



**Universidad
Zaragoza**

Trabajo Fin de Grado

Desarrollo de un módulo de Business Intelligence

ANEXOS

Autor

Víctor Beltrán Piñol

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

2017



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e
Ingeniería de Sistemas
Universidad Zaragoza



ÍNDICE DE CONTENIDOS

- Estándar de documentación
- Estándar de codificación
- Plan de gestión de configuraciones
- Especificación de requisitos
- Análisis
- Diseño
- Implementación
- Pruebas
- Manual de usuario

ESTÁNDAR DE DOCUMENTACIÓN

DESARROLLO DE UN MÓDULO DE
BUSINESS INTELLIGENCE

VERSIÓN 2.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	2.0	Modificaciones del documento para adaptarlo al TFG	Víctor Beltrán Piñol



ÍNDICE

Histórico de revisiones.....	3
Índice	4
1. Introducción	5
1.1 Propósito del documento.....	5
1.2 Alcance del documento	5
1.3 Definiciones	5
1.4 Referencias	5
1.5 Resumen.....	5
2. Formato de documentación	6
2.1 Fuentes y estilos	6
2.2 Interlineado y formato de página	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 del módulo de Business Intelligence. 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 Definiciones

Fuente: un miembro de una familia de tipo de letra.

1.4 Referencias

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

1.5 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 14, Negrita.
- Título 2: Helvética Neue 12, Negrita.
- Título 3: Helvética Neue 10.
- Texto normal: Helvética Neue 9.
- 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 página

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 guion 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).

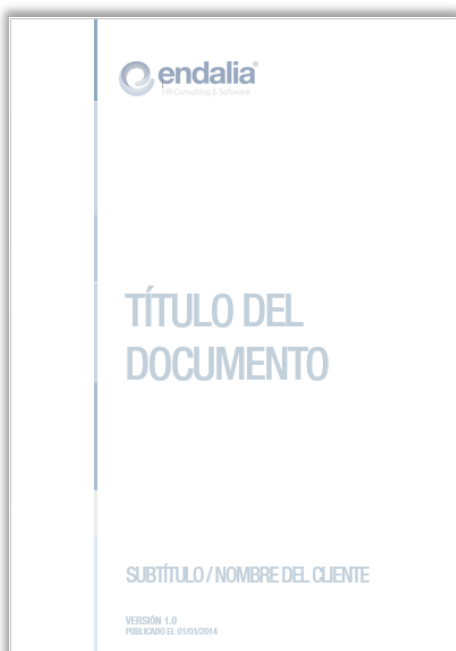


Figura 2. Hoja 1 – Portada

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).

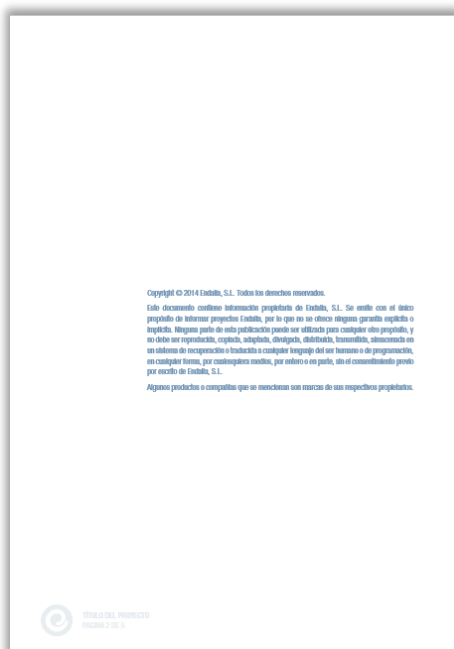


Figura 3. Hoja 2 - Información de copyright

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		

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 9.

INDICE	
Historico de revisiones.....	3
Índice.....	4
1. TÍTULO 1.....	5
1.1 TÍTULO 2.....	5
1.1.1 TÍTULO 3.....	5



TÍTULO DEL PROYECTO
PÁGINA 4 DE 12

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.

CONVOCATORIA DE LA SESIÓN

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

Fecha / hora _____

Lugar _____

Emissor _____

Fecha emisión _____

Asistentes _____

Distribución _____

Orden del día _____

 TÍTULO DEL PROYECTO (NOMBRE DEL CLIENTE)
PÁGINA 1 DE 2

Figura 6. Acta de reunión, hoja 1

ACTA DE LA SESIÓN

Sección 1

Acuerdo	Responsable	Compromiso	Estado
		01/01/2014	0%

Sección 2

Acuerdo	Responsable	Compromiso	Estado
		01/01/2014	0%


 TÍTULO DEL PROYECTO (NOMBRE DEL CLIENTE)
PÁGINA 2 DE 2

Figura 7. Acta de reunión, hoja 2

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

DESARROLLO DE UN MÓDULO DE BUSINESS
INTELLIGENCE

VERSIÓN 2.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	2.0	Modificaciones del documento para adaptarlo al TFG	Víctor Beltrán Piñol



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	7
1.6 Resumen	8
2. Estructura de los ficheros	9
2.1 Introducción	9
2.2 Codificación de archivos de interfaz de usuario .aspx.....	9
2.2.1 Declaración de página	9
2.2.2 Importación de controles	9
2.2.3 Definición de elementos de la página	9
2.3 Codificación de archivos de código subyacente .aspx.cs.....	10
2.3.1 Consideraciones generales sobre las regiones	11
2.3.2 Región directivas using.....	11
2.3.3 Región declaración namespace y class	11
2.3.4 Región declaración de variables.....	11
2.3.5 Región Page Load.....	11
2.3.6 Región I18N	12
2.3.7 Región DataBind.....	12
2.3.8 Región Eventos	12
2.3.9 Región Eventos de botones	12
2.3.10 Región Métodos de ordenación.....	12
2.3.11 Región Código generado por el diseñador de Web Forms.....	12
2.4 Codificación de archivos de clase, acceso a datos y servicios web .cs.....	12
2.4.1 Región directivas using.....	13
2.4.2 Región declaración namespace y class	13
2.4.3 Región atributos	13
2.4.4 Región propiedades.....	14
2.4.5 Región constructores	14
2.4.6 Región métodos	14



2.5	Codificación de archivos de recursos de internacionalización	14
3.	Reglas de codificación.....	15
3.1	Identación.....	15
3.1.1	Longitud de línea	15
3.1.2	Ruptura de líneas	15
3.2	Comentarios	15
3.2.1	Aplicación de los comentarios	16
3.2.2	Formatos de implementación de comentarios	17
3.3	Declaraciones	17
3.3.1	Número de declaraciones por línea	17
3.3.2	Inicialización.....	17
3.3.3	Situación	18
3.4	Sentencias.....	18
3.4.1	Sentencias simples.....	18
3.4.2	Sentencias compuestas	18
3.4.3	Sentencias de retorno	19
3.4.4	Sentencias if, if-else, if else- if else	19
3.4.5	Sentencias for	19
3.4.6	Sentencias while.....	20
3.4.7	Sentencias do-while.....	20
3.4.8	Sentencias switch	20
3.4.9	Sentencias try-catch	21
3.5	Espacios en blanco	21
3.5.1	Líneas en blanco	21
3.5.2	Espacios en blanco.....	21
3.6	Convenciones de nombres	22
3.6.1	Clases	22
3.6.2	Métodos	22
3.6.3	Variables y parámetros	22
3.6.4	Constantes	23
3.7	Hábitos de programación	23
3.7.1	Referencias a variables y métodos de clase	23
3.7.2	Constantes	23
3.7.3	Asignaciones de variables.....	23
3.7.4	Paréntesis	24
3.7.5	Variables de retorno.....	24



3.7.6	Expresiones antes de '?' en el operador condicional	24
3.7.7	Comentarios especiales.....	24
4.	Estándar de nombrado de Base de Datos.....	25
4.1	Idioma a utilizar.....	25
4.2	Convenciones de nombrado de tablas	25
4.2.1	Nombrado de tablas de entidad.....	25
4.2.2	Nombrado de tablas de relación	25
4.2.3	Nombres de tablas predefinidos	25
4.2.4	Nombrado de tablas de dimensión	26
4.2.5	Nombrado de tablas de hechos.....	26
4.3	Convenciones de nombrado de campos	26
4.3.1	Nombrado de Campos de Tablas de Entidad.....	26
4.3.2	Nombrado de campos de tablas de relación	27
5.	Bibliografía.....	28
5.1	Referencias	28
5.2	Referencias web	28



1. INTRODUCCIÓN

1.1 Propósito del documento

El presente documento describe las normas que deben seguirse en el desarrollo de cualquier tipo de elemento de codificación realizado en este 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 es importante para los programadores:

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

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

1.2 Alcance del documento

Este documento se ubica dentro de la fase inicial de desarrollo del módulo de Business Intelligence, 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. Se especifica la codificación tanto de los archivos de clases, acceso a datos y servicios web (.cs) como de los archivos de interfaz de usuario (.aspx) y los archivos de código subyacente (.aspx.cs).

2.2 Codificación de archivos de interfaz de usuario .aspx

A continuación se muestra la estructura de codificación de los archivos de interfaz de usuario. 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>Declaración de página	<pre><%@ Page Language="C#" AutoEventWireup="true" Inherits="myNamespace" MasterPageFile="myMaster" CodeBehind="myClass" %></pre>
<2>Importación de controles	<pre><%@ Register TagPrefix="myControlPrefix" TagName="myControl" Src="myControlSrc" %></pre>
<3>Definición de elementos de la página	<Código de definición de elementos HTML>

2.2.1 Declaración de página

En esta región se define la página o en su caso el control que representa el archivo. Es en esta región donde se enlaza la página con el código subyacente y con un fichero Master (en caso de definir una página y no un control).

2.2.2 Importación de controles

En esta región se definen los controles que se utilizarán a lo largo de la página. Es necesario referenciarlos mediante un prefijo y un nombre, así como indicar en qué ruta se encuentra su código fuente.

2.2.3 Definición de elementos de la página

En esta región se define la estructura de la página o del control que representa el archivo, mediante el uso de elementos tanto HTML como del framework de ASP.NET.



2.3 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. 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; <directivas using></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 I18N <declaraciones de variables de internacionalización> #endregion Variables I18N #region Variables globales <declaraciones de variables globales> #endregion Variables globales #endregion variables</pre>
<4> Región <i>Page_Load</i>	<pre> #region Page_Load <Código Page_Load> #endregion Page_load</pre>
<5> Región <i>I18N</i>	<pre> #region I18N <Código LoadI18N> <Código class_Init> #endregion I18N</pre>
<6> Región <i>DataBind</i>	<pre> #region DataBind <Código DataBind> #endregion DataBind</pre>
<7> Región Eventos	<pre> #region Eventos <Código Eventos provenientes de controles> #endregion Eventos</pre>
<8> Región Eventos de botones	<pre> #region Eventos de botones <Código Eventos clic botones> #endregion Eventos de botones</pre>



<9> Región Métodos de ordenación	<pre>#region Métodos de ordenación <Código Métodos de ordenación> #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>

2.3.1 Consideraciones generales sobre las regiones

Como se ha visto en el apartado [2.3](#) 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 [2.3](#) 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.

2.3.2 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.

2.3.3 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
/// </summary>
```

2.3.4 Región declaración de variables

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

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

2.3.5 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.



2.3.6 Región */18N*

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 *Load/18N()*: carga en los campos de texto del aspx las variables de cadenas obtenidas en el método *Init*.

2.3.7 Región *DataBind*

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

2.3.8 Región Eventos

Se utiliza esta región para definir para cada uno de los métodos que capturan los diversos eventos de diversos controles y que se especifican en la tabla del apartado [2.2](#).

2.3.9 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.

2.3.10 Región Métodos de ordenación

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

2.3.11 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.

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



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

2.4.1 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.

2.4.2 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
/// </summary>
```

2.4.3 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 guion bajo de este modo:

```
private int _requestID;
```



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

2.4.4 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 RequestID
{
    get{ return _requestID; }
    set{ _requestID = value; }
}
```

2.4.5 Región constructores

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

2.4.6 Región métodos

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

2.5 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 guion bajo seguido de un prefijo que indicará la utilización de la etiqueta seguida de un guion bajo, siguiendo la siguiente convención:
 - etiqueta o campo de texto: `_lbl_`
 - etiqueta de *hyperlink*: `_lnk_`
 - texto de botón: `_btn_`
- En el caso de que la etiqueta sea un *tooltip*, se colocará a continuación el 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.

```
BusinessIntelligence_lbl_lblSectionHeader
```



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.

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



3.2.1 Aplicación de los comentarios

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

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

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

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

Los siguientes puntos son técnicas de comentarios recomendadas.

- Cuando se modifica el código, se mantienen siempre actualizados los comentarios circundantes.
- Evitar los comentarios recargados, como las líneas enteras de asteriscos. En su lugar se utilizan espacios para separar los comentarios y el código.
- Evitar rodear un bloque de comentarios con un marco tipográfico. Puede resultar agradable, pero es difícil de mantener.
- Antes de la implementación, quitar todos los comentarios temporales o innecesarios, para evitar cualquier confusión en la futura fase de mantenimiento.
- Si se necesita realizar comentarios para explicar una sección de código compleja, examinar el código para decidir si se debería volver a escribir. Siempre que sea posible, no documentar un código malo, sino volver a escribirlo. Aunque, por regla general, no debe sacrificarse el rendimiento para hacer un código más simple para el usuario, es indispensable un equilibrio entre rendimiento y mantenibilidad.
- Usar frases completas al escribir comentarios. Los comentarios deben aclarar el código, no añadirle ambigüedad.
- Ir comentando al mismo tiempo que se programa, porque probablemente no habrá tiempo de hacerlo más tarde. Por otro lado, aunque se tuviera oportunidad de revisar el código que se ha escrito, lo que parece obvio hoy es posible que seis semanas después no lo sea.
- Evitar comentarios superfluos o inapropiados, como comentarios divertidos al margen.
- Usar los comentarios para explicar el propósito del código como si fueran traducciones interlineales.
- Comentar cualquier cosa que no sea legible de forma obvia en el código.
- Para evitar problemas recurrentes, hacer siempre comentarios al depurar errores y solucionar problemas de codificación, especialmente cuando se trabaja 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 [2.4.4](#).

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 operadores. Por ejemplo:

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

- Las expresiones de una sentencia deberán estar separadas por espacio. 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 guion 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 guion 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_Emp_Employees
```

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_Jobs' y 'Orh_Emp_Employees' se creará la tabla 'R_Employee_Job'.

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



4.2.4 Nombrado de tablas de dimensión

Las tablas que hagan referencia a las dimensiones de un Data Warehouse se comenzarán con el prefijo 'Dim'

A continuación, y en plural se colocarán usando Pascal-Casing el nombre descriptivo de las tablas de entidad que relaciona, pudiendo reducirse si es muy largo alguno de ellos.

Ejemplos:

```
DimEmployees  
DimDates
```

4.2.5 Nombrado de tablas de hechos

Las tablas de hechos de un Data Warehouse se comenzarán con el prefijo 'FactTable'.

A continuación, se escribirá mediante un código de tres letras mayúsculas el Data Mart al cual hace referencia la tabla de hechos.

Ejemplo:

```
FactTableSAL  
FactTableORH
```

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_Jobs' todos los campos de la tabla en este punto empezarían por 'Job'. Otro ejemplo: Si la tabla es 'Orh_Emp_Employees' los campos de la tabla empezarían 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



PLAN DE GESTIÓN DE CONFIGURACIONES

DESARROLLO DE UN MÓDULO DE
BUSINESS INTELLIGENCE

VERSIÓN 2.0
PUBLICADO EL 12/05/2017

Copyright © 2016 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 Estándar de Documentación de Endalia	Endalia
12/05/2017	2.0	Modificaciones del documento para adaptarlo al TFG	Víctor Beltrán Piñol



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 Wayne A. Babich, una de las personas 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

- 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.

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 explicado en los anteriores 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.

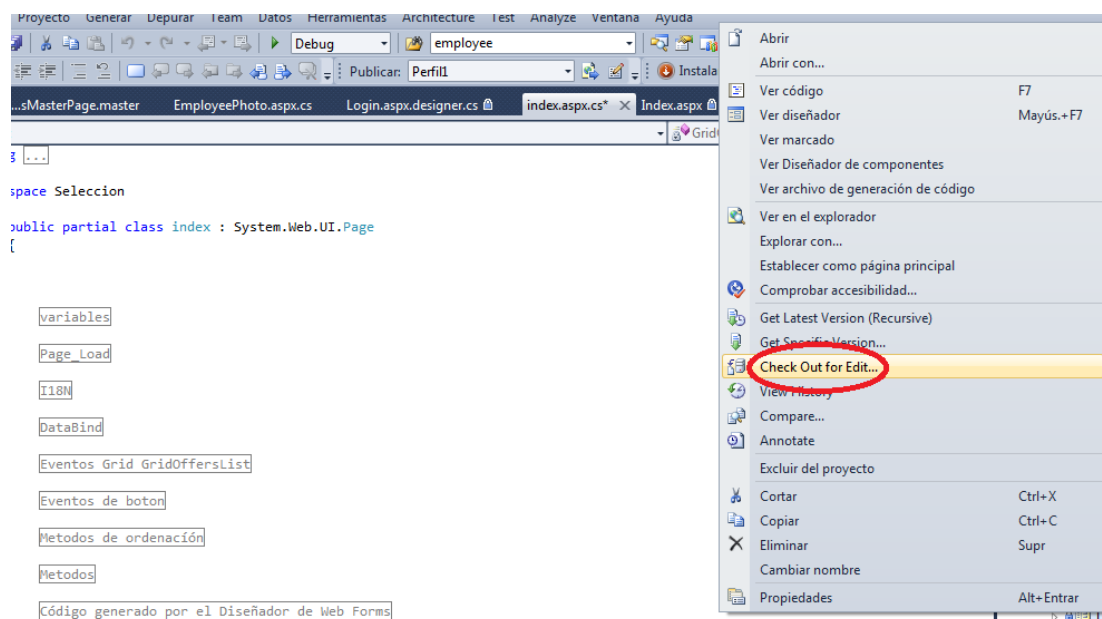


Figura 1. Bloqueo de ECS

- 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.



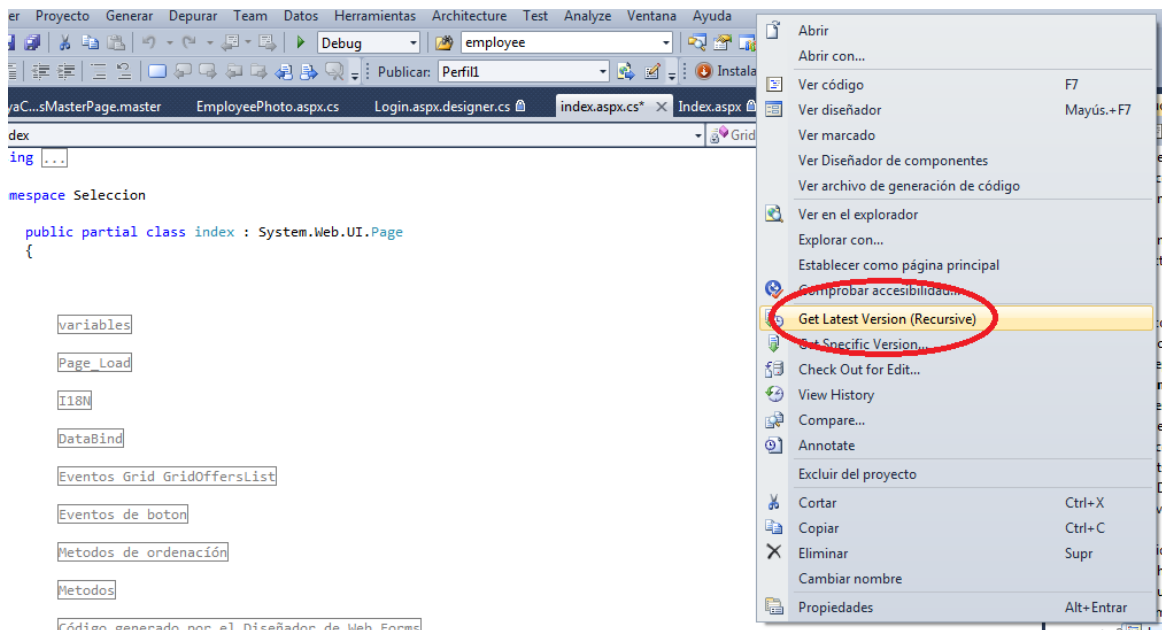


Figura 2. Descarga de versión

- 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.

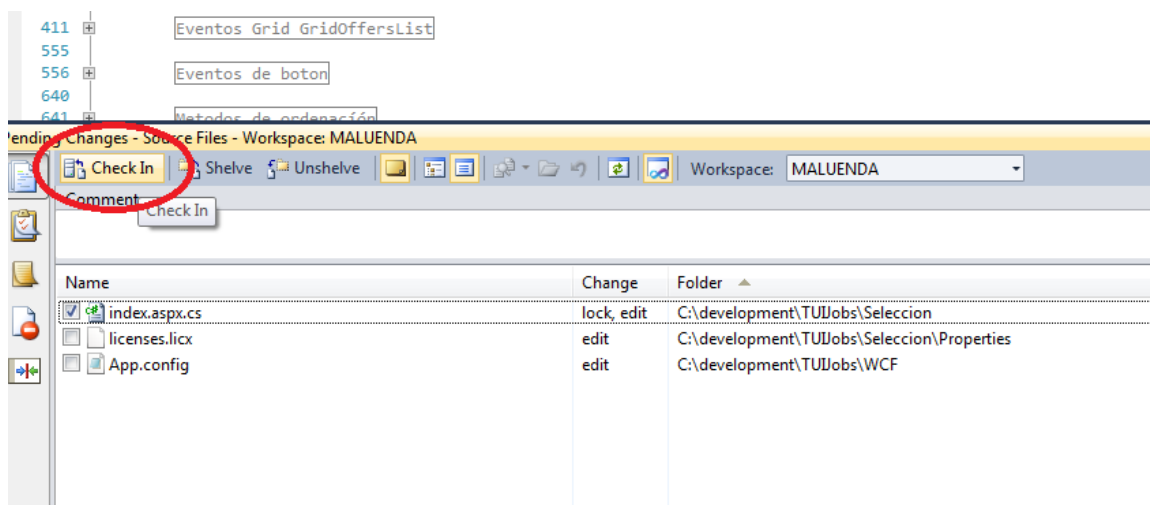


Figura 3. Modificación de ECS

- 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.



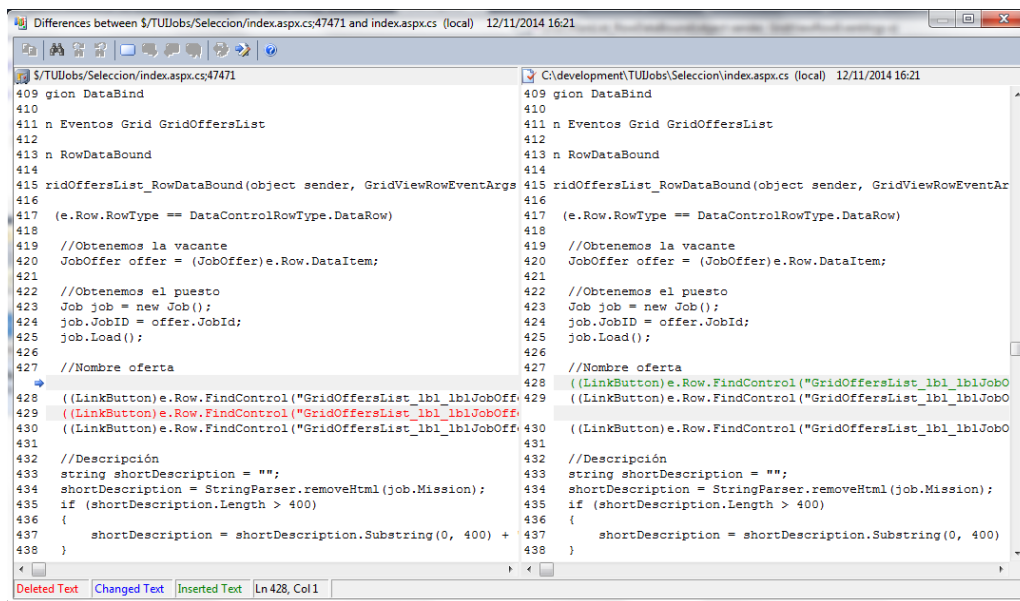


Figura 4. Resolución de conflictos

- 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.

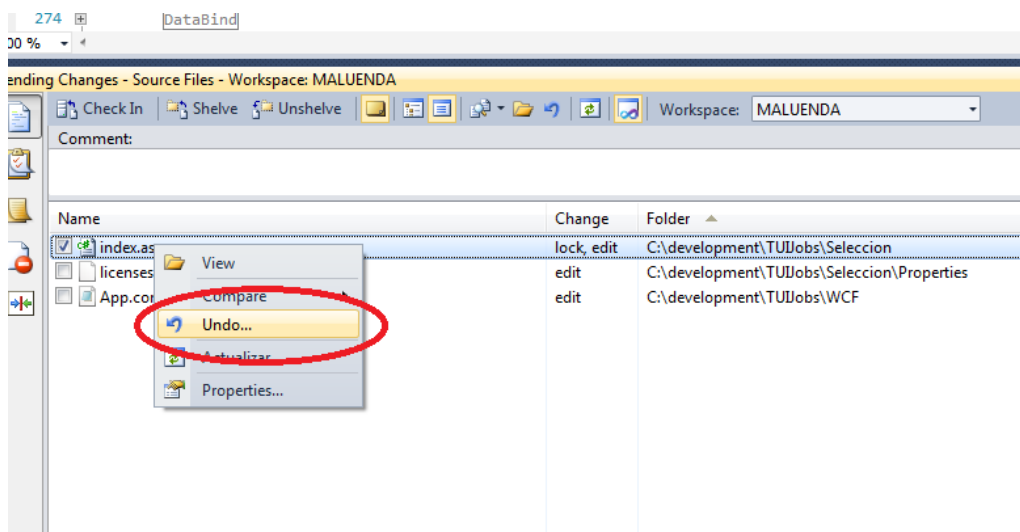


Figura 5. Descartar modificaciones

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>



ESPECIFICACIÓN DE REQUISITOS

DESARROLLO DE UN MÓDULO DE
BUSINESS INTELLIGENCE

VERSIÓN 1.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	1.0	Redacción inicial del documento	Víctor Beltrán Piñol



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 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.....	10
5. Bibliografía	12
5.1 Referencias	12



1. INTRODUCCIÓN

1.1 Propósito del documento

El presente documento pretende presentar y describir los requisitos del sistema a desarrollar. Estos requisitos se dividen en requisitos funcionales, que pretenden identificar la funcionalidad objetivo del sistema, y en requisitos no funcionales, que son relativos a la seguridad, interfaz, rendimiento o escalabilidad del sistema a desarrollar. Además, es esta especificación lo que guiará el resto del desarrollo software del proyecto.

Una buena especificación de requisitos debe ser [R1]:

- Correcta
- Inequívoca
- Completa
- Consistente
- Categorizada según su importancia
- Comprobable
- Modificable
- Identificable

1.2 Alcance del documento

Este documento describe el resultado de la fase de estudio de los requerimientos del cliente. A lo largo del mismo se describirán los distintos requisitos funcionales y no funcionales que el sistema desarrollado debe cumplir.

1.3 Acrónimos

- RF: Requisito funcional
- RNF: Requisito no funcional
- PDF: Portable Document Format
- GUI: Graphical User Interface
- KPI: Key Performance Indicator

1.4 Definiciones

- KPI: Un KPI (key performance indicator), conocido también como indicador clave o medidor de desempeño o indicador clave de rendimiento, es una medida del nivel del desempeño de un proceso.

1.5 Referencias

Este documento no contiene referencias a otros documentos del trabajo.



1.6 Resumen

El objetivo de este documento es el de presentar los resultados obtenidos del estudio de las necesidades del cliente, identificando los diferentes requisitos que el sistema diseñado debe cumplir. Se compone de los siguientes apartados:

1. Una introducción en la que se presenta el documento, su propósito y su alcance, así como una lista de acrónimos, definiciones y referencias que puedan ser de utilidad.
2. La descripción general, en la que se presenta la funcionalidad del sistema a desarrollar, así como el tipo de usuarios que lo utilizará.
3. Un listado detallado con los requisitos funcionales del sistema, su nivel de criticidad y sus dependencias.
4. Un listado detallado con los requisitos no funcionales del sistema, su nivel de criticidad y sus dependencias.
5. La bibliografía utilizada en esta fase del proyecto.



2. DESCRIPCIÓN GENERAL

2.1 Funciones generales del sistema

Este módulo de Business Intelligence es un sistema que permite realizar análisis con herramientas gráficas mediante la explotación de datos históricos de una empresa. Los usuarios podrán acceder a los informes a través del portal web de Endalia y poder filtrar estos datos por un periodo de tiempo, género, área, empresa, tipología de empleado, subtipología de empleado o rol.

2.2 Características de los usuarios

Existe un único tipo de usuario que podrá interactuar con el módulo. Este usuario será el responsable de Recursos Humanos, aunque desde la empresa se podrá dar acceso a otros usuarios en caso de considerarse oportuno.

Por último, se habrá de considerar que los usuarios no poseen una formación específica, por lo que es posible que no estén familiarizados con el uso de aplicaciones web.



3. REQUISITOS FUNCIONALES DEL SISTEMA

Nombre	RF – 01: Los informes deberán visualizarse desde Endalia web
Descripción	La aplicación Endalia web contará con una entrada en el menú para el acceso al módulo, y dentro del módulo una entrada para cada uno de los distintos informes.
Nivel de criticidad	Alto

Nombre	RF – 02: Los informes deberán ser interactivos
Descripción	Cada informe tendrá varios gráficos para el análisis en los cuales se podrán realizar filtros de datos sin tener que recargar la página. Los filtros deberán afectar a todos los gráficos de la página.
Nivel de criticidad	Alto

Nombre	RF – 03: Se deberán poder filtrar los datos por meses y años
Descripción	Los informes deberán permitir filtrar los datos por periodos de tiempo de mínimo un mes, máximo todos los años y meses disponibles.
Nivel de criticidad	Alto

Nombre	RF – 04: Se deberán poder filtrar los datos por las compañías y áreas del cliente
Descripción	Los informes deberán permitir filtrar los datos por las distintas compañías y áreas que forman la estructura organizativa de la empresa.
Nivel de criticidad	Alto



Nombre	RF – 05: Se deberán poder filtrar los datos por las distintas tipologías, subtipologías y roles de empleado
Descripción	Los informes deberán permitir filtrar los datos por las distintas tipologías de empleados, así como por las distintas subtipologías de los empleados y también por los distintos roles de los empleados.
Nivel de criticidad	Alto

Nombre	RF – 06: Se deberán poder obtener todos los KPI definidos entre Endalia y el cliente.
Descripción	El módulo deberá poder responder, mediante sus gráficos, todos los KPI definidos entre Endalia y el cliente.
Nivel de criticidad	Alto



4. REQUISITOS NO FUNCIONALES DEL SISTEMA

Nombre	RNF – 01: Eficiencia del sistema
Descripción	Toda funcionalidad del sistema y navegación entre las diferentes secciones del módulo debe responder al usuario en menos de 3 segundos.
Nivel de criticidad	Alto

Nombre	RNF – 02: Integración del módulo en la aplicación de Endalia
Descripción	El sistema deberá integrarse en el conjunto del Sistema de Gestión y Organización de Recursos Humanos de la entidad, conviviendo con el resto de módulo como uno más del conjunto global. Esto implica adecuarse al estilo, diseño y funcionalidad del resto de la aplicación.
Nivel de criticidad	Alto

Nombre	RNF – 03: Aplicación parametrizable
Descripción	El módulo será fácilmente parametrizable en función del cliente. Todos los clientes ejecutarán el mismo código, pero los informes que visualizarán estarán definidos por campos de una tabla de configuración de base de datos
Nivel de criticidad	Alto

Nombre	RNF – 04: Acceso mediante privilegios
Descripción	El acceso al módulo deberá ser controlado por privilegios que se podrán editar desde la aplicación de escritorio de la plataforma Endalia.
Nivel de criticidad	Alto



Nombre	RNF – 05: Interacción con el usuario mediante GUI consistente
Descripción	La interacción con el usuario deberá realizarse a través de una GUI consistente con la interfaz ya desarrollada en el resto de la aplicación. Es decir, deberá respetar el diseño de elementos como tablas, botones o colores, así como la distribución habitual de las pantallas.

Nivel de criticidad	Alto
---------------------	------

Nombre	RNF – 06: Aplicación intuitiva
Descripción	La interacción de los usuarios con los distintos informes del módulo deberá ser fácil e intuitiva, incluso para aquellos usuarios poco familiarizados con este tipo de aplicaciones.

Nivel de criticidad	Medio
---------------------	-------

Nombre	RNF – 07: Disponibilidad del sistema
Descripción	El módulo deberá ser accesible las 24 horas del día y los 7 días de la semana, adecuándose a la disponibilidad del sistema global de la entidad. Además, deberá permitir el acceso concurrente a múltiples usuarios sin que esto conlleve una pérdida de rendimiento o inconsistencias en los datos.

Nivel de criticidad	Medio
---------------------	-------

Nombre	RNF – 08: Resolución
Descripción	El módulo deberá ser correctamente visualizado en pantallas con resoluciones iguales o superiores a 1024x768, adecuándose a los estándares del resto del sistema de la organización.

Nivel de criticidad	Medio
---------------------	-------

Nombre	RNF – 09: Compatibilidad con distintos navegadores
Descripción	El módulo deberá funcionar como mínimo en los siguientes navegadores: Internet Explorer (versiones superiores a la 8.0), Mozilla Firefox y Google Chrome, adecuándose a los estándares de la aplicación de Endalia.

Nivel de criticidad	Medio
---------------------	-------



5. BIBLIOGRAFÍA

5.1 Referencias

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



ANÁLISIS

DESARROLLO DE UN MÓDULO DE BUSINESS INTELLIGENCE

VERSIÓN 1.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	1.0	Redacción inicial del documento	Víctor Beltrán Piñol



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. Descripción del proceso.....	7
3. Análisis deL Sistema	8
3.1 Introducción.....	8
3.2 Análisis del Data Warehouse	8
3.3 Actores de los casos de uso.....	9
3.4 Casos de uso	9
3.4.1 Diagrama de casos de uso.....	10
3.4.2 Descripción de los casos de uso	10
3.5 Análisis de la herramienta a usar para la integración de informes.....	11
4. Análisis de la infraestructura de aplicaciones	14
5. Bibliografía	15
5.1 Referencias web	15



1. INTRODUCCIÓN

1.1 Propósito del documento

En este documento se presenta la fase de análisis de este proyecto software. Partiendo de los requisitos descritos en el documento de especificación de requisitos se va a realizar un análisis del sistema a desarrollar, dividiendo este análisis en tres partes:

- El análisis del Data Warehouse.
- El análisis de la integración del módulo con el portal web de Endalia.
- El análisis de actores y casos de uso del módulo.

1.2 Alcance del documento

Este documento muestra los resultados de la fase de análisis del proyecto. En él, se describirá el proceso de análisis para a continuación realizar un estudio de los casos de uso y sus diferentes actores.

1.3 Acrónimos

- UML: Unified Modeling Language.
- KPI: Key Performance Indicator.

1.4 Definiciones

- Caso de uso: especificación de las secuencias de acciones que pueden ser efectuadas por un sistema, subsistema o clase por interacción con actores externos.
- Diagrama de casos de uso: notación gráfica que describe un caso de uso.
- UML: lenguaje de modelado de sistemas de software desarrollado por *Rational* y de uso extendido.
- KPI: Un KPI (key performance indicator), conocido también como indicador clave o medidor de desempeño o indicador clave de rendimiento, es una medida del nivel del desempeño de un proceso.

1.5 Referencias

En este documento se referencian los siguientes documentos del proyecto:

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

1.6 Resumen

En este documento se presenta la fase de análisis del trabajo. Se compone de los siguientes apartados:

1. Introducción, propósito y alcance del documento.
2. Descripción del proceso de análisis utilizado en este proyecto.
3. Análisis del sistema.
4. Análisis de la infraestructura de aplicaciones.
5. Presentación de la bibliografía utilizada para la realización de este documento.



2. DESCRIPCIÓN DEL PROCESO

El proceso de análisis del sistema se compone de distintas actividades, relacionadas con el almacén de datos, la integración de los informes con la aplicación y el análisis de casos de uso del módulo.

El análisis de los casos de uso pretende identificar los distintos actores que interactuarán con el módulo, y las acciones que podrán realizar dichos actores sobre la aplicación.

