

# Trabajo Fin de Grado

Desarrollo de un módulo de reclutamiento y  
selección implementado con servicios web seguros  
integrado en un sistema de gestión integral de  
recursos humanos

## ANEXOS

Autor

**Sergio Sanz Frías**

Director y Ponente

**Fernando Cortés Franco**

Director/CEO (Endalia)

**Santiago Velilla Marco**

Departamento de Informática e Ingeniería de sistemas (Universidad de Zaragoza)

Escuela de Ingeniería y Arquitectura

2015

Tomo 2/2



## ÍNDICE DE CONTENIDOS

- Estándar de documentación
- Estándar de codificación
- Plan de gestión de configuraciones
- Estudio de mercado
- Especificación de requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

# ESTÁNDAR DE DOCUMENTACIÓN

SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 2.2  
PUBLICADO EL 11/02/2015

Copyright © 2014 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
03/11/2014	2.0	Modificaciones del documento para adaptarlo al TFG	Sergio Sanz Frías
09/02/2015	2.1	Revisión del documento	Fernando Cortés Franco
11/02/2015	2.2	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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. Histórico de revisiones.....	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 trabajo de desarrollo de la AWSC. Se definen los formatos, diseños, tipología de fuentes y plantillas a utilizar en la elaboración de esta documentación.

## 1.2 Alcance del documento

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

## 1.3 Acrónimos

AWSC: Aplicación Web de Selección de Candidatos.

## 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 trabajo.

## 1.6 Resumen

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

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.
2. Se especifica el formato de documentación del proyecto.
3. Plantillas de documentación.
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 16, 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 9.

### 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 indicarán 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 60 y se incluye información de copyright y de control de distribución y autorización.

Todas las demás páginas del documento poseen un pie de página común, en el que aparece el logotipo de Endalia 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 del 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).

HISTÓRICO DE REVISIONES			
Fecha	Versión	Descripción	Autor
01/01/2014	1.0		


 TÍTULO DEL PROYECTO  
PÁGINA 3 DE 5

Figura 4. Hoja 3 – Histórico de revisiones




En esta hoja, aparte del 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.

INDICE	
Historico de revisiones.....	3
Indice.....	4
1. TITULO 1.....	5
1.1 TITULO 2.....	5
1.1.1 TITULO 3.....	5



TITULO DEL PROYECTO  
PÁGINA 4 DE 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.





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.

ACTA DE LA SESIÓN			
Sección 1			
Acuerdo	Responsable	Compromiso	Estado
		01/01/2014	0%
Sección 2			
Acuerdo	Responsable	Compromiso	Estado
		01/01/2014	0%

## 4. BIBLIOGRAFÍA

### 4.1 Referencias

- [R1] Edward J Huth *Scientific Style and Format: The CBE Manual for Authors, Editors, and Publishers* Cambridge University Press 1994
- [R2] Estándar de documentación de Endalia S.L.

### 4.2 Referencias Web

- [W1] <http://www.wikipedia.org>
- [W2] <http://www.monografias.com/trabajos6/dosi/dosi.shtml>
- [W3] <http://www.apa.org/journals/webref.html>



# ESTÁNDAR DE CODIFICACIÓN

SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 2.2  
PUBLICADO EL 11/02/2015

Copyright © 2014 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 del Estándar de codificación de Endalia	Endalia
03/11/2014	2.0	Modificaciones del documento para adaptarlo al TFG	Sergio Sanz Frías
09/02/2015	2.1	Revisión del documento	Fernando Cortés Franco
11/02/2015	2.2	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
1. Introducción.....	7
1.1 Propósito del documento .....	7
1.2 Alcance del documento.....	7
1.3 Acrónimos.....	7
1.4 Definiciones .....	7
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 subyacente .aspx.cs.....	9
2.2.1 Consideraciones generales sobre las regiones.....	11
2.2.2 Región directivas <i>using</i> .....	11
2.2.3 Región declaración <i>namespace</i> y <i>class</i> .....	11
2.2.4 Región declaración de variables .....	12
2.2.5 Región <i>Page Load</i> .....	12
2.2.6 Región <i>I18N</i> .....	12
2.2.7 Región <i>DataBind</i> .....	12
2.2.8 Región Eventos <i>DataGrid</i> .....	12
2.2.9 Región Eventos de botones .....	12
2.2.10 Región Métodos de ordenación .....	13
2.2.11 Región Código generado por el diseñador de <i>Web Forms</i> .....	13
2.3 Codificación de archivos de clase, acceso a datos y servicios web .cs .....	13
2.3.1 Región directivas <i>using</i> .....	14
2.3.2 Región declaración <i>namespace</i> y <i>class</i> .....	14
2.3.3 Región atributos .....	14
2.3.4 Región constructores .....	14
2.3.5 Región propiedades .....	14
2.3.6 Región métodos .....	14
2.4 Codificación de archivos de recursos de internacionalización.....	15
3. Reglas de codificación .....	16
3.1 Identación .....	16
3.1.1 Longitud de línea .....	16



3.1.2	Ruptura de líneas.....	16
3.2	Comentarios .....	16
3.2.1	Aplicación de los comentarios .....	17
3.2.2	Formatos de implementación de comentarios.....	18
3.3	Declaraciones .....	19
3.3.1	Número de declaraciones por línea.....	19
3.3.2	Inicialización .....	19
3.3.3	Situación.....	19
3.4	Sentencias .....	20
3.4.1	Sentencias simples.....	20
3.4.2	Sentencias compuestas.....	20
3.4.3	Sentencias de retorno .....	20
3.4.4	Sentencias <i>if, if-else, if else- if else</i> .....	20
3.4.5	Sentencias <i>for</i> .....	21
3.4.6	Sentencias <i>while</i> .....	21
3.4.7	Sentencias <i>do-while</i> .....	21
3.4.8	Sentencias <i>switch</i> .....	22
3.4.9	Sentencias <i>try-catch</i> .....	22
3.5	Espacios en blanco .....	23
3.5.1	Líneas en blanco.....	23
3.5.2	Espacios en blanco .....	23
3.6	Convenciones de nombres.....	24
3.6.1	Clases .....	24
3.6.2	Métodos.....	24
3.6.3	Variables y parámetros.....	24
3.6.4	Constantes.....	24
3.7	Hábitos de programación.....	25
3.7.1	Referencias a variables y métodos de clase.....	25
3.7.2	Constantes.....	25
3.7.3	Asignaciones de variables.....	25
3.7.4	Paréntesis .....	25
3.7.5	Variables de retorno .....	26
3.7.6	Expresiones antes de '?' en el operador condicional .....	26
3.7.7	Comentarios especiales .....	26
4.	Estándar de nombrado de Base de Datos.....	27
4.1	Idioma a utilizar.....	27



4.2	Convenciones de nombrado de tablas .....	27
4.2.1	Nombrado de tablas de entidad.....	27
4.2.2	Nombrado de tablas de relación.....	27
4.2.3	Nombres de tablas predefinidos .....	28
4.3	Convenciones de nombrado de campos.....	28
4.3.1	Nombrado de Campos de Tablas de Entidad.....	28
4.3.2	Nombrado de campos de tablas de relación.....	28
5.	Bibliografía.....	30
5.1	Referencias.....	30
5.2	Referencias Web .....	30





# 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 trabajo. 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 de la AWSC, 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 trabajo.

## 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:

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.
2. Se muestra la estructura de codificación de los diferentes tipos de archivos de código fuente desarrollados en el trabajo.
3. Se muestran reglas, recomendaciones y buenas prácticas para el desarrollo del código fuente del trabajo.
4. Se muestra el estándar de nombrado de los elementos de base de datos.
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, acceso a datos y servicios web (.cs). Los archivos de la interfaz de usuario (.aspx), cuyo contenido es en un 95% código HTML, son generados automáticamente por la herramienta de desarrollo por lo que este apartado especifica el estándar de codificación para los archivos de código subyacente (.aspx.cs).

### 2.2 Codificación de archivos de código subyacente .aspx.cs

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>	<pre>using System; &lt;directivas using&gt;</pre>
<2>Declaración <i>namespace</i> y <i>class</i>	<pre>namespace MyNamespacel {     public class MyClass     {</pre>
<3> Región variables	<pre>        #region variables              #region Variables de interfaz              &lt;declaraciones de variables de interfaz&gt;              #endregion Variables de interfaz              #region Variables I18N              &lt;declaraciones de variables de internacionalización&gt;              #endregion Variables I18N              #region Variables globales              &lt;declaraciones de variables globales&gt;              #endregion Variables globales          #endregion variables</pre>



<4> Región <i>Page_Load</i>	<pre>#region Page_Load  &lt;Código Page_Load&gt;  #endregion Page_load</pre>
<5> Región <i>I18N</i>	<pre>#region I18N      &lt;Código LoadI18N&gt;      &lt;Código class_Init&gt;  #endregion I18N</pre>
<6> Región <i>DataBind</i>	<pre>#region DataBind      &lt;Código DataBind&gt;  #endregion DataBind</pre>
<7> Región Eventos de Tablas <i>Grid</i>	<pre>#region Eventos Grid NombreTabla      #region ItemDataBound          &lt;Código ItemDataBound&gt;      #endregion ItemDataBound      #region ItemCommand          &lt;Código ItemCommand&gt;      #endregion ItemCommand      #region ItemEdit          &lt;Código ItemEdit&gt;      #endregion ItemEdit      #region ItemUpdate          &lt;Código ItemUpdate&gt;      #endregion ItemUpdate      #region ItemDelete          &lt;Código ItemDelete&gt;      #endregion ItemDelete      #region PageIndexChanged          &lt;Código PageIndexChanged&gt;      #endregion PageIndexChanged</pre>



	<pre> #region ItemCancel      &lt;Código ItemCancel&gt;  #endregion ItemCancel  #endregion Eventos Grid NombreTabla </pre>
<8> Región Eventos de botones	<pre> #region Eventos de botones      &lt;Código Eventos clic botones&gt;  #endregion Eventos de botones </pre>
<9> Región Métodos de ordenación	<pre> #region Métodos de ordenación      &lt;Código Métodos de ordenación&gt;  #endregion Métodos de ordenación </pre>
<10> Región Código generado Web Forms	<pre> #region Código generado por el diseñador de Web Forms  #endregion Código generado por el diseñador de Web Forms </pre>
	<pre> }  } </pre>

### 2.2.1 Consideraciones generales sobre las regiones

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

### 2.2.2 Región directivas *using*

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

### 2.2.3 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 © 2014, Endalia, S.L. Todos los derechos reservados
/// </summary>
```

#### 2.2.4 Región declaración de variables

Esta región estará integrada por tres subregiones que se especifican a continuación:

- Región variables de interfaz: en esta región aparecen las declaraciones de elementos de interfaz o controles Web, que son generados automáticamente por la herramienta de desarrollo de código al utilizarlos en el archivo de diseño aspx.
- Región variables *I18N*: en esta región aparecen las declaraciones de cadenas de internacionalización que son obtenidas del archivo de recursos y que se utilizan para definir todos los textos que son presentados al usuario.
- Región variables globales: en esta región aparecen las declaraciones de variables globales utilizadas acompañadas de una descripción de su funcionalidad.

#### 2.2.5 Región *Page Load*

En esta región se coloca el código del evento *Page\_Load* que se lanza cada vez que la página es lanzada o recargada.

#### 2.2.6 Región *I18N*

Esta región está integrada por dos métodos:

- Método *Init()*: realiza la lectura del fichero de recursos en el que se almacenan las cadenas de internacionalización, cargando estas en variables de cadena.
- Método *LoadI18N()*: carga en los campos de texto del aspx las variables de cadenas obtenidas en el método *Init*.

#### 2.2.7 Región *DataBind*

En esta región se colocan los métodos que enlazan orígenes de datos a controles de servidor como *DataGrids* o árboles.

#### 2.2.8 Región Eventos *DataGrid*

Se utilizan tantas de estas regiones como *DataGrids* aparezcan en el archivo aspx, definiendo dentro de cada una de ellas subregiones para cada uno de los métodos que capturan los diversos eventos de un *DataGrid* y que se especifican en el apartado 2.2

#### 2.2.9 Región Eventos de botones

En esta región se colocan los métodos que capturan los eventos clic de los distintos botones ubicados en el archivo aspx.



### 2.2.10 Región Métodos de ordenación

Esta región engloba los métodos que ordenan los diferentes orígenes de datos que son utilizados en el código.

### 2.2.11 Región Código generado por el diseñador de *Web Forms*

Esta región se genera automáticamente gracias a la herramienta de desarrollo, y consta de dos métodos:

- *InitializeComponent()*: contiene las declaraciones de los métodos del código que capturan los diferentes eventos producidos por los controles ubicados en el código aspx.
- *OnInit()*: este método se lanza al cargar la página y realiza la llamada al método *InitializeComponent* para comenzar la ejecución del código del servidor.

## 2.3 Codificación de archivos de clase, acceso a datos y servicios web .cs

A continuación se muestra la estructura de codificación de los archivos de definición de clases, acceso a datos y servicios web. 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>	<pre>using System; &lt;directivas using&gt;</pre>
<2>Declaración <i>namespace</i> y <i>class</i>	<pre>namespace MyNamespacel {     public class MyClass     {</pre>
<3> Región atributos	<pre>        #region atributos              &lt;Declaración atributos&gt;          #endregion atributos</pre>
<4> Región constructores	<pre>        #region constructores              &lt;Métodos constructores clase&gt;          #endregion constructores</pre>
<5> Región propiedades	<pre>        #region propiedades              &lt;declaración propiedades&gt;          #endregion propiedades</pre>
<6> Región Métodos	<pre>        #region métodos              &lt;Código Métodos&gt;          #endregion métodos</pre>
	<pre>    } }</pre>



### 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 © 2014, Endalia, 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 _offerID;
```

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 OfferID
{
    get{ return _offerID; }
    set{ _offerID = 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 .aspx 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_`
  - *tooltip* de un botón de imagen: `_btnImg_`
  - cabecera de columna de un *DataGrid*: `_gridHeader_`
- En el caso de que la etiqueta sea un *tooltip*, se colocará a continuación el sufijo `_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.

```
Login_lbl_lblDuration  
Login_btn_btnEnter  
Login_btn_btnEnter_ToolTip
```



## 3. REGLAS DE CODIFICACIÓN

### 3.1 Identación

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

#### 3.1.1 Longitud de línea

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

#### 3.1.2 Ruptura de líneas

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

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

### 3.2 Comentarios

En C# hay tres formas de escribir comentarios:

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

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

- Estos comentarios pueden abarcar tantas líneas como sea necesario. No es posible anidar comentarios de este tipo.
- En la segunda 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 sentimos obligados a llenarlo de comentarios, debemos considerar la reescritura del código para hacerlo más claro. Los comentarios no deben encerrarse en grandes cajas dibujadas con asteriscos u otros caracteres. Los comentarios nunca deberían incluir caracteres especiales como saltos de página, etc.

Los siguientes puntos son técnicas de comentarios recomendadas.

- Cuando se modifica el código, se mantienen siempre actualizados los comentarios circundantes.
- Evitar los comentarios recargados, como las líneas enteras de asteriscos. En su lugar se utilizan espacios para separar los comentarios y el código.
- Evitar rodear un bloque de comentarios con un marco tipográfico. Puede resultar agradable, pero es difícil de mantener.
- Antes de la implementación, quitar todos los comentarios temporales o innecesarios, para evitar cualquier confusión en la futura fase de mantenimiento.
- Si se necesita realizar comentarios para explicar una sección de código compleja, examinar el código para decidir si se debería volver a escribir. Siempre que sea posible, no documentar un código malo, 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 indentada al nivel del código que la sigue. Si un comentario no se puede escribir en una sola línea, debería seguir el formato de los bloques de comentario. Un comentario de una sola línea debería ir precedido de una sola línea en blanco. Para este tipo de comentario se preferirá utilizar los caracteres `/**`. A continuación se muestra un ejemplo:

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

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

```
if (a == 2)
{
    return TRUE;                // caso especial
}
else
{
    return isPrime(a);          /* otro comentario */
}
```



## 3.3 Declaraciones

### 3.3.1 Número de declaraciones por línea

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

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

se prefiere sobre:

```
int level, size;
```

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

```
int foo, fooarray[]; //Evitar
```

### 3.3.2 Inicialización

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

### 3.3.3 Situación

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

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

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

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

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

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

```
int count;

myMethod() {

    if (condition) {

        int count; // Evitar

    }

}
```



## 3.4 Sentencias

### 3.4.1 Sentencias simples

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

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

### 3.4.2 Sentencias compuestas

Las sentencias compuestas son sentencias que contienen listas de sentencias encerradas entre llaves (“{sentencias}”). Se ilustrarán ejemplos en las siguientes secciones.

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

Las únicas excepciones a 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;
}
```





## 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 operandos. Por ejemplo:

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

- Las expresiones de una sentencia deberán estar separadas por espacio. 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'; // ¡EVITAR!
```

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

```
if (c++ = d++) // ¡EVITAR!
{
    ...
}
```

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 4.2.3 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_AdministrationItems  
Orh_Job_JobOffers  
Orh_Emp_EmployeeRecruitmentPreferences
```

#### 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 'Orh\_Job\_JobOffers' y 'Orh\_Emp\_Employees' se creará la tabla 'R\_JobOffer\_Employee'.



### 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\_'. Como finalización del nombre tendrán la terminación 'ID'.
- Nombrado de claves ajenas: Las claves ajenas comenzaran con el prefijo 'fk\_'. 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\_Job\_JobOffer' todos los campos de la tabla en este punto empezarian por 'JobOffer'. Otro ejemplo: Si la tabla es 'Orh\_Emp\_Employees' los campos de la tabla empezarian por 'Employee' o 'Emp'.

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_EmpCountry  
pk_EmpID  
EmpName  
EmpGender
```

### 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\_Employee\_Language' que relaciona las tablas 'Orh\_Emp\_Employees' y 'Orh\_Emp\_Languages' se nombrarán de esta manera las claves ajenas:

```
fk_EmpLanEmployeeID  
fk_EmpLanLanguageID
```

- **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\_Employee\_Language' que relaciona las tablas 'Orh\_Emp\_Employees' y 'Orh\_Emp\_Languages' los nombres de los campos podrían ser:

```
EmpLanQualification  
EmpLanIsDeleted  
EmpLanLastUpdate
```



## 5. BIBLIOGRAFÍA

### 5.1 Referencias

[R1] Juval Lowy 2003. *C# Coding Standard*. Idesign Inc 2004.

### 5.2 Referencias Web

[W1] <http://www.dotnetspider.com/tutorials/bestpractices.aspx>

[W2] <http://msdn.microsoft.com/>

[W3] [http://vyaskn.tripod.com/object\\_naming.htm](http://vyaskn.tripod.com/object_naming.htm)





# PLAN DE GESTIÓN DE CONFIGURACIONES

SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 2.3  
PUBLICADO EL 11/02/2015

Copyright © 2014 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
04/11/2014	2.0	Modificaciones del documento para adaptarlo al TFG	Sergio Sanz Frías
12/11/2014	2.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	2.2	Revisión del documento	Fernando Cortés Franco
11/02/2015	2.3	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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. Especificaciones de gestión.....	7
3. Actividades de Gestión de Configuraciones.....	8
3.1 Identificación de la Configuración.....	8
3.2 Gestión de la Configuración.....	8
3.3 Control de cambios de la Configuración.....	9
4. Herramientas Utilizadas.....	10
5. Bibliografía.....	13
5.1 Referencias.....	13
5.2 Referencias Web .....	13



# 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 unívocamente, 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 trabajo.

## 1.2 Alcance del documento

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

## 1.3 Acrónimos

- AWSC: Aplicación Web de Selección de Candidatos.
- 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.
- 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 de la AWSC. Se compone de cinco apartados:

1. Introducción del documento, definición del propósito y alcance del mismo.
2. Descripción de la organización de la gestión de configuraciones.
3. Especificación de las tareas a realizar para la gestión de configuraciones.
4. Descripción de las herramientas utilizadas para la gestión de configuraciones.
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:** trata de 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:** para 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:** su finalidad es 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 trabajo 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; ya que los dos primeros puntos ya aportan una buena calidad en la gestión de configuraciones.



## 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 parte de 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, que resuma 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 trabajo, se definen las siguientes bibliotecas:

- Biblioteca de trabajo. Se establece al inicio del trabajo 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 en los anterior apartados, 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.

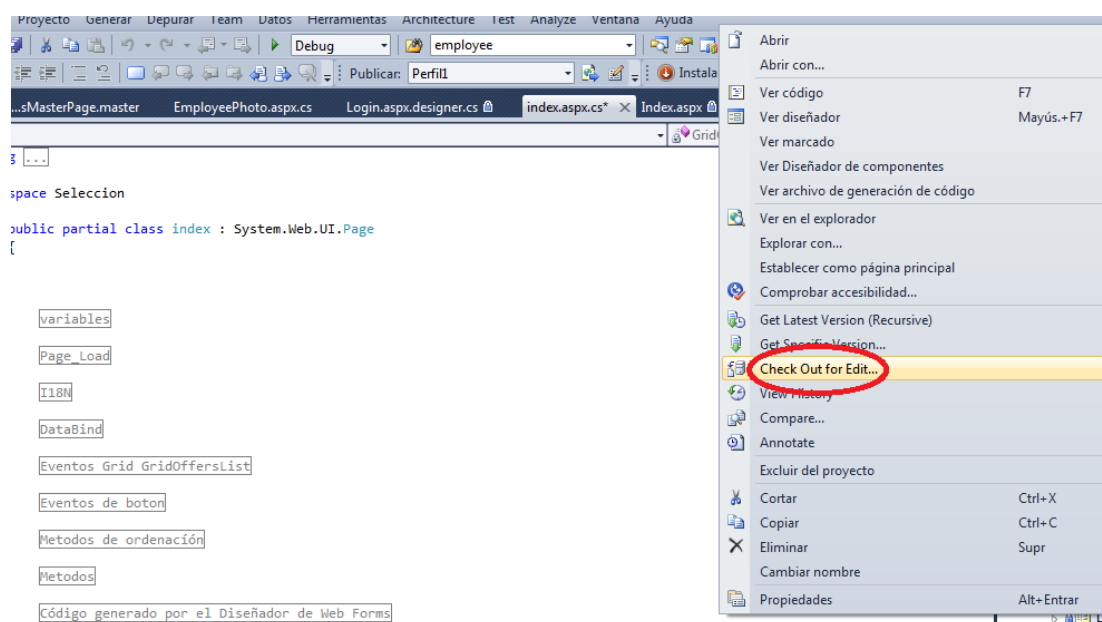


Figura 1: Bloqueo de ECS

- 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.



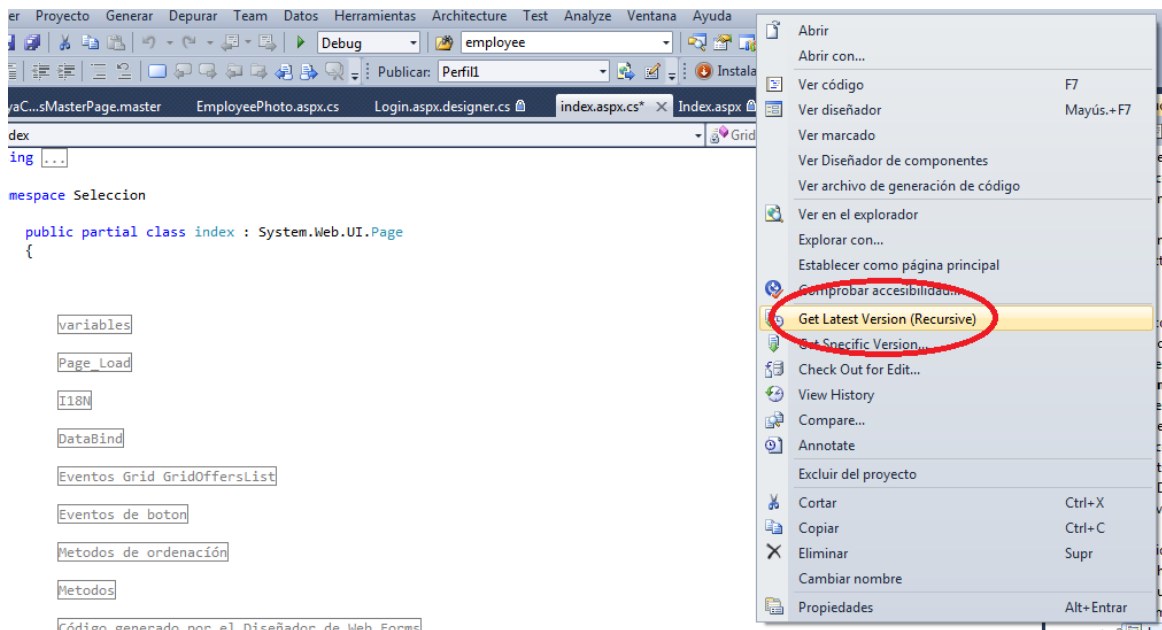


Figura 2: Descarga de versión

- 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.

