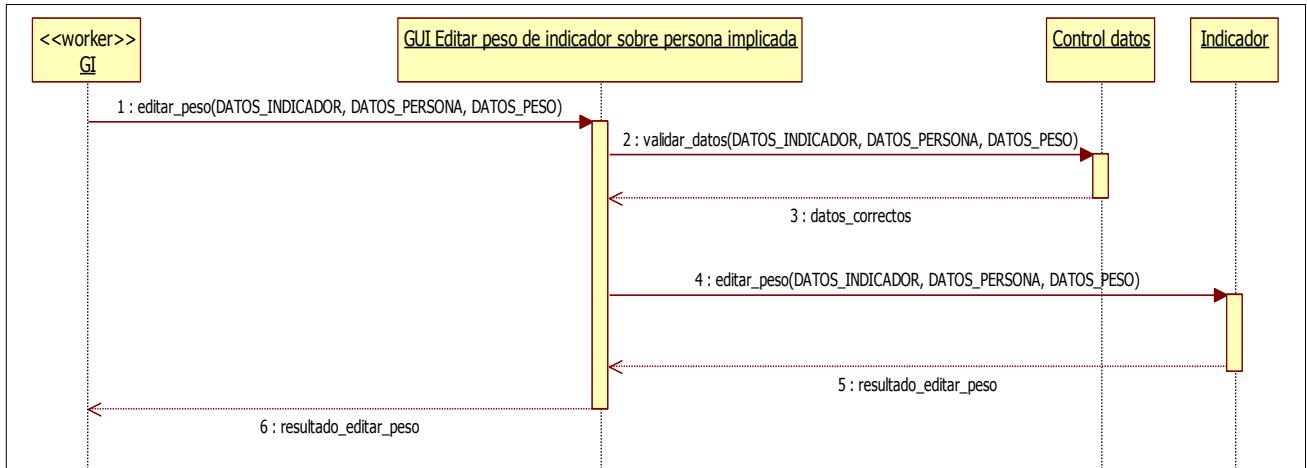


Figura 14: Diagrama de secuencia del caso de uso ‘Editar peso del indicador sobre persona implicada’

- Diagrama de actividades (Figura 15):

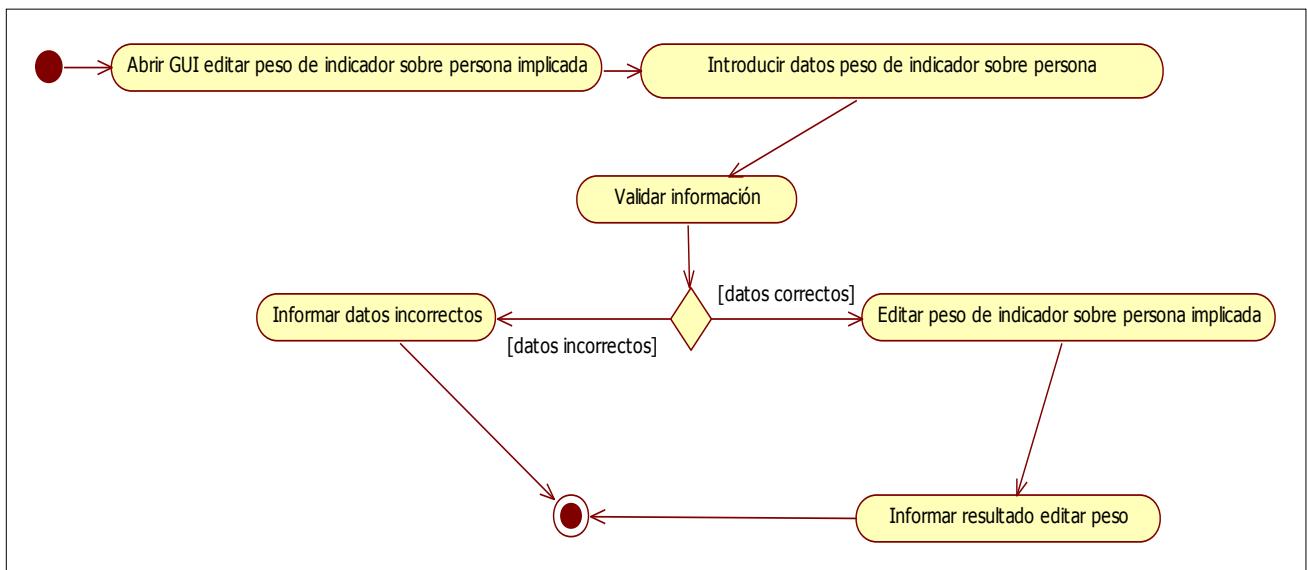


Figura 15: Diagrama de actividades del caso de uso ‘Editar peso del indicador sobre persona implicada’

3.3.2.5 Gestión de personas con privilegios sobre el indicador

En este apartado se describen los casos de uso referidos a la gestión de las personas con privilegios sobre los indicadores del SGORI por parte del gestor de indicadores (GI). En la siguiente figura (Figura 16) se observa el diagrama del caso de uso ‘Gestión de personas con privilegios’ con un mayor nivel de detalle:

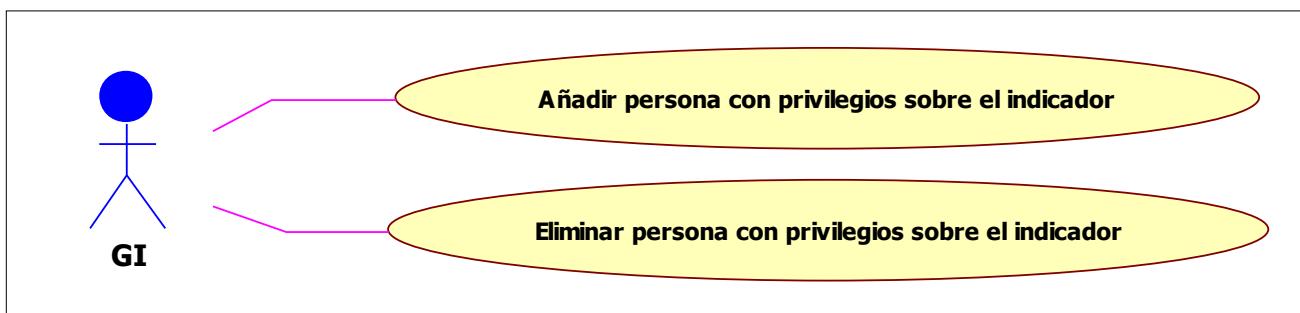


Figura 16: Caso de uso de la gestión de personas con privilegios

Como indica la figura 16, la gestión de personas implicadas consta de dos casos de uso diferenciados:

- Añadir persona con privilegios.
- Eliminar persona con privilegios.

3.3.2.5.1 Añadir persona con privilegios

En este caso de uso se define la funcionalidad de añadir una nueva persona con privilegios sobre un indicador existente en el sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, introduciendo los datos del indicador y de la persona que desea que posea privilegios sobre el mismo. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de indicadores, que será el encargado del almacenamiento de la persona en la lista de personas con privilegios sobre el indicador, así como de informar al gestor de indicadores del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.2.4.1 (figuras 10 y 11), por lo que no se considera necesario replicarlos nuevamente.

3.3.2.5.2 Eliminar persona con privilegios

En este caso de uso se define la funcionalidad de eliminar una persona definitivamente de la lista de personas con privilegios sobre el indicador por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, indicando los datos del indicador y de la persona que quiere que deje de poseer privilegios sobre el mismo. El control de datos comprueba si los datos son válidos. Si lo son, se procederá a la eliminación de la persona de la lista de personas con privilegios sobre el indicador, así como de informar al gestor de indicadores del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.2.4.2 (figuras 12 y 13), por lo que no se considera necesario replicarlos nuevamente.

3.3.2.6 Gestión de barreras de entrada que afectan al indicador

En este apartado se describen los casos de uso referidos a la gestión de los indicadores de tipo barrera de entrada que afectan al resto de los indicadores del SGORI por parte del gestor de indicadores (GI). En la siguiente figura (Figura 17) se observa el diagrama del caso de uso ‘Gestión de barreras de entrada que afectan al indicador’ con un mayor nivel de detalle:

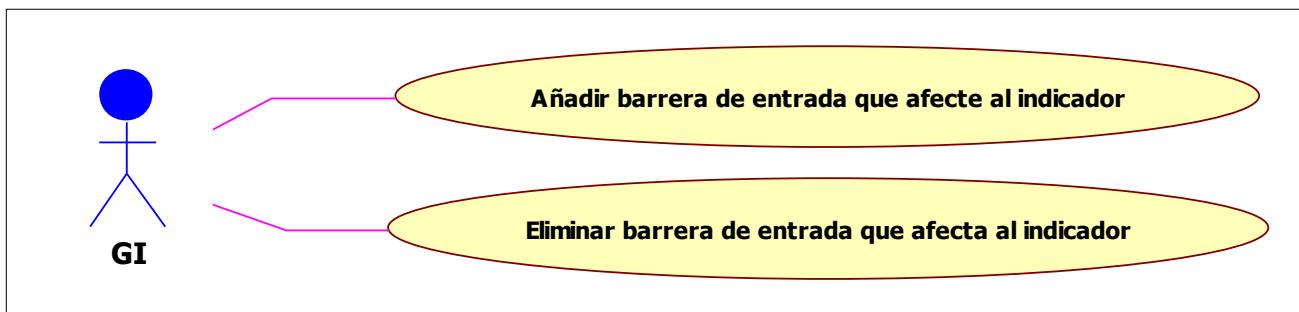


Figura 17: Caso de uso de la gestión de barreras de entrada que afectan al indicador

Como indica la figura 17, la gestión de barreras de entrada consta de dos casos de uso diferenciados:

- Añadir barrera de entrada que afecte al indicador.
- Eliminar barrera de entrada que afecta al indicador.

3.3.2.6.1 Añadir barrera de entrada que afecte al indicador

En este caso de uso se define la funcionalidad de añadir una nueva barrera de entrada que afecte a un indicador existente en el sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, introduciendo los datos del indicador y de la barrera de entrada que desea que afecte al mismo. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de indicadores, que será el encargado del almacenamiento de la barrera de entrada en la lista de barreras que afectan al indicador, así como de informar al gestor de indicadores del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.2.4.1 (figuras 10 y 11), por lo que no se considera necesario replicarlos nuevamente.

3.3.2.6.2 Eliminar barrera de entrada que afecta al indicador

En este caso de uso se define la funcionalidad de eliminar una barrera de entrada definitivamente de la lista de barreras que afectan al indicador por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, indicando los datos del indicador y de la barrera de entrada que quiere que deje de afectar al mismo. El control de datos comprueba si los datos son válidos. Si lo son, se procederá a la eliminación de la barrera de entrada de la lista de barreras que afectan al indicador, así como de informar al gestor de indicadores del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.2.4.2 (figuras 12 y 13), por lo que no se considera necesario replicarlos nuevamente.

3.3.3 Gestión de datos

En este apartado se describen los casos de uso referidos a la gestión de los datos objetivos y reales asociados a los indicadores del SGORI por parte del gestor de indicadores (GD). En la siguiente figura (Figura 18) se observa el diagrama del caso de uso ‘Gestión de datos’ con un mayor nivel de detalle:

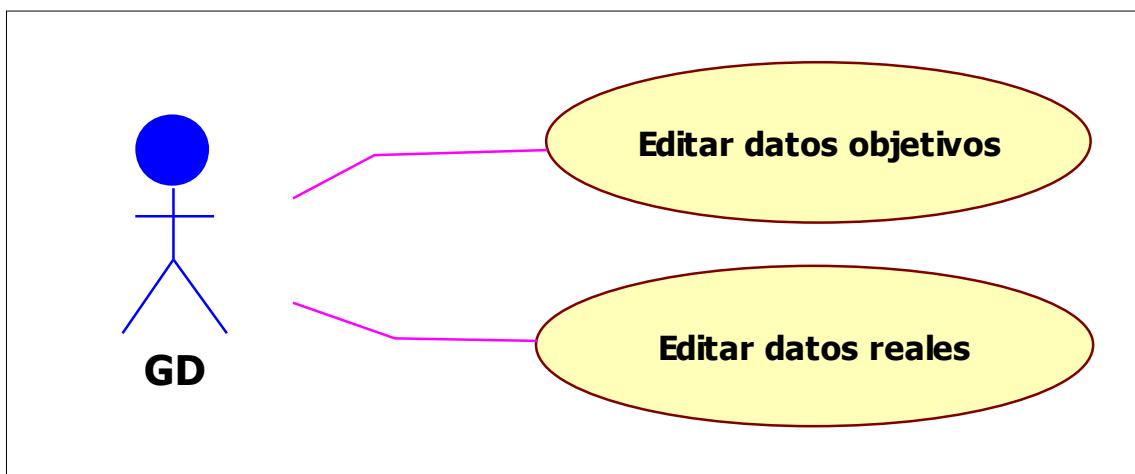


Figura 18: Caso de uso de la gestión de datos de indicadores.

Como indica la figura 18, la gestión de datos consta de dos casos de uso diferenciados:

- Editar datos objetivos.
- Editar datos reales.

En los siguientes subapartados se analizará individualmente cada uno de estos casos de uso.

3.3.3.1 Editar datos objetivos

En este caso de uso se define la funcionalidad de editar los datos a conseguir o datos objetivos asociados a un indicador ya existente en el sistema por parte del gestor de datos. Éste interactúa con la GUI del sistema, modificando mediante un formulario los objetivos. Mediante el control de datos, se asegura que los datos modificados sean válidos, pasándoselos al control de gestión de datos, que se encarga de modificar los objetivos del indicador y de informar posteriormente al gestor de datos del resultado del proceso.

- Diagrama de secuencia (Figura 19):

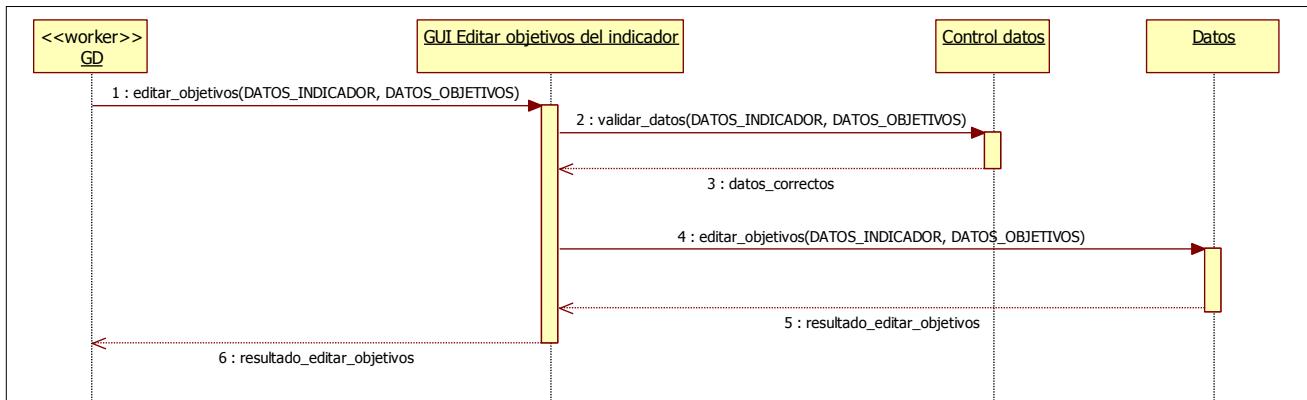


Figura 19: Diagrama de secuencia del caso de uso ‘Editar datos objetivos del indicador’

- Diagrama de actividades (Figura 20):

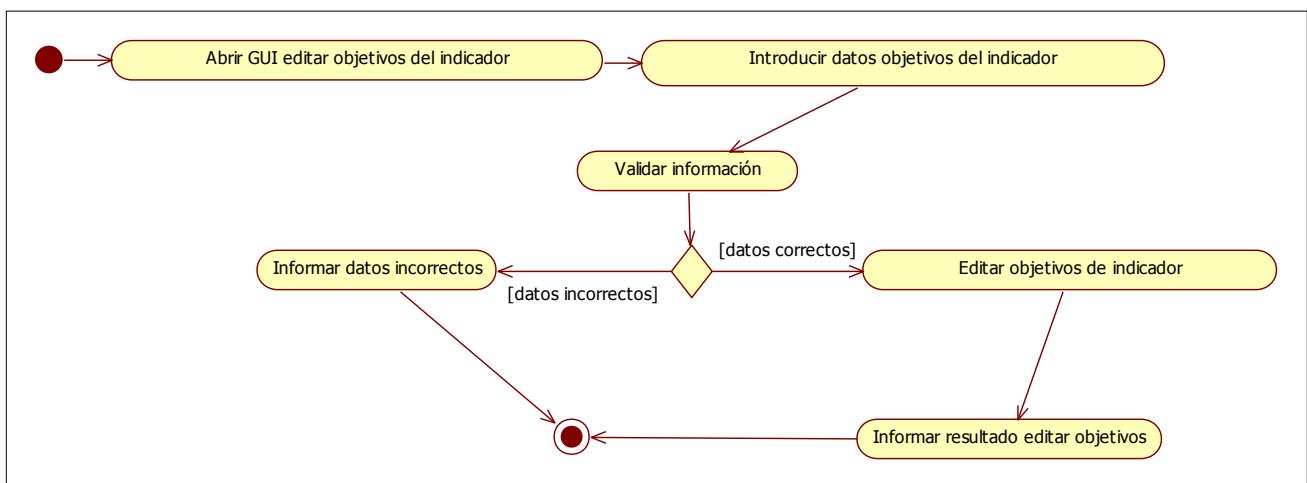


Figura 20: Diagrama de actividades del caso de uso ‘Editar datos objetivos del indicador’

3.3.3.2 Editar datos reales

En este caso de uso se define la funcionalidad de editar los datos conseguidos o datos reales asociados a un indicador ya existente en el sistema por parte del gestor de indicadores. Éste interactúa con la GUI del sistema, indicando mediante un formulario los datos reales a modificar en el indicador. Mediante el control de datos, se asegura que los datos modificados sean válidos, pasándoselos al control de gestión de datos, que se encarga de modificar los datos reales asociados al indicador y de informar posteriormente al gestor de datos del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.3.1 (figuras 19 y 20), por lo que no se considera necesario replicarlos nuevamente.

3.3.4 Gestión de remuneraciones

En este apartado se describen los casos de uso referidos a la gestión de remuneraciones del SGORI por parte del gestor de remuneraciones (GR). En la siguiente figura (Figura 21) se observa el diagrama del caso de uso ‘Gestión de remuneraciones con un mayor nivel de detalle’:

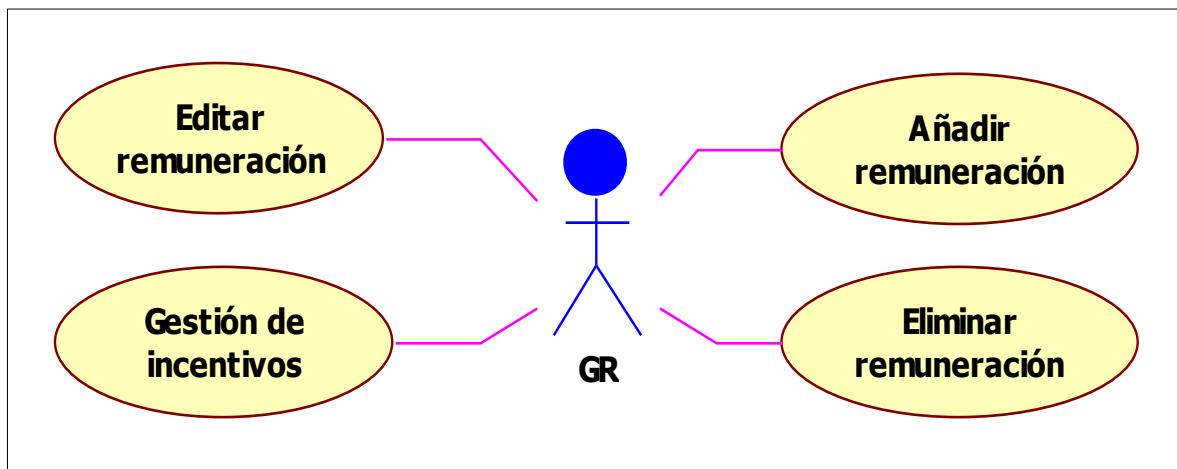


Figura 21: Caso de uso de la gestión de remuneraciones

Como indica la figura 21, la gestión de remuneraciones consta de cuatro casos de uso diferenciados:

- Añadir remuneración.
- Eliminar remuneración.
- Editar remuneración.
- Gestión de incentivos

En los siguientes subapartados se analizará individualmente cada uno de estos casos de uso.

3.3.4.1 Añadir remuneración

En este caso de uso se define la funcionalidad de añadir una nueva remuneración asociada a un indicador al sistema por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, introduciendo los datos requeridos. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de remuneraciones, que será el encargado de la creación y almacenamiento de la nueva remuneración, así como de informar al gestor de remuneraciones del resultado del proceso.

- Diagrama de secuencia (Figura 22):

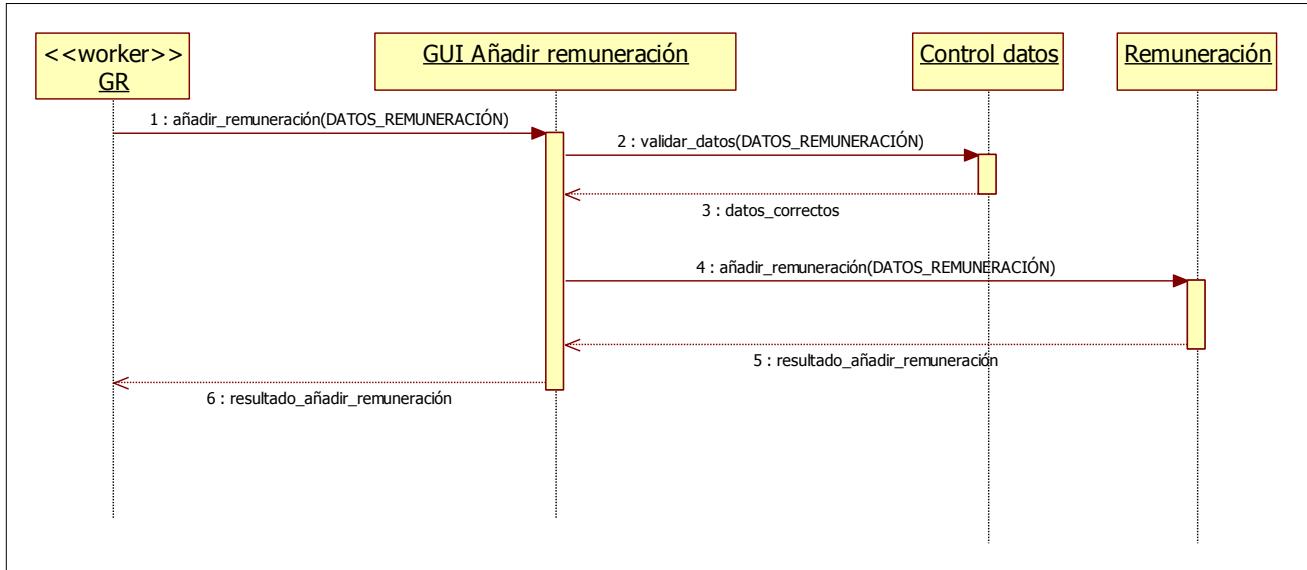


Figura 22: Diagrama de secuencia del caso de uso ‘Añadir remuneración’

- Diagrama de actividades (Figura 23):

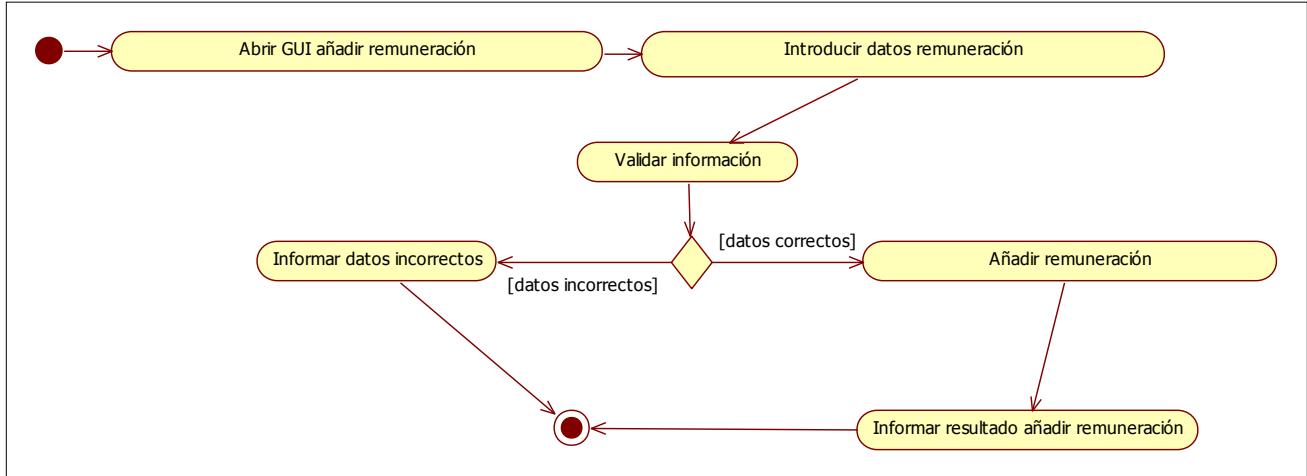


Figura 23: Diagrama de actividades del caso de uso ‘Añadir remuneración’

3.3.4.2 Eliminar remuneración

En este caso de uso se define la funcionalidad de eliminar una remuneración asociada a un indicador definitivamente del sistema por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, indicando la remuneración que quiere eliminar. El control de datos comprueba si los datos son válidos. Si lo son, se procederá a la eliminación de la remuneración del sistema, así como de informar al gestor de remuneraciones del resultado del proceso.