El análisis del Data Warehouse describe el conjunto de indicadores que son necesarios obtener a partir del Data Warehouse.

El análisis de la integración de informes con el portal web pretende identificar las ventajas e inconvenientes de las distintas alternativas acerca de la manera que se puede llevar a cabo la integración de los informes del módulo.



3. ANÁLISIS DEL SISTEMA

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.

3.2 Análisis del Data Warehouse

La primera parte del análisis del Data Warehouse a construir es determinar aquellos indicadores que podrán obtenerse a través de la realización de consultas sobre el almacén de datos.

A continuación se muestra la lista completa de indicadores (KPI) acordados entre Endalia y el primer cliente que integra el módulo dentro de su organización. Los indicadores se encuentran agrupados dentro de cada uno de los procesos de interés a analizar, es decir, cada uno de los Data Marts que conforman el Data Warehouse:

- Objetivos e indicadores generales:
 1. Número de personas en un periodo determinado.
 2. Número de personas en un periodo determinado por nivel (categoría y rol).
 3. Número de personas en un periodo determinado por tipo y subtipo.
- Salarios y nóminas:
 4. Costes salariales de plantilla mensuales por área.
 5. Coste medio real versus coste presupuestado.
 6. Costes salariales por tipo y subtipo de empleado respecto al coste salarial total
- Planificación de plantilla:
 7. Headcount y FTE por área, tipo y subtipo de empleado.
 8. Desviaciones de Headcount y FTE versus realidad.
- Organización:
 9. Tiempo medio de permanencia en el puesto por área.
- Desempeño:
 10. Porcentaje medio de desempeño por área.
 11. Porcentaje medio de alto potencial versus total de la plantilla por rol.
- Formación:
 12. Porcentaje y número de empleados que participan en formación.
- Dimensionamiento y personal no residente:
 13. Porcentaje de plantilla fija y temporal sobre el total.
 14. Personas incorporadas por país: extranjero UE y no UE.



15. Porcentaje de empleados extranjeros respecto al total de la plantilla.

- Rotación de plantilla:

16. Plantilla contratada en cada mes.

17. Número de bajas totales mensuales.

18. Número de bajas por área mensuales.

19. Del total de bajas porcentaje de bajas voluntarias y porcentaje de bajas no voluntarias.

20. Índice de rotación no deseada.

21. Índice de desvinculación no deseada.

Partiendo de la lista anterior, se localizan todas las tablas que contienen datos acerca de las variaciones de estado de los empleados que son necesarios para poder contestar a los indicadores definidos anteriormente.

Por cada uno de los grupos de indicadores anteriores, es necesario crear un data mart. Al final, todos estos data marts deberán ser integrados en el mismo almacén de datos

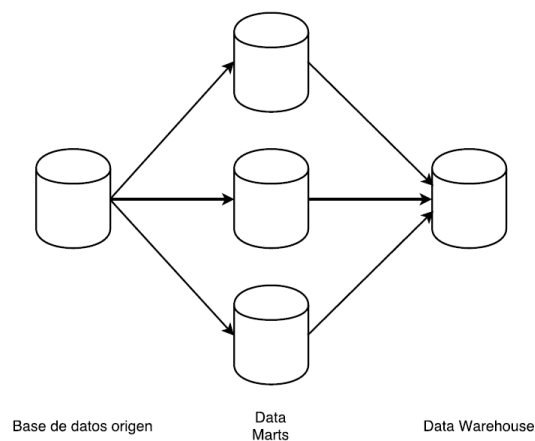


Figura 1. Flujo de creación del Data Warehouse

3.3 Actores de los casos de uso

En el ámbito de este proyecto, se denominan actores a los distintos usuarios que interaccionan con el módulo. Estos se distinguen un único tipo, que son los responsables de Recursos Humanos: Estos usuarios tienen acceso total al módulo, sin ningún tipo de restricción.

3.4 Casos de uso

En esta sección se especifican los casos de uso del módulo de Business Intelligence. Los casos de uso del módulo se representarán mediante un diagrama de casos de uso, además de una descripción de cada uno de estos casos de uso.

Primero se mostrará el diagrama completo con los casos de uso para obtener una visión completa de éstos y después se realizará una descripción detallada de cada caso de uso.



3.4.1 Diagrama de casos de uso

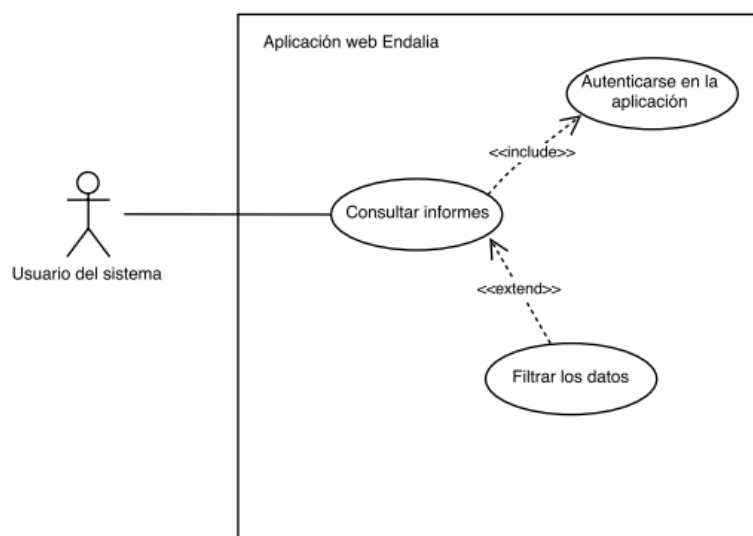


Figura 2. Diagrama de casos de uso

3.4.2 Descripción de los casos de uso

A continuación se describen con más detalle los casos de uso, analizando la secuencia de acciones que se llevan a cabo en el caso de uso.

Caso de uso	Autenticarse en la aplicación
Descripción	Este caso de uso permite a los usuarios acceder al portal web de Endalia.
Actores	Usuario del sistema
Flujo de eventos	<ol style="list-style-type: none">1. El usuario accede al portal web introduciendo la url correspondiente en el navegador2. El usuario introduce sus credenciales de acceso y es redirigido al home de la aplicación.
Flujo de eventos alternativo	<ol style="list-style-type: none">1. El usuario accede al portal web introduciendo la url correspondiente en el navegador2. El usuario introduce unas credenciales de acceso incorrectas y no pasa del login de la aplicación.

Caso de uso	Consultar informes
Descripción	Este caso de uso permite visualizar los informes del módulo de Business Intelligence a los usuarios.
Actores	Usuario del sistema

Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario accede al portal web introduciendo la url correspondiente en el navegador 2. El usuario introduce sus credenciales de acceso y es redirigido al home de la aplicación. 3. El usuario despliega la entrada del módulo de Business Intelligence del menú principal. 4. El usuario pulsa sobre cualquiera de los informes disponibles en el menú del módulo de Business Intelligence. 5. El usuario ya puede consultar el informe seleccionado
Flujo de eventos alternativo	<ol style="list-style-type: none"> 3. El usuario accede al portal web introduciendo la url correspondiente en el navegador 4. El usuario introduce unas credenciales de acceso incorrectas no pasa del login de la aplicación.

Caso de uso	Filtrar datos de los informes
Descripción	Este caso de uso permite visualizar los informes del módulo de Business Intelligence a los usuarios.
Actores	Usuario del sistema
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario accede al portal web introduciendo la url correspondiente en el navegador 2. El usuario introduce sus credenciales de acceso y es redirigido al home de la aplicación. 3. El usuario despliega la entrada del módulo de Business Intelligence del menú principal. 4. El usuario pulsa sobre cualquiera de los informes disponibles en el menú del módulo de Business Intelligence. 5. El usuario ya puede consultar el informe seleccionado. 6. El usuario cambia los valores de cualquiera de los filtros disponibles en el informe. 7. El usuario ya puede consultar el informe con los filtros aplicados.
Flujo de eventos alternativo	<ol style="list-style-type: none"> 5. El usuario accede al portal web introduciendo la url correspondiente en el navegador. 6. El usuario introduce unas credenciales de acceso incorrectas no pasa del login de la aplicación.

3.5 Análisis de la herramienta a usar para la integración de informes.

En esta fase del análisis se ha llevado a cabo un estudio de como se integrarían los informes en el módulo. Para escoger la herramienta se han estudiado las principales herramientas para crear informes interactivos y poder embeberlos en una aplicación web.



Para este estudio se han tenido en cuenta tres aplicaciones de terceros. Las tres herramientas de terceros analizadas eran las tres líderes del Cuadrante Mágico de Gartner de Business Intelligence y Business Analytics de 2017 [W2]. Este informe es el referente mundial en análisis de herramientas de Business Intelligence.

En este análisis de la consultora Gartner destacan principalmente las tres empresas situadas en el cuadrante de los líderes: Tableau, Qlik y Microsoft.

De entre estos tres líderes cabe destacar la evolución de Microsoft, que hasta el 2016 siempre ha estado por detrás de Tableau y Qlik, y año a año ha ido recortando distancias hasta situarse líder indiscutible en este año 2017.



Figura 3. Cuadrante Mágico de Gartner 2017

A continuación se describe el estudio de cada una de estas herramientas y porqué fue rechazada o la escogida:

- Tableau: Esta herramienta permite crear y embeber vistas y dashboards interactivos en una aplicación web, siempre que el usuario posea una cuenta en el servidor de Tableau que almacena los informes, Tableau Server. Los costes asociados en esta herramienta parten desde los 12.500 USD / año, divididos en las siguientes licencias:
 - Tableau Server. Coste :10.000 USD / año. Esta licencia ofrece acceso a 10 usuarios. El coste por usuario extra es de 500 USD / año
 - Tableau Desktop Professional Edition. Coste: 2.000 USD / año. Es necesaria la licencia profesional y no la básica debido a que el origen de datos se encuentra en una base de datos de SQL Server la cual no es compatible con las versión básica.
 - Tableau Online. Coste: 500 USD / año .

Finalmente se descarta esta opción por coste no asumible por parte del cliente.

- Qlik: Esta herramienta permite crear y embeber informes interactivos en una aplicación web, siempre que el usuario que acceda tenga cuenta en el servidor de Qlik donde se almacenan los informes. También permite el consume de informes en web a través de Mashups de Qlik, que son páginas web desarrolladas con el motor de Qlik. Los costes asociados a esta herramienta parten desde un mínimo de 8760 € el primer año, divididos en las siguientes licencias:
 - Licencia de desarrollo. Coste: 5000 € / año. Esta licencia incluye el software necesario para el desarrollo de informes y el software necesario para crear un servidor de informes de Qlik.
 - Licencia de producción. Coste: 33.5 € / mes por usuario. Se requiere un mínimo de 5 usuarios como punto de partida.
 - Training mínimo acerca del uso de la herramienta de cinco días en Madrid o Barcelona. Este training es exigido por parte de Qlik y necesario para poder obtener las licencias anteriores. Su coste es de 1.750 € por persona.

Finalmente se descarta esta opción por coste no asumible por parte del cliente.

- Power BI: Esta herramienta forma parte del paquete Office de Microsoft. Permite crear informes interactivos que admiten bases de datos SQL Server como origen de datos. El uso de la herramienta no tiene coste y permite embeber informes en aplicaciones web de las siguientes maneras:
 - Power BI Web: Permite hacer informes públicos publicándolos en el servicio Power BI Web [W3]. Estos informes se pueden incrustar en una aplicación web mediante un iframe. El acceso no requiere autenticación y no ofrece confidencialidad. Tampoco tiene coste. Descartada por falta de confidencialidad.
 - Power BI REST API: Mediante el uso de esta API es posible incrustar informes en una aplicación web desarrollados con la herramienta Power BI Desktop y alojados en el servicio Power BI Web. El acceso requiere autenticación en el portal de Office 365, es decir, requiere una doble autenticación, una en el portal web de Endalia y otra en Office 365. Tampoco tiene coste asociado. Descartado por tener que realizar una segunda autenticación fuera del portal web de Endalia.
 - Azure Power BI Embedded: Esta herramienta hace uso de la API REST de Power BI para el renderizado y la interacción de informes en una aplicación web, pero los informes en este caso no se publican en Power BI Web, sino en una aplicación de Azure Power BI Embedded. Esta opción ofrece seguridad mediante un sistema de tokens, y confidencialidad de los informes ya que solo pueden ser leídos a través del portal web de Endalia. El coste de esta herramienta está compuesto por sesiones de uso. Las primeras 100 sesiones de cada mes son gratis, después el coste es de 4,217 € por cada 100 sesiones (4,2 céntimos por sesión, aproximadamente).

Finalmente es la herramienta elegida, concretamente mediante el uso de Azure Power BI Embedded ya que cumple con todos los requisitos establecidos, además de ofrecer un coste asumible por parte del cliente.



4. ANÁLISIS DE LA INFRAESTRUCTURA DE APLICACIONES

Debido a que Azure Power BI Embedded requiere el uso del framework mínimo de .NET 4.5, y actualmente en el portal web se usa el framework de .NET 4.0 y no se puede subir la versión a corto plazo, es necesario crear una aplicación web intermedia que pueda embeber informes de Power BI que use el framework de .NET 4.5.

Esta aplicación se llama EmbedReports, y a su vez, es embebida por el portal web de Endalia.

A continuación, se muestra un diagrama en el que se describe la interacción entre las aplicaciones necesarias para la infraestructura a alto nivel:

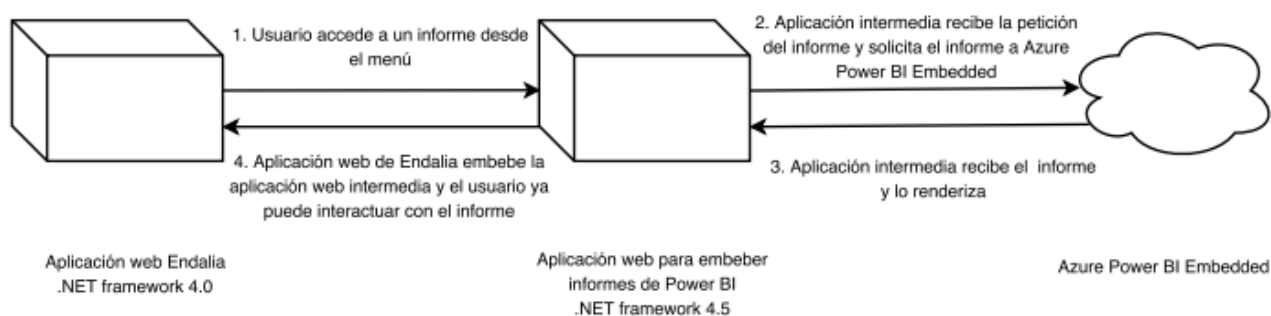


Figura 2. Flujo de visualización de informes a alto nivel

5. BIBLIOGRAFÍA

5.1 Referencias web

[W1] <https://www.ibm.com/support/knowledgecenter/>

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

[W3] <https://powerbi.microsoft.com/es-es/>



DISEÑO

DESARROLLO DE UN MÓDULO DE BUSINESS INTELLIGENCE

VERSIÓN 1.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	1.0	Redacción inicial del documento	Víctor Beltrán Piñol



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. Tecnologías utilizadas	9
3.1 Introducción.....	9
3.2 Consideraciones iniciales.....	9
3.3 SQL Server.....	9
3.4 Plataforma .NET.....	9
3.5 IIS.....	10
4. Diseño del almacén de datos.....	11
4.1 Introducción.....	11
4.2 Metodología utilizada.....	11
4.3 Diseño general de los data marts	11
4.4 Dimensiones comunes.....	12
4.4.1 Dimensión Fechas.....	12
4.4.2 Dimensión Empleados	13
4.4.3 Dimensión Áreas	14
4.4.4 Dimensión Roles	14
4.4.5 Dimensión compañías.....	15
4.4.6 Dimensión Tipos.....	15
4.4.7 Dimensión subtipos.....	15
4.5 Diseño del data mart de objetivos e indicadores generales	16
4.6 Diseño del data mart de salarios y nóminas	17
4.7 Diseño del data mart de planificación.....	18
4.8 Diseño del data mart de organización	19
4.9 Diseño del data mart de desempeño.....	20
4.10 Diseño del data mart de formación	21



4.11	Diseño del data mart de dimensionamiento y personal no residente.....	22
4.12	Diseño del data mart de rotación de plantilla.....	23
4.13	Integración de los data marts.....	24
5.	Diseño de la aplicación intermedia EmbedReports.....	25
5.1	Introducción.....	25
5.2	Arquitectura de la aplicación	25
5.3	Diseño del módulo de Business Intelligence en el portal web.....	26
5.4	Diagrama de secuencia	26
6.	Prototipado de la interfaz	27
6.1	Introducción.....	27
6.2	Interfaz básico.....	27
7.	Bibliografía	28
7.1	Referencias web	28



1. INTRODUCCIÓN

1.1 Propósito del documento

El objetivo del diseño es obtener un punto de partida para actividades de implementación, a partir de las especificaciones de requisitos y análisis previos. Esta fase de diseño está dividida en dos partes bien diferenciadas:

- Diseño del almacén de datos.
- Diseño del módulo de Business Intelligence dentro del portal web.

1.2 Alcance del documento

El alcance del documento comprende toda la fase de diseño del módulo de Business Intelligence.

1.3 Acrónimos

- ACID: Atomicity, Consistency, Isolation and Durability.
- API: Application Programming Interface.
- DDL: Data Definition Language.
- DML: Data Manipulation Language.
- ETL: Extract, Transform and Load.
- IIS: Internet Information Services.
- KPI: Key Performance Indicator.
- OLAP: On-Line Analytical Processing.
- OLTP: On-Line Transaction Processing.
- REST: Representational State Transfer.
- SGBD: Sistema Gestor de Bases de Datos.
- T-SQL: Transact – Structured Query Language.

1.4 Definiciones

- UML: lenguaje de modelado de sistemas de software desarrollado por *Rational* y de uso extendido.
- KPI: Un KPI (key performance indicator), conocido también como indicador clave o medidor de desempeño o indicador clave de rendimiento, es una medida del nivel del desempeño de un proceso.
- Diagrama de secuencia: El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML.



1.5 Referencias

Este documento contiene referencias a los siguientes documentos del trabajo, contenidos en la sección “Anexos”:

- Especificación de requisitos: documento que describe la especificación de requisitos.
- Análisis del sistema: documento que describe las fases de análisis del almacén de datos, de la integración del módulo con el portal web, junto con un listado de casos de uso.

1.6 Resumen

En este documento se describe el proceso de diseño del módulo de Business Intelligence. Se compone de los siguientes apartados:

1. Introducción del documento, definición del propósito y alcance del mismo.
2. Se describe el proceso de diseño.
3. Se describen las tecnologías utilizadas.
4. Diseño del almacén de datos.
5. Diseño de la integración del módulo con el portal web.
6. Prototipado de la interfaz.
7. Bibliografía y referencias Web.



2. DESCRIPCIÓN DEL PROCESO

Partiendo de la fase de análisis y de los requisitos identificados mediante la especificación de requisitos, se procede al diseño tanto del almacén de datos como de la integración de los informes con el portal web de Endalia [W3].

Para el diseño del almacén de datos se parte de los indicadores propuestos, mientras que para el diseño de la integración de los informes con el portal web, se parte de la herramienta elegida para llevar a cabo la integración, Power BI Embedded.



3. TECNOLOGÍAS UTILIZADAS

3.1 Introducción

En esta sección se describen las tecnologías que serán usadas en el módulo de Business Intelligence. Es necesario tener en cuenta las capacidades de las tecnologías usadas para conocer de que manera se puede llevar a cabo el desarrollo del módulo.

3.2 Consideraciones iniciales

Al ser éste un proyecto desarrollado en la empresa Endalia, y ser un módulo que debe integrarse dentro de una aplicación web mayor, es necesario tener en cuenta las tecnologías usadas en esta aplicación web, para que el módulo sea totalmente compatible y fácil de integrar.

Para llevar a cabo la construcción del almacén de datos se va a usar el gestor de base de datos Microsoft SQL Server 2014, y para la integración de los informes en el portal web, se van a usar las mismas tecnologías que se usan en el portal web, el framework .NET de Microsoft [W1], con el uso de páginas web ASP, junto al lenguaje de programación C#.

Por otra parte, se ha utilizado el IIS (Internet Information Services) como servidor de desarrollo y servidor web.

3.3 SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales, desarrollado por Microsoft. Utiliza el lenguaje de desarrollo T-SQL (Transact-SQL) para el manejo y recuperación de los datos.

Entre sus características principales destacan las siguientes:

- Soporte de transacciones ACID.
- Soporta procedimientos almacenados.
- Entorno gráfico que permite ejecutar comandos DDL y DML.
- T-SQL como lenguaje de consultas nativo.
- Permite administrar información de otros servidores de datos.

Esta es la tecnología usada para el desarrollo del almacén de datos.

3.4 Plataforma .NET

Microsoft .NET es un framework que provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

Entre los principales componentes del framework cabe destacar los siguientes:

- El conjunto de lenguajes de programación. Está formado principalmente por C# y Visual Basic, aunque no son los únicos.



- La biblioteca de clases base BCL. Esta biblioteca maneja la mayoría de operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones, como por ejemplo la administración de memoria, o las operaciones aritméticas, entre otras.
- El entorno común de ejecución para lenguajes CLR (Common Language Runtime). Es el núcleo del framework .NET. Esta herramienta de desarrollo es la encargada de compilar el código. El código ejecutable es almacenado en memoria cache.

Esta plataforma es la usada para el desarrollo del código que formará parte del módulo que será integrado dentro de la aplicación web de Endalia, usando el lenguaje de programación C#.

3.5 IIS

Internet Information Services es un servidor web y un conjunto de servicios desarrollado por Microsoft. Este servicio convierte un PC en un servidor web para Internet o una intranet.

Incluye varios módulos que permite procesar distintos tipos de páginas, destacando entre estos tipos ASP (Active Server Pages), que es una tecnología propia de Microsoft.

Esta tecnología es utilizada para el despliegue de dos servidores web, por un lado el del portal web el cual usa el framework .NET 4.0, y por otro lado también sirve como plataforma de despliegue del servidor de la aplicación web intermedia para embeber los informes mediante Power BI Embedded, que usa el framework .NET 4.5.



4. DISEÑO DEL ALMACÉN DE DATOS

4.1 Introducción

En esta sección se describe el proceso de diseño del almacén de datos. Para esta fase se tiene como punto de partida el conjunto de indicadores definidos en la fase de análisis del almacén de datos.

4.2 Metodología utilizada

La metodología seguida para la fase de diseño del almacén de datos es la metodología de Kimball, desarrollada por Ralph Kimball. Esta metodología se basa en el ciclo de vida dimensional del negocio (Business Dimensional Lifecycle), que está basado en los siguientes cuatro principios básicos:

- Centrarse en el negocio.
- Construir una infraestructura de información adecuada.
- Realizar entregas incrementales significativas.
- Ofrecer una solución integral completa.

Esta metodología sigue un diseño bottom-up, definiendo primero los distintos data marts, uno por cada uno de los grupos de indicadores definido, para después integrarlos todos en el mismo almacén de datos.

Para el diseño del almacén de datos se usa un modelo dimensional, diferente del modelo entidad-relación de las bases de datos relacionales. Este modelado no implica necesariamente una base de datos relacional y se orienta en torno al rendimiento de las consultas analíticas. En el modelado dimensional se utilizan dos conceptos de tablas, las tablas de dimensiones y las tablas de hechos.

Las tablas de hechos tienen como atributos una o más medidas del proceso organizacional, sobre las cuales se desean realizar operaciones de agregado. La granularidad es el nivel de detalle que posee cada registro de una tabla de hechos. La granularidad seleccionada para el diseño del almacén de datos es la unidad de tiempo de un día, es decir, en un registro se almacena el estado de un empleado de un día, por lo que se obtiene un registro por cada empleado por cada día que ha estado activo en la empresa.

Se han creado un total de ocho tablas de hechos, una por cada uno de los data marts que componen el almacén de datos. Las tablas de hechos almacenan distintas medidas como el FTE diario, el salario diario, o si un empleado a causado baja o alta en la empresa, por ejemplo.

Por otro lado se encuentran las tablas de dimensiones. Las tablas de dimensiones tienen un conjunto de atributos que permiten obtener una perspectiva acerca de las medidas de las tablas de hechos. Estas dimensiones, serán los filtros que se van a poder aplicar sobre los informes. Por lo tanto, entre las principales tablas de dimensiones cabe destacar el tiempo (un registro por cada día), los empleados (un registro por empleado), o las áreas de una empresa (un registro por cada una de estas áreas). Cada una de estas tablas de dimensiones tienen un identificador único, el cual aparece en la tabla de hechos cada vez que aparece en el estado del empleado, relacionando así la tabla de hechos con la de dimensiones.

4.3 Diseño general de los data marts

Partiendo de una metodología previa establecida, se procede al diseño del almacén de datos, empezando por el diseño de cada uno de los data marts. Cada data mart estará formado por una tabla de hechos con las medidas necesarias para dar respuesta a los indicadores, y un número variable de tablas de dimensiones.

El diseño de estos data marts se ha llevado de manera iterativa siguiendo los siguientes cuatro pasos en cada data mart:

1. Elegir el proceso de negocio: rotación de plantilla, planificación, salarios de empleados, etc...



2. Establecer el nivel de granularidad: estado del empleado por cada día que ha estado activo en la empresa.
3. Elegir las dimensiones: fecha, empleados, tipos, subtipos, áreas, roles, etc...
4. Identificar las medidas y tablas de hechos: FTE diario, salario diario, causa baja, etc...

Definidos los cuatro puntos anteriores de un data mart, se procede a realizar un diagrama en estrella para cada uno de estos data marts. Este esquema es conocido como esquema en estrella debido a que contiene una tabla principal en el centro, la tabla de hechos, que está conectada con cada una de las tablas de dimensiones que tiene a su alrededor.

El modelo en estrella es un modelo desnormalizado, ya que el objetivo de este modelo es obtener un rendimiento óptimo para las consultas de agregación de datos. Una característica de los almacenes de datos es que son no volátiles, es decir que sus datos no son actualizados, manteniendo así el historial de los datos, una vez insertados datos en la tabla de hechos ya no se modifican, de ahí que ya no sea necesario normalizar las tablas.

A continuación, se muestra una descripción de las dimensiones comunes, y después, el diseño de cada uno de los distintos data marts en detalle, junto a una justificación de las medidas seleccionadas en cada tabla de hechos.

4.4 Dimensiones comunes

A continuación, se muestra el detalle de cada una de las tablas de dimensiones comunes, junto a sus atributos. Estas dimensiones corresponderán a los distintos filtros disponibles en los informes.

Además, cada dimensión excepto fechas y empleado contiene un registro especial que es un campo de texto vacío. Ya que no todos los empleados tienen asociados valores de todas las dimensiones, y no pueden existir campos nulos en la tabla de hechos.

4.4.1 Dimensión Fechas

Esta dimensión contiene información de los días en los cuales se lleva a cabo el análisis de los data marts. Contiene un registro por cada día, con la información correspondiente a ese día. Debido a los pocos datos del primer cliente en el cual se implanta el módulo de Business Intelligence antes del año 2010, se decide que esta tabla de dimensiones contendrá datos acerca de las fechas posteriores al 1 de enero de 2010 hasta la actualidad.

DimDates			
	Nombre de columna	Tipo de datos	Permitir val...
🔑	pk_DateID	int	<input type="checkbox"/>
	Ddate	smalldatetime	<input type="checkbox"/>
	DateDayNumber	int	<input checked="" type="checkbox"/>
	DateWeekNumber	int	<input checked="" type="checkbox"/>
	DateMonthNumber	int	<input checked="" type="checkbox"/>
	DateQuarterNumber	int	<input checked="" type="checkbox"/>
	DateSemesterNumber	int	<input checked="" type="checkbox"/>
	DateYearNumber	int	<input checked="" type="checkbox"/>
	DateDayOfWeekES	nvarchar(20)	<input checked="" type="checkbox"/>
	DateDayOfWeekCA	nvarchar(20)	<input checked="" type="checkbox"/>
	DateDayOfWeekEN	nvarchar(20)	<input checked="" type="checkbox"/>
	DateMonthNameES	nvarchar(20)	<input checked="" type="checkbox"/>
	DateMonthNameCA	nvarchar(20)	<input checked="" type="checkbox"/>
	DateMonthNameEN	nvarchar(20)	<input checked="" type="checkbox"/>
	DateDayInWeekNumber	int	<input checked="" type="checkbox"/>
	DateDayInMonthNumber	int	<input checked="" type="checkbox"/>
	DateDayInQuarterNumber	int	<input checked="" type="checkbox"/>
	DateDayInSemesterNum...	int	<input checked="" type="checkbox"/>
	DatIsWeekEnd	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 1. Dimensión Fechas



La clave primaria de la dimensión consiste en un número entero con el siguiente formato: aaaammdd (ej: 20100101). Después tiene un campo que almacena la fecha con el tipo de datos smalldatetime. A continuación, contiene una serie de campos numéricos para obtener el número del día, del mes, de la semana, etc...

Los siguientes campos son de texto y hacen referencia al día de la semana y al nombre del mes tanto en español, como en catalán o inglés. Finalmente hay otros atributos auxiliares para saber el número de día dentro de la semana o mes por ejemplo, además de un valor booleano que indica si un día pertenece a fin de semana o no.

Diversos atributos como estos últimos, a día de hoy no son utilizados, pero han sido incluidos en la fase de diseño ya que al ser una dimensión, y no una tabla de hechos donde sí que interesa que sea lo más estrecha posible, no sobrecargan el sistema y permiten que en un futuro sean desarrollados indicadores que requieran estos campos.

4.4.2 Dimensión Empleados

Esta dimensión contiene información acerca de todos los empleados que han pasado por la organización, tanto históricos como activos.

DimEmpleees			
	Nombre de columna	Tipo de datos	Permitir val...
🔑	pk_EmplID	int	<input type="checkbox"/>
	EmpCode	nvarchar(20)	<input type="checkbox"/>
	EmpCardID	nvarchar(30)	<input checked="" type="checkbox"/>
	EmpBirthDate	smalldatetime	<input checked="" type="checkbox"/>
	EmpGender	nvarchar(1)	<input checked="" type="checkbox"/>
	EmpAddress	nvarchar(500)	<input checked="" type="checkbox"/>
	EmpPostalCode	nvarchar(10)	<input checked="" type="checkbox"/>
	EmpCity	nvarchar(100)	<input checked="" type="checkbox"/>
	EmpRegion	nvarchar(30)	<input checked="" type="checkbox"/>
	EmpEmail	nvarchar(70)	<input checked="" type="checkbox"/>
	EmpName	nvarchar(50)	<input checked="" type="checkbox"/>
	EmpSurname	nvarchar(80)	<input checked="" type="checkbox"/>
	EmpSurname1	nvarchar(40)	<input checked="" type="checkbox"/>
	EmpSurname2	nvarchar(40)	<input checked="" type="checkbox"/>
	EmpCountry	nvarchar(255)	<input checked="" type="checkbox"/>
	EmplsFromUE	bit	<input checked="" type="checkbox"/>
	EmplsForeignFromUE	bit	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 2. Dimensión Empleados

La clave primaria de esta tabla es un número entero, el cual hace referencia a la tabla de datos de empleados de la base de datos origen. Después contiene campos con información de interés acerca del empleado.

Al igual que en la dimensión de fechas, hay campos que a día de hoy no están siendo usados, pero así el modelo está preparado por si en un futuro se requieren estos atributos, como por ejemplo el nombre o el dni.

4.4.3 Dimensión Áreas

Esta dimensión contiene información acerca de las diferentes áreas de una empresa. En el primer cliente en el que se implanta el módulo coincide con los centros de coste que gestiona.

DimCostCenters			
	Nombre de columna	Tipo de datos	Permitir val...
🔑	pk_CostCenterID	int	<input type="checkbox"/>
	CostCenterName	nvarchar(255)	<input type="checkbox"/>
	CostCenterRemarks	nvarchar(255)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 3. Dimensión Áreas

Esta dimensión tan solo contiene tres campos: un identificador autoincremental, el nombre del área y una descripción del área en caso de que se requiera.

4.4.4 Dimensión Roles

Esta dimensión contiene información referente a los diferentes roles que pueden tener asociados los empleados de una empresa.

DimRoles			
	Nombre de columna	Tipo de datos	Permitir valores NULL
🔑	pk_RolID	int	<input type="checkbox"/>
	RolName	nvarchar(255)	<input type="checkbox"/>
	RolNameCA	nvarchar(255)	<input checked="" type="checkbox"/>
	RolCode	nvarchar(255)	<input checked="" type="checkbox"/>
	RolRemarks	nvarchar(300)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 4. Dimensión roles

Esta dimensión contiene un identificador autoincremental, el nombre tanto en catalán como en castellano, un código y unas observaciones. Los campos del nombre en catalán y el código actualmente no se usan, pero al estar disponibles como campos auxiliares en la base de datos origen, también se ha añadido a la dimensión por si son de utilidad en un futuro.

4.4.5 Dimensión compañías

Esta dimensión contiene información acerca de las distintas compañías que puede tener una organización, entendiendo por compañía una entidad por encima de un área.

DimCompanies			
	Nombre de columna	Tipo de datos	Permitir val...
🔑	pk_CompanyID	int	<input type="checkbox"/>
	CompanyName	nvarchar(255)	<input type="checkbox"/>
			<input type="checkbox"/>

Figura 5. Dimensión compañías

4.4.6 Dimensión Tipos

Esta dimensión contiene información acerca de las tipologías de empleados que puede tener una empresa. Al igual que la dimensión roles, esta dimensión tiene campos que no se usan en la actualidad como el nombre del tipo en catalán, o un código del tipo, pero su utilización es probable en un futuro, por lo que ya se incluyen en el diseño

DimWorkTypes			
	Nombre de columna	Tipo de datos	Permitir valores NULL
🔑	pk_WorkTypeID	int	<input type="checkbox"/>
	WTypeName	nvarchar(255)	<input type="checkbox"/>
	WTypeNameCA	nvarchar(255)	<input checked="" type="checkbox"/>
	WTypeCode	nvarchar(255)	<input checked="" type="checkbox"/>
	WtypeRemarks	nvarchar(300)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 6. Dimensión tipos

4.4.7 Dimensión subtipos

Esta dimensión contiene información acerca de las distintas subtipologías que puede tener asociadas un empleado dentro de una empresa. También, como las dimensiones roles o tipos tiene dos campos auxiliares que no se usan en la actualidad.

DimWorkSubTypes			
	Nombre de columna	Tipo de datos	Permitir valores NULL
🔑	pk_WorkSubTypeID	int	<input type="checkbox"/>
	WSubTypeName	nvarchar(255)	<input type="checkbox"/>
	WSubTypeNameCA	nvarchar(255)	<input checked="" type="checkbox"/>
	WSubTypeCode	nvarchar(255)	<input checked="" type="checkbox"/>
	WSubTypeRemarks	nvarchar(300)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 7. Dimensión subtipos



4.5 Diseño del data mart de objetivos e indicadores generales

El objetivo de este data mart es poder responder a los siguientes tres indicadores:

1. Número de personas en un periodo determinado.
2. Número de personas en un periodo determinado por nivel (categoría y rol).
3. Número de personas en un periodo determinado por tipo y subtipo.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso son los objetivos e indicadores generales de la empresa. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

A parte de las dimensiones comunes, este data mart contiene dos dimensiones específicas para poder contestar al indicador número dos: la dimensión nivel y la dimensión categoría.

Categoría y rol se han diseñado como dimensiones, y no como medidas de la tabla de hechos, ya que pueden tomar multitud de valores diferentes.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos solo tiene una medida, que es común en todas las tablas de hechos, el FTE diario del empleado. Esta medida es común a todas las tablas de hechos ya que el cliente solicitó que el dato del FTE fuese visible en todos los informes.

Además de la medida común, la tabla de hechos contiene una clave extranjera que corresponde con cada clave primaria de las dimensiones. Para el estado del empleado de cada día, solo puede tener asociado un valor de cada una de las tablas de dimensiones.