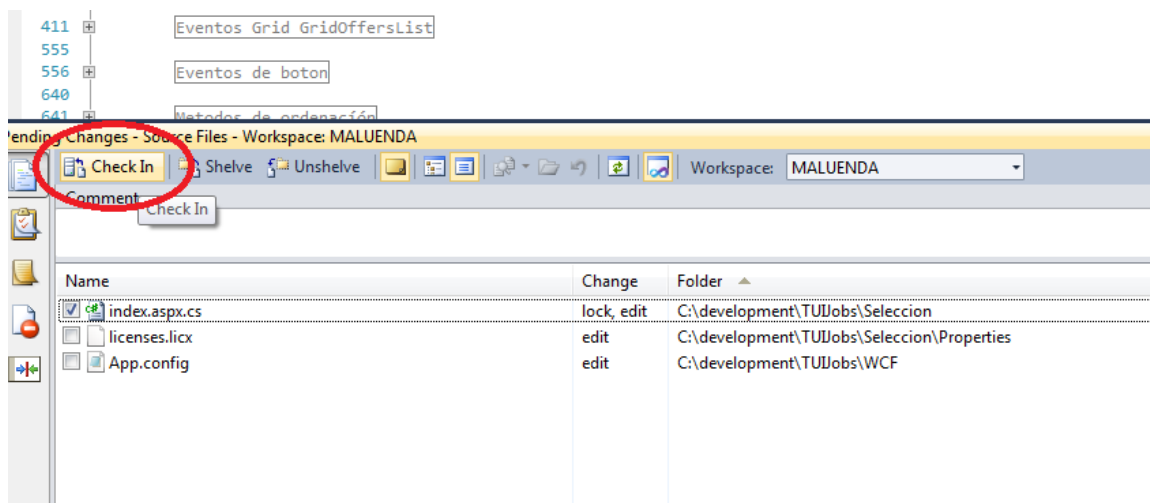


Figura 3: Modificación de ECS

- 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.



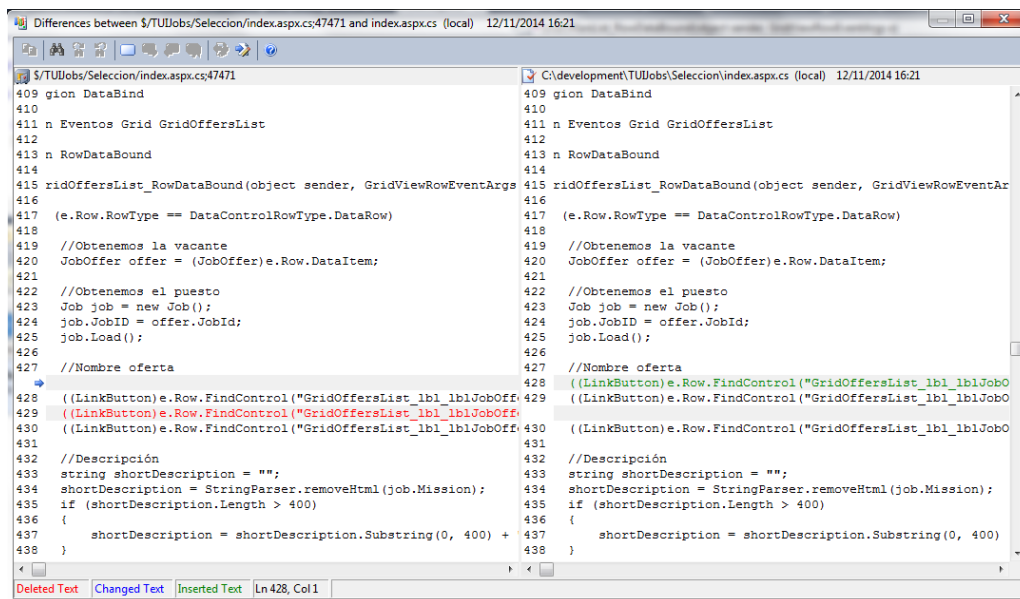


Figura 4: Resolución de conflictos

- 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.

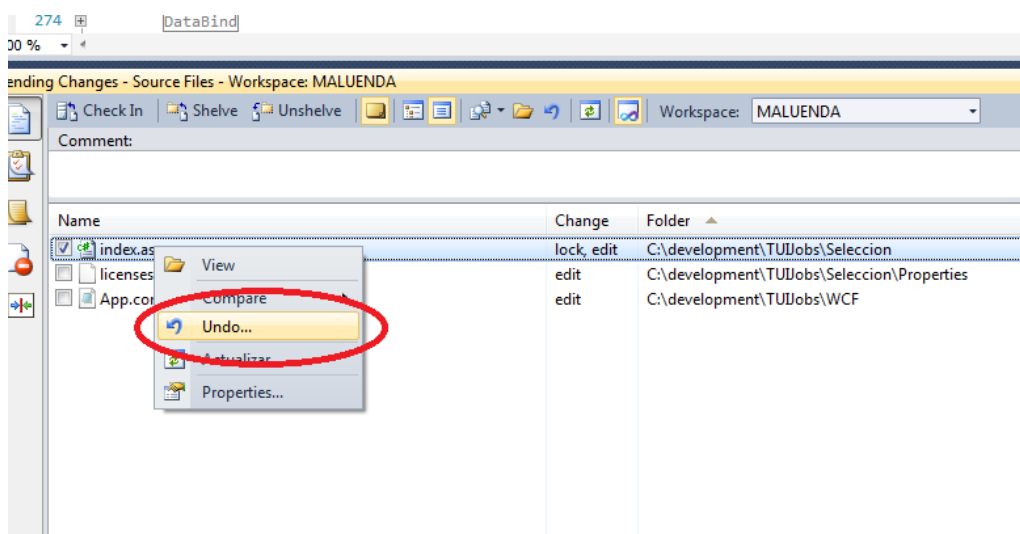


Figura 5: Descartar modificaciones

## 5. BIBLIOGRAFÍA

### 5.1 Referencias

- [R1] I. Jacobson, G. Booch, J. Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. Pearson Education

### 5.2 Referencias Web

- [W1] <http://www.wikipedia.org>  
[W2] <http://www.ieee.org>  
[W3] <http://www.histaintl.com/soluciones/configuracion/configuracion.php>



# ESTUDIO DE MERCADO

SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 1.3  
PUBLICADO EL 12/02/2015

Copyright © 2014 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
24/07/2014	1.0	Redacción inicial del documento	Sergio Sanz Frías
12/11/2014	1.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	1.2	Revisión del documento	Fernando Cortés Franco
12/02/2015	1.3	Correcciones en el documento	Sergio Sanz Frías





# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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. Bolsa de Empleo Web .....	7
3. Bolsas de Empleo Web .....	8
3.1 Introducción.....	8
3.2 Bolsas de empleo generalistas .....	8
3.2.1 <i>Infojobs.net</i> .....	8
3.2.2 <i>Trabajos.com</i> .....	13
3.3 Bolsas de empleo específicas.....	15
3.3.1 <i>Banco Santander</i> .....	15
3.4 Redes sociales.....	18
3.4.1 <i>LinkedIn</i> .....	18
3.4.2 <i>Facebook y Twitter</i> .....	18
4. Conclusiones .....	20
5. Bibliografía.....	21
5.1 Referencias web.....	21



# 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 trabajo, para así comprender mejor las necesidades y características del producto a desarrollar.

Por otra parte, se analizarán algunas bolsas de empleo web de ámbito nacional.

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 de la AWSC. 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

- AWSC: Aplicación Web de Selección de Candidatos.
- CV: Currículo Vitae.
- GUI: Graphical User Interface (Interfaz Gráfico de Usuario).
- RRHH: Recursos humanos.

## 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.

## 1.5 Referencias

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 una AWSC. Se compone de cinco apartados:



1. Propósito del documento y avance del mismo. Se proporciona una lista de acrónimos y definiciones útiles para la comprensión del documento, así como una lista de documentos del trabajo referenciados y el presente resumen.
2. Visión global del concepto de bolsa de empleo web, destacando sus principales funciones y elementos, y explicando la situación actual de la misma.
3. Se muestran algunas bolsas de empleo web existentes en el mercado, tanto generales como de organizaciones en particular o redes sociales.
4. Conclusiones obtenidas durante esta fase del trabajo.
5. Bibliografía y referencias Web utilizadas.



## 2. BOLSA DE EMPLEO WEB

Como se ha indicado en el apartado 1.1 (“Propósito del documento”) del presente texto, el punto de partida para posteriores fases del trabajo (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 selección de candidatos al desarrollo real de la aplicación que nos ocupa.

Una bolsa de empleo web es un sitio web destino a la oferta y búsqueda de empleo. En estos sitios, las organizaciones con puestos de trabajo vacantes publican ofertas de empleo para que las personas interesadas puedan ponerse en contacto con ellas.

La mayoría de las bolsas de empleo, también permiten la introducción del CV de los candidatos, para que las empresas puedan buscar en el sistema los candidatos que se ajusten a sus necesidades.

Cada vez más empresas facilitan en sus páginas web corporativas una sección de selección de candidatos, lo que ha causado una disminución de los sistemas tradicionales de reclutamiento. Las razones son muy variadas:

- La inmediata y permanente publicación de las ofertas, disponibles 24 horas al día los 7 días de la semana y accesibles desde prácticamente todas las partes del mundo; con el ahorro de tiempo y de recursos económicos que esto supone.
- Automatización de la preselección. Los RRHH se administran electrónicamente dentro de un sistema de gestión integral, consiguiendo una mayor efectividad y rentabilidad.
- Facilidad al candidato para el envío de su CV por correo electrónico o través de formularios, para poder hacer su propia gestión y actualización de sus datos, realizando además un seguimiento a las ofertas que más se acerquen a sus intereses.
- El candidato puede ser sometido a diferentes test online.

Por el contrario, también hay inconvenientes:

- Pueden producirse un gran número de solicitudes que no reúnan los requisitos requeridos.
- Algunos sectores de la población pueden quedar excluidos de los procesos de selección debido a su escasa o nula formación en el uso de las nuevas tecnologías.
- Las organizaciones no tienen la seguridad de que los CV estén actualizados o los datos sean verdaderos, ya que aún muchos usuarios de internet temen posibles engaños o desconfían del uso de sus datos.



## 3. BOLSAS DE EMPLEO WEB

### 3.1 Introducción

A continuación se muestran algunos ejemplos de bolsas de empleo que hay actualmente en funcionamiento, especificando sus características principales, y diferencias entre ellas; diferenciando las bolsas de empleo generalistas y las bolsas de empleo específicas de algunas organizaciones.

También se comenta el uso de las redes sociales en la búsqueda de trabajo.

### 3.2 Bolsas de empleo generalistas

#### 3.2.1 Infojobs.net

*Infojobs.net* es una de las principales bolsas de empleo web de ámbito nacional.

Es una bolsa de empleo privada online, especializada en el mercado español, creada en 1998 dentro del Grupo Intercom y tiene su sede central en Barcelona.

Durante el pasado mes de octubre, registró 156.406 puestos de trabajo, casi un 50% más que en el mismo mes de 2013.

Su página principal tiene una zona lateral fija en la izquierda, donde invita a los usuarios a registrarse (Figura 1); mientras que se diferencian tres zonas en el resto de la página (Figuras 2, 3 y 4), siendo necesario utilizar la barra de *scroll* vertical para visualizarlo todo.

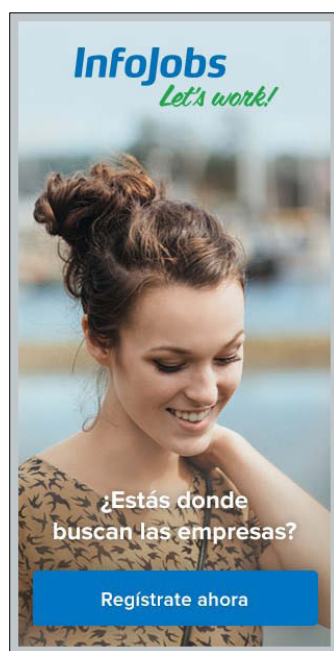


Figura 1. Zona lateral en la página principal de Infojobs



Figura 2. Primera sección en la página principal de Infojobs



Figura 3. Segunda sección en la página principal de Infojobs

## ¿Cómo me puede ayudar InfoJobs?

**Ofertas para ti cada día**  
Dinos el trabajo que buscas y te enviamos las mejores ofertas.

**Deja que te encuentren**  
Regístrate, sube tu CV y hazte visible a miles de empresas.

**Tu nuevo trabajo, aquí**  
7 de cada 10 ofertas de empleo en Internet están en InfoJobs.

[¡Regístrate gratis!](#)

156.5K likes

169.2K seguidores

3.8K seguidores

Figura 4. Tercera sección en la página principal de Infojobs

El sistema de registro es sencillo, tras rellenar algunos datos solicitados en un formulario (Figura 5), se envía al candidato un correo electrónico para confirmar su registro; redireccionándole a un nuevo formulario donde poder introducir una mayor diversidad de datos (Figura 6) divididos en secciones (Figura 7).

Estás en: [InfoJobs Empleo](#) + Alta gratis

1 Crea tu cuenta 2 Crea tu CV ¡Más fácil! Importa tu perfil de LinkedIn

### Crea tu cuenta de InfoJobs

Emplea dándonos quién eres y escoge los datos de acceso a tu cuenta. ¡Ya podrás recibir ofertas por email!

¿Ya tienes una cuenta? [Accede con tu e-mail y contraseña](#)

**Datos de acceso**

Nombre:

Primer Apellido:

Fecha de nacimiento:  /  /   
dd mm aaaa

Género: ☒ Hombre ☐ Mujer

E-mail:   
Comprueba que es correcto, porque recibirás un e-mail para verificarlo.

Crea tu contraseña:

Repite la contraseña:

Teléfono móvil:   
(opcional)

¿Cómo has llegado a InfoJobs?

Figura 5. Página de registro de Infojobs

**Completa tus datos personales**

Tipo de documento: NIF Número:  Ej: 01234567L

País de residencia:

Provincia:

Población:

Código Postal:

Teléfono de contacto preferente: ☒ Móvil ☐ Fijo ☐ Teléfono extranjero

Teléfono:

Permiso de conducir: 

- A1
- A2
- AM
- A

Autónomo: ☐ Sí ☒ No

Nacionalidad: 

- Croacia
- Dinamarca
- Eslovaquia
- Eslovenia
- España

**Tu presencia profesional en internet**

En Twitter:  Ej: http://twitter.com/mi\_usuario

En Facebook:  Ej: http://facebook.com/mi\_usuario

Tus datos personales son necesarios para que las empresas puedan contactar contigo.

Figura 6. Página de inserción de datos personales de Infojobs

✓ ESTUDIOS

✓ EXPERIENCIAS

✓ FUTURO EMPLEO

Figura 7. Secciones de CV de Infojobs

Tras rellenar todos estos datos se accede a la página del candidato, que posee un menú superior con varias opciones (Figura 8).

**InfoJobs** Empleo Freelance Executive Formación

Sergio Salir

Mi InfoJobs Mis CVs Candidaturas Perfil público Alertas Mis servicios Idiomas y tests Ajustes

Figura 8. Menu superior de la página del candidato de Infojobs





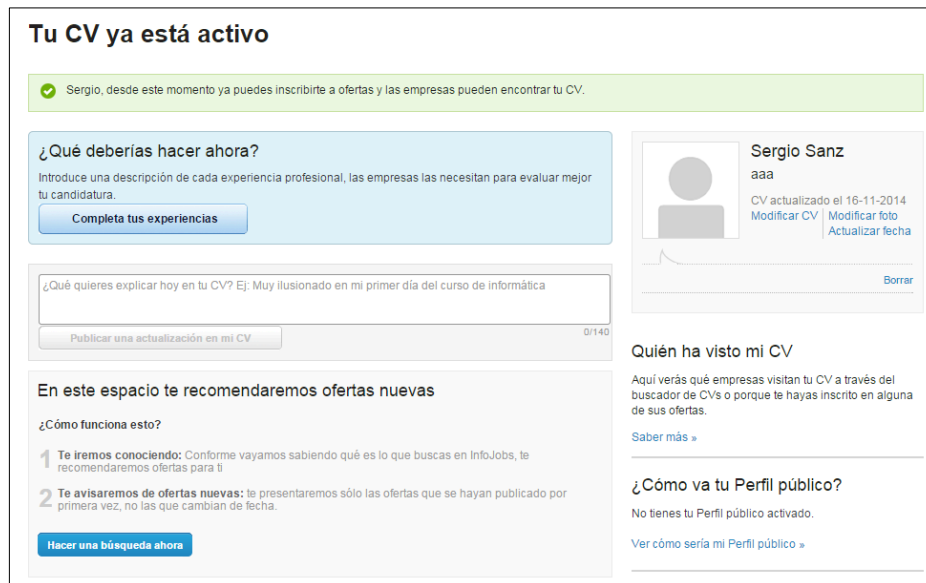


Figura 9. Página de candidato de Infojobs

Para finalizar el estudio de la página, se muestra la página de búsqueda y visualización de ofertas (Figura 10), donde aparece un menú lateral en la izquierda para realizar filtros en la búsqueda, y un menu central con la lista de ofertas con una pequeña descripción. Esta página vuelve a tener una gran altura haciendo imprescindible el uso del *scroll* vertical para visualizarla por completo.

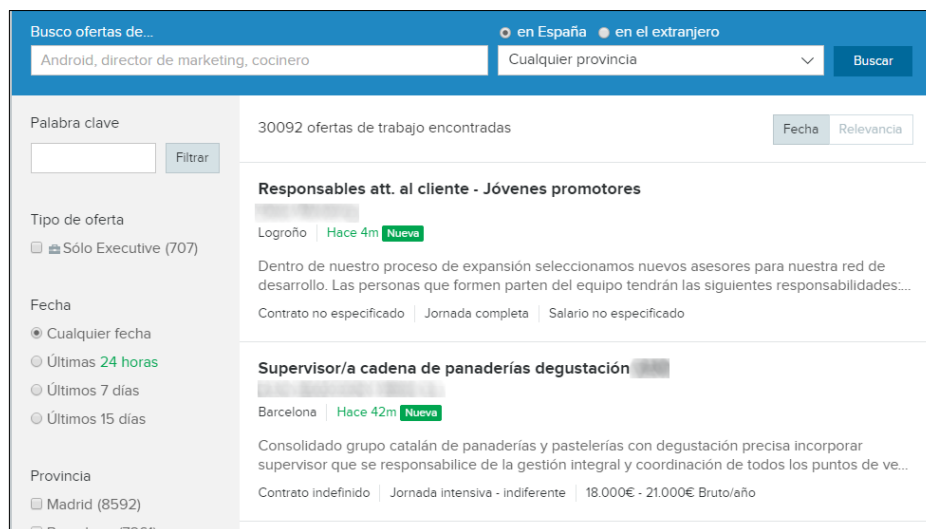


Figura 10. Página de búsqueda de ofertas de Infojobs

### 3.2.2 Trabajos.com

Trabajos.com es propiedad de Hispanista S.L. y se creó en 1996, siendo el portal pionero de la red en España. Actualmente, están registrados más de 8 millones de usuarios y de cien mil empresas.

En la página de inicio se pueden visualizar algunas ofertas de empleo (Figura 11).

The screenshot displays the 'Ofertas de trabajo' (Job Offers) section of the Trabajos.com website. It features a list of four job postings, each with a title, location, salary, and a brief description. The listings are as follows:

- Seleccionamos asesor@s nutrición deportiva** (16/11/2014): No requiere experiencia. Mayores de 25 años. Personas interesadas en mejorar su situación actual. Location: Toda España. Salary: Salario a convenir. Jornada: Indiferente.
- Captador-a Socios Fundación Lav 600e/m Mañanas** (16/11/2014): Se valora la experiencia en puestos de promotor/a-captador/a-comercial principalmente en calle. Disponibilidad inmediata, permiso de trabajo y residencia en la provincia. Location: Madrid, Madrid. Salary: 600 € - 800 €. Jornada: Indiferente.
- Cocinero En Formación Dual Con Stage En Italia.** (16/11/2014): Se precisa: - Personas interesadas y motivadas por la cocina. - Disponibilidad para participar de un proyecto de formación DUAL con inicio. Location: Madrid, Madrid. Salary: 12.000 €. Jornada: Indiferente.
- Limpiador/a En Cornellà De Llobregat** (16/11/2014): // Imprescindible experiencia como limpiador/a. // Flexibilidad para trabajar domingos y festivos. // Capacidad de trabajo en equipo. // Flexibilidad. Imprescindible. Location: Cornellà De Llobregat, Barcelona. Salary: Salario a convenir. Jornada: Indiferente.

Figura 11. Página de inicio de Trabajos.com

La página de registro (Figura 12), simplemente solicita al usuario los datos imprescindibles para su registro; y tras introducirlos se envía un email al usuario para que lo confirme, redireccionándolo al hacerlo a la página del candidato (Figura 13).

The screenshot shows the registration page of Trabajos.com. The header includes the logo 'trabajos.com' with the tagline 'Crecemos juntos', and navigation links for 'OFERTAS DE EMPLEO', 'CURSOS Y FORMACIÓN', 'ACCESO EMPRESAS', 'REGÍSTRATE' (with a LinkedIn icon), and 'ENTRAR' (with a user icon). The main content area is titled 'Regístrate en Trabajos.com' and contains a registration form with the following fields:

- ¿Eres nuevo?** Elige tus claves, introduce tu email y estarás registrado.
- \* Usuario:** A text input field.
- \* Clave:** A text input field.
- \* Repetir clave:** A text input field.
- \* Email:** A text input field.
- Terminar:** A blue button.

Below the form, there is a disclaimer: 'Al pulsar el botón "Terminar" estás aceptando expresamente las condiciones de uso y política de privacidad.' To the right of the form, there are three promotional banners:

- IMPORTA TU CV DE LINKEDIN** y ahorra tiempo.
- REGÍSTRATE GRATIS** en un sólo minuto.
- AÑADE ARCHIVOS ADJUNTOS** en tus candidaturas.
- DESCÁRGATE NUESTRA APP** y no dejes escapar ninguna oferta.

At the bottom of the page, there is a link: 'Ya estoy registrado en Trabajos.com'.

Figura 12. Página de registro en Trabajos.com

Figura 13. Página de candidato de Trabajos.com

Para completar los datos personales se accede a la sección de 'Mis datos' (Figura 14), y para crear los currículos hay varias opciones como muestra la figura 15.

Figura 14. Página de datos personales en Trabajos.com

CV Mis Currículos (5 máximo)

Importar de LinkedIn Crear currículum

Crea tu currículum (5 máximo)

Importa tu Currículo de LinkedIn

Importa los datos de tu CV desde LinkedIn. ¡Rápido y cómodo!

Importar currículum

Crear un currículum nuevo

Crea un currículum **desde cero** adaptándolo a los tipos de ofertas en las que quieras inscribirte.

Crear currículum

Cartas de presentación (5 máximo)

Crea una carta y preséntate a ti mismo para **mejorar tu candidatura** a los trabajos.

Crea tu carta

Archivos adjuntos (5 máximo)

¿Tienes títulos, acreditaciones, tu propio Currículo guardado en tu PC?

¡Añádelo!

Figura 15. Página de creación de currículos de Trabajos.com

También se puede acceder a la lista de ofertas, con una disposición similar a la vista en la pantalla de inicio.