- Diagrama de secuencia (Figura 24):

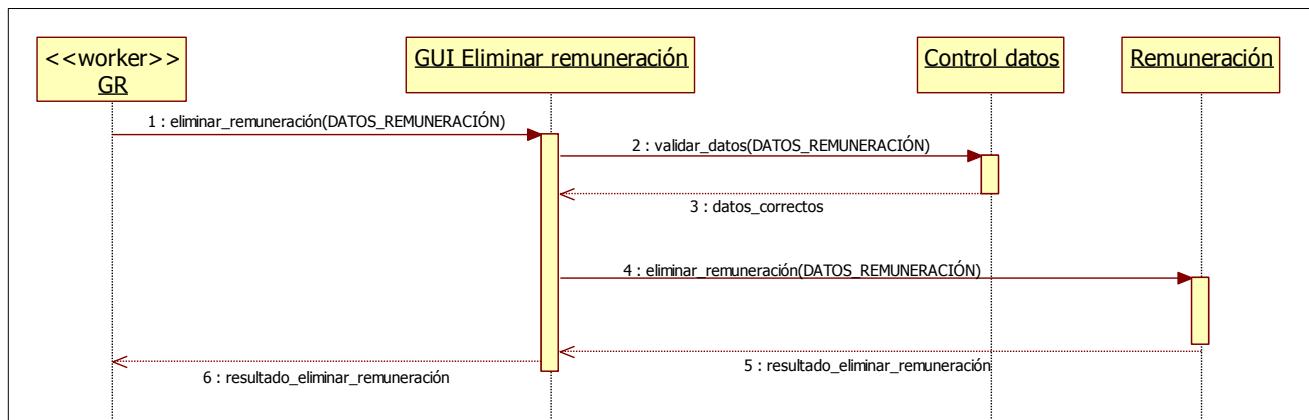


Figura 24: Diagrama de secuencia del caso de uso ‘Eliminar remuneración’

- Diagrama de actividades (Figura 25):

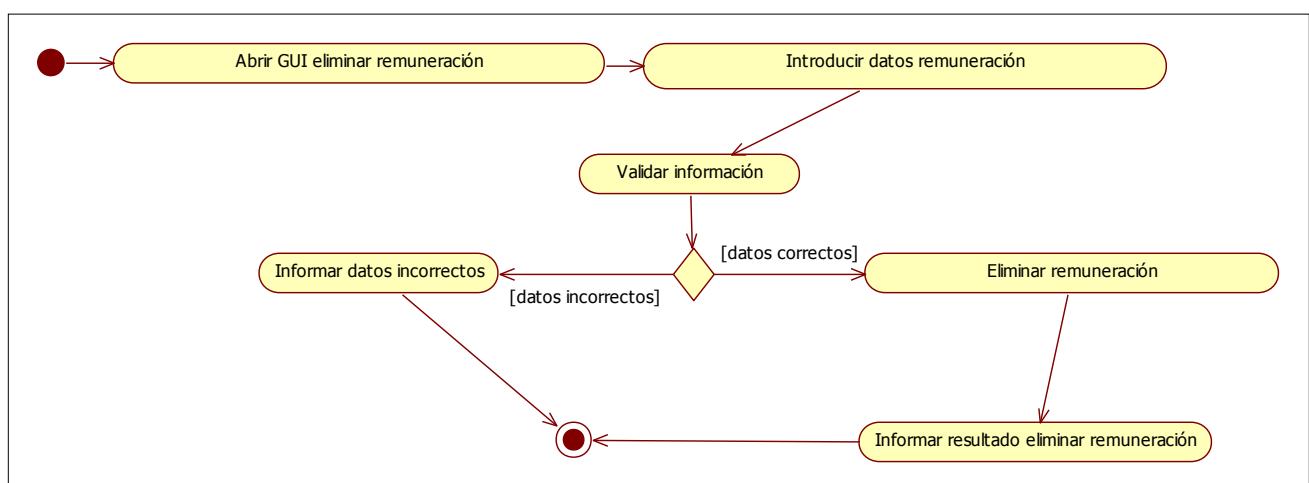


Figura 25: Diagrama de actividades del caso de uso ‘Eliminar remuneración’

3.3.4.3 Editar remuneración

En este apartado se describen los casos de uso referidos a la edición de las remuneraciones del SGORI por parte del gestor de remuneraciones (GR). En la siguiente figura (Figura 26) se observa el diagrama del caso de uso ‘Editar remuneración’ con un mayor nivel de detalle:

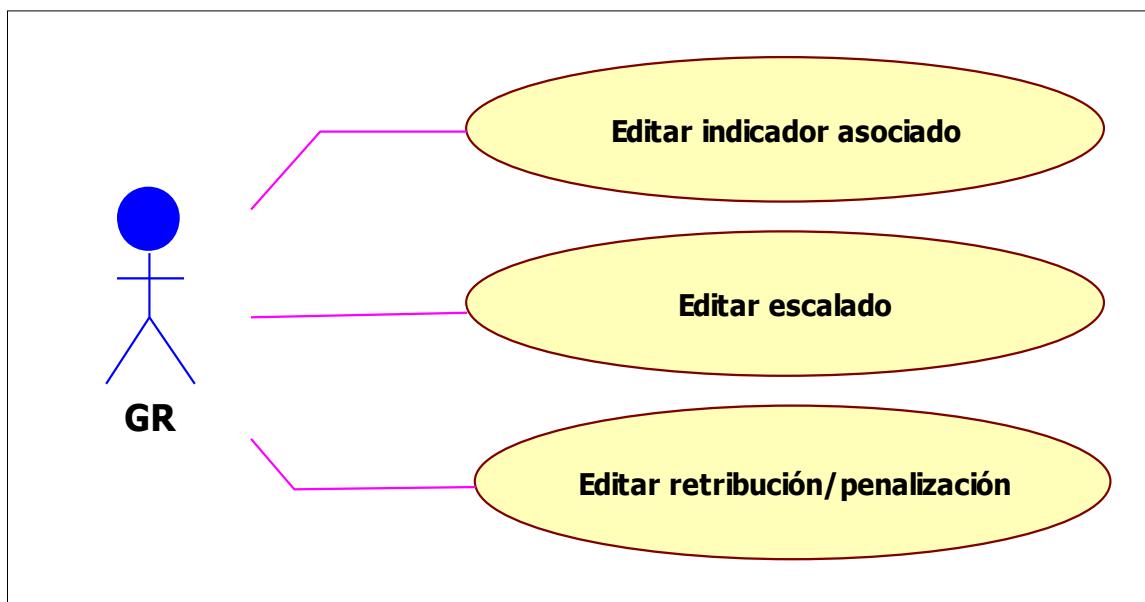


Figura 26: Caso de uso ‘Editar remuneración’

Como indica la figura 26, la edición de remuneraciones consta de tres casos de uso diferenciados:

- Editar indicador asociado.
- Editar escalado.
- Editar retribución económica o penalización porcentual.

3.3.4.3.1 Editar indicador asociado

En este caso de uso se define la funcionalidad de editar el indicador asociado a una remuneración existente en el sistema por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, introduciendo los datos de la remuneración y del indicador asociado a la misma. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de remuneraciones, que será el encargado del almacenamiento de los mismos, así como de informar al gestor de remuneraciones del resultado del proceso.

- Diagrama de secuencia (Figura 27):

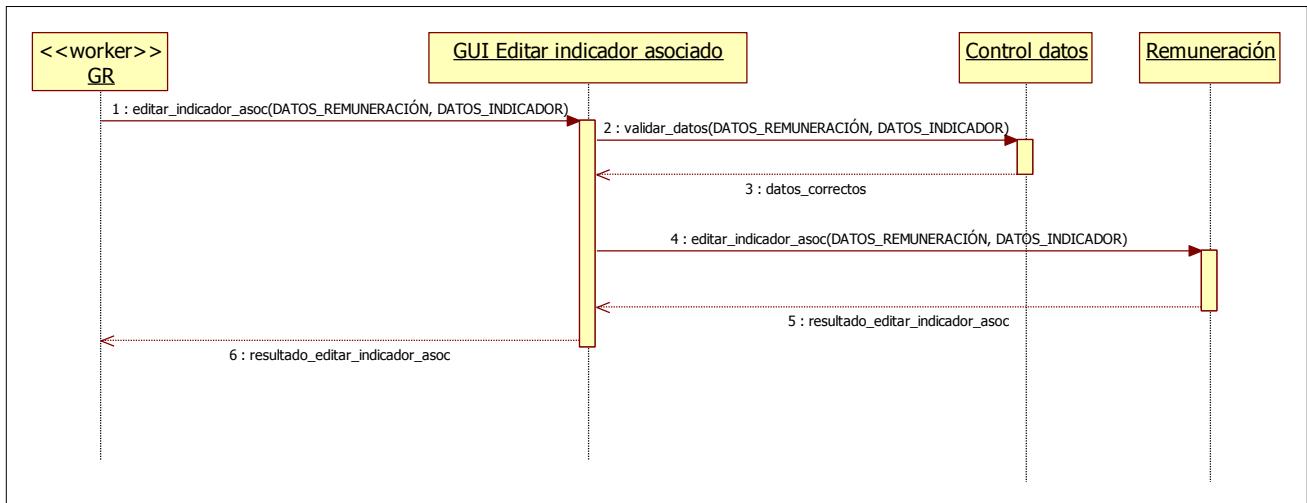


Figura 27: Diagrama de secuencia del caso de uso ‘Editar indicador asociado’

- Diagrama de actividades (Figura 28):

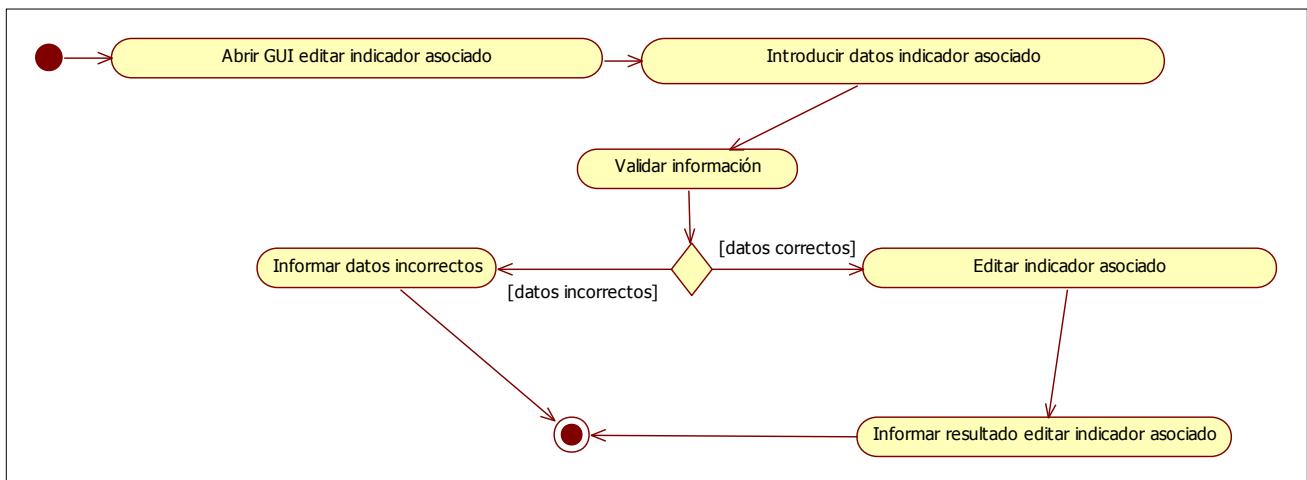


Figura 28: Diagrama de actividades del caso de uso ‘Editar indicador asociado’

3.3.4.3.2 Editar escalado

En este caso de uso se define la funcionalidad de editar el escalado asociado a una remuneración existente en el sistema por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, introduciendo los datos de la remuneración y del escalado que desea editar. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de remuneraciones, que será el encargado del almacenamiento y modificación de los mismos, así como de informar al gestor de remuneraciones del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.4.3.1 (figuras 27 y 28), por lo que no se considera necesario replicarlos nuevamente.

3.3.4.3.3 Editar retribución económica o penalización porcentual

En este caso de uso se define la funcionalidad de editar la retribución económica (para el caso de indicadores) o la penalización porcentual (para el caso de barreras de entrada) asociadas a una remuneración existente en el sistema por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, introduciendo los datos de la remuneración y de la retribución o penalización que desea editar. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de remuneraciones, que será el encargado del almacenamiento de los mismos, así como de informar al gestor de remuneraciones del resultado del proceso.

Los diagramas de secuencia y actividad de este caso de uso son análogos a los del apartado 3.3.4.3.1 (figuras 27 y 28), por lo que no se considera necesario replicarlos nuevamente.

3.3.4.4 Gestión de incentivos

En este apartado se describen los casos de uso referidos a la gestión de incentivos del SGORI por parte del gestor de remuneraciones (GR). En la siguiente figura (Figura 29) se observa el diagrama del caso de uso ‘Gestión de incentivos’ con un mayor nivel de detalle:

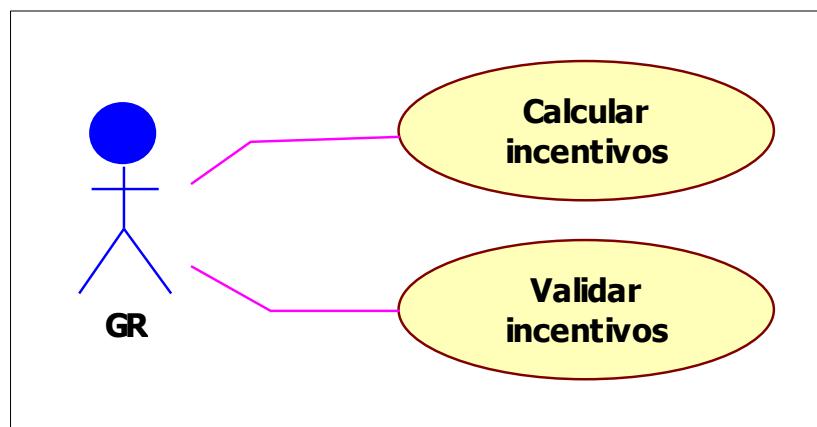


Figura 29: Caso de uso ‘Gestión de incentivos’

Como indica la figura 29, la gestión de incentivos consta de 2 casos de uso diferenciados:

- Calcular incentivos.
- Validar incentivos

En los siguientes subapartados se analizará individualmente cada uno de estos casos de uso.

3.3.4.4.1 Calcular incentivos

En este caso de uso se define la funcionalidad de calcular los incentivos a percibir por las personas implicadas en el SGORI por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, introduciendo los datos de la persona de la cual quiere realizar el cálculo de los incentivos. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de remuneraciones, que será el encargado del cálculo correspondiente, así como de informar al gestor de remuneraciones del resultado del proceso.

- Diagrama de secuencia (Figura 30):

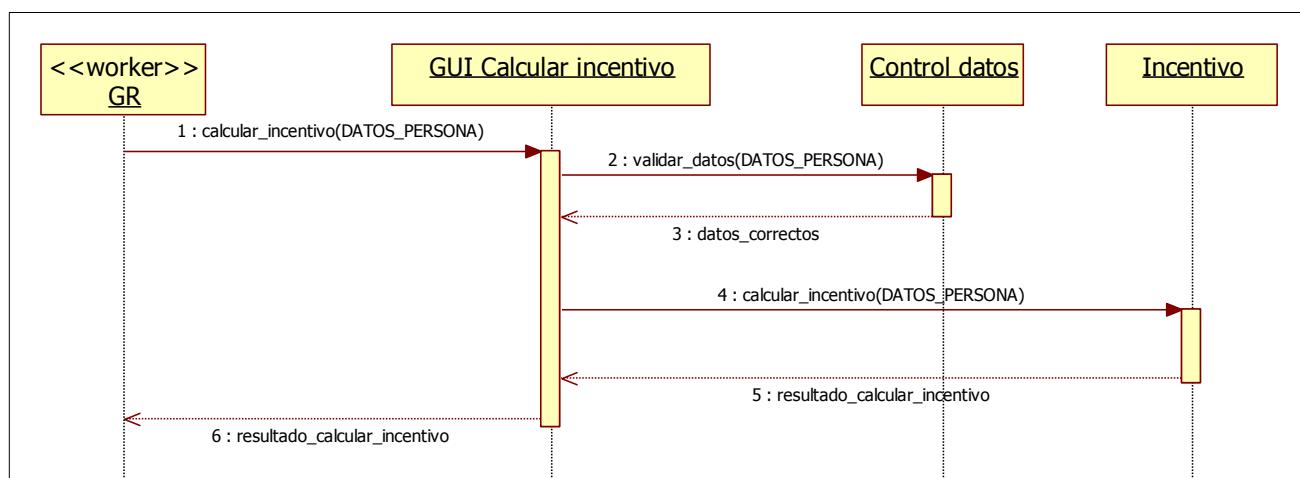


Figura 30: Diagrama de secuencia del caso de uso ‘Calcular incentivos’

- Diagrama de actividades (Figura 31):

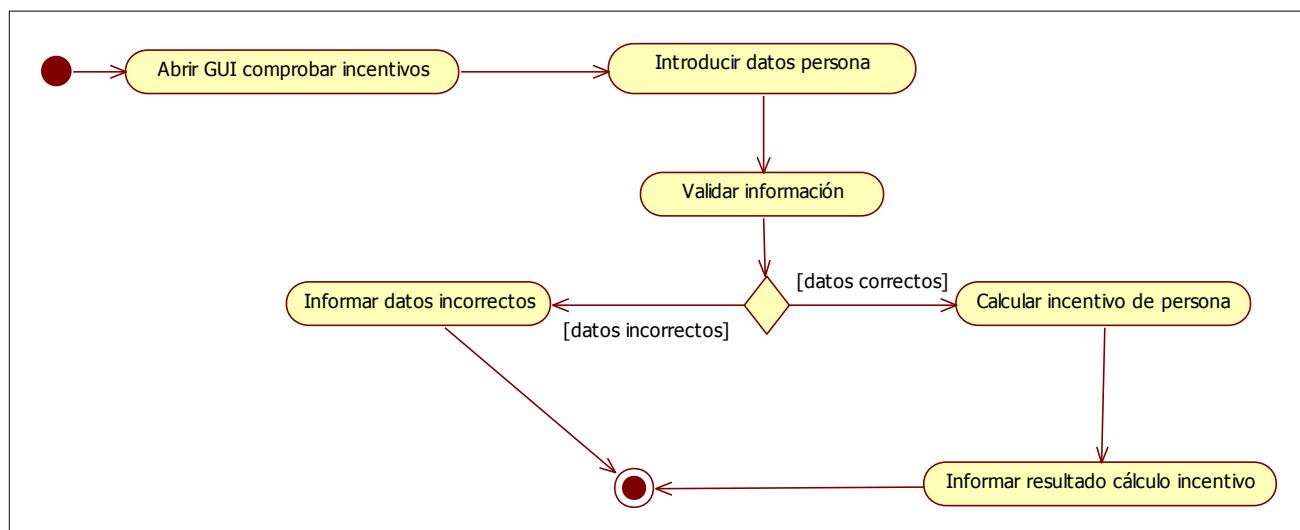


Figura 31: Diagrama de actividades del caso de uso ‘Calcular incentivos’

3.3.4.4.2 Validar incentivos

En este caso de uso se define la funcionalidad de validar el incentivo asociado a una persona implicada en el sistema de indicadores por parte del gestor de remuneraciones. Éste interactúa con la GUI del sistema, introduciendo los datos de la persona y del incentivo que desea validar. Mediante el control de datos se asegura que los datos introducidos son correctos, siendo éstos pasados al control de gestión de remuneraciones, que será el encargado del almacenamiento de los mismos, así como de informar al gestor de remuneraciones del resultado del proceso.

- Diagrama de secuencia (Figura 32):

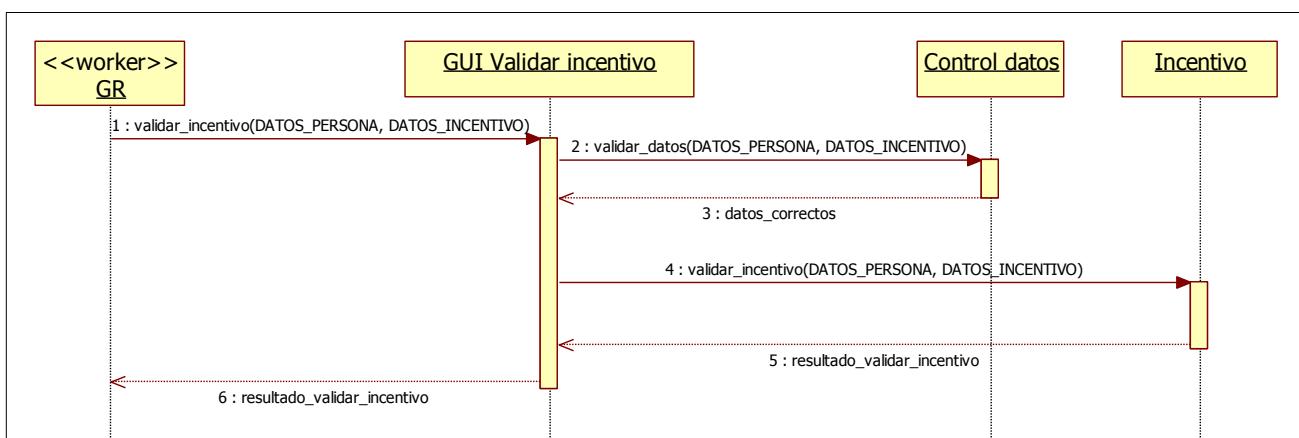


Figura 32: Diagrama de secuencia del caso de uso ‘Validar incentivos’

- Diagrama de actividades (Figura 33):

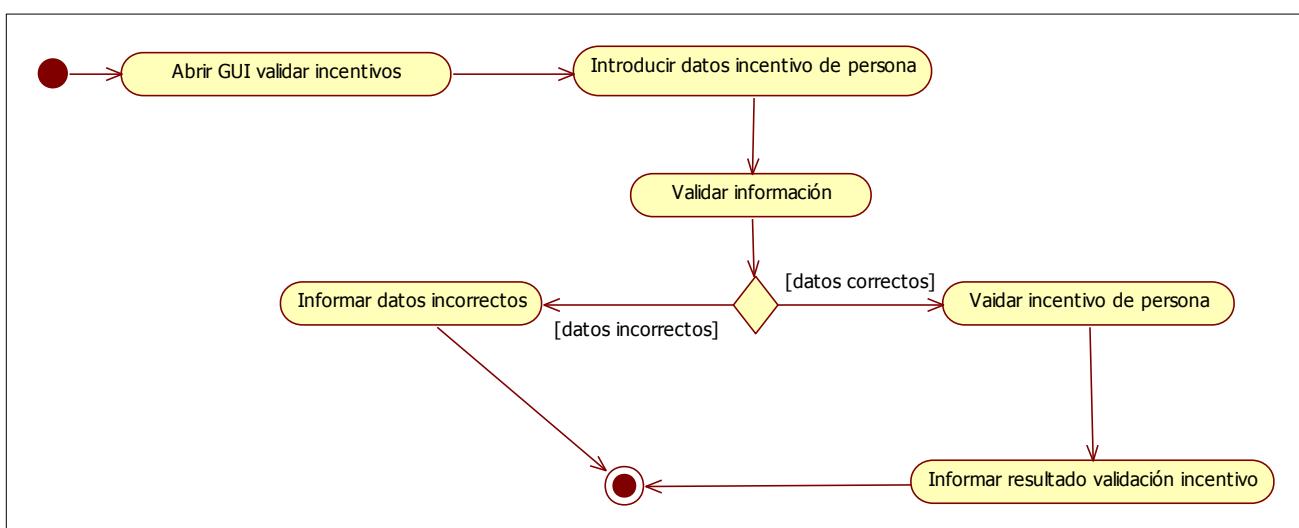


Figura 33: Diagrama de actividades del caso de uso ‘Validar incentivos’

4. Análisis de paquetes

4.1 Introducción

Una vez descritos los casos de uso, se procede a identificar las entidades relevantes y sus relaciones. Se van a definir paquetes de análisis para organizar las diferentes entidades y funcionalidades del sistema en partes más manejables.

4.2 Identificación de paquetes de análisis

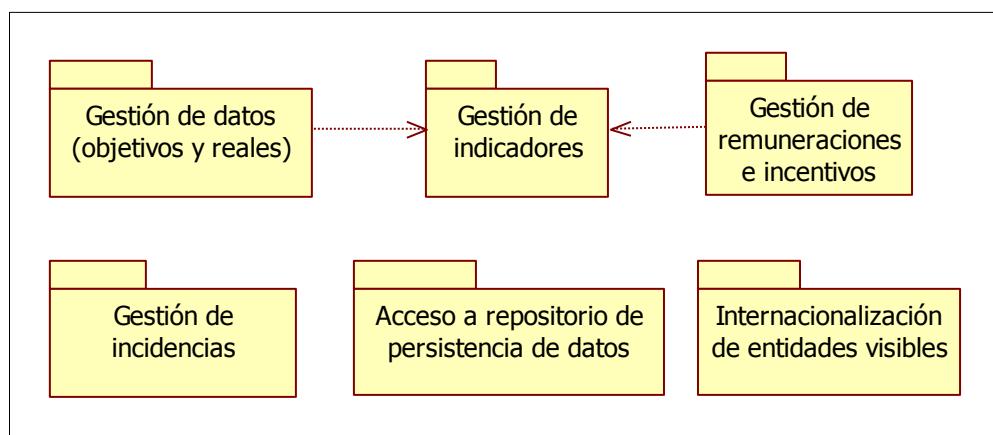


Figura 34: Paquetes de análisis del SGORI

4.2.1 Gestión de indicadores

El paquete de análisis ‘Gestión de indicadores’ engloba todas las funcionalidades relacionadas con la gestión de indicadores, agrupando los casos de uso ‘Añadir indicador’, ‘Eliminar indicador’, ‘Editar información del indicador’, ‘Gestión de personas implicadas en el indicador’, ‘Gestión de personas con privilegios sobre el indicador’ y ‘Gestión de barreras de entrada que afectan al indicador’ (Figura 35).

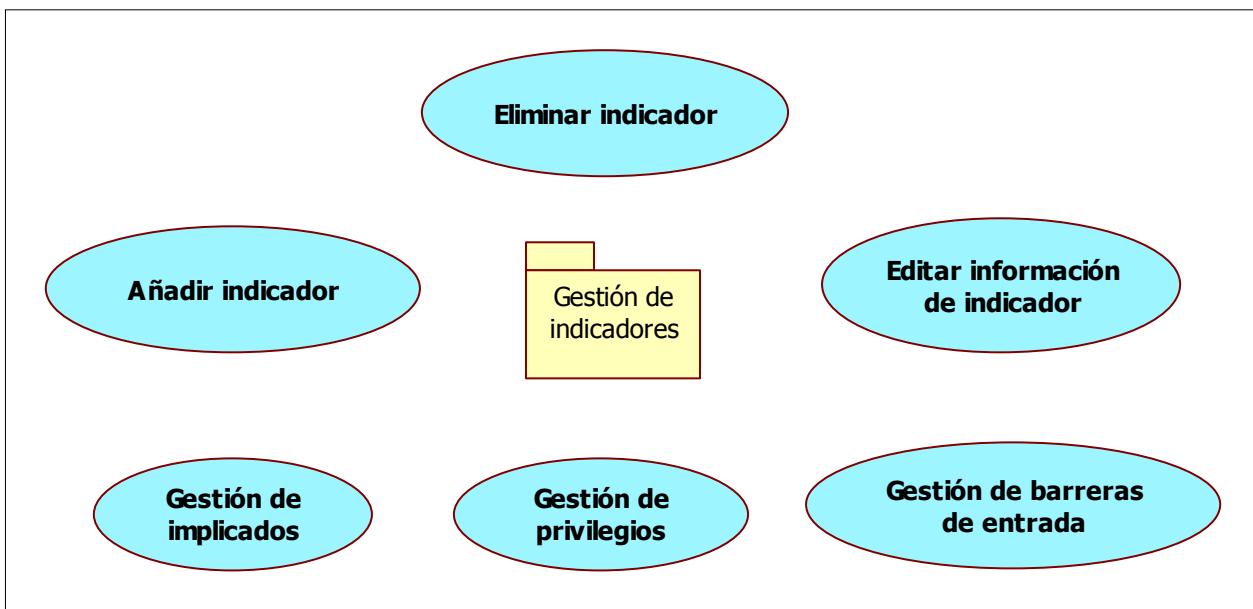


Figura 35: Paquete de análisis 'Gestión de indicadores'

4.2.2 Gestión de datos

Este paquete de análisis engloba todas las funcionalidades relacionadas con la gestión de datos, tanto objetivos como reales, asociados a los indicadores, agrupando los casos de uso 'Editar objetivos' y 'Editar reales' (Figura 36).

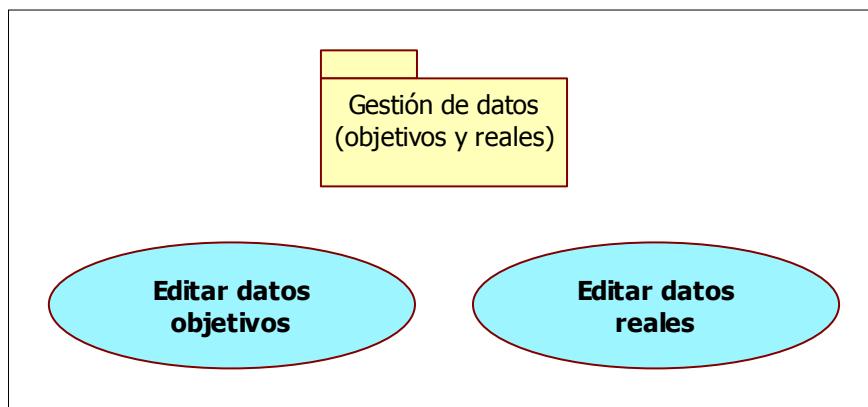


Figura 36: Paquete de análisis 'Gestión de datos'

4.2.3 Gestión de remuneraciones e incentivos

Este paquete de análisis engloba todas las funcionalidades relacionadas con la gestión de remuneraciones, agrupando los casos de uso 'Añadir remuneración', 'Eliminar remuneración', 'Editar remuneración' y 'Gestionar incentivos' (Figura 37).