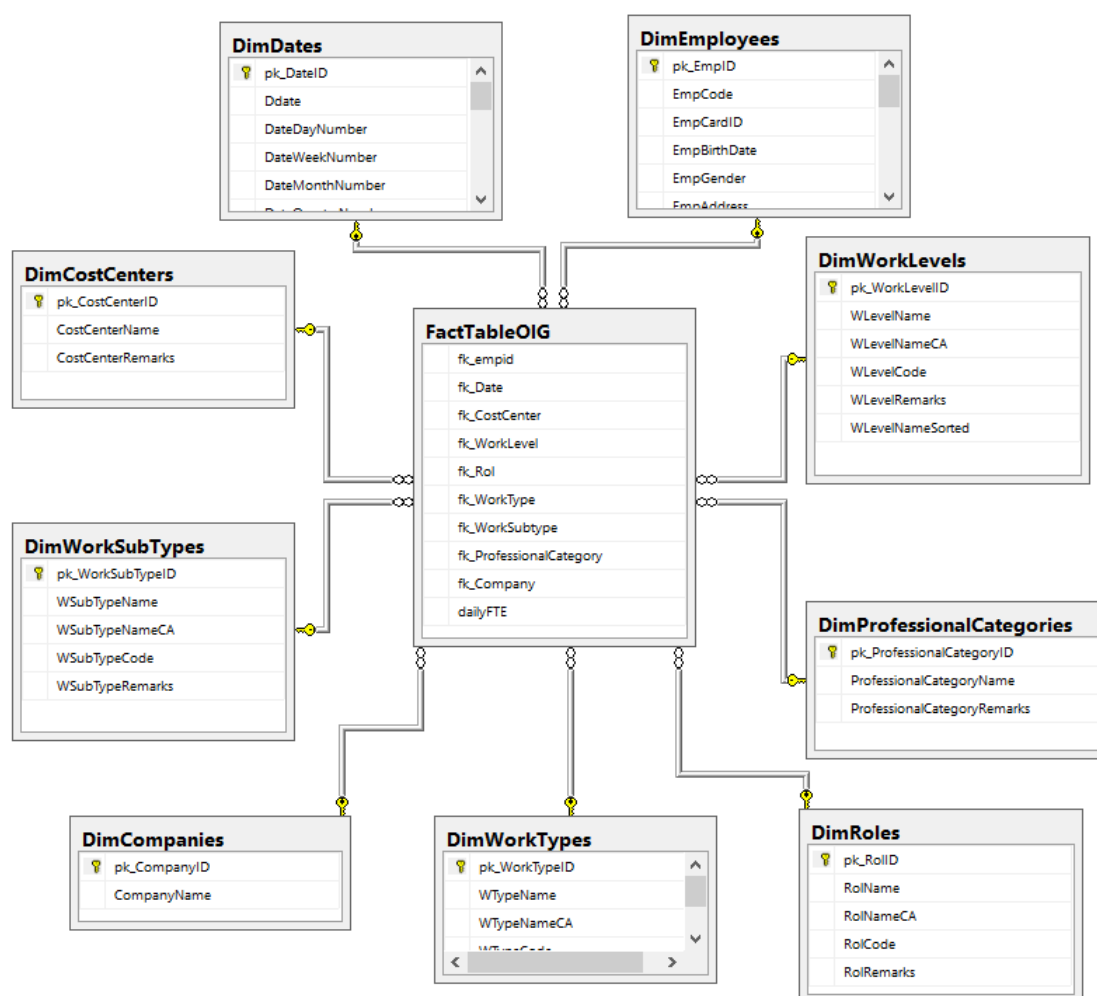


Figura 8. Esquema en estrella del data mart de objetivos e indicadores generales

4.6 Diseño del data mart de salarios y nóminas

El objetivo de este data mart es permitir contestar a los siguientes tres indicadores:

1. Costes salariales de plantilla mensuales por área.
2. Coste medio real versus coste presupuestado.
3. Costes salariales por tipo y subtipo de empleado respecto al coste salarial total.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso son los salarios, costes y costes planificados de los empleados de la empresa. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Este data mart requiere una dimensión extra a parte de las comunes. Esta dimensión contiene datos referentes a costes especiales, es decir, costes que no tienen asociados los empleados, como por ejemplo promociones de un centro de coste o costes planificados para nuevas incorporaciones. Esta dimensión es como una mini tabla de hechos ya que contiene información redundante de otras dimensiones, si bien no está conectada directamente con éstas.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos tiene varias medidas, aunque en el informe final solo se usan las medidas referentes a valores diarios. Referente a los salarios están las medidas salario anual y mensual de un empleado, la media del salario mensual porque en un mes un empleado puede tener distintos salarios y la media de este salario diario, que se obtiene al dividir la media del salario mensual entre los días que el empleado ha estado activo.

En cuanto a las medidas referentes a costes reales y presupuestados, se tienen las medidas de coste real mensual y diario, y por otro lado coste presupuestado mensual y diario.

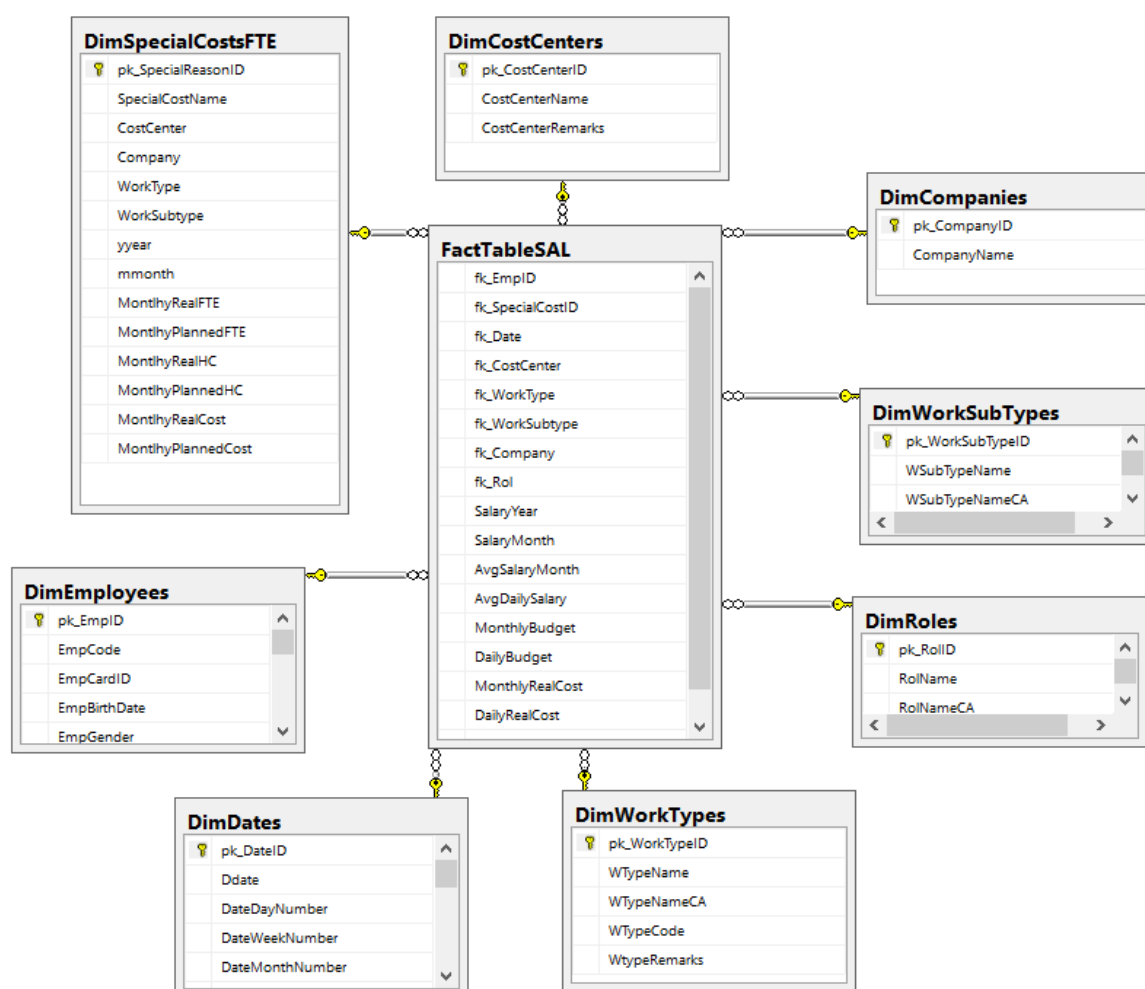


Figura 9. Esquema en estrella del data mart de salarios y nóminas

4.7 Diseño del data mart de planificación

El objetivo de este data mart es poder responder a los siguientes tres indicadores:

1. Headcount y FTE por área, tipo y subtipo de empleado.
2. Desviaciones de Headcount y FTE versus realidad.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso la planificación en cuanto a los empleados de la empresa. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Al igual que el data mart de salarios, este data mart requiere una dimensión extra a parte de las comunes. Esta dimensión contiene datos referentes a FTE y Headcount especiales, es decir, que no tienen asociados los empleados, como por ejemplo planificación para nuevas incorporaciones. Esta dimensión es como una mini tabla de hechos ya que contiene información redundante de otras dimensiones, si bien no está conectada directamente con éstas.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos tiene varias medidas, aunque en el informe final solo se usan las medidas referentes a valores diarios. Referente al FTE están las medidas de FTE mensual y diario, tanto real como planificado. Lo mismo ocurre para las medidas del Headcount, tiene los valores mensuales y diarios, tanto reales como presupuestados. Existe una medida extra que es un booleano que define si un empleado está activo o no, para identificar los casos especiales, en los que se pone que no está activo.

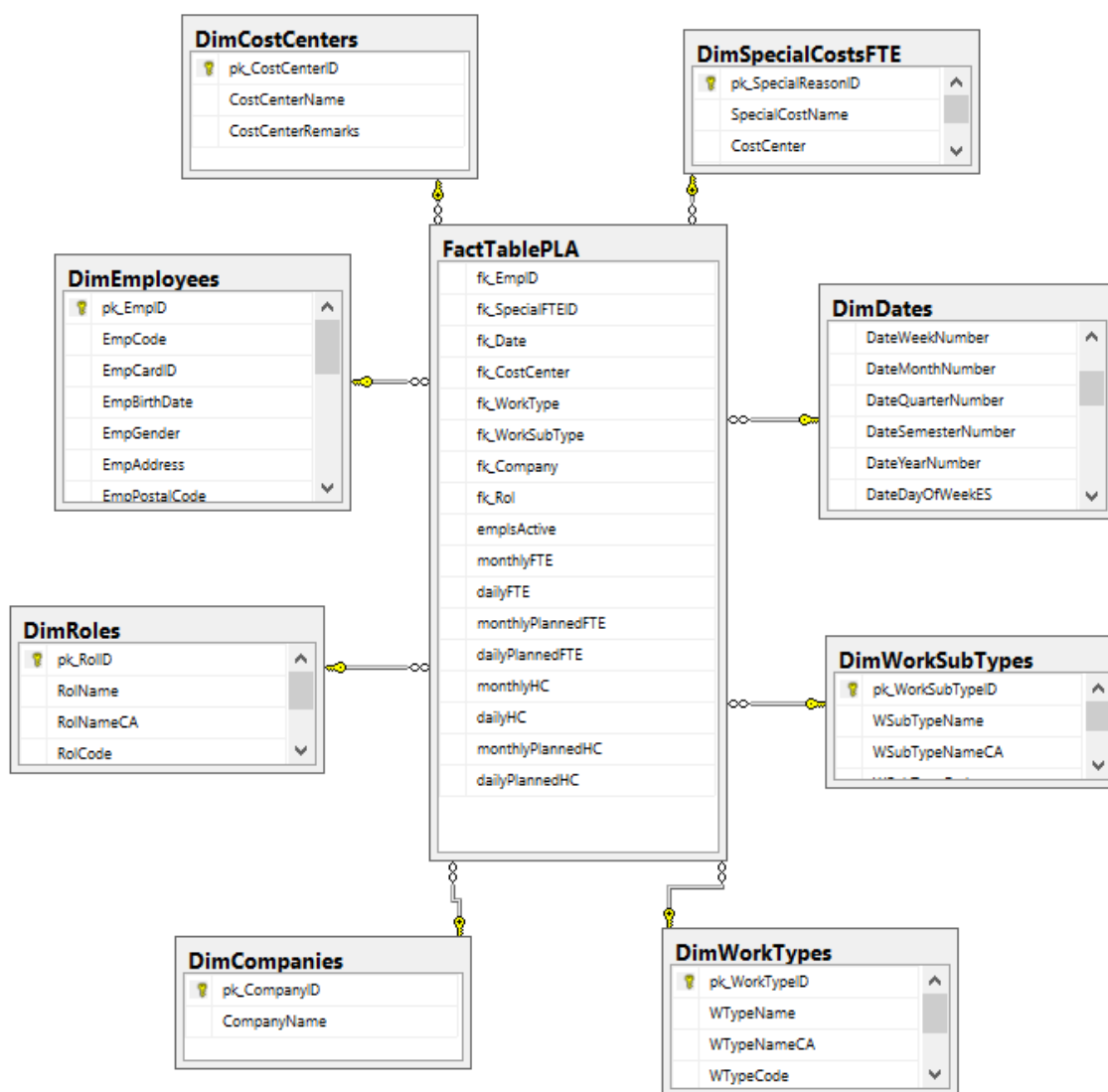


Figura 10. Esquema en estrella del data mart de planificación

4.8 Diseño del data mart de organización

El objetivo de este data mart es responder al siguiente indicador:

1. Tiempo medio de permanencia en el puesto por área.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso la organización de los empleados en cuanto a su permanencia en la empresa. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Este data mart no requiere de ninguna dimensión extra, a parte de las comunes.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos tiene varias medidas referentes al tiempo que permanece un empleado en la empresa, calculado en días, años y meses, siendo la medida calculada en años la única usada en el informe final.

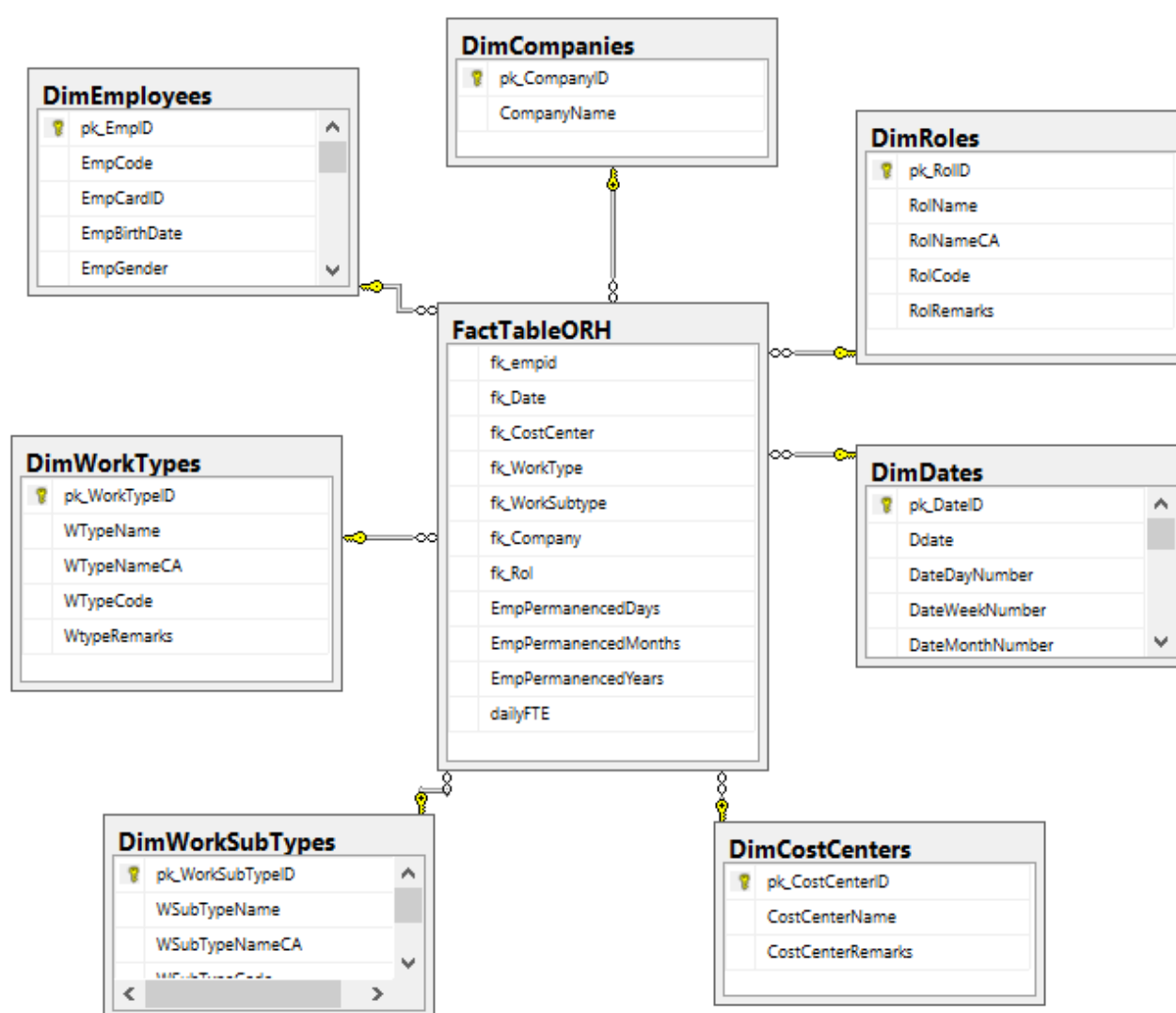


Figura 11. Esquema en estrella del data mart de organización

4.9 Diseño del data mart de desempeño

El objetivo de este data mart es responder a los siguientes indicadores:

1. Porcentaje medio de desempeño por área.
2. Porcentaje medio de alto potencial versus total de la plantilla por rol.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso el desempeño de los empleados en la empresa. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Este data mart requiere una dimensión extra, a parte de las comunes, la dimensión de potenciales. Esta dimensión contiene un registro por cada uno de los distintos potenciales que puede tener un empleado en la empresa.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos tiene medidas referentes al desempeño como valor del 0 al 4, y el desempeño como porcentaje, siendo el valor 4 el correspondiente al 100%. Estos valores permanecen constantes a lo largo de un año entero, ya que las evaluaciones de desempeño son anuales.

Por último, la tabla de hechos tiene una medida que es un campo booleano, isHighPotential, la cual indica si el empleado tiene alto potencial, es decir, el potencial más alto de los posibles.

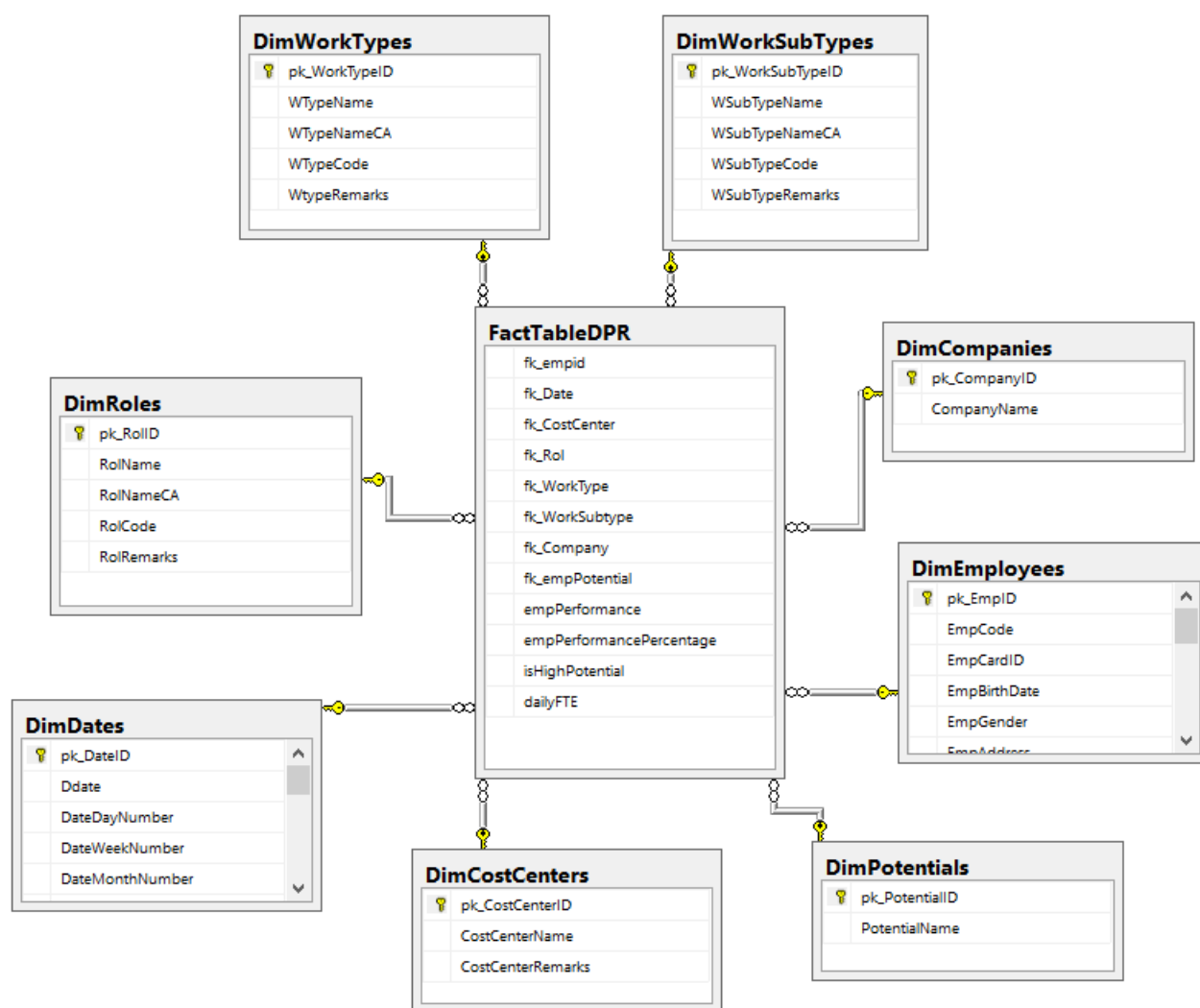


Figura 12. Esquema en estrella del data mart de desempeño

4.10 Diseño del data mart de formación

El objetivo de este data mart es responder al siguiente indicador:

1. Porcentaje y número de empleados que participan en formación.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso los cursos de formación que ofrece una empresa a sus empleados. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Este data mart no requiere de ninguna dimensión extra, a parte de las comunes.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos solo tiene una medida a parte de la común, que es el FTE diario, que consiste en un campo booleano que indica si el empleado a participado o no en cursos de formación a lo largo del año.

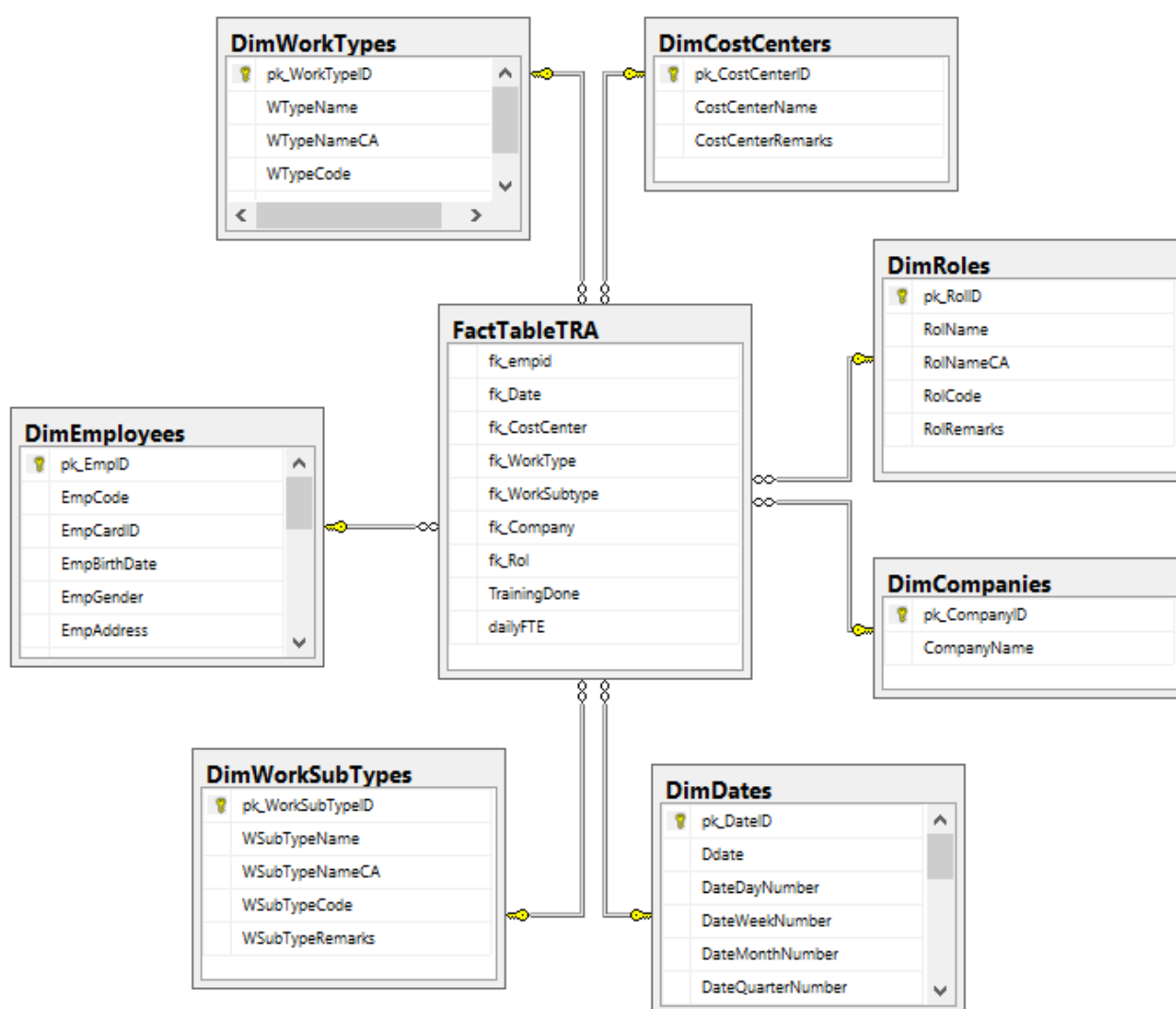


Figura 13. Esquema en estrella del data mart de formación

4.11 Diseño del data mart de dimensionamiento y personal no residente

El objetivo de este data mart es responder a los siguientes indicadores:

1. Porcentaje de plantilla fija y temporal sobre el total.
2. Personas incorporadas por país: extranjero UE y no UE.
3. Porcentaje de empleados extranjeros respecto al total de la plantilla.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso el dimensionamiento de plantilla fija y temporal y las nacionalidades de los empleados en la empresa. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Este data mart no requiere de ninguna dimensión extra, a parte de las comunes.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos tiene una medida que es un campo booleano que indica si un empleado se encuentra como empleado temporal, y otra medida que también es un campo booleano que indica si un empleado se ha incorporado un día en concreto o no.

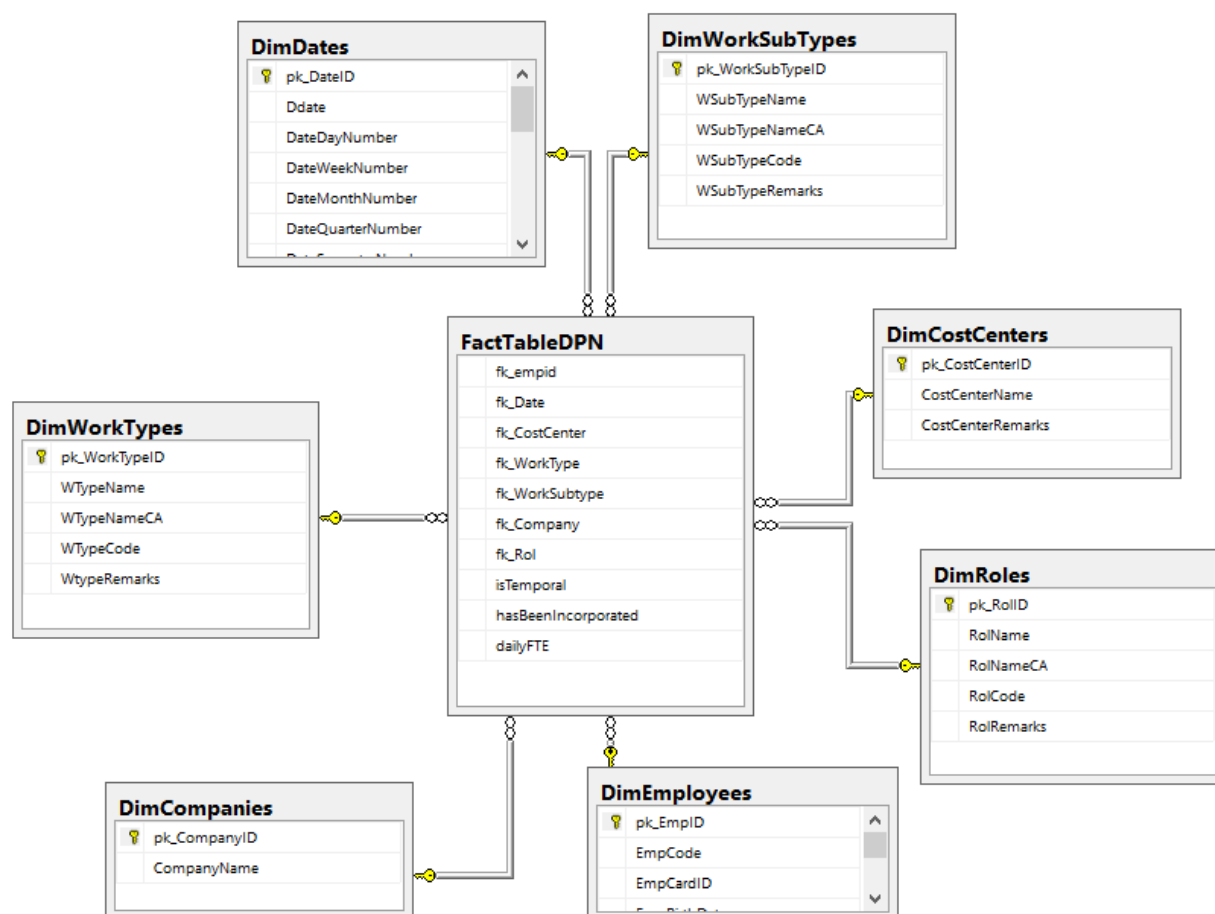


Figura 14. Esquema en estrella del data mart de dimensionamiento y personal no residente

4.12 Diseño del data mart de rotación de plantilla

El objetivo de este data mart es responder a los siguientes indicadores:

1. Plantilla contratada en cada mes.
2. Número de bajas totales mensuales.
3. Número de bajas por área mensuales.
4. Del total de bajas, porcentaje de bajas voluntarias y porcentaje de bajas no voluntarias.
5. Índice de rotación no deseada.
6. Índice de desvinculación no deseada.

Para el diseño de este data mart se empieza eligiendo el proceso del negocio, que en este caso es la rotación de los empleados en la empresa, es decir, el análisis de altas y bajas. El nivel de granularidad es el mismo en todos los data marts, el estado del empleado por cada día que ha estado activo en la empresa.

Este data mart requiere una dimensión extra, que es la dimensión tipo de baja, que contiene un registro por cada una de las distintas tipologías de bajas que se pueden dar cuando un empleado se da de baja en la empresa.

Identificadas todas las dimensiones, es hora de definir la tabla de hechos y sus medidas. Esta tabla de hechos tiene dos medidas principales que son dos campos booleanos, uno indica si un empleado ha sido dado de alta ese día, y el otro indica si el empleado ha sido dado de baja ese día.

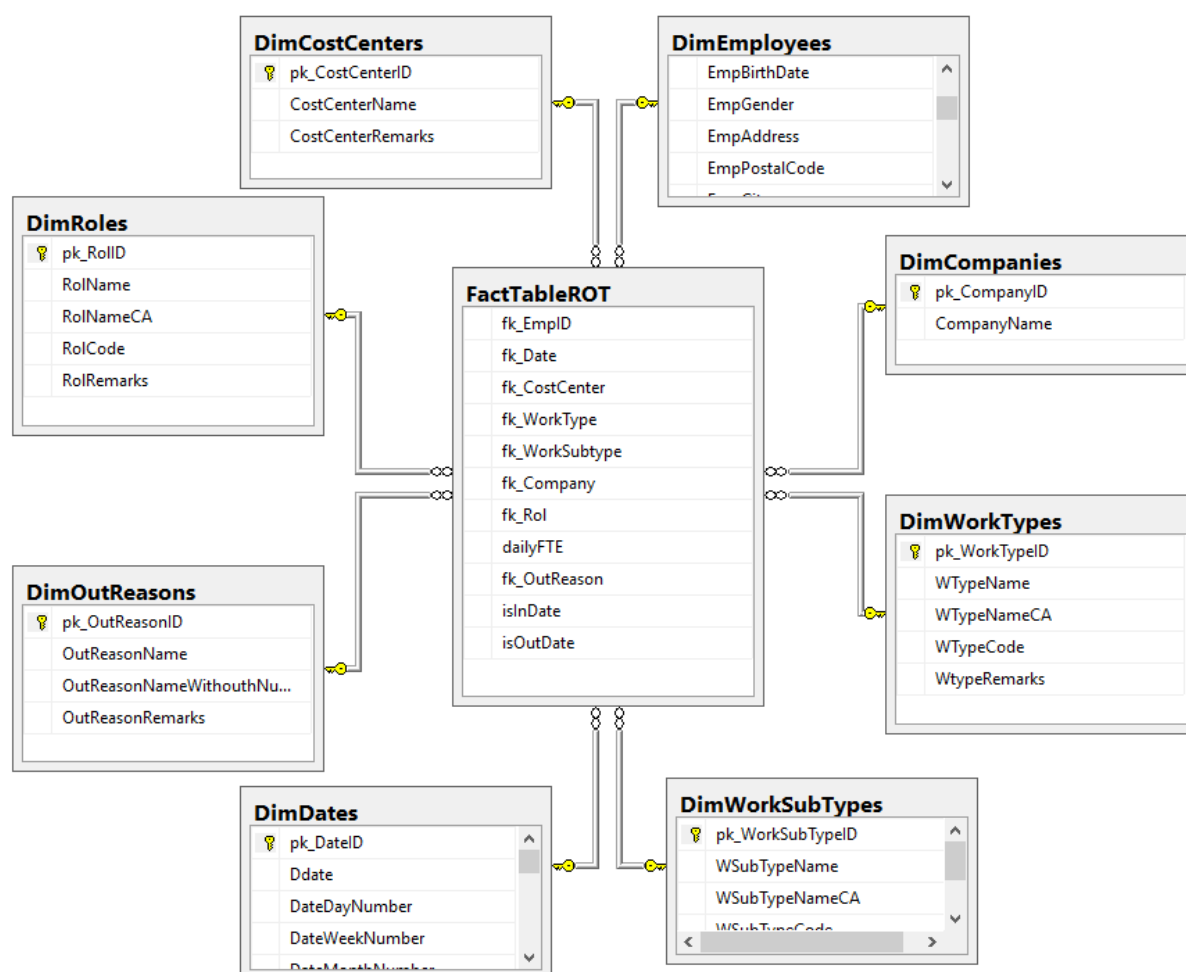


Figura 15. Esquema en estrella del data mart de rotación de plantilla

4.13 Integración de los data marts.

El diseño final del almacén de datos integra en una misma base de datos todos los diseños de los data marts anteriores. Para ello se crean todas las dimensiones una única vez, y todas las tablas de hechos.

En este diseño global las dimensiones se encuentran conectadas a varias tablas de hechos, en lugar de estar conectada a una única tabla de hechos, pero esto no afecta al diseño de cada uno de los informes, ya que se seleccionan únicamente los datos de las tablas de dimensiones y la tabla de hechos que influye en el proceso.



5. DISEÑO DE LA APLICACIÓN INTERMEDIA EMBEDREPORTS

5.1 Introducción

En este apartado se detalla el diseño del proyecto web intermedia que embebe los informes, para que a su vez, esta aplicación sea embebida por el portal web de Endalia. El motivo por el que se ha tenido que crear esta aplicación web intermedia es porque la herramienta que se usa para incrustar informes de Power BI en el portal web es Power BI Embedded, y esta herramienta requiere el uso del framework .NET 4.5, mientras que el portal web de Endalia usa el framework .NET 4.0, el cual no es posible subir de versión actualmente, debido a restricciones por parte de tecnologías utilizadas por alguno de los clientes.

También se muestra el diseño de las clases necesarias para añadir el módulo al portal web.

5.2 Arquitectura de la aplicación

El diseño de esta aplicación parte de la base ofrecida por Power BI Embedded disponible en el siguiente enlace:

<https://github.com/Azure-Samples/power-bi-embedded-integrate-report-into-web-app/tree/master/EmbedSample>

Este proyecto ofrece una aplicación web que permite incrustar informes realizados mediante Power BI alojados en Azure Power BI Embedded. La aplicación de ejemplo ha tenido que ser rediseñada para ser adaptada a las necesidades del portal web de Endalia.