### 3.3 Bolsas de empleo específicas

#### 3.3.1 Banco Santander

Banco Santander es un gran grupo financiero internacional tanto por capitalización como por beneficios. En su web corporativa facilita varias formas para que el candidato entre en su proceso de reclutamiento (Figura 16).



Figura 16. Página de inicio de Banco Santander

Este documento se centra en la sección de 'Si buscas Empleo'. En la que se puede optar por visualizar las ofertas de empleo diferenciadas en varias secciones (Figura 17), o introducir el currículum (Figura 18).





Figura 17. Página de ofertas de Banco Santander



**Darse de alta**

Todos los campos de esta pantalla son obligatorios. Recuerda que el usuario y la contraseña no pueden ser iguales. El usuario debe tener un mínimo de 5 y un máximo de 15 caracteres. La contraseña debe tener un mínimo de 4 y un máximo de 15 caracteres. El campo DNI no admite cero inicial, si tu DNI comienza por 0, omítele. La letra debe ser mayúscula y sin espacios ni guiones.

<b>Nombre de usuario *</b> <input type="text"/> <b>Confirmar contraseña *</b> <input type="text"/> <b>Nombre *</b> <input type="text"/> <b>Email *</b> <input type="text"/> <b>Introduzca código de verificación *</b> <input type="text"/>	<b>Contraseña *</b> <input type="password"/> <b>Tipo Documento *</b> NIF <input type="text"/>  <b>Apellidos *</b> <input type="text"/> <b>Confirmar email *</b> <input type="text"/> <b>Código de verificación</b> 372333 
--	---

☐ He leído y acepto la [política de privacidad](#) de Grupo Santander

[< Volver](#) [ALTA](#)

Figura 18. Página de registro de Banco Santander

Tras completar la página de registro aparece otro formulario donde poder completar más los datos del currículum del candidato con varias pestañas en un menú (Figura 19). En el lateral de la pantalla aparece un enlace a las ofertas de empleo (Figura 20).

 **Santander**


[DATOS PERSONALES](#)
[FORMACIÓN](#)
[IDIOMAS](#)
[INFORMÁTICA](#)
[EXPERIENCIA](#)
[OTROS DATOS](#)
[VER CV](#)
[MIS OFERTAS](#)

Figura 19. Menú del candidato de Banco Santander



Figura 20. Acceso a ofertas de empleo de Banco Santander

### 3.4 Redes sociales

Cada vez son más las empresas que acuden a las redes sociales para cubrir un puesto de su trabajo, para hacer llegar las ofertas a los candidatos, o para buscar información sobre los mismos antes de realizar su contratación.

Se utilizan tanto redes sociales preparadas para este fin, como *LinkedIn*, u otras como *Facebook* o *Twitter* por su amplio número de usuarios.

#### 3.4.1 *LinkedIn*

*LinkedIn* es una red social orientada a negocios fundada en diciembre de 2002.

Es utilizada por personas que buscan trabajo, personas que ya tienen trabajo, vendedores que buscan compradores o vendedores que buscan nuevos clientes.

#### 3.4.2 *Facebook y Twitter*

Ambas redes sociales, pese a no estar destinadas inicialmente al ámbito empresarial, son una buena forma de conocer a los candidatos y ofrecer ofertas de empleo. El gran número es la principal razón del uso de ambas redes sociales para el fin que nos atañe.

Hay varias formas para optar a un puesto de trabajo a través de *Facebook*. El primero es ir a su mercado de ofertas, con listas de trabajo u otras oportunidades en tu red; además de las ventanas de anuncios.



Otra manera de conseguir trabajo es usarlo para unirse a grupos y encontrar personas con intereses comunes estableciendo contacto con ellos.

*Twitter* rompe las barreras de comunicación y permite al candidato hablar directamente con los responsables de contratación sin tener que presentar un CV de inmediato a una máquina. Puede servir como una primera toma de contacto, y completarse con un blog o perfil en *LinkedIn* añadidos en el perfil.





## 4. CONCLUSIONES

El presente estudio de mercado nos ha permitido comprender mucho mejor el problema al que nos enfrentamos. Su objetivo era analizar el funcionamiento y prestaciones que ofrecen diferentes bolsas de empleo de cara a los candidatos.

Tras analizar cada una de estas bolsas de empleo y otras formas alternativas de búsqueda de trabajo en Internet, se ha llegado a la conclusión de que deben cuidarse especialmente el aspecto de web, para que resulte atractiva, y en mayor medida su usabilidad. Debe considerarse que los usuarios que accederán al sistema van a ser de tipos muy variados; debiendo evitar que personas que no están familiarizadas con el uso de aplicaciones web tengan problemas para interactuar con el sistema.

En la mayoría de los casos, mucha información se introduce en campos de texto, lo que imposibilita o dificulta en gran medida la búsqueda de candidatos con ciertas características que han sido introducidas de esta forma. Por lo tanto, en el desarrollo de nuestra aplicación, se parametrizarán al máximo todos los datos posibles.



## 5. BIBLIOGRAFÍA

### 5.1 Referencias web

- [W1] <http://www.wikipedia.org>
- [W2] <http://www.educaweb.com/contenidos/laborales/bolsas-trabajo-on-line>
- [W3] <https://www.infojobs.net/>
- [W4] <http://www.trabajos.com/>
- [W5] <https://www.bancosantander.es>
- [W6] <https://www.linkedin.com>
- [W7] <https://twitter.com/>
- [W8] <https://es-es.facebook.com/>
- [W9] <http://empleoytrabajo.org/como-usar-las-redes-sociales-para-encontrar-trabajo/>
- [W10] <http://trabajo.practicopedia.lainformacion.com/busqueda-de-trabajo/como-buscar-ofertas-de-empleo-en-facebook-11440>



# ESPECIFICACIÓN DE REQUISITOS

SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 1.3  
PUBLICADO EL 11/02/2015

Copyright © 2014 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
22/07/2014	1.0	Redacción inicial del documento	Sergio Sanz Frías
08/11/2014	1.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	1.2	Revisión del documento	Fernando Cortés Franco
11/02/2015	1.3	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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 general.....	7
2.1 Funciones generales del sistema .....	7
2.2 Características de los usuarios .....	7
3. Requisitos funcionales del sistema.....	8
4. Requisitos no funcionales del sistema.....	11
5. Bibliografía.....	14
5.1 Referencias.....	14
5.2 Referencias web.....	14



# 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 [R1]:

- Correcta.
- Inequivoca.
- Completa.
- Consistente.
- Comprobable.
- Modificable.
- Identificable.

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

## 1.2 Alcance del documento

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

## 1.3 Acrónimos

- AWSC: Aplicación Web de Selección de Candidatos.
- CV: Currículo Vitae.
- GUI: Graphical User Interface (Interfaz Gráfico de Usuario).
- PDF: Portable Document Format (Formato de Documento Portátil).
- 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.



## 1.5 Referencias

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

## 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 cuatro apartados:

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 trabajo referenciados y el presente resumen.
2. Descripción general de las funciones del producto y de la naturaleza de sus usuarios.
3. Relación de los requisitos funcionales del sistema a desarrollar, su descripción y su nivel de criticidad.
4. Descripción de los requisitos no funcionales del sistema.
5. Bibliografía utilizada a lo largo de esta fase del trabajo.





## 2. DESCRIPCIÓN GENERAL

### 2.1 Funciones generales del sistema

La bolsa de empleo web es un sistema que permite poner en contacto a empresas que buscan trabajadores, y a personas que buscan un empleo. Para ello permite la inserción de CV y la búsqueda de ofertas.

En nuestro caso es una organización en particular la interesada en publicar ofertas y buscar trabajadores.

### 2.2 Características de los usuarios

Los usuarios que interactuarán con la aplicación son personas en busca de un empleo en la organización. Podrán introducir su CV en el sistema, conocer las vacantes disponibles en la empresa e inscribirse en las ofertas que más les satisfagan.

La empresa podrá publicar las vacantes y filtrar a los candidatos mediante la información que hayan introducido utilizando el sistema software de gestión de recursos humanos desarrollado en Endalia.

Al desconocer el tipo de ofertas que la empresa publicará y por tanto, el público objetivo de la oferta, habrá que tener en cuenta que los usuarios no tendrán por qué estar familiarizados con el uso de aplicaciones web.



### 3. REQUISITOS FUNCIONALES DEL SISTEMA

Nombre	RF - 01: Alta de nuevos usuarios
Descripción	Los candidatos pueden registrarse de forma gratuita en el sistema. El acceso a la sección de alta de candidatos debe ser fácilmente localizable, y el registro lo más breve posible.
Nivel de criticidad	Alto
Dependencias	-

Nombre	RF – 02: Acceso de usuarios existentes
Descripción	Habrà un punto de acceso reconocible que llevará a la sección privada del candidato.
Nivel de criticidad	Alto
Dependencias	Alta de nuevos usuarios

Nombre	RF – 03: Inserción de CV
Descripción	<p>Sección accesible para el candidato desde el panel principal una vez que ha accedido al sistema con su usuario, y desde los menús. Debe contener las siguientes secciones:</p> <ul style="list-style-type: none"><li>• Datos personales</li><li>• Fotografía</li><li>• Formación</li><li>• Experiencia</li><li>• Expectativas</li><li>• Idioma</li><li>• Referencias</li><li>• Ficheros</li><li>• Observaciones</li></ul> <p>Parametrizar al máximo la entrada de datos para realizar búsquedas más precisas de CV.</p>
Nivel de criticidad	Alto
Dependencias	Acceso de usuarios existentes



Nombre	RF – 04: Visualización de ofertas
Descripción	Sección accesible tanto para usuarios registrados como para no registrados. Se ofrecerá una lista de los puestos vacantes de la organización junto a su descripción.
Nivel de criticidad	Alto
Dependencias	-

Nombre	RF – 05: Visualización de detalles de una oferta
Descripción	<p>Sección accesible tanto para usuarios registrados como para no registrados a partir de la visualización de todas las ofertas. La información que aparecerá dependerá de los datos que la organización haya introducido al crear la vacante. Las opciones son:</p> <ul style="list-style-type: none"> <li>• Descripción de la oferta</li> <li>• Categoría</li> <li>• Número de vacantes en el puesto</li> <li>• Requerimientos de la oferta</li> <li>• Experiencia y/o cualificación mínima</li> <li>• Fecha de publicación</li> <li>• Fecha límite de recepción de CV</li> </ul>
Nivel de criticidad	Alto
Dependencias	Visualización de ofertas

Nombre	RF – 06: Generación de un informe de detalles de una oferta
Descripción	El candidato podrá descargar un informe en PDF con la misma información disponible en los detalles de la oferta. Esta opción estará disponible tanto para usuarios registrados como no registrados.
Nivel de criticidad	Bajo
Dependencias	Visualización de detalles de una oferta



Nombre	RF – 07: Inscripción en una oferta
Descripción	Los candidatos registrados podrán inscribirse en una oferta tras acceder al sistema con su usuario
Nivel de criticidad	Alto
Dependencias	Visualización de detalles de una oferta

Nombre	RF – 08: Darse de baja de una oferta
Descripción	Los candidatos registrados podrán, al igual que inscribirse, darse de baja de una oferta en la que ya no deseen estar inscritos.
Nivel de criticidad	Alto
Dependencias	Inscripción en una oferta

Nombre	RF – 09: Eliminación de cuenta de usuario
Descripción	Los candidatos registrados podrán, en cualquier momento, eliminar su cuenta de usuario del sistema.
Nivel de criticidad	Alto
Dependencias	Alta de nuevos usuarios



## 4. REQUISITOS NO FUNCIONALES DEL SISTEMA

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 la AWSC será a través de GUI.
Nivel de criticidad	Medio
Dependencias	-

Nombre	RNF - 03: Interfaz de usuario consistente
Descripción	La AWSC debe mantener una GUI consistente con la interfaz preexistente en la organización, 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	Interacción con la aplicación a través de GUI

Nombre	RNF - 04: Aplicación escalable
Descripción	La AWSC será fácilmente escalable. Será posible añadir nuevos módulos, funcionalidades y elementos con facilidad.
Nivel de criticidad	Medio
Dependencias	-



Nombre	RNF - 05: Posibilidad de internacionalización
Descripción	La AWSC 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	La AWSC 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: Disponibilidad del sistema
Descripción	El sistema debe ser accesible las 24 horas del día y los 7 días de la semana; y permitir el acceso de múltiples usuarios de forma concurrente sin que esto conlleve una pérdida perceptible de rendimiento o inconsistencia en los datos.
Nivel de criticidad	Alto
Dependencias	-

Nombre	RNF – 08: Resolución
Descripción	La AWSC deberá ser visualizada correctamente en pantallas con resoluciones iguales o superiores a 1024x768.
Nivel de criticidad	Alto
Dependencias	-



Nombre	RNF – 09: Seguridad e integridad de los datos
Descripción	<p>La AWSC estará compuesta de secciones públicas y privadas. El acceso a las secciones privadas estará controlado por contraseña, que será almacenada de forma segura en el servidor.</p> <p>Además se debe garantizar la integridad de los datos y sus relaciones en las transacciones con la base de datos.</p>
Nivel de criticidad	Alto
Dependencias	-

Nombre	RNF – 10: Compatibilidad de navegadores
Descripción	La AWSC podrá interactuar mediante Internet Explorer 7.0 o superior, Mozilla Firefox y Google Chrome.
Nivel de criticidad	Medio
Dependencias	-

Nombre	RNF – 11: Integración en un sistema de gestión de recursos humanos
Descripción	La aplicación deberá integrarse con el software de gestión de recursos humanos desarrollado en Endalia; desde donde la organización publicará ofertas y consultará los CV de los candidatos.
Nivel de criticidad	Alto
Dependencias	-



## 5. BIBLIOGRAFÍA

### 5.1 Referencias

[R1] IEEE Recommended Practice for Software Requirements Specifications, Std-830-1998.

### 5.2 Referencias web

[W1] <http://www.wikipedia.org>

[W2] <http://www.endalia.com>





# ANÁLISIS

## SISTEMA DE RECLUTAMIENTO IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 1.3  
PUBLICADO EL 12/02/2015

Copyright © 2014 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
25/07/2014	1.0	Redacción inicial del documento	Sergio Sanz Frías
13/11/2014	1.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	1.2	Revisión del documento	Fernando Cortés Franco
12/02/2015	1.3	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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.....	9
3.3.1 Esquema general de los casos de uso .....	10
3.3.2 Candidatos .....	10
3.3.3 Empresa.....	18
4. Análisis de paquetes .....	21
4.1 Introducción.....	21
4.2 Identificación de paquetes de análisis .....	21
4.2.1 Usuarios.....	21
4.2.2 CV (Currículo Vitae).....	21
4.2.3 Ofertas .....	22
4.2.4 Gestión de incidencias .....	23
4.2.5 Acceso a repositorio de persistencia de datos.....	23
4.2.6 Internacionalización de entidades visibles.....	23
4.2.7 Servicios web .....	23
5. Requerimientos especiales .....	24
5.1 Persistencia .....	24
5.2 Tolerancia a fallos.....	24
5.3 Internacionalización.....	24
5.4 Seguridad .....	24
5.5 Disponibilidad del sistema.....	24
6. Bibliografía.....	25



6.1	Referencias.....	25
6.2	Referencias web.....	25



# 1. INTRODUCCIÓN

## 1.1 Propósito del documento

En el presente documento se muestra la fase de análisis del proyecto de desarrollo de la AWSC. A partir de los requisitos descritos en el documento de especificación de requisitos, 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

- AWSC: Aplicación Web de Selección de Candidatos.
- BD: Base de Datos.
- CV: Currículo Vitae.
- GUI: Graphic User Interface.
- RUP: Rational Unified Process.
- UML: Unified Modeling Language.

## 1.4 Definiciones

- Artefacto: cualquier tipo de información creada, producida, cambiada o utilizada por el equipo de desarrollo del proyecto.
- 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 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.
- 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.
- UML: lenguaje de modelado de sistemas de software desarrollado por *Rational* y de uso extendido.



## 1.5 Referencias

En este documento se referencian los siguientes documentos del proyecto:

- ESPREQ: documento en el que se especifican los requisitos del sistema.

## 1.6 Resumen

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

1. Introducción del documento, definición del propósito y alcance del mismo.
2. Descripción del proceso de análisis seguido para la confección de este documento.
3. Descripción de los casos de uso.
4. Identificación de paquetes de análisis.
5. Identificación de los requisitos especiales del sistema.
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 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, identificando 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. 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 son los usuarios que interaccionan con el sistema. Estos usuarios se identifican a partir del documento de especificación de requisitos (ESPREQ) y representan los diferentes roles de los usuarios en el sistema. El análisis conjunto de dichos elementos determina los siguientes actores de la AWSC:

- Candidatos:
  - Usuario que accede a la aplicación en busca de empleo.
  - Puede introducir su CV en el sistema y conocer las vacantes de ofertas e inscribirse en ellas.
- Empresa, a través de su área de recursos humanos:
  - Utiliza la AWSC indirectamente, a través del sistema software de gestión integral de recursos humanos desarrollado en Endalia.
  - Publica las ofertas de puestos vacantes en su organización.
  - Consulta la información introducida por los candidatos en su CV.

### 3.3 Casos de uso

En este apartado se especifican, mediante una descripción textual y diagramas explicativos comentados, los diferentes casos de uso de la AWSC. 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.

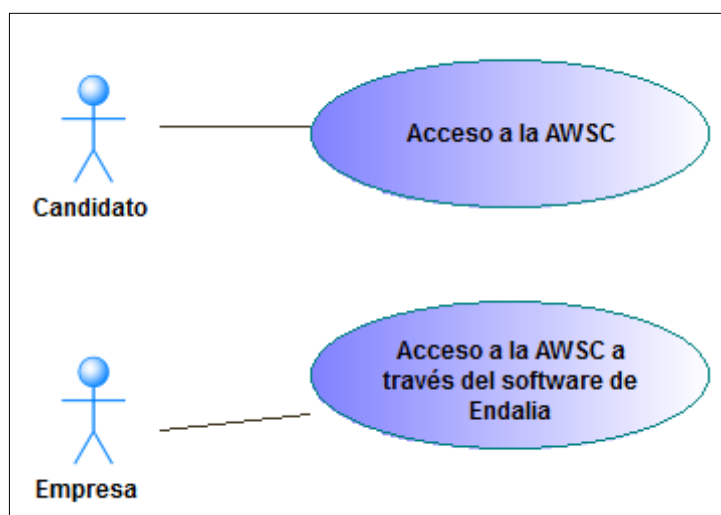


Figura 1: Esquema general de los casos de uso

### 3.3.2 Candidatos

En este apartado se describen los casos de uso referidos a los usuarios candidatos. En la siguiente figura (Figura 2) se observa el diagrama del caso de uso ‘Candidatos’ con un mayor nivel de detalle. Al tratarse de una página web siempre que se haga referencia a “Acceso al sistema”, se referirá a acceso a la parte privada del mismo, mediante la introducción de los datos de acceso (email y contraseña) del usuario.

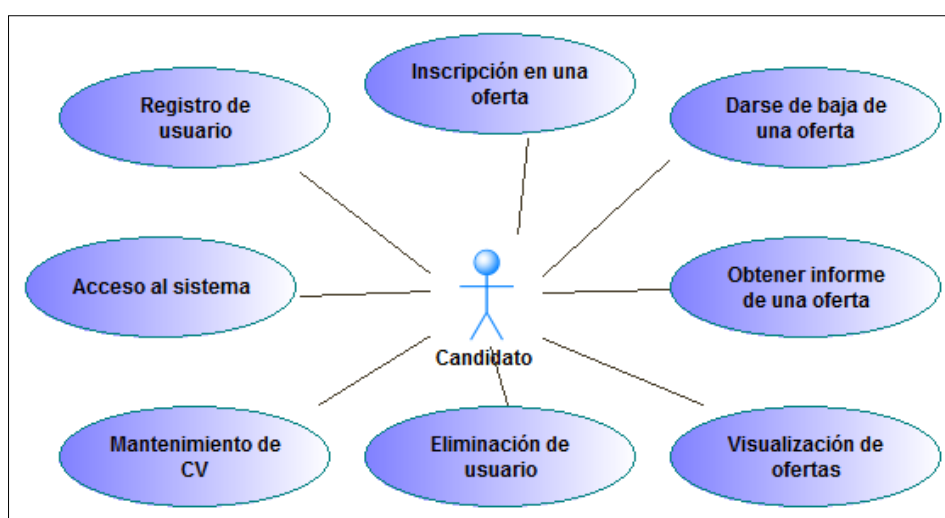


Figura 2: Casos de uso del candidato.

Como indica la figura 2, el candidato está relacionado con ocho casos de uso:

- Registro de usuario.
- Acceso al sistema.
- Mantenimiento de CV.
- Eliminación de usuario.
- Visualización de ofertas.
- Inscripción en una oferta.
- Darse de baja de una oferta.
- Obtención del informe de una oferta.

A continuación se analizará individualmente cada uno de estos casos de uso.

- Registro de usuario. El usuario interactúa con la GUI del sistema e introduce sus datos por primera vez, y en caso de ser válidos obtiene una cuenta privada para posteriores accesos al mismo, enviando el sistema automáticamente un correo electrónico al candidato facilitando un enlace para confirmar la nueva cuenta de usuario.

- Diagrama de secuencia (Figura 3):

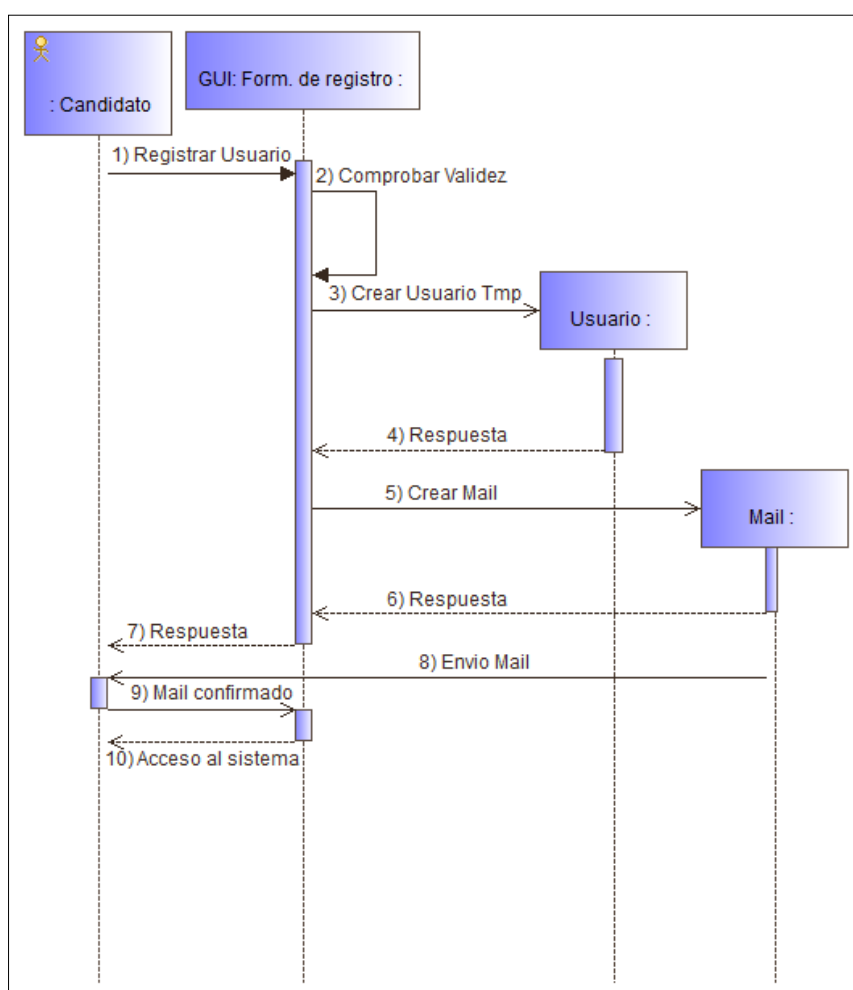


Figura 3: Diagrama de secuencia del caso de uso 'Registro de usuario'

- Diagrama de actividades (Figura 4):

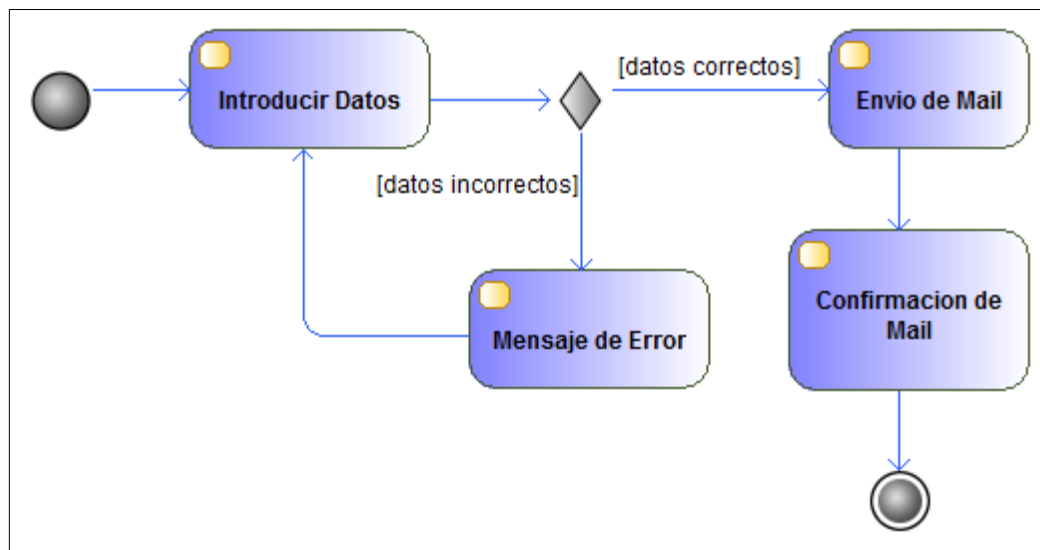


Figura 4: Diagrama de actividades del caso de uso 'Registro de usuario'

- Acceso al sistema. El usuario interactúa con la GUI para introducir su email y contraseña. Una vez comprobado que los datos son correctos, el usuario accede a la sección privada.