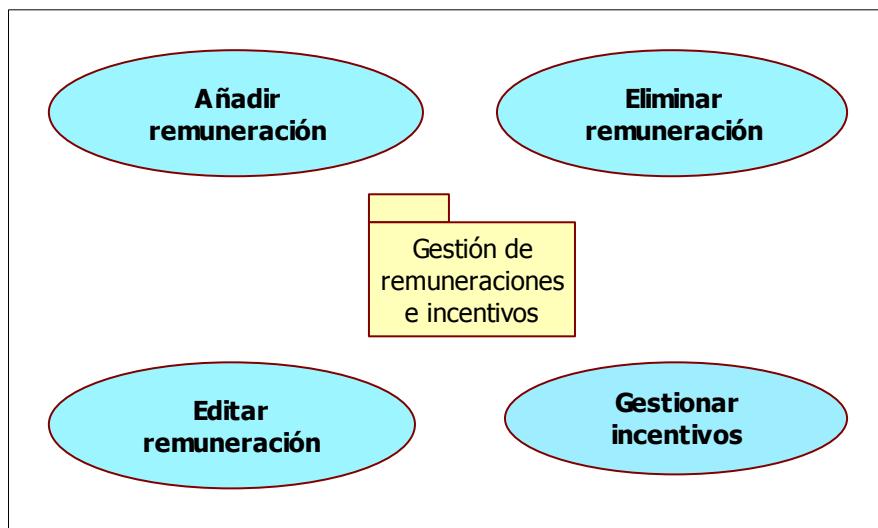


Figura 37: Paquete de análisis ‘Gestión de remuneraciones e incentivos’

4.2.4 Gestión de incidencias

Este paquete de análisis engloba la funcionalidad necesaria para dejar constancia de las incidencias que hubiere en el sistema para su correcta identificación y posterior resolución.

4.2.5 Acceso a repositorio de persistencia de datos

El paquete de análisis ‘Acceso a repositorio de persistencia de datos’ engloba la funcionalidad necesaria para asegurar la persistencia de las entidades del sistema que lo precisen. Este paquete de análisis no surge a partir de la fase de captura de requisitos, sino que proviene de las necesidades identificadas en el propio análisis del sistema, por lo que no contiene ningún caso de uso identificado.

4.2.6 Internacionalización de entidades visibles

Este paquete de análisis engloba la funcionalidad necesaria para que los textos, formato de fecha y resto de elementos que dependan del idioma o de la cultura, puedan ser definidos en función de estos.

5. Requerimientos especiales

En esta sección se describen requerimientos especiales identificados durante la fase de análisis y que son importantes para el sistema. Normalmente, no están referidos a funcionalidad final de cara al usuario, sino que son restricciones o necesidades propias del 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 se hará en fases posteriores al análisis. Asimismo, el medio sobre el cual se hará efectiva la capacidad persistente 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 la funcionalidad 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 y/o cultura del usuario. El paquete de análisis “Internacionalización de entidades visibles” será el que contenga la funcionalidad correspondiente a este punto.

6. Bibliografía

6.1 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.
- [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>
- [Ref. Web 4] <http://www.cs.ualberta.ca/~pfiguero/soo/uml/>
- [Ref. Web 5] pdfs.wke.es/4/8/0/0/pd0000014800.pdf

Sistema de Gestión de Objetivos y Retribución de Incentivos

Diseño

Versión 1.2

Publicado el 13/05/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
29/04/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
09/05/2013	1.1	Revisión del documento	Fernando Cortés Franco
13/05/2013	1.2	Corrección del documento	Santiago Pelegrín Costea

Índice

1. Introducción	6
1.1 Propósito del documento	6
1.2 Alcance del documento.....	6
1.3 Acrónimos.....	6
1.4 Definiciones	7
1.5 Referencias.....	7
1.6 Resumen.....	7
2. Descripción del proceso.....	9
3. Consideraciones iniciales.....	10
3.1 Introducción.....	10
3.2 Especificaciones tecnológicas	10
3.3 Especificaciones de diseño.....	10
3.4 Plataforma .NET.....	10
3.4.1 .Net Framework	10
3.4.2 Common Language Runtime.....	12
3.4.3 Biblioteca de Clases Base (BCL).....	12
3.5 SQL Server	14
3.6 NHibernate.....	14
4. Diseño de la arquitectura.....	17
4.1 Introducción.....	17
4.2 Estructura general del sistema	17
4.3 Estructura de capas del sistema.....	17
4.4 Estructura de subsistemas.....	18
4.4.1 Subsistema de gestión de indicadores	18
4.4.2 Subsistema de gestión de datos objetivos y reales.....	18
4.4.3 Subsistema de gestión de remuneraciones e incentivos.....	19
4.4.4 Subsistema de acceso a base de datos	19
4.4.5 Subsistema de gestión de incidencias.....	19
4.4.6 Subsistema de internacionalización 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 gestión de indicadores.....	21

5.2.2	Clases del subsistema de gestión de datos objetivos y reales	22
5.2.3	Clases del subsistema de gestión de remuneraciones e incentivos	22
5.3	Clases de acceso a datos	23
5.3.1	Obj_Ind_Indicator.cs.....	23
5.3.2	Obj_Ind_IndicatorData.cs	23
5.3.3	Obj_Ind_IndicatorRemuneration.cs.....	24
5.3.4	Obj_Ind_IndicatorEmployeePrivilege.cs.....	24
5.3.5	Obj_Ind_IndicatorArea.cs	24
5.3.6	Obj_Ind_IndicatorClosedRemuneration.cs.....	25
5.3.7	R_Indicator_Employee.cs.....	25
5.3.8	R_Indicator_EntryIndicador.cs	25
6.	Diseño de la base de datos	26
6.1	Introducción.....	26
6.2	Diseño general de la base de datos.....	26
6.3	Tablas de la base de datos	28
6.3.1	Obj_Ind_Indicator	28
6.3.2	Obj_Ind_IndicatorData.....	29
6.3.3	Obj_Ind_IndicatorRemuneration.....	29
6.3.4	Obj_Ind_IndicatorEmployeePrivileges.....	30
6.3.5	Obj_Ind_IndicatorClosedRemuneration	31
6.3.6	Obj_Ind_IndicatorAreas	31
6.3.7	R_Indicator_Employee.....	32
6.3.8	R_Indicator_EntryIndicador	32
7.	Prototipado de la interfaz	34
7.1	Introducción.....	34
7.2	Prototipos	34
8.	Bibliografía	37
8.1	Referencias.....	37
8.2	Referencias web.....	37

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

El alcance del documento comprende toda la fase de diseño del SGORI.

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.

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.
- SOAP: Simple Object Access Protocol.
- 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:

- ESTMER: estudio de mercado
- MODNEG: documento de modelo de negocio.
- ESPREQ: documento de especificación de requisitos.
- ANALISIS: documento de análisis del sistema.

1.6 Resumen

Este documento describe el proceso de diseño del SGORI. 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 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.

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 de la 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.

3. Consideraciones iniciales

3.1 Introducción

En esta sección se describen las primeras decisiones y especificaciones en el diseño del SGORI, 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 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 SGORI 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.

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...).

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.

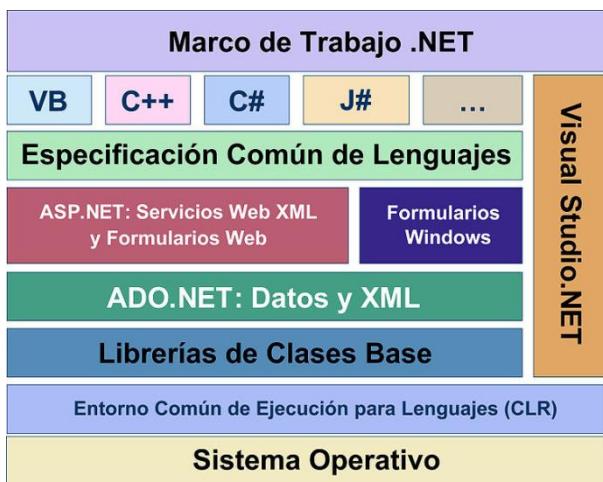


Figura 1: .NET Framework

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.
- El entorno 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.

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):

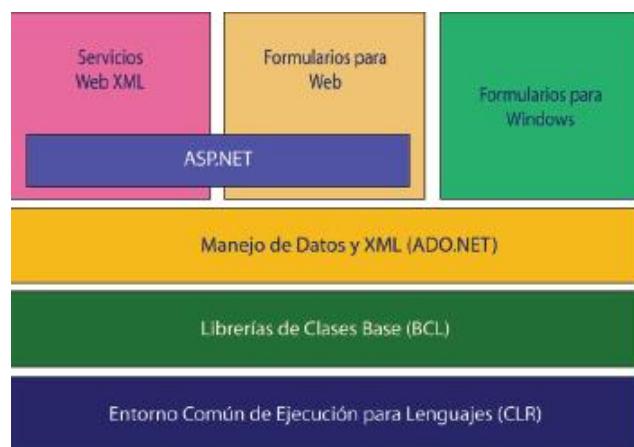


Figura 3: Esquema de distribución jerárquica de Namespaces

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 de un 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.
- 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.

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 (por ejemplo, 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 términos 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:

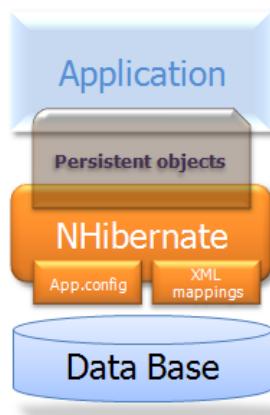


Figura 4: Arquitectura de NHibernate

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.

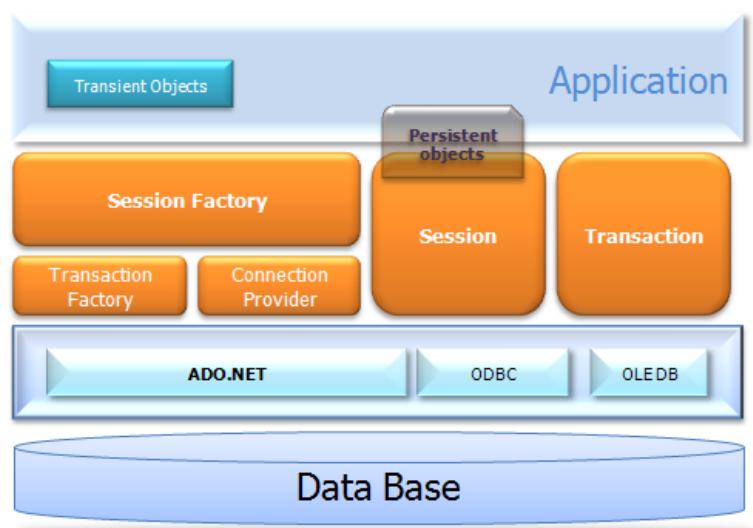


Figura 5: Arquitectura NHibernate detallada

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.

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 que se han tomado en dichos ámbitos, así como las restricciones que estaban impuestas por la organización en la que se desarrolla el SGORI.

4.2 Estructura general del sistema

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

- La aplicación de Gestión de Objetivos y Retribución de Incentivos (SGORI): los usuarios acceden al sistema, que se encontrará 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 2012.

En la siguiente figura (Figura 6) se representa 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 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 realizar 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.

En los siguientes subapartados se enumeran y explican los subsistemas de diseño que forman el SGORI.

4.4.1 Subsistema de gestión de indicadores

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en el paquete ‘Gestión de indicadores’, relacionadas con la existencia de la entidad ‘indicador’ (también existen las barreras de entrada, pero se consideran entidades indicador con algunas características especiales). Dichas funcionalidades son:

- Adición de nuevos indicadores al sistema.
- Edición de datos de indicadores.
- Eliminación de indicadores del sistema.
- Adición de personas implicadas en el indicador.
- Adición de personas con privilegios sobre el indicador.
- Adición de barreras de entrada que afectan al indicador.

4.4.2 Subsistema de gestión de datos objetivos y reales

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en el paquete ‘Gestión de datos (objetivos y reales)’, relacionadas con la existencia de las entidades ‘datos objetivos’ y ‘datos reales’. Dichas funcionalidades son:

- Edición de datos objetivos.
- Edición de datos reales.

4.4.3 Subsistema de gestión de remuneraciones e incentivos

Subsistema en el cual se desarrollan todas las funcionalidades identificadas en el paquete ‘Gestión de remuneraciones e incentivos’, relacionadas con la existencia de la entidad ‘remuneración’ (o retribución). Dichas funcionalidades son:

- Adición de nuevas remuneraciones al sistema.
- Edición de remuneraciones.
- Eliminación de remuneraciones del sistema.
- Cálculo de incentivos.
- Validación de incentivos

4.4.4 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.

4.4.5 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.

4.4.6 Subsistema de internacionalización 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.

5. Clases del sistema

5.1 Introducción

En este apartado se detallan las clases del SGORI. 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 SGORI. En la siguiente figura se observa el esquema general de clases de interfaz del sistema (Figura 8):

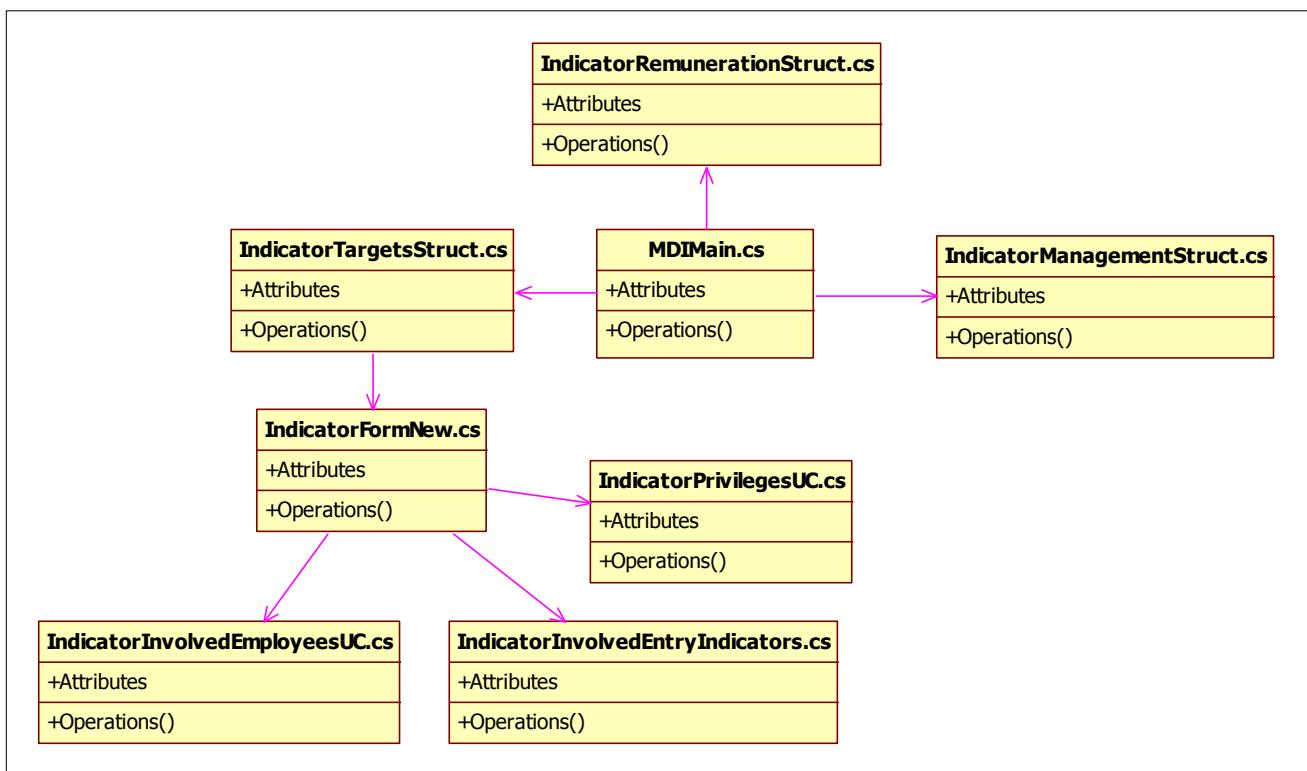


Figura 8: Esquema general de clases del sistema

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.

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

5.2.1 Clases del subsistema de gestión de indicadores

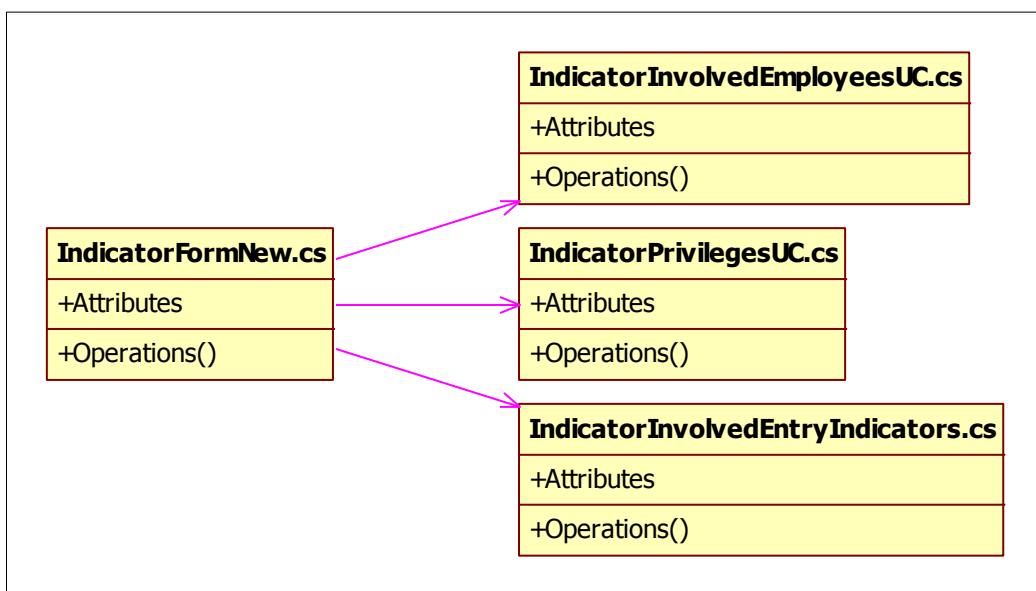


Figura 9: Clases del subsistema de gestión de indicadores

- **IndicatorFormNew.cs:** formulario que muestra la información de los indicadores, permitiendo su edición, así como el alta, baja y cambio de tipo (de indicador a barrera de entrada, o viceversa) de indicadores.
- **IndicatorInvolvedEmployeesUC.cs:** formulario que permite gestionar las personas implicadas en los indicadores.
- **IndicatorPrivilegesUC.cs:** formulario que permite la gestión de las personas que poseen privilegios de edición de los datos (objetivos y reales) sobre los indicadores.
- **IndicatorInvolvedEntryIndicators.cs:** formulario que permite gestionar los indicadores de tipo barrera de entrada que afectan a los resultados de los indicadores.

5.2.2 Clases del subsistema de gestión de datos objetivos y reales

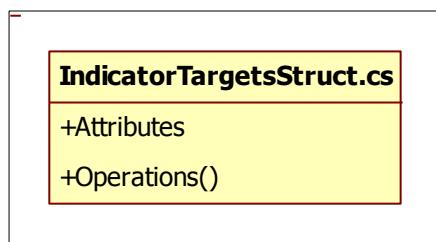


Figura 10: Clases del subsistema de gestión de datos objetivos y reales

- **IndicatorTargetsStruct.cs:** formulario que muestra la estructura de indicadores de la organización, junto con sus datos objetivos y reales asociados, permitiendo la gestión de dichos datos.

5.2.3 Clases del subsistema de gestión de remuneraciones e incentivos

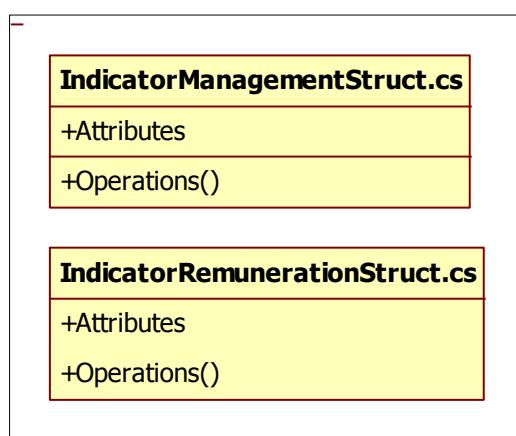


Figura 11: Clases del subsistema de gestión de remuneraciones e incentivos

- **IndicatorManagementStruct.cs:** formulario que muestra la estructura de remuneraciones asociadas a los indicadores, permitiendo la gestión de dichas remuneraciones.
- **IndicatorRemunerationStruct.cs:** formulario que muestra el listado de todos los empleados de la organización que se encuentran implicados en algún indicador, junto con los incentivos que les corresponden percibir por el alcance de los objetivos previamente definidos. Permite, además, revisar dichos incentivos y cambiarles el estado a ‘validados’.

5.3 Clases de acceso a datos

En esta sección se van a analizar las clases de acceso a datos del SGORI, describiendo los atributos de las mismas.

5.3.1 Obj_Ind_Indicator.cs

Es la clase de acceso a datos que representa la entidad ‘Indicador’. Sus atributos son:

Atributo	Descripción
_id	Identificador único del indicador
_name	Nombre del indicador
_code	Código del indicador
_status	Estado del indicador
_responsiblePerson	Persona responsable del indicador
_type	Tipo de indicador
_isIndividualIndicator	Indica si el indicador se comporta de manera individual o colectiva
_parentArea	Área a la que pertenece el indicador
_parentItem	Referencia al indicador padre

5.3.2 Obj_Ind_IndicatorData.cs

Es la clase de acceso a datos que representa los datos (tanto objetivos como reales) de los indicadores en el sistema. Sus atributos son:

Atributo	Descripción
_name	Referencia al indicador
_date	Fecha de consecución del dato objetivo y/o real
_target	Objetivo del indicador
_value	Valor real conseguido en el indicador
_updateDate	Fecha de última modificación de los datos
_lastUpdate	Persona que realizó la última modificación

5.3.3 Obj_Ind_IndicatorRemuneration.cs

Es la clase de acceso a datos que representa la entidad ‘Remuneración’. Sus atributos son:

Atributo	Descripción
_id	Identificador único de la remuneración
_indicator	Referencia al indicador
_period	Referencia al período en el que se aplica la remuneración
_indMinValue	Valor mínimo de los datos para conseguir la remuneración asociada
_indMaxValue	Valor máximo de los datos para conseguir la remuneración asociada
_totalRemuneration	Remuneración total a recibir

5.3.4 Obj_Ind_IndicatorEmployeePrivilege.cs

Es la clase de acceso a datos que representa la relación entre los empleados que poseen privilegios sobre los indicadores y dichos indicadores. Sus atributos son:

Atributo	Descripción
_name	Nombre identificador del privilegio
_indicator	Referencia al indicador
_employee	Referencia al empleado
_value	Indica el valor del permiso que posee el empleado sobre el indicador asociado

5.3.5 Obj_Ind_IndicatorArea.cs

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

Atributo	Descripción
_id	Identificador único del área
_parentArea	Referencia al área padre
_name	Nombre del área

5.3.6 Obj_Ind_IndicatorClosedRemuneration.cs

Es la clase de acceso a datos que representa la entidad ‘Incentivo’. Sus atributos son:

Atributo	Descripción
_monthlyRemuneration	Incentivo mensual conseguido
_employee	Referencia al empleado que ha conseguido el incentivo asociado
_date	Fecha de obtención del incentivo

5.3.7 R_Indicator_Employee.cs

Es la clase de acceso a datos que representa la relación entre ‘Empleado’ e ‘Indicador’. Sus atributos son:

Atributo	Descripción
_indicator	Referencia al indicador
_employee	Referencia al empleado
_percentWeight	Indica el peso o la ponderación en porcentaje que el indicador posee sobre el empleado

5.3.8 R_Indicator_EntryIndicator.cs

Es la clase de acceso a datos que representa la relación entre los indicadores de tipo barrera de entrada y el resto de indicadores. Sus atributos son:

Atributo	Descripción
_entryIndicator	Referencia al indicador de tipo barrera de entrada
_affectedIndicator	Referencia al indicador afectado por la barrera de entrada asociada

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 SGORI. 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

A continuación se muestra un esquema general de la base de datos del SGORI (Figura 12):

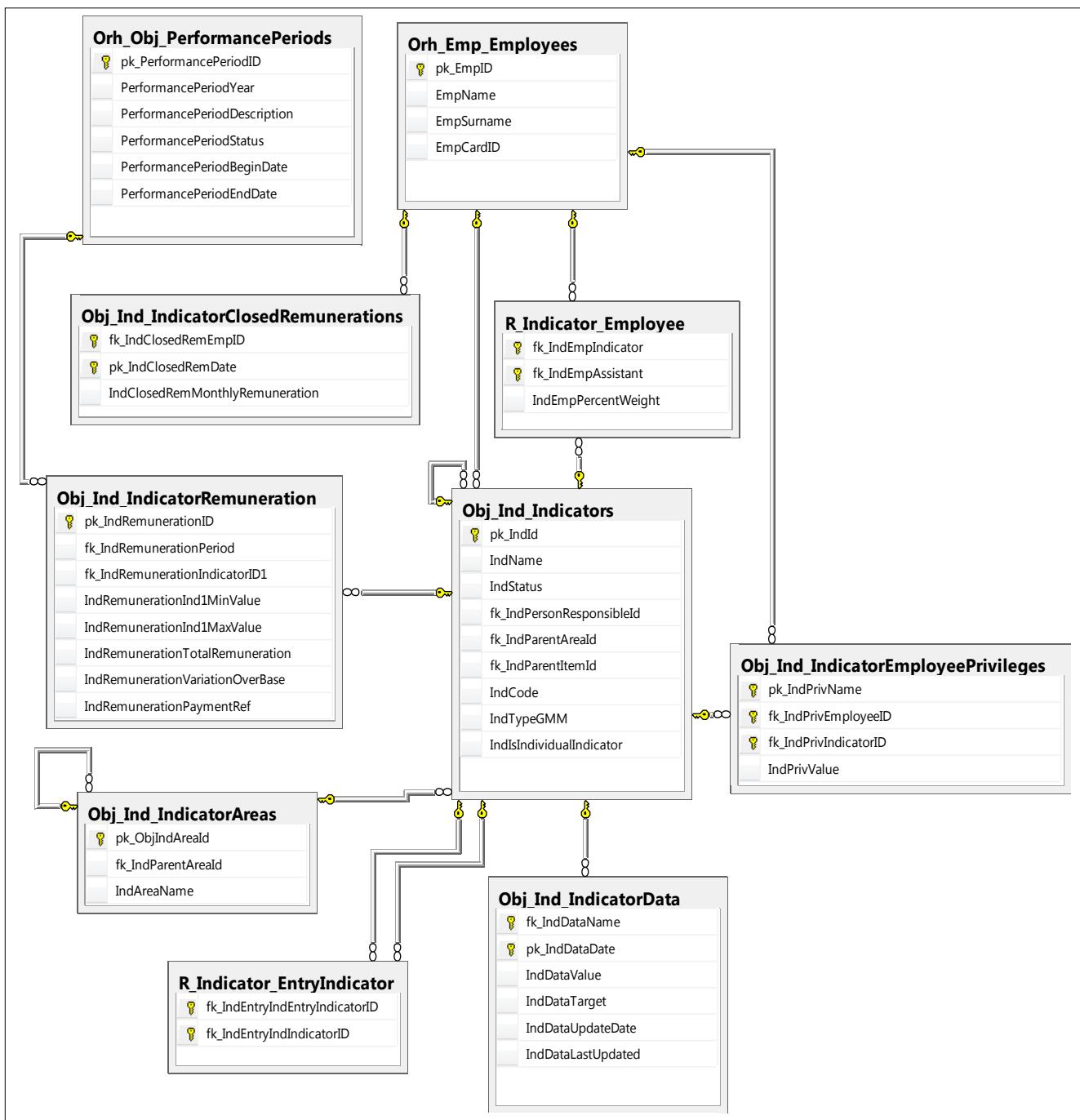


Figura 12: Esquema general de la base de datos del SGORI

Las tablas ‘Orh_Emp_Employees’ y ‘Orh_Obj_PerformancePeriods’ son tablas ajenas al SGORI, pero que guardan relación con alguna de las tablas del mismo. Por ello, se han incluido en la vista del esquema general de la base de datos, pero no se han explicado ni comentado sus características en el siguiente apartado.

6.3 Tablas de la base de datos

6.3.1 Obj_Ind_Indicator

Obj_Ind_Indicators				Descripción
Nombre de columna	Tipo de datos	Permitir valores NULL		
pk_IndId	int	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Clave primaria: ID del indicador.
IndName	nvarchar(200)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre del indicador.
IndStatus	nvarchar(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Estado del indicador.
fk_IndPersonResponsibleId	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Clave ajena: empleado responsable del indicador.
fk_IndParentAreaId	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Clave ajena: área a la que pertenece el indicador.
fk_IndParentItemId	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Clave ajena: indicador padre del indicador.
IndCode	nvarchar(15)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Código del indicador.
IndType	int	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Tipo de indicador.
IndIsIndividualIndicator	bit	<input type="checkbox"/>	<input type="checkbox"/>	Marca si el indicador tiene comportamiento individual o cole...
			<input type="checkbox"/>	
			<input type="checkbox"/>	

Figura 13: Tabla Obj_Ind_Indicator

Relaciones:

- El campo fk_IndPersonResponsibleId se corresponde con el campo pk_EmplId de la tabla ajena al SGORI Orh_Emp_Employees.
- El campo fk_IndParentAreaId se corresponde con el campo pk_ObjIndAreaId de la tabla Obj_Ind_IndicatorAreas.
- El campo fk_IndParentItemId se corresponde con el campo pk_IndId de la propia tabla (Obj_Ind_Indicators).