Esta aplicación sigue el patrón MVC (Modelo-Vista-Controlador) de Microsoft, que permite dividir la aplicación en tres partes bien diferenciadas.

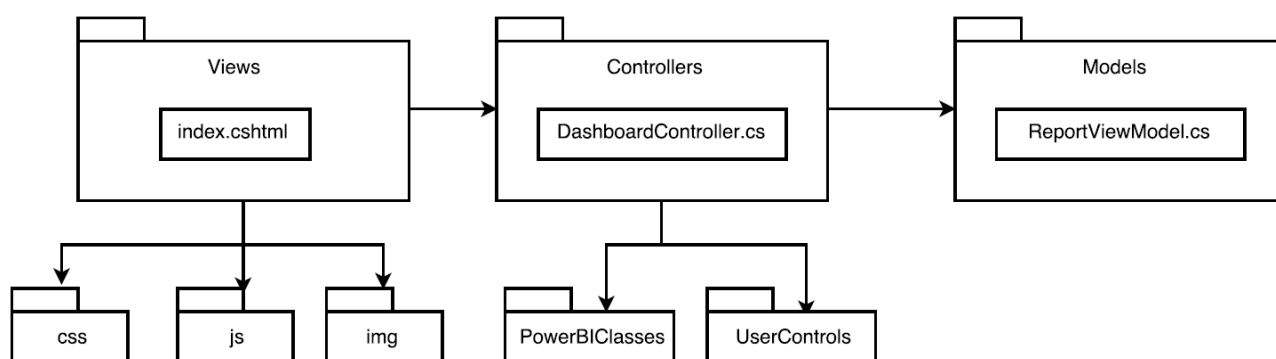


Figura 16. Arquitectura de la aplicación web intermedia

Para adaptar la aplicación al portal web de Endalia, ha sido necesario realizar modificaciones en el diseño, principalmente en el controlador, que es el encargado de la lógica de negocio.

Debido a que los identificadores de los informes, la clave de acceso y el nombre de la aplicación de Azure Power BI Embedded se encuentran en la base de datos relacional del cliente, se han añadido dos paquetes, el UserControls, que es el encargado de cargar las variables de configuración del cliente desde la base de datos, y el paquete PowerBIClasses, encargado de gestionar los atributos de los informes, es decir, sus identificadores, claves de acceso y nombre de la aplicación.

5.3 Diseño del módulo de Business Intelligence en el portal web

Para el diseño del acceso al módulo de Business Intelligence se parte del diseño de la aplicación intermedia para embeber informes.

Dentro del portal web es necesario crear un formulario aspx, para la gestión de los ocho informes que inicialmente formarán parte del módulo de Business Intelligence.

Este formulario hereda del MasterPage del portal web, y su contenido principal es un iframe con una ruta a la que apunta variable.

Al ser cargado el formulario, se comprueba desde que enlace del menú se ha accedido para comprobar cual es el informe que solicita el usuario.

Una vez comprobado el informe que solicita el usuario se cambia la dirección a la que debe apuntar el iframe, que siempre es al index del proyecto web intermedio, pero añadiendo un identificador de tres letras que hace referencia al informe solicitado como parámetro.

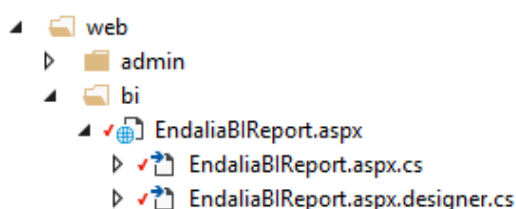


Figura 17. Estructura de ficheros añadida al portal web

5.4 Diagrama de secuencia

A continuación, se presenta un diagrama de secuencia [W2] [W4] a alto nivel de la interacción del usuario con el portal web, desde que el usuario solicita un informe, hasta que este es renderizado y disponible por el usuario.

Con este diagrama se pretende mostrar la interacción entre las diferentes partes implicadas para que el usuario pueda explotar informes desde el portal web.

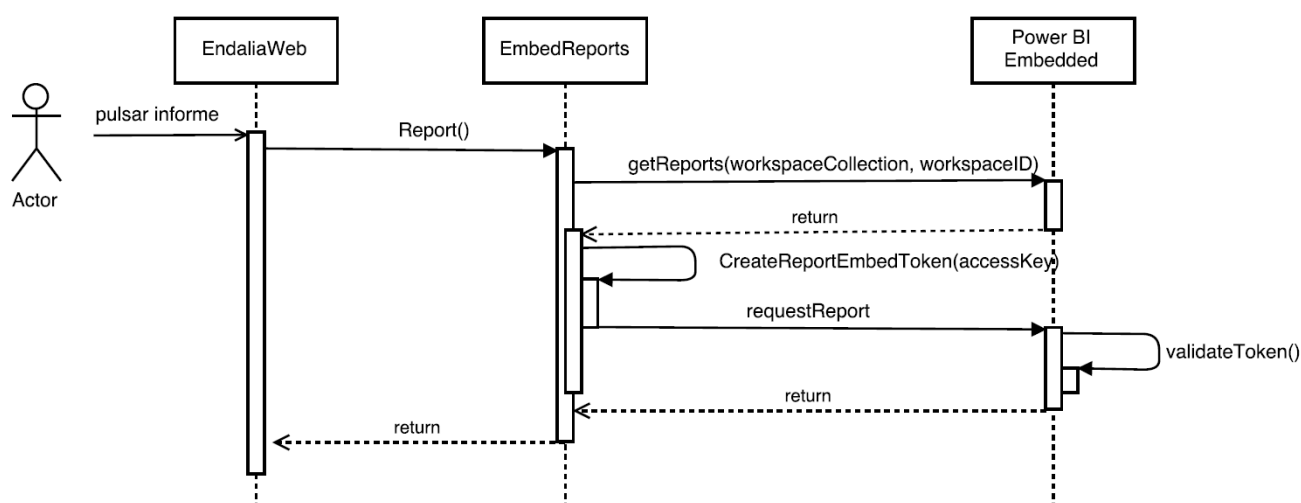


Figura 18. Diagrama de secuencia a alto nivel

6. PROTOTIPADO DE LA INTERFAZ

6.1 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.

6.2 Interfaz básico

A continuación se muestra la base del diseño del interfaz gráfico, enmarcando las diferentes secciones que lo componen

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: se trata de un menú desplegable, situado en el lateral izquierdo.
- Contenido: la parte central de la página. Estará formado por el contenido de la aplicación web EmbedReports

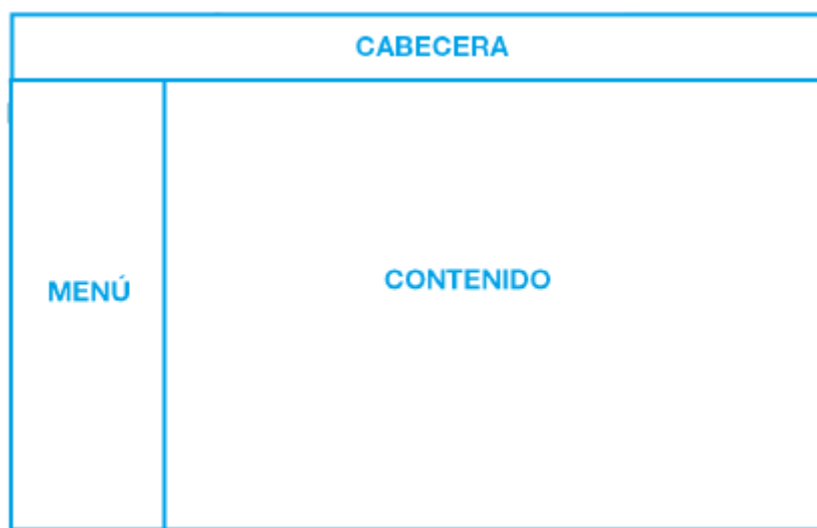


Figura 19. Prototipo de interfaz básico

7. BIBLIOGRAFÍA

7.1 Referencias web

[W1] <http://www.microsoft.com/net>

[W2] <http://www.uml.org>

[W3] <http://www.endalia.com>

[W4] <https://msdn.microsoft.com/es-es/library/dd409377.aspx>



IMPLEMENTACIÓN

DESARROLLO DE UN MÓDULO DE BUSINESS INTELLIGENCE

VERSIÓN 1.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	1.0	Redacción inicial del documento	Víctor Beltrán Piñol



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 Resumen.....	7
2. Descripción del proceso.....	8
3. Tecnologías, herramientas y lenguajes	9
4. Implementación del almacén de datos	10
4.1 Definición del proceso	10
4.2 Objetivos del almacén de datos.....	10
4.3 Proceso ETL.....	10
4.3.1 Poblar dimensiones.....	10
4.3.2 Poblar tablas de hechos.....	11
5. Automatización de subida de informes a Power BI Embedded.....	17
5.1 Definición del proceso	17
5.2 Objetivo de la aplicación.....	17
5.3 Proceso de implementación	17
6. Integración de los informes en el portal web.....	20
6.1 Definición del proceso	20
6.2 Objetivos de la aplicación.....	20
6.3 Proceso de implementación	20
7. Listado de pantallas desarrolladas.....	23
7.1 Objetivos e indicadores generales.....	23
7.1.1 Indicadores y objetivos generales.....	23
7.1.2 Personas en un periodo (anual).....	24
7.1.3 Personas en un periodo (mensual).....	24
7.1.4 Personas por rol.....	25
7.1.5 Categoría y nivel.....	25
7.2 Salarios y nóminas.....	26
7.2.1 Costes reales.....	26
7.2.2 Salarios.....	26



7.2.3	Costes salariales por tipo y subtipo	27
7.2.4	Desviación de planificación vs realidad	27
7.2.5	Costes reales y presupuestados	28
7.3	Organización	29
7.3.1	Permanencia por áreas	29
7.4	Planificación de plantilla	30
7.4.1	FTE	30
7.4.2	Headcount	30
7.4.3	Evolución anual	31
7.4.4	Evolución mensual	31
7.4.5	FTE planificado vs real	32
7.4.6	Headcount planificado vs real	32
7.5	Formación	33
7.5.1	Formación por área	33
7.5.2	Formación por rol	33
7.6	Desempeño	34
7.6.1	Desempeño por áreas	34
7.6.2	Porcentaje de alto potencial	34
7.6.3	Número de personas con alto potencial	35
7.7	Dimensionamiento y personal no residente	36
7.7.1	Porcentaje de plantilla fija y temporal	36
7.7.2	Mapa de nacionalidades	36
7.7.3	Personal no residente	37
7.8	Rotación de plantilla	38
7.8.1	Recuento de bajas	38
7.8.2	Índice de rotación no deseada (mensual)	38
7.8.3	Índice de rotación no deseada (anual)	39
7.8.4	Índice de desvinculación no deseada (mensual)	39
7.8.5	Índice de desvinculación no deseada (anual)	40
8.	Bibliografía	41



1. INTRODUCCIÓN

1.1 Propósito del documento

En este documento presenta el proceso de trabajo llevado a cabo durante la fase de implementación del módulo de Business Intelligence.

Para la fase de implementación se parte del trabajo realizado en las fases anteriores análisis y diseño de una versión funcional del módulo de Business Intelligence dentro del portal web de Endalia [W3].

1.2 Alcance del documento

Este documento describe la fase de implementación del módulo de Business Intelligence. Esta fase es la posterior a la fase de diseño, y previa a la fase de pruebas del módulo.

1.3 Acrónimos

- API: Application Programming Interface.
- ASP: Active Server Pages.
- BI: Business Intelligence
- DAX: Data Analysis eXpressions.
- ETL: Extract, Transform and Load.
- HTML: HyperText Markup Language.
- IIS: Internet Information Services.
- JSON: JavaScript Object Notation.
- KPI: Key Performance Indicator.
- OLAP: On-Line Analytical Processing.
- OLTP: On-Line Transaction Processing.
- REST: Representational State Transfer.
- SGBD: Sistema Gestor de Base de Datos.
- SQL: Structured Query Language.
- T-SQL: Transact-Structured Query Language.



1.4 Definiciones

- Control de versiones: gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.
- KPI: Un KPI (key performance indicator), conocido también como indicador clave o medidor de desempeño o indicador clave de rendimiento, es una medida del nivel del desempeño de un proceso.
- 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 Resumen

Este documento describe el proceso de implementación del módulo de Business Intelligence. Se compone de los siguientes 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 de implementación del almacén de datos y el proceso ETL.
5. Se describe el proceso de implementación del programa para subir informes a la colección de Azure Power BI Embedded, y el proceso para automatizar esta tarea.
6. Se describe el proceso de implementación del módulo dentro del portal web de Endalia.
7. Por último, se muestra un listado de capturas de pantalla con todos los informes desarrollados para el módulo.



2. DESCRIPCIÓN DEL PROCESO

Partiendo del trabajo llevado a cabo en las fases de diseño y de análisis, se comienza con el proceso de implantación del sistema.

A la vez que se va desarrollando el módulo, se van realizando pruebas para comprobar el avance correcto del sistema. Estas pruebas se complementan con las pruebas de sistema realizadas tras finalizar la fase de implementación. Las pruebas de sistema están documentadas en la sección de pruebas adjunta en los anexos.

En la fase de implementación se distinguen tres fases principales:

- Implementación del almacén de datos mediante el proceso ETL.
- Implementación de la aplicación de escritorio para la automatización de subida de informes a la colección de informes de Power BI Embedded.
- Implementación de la integración de informes en el portal web, mediante la implementación del proyecto EmbedReports.

Las dos primeras fases permiten obtener una versión funcional del módulo, mientras que la tercera fase se ha llevado a cabo para facilitar el mantenimiento del módulo.



3. TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES

Para el desarrollo del presente proyecto se han utilizado las siguientes tecnologías, lenguajes y herramientas:

- ASP .NET. Plataforma de desarrollo de aplicaciones web, que se basa en ASP para la capa de presentación.
- C#. Lenguaje de programación de propósito general nativo de la plataforma .NET.
- IIS. Servidor de aplicaciones de Microsoft [W1] que permite gestionar servidores HTTP, HTTPS, FTP o SMTP, entre otros.
- JavaScript. Lenguaje interpretado de script para páginas web y ejecutado en el navegador.
- Log4Net. Herramienta para ayudar en la generación de ficheros de log.
- Microsoft .NET Framework. Versiones 4.0 y 4.5.
- Microsoft Power BI Embedded. Servicio de Microsoft Azure [W5] que permite incrustar informes de datos de Power BI en aplicaciones web.
- Microsoft SQL Server 2014. SGBD utilizado para la gestión de la base de datos del sistema y del almacén de datos.
- Microsoft SQL Server 2014 Management Studio. Herramienta gráfica que permite realizar tareas de mantenimiento de la base de datos sobre SQL Server 2014.
- Microsoft Team Foundation Server. Herramienta usada para la gestión del control de versiones de un proyecto sobre Visual Studio.
- Microsoft Visual Studio. Entorno de programación y depuración de código de la plataforma .NET.
- Navegadores: Microsoft Internet Explorer (versiones superiores a la 8.0), Mozilla Firefox y Google Chrome.
- NHibernate. Framework de mapeo objeto-relacional que facilita la representación de entidades relacionales de base de datos en objetos accesibles desde la aplicación desarrollada.
- Power BI Desktop. Herramienta de creación de informes interactivos. Admite conexión a bases de datos SQL Server como origen de datos.
- T-SQL. Lenguaje de acceso a datos basado en SQL, propio de Microsoft.



4. IMPLEMENTACIÓN DEL ALMACÉN DE DATOS

4.1 Definición del proceso

La primera parte del desarrollo del módulo es la creación de un almacén de datos, o Data Warehouse, que recoja todos los datos necesarios de manera estructurada para poder contestar a los indicadores planteados.

El almacén de datos es una base de datos OLAP, que provee una estructura de datos óptima para realizar consultas de agregado, que son las principales consultas para realizar análisis de grandes volúmenes de datos

4.2 Objetivos del almacén de datos

La creación del almacén de datos persigue conseguir los siguientes objetivos:

- Proporcionar una estructura que almacene toda la información de la organización de forma que ésta quede accesible, completa, estructurada y completa.
- Facilitar el acceso a información corporativa.
- Dotar de consistencia a la información de la organización.
- Ser el cimiento de partida para la toma de decisiones.
- Tener una estructura óptima para poder realizar consultas analíticas que puedan contestar a los indicadores propuestos.

4.3 Proceso ETL

El proceso ETL (Extraction-Transform-Load), es el proceso mediante el cual se obtiene un almacén de datos estructurado, a partir de un conjunto de datos origen.

Existen distintas herramientas para llevar a cabo este proceso. Dentro del ecosistema Microsoft existe la herramienta MicrosoftSQL Server Integration Services (SSIS) [W5]. Se realizó un estudio del uso de esta herramienta, desconocida hasta el momento en Endalia, pero se desestimó su uso por que añadía un grado extra de complejidad al desarrollo y mantenimiento del módulo.

La tecnología usada para el proceso ETL ha sido T-SQL, usada junto al entorno de desarrollo SQL Server 2014 Management Studio.

Esta tecnología ha sido elegida debido a que es un lenguaje muy potente y altamente indicado para trabajar con almacenes de datos dentro de SQL Server, además que T-SQL es un lenguaje con el que todos los desarrolladores de la organización están familiarizados.

El punto de partida de este proceso de implementación son la fase de análisis, donde se definen los indicadores objetivo, además de la localización de las tablas que alojan los datos necesarios, junto con la fase de diseño, donde se define la estructura que tendrá el almacén de datos.

Este proceso ETL se debe ejecutar sobre cada uno de los ocho data marts que componen el almacén de datos final. A continuación, se describe el proceso ETL de un data mart genérico.

4.3.1 Poblar dimensiones

Tras definir la estructura del almacén de datos, el primer paso es poblar las dimensiones que afectan al data mart. Excepto la dimensión de fechas que se ha creado mediante un script con funciones de tiempo de SQL Server, todas las demás



dimensiones se crean a partir de la extracción de datos de la base de datos origen. Los objetos referentes a tipologías, subtipologías, roles, etc... de los empleados están definidos en la base de datos origen en una tabla llamada 'Gen_AdministrationItems'. Esta tabla guarda la información de distintos tipos de datos parametrizables desde el sistema Endalia.

Para la creación de las dimensiones, se extraen los datos de esta tabla, se transforman en caso de que sea necesario (por ejemplo, nombres que empiezan por dos dígitos a modo de código y que en la dimensión no interesa que aparezca el código), y se cargan en la dimensión.

A continuación, se muestra un script de carga de una dimensión ejemplo, concretamente la dimensión de tipologías de empleado:

```
DELETE FROM [DimWorkTypes]

insert into [dbo].[DimWorkTypes]([WTypeName],[WTypeNameCA],[WTypeCode],[WTypeRemarks])
SELECT
    CONVERT(nvarchar(4000), DecryptByPassPhrase('fraseSecreta', AdmName)) as AdmName,
    AdmNameCA,
    AdmItemCode,
    AdmRemarks
FROM
    [Base_datos_origen].[dbo].[Gen_AdministrationItems]
WHERE
    fk_AdmParTypeID='TI' and len(AdmName)>0
ORDER BY
    AdmName

insert into [dbo].[DimWorkTypes]([WTypeName],[WTypeNameCA],[WTypeCode],[WTypeRemarks])
VALUES('',' ',NULL,'Desconocido')
```

Figura 1. Script de carga de la dimensión de tipologías de empleado

En el script, primero se eliminan los registros de la dimensión en caso de que existan, después se insertan todos los valores de las distintas tipologías de empleados almacenadas en la base de datos origen, y por último, se inserta un registro con nombre la cadena vacía, que será la tipología usada para aquellos empleados que no tengan una tipología asociada, ya que en la tabla de hechos no pueden existir valores nulos.

4.3.2 Poblar tablas de hechos

Pobladas todas las tablas de dimensiones, se procede a poblar la tabla de hechos del data mart correspondiente. Para ello, todos los scripts de carga de tablas de hechos siguen la misma estructura.

A continuación, se describe el proceso de carga de una tabla de hechos, concretamente la de objetivos e indicadores generales, aunque todas siguen el mismo proceso.

Declaración de variables y tablas temporales: Se declaran tablas temporales para el manejo de datos y variables auxiliares para manejar datos como fechas o cursores.



```

DELETE FROM FACTTABLEOIG

--Tablas auxiliares
declare @tableEmployees table(empid int, area nvarchar(200), WL nvarchar(255), IJCHA nvarchar(255), TI nvarchar(255), PG nvarchar(255),
declare @tableHistory table(id int identity, fk_EmpIntHisTypeID nvarchar(5),fk_EmpIntHisEmployeeID int,pk_EmpIntHisDate date, administra
declare @auxFactTable table(empid int, ddate date,area nvarchar(200), worklevel nvarchar(255),rol nvarchar(255), worktype nvarchar(255),
declare @inoutTable table(empid int, inDate date, outDate date)
declare @tableAreaHistory table(id int identity, EmpJobEmpID int, areaName nvarchar(250), EmpJobBeginDate date , EmpJobEndDate date)

--Variables auxiliares
declare @type nvarchar(5)
declare @empid int
declare @date date
declare @auxType nvarchar(5)
declare @auxValue nvarchar(255)
declare @auxEmpid int
declare @auxDate date
declare @beginningDate date
declare @endDate date

--Cursores
declare @InternalHistoryCursor int
declare @AreaHistoryCursor int

```

Figura 2. Declaración de variables auxiliares del script de carga de una tabla de hechos

La tabla temporal @auxFactTable contiene los mismos campos que la tabla de hechos, pero sin claves extranjeras, todos los campos en formato texto completo, es decir, el nombre del registro de la dimensión al que hace referencia.

La tabla temporal @tableEmployees guarda los datos de interés a guardar de los empleados, con los valores en formato texto. Esta tabla empieza con todos los campos vacíos, y a medida que se recorren las variaciones de estados se va actualizando. Tras recorrer todas las variaciones de un día, se insertan los registros de esta tabla de los empleados que están activos en la tabla de hechos auxiliar @auxFactTable.

La tabla temporal @tableHistory guarda los valores referentes a las variaciones de estado de las dimensiones que afectan al data mart. Guarda el tipo de dato, el empleado, el dato que tiene asociado el empleado, la fecha de inicio y la fecha fin del estado.

La tabla temporal @inoutTable guarda el periodo de tiempo que los empleados han estado activos en la empresa.

La tabla temporal @tableAreaHistory guarda el periodo de tiempo que un empleado ha estado asociado a un área en concreto.

Declaradas las tablas temporales, se procede a cargar los datos que contendrán estas tablas.



```

insert into @inoutTable(empid, inDate, outDate)
SELECT fk_EmployeeID , InDate, OutDate FROM Base_de_datos_origen.[dbo].Orh_Emp_EmployeeInOut
INNER JOIN [Base_de_datos_origen].[dbo].ORH_EMP_EMPLOYEES orh ON pk_EmpID=fk_EmployeeId

insert into @tableHistory(fk_EmpIntHisTypeID,fk_EmpIntHisEmployeeID,pk_EmpIntHisDate,administrationItemValue,EmpIntHisBeginDate )
select [fk_EmpIntHisTypeID],[fk_EmpIntHisEmployeeID],[pk_EmpIntHisDate],
(SELECT CONVERT(nvarchar(4000), DecryptByPassPhrase('fraseSecreta', AdmName)) as AdmName FROM Base_de_datos_origen.[dbo].Gen_
[EmpIntHisBeginDate]
from Base_de_datos_origen.[dbo].[R_Employee_InternalHistory]
where fk_EmpIntHisTypeID in ('CP', 'WL', 'TI', 'PG', 'IJCHA', 'CT' )
ORDER BY [EmpIntHisBeginDate] asc

insert into @tableAreaHistory(EmpJobEmpID, areaName, EmpJobBeginDate, EmpJobEndDate)
select fk_EmpJobEmpID,CASE WHEN d2.deptName!='' THEN d2.deptName ELSE d.deptName END ,pk_EmpJobBeginDate , EmpJobEndDate
from Base_de_datos_origen.[dbo].[R_Employee_Job]
cross apply Base_de_datos_origen.[dbo].GetJobDepartmentByJobIsDirector(fk_EmpJobJobID) d
cross apply Base_de_datos_origen.[dbo].GetJobDepartmentByJobIsDirector2(fk_EmpJobJobID) d2
ORDER BY pk_EmpJobBeginDate asc

insert into @tableEmployees(empid, area, WL, IJCHA, TI, PG, CP, CT, empIsActive, empInDate, empOutDate)
select empid, '', '', '', '', '', '', '', 0, inDate, outDate
FROM @inoutTable
inner join DimEmployees on pk_EmpID=empid

```

Figura 3. Carga de tablas temporales en un script de carga de tabla de hechos

Una vez ya están preparadas las tablas temporales, se procede a ejecutar un bucle que va a comprobar las variaciones de estado de los empleados registradas en la tabla temporal @tableHistory, y las variaciones de áreas de los empleados. El bucle se ejecuta una vez por cada día desde el 1 de enero de 1995, que es la fecha más antigua de en la que se tienen datos, hasta el día actual, pero por poca cantidad de datos antes del año 2010, el cliente pide que solo se guarden datos posteriores a este año.

```

set @auxDate='1995-01-01'
set @beginingDate='2009-12-31'
set @endDate = getdate()

while (@auxDate <> @endDate)
begin

    update @tableEmployees set empIsActive=0 where dateadd(day,1,empOutDate)=@auxDate
    update @tableEmployees set empIsActive=1 where empInDate=@auxDate

```

Figura 4. Declaración de variables auxiliares de las fechas que se usarán en el bucle

Una vez comienza el bucle, para cada día, lo primero es comprobar que empleados hay que poner como activos y cuales hay que poner como no activos en la tabla auxiliar @tableEmployees. Esta comparación se realiza comparando los datos almacenados anteriormente en la tabla @inoutTable.

A continuación, se comprueban las variaciones de estado que se han realizado el día que se está comprobando en el bucle. Las variaciones de estado están almacenadas en la tabla @tableHistory y se actualizan los valores nuevos en la tabla temporal @tableEmployees, como se muestra en la siguiente figura:



```

----Iteración sobre las variaciones de R_Employee_InternalHistory

DECLARE MY_CURSOR CURSOR LOCAL STATIC READ_ONLY FORWARD_ONLY FOR SELECT DISTINCT id from @tableHistory
WHERE EmpIntHisBeginDate=@auxDate
OPEN MY_CURSOR
FETCH NEXT FROM MY_CURSOR INTO @InternalHistoryCursor
WHILE @@FETCH_STATUS = 0
BEGIN
    SET @auxType=(select fk_EmpIntHisTypeID from @tableHistory where @InternalHistoryCursor=id)
    SET @auxValue=(select administrationItemValue from @tableHistory where @InternalHistoryCursor=id)
    SET @auxEmpid=(select fk_EmpIntHisEmployeeID from @tableHistory where @InternalHistoryCursor=id)

    IF(@auxType='CP')
    BEGIN
        UPDATE @tableEmployees SET CP=@auxValue WHERE EMPID=@auxEmpid
    END
    ELSE IF(@auxType='WL')
    BEGIN
        UPDATE @tableEmployees SET WL=@auxValue WHERE EMPID=@auxEmpid
    END
    ELSE IF(@auxType='TI')
    BEGIN
        UPDATE @tableEmployees SET TI=@auxValue WHERE EMPID=@auxEmpid
    END
    ELSE IF(@auxType='PG')
    BEGIN
        UPDATE @tableEmployees SET PG=@auxValue WHERE EMPID=@auxEmpid
    END
    ELSE IF(@auxType='IJCHA')
    BEGIN
        UPDATE @tableEmployees SET IJCHA=@auxValue WHERE EMPID=@auxEmpid
    END
    ELSE IF(@auxType='CT')
    BEGIN
        UPDATE @tableEmployees SET CT=@auxValue WHERE EMPID=@auxEmpid
    END

    FETCH NEXT FROM MY_CURSOR INTO @InternalHistoryCursor
END
CLOSE MY_CURSOR
DEALLOCATE MY_CURSOR

```

Figura 5. Comprobación de las variaciones de estado de los empleados

A continuación, se realiza el mismo proceso con las variaciones de áreas de empleados:

```

--Iteración sobre las variaciones de R_Employee_Job (areas)

DECLARE MY_CURSOR CURSOR LOCAL STATIC READ_ONLY FORWARD_ONLY FOR SELECT DISTINCT id from @tableAreaHistory
WHERE EmpJobBeginDate=@auxDate
OPEN MY_CURSOR
FETCH NEXT FROM MY_CURSOR INTO @AreaHistoryCursor
WHILE @@FETCH_STATUS = 0
BEGIN
    SET @auxValue=(select areaName from @tableAreaHistory where @AreaHistoryCursor=id)
    SET @auxEmpid=(select EmpJobEmpID from @tableAreaHistory where @AreaHistoryCursor=id)

    UPDATE @tableEmployees SET area=@auxValue WHERE EMPID=@auxEmpid

    FETCH NEXT FROM MY_CURSOR INTO @AreaHistoryCursor
END
CLOSE MY_CURSOR
DEALLOCATE MY_CURSOR

```

Figura 6. Comprobación sobre las variaciones de las áreas de los empleados

Una vez comprobadas las variaciones que se han producido a lo largo del día y registradas en la tabla temporal @tableEmployees, se copian todos los registros de los empleados que están activos de esta tabla a la tabla temporal @auxFactTable, añadiendo la fecha al estado del empleado. Esta copia de datos se lleva a cabo solo si la fecha es a partir del año 2010, ya que el cliente no quiere que aparezcan los datos anteriores ya que considera que hay poca cantidad de datos.




```

IF(DATEDIFF(DD,@beginningDate,@auxDate) > 0)
BEGIN
    insert into @auxFactTable (empid, ddate, area, worklevel, rol, worktype, worksubtype, empcp,company)
    select empid, @auxdate, area, WL, IJCHA, TI , PG, CP, CT from @tableEmployees where empIsActive=1
END

set @auxDate=DATEADD(DAY,1,@auxDate)
end

```

Figura 7. Inserción de datos de la tabla temporal @tableEmployees en la tabla temporal @auxFactTable

Una vez almacenadas las variaciones de los estados de los empleados en la tabla temporal @auxFactTable, se debe llevar a cabo la transformación de algunos datos, en función del data mart que se esté cargando. Entre las transformaciones de datos que se deben llevar a cabo, siempre se deben transformar los campos con valores de dimensiones nulos a campos con la cadena vacía (""), ya que en la tabla de hechos final no pueden existir campos nulos, y todas las dimensiones tienen un registro con la cadena vacía como nombre que hace referencia a que el valor es desconocido o que no tiene valor de dicha dimensión asociado.

```

UPDATE @auxFactTable SET rol='' where rol not in (select rolname from dimroles)
UPDATE @auxFactTable SET area='' WHERE area is NULL
UPDATE @auxFactTable SET worklevel=(SELECT REPLACE(worklevel,'Nivel ',''))
UPDATE @auxFactTable SET company=
    case (select fk_empcompany from Base_de_datos_origen.dbo.orh_emp_employees where empid=pk_EmpID)
        when 7 then 'Test 1'
        when 8 then 'Test 2'
        when 9 then 'Test 3'
        else ''
    end
WHERE company is NULL

UPDATE @auxFactTable set company='test' where company='Ftest'

```

Figura 8. Transformación de datos de la tabla @auxFactTable

Finalmente, una vez ya existen todos los datos preparados en la tabla temporal @auxFactTable, es momento de insertarlos en la tabla de hechos. Pero los campos de valores de dimensiones no se pueden almacenar tal como están en la tabla temporal @auxFactTable, éstos se deben transformar a su clave primaria en la dimensión. Para ello, a la hora de la inserción se realiza un join con cada tabla de la dimensión.

Cabe destacar que la duración media total del proceso de carga de un data mart es inferior a 2 minutos, y que actualmente cada una de las distintas tablas de hechos tienen aproximadamente dos millones y medio de registros cada una.



```

INSERT INTO FactTable(
    fk_EmpID
    , fk_Date
    , fk_CostCenter
    , fk_WorkLevel
    , fk_Rol
    , fk_WorkType
    , fk_WorkSubtype
    , fk_ProfessionalCategory
    ,fk_Company)
(
SELECT
    empid
    , dates.pk_DateID
    , centers.pk_CostCenterID
    , levels.pk_WorkLevelID
    , roles.pk_RolID
    , typ.pk_WorkTypeID
    , sub.pk_WorkSubtypeID
    , cp.pk_ProfessionalCategoryID
    , com.pk_CompanyID
from @auxFactTable aux
inner join [dbo].[DimDates] dates on dates.Ddate=aux.ddate
inner join [dbo].[DimCostCenters] centers on centers.CostCenterName=area
inner join [dbo].[DimWorkLevels] levels on levels.WLevelName=worklevel
inner join [dbo].[DimRoles] roles on roles.RolName=rol
inner join [dbo].[DimWorkTypes] typ on typ.WTypeName=worktype
inner join [dbo].[DimWorkSubTypes] sub on sub.WSubtypeName=worksubtype
inner join [dbo].[DimProfessionalCategories] cp on cp.ProfessionalCategoryName=empcp
inner join [dbo].[DimCompanies] com on com.CompanyName=company
)

```

Figura 9. Inserción de datos en la tabla de hechos

Poblado el almacén de datos, mediante la herramienta Power BI Desktop, ya se pueden crear informes que respondan a los indicadores planteados mediante distintos gráficos y filtros interactivos.

Power BI acepta como origen de datos bases de datos SQL Server, entre otras, y tiene un gran rendimiento a la hora de realizar consultas de agregado de datos, por lo que cada vez que se realiza algún filtro sobre un informe, la respuesta por parte de Power BI es inmediata.



5. AUTOMATIZACIÓN DE SUBIDA DE INFORMES A POWER BI EMBEDDED

5.1 Definición del proceso

Una vez creados los informes en Power BI, se procede con la segunda parte del desarrollo del módulo, que es la creación de una aplicación de escritorio que permita publicar en Azure [W5] estos informes en formato .pbix, en una colección de informes de Power BI Embedded.

Cabe señalar en este momento que esta tarea no se puede realizar simplemente pulsando un botón en el panel de administración de Azure [W5], o mediante Drag and Drop, sino que tiene que llevarse a cabo mediante un desarrollo de código. Desde el panel de administración de Azure, tampoco se pueden ver ni los informes que contiene cada colección, ni borrar estos.

A continuación, se describe el proceso de implementación de esta aplicación de consola que, mediante un menú, permite subir todos los informes de golpe a la colección de Power BI Embedded [W6].

5.2 Objetivo de la aplicación

La creación de la aplicación de consola persigue la consecución de los siguientes objetivos:

- Permitir subir los informes en formato .pbix de Power BI a la colección de Azure Power BI Embedded.
- Permitir consultar que informes se encuentran en cada colección y sus identificadores.
- Permitir eliminar informes de la colección de Power BI Embedded.
- Actualizar la información de la base de datos del cliente con los identificadores de los nuevos informes.
- Facilitar la subida de informes a la colección de Power BI Embedded.

5.3 Proceso de implementación

Para la creación de esta aplicación, se parte de la aplicación mostrada como ejemplo por Microsoft en el siguiente repositorio de GitHub [W4]:

<https://github.com/Azure-Samples/power-bi-embedded-integrate-report-into-web-app/tree/master/ProvisionSample>

La aplicación de ejemplo permite subir informes a la colección de Azure Power BI, pero de una manera ineficiente, ya que se deben llevar a cabo varios pasos para la subida de un informe, y por el contrario lo que interesa es que este proceso sea lo más automatizado posible.

Para subir un solo informe mediante la aplicación de ejemplo deben llevarse a cabo los siguientes pasos:

1. Crear un workspace para el informe.
2. Crear un dataset en el workspace.
3. Especificar la ruta en la que se encuentra el fichero .pbix
4. Además, se debe actualizar manualmente el identificador del informe en la base de datos del cliente.

Todo este proceso actualmente debería llevarse a cabo ocho veces, ya que en total hay ocho informes .pbix, con varias pestañas cada uno.

Como uno de los objetivos del desarrollo del módulo es que se consiga la máxima automatización posible, esta solución no era válida, por lo que se llevaron a cabo varias modificaciones para que los cuatro pasos anteriores se realizasen en uno solo, que además se hiciese para los ocho informes en una vez. Por lo tanto, se ha pasado de realizar 24 pasos de forma manual para la actualización de los informes, a un solo paso.