- Diagrama de secuencia (Figura 5):

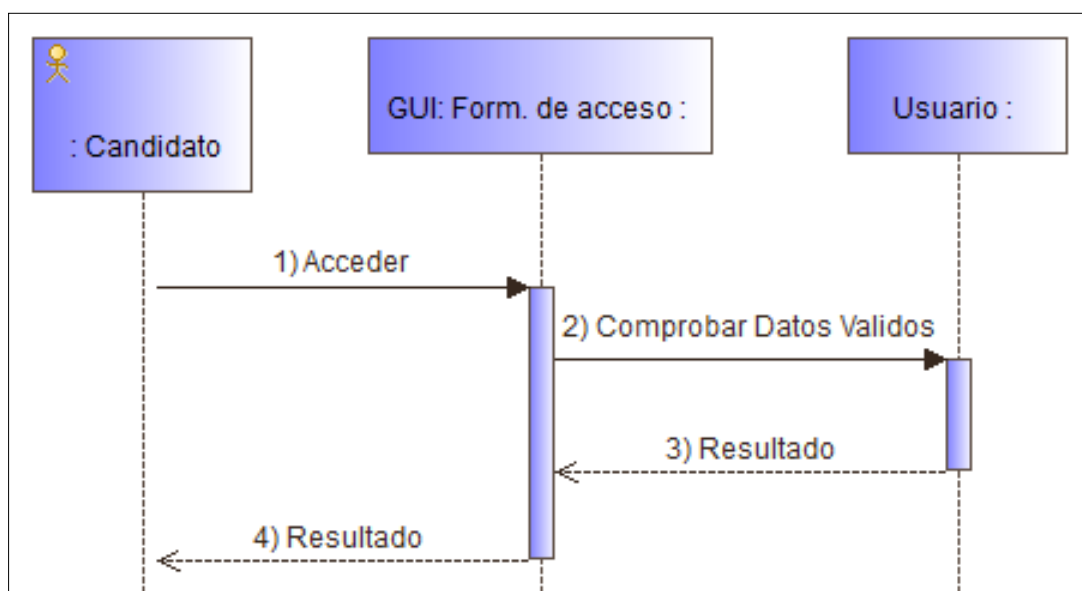


Figura 5: Diagrama de secuencia del caso de uso 'Acceso al sistema'

- Diagrama de actividades (Figura 6):

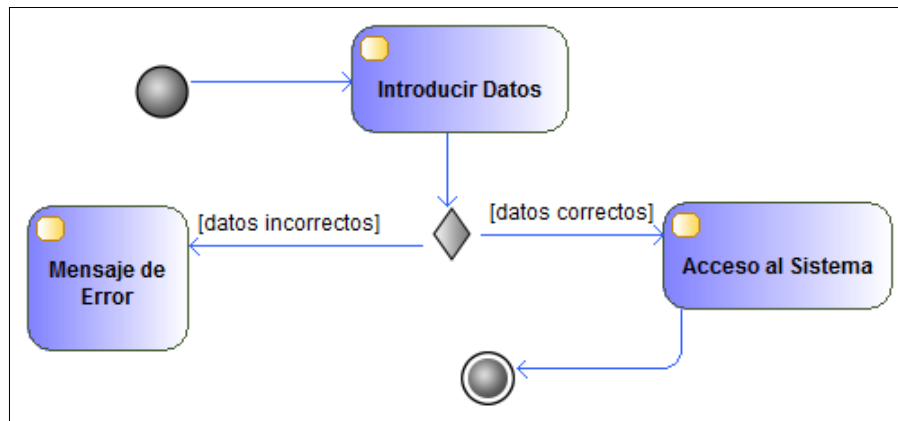


Figura 6: Diagrama de actividades del caso de uso 'Acceso al sistema'

- Mantenimiento de CV. Debido al tamaño que puede llegar a alcanzar el CV de un candidato, y aunque se trate de aspectos diferentes dentro de su CV, el proceso será prácticamente el mismo, a continuación se explica el proceso de modo genérico, sin especificar que parte del contenido del CV es la afectada. El mantenimiento de los datos del CV de un candidato, comprende tanto la consulta del mismo, como la inserción y modificación de datos.

- Diagrama de secuencia (Figura 7):

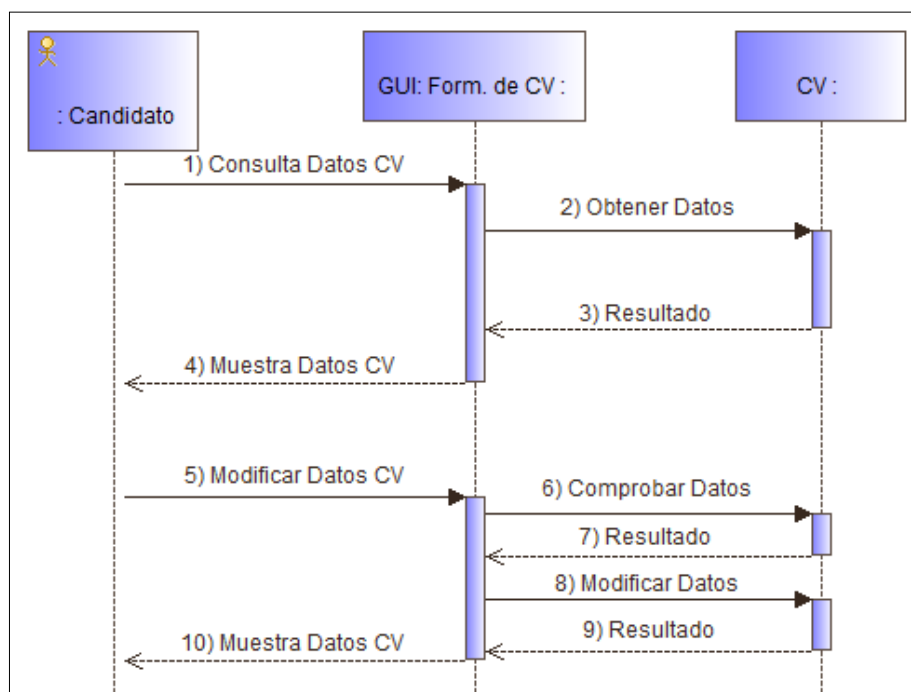


Figura 7: Diagrama de secuencia del caso de uso 'Mantenimiento de CV'

- Diagrama de actividades (Figura 8):

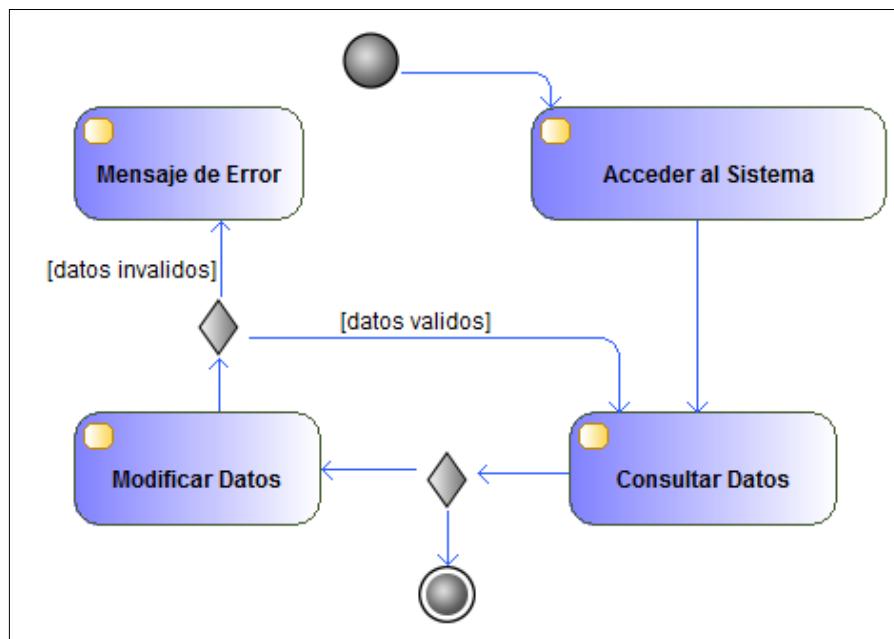


Figura 8: Diagrama de actividades del caso de uso 'Mantenimiento de CV'

- Eliminación de usuario. El usuario se da de baja en el sistema, solicitando la eliminación de su cuenta y sus datos personales. El sistema envía un correo confirmando la eliminación.

- Diagrama de secuencia (Figura 9):

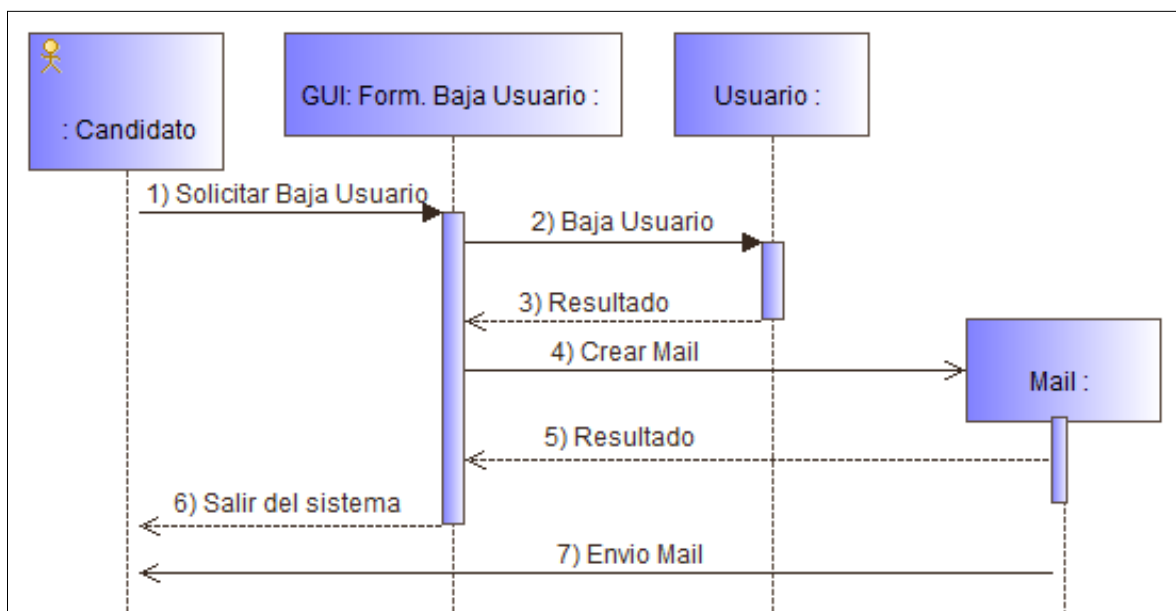


Figura 9: Diagrama de secuencia del caso de uso 'Eliminación de un usuario'

- Diagrama de actividades (Figura 10):

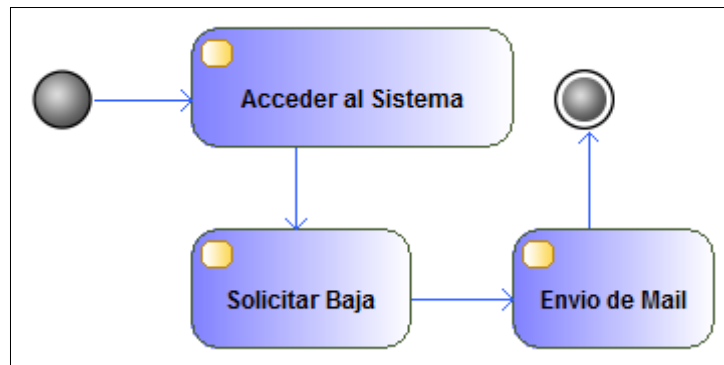


Figura 10: Diagrama de actividades del caso de uso 'Eliminación de un usuario'

- Visualización de ofertas. Tanto usuarios registrados como sin registrar pueden visualizar ofertas. Este caso de uso engloba tanto la visualización de todas las ofertas como del acceso a la información de una en particular.

- Diagrama de secuencia (Figura 11):

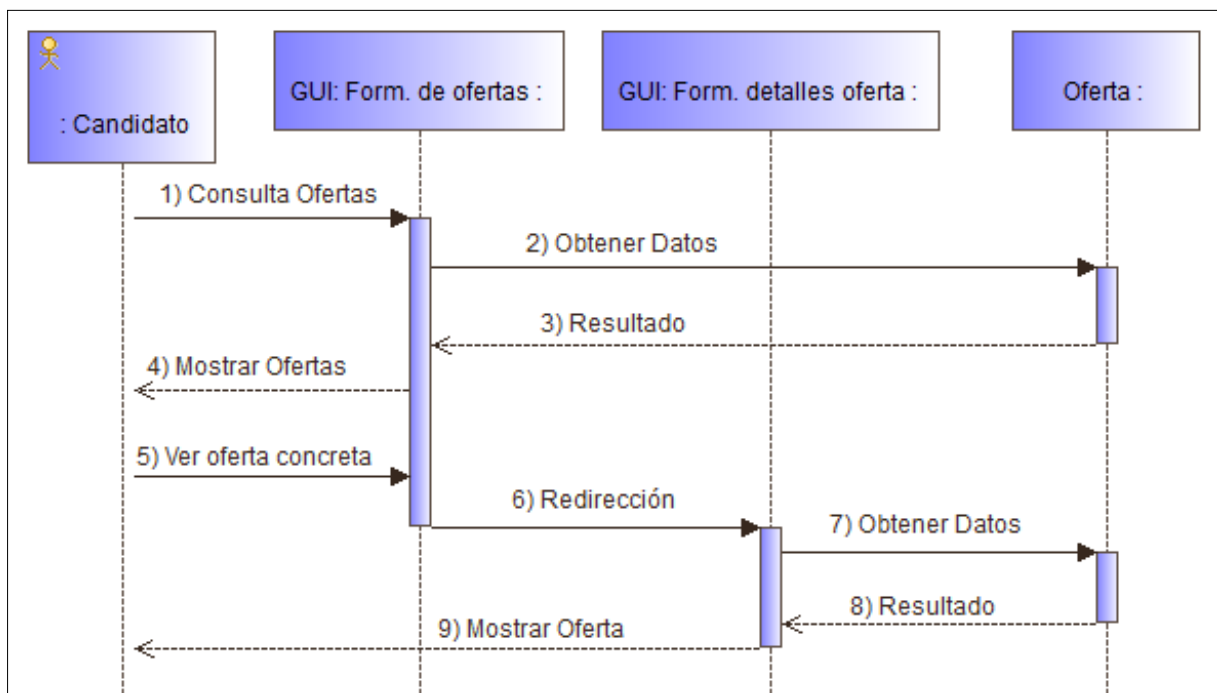


Figura 11: Diagrama de secuencia del caso de uso 'Visualización de ofertas'

- Diagrama de actividades (Figura 12):

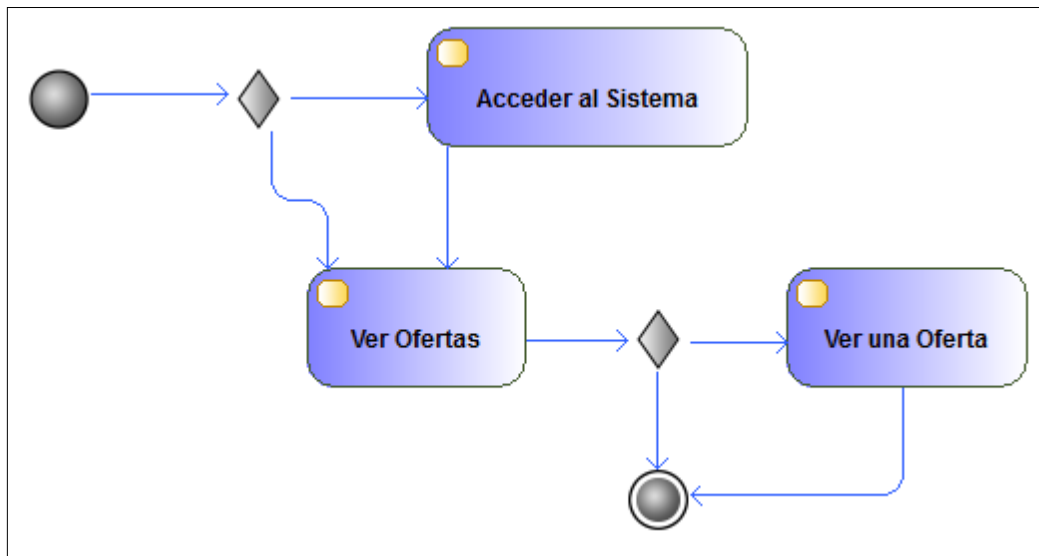


Figura 12: Diagrama de actividades del caso de uso 'Visualización de ofertas'

- Inscripción en una oferta. El usuario tras acceder al sistema puede, visualizando todas o la oferta en particular, inscribirse en una de las ofertas. Tras una validación, se envía al candidato un correo electrónico confirmando la inscripción.

- Diagrama de secuencia (Figura 13):

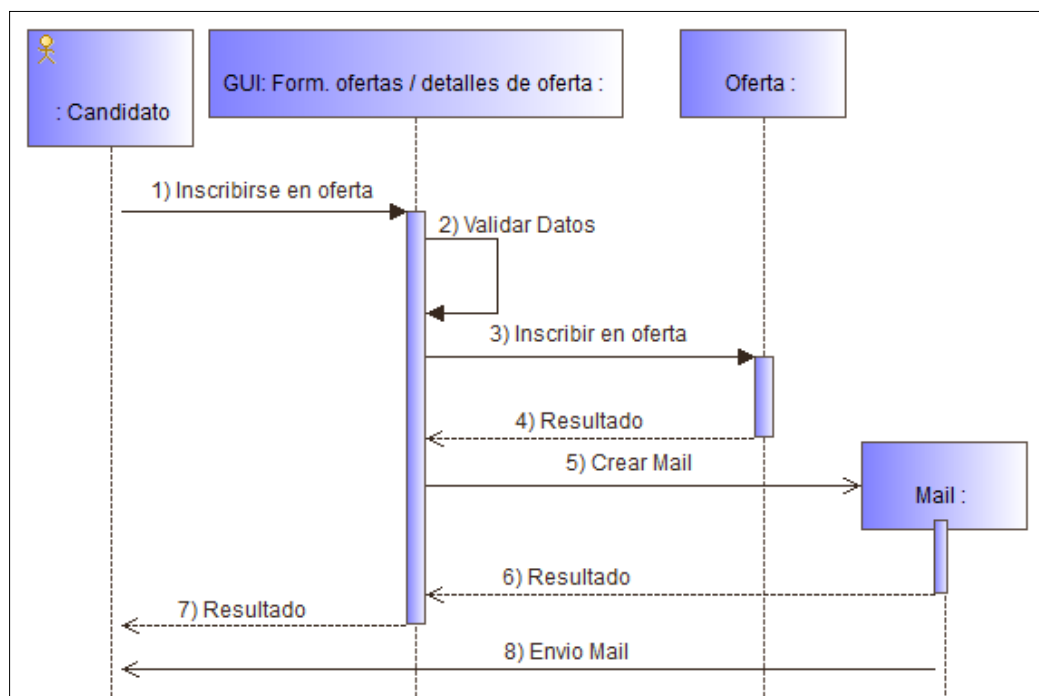


Figura 13: Diagrama de secuencia del caso de uso 'Inscripción en una oferta'



- Diagrama de actividades (Figura 14):

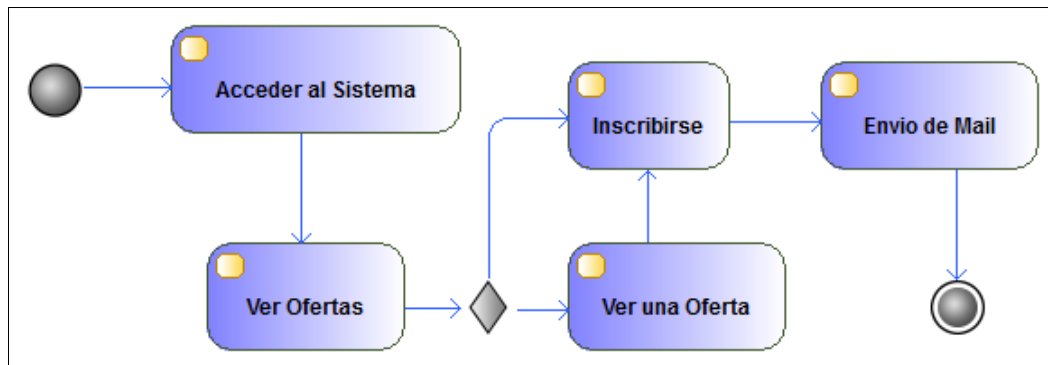


Figura 14: Diagrama de actividades del caso de uso 'Inscripción en una oferta'

- Darse de baja de una oferta. El usuario tras acceder al sistema puede, visualizando todas o la oferta en particular, darse de baja en una oferta en la que está inscrito. Tras una validación, se envía al candidato un correo electrónico confirmando la inscripción. Los diagramas de secuencia y actividades son análogos al caso de uso anterior.
- Obtención del informe de una oferta. Tanto usuarios registrados como sin registrar pueden obtener un informe en PDF con la información de una oferta.