Restricciones:

- El campo pk_IndId debe ser único para cada elemento de la tabla.
- El campo IndName debe ser único para cada elemento de la tabla.
- El campo IndStatus puede tomar los siguientes valores:
 - ‘A’: activo.
 - ‘H’: histórico.
- El campo IndType puede tomar los siguientes valores:
 - 1: indicador de tipo objetivo.
 - 2: indicador de tipo unitario.
 - 3: indicador de tipo barrera de entrada.

6.3.2 Obj_Ind_IndicatorData

Obj_Ind_IndicatorData				
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción	
 fk_IndDataName	nvarchar(200)	<input type="checkbox"/>	Clave ajena: referencia al nombre del indicador.	
 pk_IndDataDate	datetime	<input type="checkbox"/>	Clave primaria: fecha de consecución del dato.	
IndDataValue	float	<input checked="" type="checkbox"/>	Valor real obtenido.	
IndDataTarget	float	<input checked="" type="checkbox"/>	Objetivo prefijado.	
IndDataUpdateDate	datetime	<input checked="" type="checkbox"/>	Fecha de última modificación.	
IndDataLastUpdated	nvarchar(200)	<input checked="" type="checkbox"/>	Nombre del usuario que realizó la últ. modif.	
		<input type="checkbox"/>		

Figura 14: Tabla Obj_Ind_IndicatorData

Relaciones:

- El campo fk_IndDataName se corresponde con el campo IndName de la tabla Obj_Ind_Indicators.

Restricciones:

- La combinación de los valores de los campos pk_IndDataDate y fk_IndDataName debe ser única para cada elemento de la tabla.

6.3.3 Obj_Ind_IndicatorRemuneration

Obj_Ind_IndicatorRemuneration				
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción	
 pk_IndRemunerationID	int	<input type="checkbox"/>	Clave primaria: ID de la remuneración	
fk_IndRemunerationPeriod	int	<input type="checkbox"/>	Clave ajena: período en el que se aplica la remu...	
fk_IndRemunerationIndicatorID1	int	<input type="checkbox"/>	Clave ajena: indicador asociado a la remuneraci...	
IndRemunerationInd1MinValue	float	<input type="checkbox"/>	Valor mínimo a conseguir para optar a la remu...	
IndRemunerationInd1MaxValue	float	<input checked="" type="checkbox"/>	Valor máximo a conseguir para optar a la remu...	
IndRemunerationTotalRemuneration	float	<input checked="" type="checkbox"/>	Remuneración.	
		<input type="checkbox"/>		

Figura 15: Tabla Obj_Ind_IndicatorRemuneration

Relaciones:

- El campo fk_IndRemunerationPeriod se corresponde con el campo pk_PerformancePeriodId de la tabla ajena al SGORI Orh_Obj_PerformancePeriods.
- El campo fk_IndRemunerationIndicatorID1 se corresponde con el campo pk_IndId de la tabla Obj_Ind_Indicators.

Restricciones:

- El campo pk_IndRemunerationID debe ser único para cada elemento de la tabla.
- El valor del campo IndRemunerationInd1MaxValue debe ser mayor o igual al valor del campo IndRemunerationInd1MinValue.

6.3.4 Obj_Ind_IndicatorEmployeePrivileges

Obj_Ind_IndicatorEmployeePrivileges			
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción
pk_IndPrivName	nvarchar(50)	<input type="checkbox"/>	Clave primaria: nombre identificador del permiso.
fk_IndPrivEmployeeID	int	<input type="checkbox"/>	Clave ajena: empleado con privilegios.
fk_IndPrivIndicatorID	int	<input type="checkbox"/>	Clave ajena: indicador sobre el que poseer privilegios.
IndPrivValue	bit	<input type="checkbox"/>	Valor del privilegio
		<input type="checkbox"/>	
		<input type="checkbox"/>	
		<input type="checkbox"/>	

Figura 16: Tabla Obj_Ind_IndicatorEmployeePrivileges

Relaciones:

- El campo fk_IndPrivEmployeeID se corresponde con el campo pk_EmplID de la tabla ajena al SGORI Orh_Emp_Employees.
- El campo fk_IndPrivIndicatorID se corresponde con el campo pk_IndId de la tabla Obj_Ind_Indicators.

Restricciones:

- La combinación de los valores de los campos pk_IndPrivName, fk_IndPrivEmployeeID y fk_IndPrivIndicatorID debe ser única para cada elemento de la tabla.

6.3.5 Obj_Ind_IndicatorClosedRemuneration

Obj_Ind_IndicatorClosedRemunerations			
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción
fk_IndClosedRemEmpID	int	<input type="checkbox"/>	Clave ajena: empleado que recibe el incentivo.
pk_IndClosedRemDate	smalldateti...	<input type="checkbox"/>	Clave primaria: fecha de obtención del incentivo.
IndClosedRemMonthlyRemuneration	float	<input checked="" type="checkbox"/>	Cantidad total del incentivo a recibir.
		<input type="checkbox"/>	

Figura 17: Tabla Obj_Ind_IndicatorClosedRemuneration

Relaciones:

- El campo fk_IndClosedRemEmpID se corresponde con el campo pk_EmpID de la tabla ajena al SGORI Orh_Emp_Employees.

Restricciones:

- La combinación de los valores de los campos pk_IndClosedRemDate y fk_IndClosedRemEMpID debe ser única para cada elemento de la tabla.

6.3.6 Obj_Ind_IndicatorAreas

Obj_Ind_IndicatorAreas			
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción
pk_ObjIndAreaId	int	<input type="checkbox"/>	Clave primaria: ID del área.
fk_IndParentAreaId	int	<input checked="" type="checkbox"/>	Clave ajena: referencia al área padre.
IndAreaName	nvarchar(100)	<input type="checkbox"/>	Nombre del área.
		<input type="checkbox"/>	

Figura 18: Tabla Obj_Ind_IndicatorAreas

Relaciones:

- El campo fk_IndParentAreaId se corresponde con el campo pk_Obj_IndAreaId de la propia tabla (Obj_Ind_IndicatorAreas)

Restricciones:

- El campo pk_ObjIndAreaID debe ser único para cada elemento de la tabla.
- El campo IndAreaName debe ser único para cada elemento de la tabla.

6.3.7 R_Indicator_Employee

R_Indicator_Employee			
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción
fk_IndEmpIndicator	int	<input type="checkbox"/>	Clave ajena: indicador.
fk_IndEmpAssistant	int	<input type="checkbox"/>	Clave ajena: empleado
IndEmpPercentWeight	float	<input checked="" type="checkbox"/>	Indica el peso que posee el indicador sobre el emplea...
		<input type="checkbox"/>	
		<input type="checkbox"/>	

Figura 19: Tabla R_Indicator_Employee

Relaciones:

- El campo fk_IndEmpIndicator se corresponde con el campo pk_IndID de la tabla Obj_Ind_Indicators.
- El campo fk_IndEmpAssistant se corresponde con el campo pk_EmpID de la tabla ajena al SGORI Orh_Emp_Employees.

Restricciones:

- La combinación de los valores de los campos fk_IndEmpIndicator y fk_IndEmpAssistant debe ser única para cada elemento de la tabla.

6.3.8 R_Indicator_EntryIndicator

R_Indicator_EntryIndicator			
Nombre de columna	Tipo de datos	Permitir valores NULL	Descripción
fk_IndEntryIndEntryIndicatorID	int	<input type="checkbox"/>	Clave ajena: indicador de tipo barrera de entrada.
fk_IndEntryIndIndicatorID	int	<input type="checkbox"/>	Clave ajena: indicador de tipo objetivo o unitario.
		<input type="checkbox"/>	

Figura 20: Tabla R_Indicator_EntryIndicator

Relaciones:

- El campo fk_IndEntryIndEntryIndicatorID se corresponde con el campo pk_IndId de la tabla Obj_Ind_Indicators.
- El campo fk_IndEntryIndIndicatorID se corresponde con el campo pk_IndId de la tabla Obj_Ind_Indicators.

Restricciones:

- La combinación de los valores de los campos fk_IndEntryIndEntryIndicatorID y fk_IndEntryIndIndicatorID debe ser única para cada elemento de la tabla.
- El campo fk_IndEntryIndEntryIndicatorID debe apuntar a un elemento de la tabla Obj_Ind_Indicators cuyo valor del campo IndType sea igual a 3 (barrera de entrada).
- El campo fk_IndEntryIndIndicatorID debe apuntar a un elemento de la tabla Obj_Ind_Indicators cuyo valor del campo IndType sea igual a 1 o a 2 (indicador objetivo o unitario).

7. Prototipado de la interfaz

7.1 Introducción

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 prototipos con 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.

7.2 Prototipos

En la siguiente figura se observa el esquema general de todas las interfaces de la aplicación. Podemos observar tres partes diferenciadas (Figura 21):

- Menú del SGORI: 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.

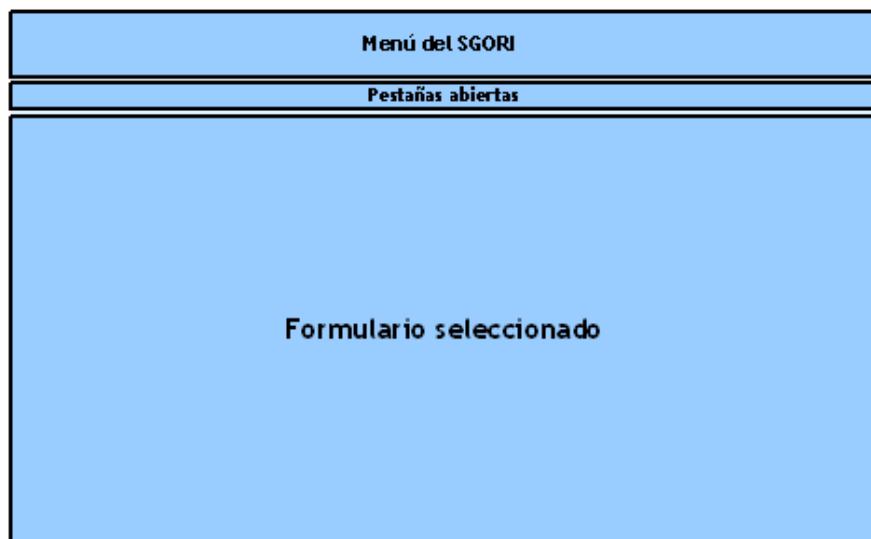


Figura 21: Esquema general de la interfaz

En la siguiente figura (Figura 22) 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: IndicatorTargetsStruct.cs, IndicatorManagementStruct.cs e IndicatorRemunerationStruct.cs.



Figura 22: Interfaz tipo barra de herramientas + contenido

En la siguiente figura (Figura 23) observamos un prototipo de interfaz menos genérico, ya que está asociado a un único formulario: IndicatorNewForm.cs. Este formulario muestra la ficha de un indicador. Consta de cuatro secciones:

- Barra de herramientas del formulario.
- Información básica. En ella aparecerán datos básicos del indicador, que serán visibles en todo momento, tales como: nombre, código, persona responsable o tipo de indicador.
- 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 indicador. Dichas secciones serán: personas implicadas en el indicador, personas con privilegios sobre el indicador y barreras de entrada que afectan al indicador.
- En la parte inferior de la pantalla se visualizará la información asociada a la sección seleccionada por el usuario.

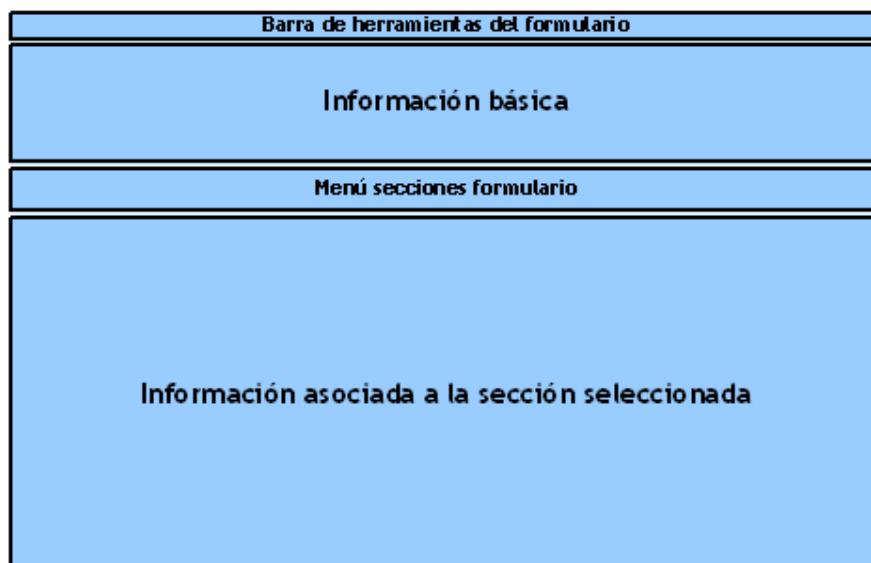


Figura 23: Interfaz de ficha de indicador

8. Bibliografía

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 Kuaté, T. Harris, C. Bauer, G. King. "NHibernate in Action". Manning Publications 2008.

8.2 Referencias web

- [Ref. Web 1] <http://www.microsoft.com/net>
- [Ref. Web 2] <http://www.wikipedia.org/>
- [Ref. Web 3] <http://www.uml.org/>
- [Ref. Web 4] <http://www.rational.com/>
- [Ref. Web 5] <http://www.hibernate.org/>
- [Ref. Web 6] <http://nhforge.org>

Sistema de Gestión de Objetivos y Retribución de Incentivos

Implementación

Versión 1.2

Publicado el 01/07/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
24/06/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
28/06/2013	1.1	Revisión del documento	Fernando Cortés Franco
01/07/2013	1.2	Corrección del documento	Santiago Pelegrín Costea

Índice

1. Introducción	6
1.1 Propósito del documento	6
1.2 Alcance del documento.....	6
1.3 Acrónimos.....	6
1.4 Definiciones	6
1.5 Referencias.....	7
1.6 Resumen.....	7
2. Descripción del proceso.....	8
3. Tecnologías, herramientas y lenguajes.....	9
4. Implementación de la internacionalización	10
4.1 Definición.....	10
4.2 Objetivos de la internacionalización.....	10
4.3 Elementos a internacionalizar.....	10
4.4 Reglas clave de desarrollo para la internacionalización	10
4.5 Proceso de internacionalización.....	11
5. Implementación de la gestión de incidencias	13
6. Implementación del acceso a base de datos	15
6.1 Introducción.....	15
6.2 SQL Server 2012	15
6.3 NHibernate.....	15
6.3.1 Configuración de NHibernate.....	16
6.3.2 Implementar el modelo de dominio.....	17
6.3.3 Mapear el modelo de dominio.....	21
6.3.4 Manipulación de datos persistentes	22
6.3.5 El Gestor de persistencia	24
6.3.6 Procedimientos almacenados.....	24
7. Implementación de la interfaz de usuario.....	28
7.1 Introducción.....	28
7.2 Elementos de interfaz.....	28
7.2.1 Iconos e imágenes.....	28
7.2.2 Tablas	28
7.2.3 Árboles.....	29
7.2.4 Pestañas	29

7.2.5	Editores de Fechas.....	29
7.3	Pantallas del sistema.....	30
7.3.1	Gestión de indicadores.....	30
7.3.2	Gestión de datos (objetivos y reales)	32
7.3.3	Gestión de remuneraciones e incentivos	33
8.	Bibliografía	36
8.1	Referencias.....	36
8.2	Referencias web.....	36

1. Introducción

1.1 Propósito del documento

El presente documento describe la fase de implementación del proyecto de desarrollo del SGORI. 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 del documento

El alcance del documento comprende la fase de implementación del SGORI, en la fase final de desarrollo del sistema.

1.3 Acrónimos

- ADO: ActiveX Data Objects.
- ANSI: Amercian 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.
- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos
- 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.

- Transacción: una transacción en un Sistema de Gestión de Bases de Datos es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.
- 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.

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.
- ESTCOD.doc: documento que especifica el estándar de codificación utilizado en el proyecto.

1.6 Resumen

Este documento describe el proceso de implementación del SGORI. Se compone de ocho apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describe el proceso de implementación seguido.
- Apartado 3: Se describen las tecnologías, herramientas y lenguajes empleados durante la implementación del sistema.
- Apartado 4: Se describe el proceso seguido para la implementación de la internacionalización de la aplicación.
- Apartado 5: Se describe el proceso seguido para la gestión de incidencias ocurridas en el sistema.
- Apartado 6: Se detallan las consideraciones necesarias para la implementación del acceso a datos.
- Apartado 7: Se describe el proceso seguido para la implementación del interfaz de usuario.
- Apartado 8: Bibliografía y referencias web utilizadas para la realización de este documento.

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.

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 2012. 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 2010. Entorno de programación y depuración de código de la plataforma .NET.
- Microsoft SQL Management Studio 2012. Herramienta gráfica que permite realizar tareas de mantenimiento de la base de datos sobre SQL Server 2012.
- Microsoft Team Foundation Server 2010. 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 2010.1. 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.

4. Implementación de la internacionalización

4.1 Definición

Uno de los requisitos iniciales planteados para el SGORI 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 SGORI 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.
- Monedas: pueden variar los símbolos monetarios que son utilizados (€, \$....).

4.4 Reglas clave de desarrollo para 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 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 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:

```
IndicatorNewForm_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 de la aplicación en que vayamos a utilizar los recursos hay que crear un campo de tipo cadena de texto con un modificador de acceso privado 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 IndicatorNewForm_btn_Ok
```

- 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 = IndicatorNewForm_lbl_neoTxtName_Label;  
neoTxtCode.Label = IndicatorNewForm_lbl_neoTxtCode_Label;
```

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).
- Etiqueta o *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("IndicatorFormNew.toolbar1_ToolClick: Tool= " + e.Tool.Key, Log.DEBUG);
```

A continuación, mostramos un ejemplo del contenido del fichero de log:

```
12/07/2013 [11:07:18]      ERROR      OSUser:      spelegrin;      AppUser:      DEMO;
Obj_Ind_Indicator.Name Asignación de NULL a campo no anulable

12/07/2013 [11:07:19]  DEBUG  OSUser: spelegrin; AppUser: DEMO ; IndicatorFormNew.
IndicatorFormNew indicatorID=8

12/07/2013 [11:07:19]  DEBUG  OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.InitializeLabels

12/07/2013 [11:07:19]  DEBUG  OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.LoadI18N
```

```
12/07/2013 [11:07:19] DEBUG OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.InitializeHeader

12/07/2013 [11:07:19] DEBUG OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.header_Modified

12/07/2013 [11:07:19] DEBUG OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.SetHeaderData

12/07/2013 [11:07:19] DEBUG OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.InitializeInfoTab

12/07/2013 [11:07:20] DEBUG OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.ultraToolbarsManager1_ToolClick: Tool = Save

12/07/2013 [11:07:20] DEBUG OSUser: spelegrin; AppUser: DEMO;
IndicatorFormNew.UpdateHeaderInfo
```

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 2012, 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 SGORI.

6.2 SQL Server 2012

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 sistemas 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 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:nhibernate-configuration-2.2">
  <session-factory>
    <property name="dialect">NHibernate.Dialect.MsSql2012Dialect</property>
    <property
      name="connection.provider">NHibernate.Connection.DriverConnectionProvider</proper
      ty>
      <property
        name="connection.driver_class">NHibernate.Driver.SqlClientDriver</property>
        <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:

- 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 “Obj_Ind_Indicator.cs”, que almacena los datos de los indicadores. El fichero de clase “Obj_Ind_Indicator.cs” es el siguiente:

```
using integradbaccess.Core;
using integradbaccess.util;
using System;
using System.Collections.Generic;

namespace integradbaccess.Domain
{
    /// <summary>
    /// Clase de acceso a datos mapeada para NHibernate de la tabla
    Obj_Ind_Indicators
    /// </summary>
    [Serializable]
    public class Obj_Ind_Indicator : DomainObject<int>
    {
```

```

#region Constantes

public struct Constants
{
    public struct Status
    {
        public const char ACTIVE = 'A';
        public const char HISTORIC = 'H';
    }

    public struct Type
    {
        public const int TARGET_IND = 1;
        public const int UNIT_INDICATOR = 2;
        public const int ENTRY_INDICATOR = 3;
    }
}

/// <summary>
/// Nombres de las propiedades de la clase Obj_Ind_Indicator
/// </summary>
public struct Properties
{
    public const string Id = "ID";
    public const string Name = "Name";
    public const string Status = "Status";
    public const string Code = "Code";
    public const string ResponsiblePerson = "ResponsiblePerson";
    public const string ParentArea = "ParentArea";
    public const string Type = "Type";
    public const string IsIndividualIndicator = "IsIndividualIndicator";
    public const string ParentItem = "ParentItem";
}

/// <summary>
/// Restricciones de las propiedades de la clase Obj_Ind_Indicator
/// </summary>
public struct Constraints
{
    public const int Name_MAXLENGTH = 200;
    public const int Code_MAXLENGTH = 15;
}

#endregion Constantes

#region Atributos

private string _name;
private char _status;
private string _code;
private Orh_Emp_Employee _responsiblePerson;
private Obj_Ind_IndicatorArea _parentArea;
private int? _type;
private bool _isIndividualIndicator;
private Obj_Ind_Indicator _parentItem;

#endregion Atributos

```

```

#region Constructores

/// <summary>
/// Constructor de Obj_Ind_Indicator
/// </summary>
public Obj_Ind_Indicator()
{
}

/// <summary>
/// Constructor de Obj_Ind_Indicator con ID
/// </summary>
public Obj_Ind_Indicator(int id)
{
    base.ID = id;
}

#endregion Constructores

#region Propiedades

public virtual string Name {
    get { return _name; }
    set
    {
        if(value == null)
        {
            string errorStr = "Obj_Ind_Indicator.Name Asignación de NULL a campo no anulable";

            Log.Write(errorStr, Log.ERROR);
        }
        else if(value.Length > Constraints.Name_MAXLENGTH)
        {
            string errorStr = "Obj_Ind_Indicator.Name MAX LEN(" +
Constraints.Name_MAXLENGTH + ") Excedida: " + value.Length + "; Valor: " + value;

            Log.Write(errorStr, Log.ERROR);
        }
        else
        {
            _name = value;
        }
    }
}

public virtual char Status {
    get { return _status; }
    set { _status = value; }
}

public virtual int? Type
{
    get { return _type; }
    set { _type = value; }
}

```

```

public virtual string Code {
    get { return _code; }
    set
    {
        if((value != null) && (value.Length > Constraints.Code_MAXLENGTH))
        {
            string errorStr = "Obj_Ind_Indicator.Code MAX LEN(" +
Constraints.Code_MAXLENGTH + ") Excedida: " + value.Length + "; Valor: " + value;
            Log.Write(errorStr, Log.ERROR);
        }
        else
        {
            _code = value;
        }
    }
}

/// <summary>
/// Obtiene un Orh_Emp_Employee a través de la relación
FK_Obj_Ind_Indicators_Orh_Emp_Employees
/// </summary>
public virtual Orh_Emp_Employee ResponsiblePerson
{
    get { return _responsiblePerson; }
    set { _responsiblePerson = value; }
}

/// <summary>
/// Obtiene un Obj_Ind_IndicatorArea a través de la relación
FK_Obj_Ind_Indicators_Obj_Ind_IndicatorAreas
/// </summary>
public virtual Obj_Ind_IndicatorArea ParentArea
{
    get { return _parentArea; }
    set { _parentArea = value; }
}

public virtual bool IsIndividualIndicator
{
    get { return _isIndividualIndicator; }
    set { _isIndividualIndicator = value; }
}

/// <summary>
/// Obtiene un Obj_Ind_Indicator a través de la relación
FK_Obj_Ind_Indicators_Obj_Ind_Indicators
/// </summary>
public virtual Obj_Ind_Indicator ParentItem
{
    get { return _parentItem; }
    set { _parentItem = value; }
}

#endregion Propiedades

```

```

    ///<summary>
    /// Obtiene el código hash de un Obj_Ind_Indicator
    ///</summary>
    public override int GetHashCode()
    {
        return ID.GetHashCode();
    }

}

```

6.3.3 Mapear el modelo de dominio

El fichero de mapeo “Obj_Ind_Indicator.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="Obj_Ind_Indicator" table="Obj_Ind_Indicators" dynamic-
update="true" >
        <id name="ID" type="int" column="pk_IndId">
            <generator class="identity"/>
        </id>
        <property name="Name" column="IndName" not-null="true" />
        <property name="Status" column="IndStatus" not-null="true" />
        <property name="Code" column="IndCode" not-null="false" />
        <property name="TypeGMM" column="IndTypeGMM" not-null="false" />
        <property name="IsIndividualIndicator" column="IndIsIndividualIndicator" not-
null="true" />
        <many-to-one name="ResponsiblePerson" class="Orh_Emp_Employee" >
            <column name="fk_IndPersonResponsibleId" />
        </many-to-one>
        <many-to-one name="ParentArea" class="Obj_Ind_IndicatorArea" >
            <column name="fk_IndParentAreaId" />
        </many-to-one>
        <many-to-one name="ParentItem" class="Obj_Ind_Indicator" >
            <column name="fk_IndParentItemId" />
        </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 Obj_Ind_Indicator es mapeada a la clase del mismo nombre. Cada fila de dicha tabla representa una instancia de tipo Obj_Ind_Indicator.
- Cada registro de la tabla Obj_Ind_Indicator 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 Name, por ejemplo, es mapeada a la columna de base de datos IndName. 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 Obj_Ind_Indicator contiene una columna llamada fk_IndPersonResponsibleId, que es una clave ajena a la tabla Orh_Emp_Employees. Dicha asociación es mapeada en el fichero xml del siguiente modo:

```
<many-to-one name="ResponsiblePerson" class="Orh_Emp_Employee" >
  <column name="fk_IndPersonResponsibleId" />
</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. Por tanto, es 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) “*persistent*” (persistente) y “*detached*” (separado, distante).

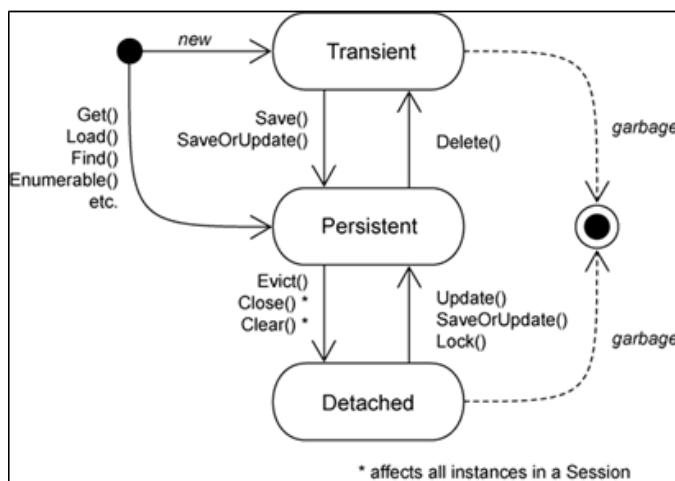


Figura 1: Ciclo de vida de la persistencia de un objeto en NHibernate

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*).
- 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.

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). Las instancias “*detached*” no tienen por qué mantener su sincronización con el estado de base de datos, ya que 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” 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.
- 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 procedimiento almacenado del siguiente modo:

```
public List<Orh_Emp_Employee> GetInvolvedEmployeesInfo ()  
{  
    try  
    {  
        IQuery namedQuery = new  
BurrowFramework().GetSession().GetNamedQuery("ListInvolvedEmployeesInfo");  
    }  
}
```

```

    return namedQuery.List<Orh_Emp_EmployeeFile>() as
List<Orh_Emp_EmployeeFile>;
}
catch
{
    return null;
}
}
  
```

- En el fichero de mapeo xml, deberá estar mapeado el procedimiento almacenado a invocar:

```

<sql-query name="ListInvolvedEmployeesInfo" >
  <return class="Orh_Emp_Employee"/>
  exec I_Indicator_GetInvolvedEmployeesInfo
</sql-query>
  
```

Donde el procedimiento I_Indicator_GetInvolvedEmployeesInfo es el siguiente:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[I_Indicator_GetInvolvedEmployeesInfo]
(
)
AS

;WITH InvolvedEmployees (EmpID, EmpFullName, EmpCode, EmpWorkCenter, EmpLevel, EmpCategory)
AS
(
  SELECT DISTINCT pk_EmpID EmpID,
                 EmpSurname + ', ' + EmpName EmpFullName,
                 EmpCode EmpCode,
                 EMPWORKCENTER.AdmName,
                 EMPLLEVEL.AdmName,
                 EMPCATEGORY.AdmName
        FROM dbo.Orh_Emp_Employees
       INNER JOIN dbo.R_Indicator_Employee ON (fk_IndEmpAssistant = pk_EmpID)
       INNER JOIN dbo.Obj_Ind_Indicators IND ON (fk_IndEmpIndicator = pk_IndID)
       LEFT JOIN Gen_AdministrationItems EMPWORKCENTER ON (fk_EmpWorkCenter =
EMPWORKCENTER.pk_AdmID)
       LEFT JOIN Gen_AdministrationItems EMPLLEVEL ON (fk_EmpWorkLevel = EMPLLEVEL.pk_AdmID)
       LEFT JOIN Gen_AdministrationItems EMPCATEGORY ON (fk_EmpType = EMPCATEGORY.pk_AdmID)
)
SELECT * FROM InvolvedEmployees ORDER BY EmpFullName
  
```

Otro modo de ejecutar procedimientos almacenados es a través del *ConfiguratioManager*, 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>
/// Metodo que devuelve la lista de empleados con privilegios sobre los
indicadores
/// </summary>
public DataTable ListEmployeePrivileges(int indicatorID)
{
    DataTable table = SqlHelper.ExecuteScalar(
        System.Configuration.ConfigurationManager.AppSettings["conn"],
        "I_Indicator_ListEmployeesPrivileges", indicatorID);
    return table;
}
  
```

Donde el procedimiento I_Indicator_ListEmployeesPrivileges es el siguiente:

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[I_Indicator_ListEmployeesPrivileges]
(
    @indicatorID int
)
AS

DECLARE @individualIndicators TABLE(id int)

INSERT @individualIndicators
    SELECT pk_IndId FROM dbo.Obj_Ind_Indicators
    WHERE (pk_IndId=@indicatorID AND fk_IndParentItemId IS NULL)
    UNION
    SELECT pk_IndId FROM dbo.Obj_Ind_Indicators
    WHERE (fk_IndParentItemId=@indicatorID AND fk_IndParentItemId IS NOT NULL)

;WITH EmployeePrivileges (EmpID, EmpFullName, EmpCode, EmpCategory,
EmpIsInvolved, PrivName, Privilege) AS
(
    SELECT pk_EmpID EmpID,
           EmpSurname + ', ' + EmpName EmpFullName,
           EmpCode EmpCode,
           EMPCATEGORY.AdmName,
           1,
           pk_IndPrivName PrivName,
           cast(IndPrivValue as int) Privilege
    FROM dbo.Orh_Emp_Employees
    INNER JOIN [dbo].[Obj_Ind_IndicatorEmployeePrivileges] ON( (
        fk_IndPrivEmployeeID = pk_EmpID
        AND fk_IndPrivIndicatorID IN (SELECT id FROm @individualIndicators))
    INNER JOIN R_Indicator_Employee ON (fk_IndPrivEmployeeID IN (SELECT
        fk_IndEmpAssistant FROM R_Indicator_Employee WHERE fk_IndEmpIndicator IN (SELECT
            id FROM @individualIndicators)))
    LEFT JOIN Gen_AdministrationItems EMPCATEGORY ON (fk_EmpWorkLevel =
EMPCATEGORY.pk_AdmID)
    UNION
  
```

```

SELECT pk_EmpID EmpID,
       EmpSurname + ' , ' +EmpName EmpFullName,
       EmpCode EmpCode,
       EMPCATEGORY.AdmName,
       0,
       pk_IndPrivName PrivName,
       cast(IndPrivValue as int) Privilege
  FROM dbo.Orh_Emp_Employees
 INNER JOIN [dbo].[Obj_Ind_IndicatorEmployeePrivileges] ON
(fk_IndPrivEmployeeID = pk_EmpId AND fk_IndPrivIndicatorID IN (SELECT id FROM
@individualIndicators))
    INNER JOIN Obj_Ind_Indicators ON (pk_IndId = fk_IndPrivIndicatorID)
    INNER JOIN R_Indicator_Employee ON (fk_IndPrivEmployeeID NOT IN (SELECT
fk_IndEmpAssistant FROM R_Indicator_Employee
WHERE fk_IndEmpIndicator IN (SELECT id FROM @individualIndicators)))
    LEFT JOIN Gen_AdministrationItems EMPCATEGORY ON (fk_EmpWorkLevel =
EMPCATEGORY.pk_AdmID)
)
,EmployeePrivilegesPivot(EmpID, EmpFullName, EmpCode, EmpCategory, EmpIsInvolved,
ChangeTargetData, ChangeValueData) AS
(
    SELECT DISTINCT(EmpID), EmpFullName, EmpCode, EmpCategory, EmpIsInvolved,
IndPrivAllowChangeTargetData, IndPrivAllowChangeValueData
      FROM EmployeePrivileges
     PIVOT
     (
        MAX(Privilege)
        FOR PrivName IN (IndPrivAllowChangeTargetData,
IndPrivAllowChangeValueData)
    ) AS PVT
)
SELECT * FROM EmployeePrivilegesPivot ORDER BY EmpFullName

```

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.
- Autenticación integrada con Microsoft-Windows.
- Instrucciones de procesado de cadenas, fechas, funciones matemáticas, etc.

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 la interfaz finalmente diseñada, y se muestran capturas de pantalla del mismo.

7.2 Elementos de interfaz

7.2.1 Iconos e imágenes

La utilización de imágenes e iconos ha sido destinada siempre a facilitar al usuario la utilidad y funcionamiento de los diferentes elementos y partes del sistema, acompañándolos siempre preferiblemente de una etiqueta textual que especifica la funcionalidad o el destino del elemento representado por la imagen.

Se ha utilizado una línea estética de tamaño y aspecto similar para todos los iconos utilizados, con el objetivo de uniformizar el diseño, usando dos tamaños de iconos: 32 píxeles de alto para los iconos grandes, utilizados en el menú principal de la aplicación, y 16 para los elementos pequeños, utilizados para barras de herramientas y otros elementos gráficos como botones o títulos.

7.2.2 Tablas

Se han utilizado tablas siempre que ha sido necesario mostrar relaciones de datos.

Las tablas que aparecen en la aplicación son configurables, otorgando gran flexibilidad al usuario para visualizar en cada momento la información que realmente le interesa. Algunas de las posibilidades más destacadas que ofrecen son:

- Ordenación por campos: el usuario puede ordenar los datos de la tabla en función de un campo determinado con tan sólo hacer click sobre dicha columna.
- Selección de campos a visualizar: en algunas tablas se permite seleccionar qué campos se desea mostrar en las columnas.
- Agrupación por columnas: es posible agrupar los resultados de cualquier tabla por determinados campos (por ejemplo, agrupar los indicadores existentes en función del área al que pertenecen).
- Selección múltiple.

- Filtrado: también es posible filtrar los resultados por uno o varios campos. Las opciones disponibles incluyen: “es igual a”, “distinto a”, “menor que”, “mayor que”, “comienza con”, “contiene”, “termina en”, etc. Según la opción elegida, se visualizarán sólo las filas que cumplan la condición elegida. 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.
- 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, podrá ser abierto para ver su contenido.

7.2.3 Árboles

Se han utilizado vistas en forma de árbol siempre que ha sido necesario mostrar estructuras, tales como la estructura de áreas, los indicadores de la organización, o la estructura de empleados y puestos. De este modo, se facilita al usuario una identificación rápida de los diferentes elementos de la estructura y una navegación eficiente y clara a través de los mismos.

7.2.4 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.

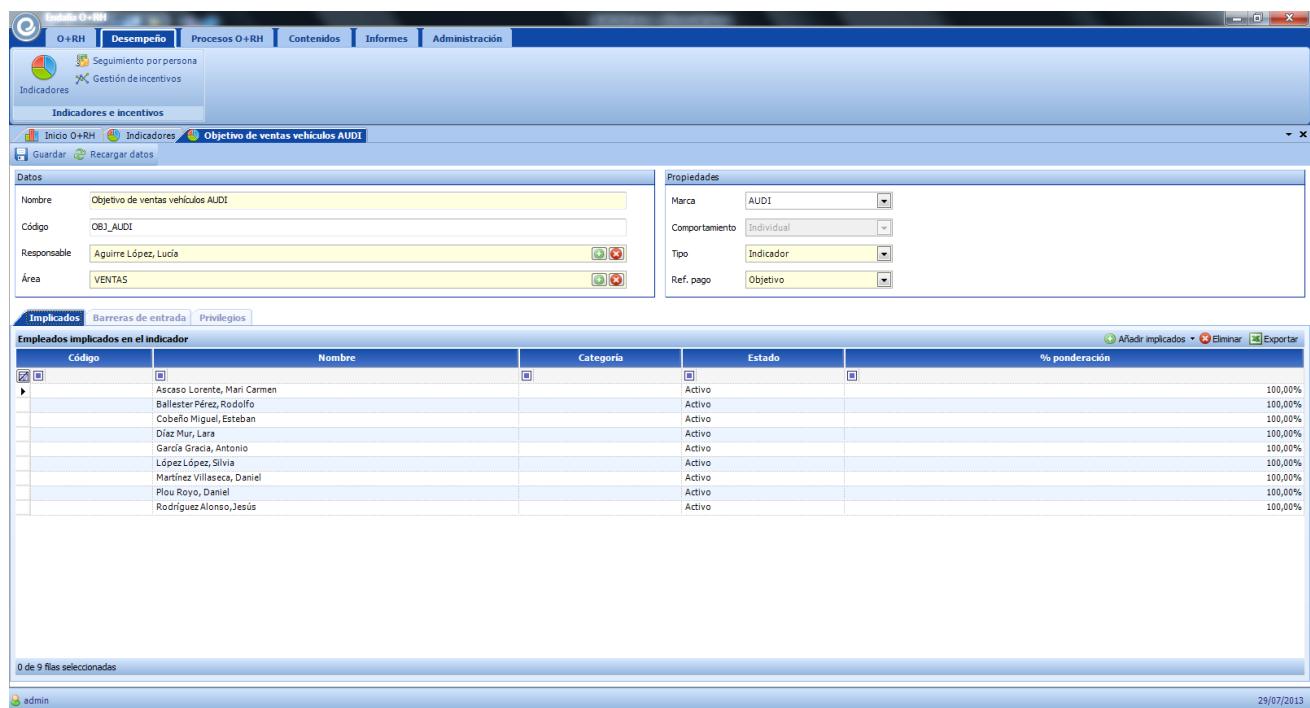
7.2.5 Editores de Fechas

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.

7.3 Pantallas del sistema

A continuación, se presenta el diseño definitivo de las principales interfaces del sistema:

7.3.1 Gestión de indicadores



Código	Nombre	Categoría	Estado	% ponderación
	Ascaso Lorente, Mari Carmen		Activo	100,00%
	Ballester Pérez, Rodolfo		Activo	100,00%
	Cobello Miguel, Esteban		Activo	100,00%
	Díaz Mur, Lara		Activo	100,00%
	García Gracia, Antonio		Activo	100,00%
	López López, Silva		Activo	100,00%
	Martínez Villaseca, Daniel		Activo	100,00%
	Plou Royo, Daniel		Activo	100,00%
	Rodríguez Alonso, Jesús		Activo	100,00%

Figura 2: Gestión de personas implicadas en el indicador

Indicadores e incentivos

Unidad Vendida Vehículos JAGUAR

Datos

Nombre	Unidad Vendida Vehículos JAGUAR
Código	UNID_JAGUAR
Responsable	Aguirre López, Lucía
Área	VENTAS

Propiedades

Marca	JAGUAR
Comportamiento	Individual
Tipo	Indicador
Ref. pago	Unidad

Barreras de entrada

Código

BARR_AUS	Ausencias mensuales
BARR_CLIENTINS	Clientes insatisfechos mensuales
BARR_PART	Partes negativas equipo
BARR_PARTNOK	Partes e incidencias entregadas a destiempo o no entregadas
OBJ_CLIENT	Clientes satisfechos mensuales
UNID_NOSIG	Clientes sin seguimiento
UNID_PART	Partes negativas individuales
UNID_PARTOK	Partes e incidencias entregadas a tiempo
UNID_RETRO	Retrasos mensuales
UNIT_REPINS	Reparaciones insatisfactorias para el cliente

Barrera de entrada

Implicados **Barreras de entrada** **Privilegios**

0 de 10 filas seleccionadas

admin 29/07/2013

Figura 3: Gestión de barreras de entrada que afectan al indicador

Indicadores e incentivos

Objetivo de ventas vehículos AUDI

Datos

Nombre	Objetivo de ventas vehículos AUDI
Código	OBJ_AUDI
Responsable	Aguirre López, Lucía
Área	VENTAS

Propiedades

Marca	AUDI
Comportamiento	Individual
Tipo	Indicador
Ref. pago	Objetivo

Empleados con privilegios sobre el indicador

Empleado

Código	Nombre	Categoría	Modificación objetivo	Modificación real	Observaciones
Ascaso Lorente, Mari Carmen			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Ballester Pérez, Rodolfo			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Cobello Miguel, Esteban			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Díaz Mur, Lara			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
García Gracia, Antonio			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
López López, Silvia			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Martínez Villaseca, Daniel			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Piou Rojo, Daniel			<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Rodríguez Alonso, Jesús			<input type="checkbox"/>	<input checked="" type="checkbox"/>	

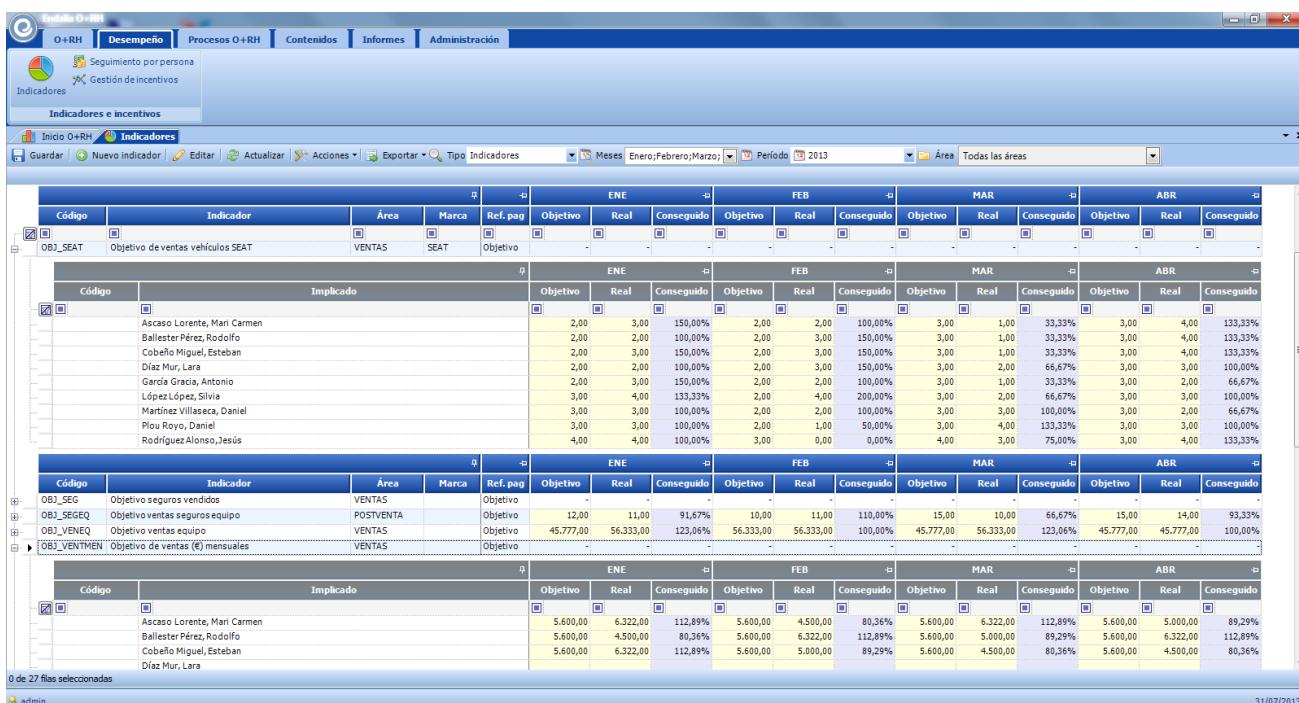
Privilegios

0 de 9 filas seleccionadas

admin 29/07/2013

Figura 4: Gestión de personas con privilegios sobre el indicador

7.3.2 Gestión de datos (objetivos y reales)



The screenshot shows the Endalia O+RH software interface. The main menu includes 'O+RH', 'Desempeño', 'Procesos O+RH', 'Contenidos', 'Informes', and 'Administración'. The current view is under 'Indicadores e incentivos'.

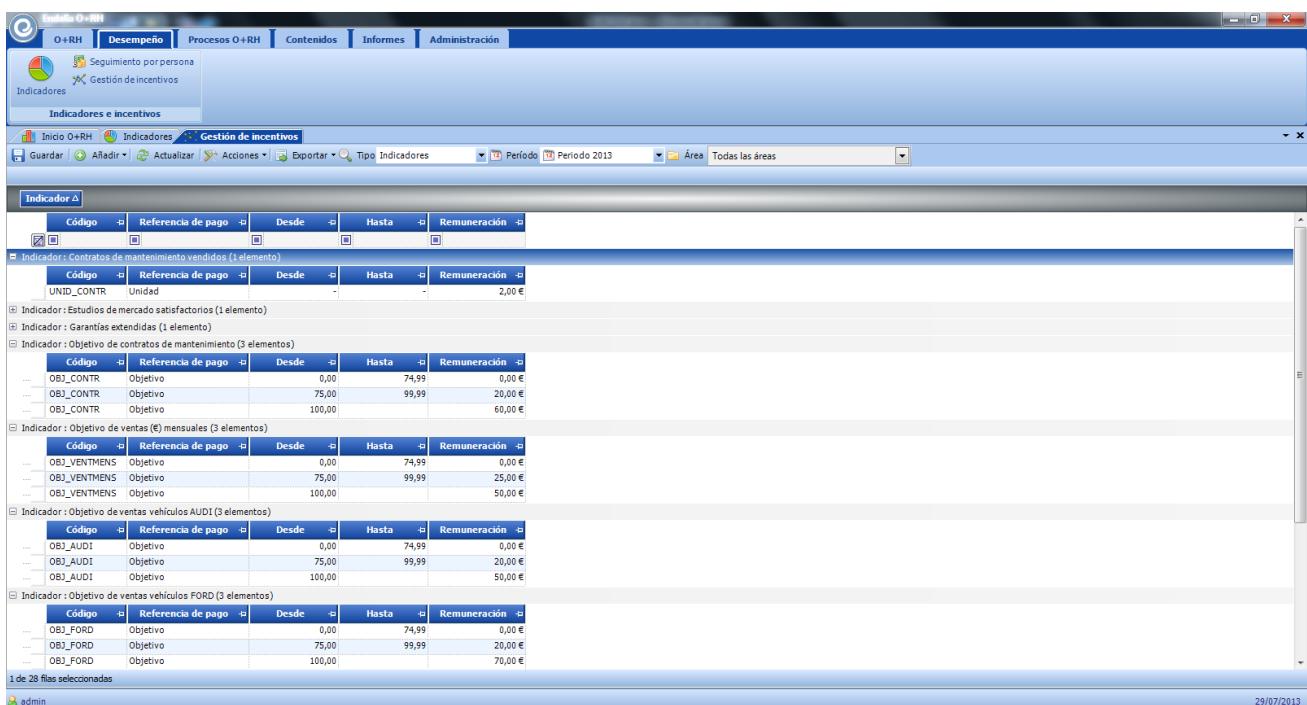
The main content area displays several tables of performance data:

- Table 1:** Shows data for 'Objetivo de ventas vehículos SEAT' across 'VENTAS' and 'SEAT' categories. It includes columns for 'Código', 'Indicador', 'Área', 'Marca', 'Ref. pag', 'Objetivo', 'Real', and 'Conseguido' for the months of ENE, FEB, MAR, and ABR.
- Table 2:** Shows data for 'Implicado' (employees) under 'Objetivo de ventas vehículos SEAT'. It includes columns for 'Código', 'Indicador', 'Área', 'Marca', 'Ref. pag', 'Objetivo', 'Real', and 'Conseguido' for the months of ENE, FEB, MAR, and ABR.
- Table 3:** Shows data for 'Objetivo seguros vendidos' across 'VENTAS' categories. It includes columns for 'Código', 'Indicador', 'Área', 'Marca', 'Ref. pag', 'Objetivo', 'Real', and 'Conseguido' for the months of ENE, FEB, MAR, and ABR.
- Table 4:** Shows data for 'Objetivo ventas seguros equipo' across 'POSTVENTA' categories. It includes columns for 'Código', 'Indicador', 'Área', 'Marca', 'Ref. pag', 'Objetivo', 'Real', and 'Conseguido' for the months of ENE, FEB, MAR, and ABR.
- Table 5:** Shows data for 'Objetivo ventas equipo' across 'VENTAS' categories. It includes columns for 'Código', 'Indicador', 'Área', 'Marca', 'Ref. pag', 'Objetivo', 'Real', and 'Conseguido' for the months of ENE, FEB, MAR, and ABR.
- Table 6:** Shows data for 'Objetivo de ventas (€) mensuales' across 'VENTAS' categories. It includes columns for 'Código', 'Indicador', 'Área', 'Marca', 'Ref. pag', 'Objetivo', 'Real', and 'Conseguido' for the months of ENE, FEB, MAR, and ABR.

At the bottom left, it says '0 de 27 filas seleccionadas'. At the bottom right, it says '31/07/2013'.

Figura 5: Gestión de datos objetivos y reales

7.3.3 Gestión de remuneraciones e incentivos



The screenshot shows the Endalia O+RH software interface with the 'Gestión de incentivos' (Incentive Management) module selected. The main window displays several tables of data, each with columns for Código (Code), Referencia de pago (Payment Reference), Desde (From), Hasta (To), and Remuneración (Remuneration). The data is categorized by indicator type:

- Indicador: Contratos de mantenimiento vendidos (1 elemento)**

Código	Referencia de pago	Desde	Hasta	Remuneración
UNID_CONTR	Unidad	-	-	2,00 €
- Indicador : Estudios de mercado satisfactorios (1 elemento)**
- Indicador : Garantías extendidas (1 elemento)**
- Indicador : Objetivo de contratos de mantenimiento (3 elementos)**

Código	Referencia de pago	Desde	Hasta	Remuneración
OBJ_CONTR	Objetivo	0,00	74,99	0,00 €
OBJ_CONTR	Objetivo	75,00	99,99	20,00 €
OBJ_CONTR	Objetivo	100,00	-	60,00 €
- Indicador : Objetivo de ventas (€) mensuales (3 elementos)**

Código	Referencia de pago	Desde	Hasta	Remuneración
OBJ_VENTMENS	Objetivo	0,00	74,99	0,00 €
OBJ_VENTMENS	Objetivo	75,00	99,99	25,00 €
OBJ_VENTMENS	Objetivo	100,00	-	50,00 €
- Indicador : Objetivo de ventas vehículos AUDI (3 elementos)**

Código	Referencia de pago	Desde	Hasta	Remuneración
OBJ_AUDI	Objetivo	0,00	74,99	0,00 €
OBJ_AUDI	Objetivo	75,00	99,99	20,00 €
OBJ_AUDI	Objetivo	100,00	-	50,00 €
- Indicador : Objetivo de ventas vehículos FORD (3 elementos)**

Código	Referencia de pago	Desde	Hasta	Remuneración
OBJ_FORD	Objetivo	0,00	74,99	0,00 €
OBJ_FORD	Objetivo	75,00	99,99	20,00 €
OBJ_FORD	Objetivo	100,00	-	70,00 €

At the bottom left, it says "1 de 28 filas seleccionadas" (1 of 28 rows selected). At the bottom right, it says "29/07/2013".