Para realizar esta mejora sobre la aplicación base, lo primero necesario es crear una clase que represente la entidad PowerBIReport, que además de implementar los métodos necesarios para la gestión de la entidad, tenga los siguientes atributos que permitan identificar un informe de Power BI:

- `_name`: Hace referencia al nombre del informe. Este nombre consiste en un identificador único de tres letras que hace referencia al proceso de la organización al cual hace referencia el informe. Ejemplo: SAL es el nombre del informe de salarios y nóminas, ROT hace referencia al informe de rotación de plantilla, etc...
- `_reportID`: Hace referencia al identificador único del informe en Power BI Embedded.

Junto a sus dos atributos, también se han declarado los siguientes métodos para la gestión de los informes:

- `GetPowerBIReports()`: Devuelve una lista con todos los informes (objetos PowerBIReport) almacenados en la base de datos, donde están almacenados en un campo en formato JSON ({ `_name` : `_reportID` }).
- `GetPowerBIReportByName(string name)`: Devuelve un objeto PowerBIReport cuyo nombre se el pasado como parámetro.
- `UpdateReportList(PowerBIReport report)`: Inserta un objeto PowerBIReport pasado como parámetro en la lista de todos los informes almacenado en la base de datos del cliente. En caso de que ya exista un informe con ese nombre, primero lo borra y después inserta el nuevo.
- `GetPowerBICollection()`: Devuelve el nombre de la colección de informes de Power BI Embedded.
- `GetPowerBIAccessKey()`: Devuelve la clave de acceso a la colección de informes de Power BI Embedded.
- `GetPowerBIReportsPath()`: Devuelve la ruta en la que están almacenados todos los informes de Power BI en formato .pbix

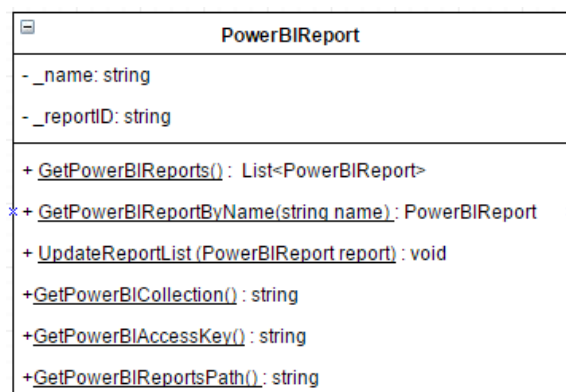


Figura 10. Diagrama de Clases [W2] de la clase PowerBIReport

Una vez la clase PowerBIReport se encuentra implementada, ya es posible modificar el programa de consola para poder subir todos los informes de golpe, además de conectar el programa con la base de datos del cliente para que actualice ésta automáticamente. Para ello, se ha desarrollado un método seleccionable en el menú de la aplicación que realiza los siguientes pasos para actualizar todos los informes.

1. Comprueba el nombre de la colección Azure Power BI Embedded y que la clave de acceso sea la correcta.
2. Obtiene una lista con todos los informes actuales almacenados en la base de datos mediante el método `GetPowerBIReports()`.
3. Actualiza, mediante el método `UpdateReportList(PowerBIReport report)`, cada uno de los informes de la lista, borrándolos e insertando los que se encuentran en la ruta donde se almacenan los informes en formato .pbix. Esta ruta se obtiene mediante el método `GetPowerBIReportsPath()` y se encuentra almacenada en la base de datos del cliente,

Se sigue un estándar a la hora de nombrar los informes mediante las tres letras del identificador seguido de la fecha en formato YYYYMMDD.



Tras ejecutar este método ya estarán actualizados todos los informes, tanto en la colección Azure Power BI Embedded como en la base de datos del cliente.

```
file:///C:/development/Integra/DEV/EndaliaBI/ProvisionReports/bin/Debug/ProvisionSample.EXE

What do you want to do (select numeric value)?
Select command group:
=====
1  Collection management
2  Report management
3  Misc.
4  Settings
1

What do you want to do (select numeric value)?
Current group = Collection management
You can use #1, #2 ... to quickly switch to another group
=====
1  Get Workspace Collections
2  Get metadata for a Workspace Collection
3  Get API keys for a Workspace Collection
4  Provision a new Workspace Collection
5  Get Workspaces within a collection
6  Provision a new Workspace
7  Update all Workspaces
8  Exit group
```

Figura 11. Programa de actualización de informes de Power BI

```
What do you want to do (select numeric value)?
Current group = Collection management
You can use #1, #2 ... to quickly switch to another group
=====
1  Get Workspace Collections
2  Get metadata for a Workspace Collection
3  Get API keys for a Workspace Collection
4  Provision a new Workspace Collection
5  Get Workspaces within a collection
6  Provision a new Workspace
7  Update all Workspaces
8  Exit group

7
Workspace Collection Name is required. Enter value: OIG170608
Access Key (optional). Enter value (or press Enter to = Auto select):
=====
Workspace Name:OIG170608      Workspace Id: 52B3D800-46E2-44B8-94D6-52B010101010
Checking import state... Publishing
Checking import state... Succeeded
Import: 0618-0000-0000-0000-0000-0000-0000-0000
Workspace Name:SAL170608      Workspace Id: 52B3D800-46E2-44B8-94D6-52B010101010
Checking import state... Publishing
Checking import state... Succeeded
Import: 0618-0000-0000-0000-0000-0000-0000-0000
Workspace Name:ORH170608      Workspace Id: 52B3D800-46E2-44B8-94D6-52B010101010
Checking import state... Publishing
Checking import state... Succeeded
Import: 0618-0000-0000-0000-0000-0000-0000-0000
Workspace Name:TRA170608      Workspace Id: 52B3D800-46E2-44B8-94D6-52B010101010
Checking import state... Succeeded
```

Figura 12. Proceso de actualización de informes



6. INTEGRACIÓN DE LOS INFORMES EN EL PORTAL WEB

6.1 Definición del proceso

Una vez creados los informes en Power BI, se procede con la última parte del desarrollo del módulo, que es la creación de una aplicación web que permita incrustar informes de Power BI en el portal web de Endalia.

Es necesario crear un proyecto web a parte del proyecto del portal web debido a que para embeber informes es necesario el uso de la API de Power BI, y ésta requiere una versión mínima del framework de .NET 4.5, mientras que el portal web está desarrollado sobre el framework .NET 4.0 y no es posible una subida inmediata por restricciones de tecnologías usadas en algunos clientes.

A continuación, se describe el proceso de implementación de esta aplicación web para embeber informes, que, a su vez, será embebida por el portal web de Endalia. Esta nueva aplicación se ha llamado EmbedReports.

6.2 Objetivos de la aplicación

La creación de la aplicación para embeber informes de Power BI persigue la consecución de los siguientes objetivos:

- Permitir visualizar con los informes de la aplicación Azure Power BI Embedded desde el portal web.
- Permitir interactuar con los informes de la aplicación Azure Power BI Embedded desde el portal web.

6.3 Proceso de implementación

Para la creación de esta aplicación, se parte de la aplicación mostrada como ejemplo por Microsoft en el siguiente repositorio de GitHub [W4]:

<https://github.com/Azure-Samples/power-bi-embedded-integrate-report-into-web-app/tree/master/EmbedSample>

Esta aplicación está desarrollada mediante el modelo arquitectónico Modelo-Vista-Controlador. También ha sido realizar varias modificaciones para poder adaptar la aplicación al portal web de Endalia.

La primera de estas modificaciones se ha realizado sobre la vista. En la vista original de la aplicación base se muestran unos campos de texto para escribir la colección Power BI Embedded [W6] a la que se quiere acceder, y la clave de acceso a esta. También posee un menú desplegable que permite elegir el informe que se quiera visualizar. Como el informe a visualizar se elegirá desde una entrada en el menú del portal web, se eliminan todos los campos de la vista excepto el campo que muestra el informe en cuestión, el cual pasa a ocupar toda la pantalla.

Como se han eliminado los campos de texto de la colección y de la clave de acceso, estos dos campos se almacenan en la base de datos, y son gestionados en la aplicación mediante la clase PowerBIReport desarrollada en el apartado anterior.

Otra parte de la aplicación que ha requerido ser construida de nuevo es el controlador de la aplicación. Este controlador antes actuaba en función de las peticiones del usuario desde los campos de texto y el desplegable con los informes disponibles. Ahora el controlador debe actuar en función de las peticiones que le llegan desde el portal web de Endalia, que es una aplicación web distinta.

Debido a que la aplicación EmbedReports recibe las peticiones desde el portal web de Endalia, se han añadido las entradas de cada informe en el menú principal del portal, como se puede ver a continuación:





Figura 13. Entradas del menú a cada uno de los ocho informes

Para gestionar el acceso a estos informes, se ha creado un formulario web en el portal web de Endalia, al cual se accede cuando se pulsa sobre las entradas del menú.

Este formulario hereda del formulario master del portal web, para mantener el diseño de Endalia en el módulo. El formulario está compuesto únicamente por un elemento iframe que apunta a la aplicación EmbedReports, y que ocupa la sección de contenido de la web.

Entonces, al ser cargado este formulario cuando se pulsa sobre alguna de las entradas del menú referentes a los informes, se realizan los siguientes pasos para realizar la petición del informe correspondiente a la aplicación EmbedReports:

1. Se lee de la base de datos la ruta de la aplicación EmbedReports en el servidor.
2. Se lee el nombre del informe que se solicita desde el menú. El nombre se encuentra encriptado mediante el método Rijndael al final de la ruta a la que se tiene que acceder tras pulsar sobre una de las entradas del menú. El nombre sigue siendo un identificador de tres letras.
3. Por último, se cambia la dirección a la que tiene que apuntar el iframe, que se forma mediante la ruta de la aplicación EmbedReports en el servidor, concatenado con el nombre del informe.

En este momento se realiza la petición desde el portal web de Endalia a la aplicación EmbedReports. Esta aplicación intermedia lleva a cabo los siguientes pasos para renderizar el informe solicitado por parte del usuario:

1. Lee el nombre del informe que se está solicitando, que se había añadido al final de la url a la que se realiza la petición.
2. Obtiene un objeto PowerBIReport con el nombre del informe y su identificador en la colección Power BI Embedded, almacenados en la base de datos del cliente, mediante el método de la clase PowerBIReport `GetPowerBIReportByName(name)` descrito anteriormente.
3. Lee el nombre de la colección de informes de Power BI Embedded, almacenado en la base de datos del cliente, mediante el método de la clase PowerBIReport `GetPowerBICollection()` descrito anteriormente.
4. Lee el nombre de la clave de acceso a la colección de informes de Power BI Embedded, almacenado en la base de datos del cliente, mediante el método de la clase PowerBIReport `GetPowerBIAccessKey()` descrito anteriormente.
5. Crea un cliente de acceso a la colección de informes de Power BI Embedded con el nombre de la colección y la clave de acceso mediante el método `CreatePowerBIClient()` de la API de Azure Power BI Embedded.
6. El cliente de acceso obtiene la información del informe solicitado de la colección, así como la url para embeberlo. Esta url solo es accesible mediante un token de seguridad.
7. Se crea el token de seguridad para acceder al informe desde la aplicación EmbedReports.
8. Se crea un objeto ReportViewModel de la API de Power BI, mediante la información del informe y el token de acceso.



9. Se devuelve una vista con el objeto ReportViewModel.
10. Finalmente el usuario ya puede ver el informe e interactuar con él.



7. LISTADO DE PANTALLAS DESARROLLADAS

A continuación, se muestran capturas de pantalla de todos los informes del módulo dentro del portal web de Endalia.

Los datos mostrados a continuación son ficticios, no corresponden a ninguna empresa real. Aún así, sirven como muestra de como quedarían los informes en una empresa real.

El detalle de todas las vistas y gráficos de los informes accesibles por parte de los usuarios se puede consultar en la sección “Manual de usuario” adjunta en los anexos.

7.1 Objetivos e indicadores generales

7.1.1 Indicadores y objetivos generales



7.1.2 Personas en un periodo (anual)



7.1.3 Personas en un periodo (mensual).



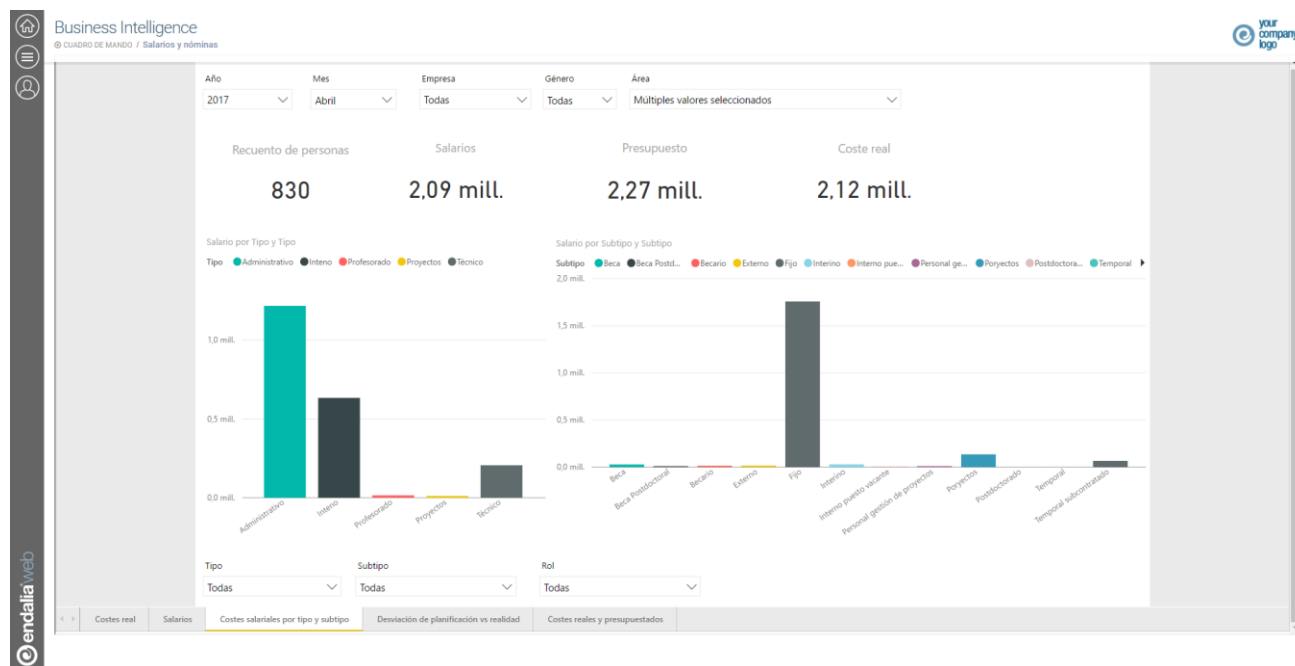
7.1.4 Personas por rol



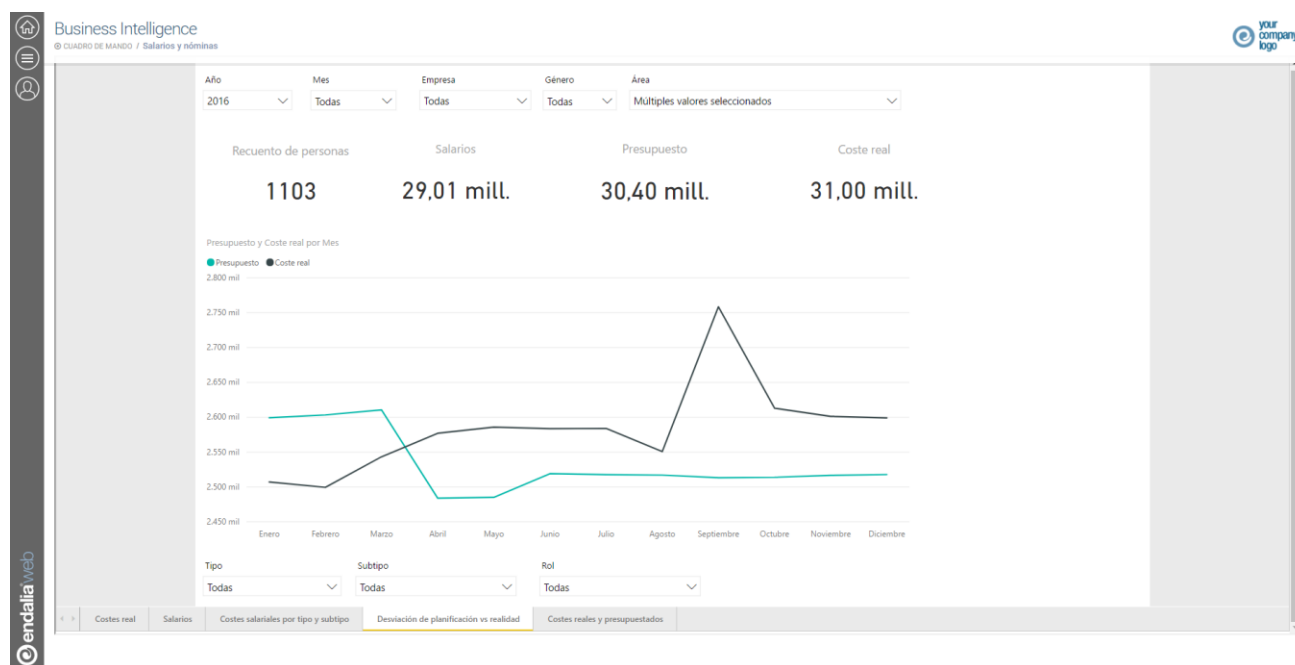
7.1.5 Categoría y nivel



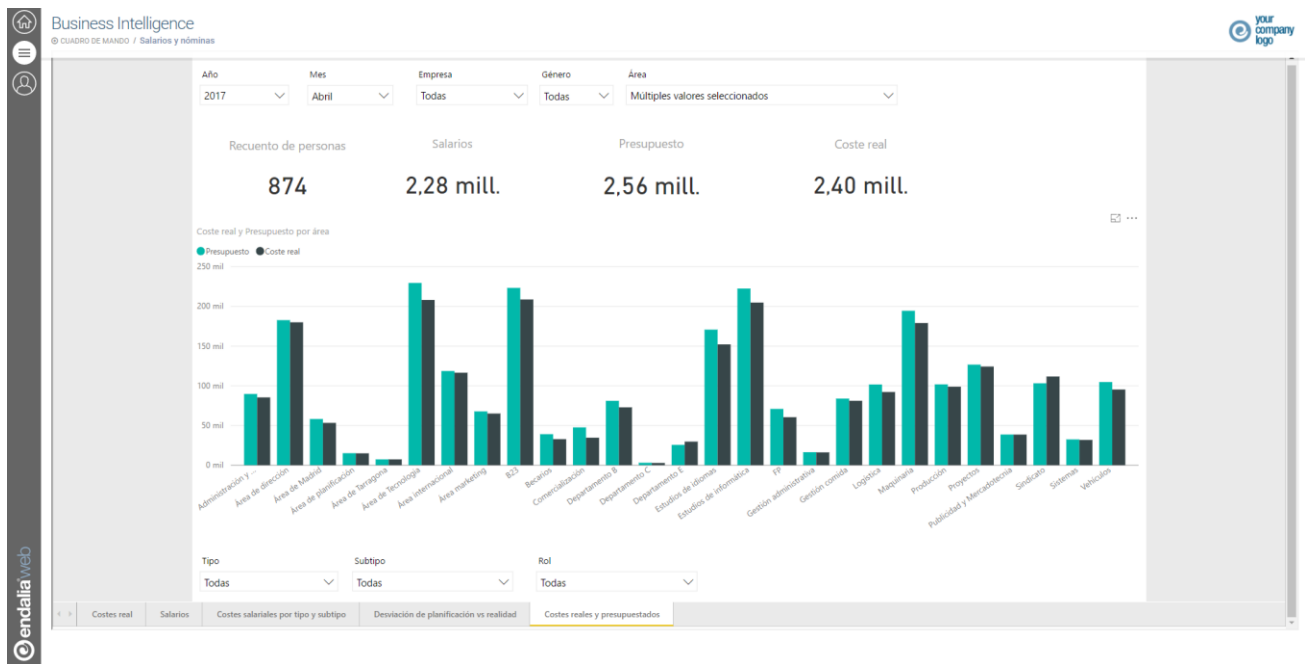
7.2.3 Costes salariales por tipo y subtipo