- Diagrama de secuencia (Figura 15):

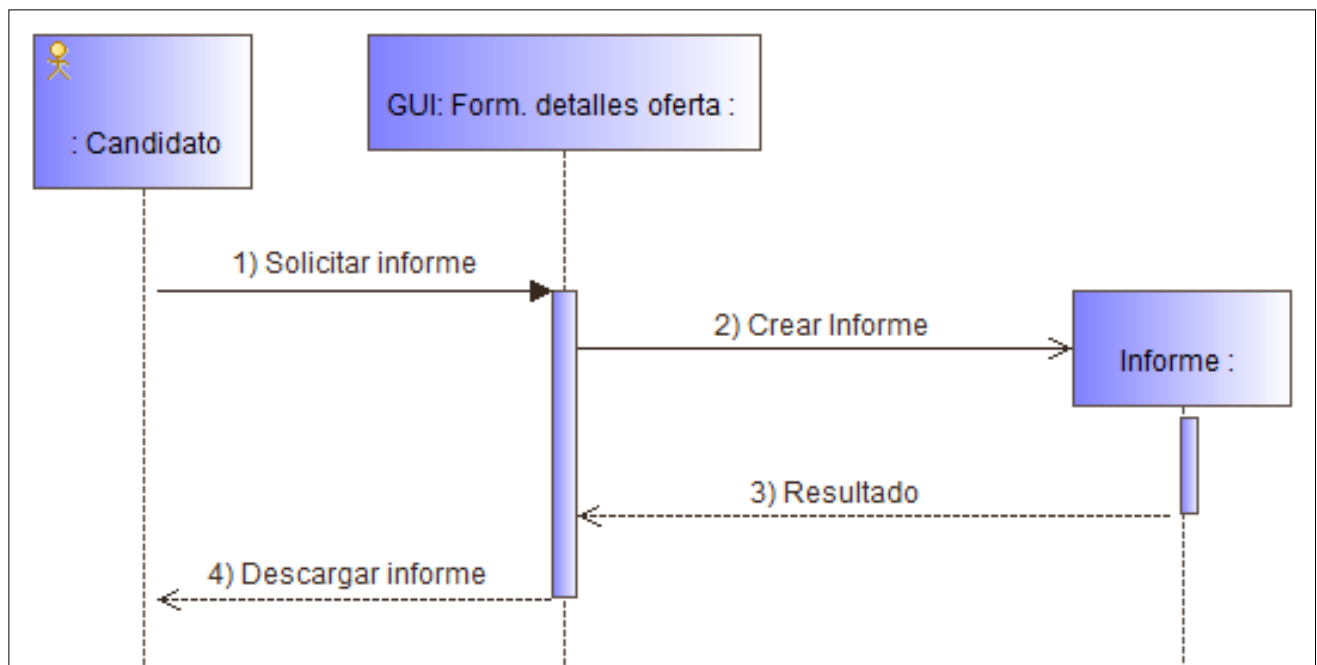
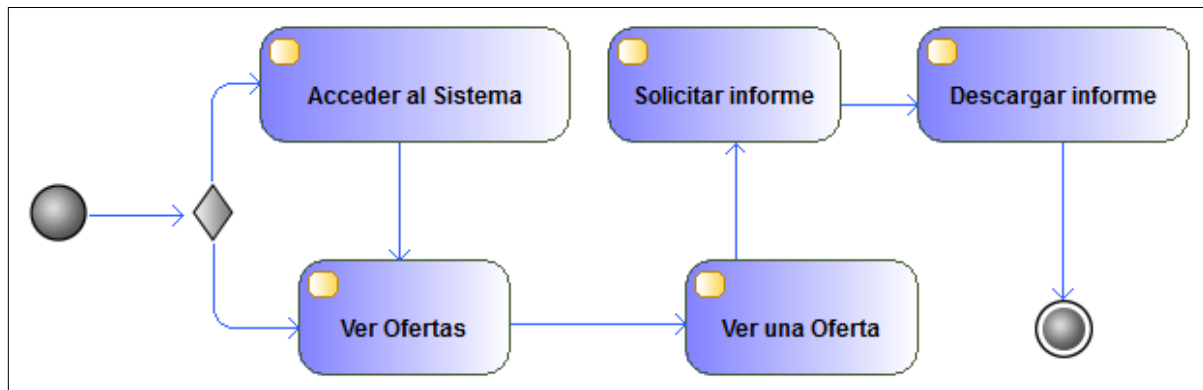


Figura 15: Diagrama de secuencia del caso de uso 'Obtención del informe de una oferta'

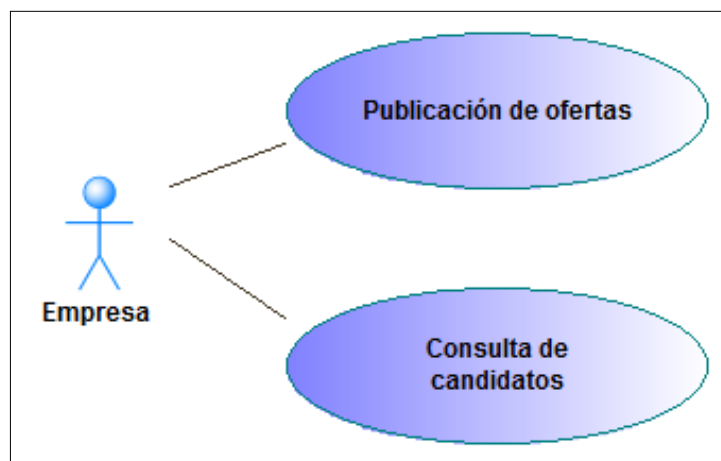
- Diagrama de actividades (Figura 16):



*Figura 16: Diagrama de actividades del caso de uso 'Obtención del informe de una oferta'*

### 3.3.3 Empresa

En este apartado se describen los casos de uso referidos la empresa mediante el sistema software de gestión de recursos humanos de Endalia. En la siguiente figura (Figura 17) se observa el diagrama del caso de uso 'Empresa' con un mayor nivel de detalle. Hay que tener en cuenta en todo momento que la AWSC está integrada dentro del sistema de gestión desarrollado por Endalia, y que es este sistema quien interactúa con la aplicación.



*Figura 17: Caso de uso de la empresa.*

Como indica la figura 17, la empresa posee tres casos de uso:

- Publicación de ofertas
- Consulta de candidatos

A continuación se analizará individualmente cada uno de estos casos de uso.

- Publicación de ofertas. Engloba la creación, modificación y eliminación de las ofertas relativas a las vacantes de la organización.

- Diagrama de secuencia (Figura 18):

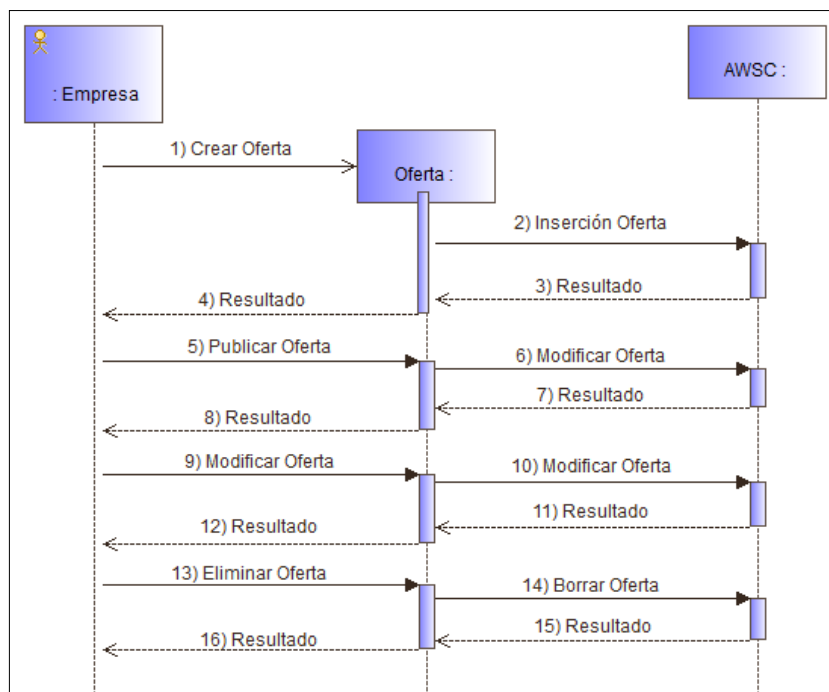


Figura 18: Diagrama de secuencia del caso de uso 'Publicación de ofertas'

- Diagrama de actividades (Figura 19):

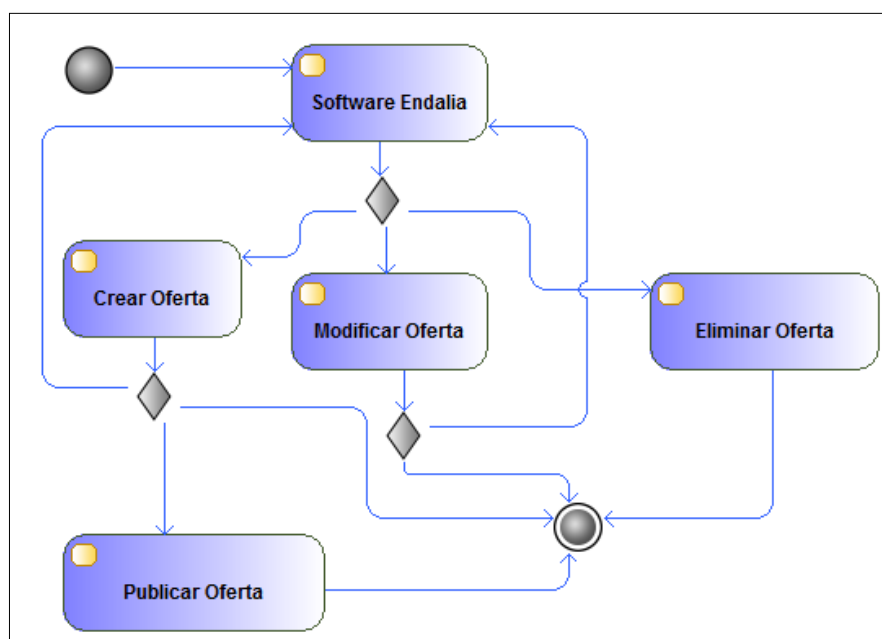


Figura 19: Diagrama de actividades del caso de uso 'Publicación de ofertas'

- Consulta de candidatos. La empresa puede consultar los CV o conocer en qué ofertas está inscrito cada uno.
  - Diagrama de secuencia (Figura 20):

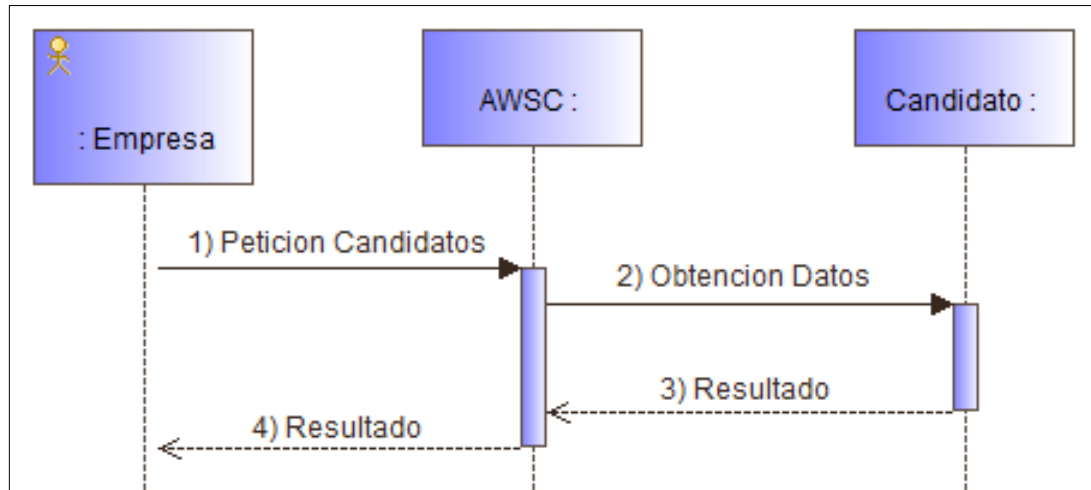


Figura 20: Diagrama de secuencia del caso de uso 'Consulta de candidatos'

- Diagrama de actividades (Figura 21):

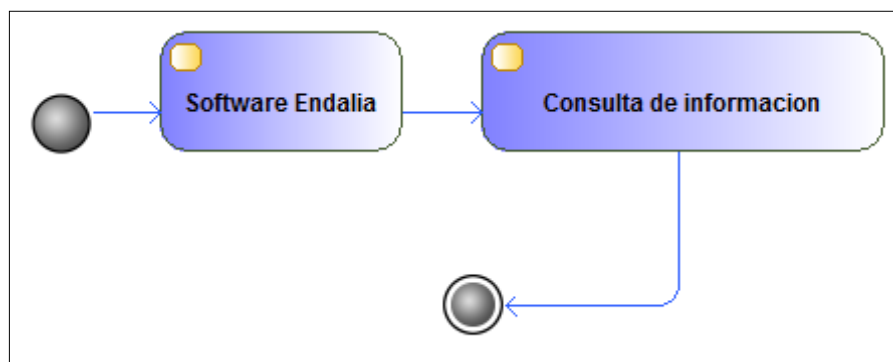


Figura 21: Diagrama de actividades del caso de uso 'Consulta de candidatos'

## 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

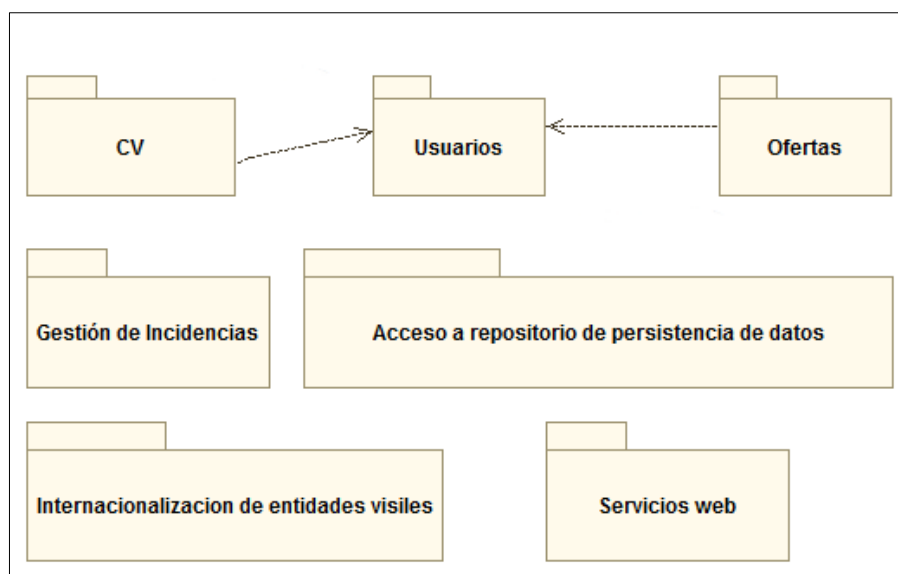


Figura 22: Paquetes de análisis de la AWSC

#### 4.2.1 Usuarios

El paquete de 'Usuarios', engloba todo lo relacionado con los candidatos registrados en el sistema. Se agrupan en este paquete los casos de uso de 'Registro de usuario', 'Acceso al sistema', 'Eliminación del usuario' y 'Consulta de candidatos'.

#### 4.2.2 CV (Currículo Vitae)

Por la relevancia de los CV de los candidatos, se ha separado en un paquete individual. El CV de los candidatos debe diseñarse de tal forma que sea especialmente escalable por las posibles modificaciones que pueda sufrir en un futuro, adaptándose a posibles necesidades. El paquete engloba el caso de uso 'Mantenimiento de CV'.

### 4.2.3 Ofertas

Las ofertas son otro de los puntos principales en el sistema, y por ello se ha separado en otro paquete individual. En dicho paquete se agrupan los casos de uso de 'Visualización de ofertas', 'Obtención de informe de oferta' y 'Publicación de ofertas'. También se incluyen 'Inscripción en una oferta' y 'Darse de baja de una oferta'.

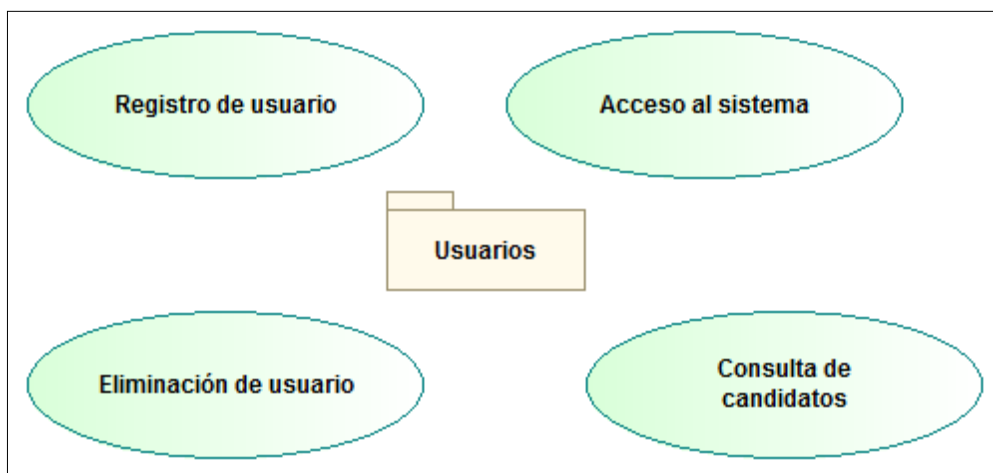


Figura 23: Paquete de análisis 'Usuarios'

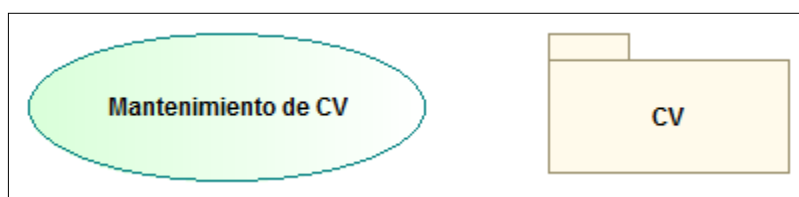


Figura 24: Paquete de análisis 'CV'

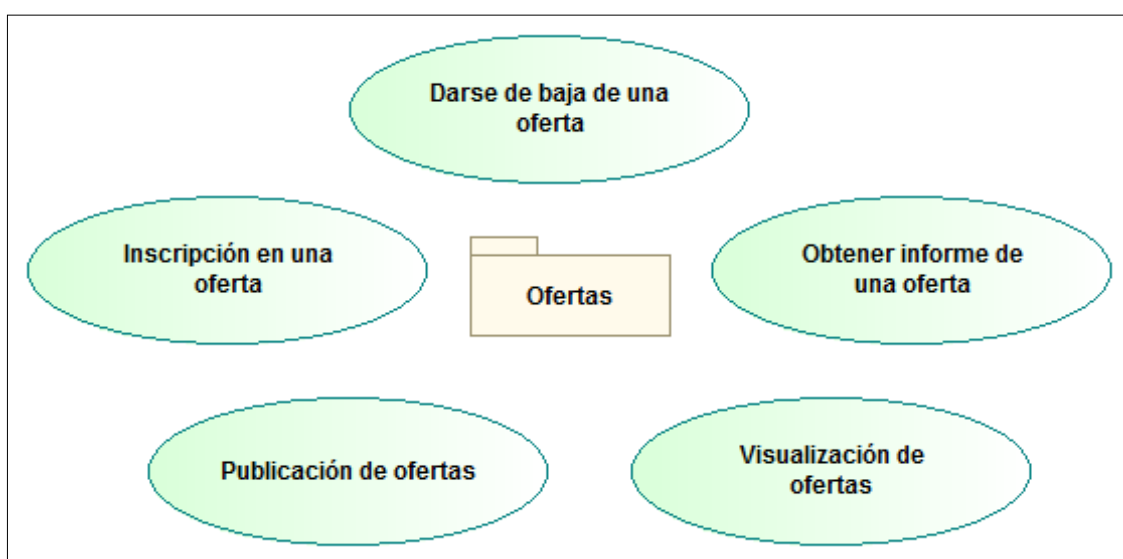


Figura 25: Paquete de análisis 'Ofertas'

#### 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.

#### 4.2.7 Servicios web

Este paquete de análisis engloba la funcionalidad necesaria para asegurar la seguridad e integridad de los datos.



## 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.

### 5.4 Seguridad

En cualquier proyecto software es un requisito fundamental que el sistema garantice la seguridad de las comunicaciones entra la GUI y el repositorio de almacenamiento, así como la privacidad de los datos. En el caso de este trabajo, este requerimiento se va a llevar a cabo con el uso de servicios web seguros, que están incluidos en el paquete de análisis “Servicios web”.

### 5.5 Disponibilidad del sistema

El sistema debe ser multiusuario, y además, debe acceder a un repositorio de información, por lo que se debe controlar el acceso de cada usuario a la información almacenada, evitando escrituras simultáneas en el repositorio de datos. También se debe garantizar la concurrencia en las interacciones de los usuarios con la aplicación, sin que empeore el rendimiento.





## 6. BIBLIOGRAFÍA

### 6.1 Referencias

- [R1] I. Jacobson, G. Booch, J. Rambaugh. 2000. "El Proceso Unificado de Desarrollo de Software". Pearson Education
- [R2] I. Jacobson, G. Booch, J. Rambaugh. 1999. "El lenguaje unificado de modelado. Manual de referencia". Ed. Addison Wesley.
- [R3] Martin Fowler. 1999. "UML Destilled". Addison-Wesley 1999 (2nd ed.).

### 6.2 Referencias web

- [W1] <http://www.uml.org>
- [W2] <http://www.wikipedia.org>
- [W3] <http://www.rational.com>
- [W4] <http://www.cs.ualberta.ca/~pfiguero/soo/uml/>
- [W5] <http://www.endalia.com>



# DISEÑO

## SISTEMA DE RECLUTAMIENTO IMPLEMENTADO CON SERVICIOS WEB

VERSIÓN 1.3  
PUBLICADO EL 12/02/2015

Copyright © 2014 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
13/08/2014	1.0	Redacción inicial del documento	Sergio Sanz Frías
14/11/2014	1.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	1.2	Revisión del documento	Fernando Cortés Franco
12/02/2015	1.3	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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.....	8
1.6 Resumen.....	8
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 .....	11
3.4.2 <i>Common Language Runtime</i> .....	12
3.4.3 Biblioteca de Clases Base (BCL).....	12
3.5 SQL Server .....	14
3.6 IIS.....	14
3.7 WCF.....	14
4. Diseño de la arquitectura .....	15
4.1 Introducción.....	15
4.2 Estructura general del sistema.....	15
4.3 Estructura de capas del sistema .....	15
4.4 Estructura de subsistemas.....	16
4.4.1 Subsistema de usuarios .....	16
4.4.2 Subsistema de currículos vitae .....	16
4.4.3 Subsistema de ofertas.....	16
4.4.4 Subsistema de gestión de incidencias.....	16
4.4.5 Subsistema de acceso a base de datos .....	16
4.4.6 Subsistema de internacionalización de entidades visibles.....	16
4.4.7 Subsistema de servicios web.....	16
5. Clases del sistema.....	17



5.1	Introducción.....	17
5.2	Clases de interfaz.....	17
5.2.1	Clases del subsistema de usuarios.....	17
5.2.2	Clases del subsistema de CV.....	17
5.2.3	Clases del subsistema de ofertas .....	18
5.3	Clases de acceso a datos .....	18
5.4	Clases de servicios web.....	18
5.5	Clases auxiliares.....	19
6.	Diseño de la base de datos.....	20
6.1	Introducción.....	20
6.2	Diseño general de la base de datos.....	20
6.2.1	Usuarios.....	20
6.2.2	Ofertas .....	21
6.2.3	CV .....	21
7.	Prototipado de la interfaz .....	22
7.1	Introducción.....	22
7.2	Interfaz básico .....	22
7.3	Interfaz del contenido de página de registro .....	23
7.4	Interfaz del contenido de página de visualización de ofertas .....	23
7.5	Interfaz del contenido de página de detalles de una oferta.....	24
7.6	Interfaz del contenido de página de datos personales del CV .....	24
7.7	Interfaz del contenido de página de información del CV .....	25
8.	Bibliografía.....	26
8.1	Referencias.....	26
8.2	Referencias web.....	26



# 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 de la AWSC.

## 1.3 Acrónimos

- ACID: Atomicity- Consistency-Isolation-Durability.
- API: Application Programming Interface.
- ASP: Active Server Pages.
- AWSC: Aplicación Web de Selección de Candidatos.
- BCL: Base Class Library.
- CIL: Common Intermediate Language.
- CLI: Common Language Infrastructure.
- CLR: Common Language Runtime.
- CLS: Common Language Specification.
- CTS: Common Type System.
- DDL: Data Definition Language.
- DML: Data Manipulation Language.
- ECMA: European Computer Manufacturer Association.
- FTP: File Transfer Protocol.
- FTPS: File Transfer Protocol Secure.
- GDI: Graphics Device Interface.
- GUI: Graphic User Interface.
- HTTP: HyperText Transfer Protocol.
- HTTPS: HyperText Transfer Protocol Secure.
- IEC: International Electrotechnical Commission.
- IIS: Internet Information Services.
- ISO: International Organization for Standardization.
- JIT: Just-in-Time.