Figura 6: Gestión de remuneraciones asociadas a los indicadores

Seguimiento por persona															
Indicadores e incentivos		Gestión de incentivos													
Indicadores		Gestión de incentivos													
Código	Empleado	Nivel	Centro de trabajo												
	Ascaso Lorente, Mari Carmen		COCHES BUFALO SL												
				ENE	Real	Conseguido - €/ud.	Variación barreras	Total (€)	Objetivo	Real	Conseguido - €/ud.	Variación barreras	Total (€)	Objetivo	
OBJ_AUDI	Objetivo de ventas vehículos AUDI	100,00%	VENTAS	AUDI	Objetivo										
OBJ_CONTR	Objetivo de contratos de mantenimiento	100,00%	POSTVENTA		Objetivo										
OBJ_FORD	Objetivo de ventas vehículos FORD	100,00%	VENTAS	FORD	Objetivo										
OBJ_GAR	Objetivo garantías extendidas	100,00%	POSTVENTA		Objetivo										
OBJ_JAG	Objetivo de ventas vehículos JAGUAR	100,00%	VENTAS	JAGUAR	Objetivo										
OBJ_SEAT	Objetivo de ventas vehículos SEAT	100,00%	VENTAS	SEAT	Objetivo	2,00	3,00	150,00%		2,00	2,00	100,00%		3,00	
OBJ_SEG	Objetivo seguros vendidos	100,00%	VENTAS		Objetivo										
OBJ_SEGEQ	Objetivo ventas seguros equipo	100,00%	POSTVENTA		Objetivo	12,00	11,00	91,67%	0,00 €	20,00 €	10,00	11,00	110,00%	0,00 €	
OBJ_VENEQ	Objetivo ventas equipo	100,00%	VENTAS		Objetivo	45777,00	56333,00	123,05%			56333,00	56333,00	100,00%		45777,00
OBJ_VENTMEN	Objetivo de ventas (€) mensuales	100,00%	VENTAS		Objetivo	5600,00	6322,00	112,89%	0,00 €	50,00 €	5600,00	4500,00	80,36%	0,00 €	25,00 €
UNID_2M	Unidad vendida vehículos segunda mano	100,00%	VENTAS		Unidad	-									
UNID_AUDI	Unidad Vendida Vehículos AUDI	100,00%	VENTAS	AUDI	Unidad	-									
UNID_BUJ	Unidad vendida bujías	100,00%	VENTAS		Unidad	-	15,00	5,00 €/ud.	0,00 €	75,00 €	-	10,00	5,00 €/ud.	0,00 €	50,00 €
UNID CONTR	Contratos de mantenimiento vendidos	100,00%	POSTVENTA		Unidad	-									
UNID_ESTMET	Estudios de mercado satisfactorios	100,00%	POSTVENTA		Unidad	-									
UNID_FORD	Unidad Vendida Vehículos FORD	100,00%	VENTAS	FORD	Unidad	-									
UNID_GAR	Garantías extendidas	100,00%	POSTVENTA		Unidad	-									
UNID_JAGUA	Unidad Vendida Vehículos JAGUAR	75,00%	VENTAS	JAGUAR	Unidad	-									
UNID_LUN	Reparaciones de lunas	100,00%	POSTVENTA		Unidad	-									
UNID_NEUM	Unidad vendida neumáticos	100,00%	VENTAS		Unidad	-	12,00	2,00 €/ud.	0,00 €	24,00 €	-	20,00	2,00 €/ud.	0,00 €	40,00 €
UNID REP	Reparaciones de lunas equipo	100,00%	POSTVENTA		Unidad	-									
UNID_RPGEN	Reparaciones generales equipo	100,00%	POSTVENTA		Unidad	-									
UNID_REPMOT	Reparaciones de motor	100,00%	POSTVENTA		Unidad	-									
UNID_REPMOTOR	Reparaciones de motor equipo	100,00%	POSTVENTA		Unidad	-									
UNID_SEAT	Unidad Vendida Vehículos SEAT	100,00%	VENTAS	SEAT	Unidad	-	1,00	120,00 €/ud.	0,00 €	120,00 €	-	2,00	120,00 €/ud.	0,00 €	240,00 €
UNID_SEG	Seguros auto vendidos	100,00%	POSTVENTA		Unidad	-									
BARR_AUS	Ausencias mensuales	-	POSTVENTA	-	-	2,00	-	-	-	-	1,00	-	-	-	
BARR_CLIENT	Clientes insatisfechos mensuales	-	POSTVENTA	-	-	0,00	-	-	-	-	1,00	-	-	-	
BARR_PART	Partes negativas equipo	-	POSTVENTA	-	-	3,00	-	-40,00%	-	-	3,00	-	-40,00%	-	
BARR_PARTNO	Partes e incidencias entregadas a destiempo o no entregadas	-	POSTVENTA	-	-	2,00	-	-	-	-	1,00	-	-	-	
OBL_CLIENT	Clientes satisfechos mensuales	-	POSTVENTA	-	-	6,00	-	10,00%	-	-	6,00	-	10,00%	-	
UNID_NOSEG	Clientes sin seguimiento	-	POSTVENTA	-	-	0,00	-	-	-	-	0,00	-	-	-	
UNID_PART	Partes negativas individuales	-	POSTVENTA	-	-	1,00	-	-	-	-	1,00	-	-	-	
UNID_PARTOK	Partes e incidencias entregadas a tiempo	-	POSTVENTA	-	-	4,00	-	-	-	-	4,00	-	-	-	
UNID_RET	Retrasos mensuales	-	POSTVENTA	-	-	1,00	-	-50,00%	-	-	0,00	-	-	-	
UNIT_REPINS	Reparaciones insatisfactorias para el cliente	-	POSTVENTA	-	-	1,00	-	-	-	-	2,00	-	-	-	
REMUNERACIÓN MENSUAL															
					PAGADO	289,00 €				PAGADO	405,00 €				
	Ballester Pérez, Rodolfo		COCHES BUFALO SL												
	Coboño Miguel, Esteban		COCHES BUFALO SL												
	Díaz Mur, Lara		COCHES BUFALO SL												

Figura 7: Visualización detallada de los incentivos recibidos por las personas

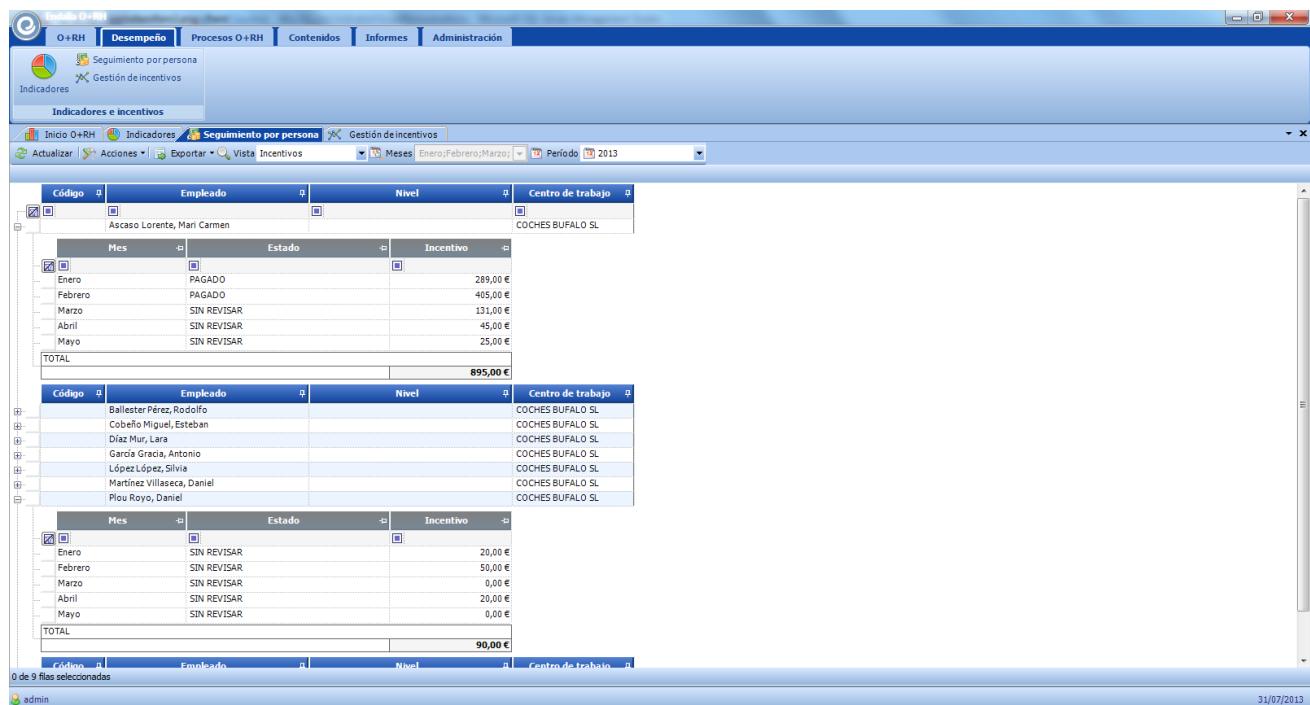


Figura 8: Visualización general de los incentivos mensuales recibidos por las personas

8. Bibliografía

8.1 Referencias

- [IGJ, 2000] I. Jacobson, G. Booch, J. Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. Pearson Education
- [HHBK 2008] P. Henri Kuaté, 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 de Gestión de Objetivos y Retribución de Incentivos

Plan de pruebas

Versión 1.3

Publicado el 15/07/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
17/06/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
10/07/2013	1.1	Modificación del documento	Santiago Pelegrín Costea
12/07/2013	1.2	Revisión del documento	Fernando Cortés Franco
15/07/2013	1.3	Corrección del documento	Santiago Pelegrín Costea

Índice

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.....	6
1.6 Resumen.....	6
2. Descripción del proceso.....	7
3. Pruebas unitarias y de integración.....	8
3.1 Pruebas de gestión de indicadores.....	8
3.2 Pruebas de gestión de datos	15
3.3 Pruebas de gestión de remuneraciones e incentivos	17
3.4 Pruebas de gestión de incidencias	20
4. Pruebas de sistema y aceptación.....	21
4.1 Pruebas de operación	21
4.2 Pruebas de facilidad de uso.....	21
5. Bibliografía	22
5.1 Referencias.....	22
5.2 Referencias web.....	22

1. Introducción

1.1 Propósito del documento

El presente documento describe la fase de pruebas del SGORI. 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 SGORI.

1.3 Acrónimos

- PGI: Prueba de Gestión de Indicadores.
- PGD: Prueba de Gestión de Datos.
- PGRI: Prueba de Gestión de Remuneraciones e Incentivos.
- PGINC: Prueba de Gestión de Incidencias.
- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.

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.

1.5 Referencias

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

- ESPREQ: documento en el que se especifican los requisitos del sistema.
- ANALISIS: documento de análisis del sistema.
- DISEÑO: documento de diseño del sistema.
- IMPLEM: documento de implementación del sistema.

1.6 Resumen

Este documento describe el proceso de pruebas del SGORI. 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.

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 SGORI, 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 remuneraciones e incentivos si no se desarrollan los subsistemas de gestión de indicadores y de gestión de datos). 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.

3. Pruebas unitarias y de integración

3.1 Pruebas de gestión de indicadores

Identificador	PGI – 01
Descripción	Carga inicial de la estructura de indicadores.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Seleccionar dentro del desplegable ‘Tipo’ de la barra de herramientas la opción ‘Indicadores’.
Casos especiales a comprobar	
Criterio de aceptación	En la tabla, aparecen listados todos los indicadores activos del sistema, junto con su información asociada, ordenados por código.
Resultado de la prueba	Aceptada.

Identificador	PGI - 02
Descripción	Carga inicial de la estructura de barreras de entrada.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Seleccionar dentro del desplegable ‘Tipo’ de la barra de herramientas la opción ‘Barreras de entrada’.
Casos especiales a comprobar	
Criterio de aceptación	En la tabla, aparecen listados los datos de todos los indicadores de tipo barrera de entrada activos del sistema, ordenados por código.
Resultado de la prueba	Aceptada.

Identificador	PGI - 03
Descripción	Añadir indicador/barrera de entrada
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Pulsar botón ‘Añadir’. 3. Rellenar los datos del nuevo indicador. 4. En el campo ‘Tipo’, introducir si se trata de indicador o barrera de entrada. 5. Pulsar el botón ‘Guardar’.
Casos especiales a comprobar	<ul style="list-style-type: none"> • No se introducen todos los campos obligatorios en el formulario de alta (paso 3). Resultado esperado: el sistema advierte al usuario de que debe llenar todos los campos obligatorios.
Criterio de aceptación	El nuevo indicador/barrera de entrada es almacenado en el sistema y visible en el listado de datos.
Resultado de la prueba	Aceptada.

Identificador	PGI - 04
Descripción	Editar datos del indicador/barrera de entrada.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Seleccionar un indicador/barrera de entrada del listado. 3. Pulsar el botón ‘Editar’. 4. En la ficha abierta, modificar los datos del indicador/barrera de entrada. 5. Pulsar el botón ‘Guardar’.
Casos especiales a comprobar	<ul style="list-style-type: none"> • No se completan todos los campos obligatorios en el formulario de edición del indicador (paso 3). Resultado esperado: el sistema advierte al usuario de que debe llenar todos los campos obligatorios. • El usuario introduce datos incoherentes o en formato erróneo. Resultado esperado: el sistema controla la entrada de datos e informa al usuario de aquellos campos que se han introducido de manera incorrecta
Criterio de aceptación	El sistema almacena correctamente los nuevos datos del indicador.
Resultado de la prueba	Aceptada.

Identificador	PGI - 05
Descripción	Cambiar de tipo un indicador a barrera de entrada.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Seleccionar dentro del desplegable ‘Tipo’ de la barra de herramientas la opción ‘Indicadores’. 3. Seleccionar un indicador del listado. 4. Pulsar el botón ‘Editar’. 5. En la ficha del indicador abierta, modificar el tipo a ‘Barrera de entrada’. 6. Pulsar el botón ‘Guardar’.
Casos especiales a comprobar	
Criterio de aceptación	El sistema almacena correctamente el cambio de tipo en el indicador.
Resultado de la prueba	Aceptada.

Identificador	PGI - 06
Descripción	Cambiar de tipo una barrera de entrada a indicador
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Seleccionar dentro del desplegable ‘Tipo’ de la barra de herramientas la opción ‘Barreras de entrada’. 3. Seleccionar una barrera de entrada del listado. 4. Pulsar el botón ‘Editar’. 5. En la ficha del indicador abierta, modificar el tipo a ‘Indicador’. 6. Pulsar el botón ‘Guardar’.
Casos especiales a comprobar	
Criterio de aceptación	El sistema almacena correctamente el cambio de tipo en la barrera de entrada.
Resultado de la prueba	Aceptada.

Identificador	PGI - 07
Descripción	Eliminar indicador/barrera de entrada.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Seleccionar un indicador/barrera de entrada del listado. 3. Pulsar el botón ‘Eliminar’, dentro del menú emergente ‘Acciones’ de la barra de herramientas. 4. Pulsar el botón ‘Sí’ en el diálogo de confirmación de eliminación.
Casos especiales a comprobar	
Criterio de aceptación	El indicador/barrera de entrada es eliminado definitivamente del sistema, desapareciendo del listado.
Resultado de la prueba	Aceptada.
Identificador	PGI - 08
Descripción	Añadir persona implicada en el indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Implicados’ dentro de la ficha del indicador. 4. En la tabla, pulsar el botón ‘Añadir implicados’. 5. Dentro de la ventana emergente abierta, seleccionar una o varias personas. 6. Pulsar el botón ‘Aceptar’ de la venta emergente. 7. Pulsar el botón ‘Guardar’ de la ficha del indicador.
Casos especiales a comprobar	<ul style="list-style-type: none"> • Se intenta añadir una persona que ya aparece como implicada en el listado del indicador. Resultado esperado: el sistema no añade de nuevo a la persona en el listado de implicados del indicador.
Criterio de aceptación	El listado de implicados del indicador es almacenado correctamente en el sistema, y mostrado dentro de la ficha del indicador, pestaña ‘Implicados’.
Resultado de la prueba	Aceptada.

Identificador	PGI - 09
Descripción	Editar el peso porcentual del indicador sobre las personas.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Implicados’ dentro de la ficha del indicador. 4. Seleccionar un implicado de la lista. 5. Pulsar el botón ‘Eliminar’ de la tabla de implicados. 6. Pulsar el botón ‘Sí’ del diálogo de confirmación de eliminación.
Casos especiales a comprobar	
Criterio de aceptación	El implicado seleccionado es eliminado definitivamente del listado de implicados en el indicador, y desaparece de la tabla dentro de la pestaña ‘Implicados’.
Resultado de la prueba	Aceptada.

Identificador	PGI - 10
Descripción	Eliminar persona implicada en el indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Implicados’ dentro de la ficha del indicador. 4. En la tabla, modificar el valor de una de las celdas de la columna ‘% ponderación’. 5. Pulsar el botón ‘Guardar’ de la ficha del indicador.
Casos especiales a comprobar	
Criterio de aceptación	El nuevo peso del indicador sobre el implicado modificado es almacenado correctamente en el sistema, y mostrado dentro de la ficha del indicador, pestaña ‘Implicados’.
Resultado de la prueba	Aceptada.

Identificador	PGI - 11
Descripción	Añadir persona con privilegios de edición de datos del indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Privilegios’ dentro de la ficha del indicador. 4. En la tabla, pulsar el botón ‘Añadir’. 5. Dentro de la ventana emergente abierta, seleccionar una o varias personas. 6. Pulsar el botón ‘Aceptar’ de la venta emergente. 7. Pulsar el botón ‘Guardar’ de la ficha del indicador.
Casos especiales a comprobar	<ul style="list-style-type: none"> • Se intenta añadir una persona que ya aparece en el listado de personas con permisos sobre el indicador. Resultado esperado: el sistema no añade a la persona que ya aparece en el listado.
Criterio de aceptación	El listado de personas que poseen privilegios sobre el indicador es almacenado correctamente en el sistema.
Resultado de la prueba	Aceptada.

Identificador	PGI - 12
Descripción	Modificar los privilegios de una persona sobre el indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Privilegios’ dentro de la ficha del indicador. 4. En la tabla, modificar los privilegios, seleccionando o deseleccionando las opciones correspondientes en las celdas del grupo de columnas ‘Privilegios’. 5. Pulsar el botón ‘Guardar’ de la ficha del indicador.
Casos especiales a comprobar	
Criterio de aceptación	Los nuevos privilegios de la persona que han sido modificados son almacenados correctamente en el sistema.
Resultado de la prueba	Aceptada.

Identificador	PGI - 13
Descripción	Eliminar persona con privilegios sobre el indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Privilegios’ dentro de la ficha del indicador. 4. Seleccionar una persona de la lista. 5. Pulsar el botón ‘Eliminar’ de la tabla. 6. Pulsar el botón ‘Sí’ del diálogo de confirmación de eliminación.
Casos especiales a comprobar	
Criterio de aceptación	La persona seleccionada es eliminada definitivamente del listado de personas con privilegios sobre el indicador, y desaparece de la tabla dentro de la pestaña ‘Privilegios’.
Resultado de la prueba	Aceptada.

Identificador	PGI - 14
Descripción	Añadir barrera de entrada que afecte al indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Barreras de entrada’ dentro de la ficha del indicador. 4. Dentro de la tabla, pulsar el botón ‘Añadir’. 5. Dentro de la ventana emergente abierta, seleccionar una o varios indicadores de tipo barrera de entrada. 6. Pulsar el botón ‘Aceptar’ de la venta emergente. 7. Pulsar el botón ‘Guardar’ de la ficha del indicador.
Casos especiales a comprobar	<ul style="list-style-type: none"> • Se intenta añadir una barrera de entrada que ya aparece en el listado de barreras de entrada que afectan al indicador. Resultado esperado: el sistema no añade la barrera de entrada que ya aparece en el listado.
Criterio de aceptación	El listado de barreras de entrada que afectan al indicador es almacenado correctamente en el sistema.
Resultado de la prueba	Aceptada.

Identificador	PGI - 15
Descripción	Eliminar barrera de entrada que afecta al indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Hacer doble click en un indicador del listado para abrir su ficha. 3. Seleccionar la pestaña ‘Barreras de entrada’ dentro de la ficha del indicador. 4. Seleccionar una barrera de entrada de la lista. 5. Pulsar el botón ‘Eliminar’ de la tabla. 6. Pulsar el botón ‘Sí’ del diálogo de confirmación de eliminación.
Casos especiales a comprobar	
Criterio de aceptación	La barrera de entrada seleccionada es eliminada definitivamente del listado de barreras de entrada que afectan al indicador, y desaparece de la tabla.
Resultado de la prueba	Aceptada.

3.2 Pruebas de gestión de datos

Identificador	PGD – 01
Descripción	Carga inicial de la estructura de datos objetivos y reales de indicadores.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal.
Casos especiales a comprobar	
Criterio de aceptación	En la tabla, aparecen listados los datos objetivos y reales de todos los indicadores activos del sistema, agrupados por meses.
Resultado de la prueba	Aceptada.

Identificador	PGD - 02
Descripción	Editar datos objetivos del indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Modificar el valor en la celda de la columna ‘Objetivo’. 3. Pulsar el botón ‘Guardar’ de la barra de herramientas.
Casos especiales a comprobar	<ul style="list-style-type: none"> • El usuario introduce datos incoherentes o incorrectos en la celda ‘Objetivo’. Resultado esperado: el sistema no permite la introducción de datos que no sean números reales.
Criterio de aceptación	El dato objetivo modificado es almacenado correctamente en el sistema y visible en la tabla de datos.
Resultado de la prueba	Aceptada.

Identificador	PGD - 03
Descripción	Editar datos reales del indicador.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Indicadores’ del menú principal. 2. Modificar el valor en la celda de la columna ‘Real’. 3. Pulsar el botón ‘Guardar’ de la barra de herramientas.
Casos especiales a comprobar	<ul style="list-style-type: none"> • El usuario introduce datos incoherentes o incorrectos en la celda ‘Real’. Resultado esperado: el sistema no permite la introducción de datos que no sean números reales.
Criterio de aceptación	El dato real modificado es almacenado correctamente en el sistema y visible en la tabla de datos.
Resultado de la prueba	Aceptada.

3.3 Pruebas de gestión de remuneraciones e incentivos

Identificador	PGRI – 01
Descripción	Carga inicial de la estructura de remuneraciones.
Método	1. Seleccionar la opción ‘Gestión de incentivos’ del menú principal.
Casos especiales a comprobar	
Criterio de aceptación	En la tabla, aparecen listados todas las remuneraciones del sistema, agrupados por indicador y ordenados por valores mínimo y máximo de consecución (escalado).
Resultado de la prueba	Aceptada.
Identificador	PGRI – 02
Descripción	Añadir remuneración.
Método	1. Seleccionar la opción ‘Gestión de incentivos’ del menú principal. 2. Pulsar el botón ‘Añadir’ de la barra de herramientas. 3. En la nueva fila añadida, llenar los datos correspondientes a la nueva remuneración. 4. Pulsar el botón ‘Guardar’ de la barra de herramientas’.
Casos especiales a comprobar	<ul style="list-style-type: none"> • No se introducen todos los campos obligatorios en la fila de la nueva remuneración (paso 3). Resultado esperado: el sistema advierte al usuario de que debe llenar todos los campos obligatorios. • El usuario introduce datos incoherentes o incorrectos (como por ejemplo, escalados superpuestos) al dar de alta la nueva remuneración. Resultado esperado: el sistema advierte al usuario de que se han introducido datos incorrectos.
Criterio de aceptación	La nueva remuneración es almacenada correctamente en el sistema y visible en el listado de remuneraciones.
Resultado de la prueba	Aceptada.

Identificador	PGRI - 03
Descripción	Editar datos de la remuneración.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Gestión de incentivos’ del menú principal. 2. En la tabla, modificar los valores en las celdas que muestran los datos de la remuneración. 3. Pulsar el botón ‘Guardar’ de la barra de herramientas’
Casos especiales a comprobar	<ul style="list-style-type: none"> • No se completan todos los campos obligatorios en la fila de la remuneración (paso 2). Resultado esperado: el sistema advierte al usuario de que debe llenar todos los campos obligatorios. • El usuario introduce datos incoherentes o incorrectos (como por ejemplo, escalados superpuestos) al dar de alta la nueva remuneración. Resultado esperado: el sistema advierte al usuario de que se han introducido datos incorrectos.
Criterio de aceptación	El sistema almacena correctamente los nuevos datos de la remuneración.
Resultado de la prueba	Aceptada.

Identificador	PGRI - 04
Descripción	Eliminar remuneración.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Gestión de incentivos’ del menú principal. 2. Seleccionar una remuneración del listado. 3. Pulsar el botón ‘Eliminar’, dentro del menú emergente ‘Acciones’ de la barra de herramientas. 4. Pulsar el botón ‘Sí’ en el diálogo de confirmación de eliminación de la remuneración.
Casos especiales a comprobar	
Criterio de aceptación	La remuneración es eliminada definitivamente del sistema, desapareciendo del listado de remuneraciones.
Resultado de la prueba	Aceptada.

Identificador	PGRI - 05
Descripción	Carga de los incentivos recibidos detallados.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Seguimiento por persona’ del menú principal. 2. Seleccionar dentro del desplegable ‘Vista’ de la barra de herramientas la opción ‘Datos de indicadores’.
Casos especiales a comprobar	
Criterio de aceptación	En la tabla aparecen listadas las personas junto con todos los indicadores en las que están implicadas, los datos objetivos y reales conseguidos, las penalizaciones por barreras de entrada y los incentivos totales a recibir.
Resultado de la prueba	Aceptada.
Identificador	PGRI - 06
Descripción	Carga de los incentivos recibidos mensuales totales.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Seguimiento por persona’ del menú principal. 2. Seleccionar dentro del desplegable ‘Vista’ de la barra de herramientas la opción ‘Incentivos’.
Casos especiales a comprobar	
Criterio de aceptación	En la tabla aparecen listadas las personas junto con los incentivos mensuales totales a recibir.
Resultado de la prueba	Aceptada.

Identificador	PGRI - 07
Descripción	Validar incentivo mensual.
Método	<ol style="list-style-type: none"> 1. Seleccionar la opción ‘Seguimiento por persona’ del menú principal. 2. Seleccionar dentro del desplegable ‘Vista’ de la barra de herramientas la opción ‘Datos de indicadores’.
Casos especiales a comprobar	<ul style="list-style-type: none"> • Intentar validar los incentivos mensuales de un empleado cuyos datos de indicadores y barreras de entrada no se encuentran completos. Resultado esperado: el sistema advierte al usuario de que el incentivo mensual no puede ser validado hasta completar todos los datos.
Criterio de aceptación	El incentivo mensual es almacenado en el sistema, y mostrado como ‘Pagado’ en la tabla de incentivos.
Resultado de la prueba	Aceptada.

3.4 Pruebas de gestión de incidencias

Identificador	PGINC - 01
Descripción	Comprobar escritura de log.
Método	<ol style="list-style-type: none"> 1. Abrir el directorio donde se almacena el registro de log (indicador en el fichero de configuración). 2. Abrir el documento más reciente almacenado.
Casos especiales a comprobar	
Criterio de aceptación	El directorio contiene ficheros de log, y el fichero de log abierto contiene información asociada a las últimas operaciones realizadas en la aplicación.
Resultado de la prueba	Aceptada.

4. Pruebas de sistema y 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 objetivos y de retribución de incentivos que se establece con la utilización del SGORI 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.

5. Bibliografía

5.1 Referencias

[DUS 2002] Elfriede Dustin 2002. *Effective Software Testing*. Ed. Addison Wesley.

5.2 Referencias web

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

Sistema de Gestión de Objetivos y Retribución de Incentivos

Manual de usuario

Versión 1.2

Publicado el 29/07/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
25/07/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
26/07/2013	1.1	Revisión del documento	Fernando Cortés Franco
29/07/2013	1.2	Corrección del documento	Santiago Pelegrín Costea

Índice

1. Introducción	6
1.1 Propósito del documento	6
1.2 Alcance del documento.....	6
1.3 Acrónimos.....	6
1.4 Definiciones	6
1.5 Referencias.....	6
1.6 Resumen.....	7
2. Controles genéricos de la aplicación	8
2.1 Barra de herramientas	8
2.2 Uso de tablas.....	8
3. Indicadores.....	13
3.1 Indicadores / Barreras de entrada (pantalla principal)	13
3.1.1 Modos de visualización de datos.....	13
3.1.2 Gestión de datos objetivos y reales	14
3.1.3 Eliminar indicador/barrera de entrada.....	15
3.1.4 Cambiar comportamiento del indicador/barrera de entrada	16
3.1.5 Editar indicador/barrera de entrada	16
3.1.6 Añadir indicador/barrera de entrada	16
3.2 Ficha del indicador/barrera de entrada.....	16
3.2.1 Implicados	17
3.2.2 Barreras de entrada.....	18
3.2.3 Privilegios.....	19
4. Gestión de incentivos.....	20
4.1 Pantalla de gestión de incentivos/penalizaciones	20
4.1.1 Visualización de incentivos.....	20
4.1.2 Añadir incentivo/penalización.....	21
4.1.3 Editar incentivos	22
4.1.4 Eliminar indicador/barrera de entrada.....	22
5. Seguimiento por persona	23
5.1 Pantalla de gestión de remuneraciones recibidas por las personas	23
5.1.1 Modos de visualización de remuneraciones.....	23
5.1.2 Validar remuneración total mensual	25
6. Bibliografía	27

6.1	Referencias.....	27
6.2	Referencias web.....	27

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 SGORI, 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

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.
- O+RH: Organización y Recursos Humanos
- RRHH: Recursos Humanos

1.4 Definiciones

- Indicador: herramienta básica del SGORI que permite medir la consecución o evolución de los objetivos de una organización basados principalmente en los aspectos clave del negocio. Pueden influir y ser aplicados sobre un grupo o colectivo, o sobre individuos en concreto.
- Barrera de entrada: indicador que establece el límite a partir del cual se tiene derecho al devengo del incentivo por la superación de los objetivos asignados a los indicadores.

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 SGORI. Está compuesto de 8 apartados:

- Apartado 1: Introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: Se describen los controles genéricos de la aplicación (barra de herramientas y tablas), ilustrando las posibilidades que ofrecen.
- Apartado 3: Este apartado se centra en la descripción del módulo de gestión de indicadores y sus distintas funcionalidades, así como el módulo de gestión de datos.
- Apartado 4: Se describe el módulo de gestión de incentivos.
- Apartado 5: Se describe el módulo de gestión de remuneraciones.
- Apartado 6: Se muestra la bibliografía utilizada para la elaboración del presente documento.

2. Controles genéricos de la aplicación

2.1 Barra de herramientas

Desde la barra de herramientas (Figura 1), podemos acceder a los diversos módulos de la aplicación. El SGORI se encuentra dentro del módulo ‘Desempeño’, accediendo a través de la pestaña con el mismo nombre.

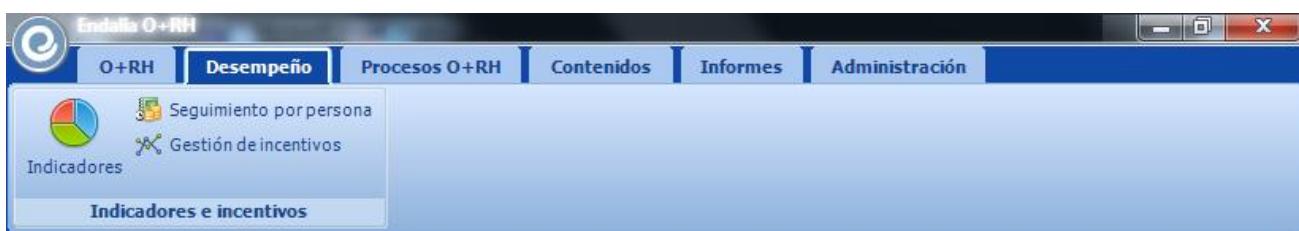


Figura 1: Barra de herramientas

Dentro del módulo que contiene el SGORI, pueden visualizarse los accesos a las siguientes partes del sistema:

- Indicadores: gestión de indicadores y de datos objetivos y reales de los mismos.
- Seguimiento por persona: gestión de las remuneraciones a recibir por las personas.
- Gestión de incentivos: gestión de las remuneraciones asociadas a los indicadores.