7.2.4 Desviación de planificación vs realidad

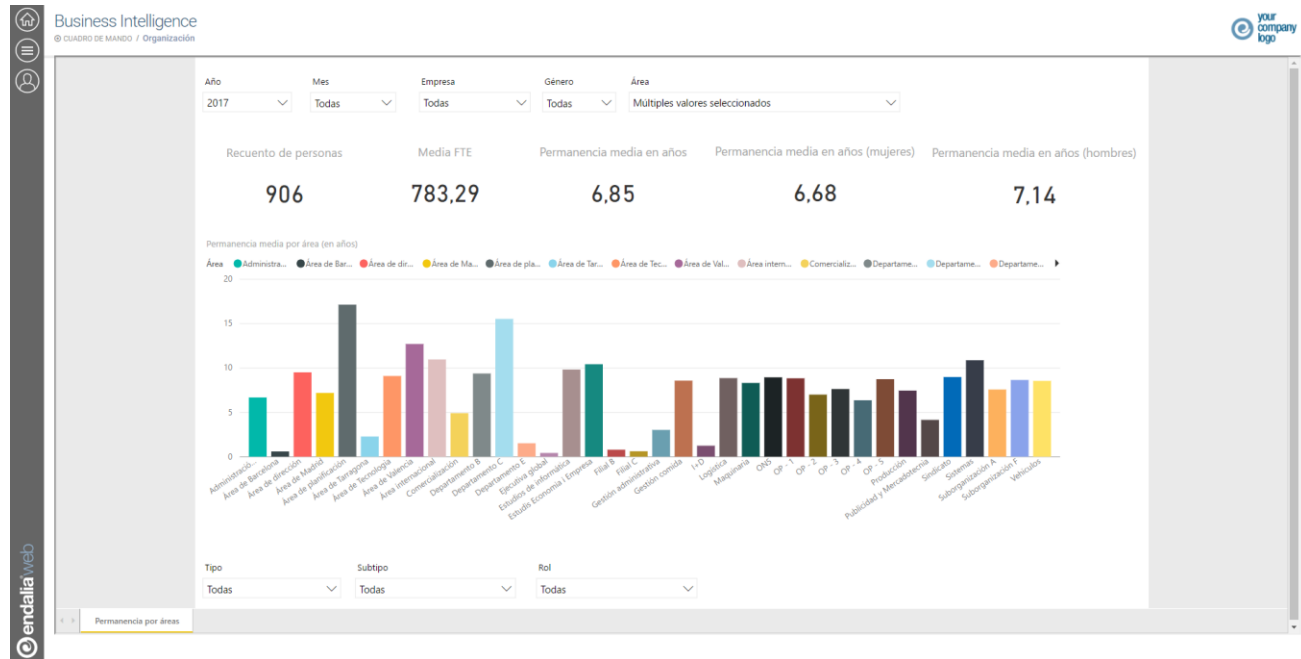


7.2.5 Costes reales y presupuestados



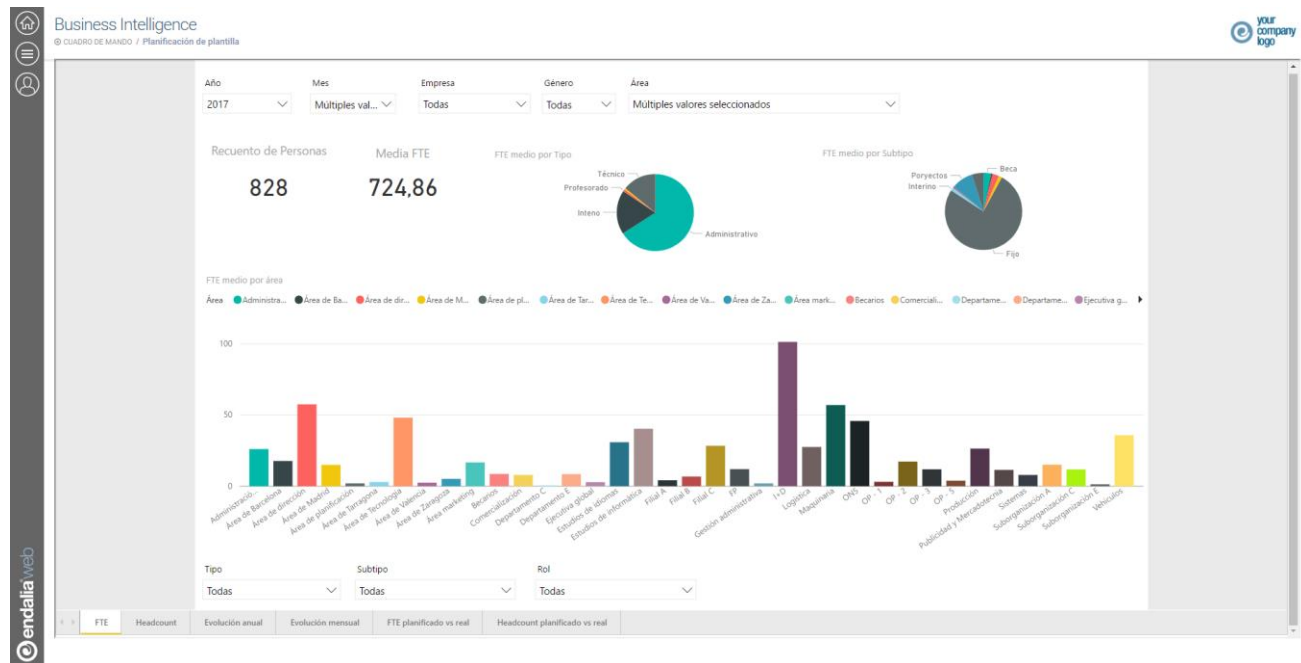
7.3 Organización

7.3.1 Permanencia por áreas

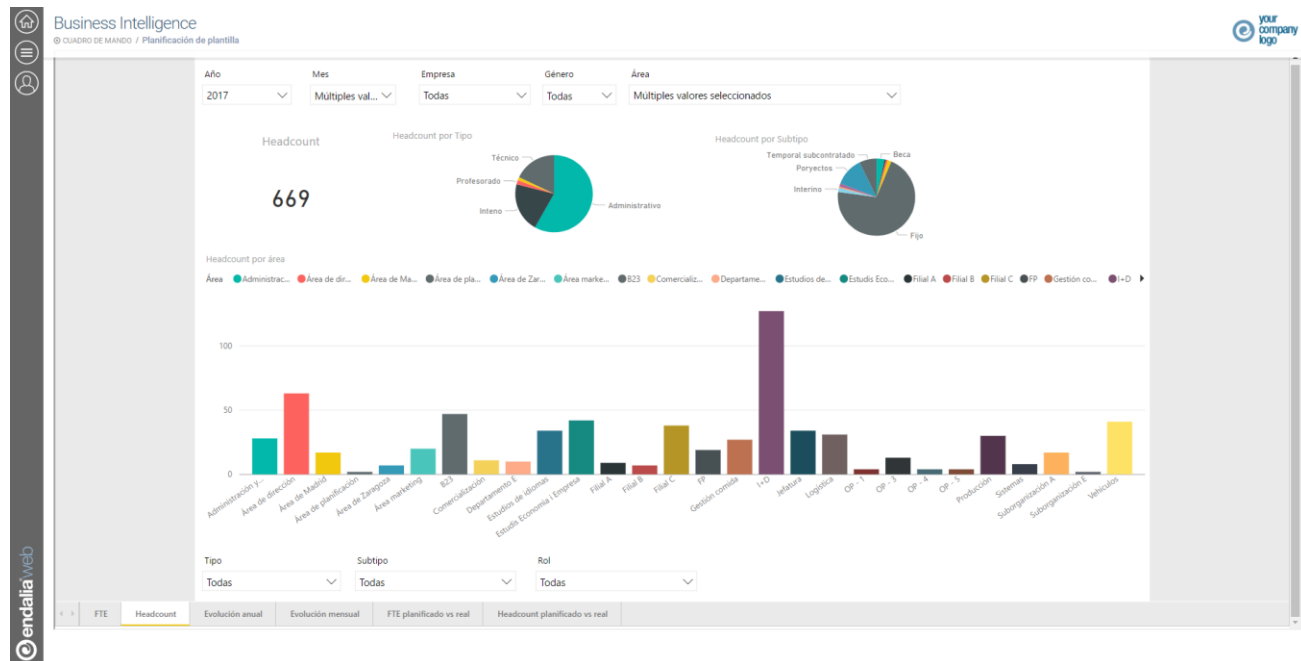


7.4 Planificación de plantilla

7.4.1 FTE



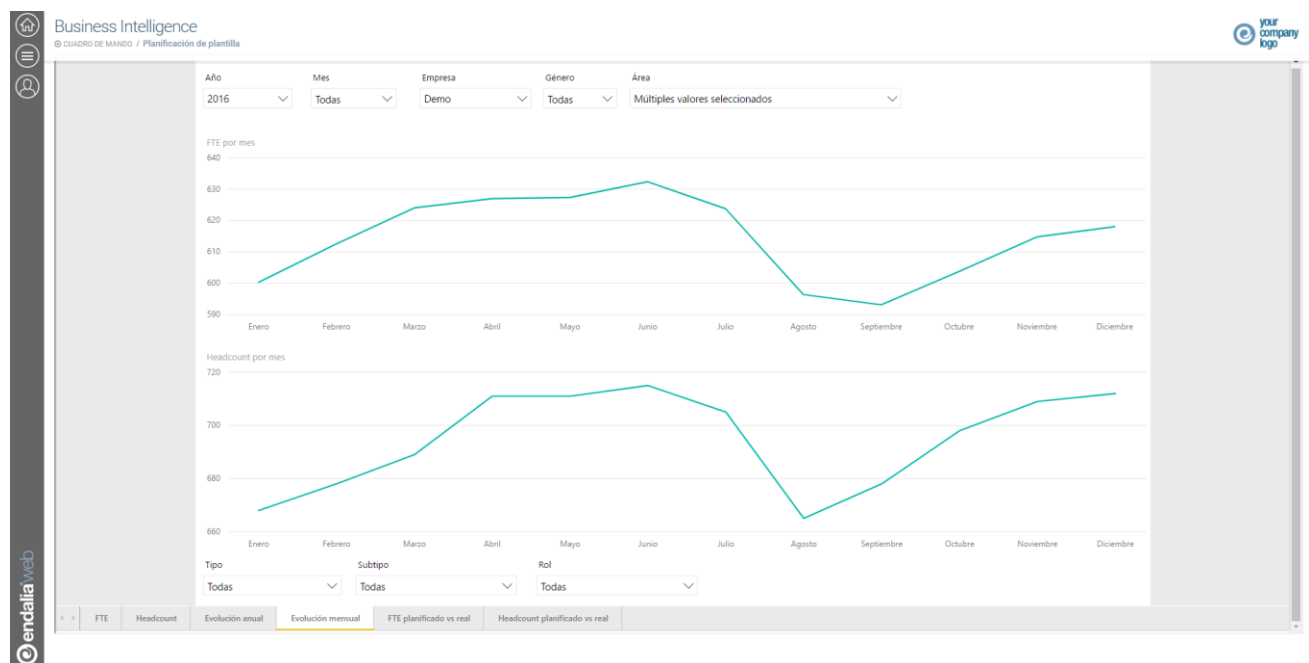
7.4.2 Headcount



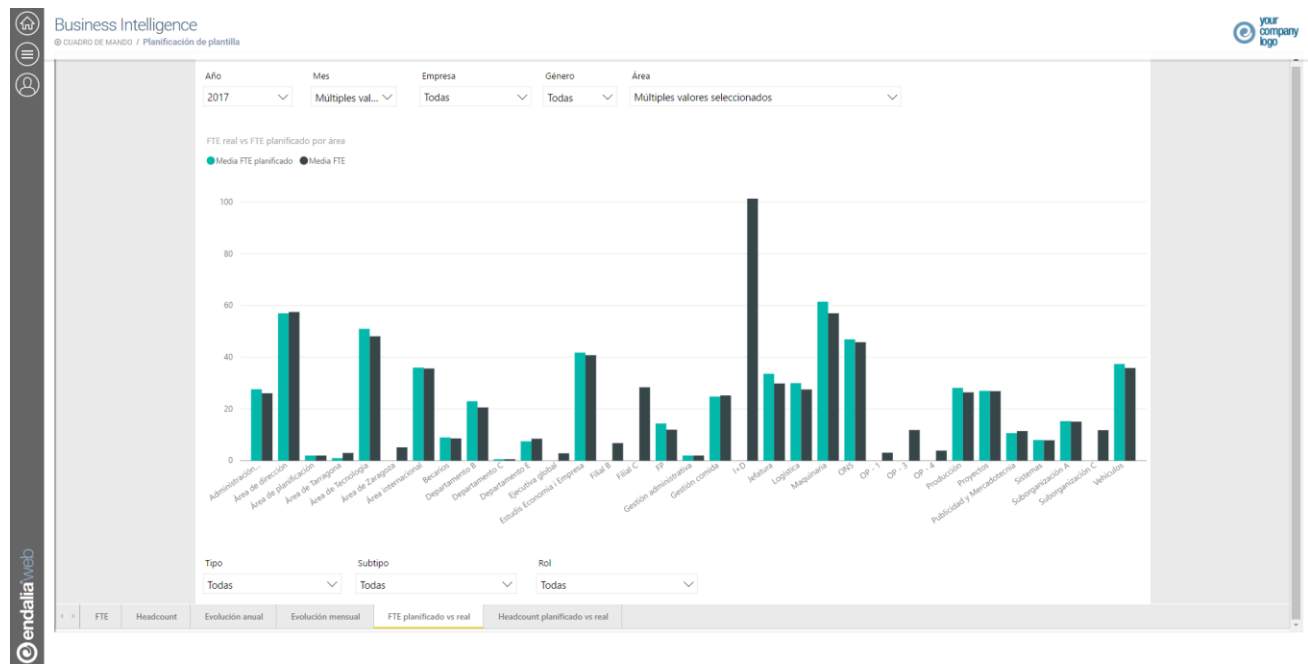
7.4.3 Evolución anual



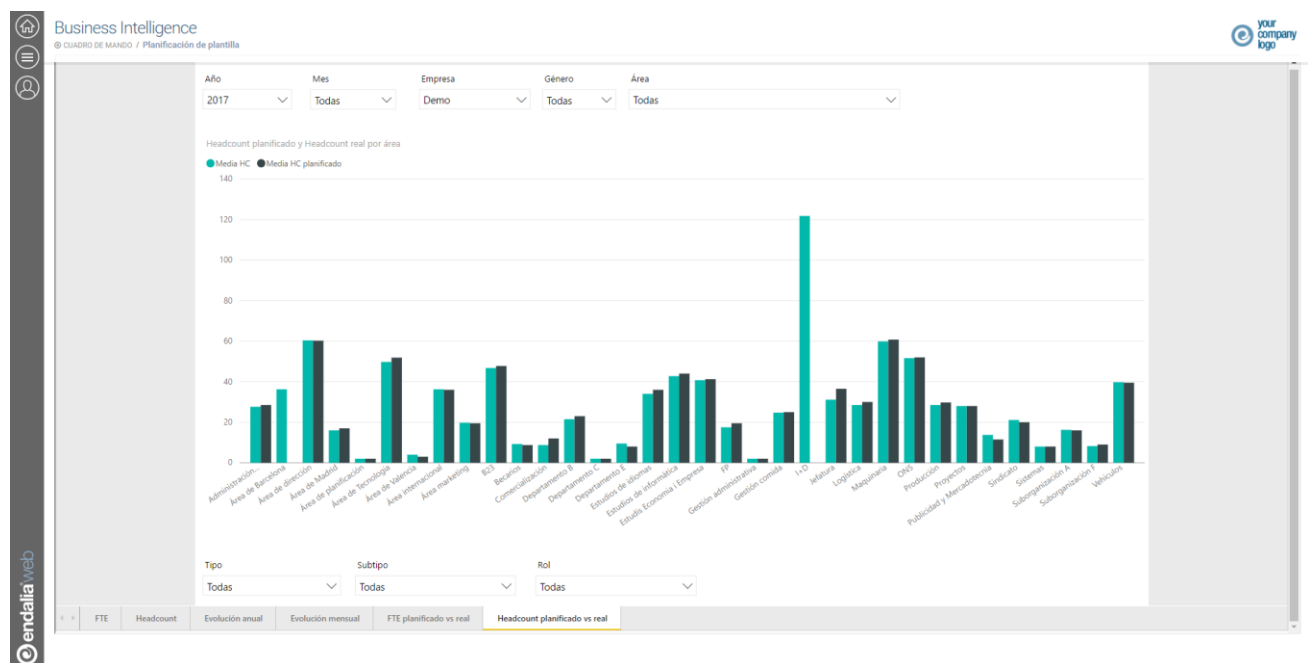
7.4.4 Evolución mensual



7.4.5 FTE planificado vs real

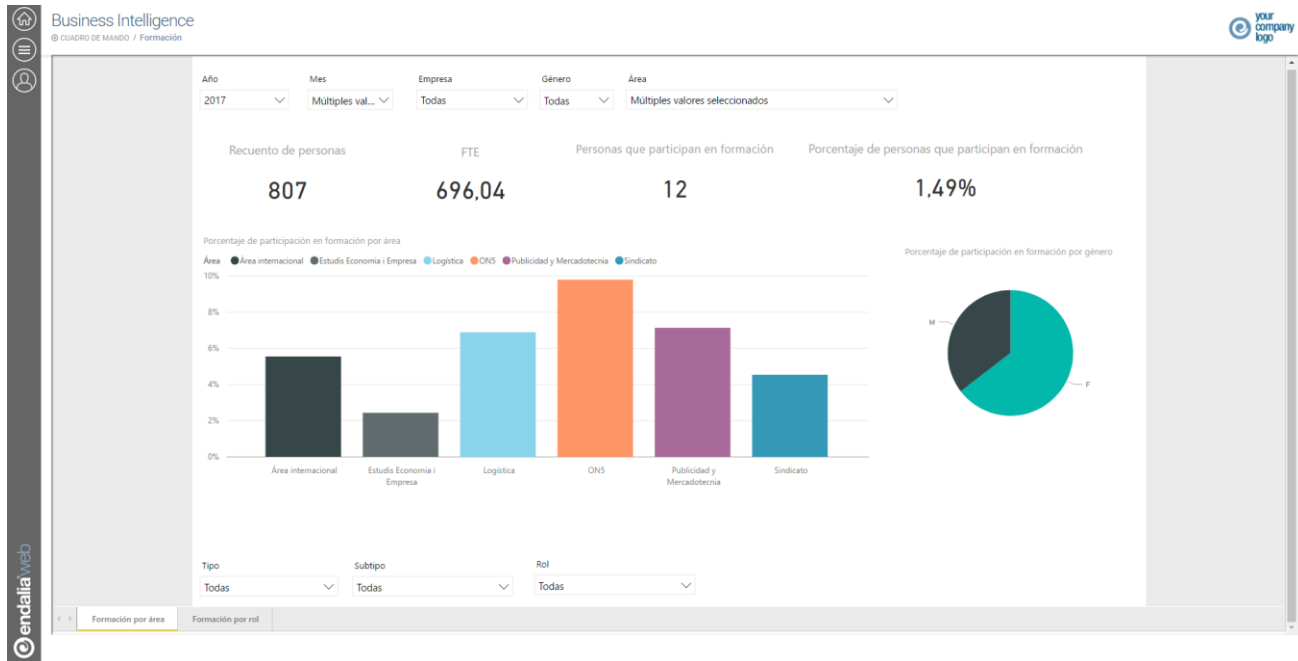


7.4.6 Headcount planificado vs real

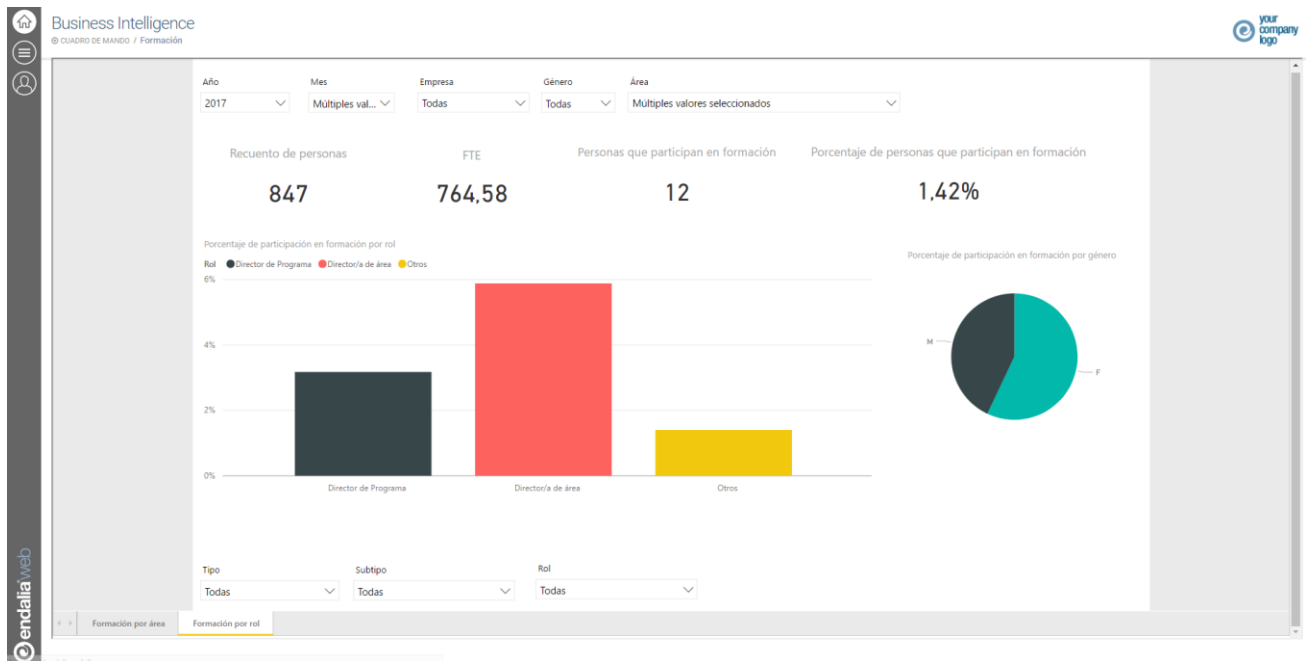


7.5 Formación

7.5.1 Formación por área

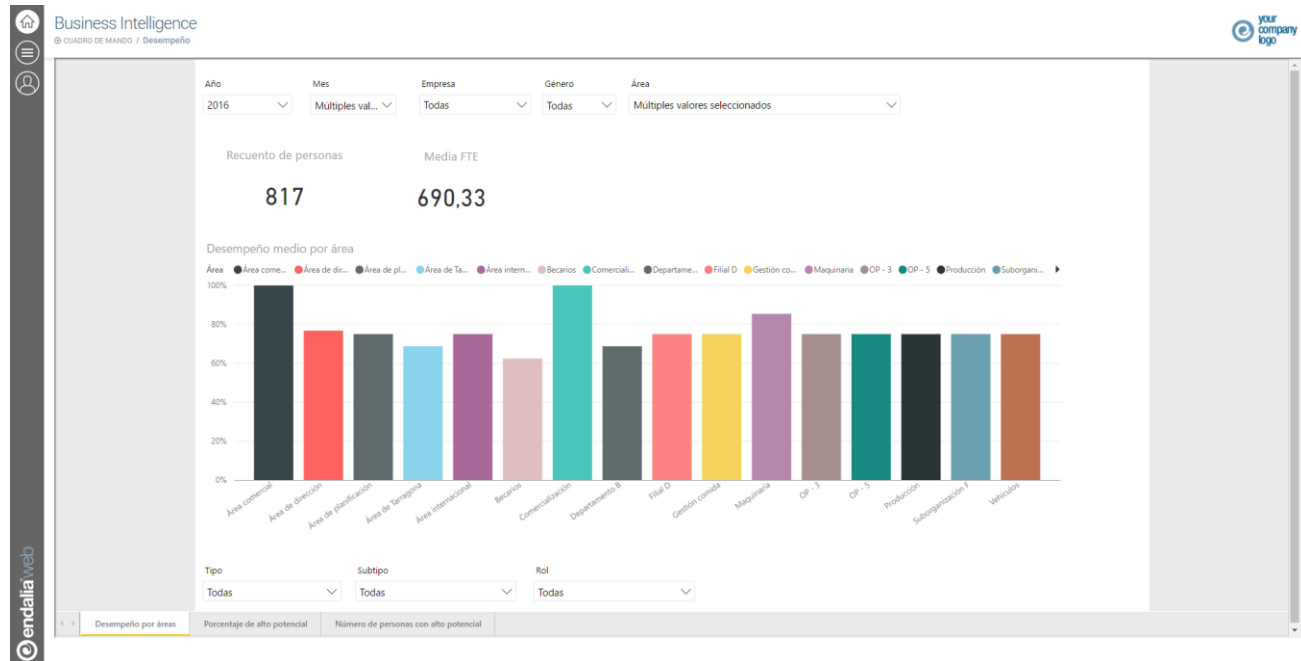


7.5.2 Formación por rol

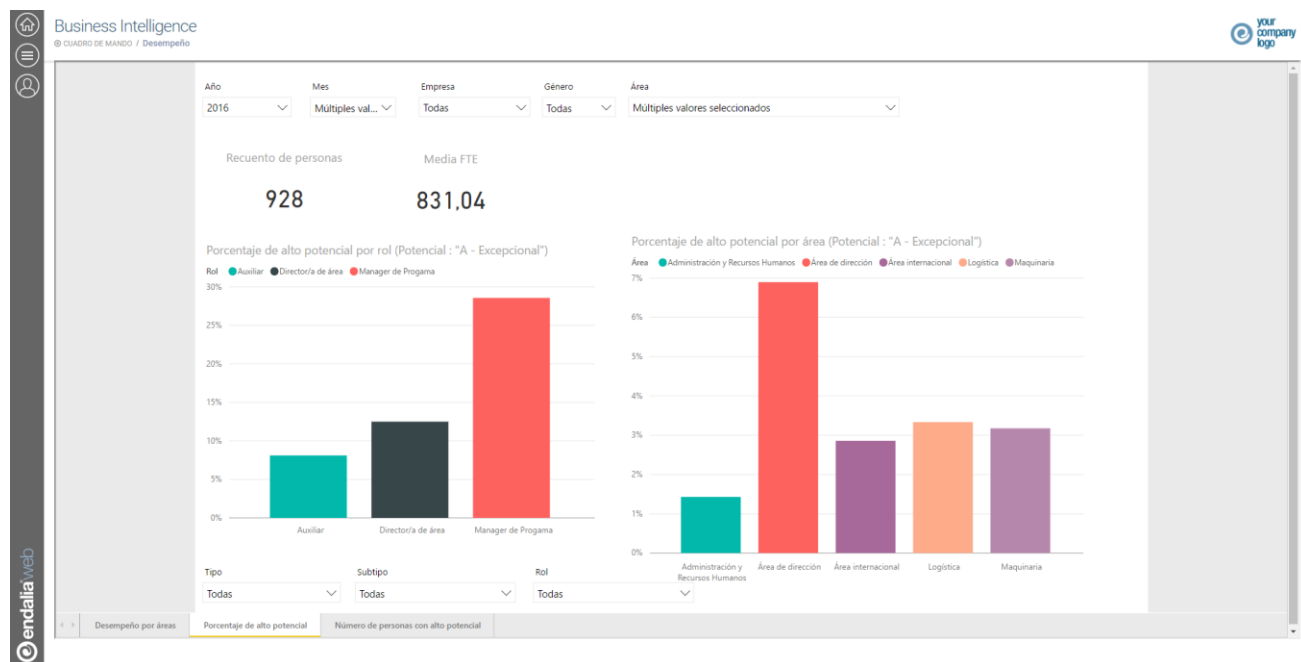


7.6 Desempeño

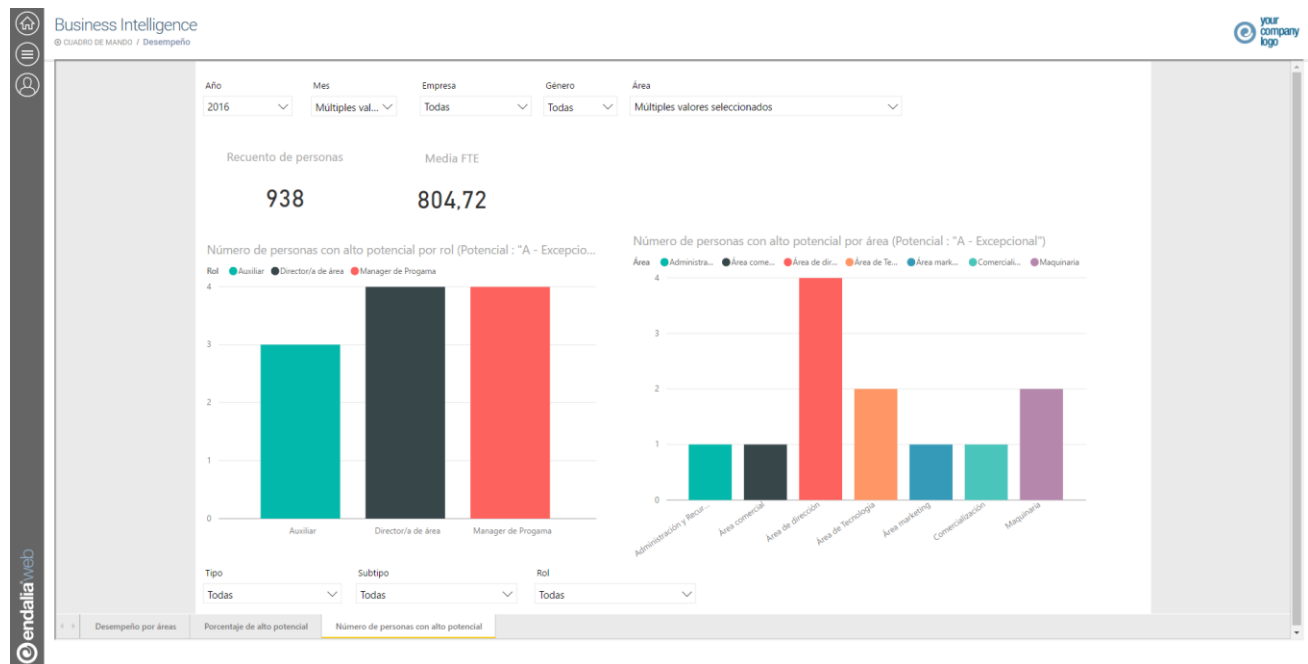
7.6.1 Desempeño por áreas



7.6.2 Porcentaje de alto potencial

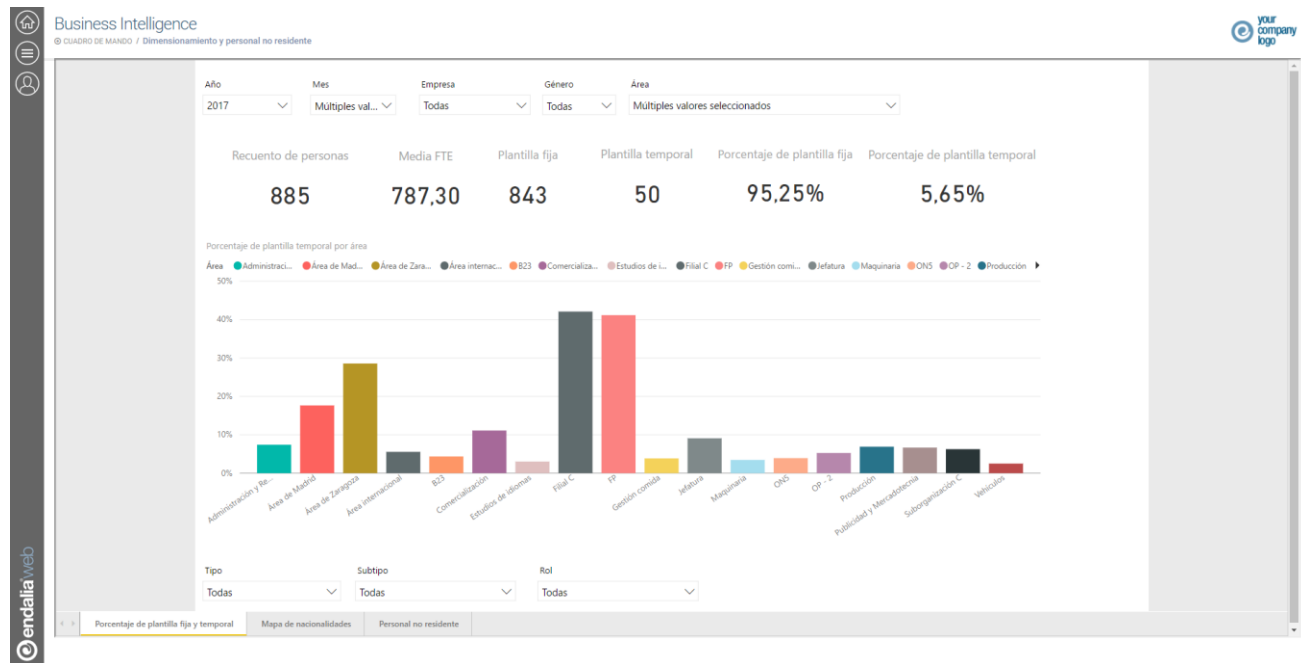


7.6.3 Número de personas con alto potencial

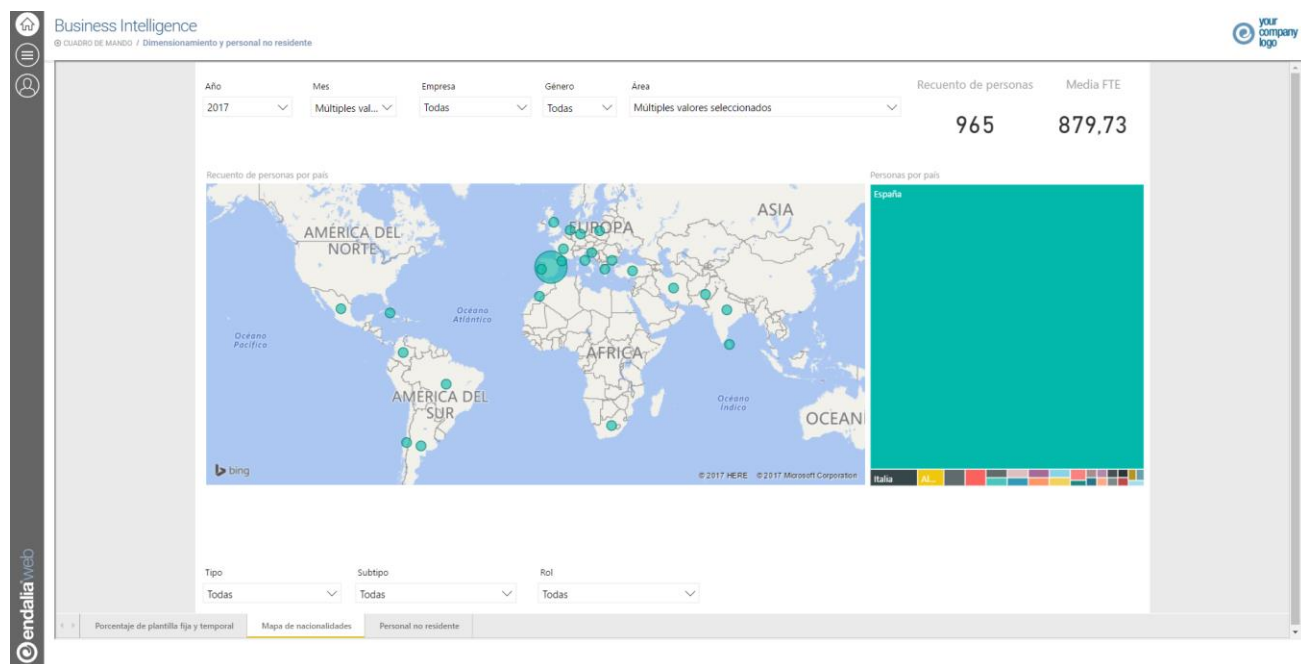


7.7 Dimensionamiento y personal no residente

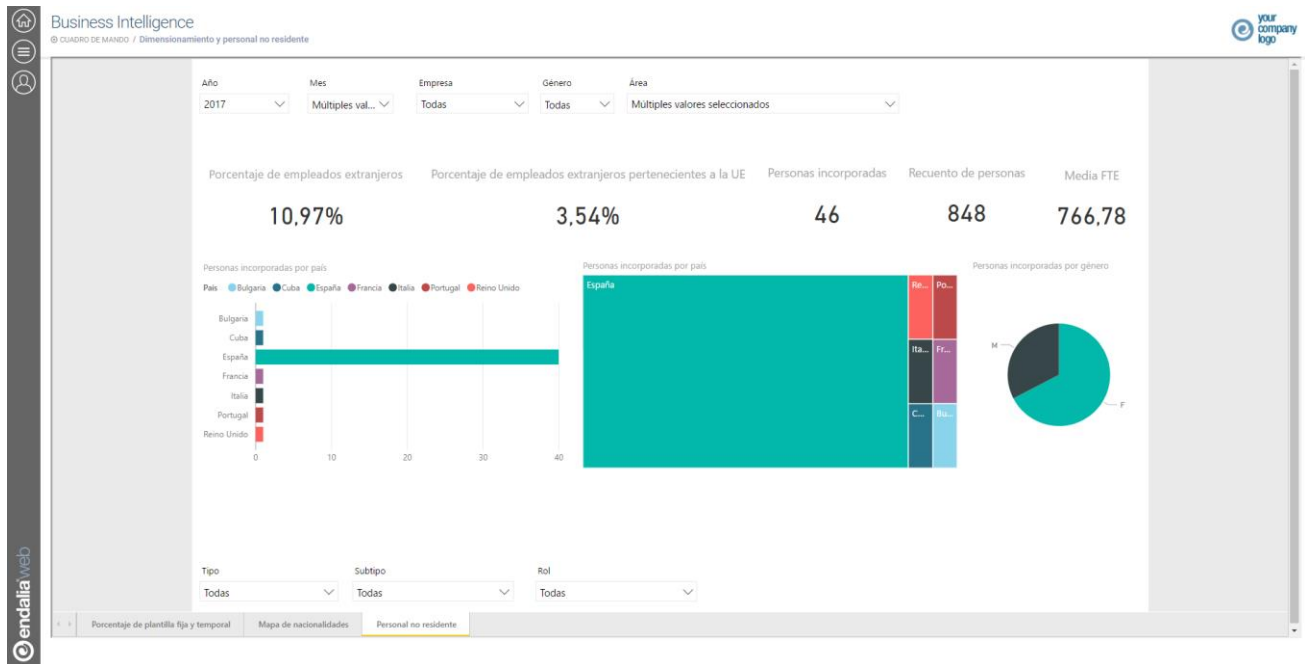
7.7.1 Porcentaje de plantilla fija y temporal



7.7.2 Mapa de nacionalidades

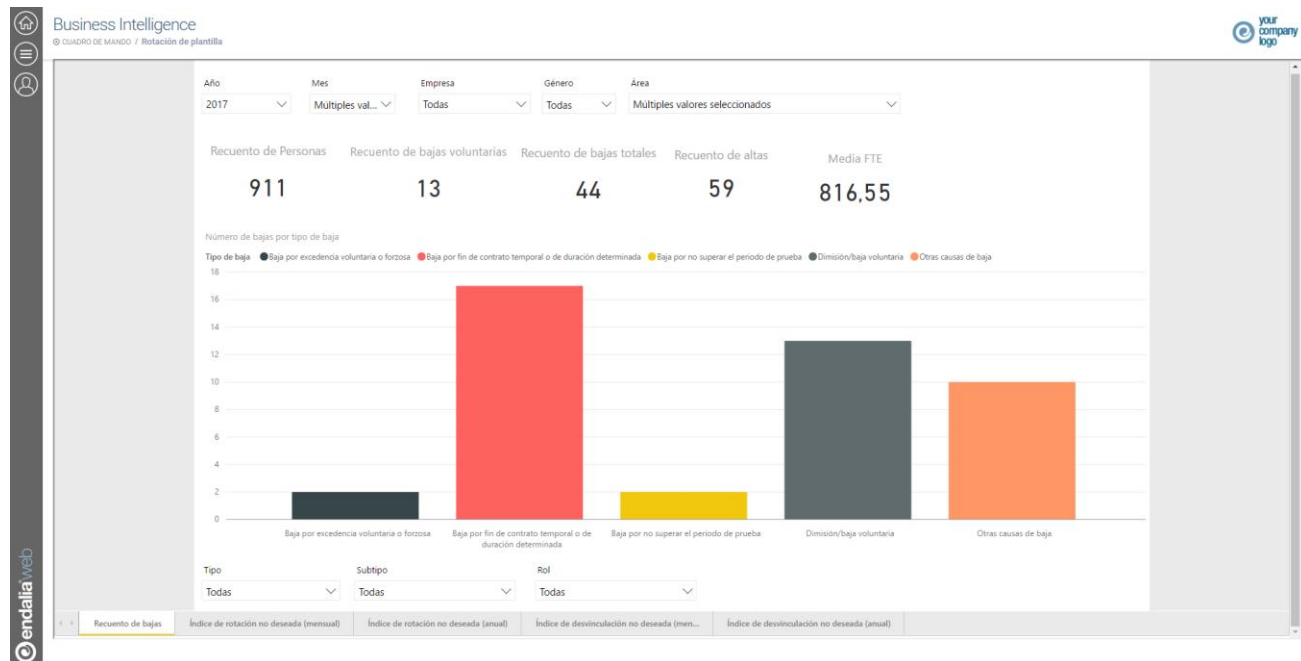


7.7.3 Personal no residente

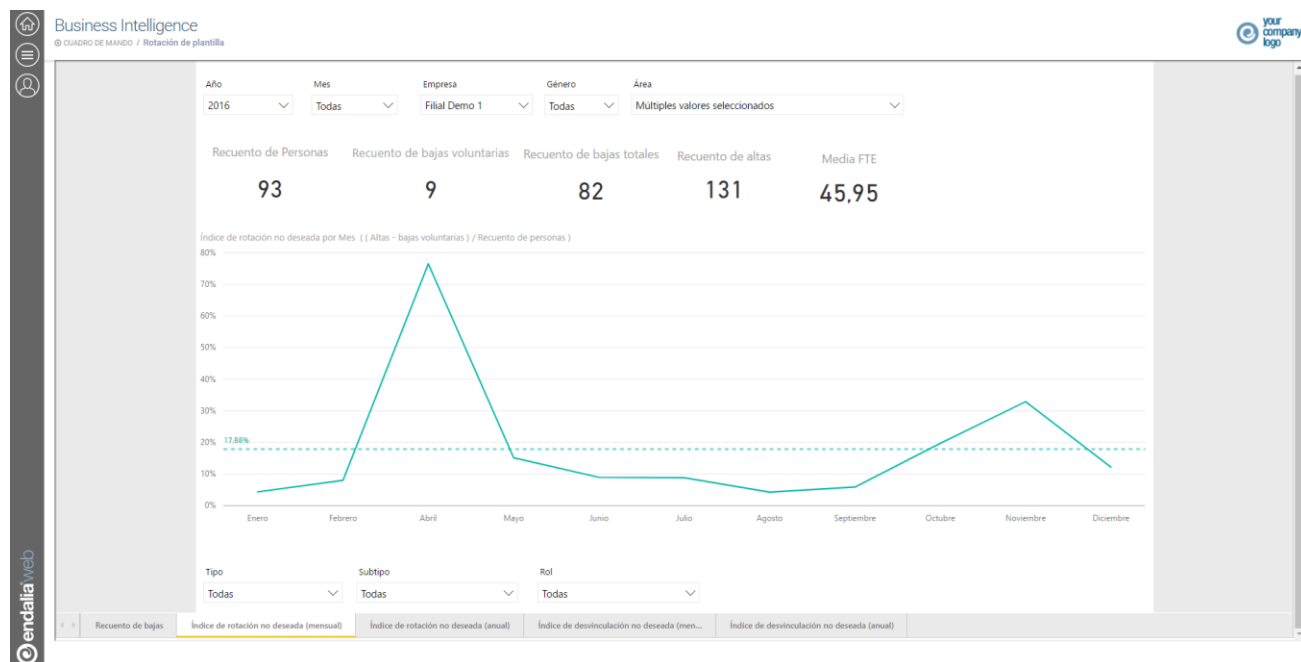


7.8 Rotación de plantilla

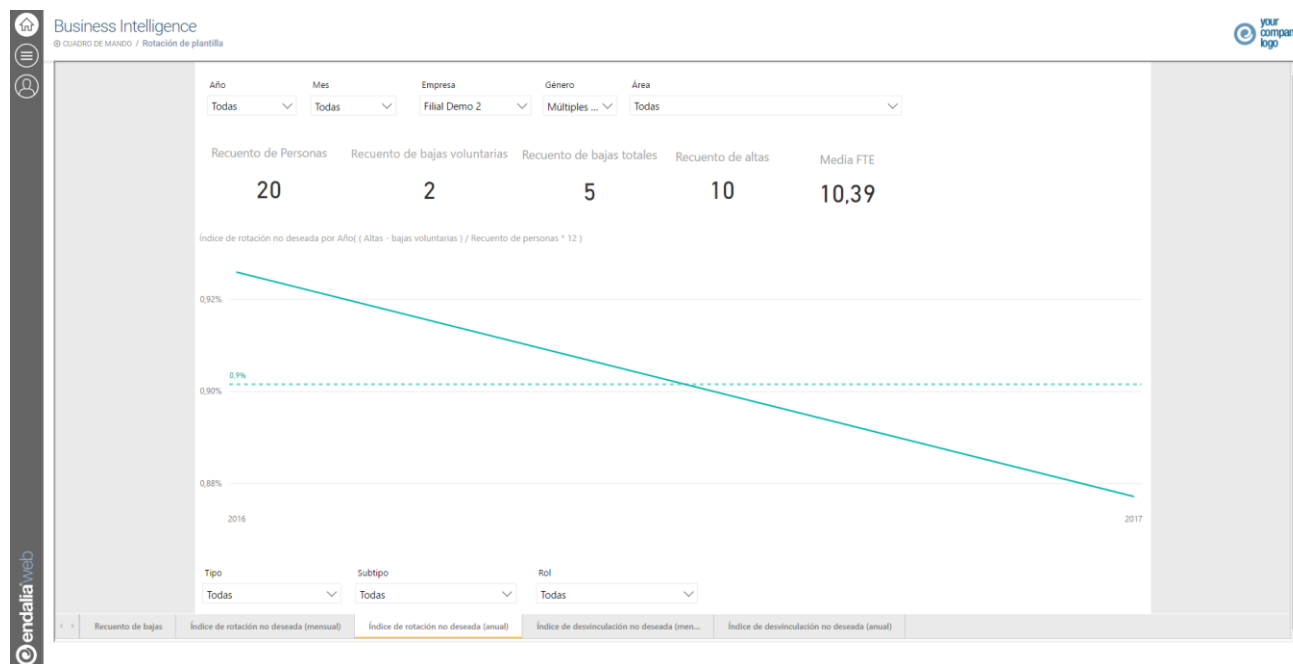
7.8.1 Recuento de bajas



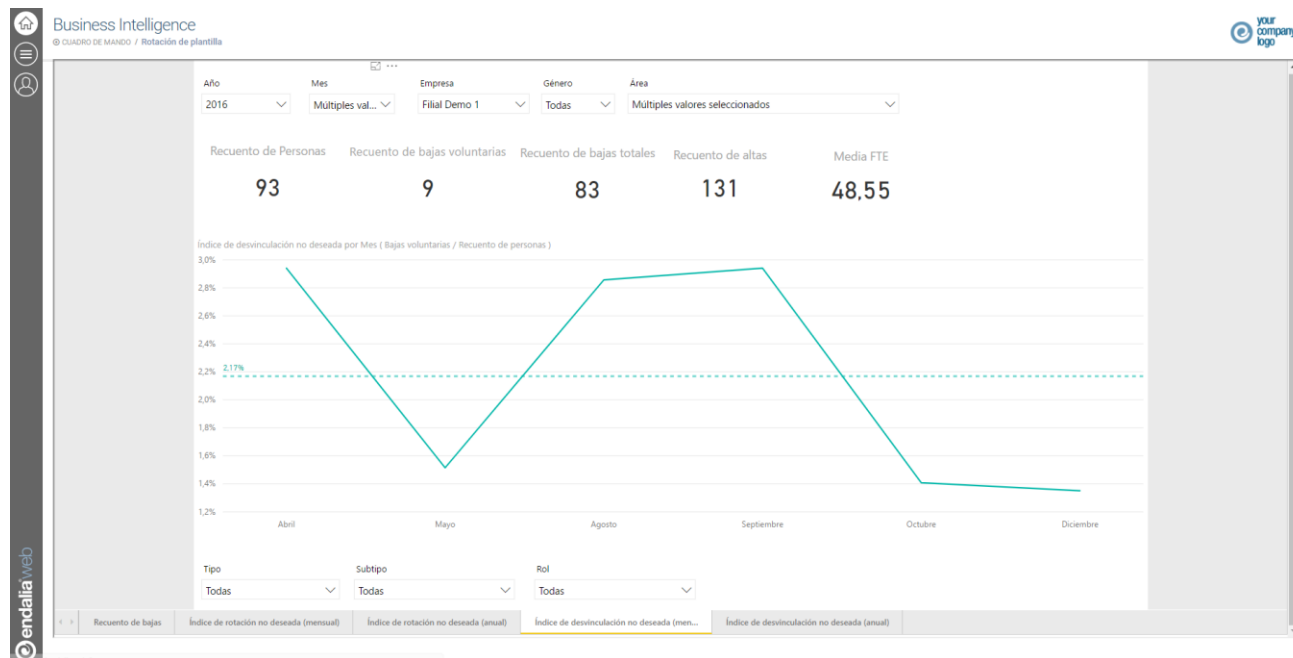
7.8.2 Índice de rotación no deseada (mensual)



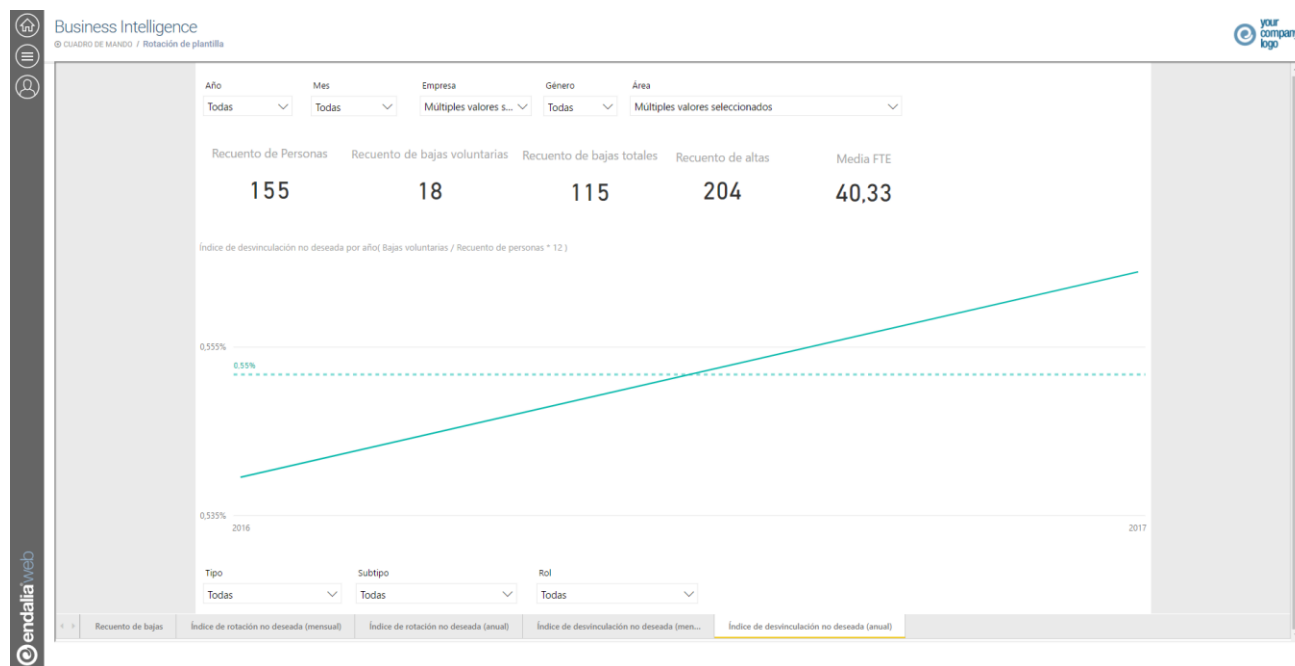
7.8.3 Índice de rotación no deseada (anual)



7.8.4 Índice de desvinculación no deseada (mensual)



7.8.5 Índice de desvinculación no deseada (anual)



8. BIBLIOGRAFÍA

[W1] <http://www.microsoft.com/net>

[W2] <http://www.uml.org>

[W3] <http://www.endalia.com>

[W4] <https://github.com/>

[W5] <https://azure.microsoft.com/es-es/>

[W6] <https://azure.microsoft.com/es-es/services/power-bi-embedded/>



PRUEBAS

DESARROLLO DE UN MÓDULO DE BUSINESS INTELLIGENCE

VERSIÓN 1.0
PUBLICADO EL 12/06/2017

Copyright © 2016 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
12/06/2017	1.0	Redacción inicial del documento	Víctor Beltrán Piñol



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 [W1].....	5
1.5 Resumen.....	5
2. Descripción del proceso.....	6
3. Pruebas de sistema.....	7
4. Bibliografía	8
4.1 Referencias web	8



1. INTRODUCCIÓN

1.1 Propósito del documento

El presente documento describe la fase de pruebas del Business Intelligence integrado dentro del portal web de Gestión Integral de Organización y Recursos Humanos de Endalia [W2]. El objetivo de esta fase es el de asegurar la calidad del software y el correcto funcionamiento del desarrollo del módulo, así como garantizar que se cumplen los requisitos planteados al comienzo de la realización de este trabajo.