- LDAP: Lightweight Directory Access Protocol.
- LINQ: Language Integrated Query.
- LGPL: Lesser GNU Public License.
- MSIL: Microsoft Intermediate Language.
- NNTP: Network News Transfer Protocol.
- ORM: Object Relational Mapping.
- SGBD: Sistema Gestor de Bases de Datos.
- SMTP: Simple Mail Transfer Protocol.
- SOAP: Simple Object Access Protocol.
- TCP/IP: Transmission Control Protocol / Internet Protocol.
- T-SQL: Transact – Structured Query Language.
- WCF: Windows Communication Foundation.
- 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.
- *Framework*: es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado.
- Mapeo objeto-relacional (ORM): es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos.
- *ToolTip*: es una herramienta de ayuda que funciona al situar o pulsar con el ratón sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra. Los *tooltip* son una variación de los globos de ayuda y es un complemento muy usado en programación, dado que proporcionan información adicional sin necesidad de que el usuario





la solicite.

## 1.5 Referencias

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

- ESTMER: estudio de mercado
- ESPREQ: documento de especificación de requisitos
- ANALISIS: documento de análisis del sistema

## 1.6 Resumen

Este documento describe el proceso de diseño de la AWSC. Se compone de ocho apartados:

1. Introducción del documento, definición del propósito y alcance del mismo.
2. Se describe el proceso de diseño seguido para la confección de este documento.
3. Se describen las decisiones y restricciones iniciales del diseño, y se comparan y describen diferentes alternativas.
4. Descripción de la arquitectura del sistema tanto a nivel físico como a nivel de organización del mismo.
5. Detalle de las clases identificadas.
6. Detalle del diseño de la base de datos.
7. Prototipado del interfaz de las diferentes pantallas con las que ha de interactuar el usuario del sistema.
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 de la AWSC, 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 trabajo 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 la AWSC se va a desarrollar en la plataforma .NET de Microsoft, con ASP en las páginas Web usando el lenguaje de programación C# y el gestor de base de datos Microsoft SQL Server. Para la implementación de los servicios web se utilizará WCF.

### 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:

- Acceso a base de datos de forma transaccional, cumpliendo el principio ACID.
- 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.
- Seguridad: el sistema debe ser fiable, tanto a nivel de autenticación, como de autorización, y debe mantener la privacidad de la información confidencial.
- Rendimiento: el sistema debe soportar un incremento en la carga de trabajo sin que ello repercuta notablemente en el usuario.

### 3.4 Plataforma .NET

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



### 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.

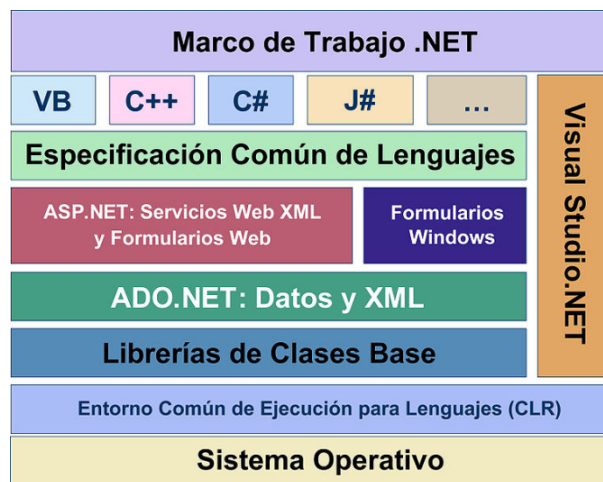


Figura 1: .NET Framework

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):

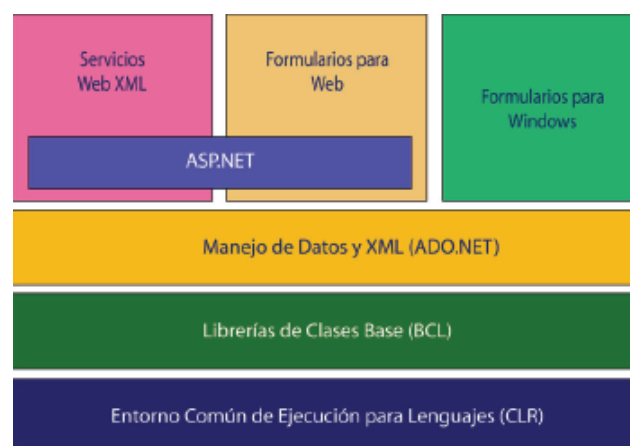


Figura 3: Esquema de distribución jerárquica de Namespaces

### 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 IIS

*Internet Information Services*, desarrollado por Microsoft, es un conjunto de servicios sobre TCP/IP que permiten a la máquina donde está alojado proporcionar funcionalidad de servidor de aplicaciones. Actualmente soporta los siguientes protocolos: HTTP, HTTPS, FTP, FTPS, SMTP y NNTP.

Como servidor web, permite definir una serie de directorios virtuales, donde se encuentra cada una de las aplicaciones indexadas por él, resolviendo las peticiones de páginas estáticas, y gestionando las necesidades de código dinámico de las mismas. Asimismo, proporciona mecanismos de seguridad y autenticación de usuarios, donde permite una total integración con el *Active Directory* de Microsoft Windows.

### 3.7 WCF

*Windows Communication Foundation* (WCF) es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Con WCF, es posible enviar datos como mensajes asincrónicos de un extremo de servicio a otro. Un extremo de servicio puede formar parte de un servicio disponible continuamente hospedado por IIS, o puede ser un servicio hospedado en una aplicación. Un extremo puede ser un cliente de un servicio que solicita datos de un extremo de servicio web. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios.

En resumen, WCF se ha diseñado para ofrecer un enfoque manejable para la creación de servicios web y clientes de servicios web.



## 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 la AWSC.

### 4.2 Estructura general del sistema

La estructura general del sistema está definida por la plataforma de otros servicios web de Endalia. Por lo tanto, la estructura del proyecto debe adaptarse a la de estos servicios. A continuación se detallan las restricciones que deben mantenerse, referente a la estructura:

- Servidor IIS: servidor de aplicaciones donde se aloja la plataforma de servicios web. Resuelve peticiones (https en este caso, pero también de otros tipos) y las asigna a su aplicación correspondiente, en función de los permisos previamente establecidos.
- Base de datos: repositorio de datos de la aplicación, implementado desde el sistema gestor de bases de datos SQL Server 2012.

### 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; lo que proporciona una óptima escalabilidad y mejorar el rendimiento.

El sistema está basado en la siguiente arquitectura multicapa típica:

- Capa de cliente: formada por los componentes que se ejecutan en el cliente. En nuestro caso es el navegador web desde el que se accede a la plataforma y el código JavaScript que se ejecuta en algunas páginas de la aplicación.
- Capa de presentación. Es la capa que se encarga de crear el interfaz gráfico que renderiza la capa de cliente y de gestionar las interacciones de éste con el sistema. Esto se corresponde en el sistema con las páginas aspx y su código subyacente.
- 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 servicios web: es la capa que ofrece a la aplicación el acceso seguro a los datos.
- Capa de integración. Contiene objetos que automatizan el acceso a datos. Esto se corresponde con los procedimientos almacenados en la base de datos.
- 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.





## 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 la aplicación.

### 4.4.1 Subsistema de usuarios

En este subsistema se realizan las funcionalidades propias de los usuarios del sistema, su acceso al mismo junto con su modificación de datos de acceso.

### 4.4.2 Subsistema de currículos vitae

En este subsistema se engloban todas las funcionalidades necesarias para la inserción de información del candidato en su CV.

### 4.4.3 Subsistema de ofertas

En este subsistema se realizan las funcionalidades necesarias para el mantenimiento de la información de las ofertas y su consulta por parte de los candidatos.

### 4.4.4 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

### 4.4.5 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.

### 4.4.6 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.

### 4.4.7 Subsistema de servicios web

En este subsistema se desarrollan las funcionalidades necesarias para que la aplicación sea segura y consiga una integridad completa de los datos.



## 5. CLASES DEL SISTEMA

### 5.1 Introducción

En este apartado se detallan las clases de la AWSC. Estas clases se dividen en cuatro 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.
- Clases de servicios web: relacionadas con las clases de acceso a datos.
- Clases auxiliares.

### 5.2 Clases de interfaz

En esta sección se van a analizar las clases de interfaz de la aplicación. Como ya se ha comentado, el proyecto se desarrolla en .NET y el interfaz en asp. La estructura básica de la página web que contiene cabecera, menús y pie de página está contenida en la página “*master*”. Esto permite la creación del resto de las páginas del sistema como un contenido de los master, y por lo tanto la reutilización de la estructura base, permitiendo la realización de modificaciones en la estructura básica de forma rápida y sencilla.

También facilita la adaptación de la aplicación para otra organización distinta a la que se ha destinado en este trabajo.

A continuación se comentan las distintas clases de interfaz agrupándolas por subsistemas.

#### 5.2.1 Clases del subsistema de usuarios

- Login.aspx: formulario utilizado por el usuario para registrarse en el sistema.
- RegisterEnd.aspx: este formulario sirve como página a la que se redirecciona desde el correo electrónico enviado al candidato al registrarse.
- ForgotPass.aspx: formulario donde el candidato puede solicitar la contraseña en caso de no recordarla (se puede encontrar habitualmente en todo tipo de web que requiera un acceso con contraseña).
- NewPass.aspx: formulario al que se redirecciona al candidato a través de un correo electrónico tras completar el anterior; aquí establece su nueva contraseña.
- ChangePass.aspx: formulario donde el usuario puede cambiar su contraseña o darse de baja del sistema.

#### 5.2.2 Clases del subsistema de CV

- CurriculumMenu.ascx: menú lateral donde acceder a las distintas secciones del currículum.
- UserPersonalData.aspx: formulario donde aparecen los principales datos personales del candidato.
- UserPhoto.aspx: formulario donde el candidato puede visualizar y/o modificar su fotografía.
- UserTrainingData.aspx: formulario donde aparecen los datos de formación del candidato.
- UserExperiencia.aspx: formulario donde aparece la información del candidato sobre su experiencia.



- UserExpectations.aspx: formulario donde aparece las áreas de interés del candidato.
- UserLanguages.aspx: formulario donde aparecen los idiomas conocidos por el candidato.
- UserReferences.aspx: formulario donde aparecen referencias externas del candidato.
- UserFiles.aspx: formulario donde aparecen ficheros que pueden ser de utilidad para la organización o para completar algún información de las demás secciones del currículum vitae.
- UserRemarks.aspx: formulario donde poder introducir un comentario en un campo de texto libre.

### 5.2.3 Clases del subsistema de ofertas

- Index.aspx: formulario principal y en el cual aparecen los puestos vacantes disponibles en la organización que pueden visualizarse sin necesidad de registrarse.
- SearchOffers.aspx: formulario similar al anterior pero preparado para usuarios ya registrados.
- RecruitmentMyOffers.aspx: formulario donde se muestran las ofertas en las que el usuario está inscrito.
- OfferDetails.aspx: formulario donde aparece la información sobre la oferta y desde el que es posible obtener un informe PDF de la misma, disponible tanto para usuarios registrados como no registrados, aunque solo los registrados podrán inscribirse en la oferta.

## 5.3 Clases de acceso a datos

En esta sección se van a analizar las clases de acceso a datos de la AWSC. Las más destacadas son:

- AdministrationItem.cs: clase importante ya existente previamente para poder controlar el uso de parametrizaciones en el sistema software de Endalia.
- EmployeeExternalRecruitmentTmp.cs: clase que representa los datos de los usuarios que han completado los datos de registro pero que aún no lo han confirmado a través del enlace del correo electrónico enviado.
- Employee.cs: clase que representa los datos de los usuarios ya registrados completamente.
- EmployeeFiles.cs: clase que representa datos de un fichero que los usuarios introducen en su CV.
- EmployeeLanguage.cs: clase que representa datos de un idioma que los usuarios introducen en su CV.
- EmployeePhoto.cs: clase que representa datos de la foto introducida por los usuarios.
- EmployeeReference.cs: clase que representa datos de una referencia que los usuarios introducen en su CV.
- R\_Employee\_AreasOfInterest.cs: es la clase de acceso a datos que representa la relación entre un candidato y sus áreas de interés.
- R\_Employee\_ExternalExperience.cs: es la clase de acceso a datos que representa la relación entre un candidato y su experiencia anterior.
- Job.cs: clase que representa los datos de un puesto de trabajo de la organización.
- JobOffer.cs: clase que representa los datos de una oferta de una vacante de la organización.
- SystemOption.cs: clase importante ya existente previamente para poder controlar el uso de opciones de sistema en el sistema software de Endalia.

## 5.4 Clases de servicios web

A las clases de acceso a datos anteriores les corresponden las siguientes clases de servicios web:

- AdministrationItemWCF.cs



- EmployeeExternalRecruitmentTmpWCF.cs
- EmployeeWCF.cs
- EmployeeFilesWCF.cs
- EmployeeLanguageWCF.cs
- EmployeePhotoWCF.cs
- EmployeeReferenceWCF.cs
- R\_Employee\_AreasOfInterestWCF.cs
- R\_Employee\_ExternalExperienceWCF.cs
- JobWCF.cs
- JobOfferWCF.cs
- SystemOptionWCF.cs

## 5.5 Clases auxiliares

Las clases auxiliares son un tipo de clases con diferentes utilidades sobre los datos del sistema.

- WCFGenerics.cs: clase utilizada con objetos genéricos para realizar las conversiones desde los objetos utilizados en la comunicación con los servicios web a los objetos de las clases utilizadas por el sistema de software de Endalia, o viceversa.
- WCFUtils.cs: clase con métodos útiles en el uso de WCF.
- ToolTipUtil.cs: clase que permite añadir y quitar *tooltips*.
- StringUtil.cs: clase que contiene funciones de comprobación de sintaxis de correos electrónicos, cálculo de tiempo transcurrido en modo texto, etc.



## 6. DISEÑO DE LA BASE DE DATOS

### 6.1 Introducción

En este apartado se presenta una vista global del diseño de la base de datos, resultante de las entidades identificadas en el análisis, adecuando al repositorio de datos que utiliza la aplicación. Como ya se ha comentado anteriormente, el repositorio de datos a utilizar, es una base de datos relacional gestionada desde el SGBD MSSQL Server 2012.

También se describen las tablas de la base de datos junto con sus atributos y las relaciones existentes entre ellas.

### 6.2 Diseño general de la base de datos

Debido al tamaño de la base de datos, se va a separar en subconjuntos de menor tamaño, ya que no sería posible representarlo de otro modo en este formato de documento.

#### 6.2.1 Usuarios

Contiene la información relacionada con los usuarios del sistema y sus datos de acceso. Todos los usuarios del sistema tienen una entrada en la tabla *Gen\_Users*, ya que es la que contiene los datos de acceso al mismo. Los candidatos tienen otra entrada en la tabla *Orh\_Emp\_Employees*. Los candidatos que no han confirmado su registro se almacenan en la tabla *Orh\_Emp\_EmployeeExternalRecruitmentTmp*.

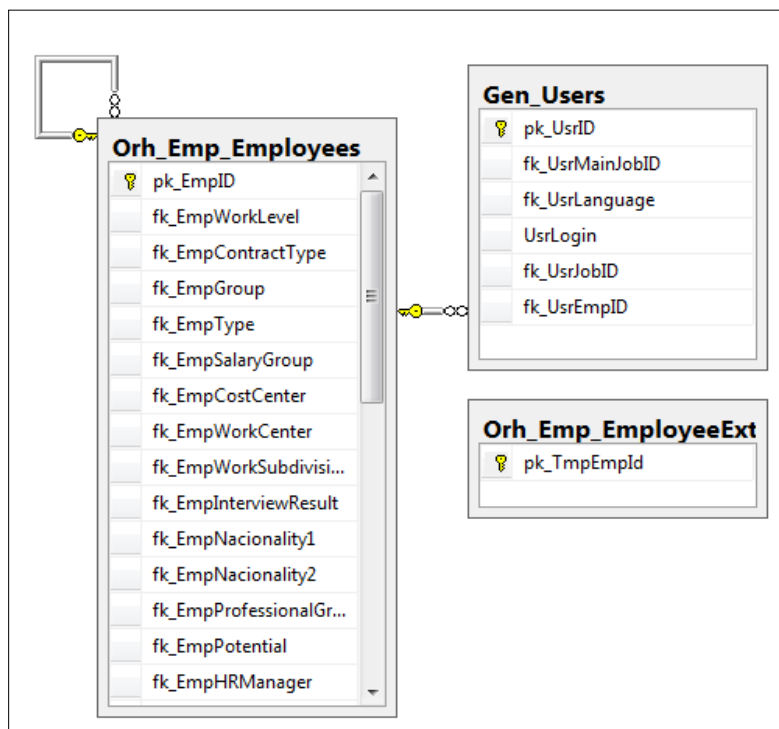


Figura 4: Esquema general de usuarios de la AWSC



### 6.2.2 Ofertas

Contiene la información relacionada con las ofertas. Las ofertas se relacionan con un puesto de trabajo, y el puesto de trabajo con los candidatos (empleados en otras aplicaciones de Endalia). Otras características de las ofertas están en relaciones a parte como se muestran en la Figura 5:

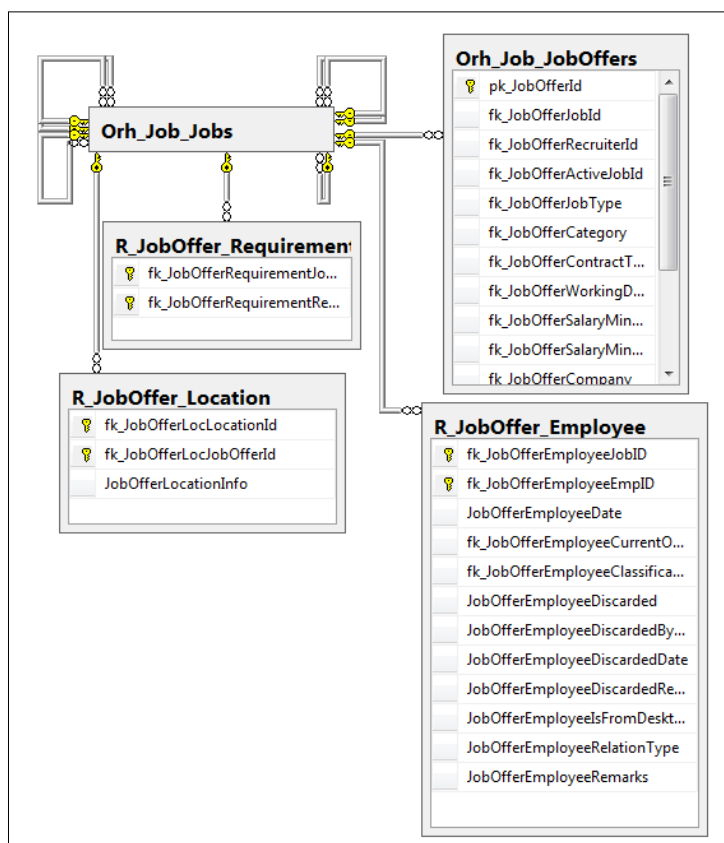


Figura 5: Esquema general de ofertas de la AWSC

### 6.2.3 CV

Contiene la información del currículum vitae de los candidatos. Hay numerosas relaciones con clases para almacenar la información parametrizada del candidato.

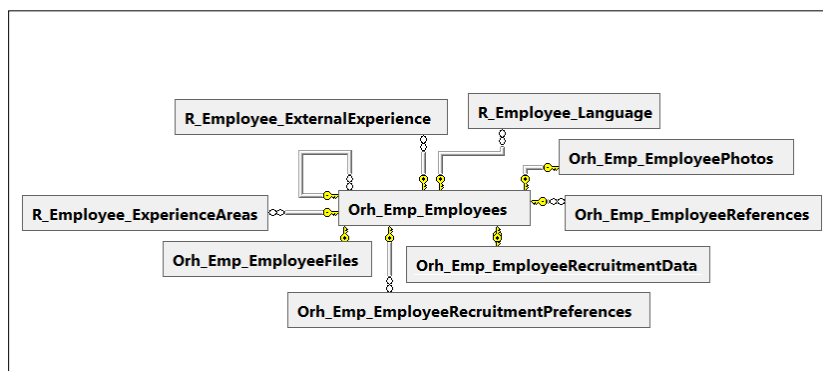


Figura 5: Esquema general de currículum vitae de la AWSC

## 7. PROTOTIPADO DE LA INTERFAZ

### 7.1 Introducción

En este apartado se muestran los prototipos de interfaz de pantalla diseñados para el sistema, teniendo en cuenta el interfaz de la web corporativa de la organización a la que va destinado este proyecto realizado dentro de Endalia.

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, proporcionando un entorno claro e intuitivo para los usuarios.
- Evitar la aparición de barras de desplazamiento horizontal, y limitar dentro de lo posible el uso de la barra de desplazamiento vertical, ocupando a poder ser únicamente la pantalla visible.

### 7.2 Interfaz básico

A continuación se muestra la base del diseño del interfaz gráfico, enmarcando las diferentes secciones que lo componen (Figura 6).

Los aspectos más significativos son los siguientes:

- Cabecera: debe ocupar la parte superior de la pantalla. El diseño de la misma es el utilizado por la organización en su web corporativa.
- Menú: situado en el lateral izquierdo, siguiendo el estilo corporativo de la organización una vez más.
- Contenido: la parte central de la página.
- Pie de página: debe ocupar la parte inferior de la pantalla. Al igual que la cabecera, su diseño es el mismo utilizado en la organización en su web corporativa.

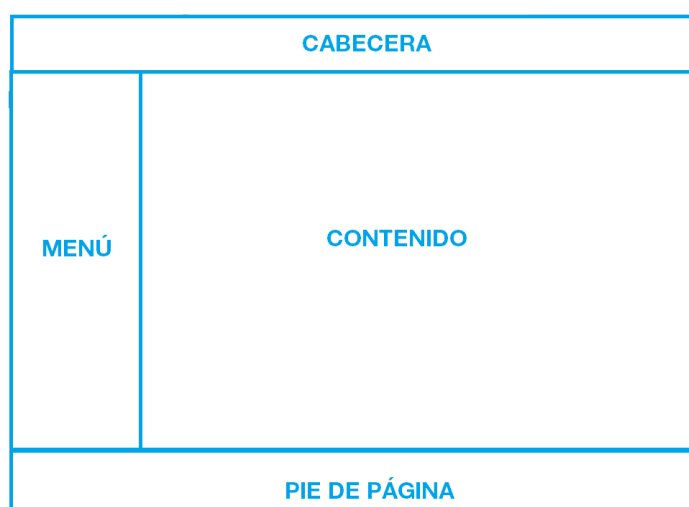


Figura 6: Prototipo de interfaz básico

## 7.3 Interfaz del contenido de página de registro

A continuación se muestra el prototipo del contenido en la página de registro (Figura 7):

**Registro de nuevos usuarios**

¿Aún no te has registrado en nuestra bolsa de empleo?  
Regístrate aquí y podrás:

- ✓ Introducir tu curriculum
- ✓ Apuntarte a los procesos de selección
- ✓ Recibir ofertas en tu e-mail

La dirección de correo electrónico que introduzcas será tu nombre de usuario en el sistema.  
Un correo electrónico con los datos para activar tu usuario será enviado a esta dirección una vez completado el registro.

---

Email*	<input type="text"/>	Apellidos*	<input type="text"/>
Nombre*	<input type="text"/>	Teléfono	<input type="text"/>
Fecha nac.*	<input type="text"/>	Confirmar contraseña*	<input type="text"/>
Contraseña*	<input type="password"/>		

Solicitante de empleo ☐      Solicitante de prácticas remuneradas ☐

[Política de privacidad](#)

☐ He leído y acepto la política de privacidad de

(\*) Campos obligatorios

[Finalizar registro](#)

Figura 7: Prototipo de página de registro

## 7.4 Interfaz del contenido de página de visualización de ofertas

A continuación se muestra el prototipo del contenido en la página de visualización de ofertas (Figura 8):

**Ofertas de empleo** test@endalia.com | [Desconectar](#)

¿Quieres trabajar en  
En esta sección se publican nuestras ofertas de empleo.