IMPORTANTE: Si desea agrandar el espacio de trabajo de la aplicación, puede ocultar esta barra de herramientas, haciendo doble clic sobre la pestaña seleccionada. Para volver a mostrar la barra de herramientas, basta con hacer doble clic de nuevo en cualquiera de las pestañas.

2.2 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, agrupar por resultados, etc.

- Ordenación por campos (Figura 2): 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).



Figura 2: Ordenación por campos

- Fijado de columnas y filas (Figura 3): 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.



Código	Indicador	Área	Marca	Ref. pag	Objetivo	Real	Conseguido
--------	-----------	------	-------	----------	----------	------	------------

Figura 3: Fijado de columnas

En este ejemplo, hemos bloqueado el grupo que contiene las columnas ‘Código’, ‘Indicador’, ‘Área’ y ‘Marca’, de manera que aunque movamos los resultados con la barra de desplazamiento inferior, el grupo de columnas correspondiente no se moverá, de forma que es más fácil comparar con el resto de columnas. Este funcionamiento es exactamente igual para tratar con filas, con la única diferencia de que en este caso, las filas quedarían fijas en la parte superior de la tabla.

- Agrupación (Figuras 4 y 5): 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.

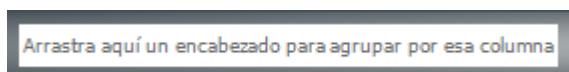
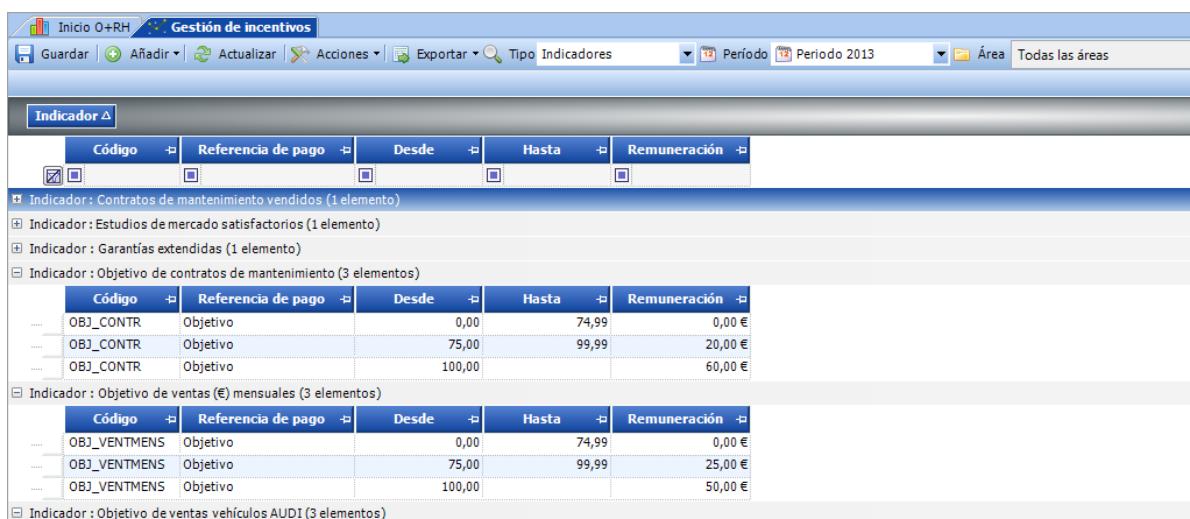


Figura 4: Zona de agrupación



Indicador					
Código	Referencia de pago	Desde	Hasta	Remuneración	
...
...
...
Indicador : Contratos de mantenimiento vendidos (1 elemento)					
...
Indicador : Estudios de mercado satisfactorios (1 elemento)					
...
Indicador : Garantías extendidas (1 elemento)					
...
Indicador : Objetivo de contratos de mantenimiento (3 elementos)					
Código	Referencia de pago	Desde	Hasta	Remuneración	
....	OBJ_CONTR	Objetivo	0,00	74,99	0,00 €
....	OBJ_CONTR	Objetivo	75,00	99,99	20,00 €
....	OBJ_CONTR	Objetivo	100,00		60,00 €
Indicador : Objetivo de ventas (€) mensuales (3 elementos)					
Código	Referencia de pago	Desde	Hasta	Remuneración	
....	OBJ_VENTMENS	Objetivo	0,00	74,99	0,00 €
....	OBJ_VENTMENS	Objetivo	75,00	99,99	25,00 €
....	OBJ_VENTMENS	Objetivo	100,00		50,00 €
Indicador : Objetivo de ventas vehículos AUDI(3 elementos)					

Figura 5: Ejemplo de agrupación por nombre de indicador

- 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, seleccionando desde el primer elemento hasta el último (un clic al principio y otro al final).
- Filtrado: las tablas del sistema permiten filtrar los resultados por uno o varios campos. Para realizar esto, buscaremos la fila de filtrado (Figura 6), situada entre los encabezados de las columnas y los datos de la tabla, en un color gris para una mejor localización y distinción respecto a los datos.

Código	Indicador
<input type="checkbox"/>	<input type="checkbox"/>

Figura 6: Fila de filtrado

La línea de filtrado contiene los siguientes elementos:

- El icono que tacha un embudo a la izquierda borrará todos los filtros que tengamos creados en el momento (Figura 7).



Figura 7: Eliminar todos los filtros

- Cuando coloque el cursor sobre un campo de filtro, aparecerá el desplegable, el cual contiene tanto cada uno de los distintos elementos que son mostrados en la columna como elementos genéricos referidos a “Todos”, “Vacíos” y “No vacíos” (Figura 8).

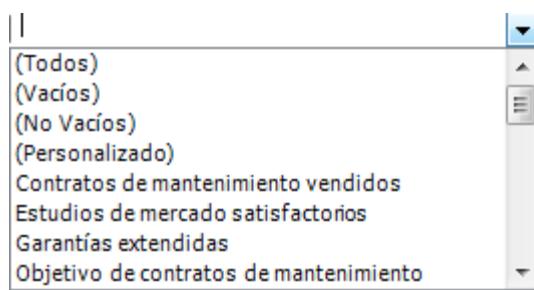


Figura 8: Desplegable de la línea de filtrado

- El icono de un cuadrado determinará qué comparación se hace entre los resultados y el campo de filtro propiamente dicho (Figura 9). 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

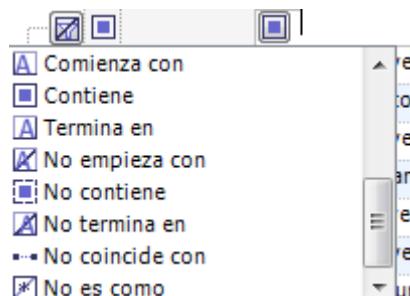


Figura 9: Distintos tipos de filtrado

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.

- Cuando coloque el cursor sobre un campo de filtro, aparecerá el icono con el que podrá eliminar el filtro establecido para esa columna en concreto (Figura 10).



Figura 10: Eliminar filtro de la columna

- 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.
- Copiar / pegar: En alguno de los campos de estas vistas, se puede utilizar el “Control + C” y posteriormente “Control + V” para extraer datos directamente de estas tablas a otras aplicaciones. Estos campos se reconocen porque al hacer clic sobre su celda, se colocan en modo edición para copiar su contenido (Figura 11).

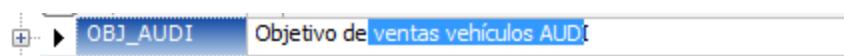


Figura 11: Selección del texto en la celda para copiar/pegar

- Exportación a Excel: En todas las tablas aparecerá un botón (Figura 12) 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.

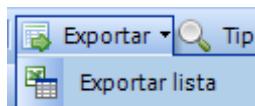


Figura 12: Botón de exportación a Excel

3. Indicadores

3.1 Indicadores / Barreras de entrada (pantalla principal)

A este módulo se accede mediante el botón de “Indicadores” colocado en la barra superior de herramientas, accediendo al listado de indicadores o barreras de entrada actuales de la organización (Figura 13), donde se muestran, además, los datos objetivos y reales conseguidos por las personas implicadas para esos indicadores o barreras de entrada, permitiéndose la edición de dichos datos (dependiendo de los privilegios que el usuario posea sobre los indicadores mostrados).

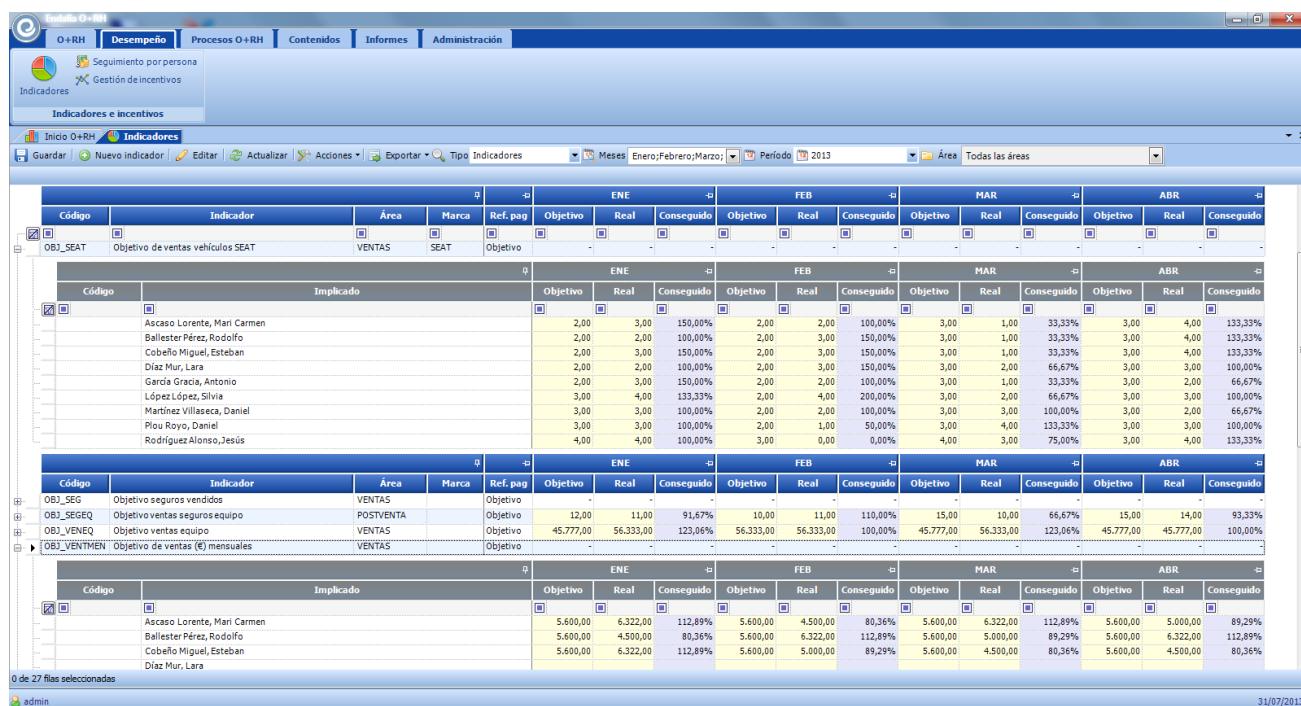


Figura 13: Indicadores y datos objetivos/reales

3.1.1 Modos de visualización de datos

En la barra de herramientas de la pantalla se encuentran, a disposición del usuario, distintas opciones de visualización de los datos:

- Opción “Tipo” (Figura 14): permite visualizar los indicadores o las barreras de entrada actuales de la organización.

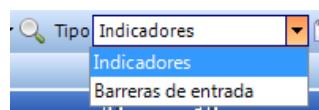


Figura 14: Tipo

- Opción “Período” (Figura 15): permite la visualización de los datos, tanto objetivos como reales, pertenecientes a los distintos meses del año seleccionado.

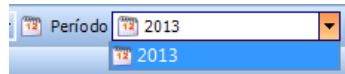


Figura 15: Período

- Opción “Meses” (Figura 16): permite mostrar únicamente los grupos de columnas que contienen los datos correspondientes a los meses seleccionados



Figura 16: Meses

- Opción “Área” (Figura 17): este desplegable muestra la estructura de áreas de la organización en forma de árbol, permitiendo visualizar únicamente los indicadores o barreras de entrada englobados dentro del área seleccionada.

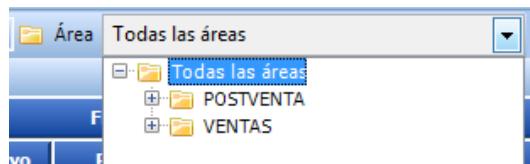


Figura 17: Área

3.1.2 Gestión de datos objetivos y reales

Para la gestión de los datos objetivos y reales de cada indicador o barrera de entrada, bastará con introducir un valor numérico en la celda editable correspondiente (marcada en color amarillo): si el dato a introducir pertenece a un indicador colectivo (esto es, afecta a las personas implicadas por igual), se deberá introducir el dato a nivel del indicador. Si por el contrario el indicador es individual (puede afectar de modo distinto a cada persona implicada), el dato deberá introducirse a nivel de la persona implicada. Para evitar posibles confusiones y, además, identificar fácilmente el comportamiento de un indicador, cuando éste posee comportamiento colectivo las celdas a nivel de indicador aparecen marcadas como editables (color amarillo y siempre que se posean los privilegios de usuario necesarios) y las celdas a nivel de personas implicadas aparecen deshabilitadas (Figura 18), mientras que si el comportamiento del indicador es individual, las celdas a nivel de indicador aparecerán deshabilitadas mientras que las celdas a nivel de empleados serán

editables (Figura 19). Una vez editados los datos, se podrán almacenar en el sistema pulsando sobre el botón “Guardar” de la barra de herramientas.

Código	Indicador	Área	Marca	Ref. pag	ENE			FEB			MAR			ABR			Obj
					Objetivo	Real	Conseguido										
OBJ_SEGEQ	Objetivo ventas segurosequipos	POSTVENTA		Objetivo	12,00	11,00	91,67%	10,00	11,00	110,00%	15,00	10,00	66,67%	15,00	14,00	93,33%	
Código	Implicado				Objetivo	Real	Conseguido	Obj									
	Ascaso Lorente, Mari Carmen				-	-	-	-	-	-	-	-	-	-	-	-	
	Ballester Pérez, Rodolfo				-	-	-	-	-	-	-	-	-	-	-	-	
	Cobello Miguel, Esteban				-	-	-	-	-	-	-	-	-	-	-	-	
	Díaz Mur, Lara				-	-	-	-	-	-	-	-	-	-	-	-	
	García Gracia, Antonio				-	-	-	-	-	-	-	-	-	-	-	-	
	López López, Silvia				-	-	-	-	-	-	-	-	-	-	-	-	
	Martínez Villaseca, Daniel				-	-	-	-	-	-	-	-	-	-	-	-	
	Plou Royo, Daniel				-	-	-	-	-	-	-	-	-	-	-	-	
	Rodríguez Alonso, Jesús				-	-	-	-	-	-	-	-	-	-	-	-	

Figura 18: Edición de datos de indicador colectivo

Código	Indicador	Área	Marca	Ref. pag	ENE			FEB			MAR			ABR			Obj
					Objetivo	Real	Conseguido										
OBJ_SEAT	Objetivo de ventas vehículos SEAT	VENTAS	SEAT	Objetivo	-	-	-	-	-	-	-	-	-	-	-	-	
Código	Implicado				Objetivo	Real	Conseguido	Obj									
	Ascaso Lorente, Mari Carmen				2,00	3,00	150,00%	2,00	2,00	100,00%	3,00	1,00	33,33%	3,00	4,00	133,33%	
	Ballester Pérez, Rodolfo				2,00	2,00	100,00%	2,00	3,00	150,00%	3,00	1,00	33,33%	3,00	4,00	133,33%	
	Cobello Miguel, Esteban				2,00	3,00	150,00%	2,00	3,00	150,00%	3,00	1,00	33,33%	3,00	4,00	133,33%	
	Díaz Mur, Lara				2,00	2,00	100,00%	2,00	3,00	150,00%	3,00	2,00	66,67%	3,00	3,00	100,00%	
	García Gracia, Antonio				2,00	3,00	150,00%	2,00	2,00	100,00%	3,00	1,00	33,33%	3,00	2,00	66,67%	
	López López, Silvia				3,00	4,00	133,33%	2,00	4,00	200,00%	3,00	2,00	66,67%	3,00	3,00	100,00%	
	Martínez Villaseca, Daniel				3,00	3,00	100,00%	2,00	2,00	100,00%	3,00	3,00	100,00%	3,00	2,00	66,67%	
	Plou Royo, Daniel				3,00	3,00	100,00%	2,00	1,00	50,00%	3,00	4,00	133,33%	3,00	3,00	100,00%	
	Rodríguez Alonso, Jesús				4,00	4,00	100,00%	3,00	0,00	0,00%	4,00	3,00	75,00%	3,00	4,00	133,33%	

Figura 19: Edición de datos de indicador individual

Si se han introducido datos erróneos o no deseados, puede recuperarse la última información almacenada pulsando el botón “Actualizar” (Figura 20) de la barra de herramientas, siendo recargados los datos siempre que descartemos el guardado que aparece en el mensaje de advertencia por haber realizado modificaciones y no haber sido guardadas.

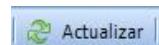


Figura 20: Botón “Actualizar”

3.1.3 Eliminar indicador/barrera de entrada

Para eliminar definitivamente uno o varios indicadores (o barreras de entrada), seleccione el dato o datos a eliminar, acceda al menú superior “Acciones” en la barra de herramientas y seleccione la opción “Eliminar” (Figura 21). Esta acción provocará, tras la confirmación de la misma, la eliminación definitiva de los indicadores seleccionados, junto con todos sus datos asociados, que no podrán ser recuperados.



Figura 21: Botón “Eliminar” indicador/barrera de entrada

3.1.4 Cambiar comportamiento del indicador/barrera de entrada

El menú “Acciones” de la barra de herramientas permite también la opción de cambiar el comportamiento de individual a colectivo de uno o varios indicadores o barreras de entrada, sin tener que acceder a la correspondiente ficha. Para ello, seleccione el dato o datos cuyo comportamiento desee modificar, acceda al menú “acciones” y seleccione la opción “Cambiar comportamiento a individual” (Figura 22). Esta acción provocará, tras la confirmación de la misma, el cambio de comportamiento de los indicadores seleccionados a modo individual.

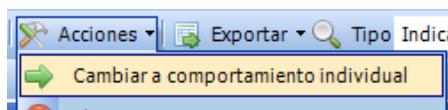


Figura 22: Botón “Cambiar a comportamiento individual”

3.1.5 Editar indicador/barrera de entrada

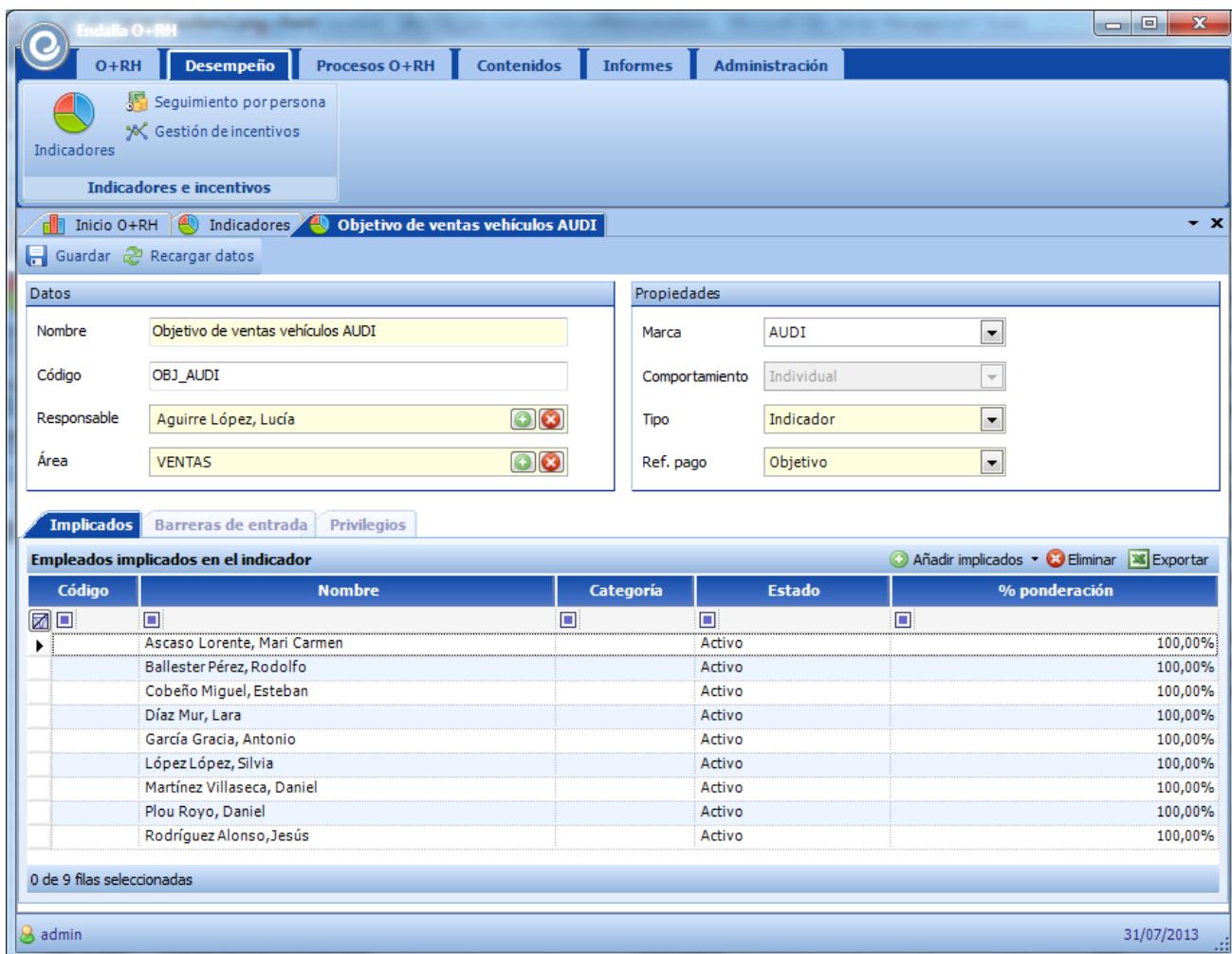
Para editar la información de un indicador o barrera de entrada, pulse el botón “Editar” de la barra de herramientas, el cual abrirá la ficha del indicador o barrera de entrada seleccionada, permitiéndole editar la información correspondiente, tal como se detallará en el punto 3.2.

3.1.6 Añadir indicador/barrera de entrada

Para añadir un nuevo indicador o barrera de entrada, pulse el botón “Añadir indicador” de la barra de herramientas, el cual abrirá la ficha del indicador o barrera de entrada seleccionada, permitiéndole introducir la información correspondiente, tal como se detallará en el punto 3.2.

3.2 Ficha del indicador/barrera de entrada

Al hacer doble clic sobre un indicador o barrera de entrada, o hacer clic en “Editar” cuando tenemos una o varias filas seleccionadas, aparecerá su ficha (Figura 23). Ésta ficha aparecerá también cuando pulsemos el botón “Añadir indicador”, con la diferencia de que todos los campos estarán vacíos y el título de la pestaña que contiene la ficha será “Nuevo indicador” en lugar del nombre del indicador o barrera de entrada seleccionada.



Datos		Propiedades		
Nombre	Objetivo de ventas vehículos AUDI	Marca	AUDI	
Código	OBJ_AUDI	Comportamiento	Individual	
Responsable	Aguirre López, Lucía	Tipo	Indicador	
Área	VENTAS	Ref. pago	Objetivo	

Empleados implicados en el indicador				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ascaso Lorente, Mari Carmen	Activo	100,00%
		Ballester Pérez, Rodolfo	Activo	100,00%
		Cobeño Miguel, Esteban	Activo	100,00%
		Díaz Mur, Lara	Activo	100,00%
		García Gracia, Antonio	Activo	100,00%
		López López, Silvia	Activo	100,00%
		Martínez Villaseca, Daniel	Activo	100,00%
		Plou Royo, Daniel	Activo	100,00%
		Rodríguez Alonso, Jesús	Activo	100,00%

0 de 9 filas seleccionadas

admin 31/07/2013

Figura 23: Ficha del indicador/barrera de entrada

En la ficha de un indicador podremos editar los datos de varios ámbitos, como el nombre, el código identificativo, la persona de la organización responsable del mismo, el área al que pertenece, el comportamiento del indicador (individual/colectivo), el tipo (indicador o barrera de entrada) o la referencia de pago de los incentivos asociados (por cumplimiento de objetivo o por unidad conseguida).

La ficha de un indicador posee, además, 3 pestañas que permiten gestionar el resto de información asociada al mismo: “Implicados”, “Barreras de entrada” y “Privilegios”.

3.2.1 Implicados

Esta pestaña permite gestionar la lista de personas de la organización que se encuentran implicadas en el indicador o barrera de entrada (Figura 24). En esta pestaña podremos añadir personas implicadas (menú “Añadir implicados”, clic en “Buscar por nombre” o “Buscar por puesto activo”, y seleccionando una o varias personas de la lista mostrada), eliminar personas existentes en la lista de implicados (botón “Eliminar”), o, modificando el valor de la celda correspondiente a la columna “% ponderación”, editar el peso que el indicador posee sobre cada persona, es decir, el porcentaje del incentivo asociado al indicador que la persona recibirá en caso de alcanzar el objetivo correspondiente.

Empleados implicados en el indicador				
Código	Nombre	Categoría	Estado	% ponderación
	Ascaso Lorente, Mari Carmen		Activo	100,00%
	Ballester Pérez, Rodolfo		Activo	100,00%
	Cobeño Miguel, Esteban		Activo	100,00%
	Díaz Mur, Lara		Activo	100,00%
	García Gracia, Antonio		Activo	100,00%
	López López, Silvia		Activo	100,00%
	Martínez Villaseca, Daniel		Activo	100,00%
	Plou Royo, Daniel		Activo	100,00%
	Rodríguez Alonso, Jesús		Activo	100,00%

0 de 9 filas seleccionadas

Figura 24: Pestaña “Implicados”

3.2.2 Barreras de entrada

Cuando la ficha abierta pertenezca a un indicador, esta pestaña permitirá gestionar las barreras de entrada que afectan a dicho indicador (Figura 25). En esta pestaña se permitirá añadir nuevas barreras de entrada que afectan al indicador (botón “Añadir” y seleccionando una o varias barreras de entrada en el listado que se muestra) o eliminar barreras de entrada de la lista de barreras que afectan al indicador (botón “Eliminar”).

Cuando la ficha abierta pertenezca a una barrera de entrada, esta pestaña se comporta de modo inverso al anteriormente descrito (Figura 26), es decir, permitirá gestionar los indicadores a los que la barrera de entrada seleccionada afecta (el texto de la pestaña cambiará a “Indicadores afectados”).

Barreras de entrada que afectan al indicador 'Objetivo de ventas vehículos AUDI'	
Código	Barrera de entrada
BARR_AUS	Ausencias mensuales
BARR_CLIENTINS	Clientes insatisfechos mensuales
OBJ_CLIENT	Clientes satisfechos mensuales
UNID_NOSEG	Clientes sin seguimiento

0 de 4 filas seleccionadas

Figura 25: Pestaña “Barreras de entrada”

Indicadores afectados		Privilegios
Indicadores afectados por la barrera de entrada 'Ausencias mensuales'		
Código	Indicador	
<input checked="" type="checkbox"/> <input type="checkbox"/> ► OBJ_AUDI	Objetivo de ventas vehículos AUDI	
<input type="checkbox"/> UNID_JAGUAR	Unidad VendidaVehículos JAGUAR	

0 de 2 filas seleccionadas

Figura 26: Pestaña “Indicadores afectados”

3.2.3 Privilegios

Esta pestaña permite gestionar la lista de personas que poseen privilegios de modificación de los datos (objetivos y reales) sobre el indicador o barrera de entrada seleccionada (Figura 27). En esta pestaña podremos añadir personas con privilegios (menú “Añadir implicados”, clic en “Buscar por nombre” o “Buscar por puesto activo”, y seleccionando una o varias personas de la lista mostrada), eliminar personas existentes en la lista (botón “Eliminar”), o, modificando el valor de una de las celdas correspondientes al grupo de columnas “Privilegios”, editar el tipo de privilegio que la persona poseerá sobre la edición de datos asociados al indicador.