1.2 Alcance del documento

Este documento contiene los resultados obtenidos en la última de las fases del trabajo: la realización de pruebas del módulo de Business Intelligence.

1.3 Acrónimos

- PS: Prueba de Sistema

1.4 Definiciones [W1]

- Caso de prueba: conjunto de condiciones o variables bajo las cuáles un analista determinará si una aplicación, un sistema software, o una característica de éstos es parcial o completamente satisfactoria.
- Prueba de sistema: conjunto de verificaciones de la interconexión de varios subsistemas dentro de una aplicación o sistema software.

1.5 Resumen

En este documento se presentan el proceso y los resultados de la fase de pruebas del Módulo de Gestión de Viajes y Gastos. Se compone de los cinco apartados siguientes:

1. Introducción, propósito y alcance del documento.
2. Descripción del proceso de pruebas realizado en este proyecto.
3. Descripción y resultados de las pruebas de sistema.
4. Presentación de la bibliografía utilizada para la realización de este documento.



2. DESCRIPCIÓN DEL PROCESO

Durante la implementación del módulo de Business Intelligence se han ido desarrollando distintas pruebas para asegurar el correcto funcionamiento del desarrollo. Además de las pruebas desarrolladas durante la implementación, se han llevado a cabo una serie de pruebas de sistema.

Por otro lado, al ser un módulo de análisis y explotación de datos también se han realizado distintas pruebas para comprobar que los datos que se muestran en los informes son correctos.

Para la comprobación de estos datos, se han seleccionado varios grupos de datos en cada informe, como por ejemplo datos de todos los empleados de un área o datos referentes a los empleados que tienen asociado un rol en concreto. Para estos grupos de datos se han realizado diversas consultas SQL contra la base de datos origen, para comprobar que las funciones de agregado que se muestran en el informe, muestran los datos correctamente.

Además de realizar estas pruebas para todos los informes, se ha tenido especial cuidado con los informes de costes reales y planificados, y FTE real y planificado. En estas dos vistas se han comprobado los datos que aparecen uno a uno, contrastándolo con un archivo en formato Excel donde se podía comprobar el resultado correcto de los datos agregados.

Mediante estas pruebas se han descubierto diversos casos especiales, que han permitido obtener los resultados correctos, mediante cálculos distintos a los que se realizan en el fichero Excel, ya que en el fichero Excel se calculan los totales a partir de las cantidades mensuales de los empleados, mientras que en los informes Power BI se calculan mediante el agregado de los valores diarios de los empleados.

Entre los casos más delicados para los cálculos se pueden destacar empleados, que aún estando inactivos siguen con nómina en la empresa, empleados que se planificaron pero finalmente no estuvieron activos, o empleados que ocupan múltiples puestos en distintas áreas de la organización.



3. PRUEBAS DE SISTEMA

Las pruebas de sistema tienen como objetivo validar que se cumplen correctamente los casos de uso definidos anteriormente en la fase de análisis del desarrollo.

Mediante estas pruebas se pretende probar y confirmar el correcto funcionamiento de los diferentes subsistemas que componen el módulo de Business Intelligence.

Identificador	PS – 01
Nombre	Consultar informes.
Descripción	Se ha de comprobar la correcta visualización de todos los informes que tiene acceso el usuario.
Resultados esperados	La aplicación debe mostrar el informe seleccionado por el usuario.
¿Prueba superada?	Sí.

Identificador	PS – 02
Nombre	Filtrar los datos.
Descripción	Se ha de comprobar que el informe posee distintos filtros que permiten filtrar los datos como el usuario considere necesario
Resultados esperados	La aplicación debe filtrar los datos por la selección de filtros del usuario.
¿Prueba superada?	Sí.

Identificador	PS – 03
Nombre	Disponibilidad de informes.
Descripción	Se ha de comprobar que las entradas del menú referentes a los informes solo está visibles por parte de los usuarios que tienen los privilegios correspondientes.
Resultados esperados	La aplicación debe mostrar las entradas de los informes en el menú solo en caso de que el usuario tenga privilegios para ello.
¿Prueba superada?	Sí.



4. BIBLIOGRAFÍA

4.1 Referencias web

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

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



MANUAL DE USUARIO DEL MÓDULO DE BUSINESS INTELLIGENCE

Copyright © 2015 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.



RESUMEN

Este documento presenta una descripción de las vistas y gráficas de desarrolladas en los diferentes informes disponibles en el módulo de Business Intelligence.

Para implantar este sistema se ha tenido en cuenta la naturaleza de las distintas necesidades reflejadas en los indicadores iniciales propuestos por entre Endalia y el cliente.

A continuación se procede a presentar las gráficas ordenadas por grupos de indicadores.



HISTÓRICO DE REVISIONES

Fecha	Versión	Descripción	Autor
12/06/2017	1.0	Redacción inicial del documento	Víctor Beltrán Piñol



ÍNDICE

RESUMEN.....	3
Histórico de revisiones.....	4
Índice	5
1. Filtros genéricos.....	9
1.1 Año.....	9
1.2 Mes.....	9
1.3 Empresa.....	9
1.4 Género	10
1.5 Área.....	10
1.6 Tipo.....	10
1.7 Subtipo	11
1.8 Rol.....	11
2. Tarjetas genéricas	12
2.1 Número de personas	12
2.2 Media FTE.....	12
3. Objetivos e indicadores generales	13
3.1 Filtros específicos de estos indicadores.....	13
3.1.1 Nivel:	13
3.1.2 Categoría:.....	13
3.2 Indicadores y objetivos generales.....	14
3.2.1 Personas por género:.....	14
3.2.2 Personas por Subtipo:	14
3.2.3 Personas por Tipo:.....	15
3.2.4 Personas por área:	15
3.3 Personas en un periodo (anual).....	16
3.3.1 Recuento de personas y media FTE por año:	16
3.4 Personas en un periodo (mensual).....	16
3.4.1 Recuento de personas y media FTE por mes	16
3.5 Personas por rol.....	17
3.5.1 Personas por rol:.....	17
3.6 Categoría y nivel	17
3.6.1 Personas por nivel.....	17
3.6.2 Personas por categoría.....	18
4. Salarios y nóminas	19



4.1	Tarjetas específicas	19
4.1.1	Salarios.....	19
4.1.2	Presupuesto	19
4.1.3	Coste real	19
4.2	Coste real.....	20
4.2.1	Coste real por área.....	20
4.3	Salarios	21
4.3.1	Salarios por mes	21
4.4	Costes salariales por tipo y subtipo.....	21
4.4.1	Salarios por tipo.....	21
4.4.2	Salarios por subtipo	22
4.5	Desviación de planificación vs realidad	22
4.5.1	Coste real y presupuesto por mes.	22
4.6	Costes reales y presupuestados.....	23
4.6.1	Coste real y presupuesto por área.....	23
5.	PLANificación de plantilla.....	24
5.1	FTE.....	24
5.1.1	FTE medio por tipo:.....	24
5.1.2	FTE medio por subtipo.....	24
5.1.3	FTE medio por área.....	25
5.2	Headcount	25
5.2.1	Headcount por tipo	25
5.2.2	Headcount por subtipo	26
5.2.3	Headcount por área	26
5.3	Evolución anual.....	27
5.3.1	FTE por año:.....	27
5.3.2	Headcount por año:	27
5.4	Evolución mensual.....	27
5.4.1	FTE por mes:.....	27
5.4.2	Headcount por mes:.....	28
5.5	FTE planificado vs real.....	28
5.5.1	FTE real vs FTE planificado por área:.....	28
5.6	Headcount planificado vs real.....	29
5.6.1	Headcount planificado y Headcount real por área:.....	29
6.	Organización	30
6.1	Tarjetas específicas	30



6.1.1	Permanencia media en años:.....	30
6.1.2	Permanencia media en años (mujeres):	30
6.1.3	Permanencia media en años (hombres):	30
6.2	Permanencia por áreas	31
6.2.1	Permanencia media por área (en años):.....	31
7.	desempeño / people review	32
7.1	Desempeño por áreas.....	32
7.1.1	Desempeño medio por área.....	32
7.2	Porcentaje de alto potencial	32
7.2.1	Porcentaje de alto potencial por rol (Potencial: “A - Excepcional”)	32
7.2.2	Porcentaje de alto potencial por área (Potencial: “A - Excepcional”).....	33
7.3	Número de personas con alto potencial.....	33
7.3.1	Número de personas con alto potencial por rol (Potencial: “A - Excepcional”):	33
7.3.2	Número de personas con alto potencial por área (Potencial: “A - Excepcional”):	34
8.	formación	35
8.1	Tarjetas específicas	35
8.1.1	Personas que participan en formación:.....	35
8.1.2	Porcentaje de personas que participan en formación:.....	35
8.2	Formación por área.....	35
8.2.1	Porcentaje de participación en formación por área:	35
8.2.2	Porcentaje de participación en formación por género:	36
8.3	Formación por rol.....	36
8.3.1	Porcentaje de participación en formación por rol:	36
9.	dimensionamiento y personal no residente.....	37
9.1	Tarjetas específicas	37
9.1.1	Plantilla fija:	37
9.1.2	Plantilla temporal:.....	37
9.1.3	Porcentaje de plantilla fija:	37
9.1.4	Porcentaje de plantilla temporal:.....	37
9.1.5	Porcentaje de empleados extranjeros:.....	38
9.1.6	Porcentaje de empleados extranjeros pertenecientes a la UE:.....	38
9.1.7	Personas incorporadas:	38
9.2	Porcentaje de plantilla fija y temporal	39
9.2.1	Porcentaje de plantilla temporal por área	39
9.3	Mapa de nacionalidades	39
9.3.1	Recuento de personas por país:	39



9.3.2	Personas por país:	40
9.4	Personal no residente	40
9.4.1	Personas incorporadas por país:	40
9.4.2	Personas incorporadas por país:	41
9.4.3	Personas incorporadas por género:	41
10.	rotación de plantilla.....	42
10.1	Tarjetas específicas	42
10.1.1	Recuento de bajas voluntarias:	42
10.1.2	Recuento de bajas totales:.....	42
10.1.3	Recuento de altas:	42
10.2	Recuento de bajas.....	43
10.2.1	Número de bajas por tipo de baja:.....	43
10.3	Índice de rotación no deseada (mensual).....	43
10.3.1	Índice de rotación por mes:	43
10.4	Índice de rotación no deseada (anual).....	44
10.4.1	Índice de rotación por año:	44
10.5	Índice de desvinculación no deseada (mensual).....	45
10.5.1	Índice de desvinculación no deseada por mes:	45
10.6	Índice de desvinculación no deseada (anual).....	46
10.6.1	Índice de desvinculación no deseada por año:.....	46



1. FILTROS GENÉRICOS

1.1 Año

Permite filtrar entre los datos de uno o varios años concretos.

Año

2017 ^

- ☐ Seleccionar t...
- ☐ 2010
- ☐ 2011
- ☐ 2012
- ☐ 2013
- ☐ 2014
- ☐ 2015
- ☐ 2016
- ☒ 2017

1.2 Mes

Permite filtrar entre los datos de uno o varios meses de los años seleccionados.

Mes

Todas ^

- ☐ Seleccionar todo
- ☐ Enero
- ☐ Febrero
- ☐ Marzo
- ☐ Abril

1.3 Empresa

Permite filtrar entre los datos de las personas pertenecientes a una empresa en concreto o varias. El campo vacío permite filtrar aquellas personas que no están asociadas a una empresa en el periodo de tiempo seleccionado.

Empresa

All ^

- ☐ Select All
- ☐
- ☐ Demo
- ☐ Filial Demo 1
- ☐ Filial Demo 2



1.4 Género

Permite filtrar los datos en función del género de las personas (F femenino, M masculino).

Género

All

☐ Select All

☐ F

☐ M

1.5 Área

Este campo permite filtrar por las distintas áreas de la estructura organizativa. El campo vacío permite filtrar aquellas personas que no están asociadas a un área en el periodo de tiempo seleccionado.

Área

All

☐ Select All

☐

☐ Administración y Recursos Humanos

☐ Área comercial

☐ Área corporativa

☐ Área de Barcelona

☐ Área de calidad

☐ Área de dirección

☐ Área de Madrid

1.6 Tipo

Permite filtrar por los distintos tipos que pueden tener asociados las personas. El campo vacío permite filtrar aquellas personas que no están asociadas a un tipo en el periodo de tiempo seleccionado.

Tipo

All

☐ Select All

☐

☐ Administrativo

☐ Inteno

☐ Profesorado

☐ Proyectos

☐ Técnico



1.7 Subtipo

Permite filtrar por los distintos subtipos que pueden tener asociados las personas. El campo vacío permite filtrar aquellas personas que no están asociadas a un subtipo en el periodo de tiempo seleccionado.

Subtipo

All

☐ Select All

☐

☐ Beca

☐ Beca Postdoctoral

☐ Becario

☐ Externo

☐ Fijo

☐ Interino

☐ Interno puesto vacante

1.8 Rol

Permite filtrar por los distintos roles que pueden tener asociados las personas. El campo vacío permite filtrar aquellas personas que no están asociadas a un rol en el periodo de tiempo seleccionado. En cambio, el rol “-” es un rol definido en el sistema y existen personas que se encuentran asociadas a dicho rol.

Rol

All

☐ Select All

☐

☐ Auxiliar

☐ Becario

☐ Cocina

☐ Contabilidad

☐ Delegado sede

☐ Director de docencia

☐ Director de Producción



2. TARJETAS GENÉRICAS

2.1 Número de personas

Muestra el número de personas distintas que se encuentran o se han encontrado activas durante el periodo de tiempo y demás filtros seleccionados.

Si filtramos por más de un mes, aparece el número de personas distintas que han estado en activas en el periodo seleccionado.

Ejemplo:

Al filtrar por los meses de enero y febrero del año 2017 aparecen 1078 personas en el recuento de personas. Si se filtra solo por el mes de enero se obtienen 1056 personas, mientras que si se filtra por febrero aparecen 1068 personas, de las cuales 22 personas no estaban en enero. Entonces el resultado final proviene de la suma del número de personas en enero más el número de incorporaciones en febrero: $1056 + 22 = 1078$

Número de personas

1071

2.2 Media FTE

Muestra la media del FTE (Full Time Equivalent) de las personas que aparecen en los filtros seleccionados para el periodo seleccionado. Ejemplo:

La media del FTE en enero de 2017 es de 967.15, mientras que la media del FTE para febrero de 2017 ha sido de 972.45. Si se filtra por los datos de enero y febrero de 2017 se obtiene una media de FTE de 969.80 , la cual se obtiene de la siguiente operación : $(967.15 + 972.45) / 2$

Media FTE

972,45



3. OBJETIVOS E INDICADORES GENERALES

3.1 Filtros específicos de estos indicadores

3.1.1 Nivel:

Permite filtrar por los distintos niveles que pueden tener asociados las personas. El campo vacío permite filtrar aquellas personas que no están asociadas a un nivel en el periodo de tiempo seleccionado.

Nivel

All ^

☐ Select All

☐ Nivel 01

☐ Nivel 02

☐ Nivel 03

☐ Nivel 04

☐ Nivel 05

☐ Nivel 06

☐ Nivel 07

☐ Nivel 08

3.1.2 Categoría:

Permite filtrar por los distintos categorías (Grupo profesional) que pueden tener asociados las personas. El campo vacío permite filtrar aquellas personas que no están asociadas a una categoría en el periodo de tiempo seleccionado.

Categoría

All ^

☐ Select All

☐

☐ Jefes Administrativos

☐ Ayudante no Titulado

☐ Jefe de taller

☐ Auxiliar administrativo A

☐ Auxiliar administrativo B

☐ Auxiliar administrativo C

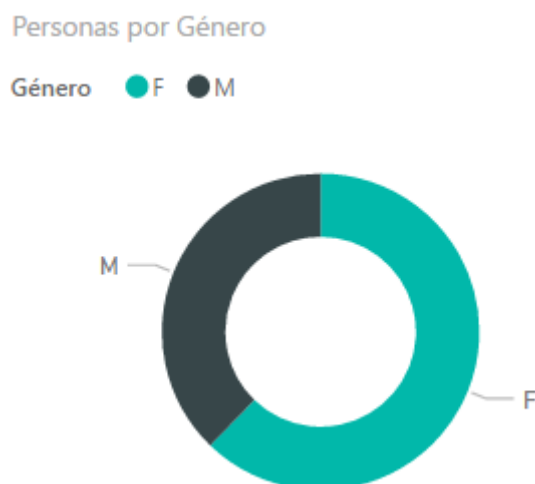


3.2 Indicadores y objetivos generales

3.2.1 Personas por género:

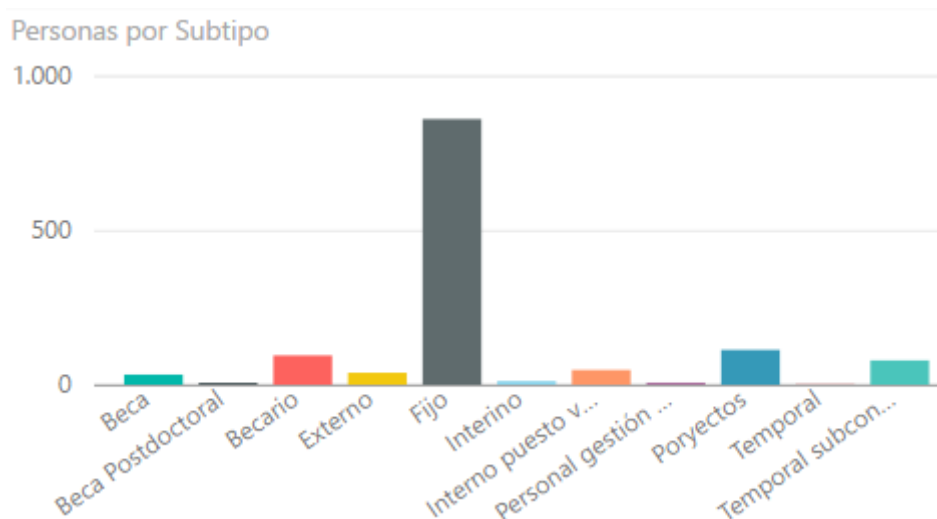
Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por género.

Ejemplo: Seleccionando los meses de enero a abril de 2017 en el recuento de personas aparecen 1118. Si se pulsa sobre cada uno de los géneros del gráfico o se pasa el ratón por encima, se puede apreciar que en total han estado activas 696 mujeres y 422 hombres (1118 en total)



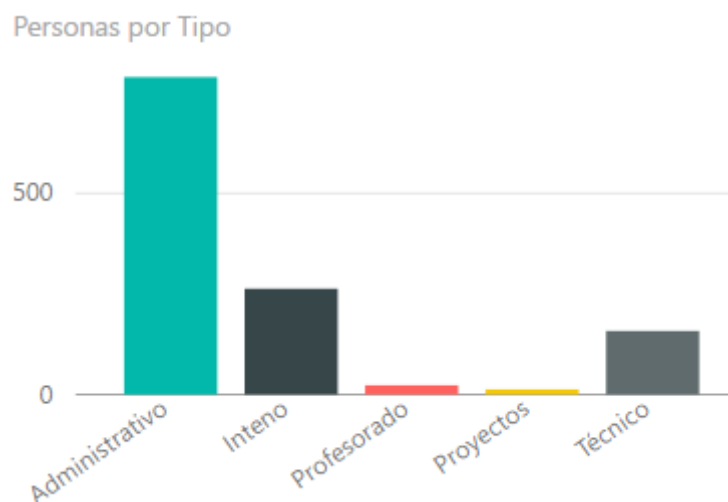
3.2.2 Personas por Subtipo:

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por subtipo. En este gráfico pueden aparecer un total de personas mayor que el número que aparece en la tarjeta de Número de personas. Esto es debido a que una persona ha podido ocupar más de un subtipo a lo largo del periodo de tiempo seleccionado. Entonces dicha persona sumaría 1 en la columna de cada subtipo al que haya pertenecido.



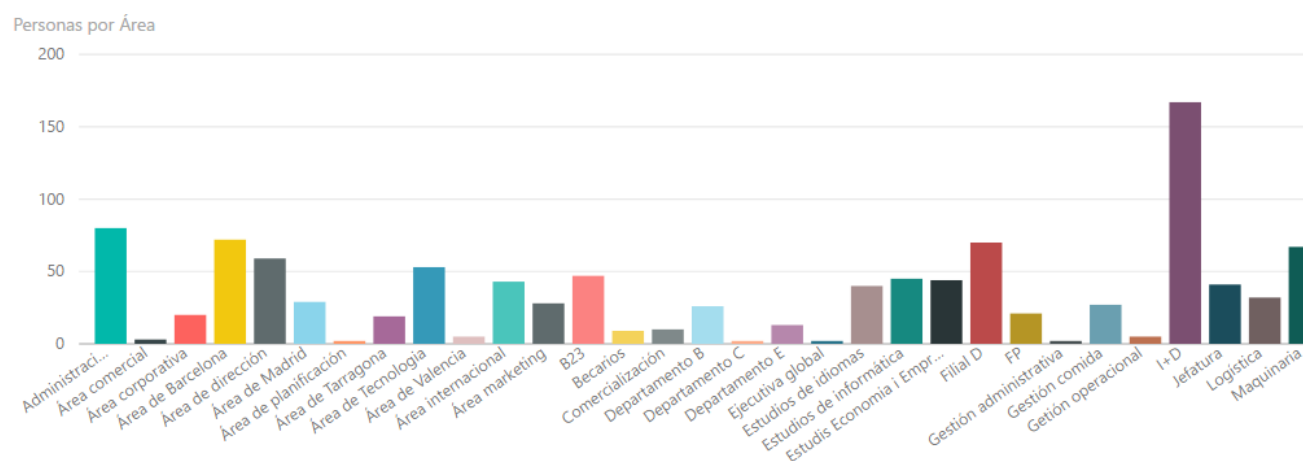
3.2.3 Personas por Tipo:

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por tipo. En este gráfico pueden aparecer un total de personas mayor que el número que aparece en la tarjeta de Número de personas. Esto es debido a que una persona ha podido ocupar más de un tipo a lo largo del periodo de tiempo seleccionado. Entonces dicha persona sumaría 1 en la columna de cada tipo al que haya pertenecido.



3.2.4 Personas por área:

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por área. En este gráfico pueden aparecer un total de personas mayor que el número que aparece en la tarjeta de Número de personas. Esto es debido a que una persona ha podido pertenecer a más de un centro de coste a lo largo del periodo de tiempo seleccionado. Entonces dicha persona sumaría 1 en la columna de cada centro de coste al que haya pertenecido.

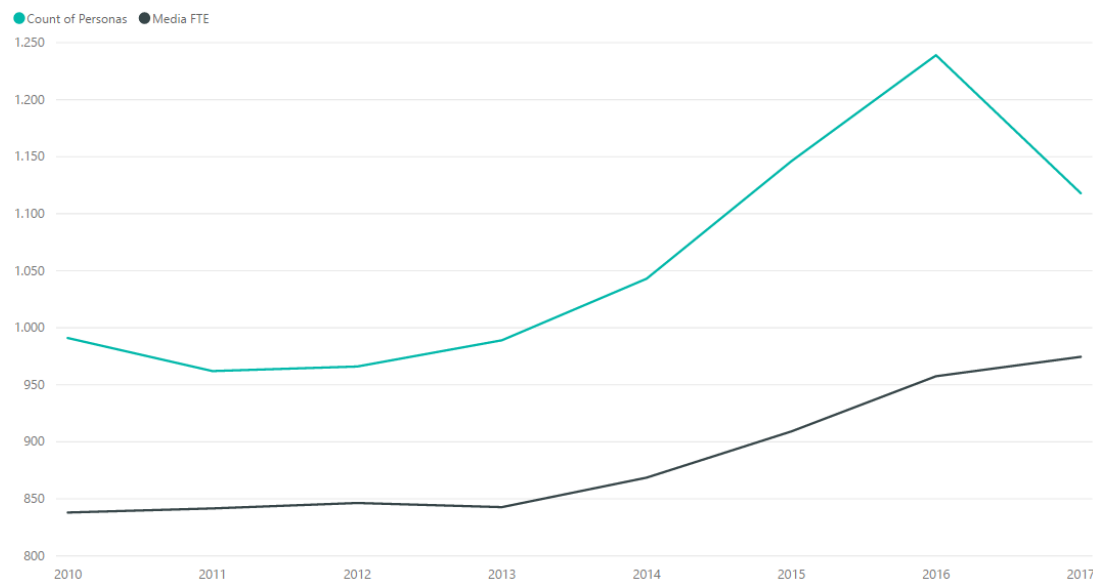


3.3 Personas en un periodo (anual)

3.3.1 Recuento de personas y media FTE por año:

Muestra la evolución del número de personas distintas que ha habido en cada año. Además, debajo, muestra la evolución de la media del FTE de cada año.

Recuento de personas y media FTE por año

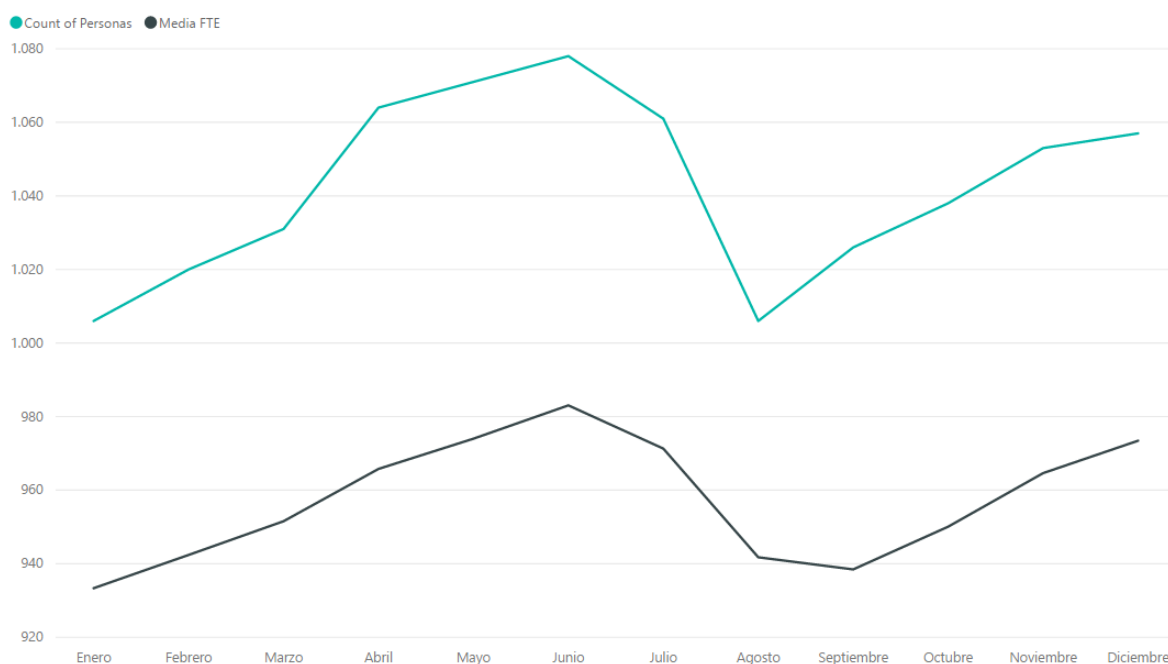


3.4 Personas en un periodo (mensual)

3.4.1 Recuento de personas y media FTE por mes

Muestra la evolución del número de personas distintas que ha habido en durante un año concreto. Además, debajo, muestra la evolución de la media del FTE mensual.

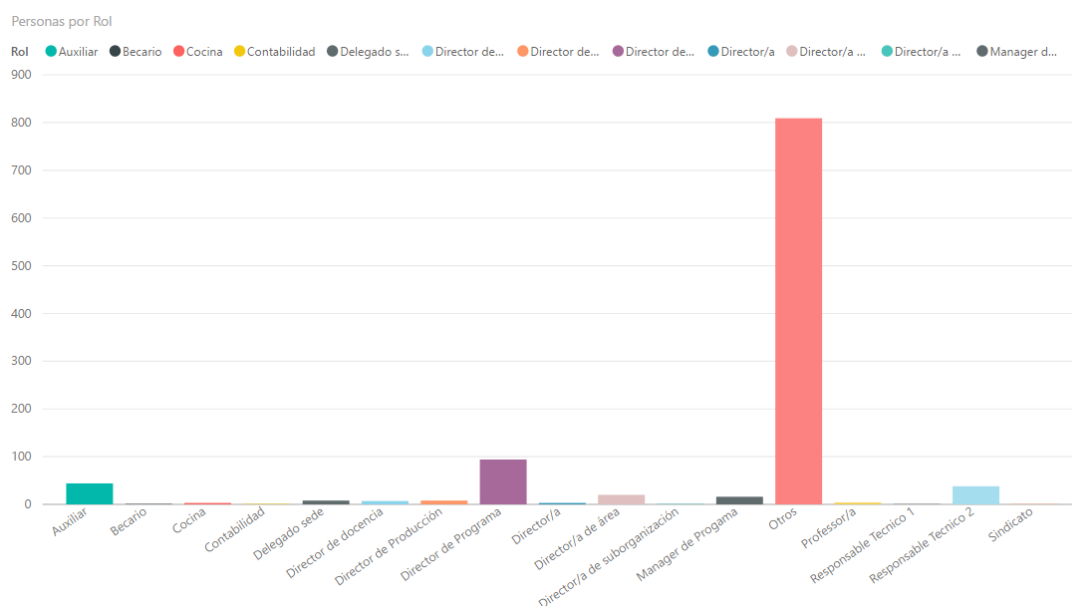
Recuento de personas y media FTE por mes



3.5 Personas por rol

3.5.1 Personas por rol:

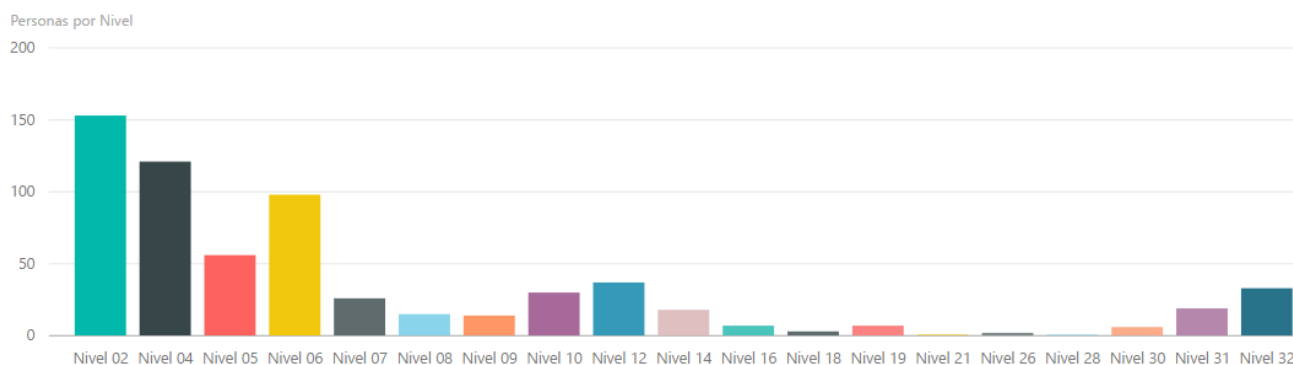
Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por rol. En este gráfico pueden aparecer un total de personas mayor que el número que aparece en la tarjeta de Número de personas. Esto es debido a que una persona ha podido ocupar más de un rol a lo largo del periodo de tiempo seleccionado. Entonces dicha persona sumaría 1 en la columna de cada rol al que haya pertenecido. También podría aparecer un total de personas menor que el número que aparece en la tarjeta de Número de personas ya que no todas las personas tienen un rol asociado.



3.6 Categoría y nivel

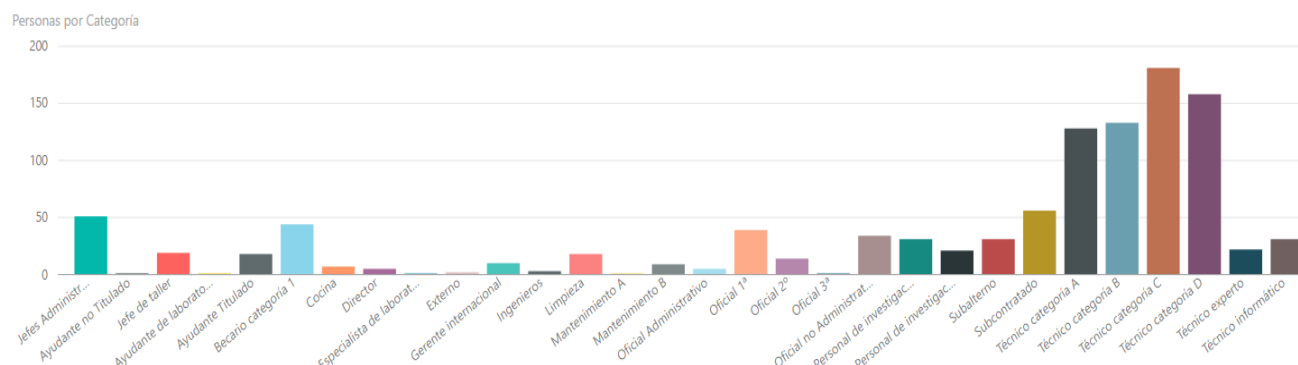
3.6.1 Personas por nivel

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por nivel. En este gráfico pueden aparecer un total de personas mayor que el número que aparece en la tarjeta de Número de personas. Esto es debido a que una persona ha podido ocupar más de un nivel a lo largo del periodo de tiempo seleccionado. Entonces dicha persona sumaría 1 en la columna de cada nivel al que haya pertenecido. También podría aparecer un total de personas menor que el número que aparece en la tarjeta de Número de personas ya que no todas las personas tienen un nivel asociado.



3.6.2 Personas por categoría

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por categoría (grupo profesional). En este gráfico pueden aparecer un total de personas mayor que el número que aparece en la tarjeta de Número de personas. Esto es debido a que una persona ha podido ocupar más de una categoría a lo largo del periodo de tiempo seleccionado. Entonces dicha persona sumaría 1 en la columna de cada categoría a la que haya pertenecido. También podría aparecer un total de personas menor que el número que aparece en la tarjeta de Número de personas ya que no todas las personas tienen una categoría asociada.



4. SALARIOS Y NÓMINAS

4.1 Tarjetas específicas

4.1.1 Salarios

Muestra el coste salarial total (sin bonus) del periodo seleccionado y los filtros aplicados.

Salarios

2,08M

4.1.2 Presupuesto

Muestra el coste total presupuestado del periodo seleccionado y los filtros aplicados.

Presupuesto

2,24M

4.1.3 Coste real

Muestra el coste total real del periodo seleccionado y los filtros aplicados.