[Lista de puestos vacantes \(2 puestos encontrados\)](#)

**Administrativo área de formación** Hace 2 meses

Se precisa administrativo para el área de formación de sistemas, tabulación de encuestas y reporting. , para el envío comunicados, confirmación de participantes, actualización de

[Ver en detalle](#) [Darse de baja](#)

**Beca en el área de comunicación** Hace 2 meses

ofrece una beca para la colaboración en la elaboración de convocatoria y notas de prensa, búsqueda, recopilación y gestión de información corporativa y seguimiento de alerta de noticias

[Ver en detalle](#) [Inscribirse](#)

Figura 8: Prototipo de visualización de ofertas





## 7.5 Interfaz del contenido de página de detalles de una oferta

A continuación se muestra el prototipo del contenido en la página de detalles de una oferta (Figura 9):

test@endalia.com | [Desconectar](#)

**Proceso de selección**  
**Técnico/a de Relaciones con Empresas y Fundaciones**

**Descripción**

Puesto:	Técnico/a de Relaciones con Empresas y Fundaciones
Área:	Comercial / Marketing
Tipo de oferta:	Externa
Número de vacantes:	0
Descripción del puesto:	precisa incorporar un/a Técnico/a de Relaciones con Empresas y Fundaciones, para la oficina central de Barcelona. La misión del puesto consiste en Planificar y ejecutar la captación y la fidelización de Empresas, Fundaciones y otras instituciones privadas de acuerdo con el Plan de Acción establecido y los objetivos estratégicos de la , con el objetivo de aumentar el número de donantes así como su fidelización.
Fecha de publicación:	01/09/2014
Fecha límite envío CV:	30/09/2014

[Darse de baja](#)

Figura 9: Prototipo de detalles de una oferta

## 7.6 Interfaz del contenido de página de datos personales del CV


A continuación se muestra el prototipo del contenido en la página de datos personales del currículum vitae (Figura 10):

test@endalia.com | [Desconectar](#)

**Datos personales**

**Información personal**

Nombre:*	<input type="text" value="Sergio"/>
Apellidos:*	<input type="text" value="Sanz Frías"/>
DNI:*	<input type="text" value="122345722a"/>
Nacionalidad:*	<input type="text" value="España"/>
Fecha nacimiento:*	<input type="text" value="05/09/1990"/>
Sexo:*	<input type="text" value="Hombre"/>

  
(Haz clic sobre la foto para editar)

**Información de contacto**

Email:	test@endalia.com
País:*	<input type="text" value="España"/>
Provincia:*	<input type="text" value="Zaragoza"/>
Población:*	<input type="text" value="Zaragoza"/>
Dirección:	<input type="text" value="Paseo Independencia"/>
Código postal:*	<input type="text" value="50010"/>
Teléfono fijo:*	<input type="text" value="976334567"/>
Teléfono móvil:	<input type="text" value="666123456"/>
Web/blog:	<input type="text" value="www.endalia.com"/>

**Información adicional**

☐ Trabajo actualmente

☒ Disponibilidad para viajar

☒ Vehículo propio

☒ Carné de conducir

Figura 10: Prototipo de datos personales del CV



## 7.7 Interfaz del contenido de página de información del CV

A continuación se muestra el prototipo del contenido en las otras páginas del CV donde se introduce información que se encuentra parametrizada (Figura 11):

test@endalia.com | [Desconectar](#)

### Formación

En esta sección puedes introducir la formación que poseas. Si no deseas incluir ningún dato, marca la siguiente opción para dar por completada esta sección.

#### Titulaciones

☐ No aplicable en mi caso. Dar esta información por completada.

Introduce las titulaciones académicas que posees.

Fecha inicio	Fecha fin	Titulación	Centro	Estado		
01/09/2005	30/09/2009	MASTER	Master RRHH	Reconocido		
16/09/1998	01/09/2002	RELACIONES LABORALES	Universidad de Zaragoza	Reconocido		
01/09/1989	30/06/1998	BACHILLERATO	Escolapios	Reconocido		

Añadir titulación

#### Cursos de formación

☐ No aplicable en mi caso. Dar esta información por completada.

Introduce los cursos de formación que has realizado. Nombre del curso, el centro formador, el área temática del curso, fecha de finalización del curso, su duración en horas y los detalles del curso.

Fecha fin	Curso	Área	Horas		
30/09/2004	Experto en Gestión de Salarios y Recursos Sociales	Gestión	200		

Añadir cursos

Figura 11: Prototipo de información en el CV

DISEÑO  
PÁGINA 25 DE 26

## 8. BIBLIOGRAFÍA

### 8.1 Referencias

- [R1] I. Jacobson, G. Booch, J. Rumbaugh. 2000. “El Proceso Unificado de Desarrollo de Software”. Pearson Education.
- [R2] I. Jacobson, G. Booch, J. Rumbaugh. 1999. “El lenguaje unificado de modelado. Manual de referencia”. Ed. Addison Wesley.
- [R3] J. Rumbaugh 1991. “Modelado y Diseño Orientado a Objetos”. Ed. Prentice Hall, 1991.
- [R4] Hoang Lam, Thuan L. Thai. “.NET Framework Essentials, 3rd Edition”. O’Reilly 2003.
- [R5] Art Gittleman. “Computing With C# and the .Net Framework”. Jones and Bartlett Publishers 2003.

### 8.2 Referencias web

- [W1] <http://www.microsoft.com/net>
- [W2] <http://www.wikipedia.org/>
- [W3] <http://www.uml.org/>
- [W4] <http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/>
- [W5] <http://www.rational.com/>
- [W6] <http://www.webstyleguide.com/index.html>
- [W7] <http://www.endalia.com>



# IMPLEMENTACIÓN

**SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB**

VERSIÓN 1.3  
PUBLICADO EL 12/02/2015

Copyright © 2014 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
08/09/2014	1.0	Redacción inicial del documento	Sergio Sanz Frías
15/11/2014	1.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	1.2	Revisión del documento	Fernando Cortés Franco
12/02/2015	1.3	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
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.....	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 .....	14
6.1 Introducción.....	14
6.2 SQL Server 2012 .....	14
6.2.1 T-SQL .....	14
6.2.2 Transacciones .....	14
6.2.3 Procedimientos almacenados.....	15
6.2.4 Desencadenadores.....	16
6.3 <i>Windows Communication Foundation</i> .....	16
6.3.1 Configuración de WCF .....	16
6.3.2 Archivos de clases de WCF .....	20
6.3.3 Seguridad de WCF .....	21
6.4 Acceso a datos .....	22
6.4.1 Configuración para acceder a los servicios web .....	22
6.4.2 Clases de acceso a datos .....	22
7. Implementación de la interfaz de usuario .....	24
7.1 Introducción.....	24



7.2	Elementos de interfaz.....	24
7.2.1	Iconos e imágenes.....	24
7.2.2	Tablas .....	24
7.2.3	Editores de Fechas.....	24
7.2.4	Colores y fuentes.....	25
7.2.5	Estilos .....	25
7.3	Pantallas del sistema.....	25
7.3.1	Página de inicio .....	25
7.3.2	Página de registro.....	26
7.3.3	Página de candidato.....	26
7.3.4	Página del currículo del candidato .....	27
8.	Bibliografía.....	31
8.1	Referencias.....	31
8.2	Referencias web.....	31





# 1. INTRODUCCIÓN

## 1.1 Propósito del documento

El presente documento describe la fase de implementación del trabajo de desarrollo de la AWSC. 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 de la AWSC, en la fase final del desarrollo del sistema.

## 1.3 Acrónimos

- AES: Advanced Encryption Standard.
- AJAX: Asynchronous JavaScript And XML.
- ANSI: American National Standards Institute.
- API: Application Programming Interface.
- ASP: Active Server Pages.
- AWSC: Aplicación Web de Selección de Candidatos.
- CRUD: Create, Read, Update, Delete.
- DAO: Data Access Object.
- DDL: Data Definition Language.
- DML: Data Manipulation Language.
- FTP: File Transfers Protocol.
- GUI: Graphical User Interface.
- HTML: HyperText Markup Language.
- HTTP: HyperText Transfer Protocol.
- HTTPS: HyperText Transfer Protocol Secure.
- IIS: Internet Information Services.
- RIA: Rich Internet Applications.
- SDK: Software Development Kit.
- SGBD: Sistema Gestor de Base de Datos.
- SMTP: Simple Mail Transfer Protocol.
- SQL: Structured Query Language.
- T-SQL: Transact Structured Query Language.
- WCF: Windows Communication Foundation.
- WPF: Windows Presentation Foundation.
- XML: Extensible Markup Language.



## 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.
- Hoja de estilo: lenguaje formal usado para definir la presentación de un documento descrito en HTML o XML.
- 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.
- Método Rijndael o AES: es un esquema de cifrado de bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. Es uno de los algoritmos más populares usados en criptografía simétrica.
- 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 trabajo:

- DISEÑO: documento que describe la fase de diseño del trabajo.
- ESTCOD: documento que especifica el estándar de codificación utilizado en el trabajo.

## 1.6 Resumen

Este documento describe el proceso de implementación de la AWSC. Se compone de ocho apartados:

1. Introducción del documento, definición del propósito y alcance del mismo.
2. Se describe el proceso de implementación seguido.
3. Se describen las tecnologías, herramientas y lenguajes empleados durante la implementación del sistema.
4. Se describe el proceso seguido para la implementación de la internacionalización de la aplicación.
5. Se describe el proceso seguido para la gestión de incidencias ocurridas en el sistema.
6. Detalle de las consideraciones necesarias para la implementación del acceso a datos.
7. Se describe el proceso seguido para la implementación del interfaz de usuario.
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 objetivo 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 dos fases:

- Implementación de la base de datos: creación de las tablas y las relaciones necesarias para el nuevo sistema en la base de datos, el conjunto de clases de acceso a datos correspondientes y sus respectivos servicios web.
- Implementación de los subsistemas identificados en el diseño, adecuándolos a la interfaz definida previamente.



### 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.
- ASP .NET. Plataforma de desarrollo de aplicaciones web, que se basa en ASP para la capa de presentación.
- JavaScript. Lenguaje interpretado normalmente embebido en páginas HTML y que puede ser entendido y ejecutado por la práctica totalidad de los navegadores Web modernos.
- 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 2010.
- IIS. Servidor de aplicaciones de Microsoft que permite gestionar servidores HTTP, HTTPS, FTP o SMTP entre otros. Se describe en el documento de Diseño en el apartado 3.6.
- Windows Communication Foundation. Marco de trabajo de Microsoft orientado para el trabajo con servicios web. Se describe en el documento de Diseño en el apartado 3.7.
- T-SQL. Lenguaje de acceso a datos basado en SQL. Está descrito en el apartado 6.3.6.1 del presente documento.
- AutoMapper. Librería utilizada para el mapeado entre objetos.
- Infragistics NetAdvantage 8. Librería para .NET que contiene controles para diversos entornos de desarrollo dentro de dicha plataforma, como ASP.NET, Winforms o WPF.
- Log4Net. Herramienta para ayudar en la generación de ficheros de registro.
- Navegadores: Microsoft Internet Explorer (versiones superiores a la 8.0), Mozilla Firefox y Google Chrome.



## 4. IMPLEMENTACIÓN DE LA INTERNACIONALIZACIÓN

### 4.1 Definición

Uno de los requisitos iniciales planteados para la AWSC 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 de la aplicación, sino que debe ser almacenado de forma externa y ser recuperable 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 la AWSC 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, como ya se ha comentado en el apartado anterior.
- 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:

```
lblName.Text = "Nombre";
```

- 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:

```
Index_lbl_lblUserName=Usuario
```

- 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 Index_lbl_lblUserName;
```

- Cargar en el código fuente los valores correspondientes del fichero de recursos compilado:



```
Assembly a = Assembly.Load("labels");
rm = new ResourceManager("labels", a);

if (rm != null)
{
    Index_lbl_lblUserName = rm.GetString("Index_lbl_lblUserName ");
}
}
```

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 *Load18N* carga en los campos de texto del formulario las variables de cadenas obtenidas de la siguiente forma:

```
btnRegister.Text = Index_btn_btnRegister;
```



## 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("DEBUG: Index.GridOffersList_RowCommand command=" + e.CommandName);  
  
Log.Write("DEBUG: Employee.Remove EmployeeID=" + EmployeeID.ToString());  
  
Log.Write("ERROR: Employee.Remove: EmployeeID=" + EmployeeID.ToString() + " " +  
e.ToString());
```





## 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, al que se accede a través de servicios web.

En este apartado se van a describir las principales características e implicaciones del uso conjunto de ambos sistemas en la implementación propiamente dicha de la AWSC.

### 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 sistema Operativos de Microsoft.

#### 6.2.1 T-SQL

Transact-SQL o T-SQL es una extensión a SQL con las siguientes características:

- Lenguaje de control de flujo. Instrucciones para permitir la ejecución de bloques condicionados e iterativos.  
`IF-THEN-ELSE, CASE-WHEN, WHILE, ...`
- Variables locales  
`DECLARE @Error int`  
Funciones nativas para procesamiento de cadenas, gestión de fechas, funciones matemáticas, etcétera.
- Mejoras en sentencias *DELETE* y *UPDATE*, para permitir uniones en la cláusula *FROM*.

#### 6.2.2 Transacciones

Una transacción es un conjunto de operaciones que debe ejecutarse como una sola unidad, según el principio ACID expuesto en el documento de Diseño.



En SQL Server, las sentencias SQL son tratadas como transacciones, y si ocurre cualquier tipo de incidencia durante su ejecución, el sistema es capaz de volver al estado anterior a su ejecución.

En ocasiones se requiere ejecutar un conjunto de sentencias SQL de forma transaccional, para lo que el lenguaje proporciona las sentencias de gestión de transacciones:

- *BEGIN TRAN*. Indica el comienzo de una transacción.
- *ROLLBACK TRAN*. deshace todos los cambios realizados por la transacción activa.
- *COMMIT TRAN*. Una vez que las operaciones de la transacción se han completado con éxito, se indica a la base de datos que vuelva a un estado consistente.

```
DECLARE @Error int
BEGIN TRANSACTION
INSERT INTO
NombreTabla
(
    Campo1,
    Campo2
)
VALUES
(
    @Valor1,
    @Valor2
)
SET @Error = @@ERROR
IF @Error != 0 GOTO ERROR_HANDLER

SELECT SCOPE_IDENTITY() AS 'Identity'
SET @Error = @@ERROR
IF @Error != 0 GOTO ERROR_HANDLER

COMMIT TRANSACTION
RETURN SCOPE_IDENTITY()
ERROR_HANDLER:
IF @@TRANCOUNT != 0 ROLLBACK TRANSACTION
RETURN @Error
```

### 6.2.3 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 los siguientes:

- Ejecución directamente en el motor de base de datos que suele estar en un servidor separado, evitando tráfico de datos entre el SGBD y la aplicación cliente.
- Posibilidad de almacenamiento del plan de ejecución por parte del SGBD, lo que repercute positivamente en la eficiencia de la consulta.
- Encapsula 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 pueden realizar en el código de la aplicación.



```

CREATE PROCEDURE [NombreProcedimiento]
(
    @parametro1 int,
    @parametro2 int
)
AS SELECT
    SUM(EnvPollLevelValue) AS IndicatorSUM
FROM
    NombreTabla1,
    NombreTabla2
WHERE
    <condición booleana>

```

#### 6.2.4 Desencadenadores

Los desencadenadores (o *triggers*) son, al igual que los procedimientos almacenados, un conjunto de sentencias T-SQL almacenadas en base de datos, pero con la particularidad de ser ejecutadas automáticamente cuando un usuario realiza una acción en la tabla de la base de datos que lleva asociado el desencadenador.

Su principal utilidad es el control de actualizaciones y borrados en cascada, que no pueden ser controlados por el SGBD debido a ciclos múltiples.

Se pueden crear desencadenadores para los siguientes tipos de instrucciones:

- *INSERT*: Inserción de información en la base de datos.
- *UPDATE*: Actualización de información de la base de datos.
- *DELETE*: Eliminación de información de la base de datos.

A continuación se muestra un ejemplo de un *trigger* que controla que al eliminar una entrada de la tabla 1, se elimine una entrada asociada a esa tabla, en la tabla 2.

```

CREATE TRIGGER [NombreTrigger] ON [NombreTabla1] FOR DELETE AS DELETE FROM NombreTabla2
WHERE <condición booleana>

```

### 6.3 Windows Communication Foundation

*Windows Communication Foundation* (WCF) es la tecnología utilizada para implementar los servicios web y pretende ser la solución al problema de mantener la seguridad de los datos. Los servicios web median entre la aplicación y la base de datos. Se ha implementado una clase de WCF con su respectiva interfaz por cada clase de acceso a datos utilizada en el sistema.

#### 6.3.1 Configuración de WCF

Para configurar los servicios web se ha utilizado el fichero de configuración “App.config”, donde se asigna a cada servicio una dirección y el contrato de seguridad para la comunicación. Se tiene en cuenta el acceso con HTTPS a los servicios web utilizando además seguridad a nivel de mensaje con credencial de usuario y contraseña que se valida en un método implementado para ello. La gestión de la seguridad se tratará con más detalle en su sección correspondiente. Se asigna también la base de datos a la que accederá el servicio.



A continuación, se muestra parte del contenido del fichero “App.config”, donde aparecen los servicios web utilizados:

```
<?xml version="1.0"?>
<configuration>
  <appSettings>
    <add key="conn" type="server=.\\SQL2012;database=Seleccion;uid=**;pwd=**" />
    <add key="userName" type="**" />
    <add key="password" type="**" />
  </appSettings>
  <system.serviceModel>
    <serviceHostingEnvironment multipleSiteBindingsEnabled="true"/>
    <bindings>
      <basicHttpBinding>
        <binding name="Security" ... >
          <security mode="TransportWithMessageCredential" >
            <message clientCredentialType="UserName"/>
          </security>
        </binding>
      </basicHttpBinding>
    </bindings>
    <services>
      <service name="WCF.administration.AdministrationItemWCF">
        <endpoint address="" binding="basicHttpBinding"
contract="WCF.administration.IAdministrationItemWCF" bindingConfiguration="Security">
        </endpoint>
        <endpoint address="mex" binding="mexHttpsBinding" contract="IMetadataExchange" />
        <host>
          <baseAddresses>
            <add
baseAddress="https://www.demoseleccion.endalia.com/WCF/WCF.administration.AdministrationIt
emWCF.svc" />
          </baseAddresses>
        </host>
      </service>
      <service name="WCF.orh.competence.competence.CompetenceWCF">
        <endpoint address="" binding="basicHttpBinding"
contract="WCF.orh.competence.competence.ICompetenceWCF" bindingConfiguration="Security">
        </endpoint>
        <endpoint address="mex" binding="mexHttpsBinding" contract="IMetadataExchange" />
        <host>
          <baseAddresses>
            <add baseAddress="https://www.demoseleccion.endalia.com/WCF/
WCF.orh.competence.competence.CompetenceWCF.svc" />
          </baseAddresses>
        </host>
      </service>
      <service name=" WCF.orh.recruitment.JobOfferWCF ">
        <endpoint address="" binding="basicHttpBinding"
contract=" WCF.orh.recruitment.IJobOfferWCF " bindingConfiguration="Security">
        </endpoint>
        <endpoint address="mex" binding="mexHttpsBinding" contract="IMetadataExchange" />
        <host>
          <baseAddresses>
            <add baseAddress="https://www.demoseleccion.endalia.com/WCF/
WCF.orh.recruitment.JobOfferWCF.svc" />
          </baseAddresses>
        </host>
      </service>
      <service name=" WCF.orh.recruitment.JobFileWCF ">
        <endpoint address="" binding="basicHttpBinding" contract="
WCF.orh.recruitment.IJobFileWCF " bindingConfiguration="Security">
        </endpoint>
        <endpoint address="mex" binding="mexHttpsBinding" contract="IMetadataExchange" />
        <host>
          <baseAddresses>
```



```

        <add baseAddress="https://www.demoseleccion.endalia.com/WCF/
WCF.orch.recruitment.JobFileWCF.svc" />
    </baseAddresses>
</host>
</service>
<service name="WCF.orch.recruitment.R_JobOffer_LocationWCF">
    <endpoint address="" binding="basicHttpBinding" contract="
WCF.orch.recruitment.IR_JobOffer_LocationWCF " bindingConfiguration="Security">
    </endpoint>
    <endpoint address="mex" binding="mexHttpsBinding" contract="IMetadataExchange" />
    <host>
        <baseAddresses>
            <add baseAddress="https://www.demoseleccion.endalia.com/WCF/
WCF.orch.recruitment.R_JobOffer_LocationWCF.svc" />
        </baseAddresses>
    </host>
</service>
<service name="WCF.orch.recruitment.R_JobOffer_EmployeeWCF">
    (...)
</service>
<service name="WCF.orch.recruitment.R_JobOffer_RequirementWCF">
    (...)
</service>
<service name="WCF.orch.job.JobWCF">
    (...)
</service>
<service name="WCF.orch.job.JobResponsabilityWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_CompetenceWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_QualificationWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_ExternalExperienceWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_ExternalTrainActionWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_GenericCourseWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_LanguageWCF">
    (...)
</service>
<service name="WCF.orch.job.R_Job_JobResponsabilityWCF">
    (...)
</service>
<service name="WCF.util.SystemOptionsWCF">
    (...)
</service>
<service name="WCF.util.MIMETYPEWCF">
    (...)
</service>
<service name="WCF.util.MailAddressInfoWCF">
    (...)
</service>
<service name="WCF.util.MailConstantWCF">
    (...)
</service>
<service name="WCF.util.MailSentMessageWCF">
    (...)
</service>
<service name="WCF.util.MailSentMessageAttachmentWCF">
    (...)
</service>

```



```

<service name="WCF.orh.user.UserWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeLanguageWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeExternalRecruitmentTmpWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeCurriculumNotApplicableWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeFilesWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeePhotoWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeReferenceWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_Employee_AreasOfInterestWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_Employee_GeographicAvailabilityWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_Employee_ExternalExperienceWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_EmployeeQualificationWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_EmployeeLanguageWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_Employee_DrivingLicenseWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_Employee_ExternalTrainActionsWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeRecruitmentDataWCF">
  (...)
</service>
<service name="WCF.orh.employee.EmployeeRecruitmentPreferenceWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_Employee_ExperienceAreasWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_EmployeeRecruitmentPreference_LocationsWCF">
  (...)
</service>
<service name="WCF.orh.employee.R_EmployeeRecruitmentPreference_CategoriesWCF">
  (...)
</service>
<service name="WCF.orh.training.TrainingGenericCourseWCF">
  (...)
</service>
<service name="WCF.orh.recruitment.JobOfferPersonalStageWCF">
  (...)
</service>
<service name="WCF.orh.recruitment.JobOfferStageWCF">
  (...)

```



```

</service>
</services>
<behaviors>
  <serviceBehaviors>
    <behavior>
      <serviceMetadata httpsGetEnabled="True" />
      <serviceDebug includeExceptionDetailInFaults="true" />
      <serviceCredentials>
        <userNameAuthentication
          userNamePasswordValidationMode="Custom"
          customUserNamePasswordValidatorType="WCF.CustomUserNameValidator, WCF"
        />
      </serviceCredentials>
    </behavior>
  </serviceBehaviors>
</behaviors>
</system.serviceModel>

(...)

</configuration>

```

### 6.3.2 Archivos de clases de WCF

Las clases de los servicios web están definidas por una interfaz y la clase que la implementa.

Por ejemplo, en la interfaz “IJobWCF.cs” se definen los métodos:

- Job[] GetJobList();
- Job Load(Job item);
- Job LoadNoGeneric(Job item);

Estos métodos se etiquetan con el atributo *OperationContract*, para indicar que estos métodos se definen en una clase que define un contrato de servicio en una aplicación WCF; mientras que la clase se etiqueta con *ServiceContract* indicando que define un contrato de servicio.