Empleados		Privilegios				Observaciones
Código	Nombre	Categoría	Modificación objetivo	Modificación real		
<input checked="" type="checkbox"/> <input type="checkbox"/> ► Ascaso Lorente, Mari Carmen			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Ballester Pérez, Rodolfo			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/> Cobeño Miguel, Esteban			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> Díaz Mur, Lara			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> García Gracia, Antonio			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> López López, Silvia			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> Martínez Villaseca, Daniel			<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Empleado no incluido en la lista de implicados del indicador.
<input type="checkbox"/> Plou Royo, Daniel			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Empleado no incluido en la lista de implicados del indicador.
<input type="checkbox"/> Rodríguez Alonso, Jesús			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Empleado no incluido en la lista de implicados del indicador.

0 de 9 filas seleccionadas

Figura 27: Pestaña “Privilegios”

4. Gestión de incentivos

4.1 Pantalla de gestión de incentivos/penalizaciones

Desde la pantalla de gestión de incentivos podremos gestionar las remuneraciones asociadas a los indicadores (Figura 28).

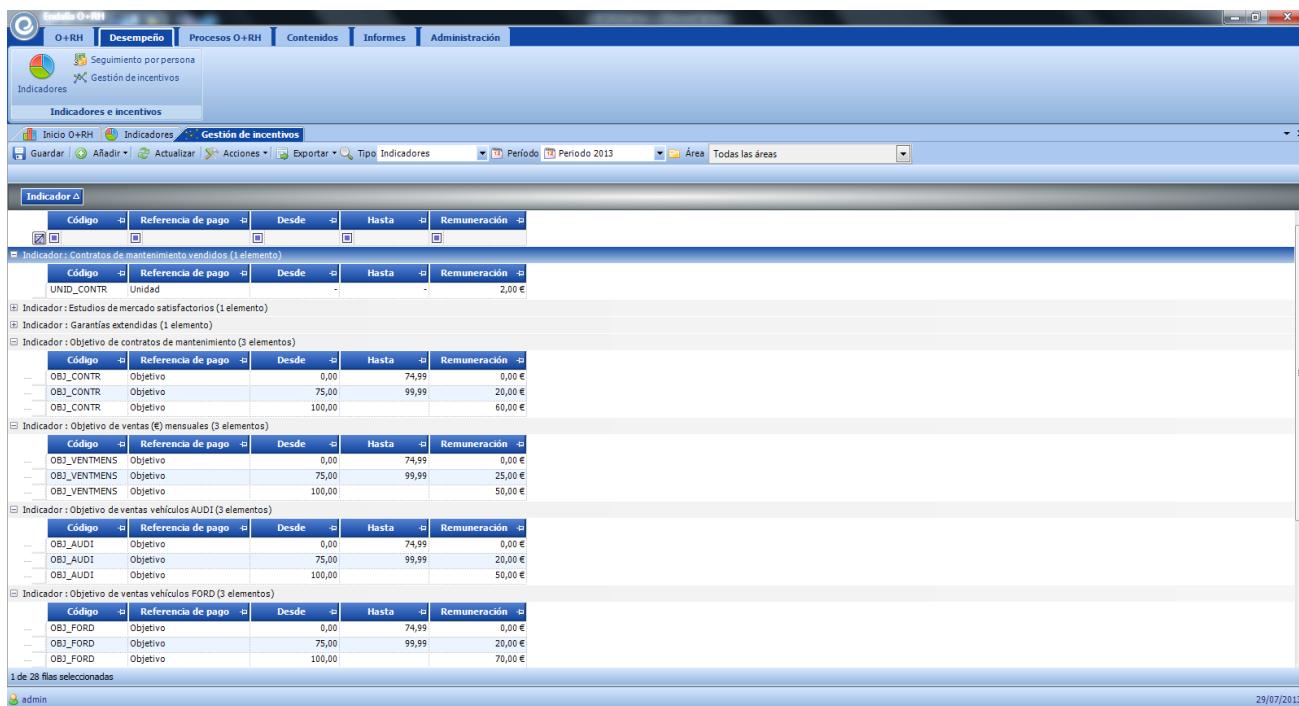


Figura 28: Gestión de incentivos

4.1.1 Visualización de incentivos

De modo análogo a la pantalla de indicadores, en la barra de herramientas de la pantalla de gestión de incentivos se encuentran, a disposición del usuario, distintas opciones de visualización de los datos:

- Opción “Tipo” (Figura 29): permite visualizar los incentivos asociados a los indicadores o las penalizaciones asociadas a las barreras de entrada actuales de la organización.

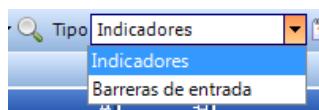


Figura 29: Tipo

- Opción “Período” (Figura 30): permite la visualización de los incentivos asociados al período concreto.

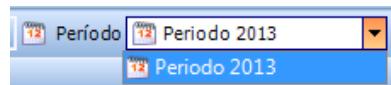


Figura 30: Período

- Opción “Área” (Figura 31): este desplegable muestra la estructura de áreas de la organización en forma de árbol, permitiendo visualizar únicamente los incentivos de los indicadores o las penalizaciones de las barreras de entrada englobadas dentro del área seleccionada.

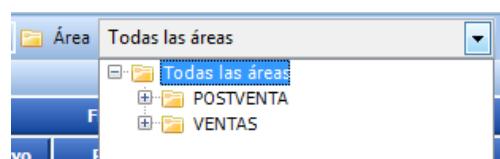


Figura 31: Área

4.1.2 Añadir incentivo/penalización

Para añadir un nuevo incentivo asociado a un indicador (o penalización asociada a una barrera de entrada), acceda al menú “Añadir” donde se le ofrecerán 2 opciones (Figura 32):

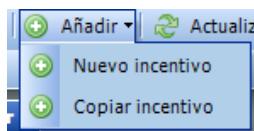


Figura 32: Menú “Añadir incentivo”

- “Añadir nuevo incentivo”: a partir de una nueva fila en blanco, podrá introducir los datos del nuevo incentivo (indicador asociado, valores de escalado “Desde” y “Hasta” y remuneración).
- “Copiar incentivo”: se realizará una copia de la información contenida en la fila o filas seleccionadas referentes al indicador asociado, pudiendo añadir nuevos valores de escalado (“Desde” y “Hasta”) para un incentivo asociado a un indicador ya existente en el listado.

4.1.3 Editar incentivos

Si desea editar tanto los valores de escalado (“Desde” y “Hasta”) como la remuneración asociada a un incentivo, bastará con introducir los nuevos datos en las celdas editables correspondientes y pulsar el botón “Guardar” de la barra de herramientas para almacenar la información.

Si desea cambiar el indicador al que se encuentra asociado un incentivo existente, deberá hacer clic en el botón de la celda con el símbolo “+” (Figura 33), y seleccionar, de la lista de indicadores disponibles que se muestra, el nuevo indicador al que asociar el incentivo seleccionado. Una vez aceptado el indicador, bastará pulsar el botón “Guardar” para almacenar el cambio en el incentivo.

Indicador	Código
► Objetivo de ventas vehículos AUDI	   

Figura 33: Edición del indicador asociado al incentivo

Si se han introducido datos erróneos o no deseados en la edición de los incentivos, puede recuperarse la última información correcta almacenada pulsando el botón “Actualizar” de la barra de herramientas, siendo recargados los datos de incentivos en la tabla.

4.1.4 Eliminar incentivo/penalización

Para eliminar definitivamente uno o varios incentivos (o penalizaciones asociadas a barreras de entrada), seleccione el dato o datos a eliminar, acceda al menú superior “Acciones” en la barra de herramientas y seleccione la opción “Eliminar”. Esta acción provocará, tras la confirmación de la misma, la eliminación definitiva de los incentivos seleccionados, junto con todos sus datos asociados, que no podrán ser recuperados.

5. Seguimiento por persona

5.1 Pantalla de gestión de remuneraciones recibidas por las personas

Desde esta pantalla podremos visualizar las remuneraciones a recibir por las personas de la organización implicadas en el sistema de indicadores, así como realizar la validación de dichas remuneraciones.

5.1.1 Modos de visualización de remuneraciones

La barra de herramientas de la pantalla ofrece distintas opciones de visualización de los datos:

- Opción “Vista” (Figura 34): permite la visualización de las remuneraciones a recibir por las personas de la organización de dos modos distintos:

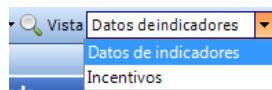


Figura 34: Vista

- De modo detallado (Figura 35), mostrando los incentivos desglosados junto con los indicadores de los que provienen y sus datos, así como las penalizaciones que producen las barreras de entrada (vista “Datos de indicadores”) y el incentivo total mensual a recibir por la persona. En esta vista, los distintos tipos de indicadores de acuerdo a su referencia de pago (indicadores de referencia de pago por cumplimiento de objetivos, indicadores de referencia de pago por unidad conseguida, y barreras de entrada que penalizan) se muestran en distintos colores, para permitir una distinción rápida y a primera vista del origen y el significado de los datos mostrados.
- De modo general (Figura 36), agrupando las remuneraciones totales a recibir por los meses del año seleccionado.

Screenshot of the Endalia O+RH software interface showing a detailed view of employee compensation.

The main menu tabs include: Inicio O+RH, Indicadores, Seguimiento por persona, Gestión de incentivos, Desempeño, Procesos O+RH, Contenidos, Informes, and Administración.

The current view is under the "Seguimiento por persona" tab, specifically the "Gestión de incentivos" section.

Filter settings at the top: Meses (Enero;Febrero;Marzo), Período (2013).

Table headers: Código, Empleado, Nivel, Centro de trabajo.

Employee details: Ascaso Lorente, Mari Carmen, Centro de trabajo: COCHES BUFALO SL.

Table rows show various incentive objectives and their performance metrics (Objetivo, Real, Conseguido - €/ud., Variación barreras, Total (€)) for different categories like VENTAS, AUDI, FORD, etc.

Section header: REMUNERACIÓN MENSUAL.

Table headers: Código, Empleado, Nivel, Centro de trabajo.

Employee details: Ballester Pérez, Rodolfo, Centro de trabajo: COCHES BUFALO SL.

Employee details: Cobello Miguel, Estéban, Centro de trabajo: COCHES BUFALO SL.

Employee details: Diaz Mur, Lara, Centro de trabajo: COCHES BUFALO SL.

Message: 0 de 9 filas seleccionadas.

User information: admin, Date: 31/07/2013.

Figura 35: Vista detallada de las remuneraciones a recibir por las personas

Screenshot of the Endalia O+RH software interface showing a general view of monthly compensation totals.

The main menu tabs are the same as in Figure 35.

The current view is under the "Seguimiento por persona" tab, specifically the "Incentivos" section.

Filter settings at the top: Meses (Enero;Febrero;Marzo), Período (2013).

Table headers: Código, Empleado, Nivel, Centro de trabajo.

Employee details: Ascaso Lorente, Mari Carmen, Centro de trabajo: COCHES BUFALO SL.

Table rows show monthly incentive amounts (Mes, Estado, Incentivo) for January, February, March, April, and May, totaling 895,00 €.

Table headers: Código, Empleado, Nivel, Centro de trabajo.

Employee details: Ballester Pérez, Rodolfo, Centro de trabajo: COCHES BUFALO SL.

Employee details: Cobello Miguel, Estéban, Centro de trabajo: COCHES BUFALO SL.

Employee details: Diaz Mur, Lara, Centro de trabajo: COCHES BUFALO SL.

Employee details: Garcia Gracia, Antonio, Centro de trabajo: COCHES BUFALO SL.

Employee details: Lopez Lopez, Silvia, Centro de trabajo: COCHES BUFALO SL.

Employee details: Martinez Villaseca, Daniel, Centro de trabajo: COCHES BUFALO SL.

Employee details: Piou Royo, Daniel, Centro de trabajo: COCHES BUFALO SL.

Table rows show monthly incentive amounts (Mes, Estado, Incentivo) for January, February, March, April, and May, totaling 90,00 €.

Message: 0 de 9 filas seleccionadas.

User information: admin, Date: 31/07/2013.

Figura 36: Vista general de las remuneraciones totales mensuales

- Opción “Período” (Figura 37): permite la visualización de los incentivos a recibir en los meses del año seleccionado.

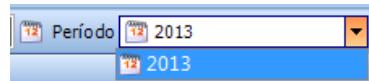


Figura 37: Período

- Opción “Meses” (Figura 38): permite mostrar únicamente los grupos de columnas que contienen los datos correspondientes a los meses seleccionados



Figura 38: Meses

5.1.2 Validar remuneración total mensual

Para validar la remuneración a recibir por una de las personas implicadas en el sistema de indicadores de la organización, desde la vista “Datos de indicadores”, seleccione la persona de la cual desea validar el pago mensual, acceda al menú “Acciones” y seleccione la opción “Cerrar pago mensual”. Se le mostrará un nuevo menú con varias opciones (Figura 39), una por cada mes del año seleccionado en el desplegable “Período” de la barra de herramientas. Haga clic en el botón correspondiente al mes del que desea validar el pago. Si la persona seleccionada, para el mes elegido, posee todos los datos de indicadores y barreras de entrada completados correctamente, la aplicación le pedirá la confirmación del cierre del pago mensual. Tras pulsar el botón “Aceptar” del mensaje de confirmación, el incentivo total mensual será almacenado y mostrado como “Pagado” tanto en la vista “Datos de indicadores” (Figura 40) como en la vista general “Incentivos” (Figura 41), y no podrá sufrir cambios posteriores aunque se modifiquen los datos de los incentivos asociados a los indicadores que afectaban a la persona, o los datos objetivos y/o reales de dichos indicadores para el mes que ha sido validado.

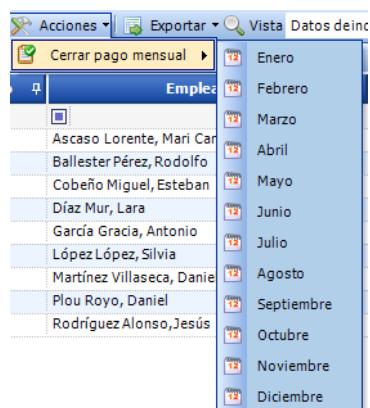


Figura 39: Menú “Cerrar pago mensual”

UNID_RETIR	Retrasos mensuales	- POSTVENTA	-	-	1,00	-	-	-	0,00	-	-
UNIT_REPINS	Reparaciones insatisfactorias para el cliente	- POSTVENTA	-	-	1,00	-	-	-	2,00	-	-
REMUNERACIÓN MENSUAL											
						PAGADO	70,00 €			PAGADO	75,00 €

Figura 40: Pagos mensuales validados en la vista “Datos de indicadores”

Código	Empleado	Nivel
	Ascaso Lorente, Mari Carmen	
Mes	Estado	Incentivo
Enero	PAGADO	70,00 €
Febrero	PAGADO	75,00 €

Figura 41: Pagos mensuales validados en la vista “Incentivos”

6. Bibliografía

6.1 Referencias

En este documento no se realizan referencias a otros documentos.

6.2 Referencias web

En este documento no se realizan referencias a recursos disponibles en la Web.

Sistema de Gestión de Objetivos y Retribución de Incentivos

Instalación

Versión 1.2

Publicado el 02/08/2013

Copyright © 2013 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 Endalia, 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 Endalia, S.L..

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

Histórico de revisiones

Fecha	Versión	Descripción	Autor
28/07/2013	1.0	Redacción inicial del documento	Santiago Pelegrín Costea
29/07/2013	1.1	Revisión del documento	Fernando Cortés Franco
02/08/2013	1.2	Corrección del documento	Santiago Pelegrín Costea

Índice

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. Características del instalador	6
3. Instalación de la aplicación	7
4. Bibliografía	11
4.1 Referencias.....	11
4.2 Referencias Web	11

1. Introducción

1.1 Propósito del documento

El objetivo principal del documento se basa en describir las características principales del instalador que se ha desarrollado para el SGORI, así como ilustrar el proceso de instalación de la aplicación.

1.2 Alcance del documento

El alcance del presente documento comprende la fase final del proceso de desarrollo del SGORI.

1.3 Acrónimos

- SGORI: Sistema de Gestión de Objetivos y Retribución de Incentivos.

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 realiza ninguna referencia a otros documentos del proyecto.

1.6 Resumen

Este documento describe el proceso de instalación de la aplicación que contiene el SGORI. Se compone de cuatro apartados:

- Apartado 1: introducción del documento, definición del propósito y alcance del mismo.
- Apartado 2: características principales del instalador desarrollado para el SGORI.
- Apartado 3: descripción del proceso de instalación de la aplicación.
- Apartado 4: bibliografía y referencias Web utilizadas en esta fase del proyecto.

2. Características del instalador

Para el desarrollo del instalador de la aplicación, se ha utilizado “Inno Setup”, un generador gratuito de instaladores para Windows.

Las principales características de InnoSetup son:

- Instalador basado en scripts.
- Soporta todas las versiones modernas de Windows (desde Windows 95 hasta Windows 7).
- Soporta arquitecturas de 64 bits.
- Permite crear un único ejecutable (.exe) de instalación, para facilitar la distribución sencilla de programas online.
- Permite crear distintos tipos de instalación: completa, mínima, configurable.
- Permite desinstalar completamente el programa.
- Permite crear accesos directos en cualquier lugar, incluyendo el menú de inicio y el escritorio.
- Permite crear entradas .INI y en el registro.
- Permite ejecutar otros programas antes, durante o después de la instalación.
- Soporte para instalaciones multilingües.
- Soporte para instalaciones restringidas por contraseña o encriptación.
- Soporte para crear instaladores firmados digitalmente.

Los requerimientos necesarios para poder ejecutar la aplicación son:

- Sistema operativo Windows.
- .NET Framework 2.0 o superior.
- SQL Express 2005 o superior.

Es por ello que, durante el proceso de instalación, se analiza el registro de Windows para descubrir si el usuario ya tiene instalado .NET Framework 3.5 (o una versión posterior) y SQL Express 2008 (o una versión posterior). Si no es así, el instalador descarga ambos componentes de Internet.

Una vez que .NET Framework y el motor de base de datos están correctamente instalados, se crea e inicializa la base de datos necesaria para persistir los datos manejados por la aplicación.

En último lugar, se procede a la instalación de la aplicación desarrollada.

3. Instalación de la aplicación

En este apartado se describe el proceso de instalación de la aplicación.

Para iniciar el asistente de instalación, basta con hacer doble clic sobre su ícono. Se abrirá la siguiente ventana (Figura 1), que muestra la pantalla de bienvenida del asistente. Para continuar, se debe pulsar el botón ‘Siguiente’.

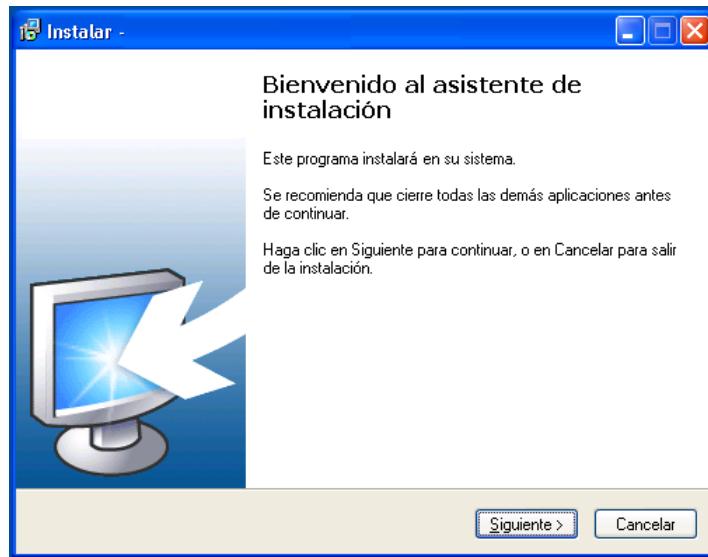


Figura 1: Ventana de bienvenida al asistente de instalación

La siguiente ventana (Figura 2), permite seleccionar el directorio destino de la instalación. Para buscar el directorio deseado, se puede pulsar el botón ‘Examinar’, para visualizar la estructura de carpetas del equipo, o bien teclear directamente la ruta en el cuadro de texto. Para continuar, se debe pulsar el botón ‘Siguiente’.

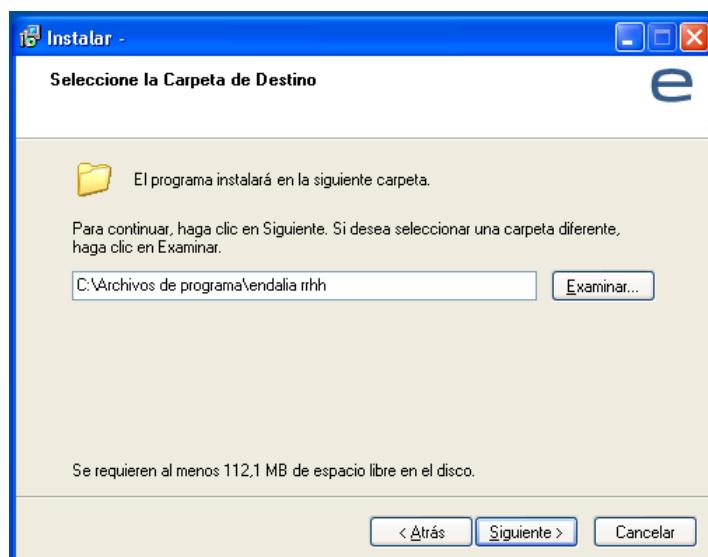


Figura 2: Ventana de selección de carpeta destino

La ventana que aparece a continuación (Figura 3), permite seleccionar la carpeta del menú inicio donde se creará el acceso directo al programa. También es posible no crear ninguna carpeta en el menú inicio, seleccionando el checkbox de la parte inferior de la pantalla. Una vez más, se debe pulsar el botón ‘Siguiente’ para continuar con la instalación.

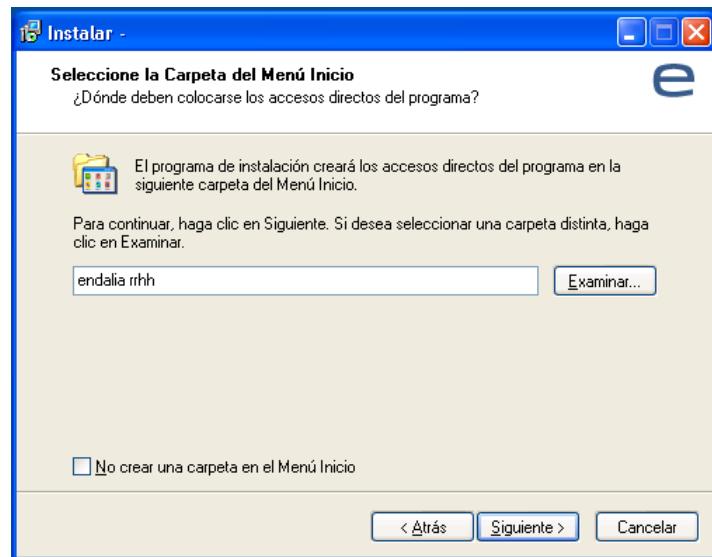


Figura 3: Ventana de selección de carpeta del menú Inicio

En la siguiente ventana (Figura 4), se pueden especificar tareas adicionales a realizar por el asistente de instalación, como son: crear un ícono en el escritorio y crear un ícono de inicio rápido en la barra de herramientas. Para comenzar con la instalación propiamente dicha, se debe pulsar el botón ‘Siguiente’.

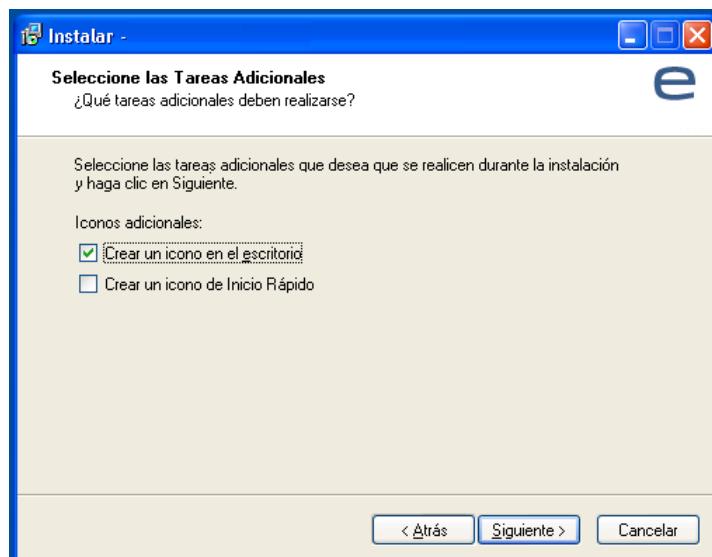


Figura 4: Ventana de tareas adicionales

Una vez seleccionadas las opciones de instalación deseadas, el asistente de instalación comprueba qué componentes requeridos no están actualmente instalados, y procede a descargarlos de Internet y a instalarlos en el equipo. La siguiente ventana (Figura 5) se muestra mientras se llevan a cabo dichas acciones, informando al usuario del proceso de descarga e instalación de los diferentes componentes.

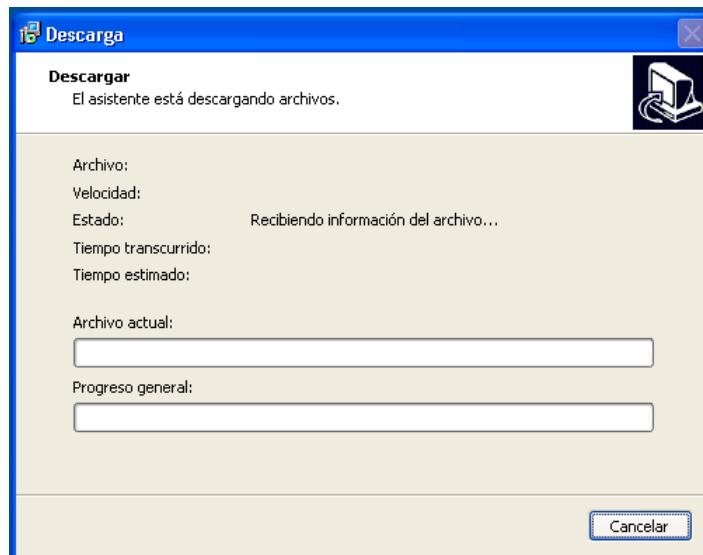


Figura 5: Ventana de descarga de componentes adicionales

Una vez descargados e instalados los componentes necesarios, comienza la instalación de la aplicación. La siguiente ventana (Figura 6), muestra la evolución de la instalación de la aplicación.

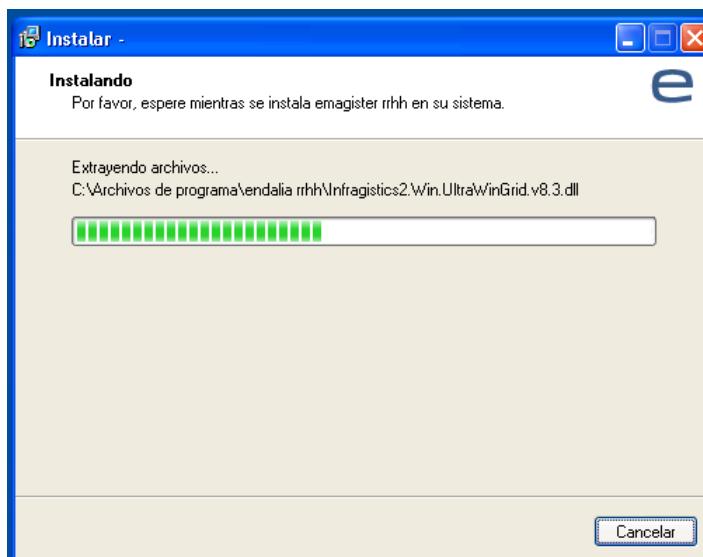


Figura 6: Ventana de progreso de la instalación

Cuando el proceso de instalación concluye, se muestra la siguiente ventana (Figura 7) de despedida del asistente de instalación. Dicha ventana informa de que el proceso ha concluido satisfactoriamente, y permite ejecutar directamente la aplicación. Para concluir el proceso de instalación, basta con pulsar el botón 'Finalizar'.



Figura 7: Ventana de despedida del asistente de instalación

4. Bibliografía

4.1 Referencias

En este documento no se realizan referencias.

4.2 Referencias Web

- [Ref. Web 1] <http://www.wikipedia.org>
- [Ref. Web 2] <http://www.jrsoftware.org/isinfo.php>
- [Ref. Web 3] <http://www.vincenzo.net/>