El archivo de la clase que implementa la interfaz es “JobWCF.cs”, y sus métodos se implementan así:

```

public Job[] GetJobList()
{
    JobCollection items = Job.GetJobList();

    if (items != null)
    {
        return (Job[])items.ToArray(typeof(Job));
    }

    return null;
}

public Job Load(Job item)
{
    item.Load();
    return item;
}

```



```

public Job LoadNoGeneric(Job item)
{
    item.LoadNoGeneric();
    return item;
}

```

En estos métodos se hacen referencia a métodos de las clases de la capa de acceso a datos que veremos más adelante.

### 6.3.3 Seguridad de WCF

El acceso a los servicios web se realiza a través de HTTPS, proporcionando seguridad en el transporte; mientras que el mensaje va protegido con credenciales de usuario y contraseña que se indican en el fichero de configuración. En este mismo fichero, se indica la clase que contiene el método que realizará la validación del usuario y contraseña, la clase “*CustomUserNameValidator*” que hereda de la clase “*UserNamePasswordValidator*” la posibilidad de sobrescribir el método “*Validate*”. El usuario y contraseña van además encriptados con el método *Rijndael*, también conocido como AES, utilizado en Endalia:

```

public override void Validate(string userName, string password)
{
    var validateUserName =
ConfigurationManager.AppSettings[USERNAME_ELEMENT_NAME];

    var validatePassword =
ConfigurationManager.AppSettings[PASSWORD_ELEMENT_NAME];

    validateUserName = RijndaelMethod.Decrypt(validateUserName,
RijndaelMethod.PK_DES);

    validatePassword = RijndaelMethod.Decrypt(validatePassword,
RijndaelMethod.PK_DES);

    if (string.IsNullOrEmpty(userName) || string.IsNullOrEmpty(password))
    {
        throw new FaultException("UserName and Password is required");
    }

    var validateCondition =
        userName.Equals(validateUserName) && password.Equals(validatePassword);
    if (!(userName.Equals(validateUserName) && password.Equals(validatePassword)))
    {
        throw new FaultException("Invalid UserName or Password");
    }
}

```





## 6.4 Acceso a datos

Las clases de acceso a datos están asociados a cada una de las tablas de la base de datos y contiene diversos métodos que pueden ser de actualización, consulta, eliminación o inserción de datos.

En este trabajo se han reutilizado algunas clases de acceso a datos ya implementadas anteriormente en el software de Endalia, pero realizando modificaciones en los métodos utilizados en la AWS para adaptarlos a los servicios web implementados.

### 6.4.1 Configuración para acceder a los servicios web

Se ha utilizado el fichero de configuración “app.config” para definir los servicios web utilizados y la dirección donde se encuentran alojados. En este fichero se define además el mismo contrato de seguridad que en la configuración de los servicios web y del mismo modo el usuario y su contraseña.

A continuación, se muestra la parte del fichero donde se indican los servicios web utilizados:

```
<?xml version="1.0"?>
<configuration>
  (...)
  <system.serviceModel>
    (...)
    <client>
      <endpoint
address="https://www.demoseleccion.endalia.com/WCF/WCF.administration.AdministrationItemWC
F.svc" binding="basicHttpBinding" contract="
administration.AdministrationItemWCF.IAdministrationItemWCF "
bindingConfiguration="Security" name=" BasicHttpBinding_IAdministrationItemWCF">
      </endpoint>
      (...)
    </client>
  </system.serviceModel>

  (...)
</configuration>
```

### 6.4.2 Clases de acceso a datos

Existe una clase de acceso a datos para su correspondiente servicio web. Si los métodos de la clase son utilizados desde el servicio web, se realiza el acceso a la base de datos; mientras que si son referenciados desde la capa de presentación utilizan los servicios web.

Como ejemplo se muestra el contenido del método *GetJobList()*, nombrado anteriormente en este documento:

```
if (WCFUtils.IsWebServiceAccess())
{
    IntegraDataAccess.orh.job.JobWCF.JobWCFClient client = generateClient();
    try
    {
        IntegraDataAccess.orh.job.JobWCF.Job[] toMap = client.GetJobList();
        Job[] mapped = WCFGenerics.FromWCFArray(toMap);
    }
}
```



```

        JobCollection collection = new JobCollection();

        collection.AddRange(mapped);

        Log.Write("DEBUG: Job.GetJobList");

        client.Close();

        return collection;
    }

    catch (Exception e)
    {
        client.Abort();

        MailSend.SendMailErrorOnCatchOnDataAccess(e.Message);

        Log.Write("ERROR: Job.GetJobList:" + e.ToString());

        return (null);
    }
}

else
{
    // acceso a base de datos
}

```

En cada método se genera un objeto cliente que accede al servicio web, la generación de este cliente se muestra a continuación:

```

IntegraDataAccess.orh.job.JobWCF.JobWCFClient client = new
IntegraDataAccess.orh.job.JobWCF.JobWCFClient();

var validateUserName =
RijndaelMethod.Decrypt(ConfigurationManager.AppSettings[WCFUtils.USERNAME_ELEMENT_NAME],
RijndaelMethod.PK_DES);

var validatePassword =
RijndaelMethod.Decrypt(ConfigurationManager.AppSettings[WCFUtils.PASSWORD_ELEMENT_NAME],
RijndaelMethod.PK_DES);

client.ClientCredentials.UserName.UserName = validateUserName;

client.ClientCredentials.UserName.Password = validatePassword;

return client;

```

Estas clases se etiquetan con el atributo *DataContract*, que especifica que se implementa un contrato de datos que puede hacer que el tipo sea serializable. Además las propiedades de estas clases se etiquetan con el atributo *DataMember*, especificando que pertenecen a un tipo de contrato de datos y lo pueden serializar.



## 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.

La interfaz de usuario tendrá en cuenta el diseño ya existente en la web corporativa de la organización, que por motivos de confidencialidad no se especificará.

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, llamada *tooltip*.

Los iconos de acceso adecuadamente escogidos facilitan a los usuarios encontrar las diferentes secciones del sistema con mayor rapidez.

#### 7.2.2 Tablas

Se han utilizado tablas siempre que ha sido necesario mostrar listado de datos. El texto de la cabecera destaca sobre el resto haciéndolo en negrita, y las filas alternas deben tener colores de fondo alternos para facilitar la legibilidad de las tablas.

#### 7.2.3 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.



## 7.2.4 Colores y fuentes

Los colores a utilizar vienen determinados por los colores corporativos de la organización a la que va destinado el proyecto de la AWSC; del mismo modo que ocurre con las fuentes y estilos.

## 7.2.5 Estilos

Todos los estilos CSS del sistema deben estar registrados en archivos CSS u hojas de estilo, facilitando al máximo las futuras modificaciones de estilo y permitiendo su adaptación a otras organizaciones. En esta hoja de estilos también deben de estar incluidos los colores y fuentes comentados en el punto anterior.

# 7.3 Pantallas del sistema

A continuación, se presenta el diseño definitivo de las principales interfaces del sistema; ya que por el tamaño del mismo, entrar en detalle en cada una de las pantallas existente sería excesivo para este documento.

En las primeras capturas se mostrarán la cabecera y pie de página difuminados, y en las siguientes se mostrará solo el contenido central junto al menú lateral.

## 7.3.1 Página de inicio

Esta es la página de inicio del sistema, donde pueden verse las vacantes disponibles, acceder a la sección privada o pasar a registrarse. Tiene las estructuras comunes descritas en el documento de diseño.

Figura 1: Página de inicio



### 7.3.2 Página de registro

Esta es la página de registro en el sistema, también se puede acceder desde ella a la sección privada.

**Acceso candidatos**

Usuario  
Contraseña  
**Acceder**  
☐ He olvidado mi contraseña  
[Ir a ver ofertas](#)

**Registro de nuevos usuarios**

¿Aún no te has registrado en nuestra bolsa de empleo?  
Regístrate aquí y podrás:

- ✓ Introducir tu currículum
- ✓ Apuntarte a los procesos de selección
- ✓ Recibir ofertas en tu e-mail

La dirección de correo electrónico que introduzcas será tu nombre de usuario en el sistema.  
Un correo electrónico con los datos para activar tu usuario será enviado a esta dirección una vez completado el registro.

E-mail:\*  
Nombre:\*  
Apellidos:\*  
Fecha nac.:\*  
Teléfono:  
Contraseña:\*  
Confirma contraseña:\*  
Solicitante de empleo ☐  
Solicitante de prácticas remuneradas ☐

[Política de privacidad](#)

☐ He leído y acepto la política de privacidad\*  
(\*) Campos obligatorios

**Finalizar registro**

Figura 2: Página de registro de usuario

### 7.3.3 Página de candidato

Similar a la pantalla de inicio pero con el menú lateral con las opciones del candidato y la posibilidad de inscribirse en ofertas.

test@endalia.com | [Desconectar](#)

**Mi cuenta**

- ☒ Mi currículum
- ☒ Mis ofertas (2)
- ☒ Datos de usuario

**Ofertas de empleo**

¿Quieres trabajar en...?  
En esta sección se publican nuestras ofertas de empleo.

[Lista de puestos vacantes \(2 puestos encontrados\)](#)

**Administrativo área de formación** Hace 2 meses

Se precisa administrativo para el área de formación de... para el envío de comunicados, confirmación de participantes, actualización de sistemas, tabulación de encuestas y reporting.

[Ver en detalle](#) **Darse de baja**

**Beca en el área de comunicación** Hace 3 meses

... ofrece una beca para la colaboración en la elaboración de convocatoria y notas de prensa, búsqueda, recopilación y gestión de información corporativa y seguimiento de alerta de noticias

[Ver en detalle](#) **Inscribirse**

Figura 3: Página de candidato



### 7.3.4 Página del currículum del candidato

A continuación se muestra el menú lateral que aparece en la página del currículum del candidato y las distintas secciones que posee:



Figura 4: Menú lateral en currículum

test@endalia.com | Desconectar

#### Datos personales

---

##### Información personal

Nombre:\*

Apellidos:\*

DNI:\*

Nacionalidad:\*

Fecha nacimiento:\*

Sexo:\*

 (Haz clic sobre la foto para editar)

---

##### Información de contacto

Email:

País:\*

Provincia:\*

Población:\*

Dirección:

Código postal:\*

Teléfono fijo:\*

Teléfono móvil:

Web/blog:

---

##### Información adicional

☐ Trabajo actualmente

☒ Disponibilidad para viajar

☒ Vehículo propio

☒ Carné de conducir

Figura 5: Página de datos personales del currículum

## Fotografía

test@endalia.com | [Desconectar](#)

Selecciona la imagen en el selector de fichero situado debajo de la foto. Puedes previsualizarla y compararla con la imagen actual pulsando sobre visualizar. Pulsando sobre el botón guardar se guarda la imagen del último fichero seleccionado, o de la última imagen previsualizada. (Tamaño máximo 500KB)

Foto actual



Selecciona el fichero con la nueva foto  Ningún archivo seleccionado

Visualizar

Guardar

Figura 6: Página de fotografía del currículo

## Formación

test@endalia.com | [Desconectar](#)

En esta sección puedes introducir la formación que poseas. Si no deseas incluir ningún dato, marca la siguiente opción para dar por completada esta sección.

### Titulaciones

☐ No aplicable en mi caso. Dar esta información por completada.

Introduce las titulaciones académicas que posees.

Fecha inicio	Fecha fin	Titulación	Centro	Estado		
01/09/2005	30/09/2009	MASTER	Master RRHH	Reconocido		
16/09/1998	01/09/2002	RELACIONES LABORALES	Universidad de Zaragoza	Reconocido		
01/09/1989	30/06/1998	BACHILLERATO	Esc. olapios	Reconocido		

Añadir titulación

### Cursos de formación

☐ No aplicable en mi caso. Dar esta información por completada.

Introduce los cursos de formación que has realizado. Nombre del curso, el centro formador, el área temática del curso, fecha de finalización del curso, su duración en horas y los detalles del curso.

Fecha fin	Curso	Área	Horas		
30/09/2004	Experto en Gestión de Salarios y Recursos Sociales	Gestión	200		

Añadir cursos

Figura 7: Página de formación del currículo



## Experiencia

test@endalia.com | [Desconectar](#)

En esta sección puedes introducir la experiencia que poseas. Si no deseas incluir ninguna, puedes marcar más abajo la opción para dar por completada esta sección.

### Experiencia profesional

☐ No aplicable en mi caso. Dar esta información por completada.

Introduce la experiencia profesional que poseas.

Fecha inicio	Fecha fin	Puesto	Empresa	Área		
01/09/2012	30/06/2014	Administrativo en el Depto. de contabilidad y fiscalidad	Gestoria CONFISA	Administration		
03/09/2010	01/08/2012	Gestor laboral	Endalia	Human Resources		

[Añadir experiencia](#)

Figura 8: Página de experiencia del currículum

## Idiomas

test@endalia.com | [Desconectar](#)

En esta sección puedes introducir los idiomas que conozcas. Si no deseas incluir ninguno, marca la siguiente opción para dar por completada esta sección.

### Idiomas

☐ No aplicable en mi caso. Dar esta información por completada.

Introduce el idioma y tu grado de dominio del mismo.

Idioma	Lectura	Escritura	Conversación		
ALEMÁN	3 - Medio	3 - Medio	3 - Medio		
FRANCÉS	3 - Medio	3 - Medio	3 - Medio		
INGLÉS	2 - Alto	2 - Alto	2 - Alto		

[Añadir idioma](#)

Figura 9: Página de idiomas del currículum

## Expectativas

test@endalia.com | [Desconectar](#)

En esta sección puedes introducir las expectativas de carrera en la organización. Si no deseas incluir ninguna, marca más abajo las opciones para dar por completada esta sección.

### Áreas de interés

☐ No aplicable en mi caso. Dar esta información por completada.

Selecciona el área de interés de nuestra organización que más se ajuste a tu perfil.

Área		
Social		
Educación		

[Añadir área](#)

Figura 10: Página de expectativas del currículum





## Referencias

En esta sección puedes introducir las referencias profesionales que poseas. Si no deseas incluir ninguna, puedes marcar más abajo la opción para dar por completada esta sección.

### Referencias profesionales

☐ No aplicable en mi caso. Dar esta información por completada.

Indica referencias de personas que pueden ser contactadas para informar sobre tí.

Nombre	Organización	Teléfono	Email		
Joaquín Álvarez Marquez	Endalia	976123356	test@endalia.com		

[Añadir referencia](#)

Figura 11: Página de referencias del currículo

## Ficheros

En esta sección puedes introducir ficheros que puedan aportarnos más detalles sobre tu perfil. Si no deseas incluir ningún fichero, puedes marcar más abajo la opción para dar por completada esta sección.

☐ No aplicable en mi caso. Dar esta información por completada.

Nombre	Descripción	Tamaño	
Ejemplo de Curriculum Vitae.pdf	Mi Curriculum	59 KB	

[Añadir fichero](#)

Figura 12: Página de ficheros del currículo

## Observaciones

En esta sección puedes introducir cualquier observación o comentario que consideres de relevancia. Si no deseas incluir ningún comentario, puedes marcar más abajo la opción para dar por completada esta sección.

☒ No aplicable en mi caso. Dar esta información por completada.

[Guardar](#)

Figura 13: Página de observaciones del currículo

## 8. BIBLIOGRAFÍA

### 8.1 Referencias

- [R1] I. Jacobson, G. Booch, J. Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software*. Pearson Education

### 8.2 Referencias web

- [W1] <http://www.uml.org>  
[W2] <http://www.wikipedia.org>  
[W3] <http://www.microsoft.com/net>  
[W4] <http://www.endalia.com>



# PRUEBAS

**SISTEMA DE RECLUTAMIENTO  
IMPLEMENTADO CON SERVICIOS WEB**

VERSIÓN 1.3  
PUBLICADO EL 12/02/2015

Copyright © 2014 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
19/09/2014	1.0	Redacción inicial del documento	Sergio Sanz Frías
15/11/2014	1.1	Modificaciones en el documento	Sergio Sanz Frías
09/02/2015	1.2	Revisión del documento	Fernando Cortés Franco
12/02/2015	1.3	Correcciones en el documento	Sergio Sanz Frías



# ÍNDICE

Histórico de revisiones .....	3
Índice .....	4
1. Introducción.....	5
1.1 Propósito del documento.....	5
1.2 Alcance del documento.....	5
1.3 Acrónimos.....	5
1.4 Definiciones .....	5
1.5 Referencias.....	5
1.6 Resumen.....	6
2. Descripción del proceso.....	7
3. Pruebas unitarias y de integración.....	8
4. Pruebas de sistema.....	9
5. Bibliografía.....	14
5.1 Referencias.....	14
5.2 Referencias web.....	14



# 1. INTRODUCCIÓN

## 1.1 Propósito del documento

El presente documento describe la fase de pruebas de la AWSC. 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 de la AWSC.

## 1.3 Acrónimos

- AWSC: Aplicación Web de Selección de Candidatos.
- PS: Prueba de Sistema.
- PUI: Prueba Unitaria y de Integración.

## 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 de la AWSC. Se compone de cinco apartados:

1. Introducción del documento, definición del propósito y alcance del mismo.
2. Se describe el proceso de diseño seguido para la confección de este documento.
3. Se describen las pruebas unitarias y de integración realizadas.
4. Se describen las pruebas de sistema y de aceptación realizadas.
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. Esta fase, no tiene por qué ser la fase final del trabajo; de hecho, desde esta fase se persigue obtener retroalimentación para la fase de implementación, con objeto de depurar el resultado de ésta. Se trata de un proceso circular.

En el caso de la AWSC, 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. 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 trabajo, el proyectando debe ser el responsable del proceso, y por ello ha desempeñado esta tarea.



### 3. PRUEBAS UNITARIAS Y DE INTEGRACIÓN

Las pruebas unitarias deben ser realizadas por el programador. Cada método que no sea trivial, debe ser probado individualmente y en el conjunto de su módulo. Como ya se ha comentado, cada uno de estos casos de uso se debe hacer a modo de caja negra.

Debido a la cantidad de casos de uso unitarios, y a que son muy parecidos entre si, se detalla a continuación uno de ellos, a modo de ejemplo.

Identificador	PUI – 01
Nombre	Modificación nombre de candidato
Descripción	Modificación del nombre del candidato desde la sección de CV del mismo.
Resultados esperados	Sustitución del nombre del candidato por el nuevo nombre.
Resultados obtenidos	Correcto

Las pruebas de integración, deben realizarse al finalizar cada módulo, para comprobar que todo funciona correctamente, en los diferentes módulos del sistema. Si los módulos tienen interfaz visual, debe comprobarse que su aspecto es adecuado con respecto al resto de la aplicación. Si el módulo tiene que interactuar con la base de datos, debe comprobarse su correcta comunicación con la misma. Si el módulo interactúa con servicios web, se comprueba que la conexión es correcta y segura.



## 4. PRUEBAS DE SISTEMA

Las pruebas de sistema tienen por objeto verificar el correcto funcionamiento de la aplicación en su conjunto, y comprobar que se cumplen los requisitos especificados.

Además de los casos de prueba que se detallan a continuación, se han tenido en cuenta las siguientes consideraciones:

- Rendimiento: se ha comprobado que el sistema se comporta de forma aceptable con diversas cargas de trabajo.
- Integración con el entorno: se ha verificado que cada módulo por separado, y el sistema en su conjunto son coherentes tanto en interfaz como en sensación visual con el resto de la plataforma donde está contenido.
- Seguridad: se ha comprobado que el sistema de control de usuarios cumple con las especificaciones definidas en la captura de requisitos.
- Facilidad de uso: se ha evaluado la interfaz del sistema de modo que el usuario final la perciba de la forma más intuitiva posible.

Identificador	PS – 01
Nombre	Registro de nuevo candidato
Descripción	Registro en el sistema de un nuevo candidato.
Resultados esperados	Debe comprobar que se introducen todos los datos obligatorios con valores correctos. En caso contrario se informa al usuario, y si está todo bien: <ul style="list-style-type: none"><li>• Crea un usuario temporal</li><li>• Envía mail al candidato para que confirme el registro</li></ul>
Resultados obtenidos	Correcto

Identificador	PS – 02
Nombre	Confirmación de registro de nuevo candidato
Descripción	El usuario confirma su registro en el sistema a través del correo electrónico que se le ha enviado y se le redirecciona a su sección privada.
Resultados esperados	Debe comprobarse que al acceder al enlace del correo electrónico el usuario accede correctamente a su sección privada.
Resultados obtenidos	Correcto



Identificador	PS – 03
Nombre	CV del candidato
Descripción	Comprobar la carga correcta de todos los datos del candidato, su modificación y eliminación.
Resultados esperados	<p>Debe comprobarse la visualización de todos los datos del candidato, pasando por cada una de las pestañas; comprobando que los datos se guardan correctamente, y se informa de datos incorrectos o inválidos.</p> <p>Debe comprobarse también la funcionalidad correcta de los controles de fotografía y subida de ficheros en varios navegadores sin que se produzcan incompatibilidades.</p>
Resultados obtenidos	Correcto

Identificador	PS – 04
Nombre	Visualización de ofertas
Descripción	Comprobar la visualización de las ofertas.
Resultados esperados	<p>Debe comprobarse la visualización correcta del listado de ofertas junto a su descripción.</p> <p>Del mismo modo debe accederse a cada oferta para verificar que su información se muestra correctamente.</p>
Resultados obtenidos	Correcto

Identificador	PS – 05
Nombre	Inscripción en ofertas
Descripción	Comprobar que el usuario se inscribe correctamente en una oferta.
Resultados esperados	Un mensaje en la pantalla informa al usuario que se ha inscrito en la oferta.



La oferta aparece en la sección 'Mis ofertas' y el usuario puede darse de baja de la misma.

También se envía un correo electrónico confirmando esta inscripción.

Resultados obtenidos	Correcto
Identificador	PS – 06
Nombre	Darse de baja de una oferta
Descripción	Comprobar que el usuario se da de baja correctamente de una oferta.
Resultados esperados	Un mensaje en la pantalla informa al usuario que se ha dado de baja de la oferta. La oferta ya no aparece en la sección de 'Mis ofertas' y puede volver a inscribirse.
Resultados obtenidos	Correcto

Identificador	PS – 07
Nombre	Obtener informe de una oferta
Descripción	Comprobar que el informe se genera correctamente.
Resultados esperados	El usuario podrá descargar el informe y en él aparece la información de la oferta igual que en la página de detalles de la misma.
Resultados obtenidos	Correcto

Identificador	PS – 08
Nombre	Acceso al sistema con usuario y contraseña



Descripción	Comprobar que los datos de acceso son correctos, y de ser así se redirecciona a la página principal correspondiente.
Resultados esperados	Si los datos son correctos se redirecciona a la sección privada. Informa correctamente si los datos no son correctos.
Resultados obtenidos	Correcto

Identificador	PS – 09
Nombre	Darse de baja del sistema
Descripción	Comprobar que el usuario tras darse de baja no puede volver a acceder al sistema.
Resultados esperados	El usuario solicita la baja y se le redirecciona a la página principal del sistema, desde la que ya no puede volver a acceder a la sección privada.  El usuario recibe un correo electrónico confirmando la eliminación de sus datos.
Resultados obtenidos	Correcto

Identificador	PS – 10
Nombre	Publicación de ofertas desde el sistema software de Endalia
Descripción	Comprobar que las ofertas publicadas desde otro sistema aparecen en la AWSC.
Resultados esperados	Tras publicar una oferta de una vacante desde el software de gestión desarrollado por Endalia, se comprueba que está misma oferta aparece en la aplicación web.
Resultados obtenidos	Correcto



Identificador	PS – 11
Nombre	Visualización de candidatos desde el sistema software de Endalia
Descripción	Comprobar que los candidatos de la AWSC aparecen en el sistema software de gestión de Endalia.
Resultados esperados	Tras registrarse en la AWSC, se consultan los datos del candidato desde el software de gestión desarrollado por Endalia. Se comprueban también las ofertas en las que está inscrito el candidato.
Resultados obtenidos	Correcto



## 5. BIBLIOGRAFÍA

### 5.1 Referencias

[R1] Elfriede Dustin 2002. *Effective Software Testing*. Ed. Addison Wesley.

### 5.2 Referencias web

[W1] <http://www.wikipedia.org>

[W2] <http://www.endalia.com>