Coste real

2,11M

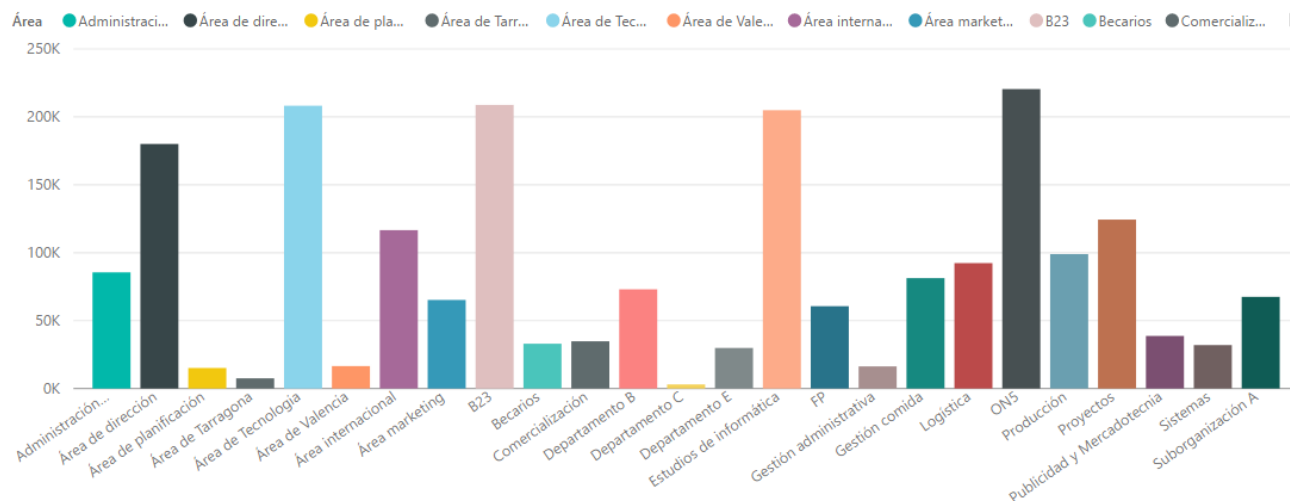


4.2 Coste real

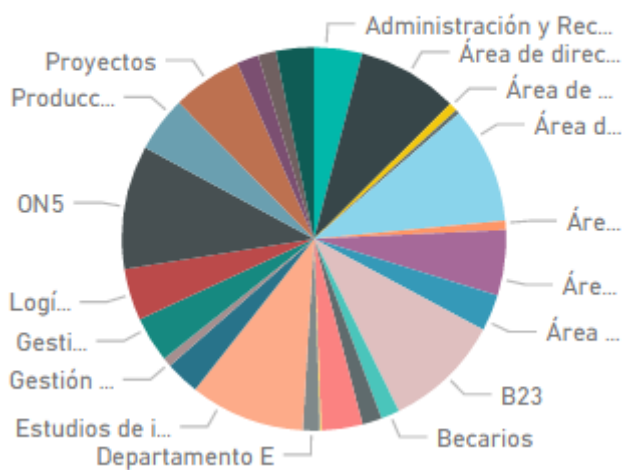
4.2.1 Coste real por área

Muestra el coste real tras aplicar los distintos filtros, dividido por las distintas áreas. Este indicador se muestra tanto en un gráfico de barras como en uno de tarta.

Coste real por Área



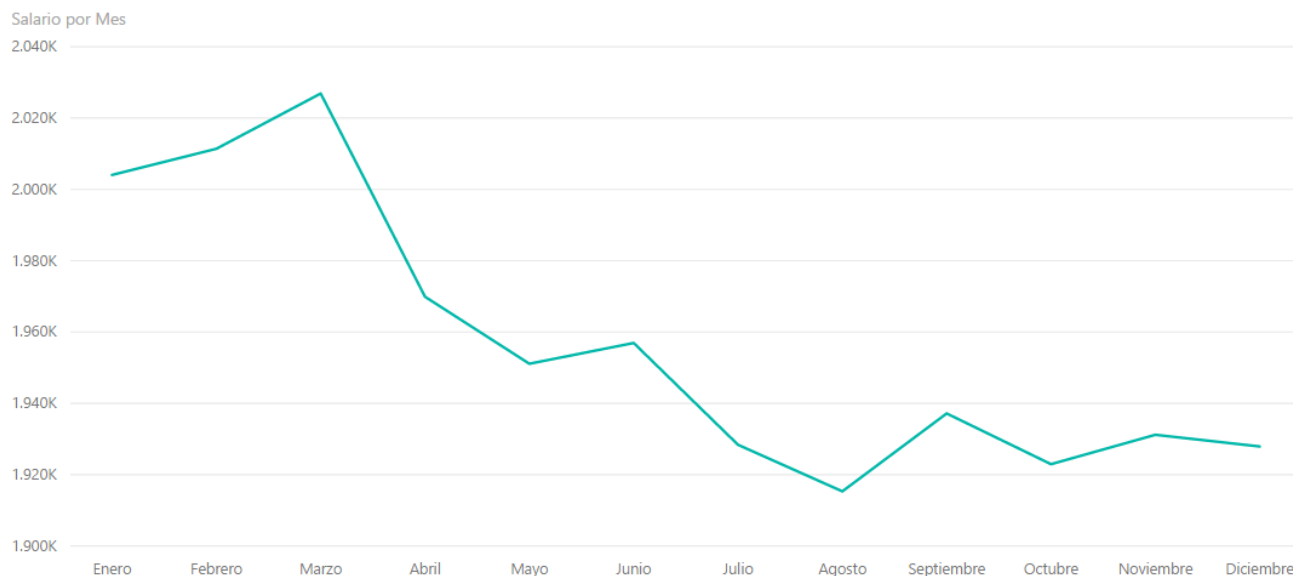
Coste real por Área



4.3 Salarios

4.3.1 Salarios por mes

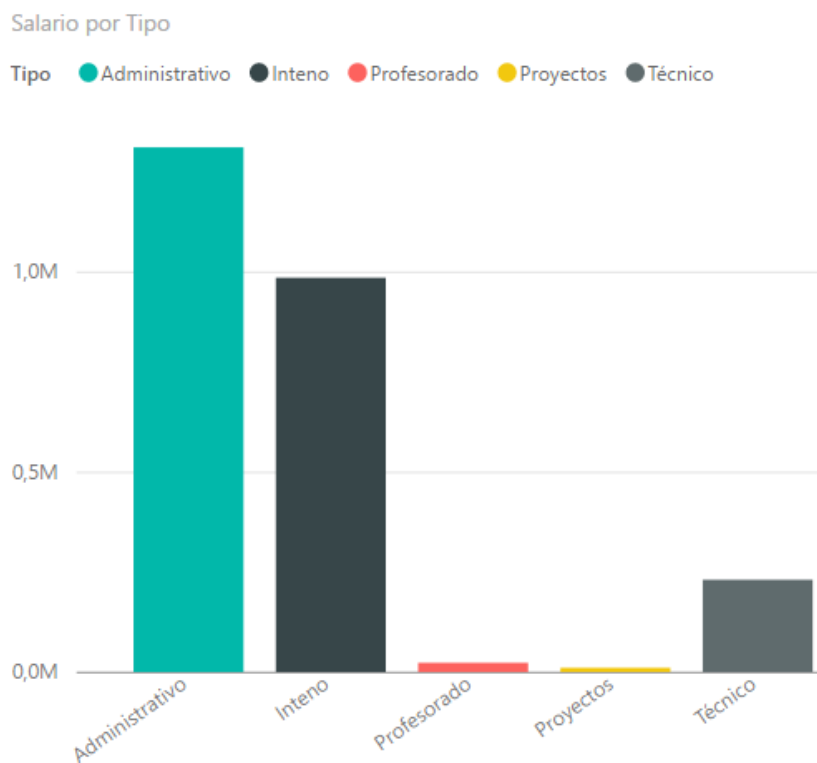
Muestra la evolución de los costes salariales (sin bonus) a lo largo de un año completo. Al pasar el ratón por encima del gráfico aparece el coste salarial concreto de cada mes.



4.4 Costes salariales por tipo y subtipo

4.4.1 Salarios por tipo

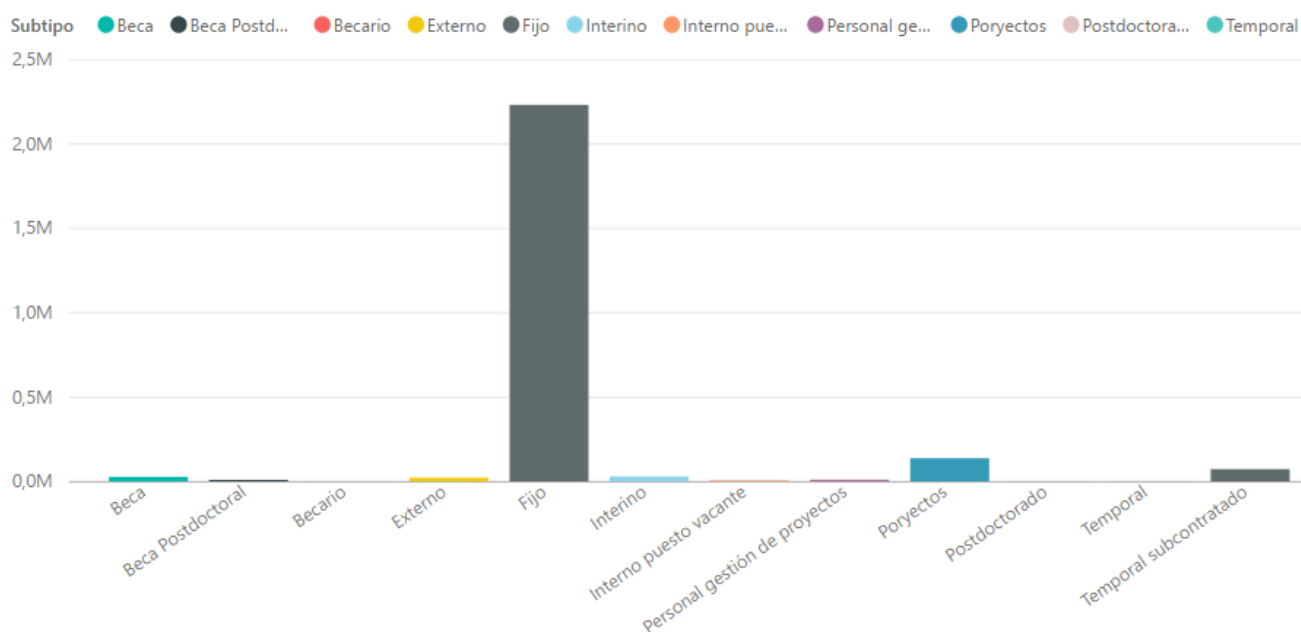
Muestra los costes salariales (sin bonus) dividido por los distintos tipos.



4.4.2 Salarios por subtipo

Muestra los costes salariales (sin bonus) dividido por los distintos subtipos.

Salario por Subtipo

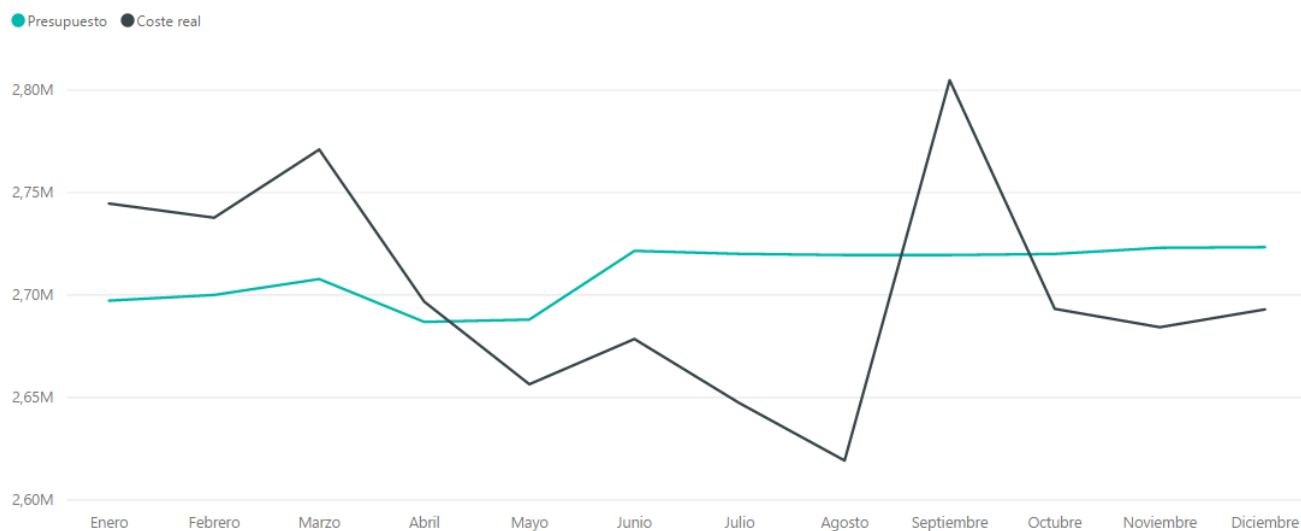


4.5 Desviación de planificación vs realidad

4.5.1 Coste real y presupuesto por mes.

Muestra en un mismo gráfico la evolución del coste real y el presupuestado a lo largo de un año. Al pasar el ratón por encima del gráfico aparecen los valores concretos de coste real y presupuestado para cada mes.

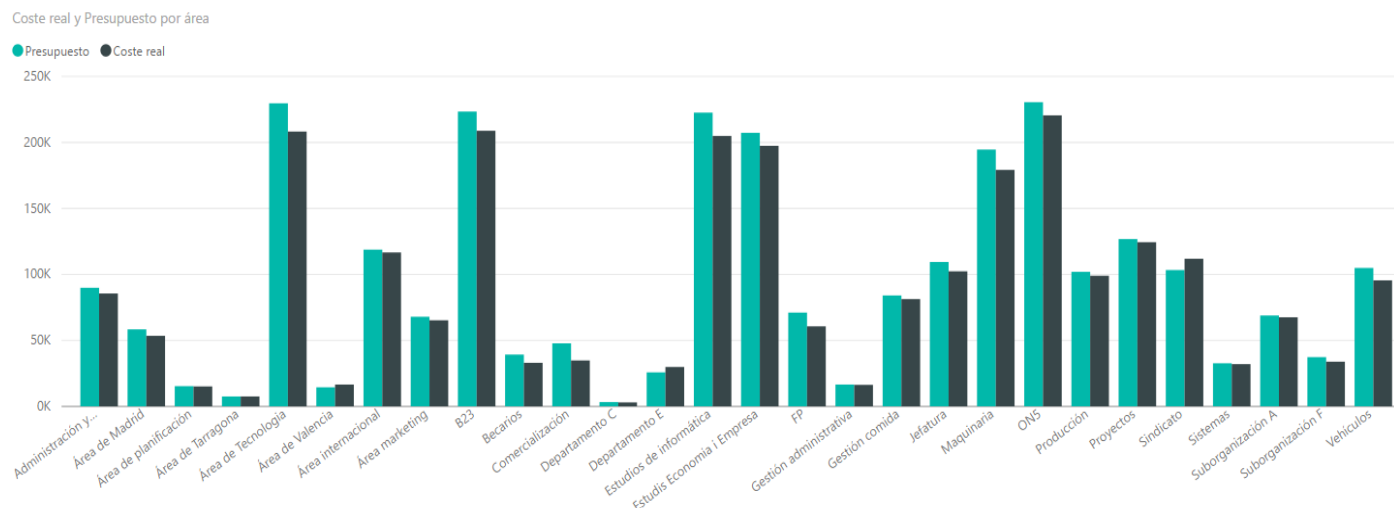
Presupuesto y Coste real por Mes



4.6 Costes reales y presupuestados

4.6.1 Coste real y presupuesto por área

Muestra el coste real total y el presupuestado total tras aplicarse los filtros y un periodo de tiempo, dividido por las distintas áreas. En verde aparece el coste presupuestado y en negro el coste real. Siempre coge los datos del último forecast (última planificación).



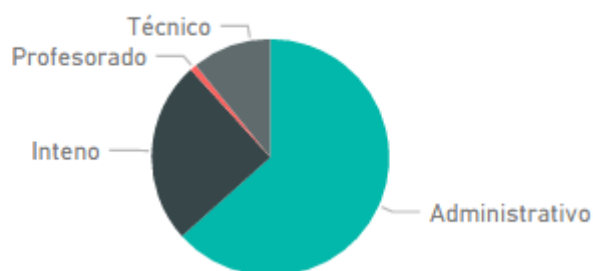
5. PLANIFICACIÓN DE PLANTILLA

5.1 FTE

5.1.1 FTE medio por tipo:

Muestra el FTE medio dividido por los distintos tipos, tras aplicar los filtros.

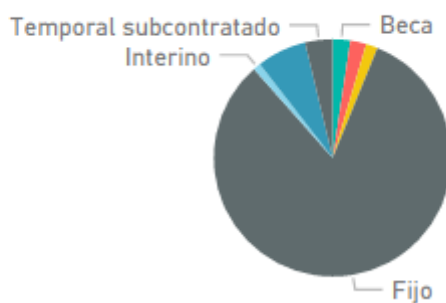
FTE medio por Tipo



5.1.2 FTE medio por subtipo

Muestra el FTE medio dividido por los distintos subtipos, tras aplicar los filtros.

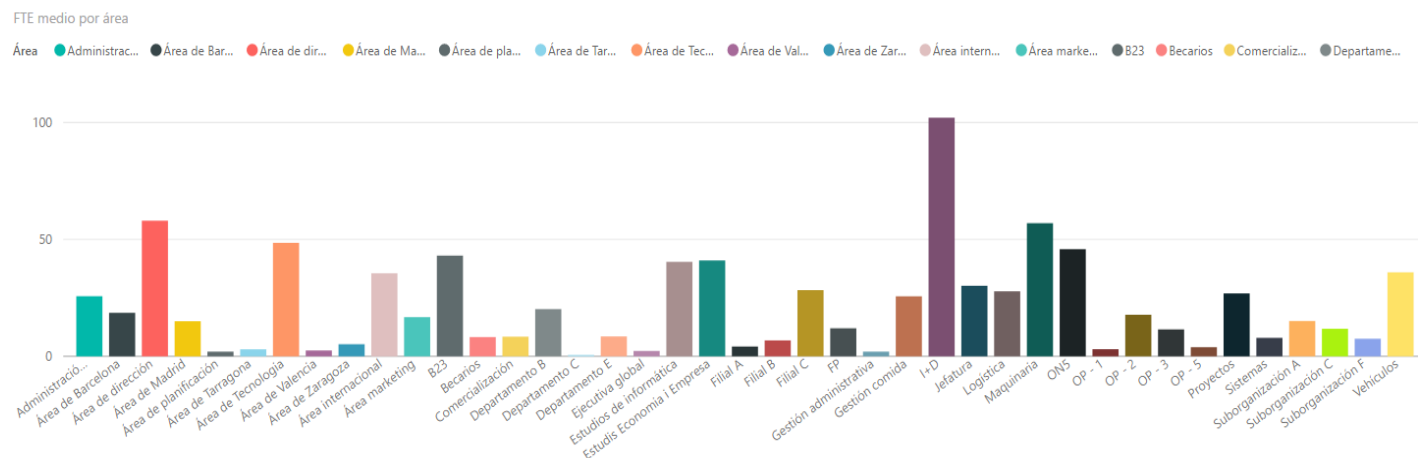
FTE medio por Subtipo



5.1.3 FTE medio por área

Muestra el FTE medio dividido por los distintos centros de coste, tras aplicar los filtros.

Ejemplo: Para el mes de marzo de 2017 el FTE medio de las personas asociadas al centro de coste Área de Administración y recursos humanos es de 26.48, mientras que en el mes de abril del mismo año tienen un FTE medio de 25.71. En caso de seleccionarse ambos meses el FTE medio de las personas asociadas al centro de coste Área de Administración sería la media de ambos meses, es decir, $(26.48 + 25.71) / 2 = 26.09$

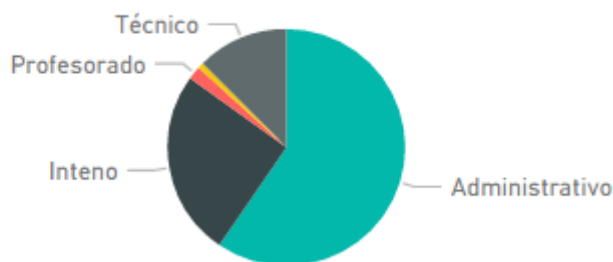


5.2 Headcount

5.2.1 Headcount por tipo

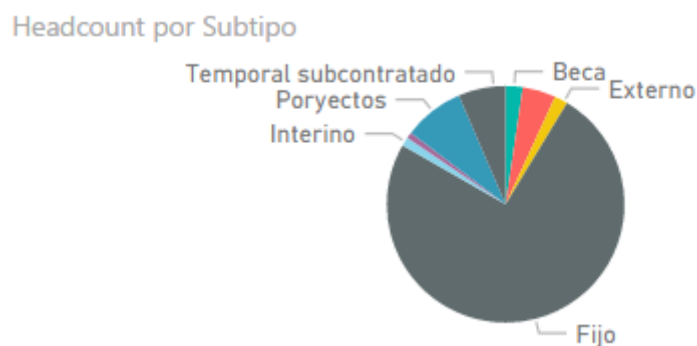
Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por tipo. Al pasar el ratón por encima del gráfico aparecen los valores concretos de Headcount de cada tipo

Headcount por Tipo



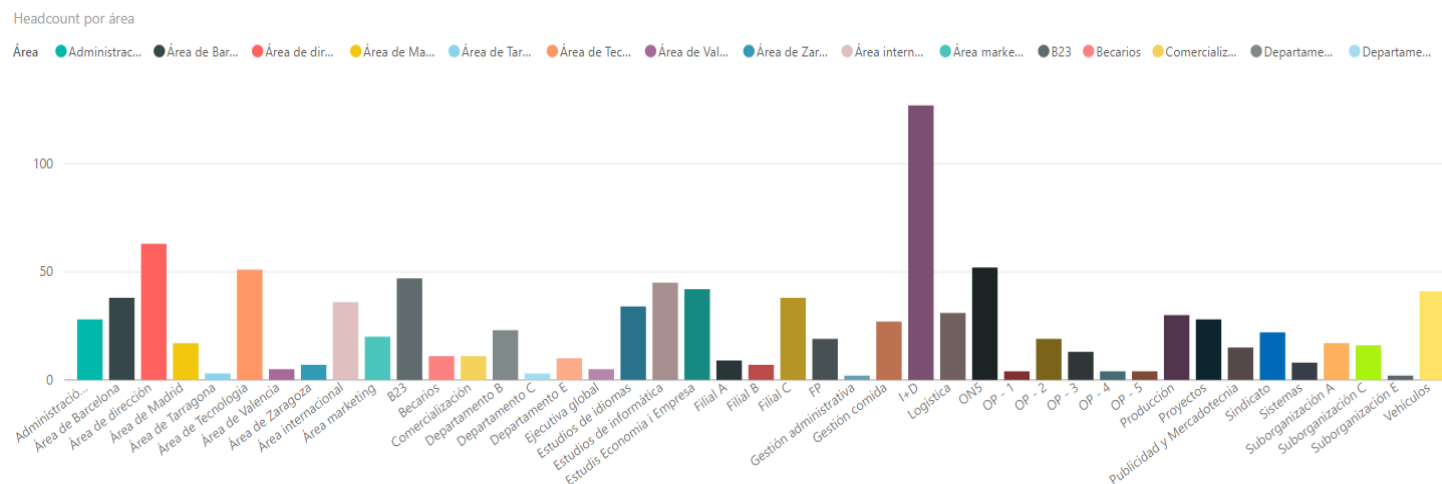
5.2.2 Headcount por subtipo

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por subtipo. Al pasar el ratón por encima del gráfico aparecen los valores concretos de Headcount de cada subtipo



5.2.3 Headcount por área

Muestra el número de personas distintas que han estado activas durante el periodo de tiempo seleccionado, divididas por centro de coste tras aplicarse los filtros. Al pasar el ratón por encima del gráfico aparecen los valores concretos de Headcount de cada centro de coste

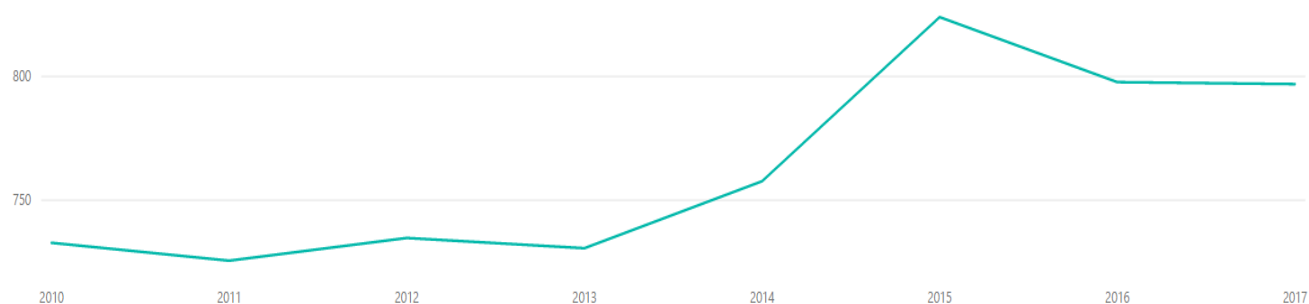


5.3 Evolución anual

5.3.1 FTE por año:

Muestra la evolución anual de la media del FTE tras aplicar los filtros.

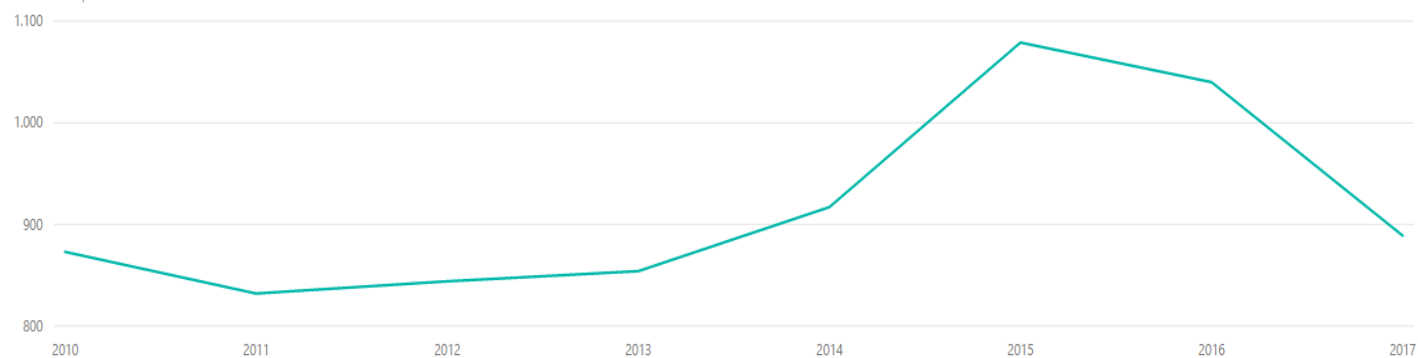
FTE por año



5.3.2 Headcount por año:

Muestra la evolución anual del Headcount de cada año, es decir, del número de personas distintas que han estado activas durante cada año.

Headcount por año

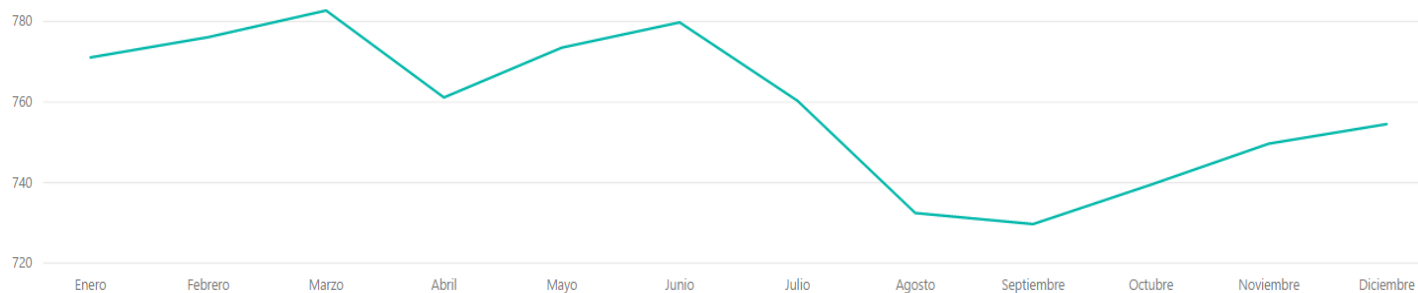


5.4 Evolución mensual

5.4.1 FTE por mes:

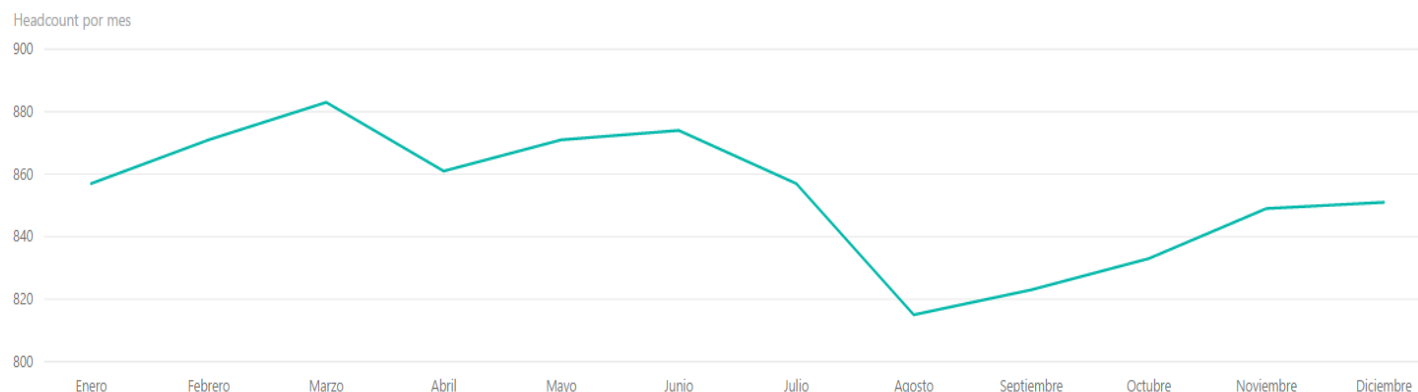
Muestra la evolución del FTE a lo largo de un año tras aplicar los filtros. Al pasar el ratón por encima aparecen los valores concretos de cada mes.

FTE por mes



5.4.2 Headcount por mes:

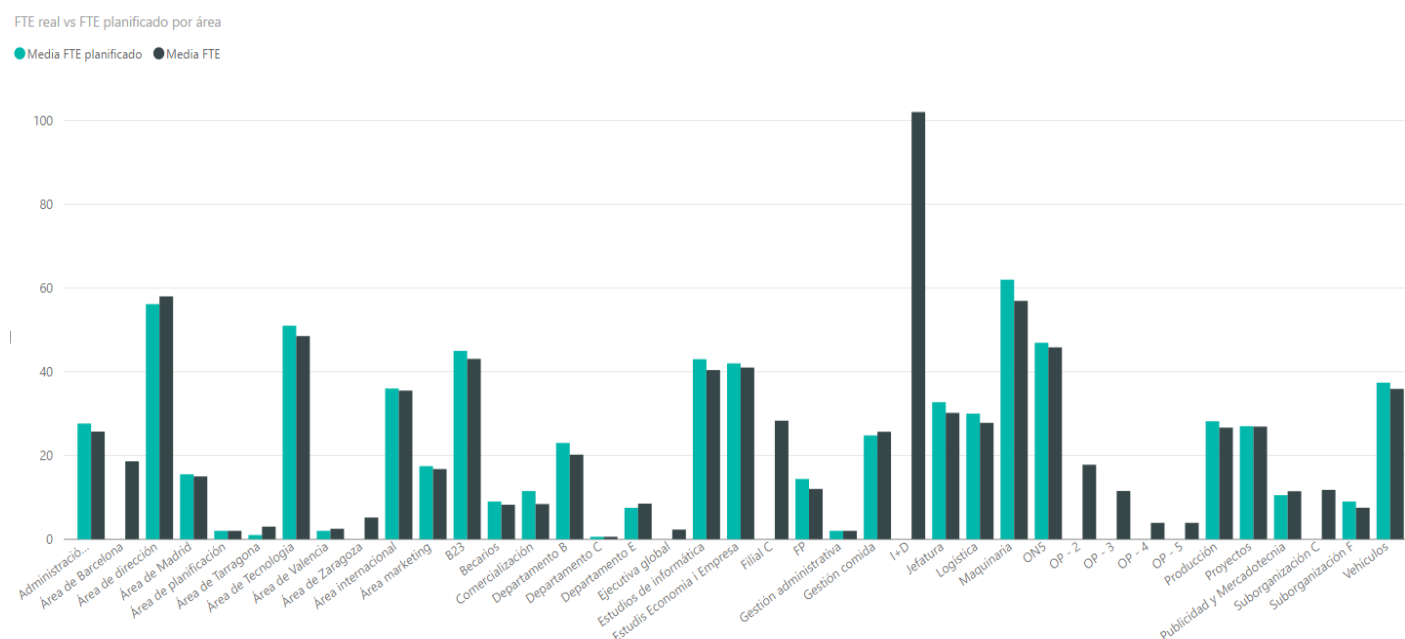
Muestra la evolución del Headcount a lo largo de un año en concreto tras aplicar los filtros. Al pasar el ratón por encima aparecen los valores concretos de cada mes.



5.5 FTE planificado vs real

5.5.1 FTE real vs FTE planificado por área:

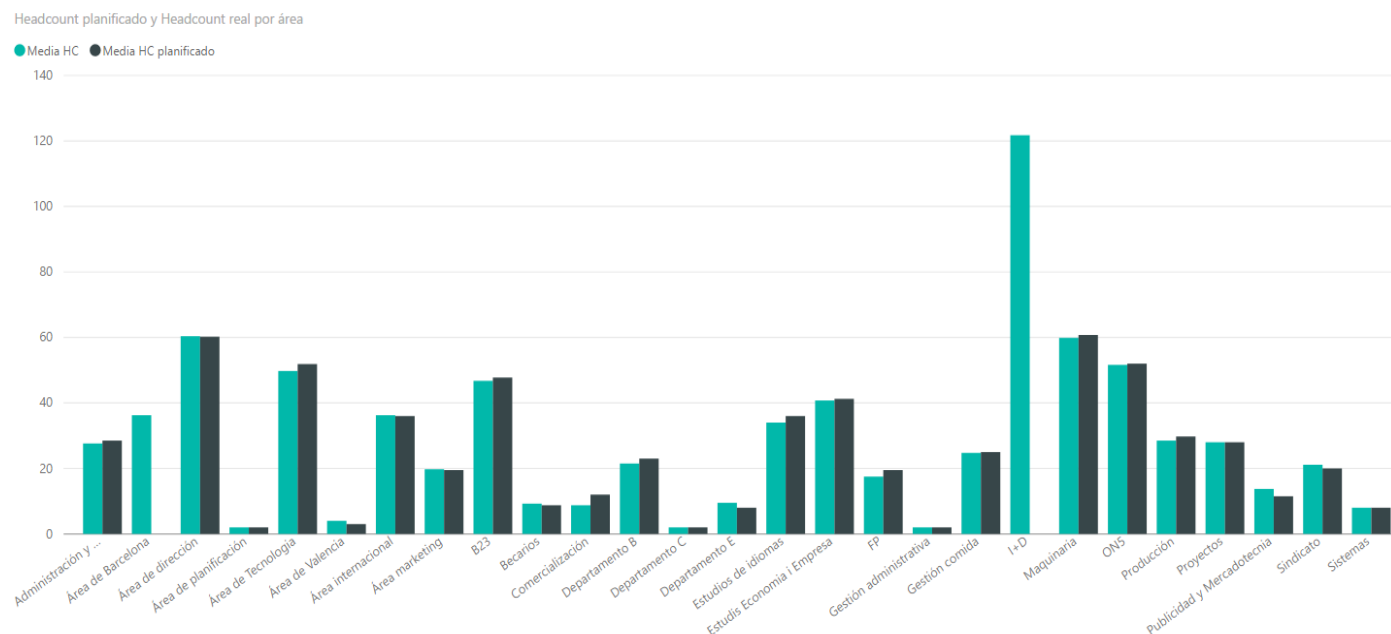
Muestra el FTE medio real y el FTE medio planificado total tras aplicarse los filtros y un periodo de tiempo, dividido por las distintas áreas. En verde aparece el FTE medio planificado y en negro el FTE medio real. Siempre coge los datos del último forecast (última planificación).



5.6 Headcount planificado vs real

5.6.1 Headcount planificado y Headcount real por área:

Muestra el Headcount medio real y el Headcount medio planificado total tras aplicarse los filtros y un periodo de tiempo, dividido por las distintas áreas. En verde aparece el Headcount medio real y en negro el Headcount medio planificado. Siempre coge los datos del último forecast.



6. ORGANIZACIÓN

6.1 Tarjetas específicas

6.1.1 Permanencia media en años:

Muestra la permanencia media en años de los empleados activos durante el periodo de tiempo seleccionado.

Ejemplo: Si una persona empezó el 1 de enero de 2010 y terminó en 31 de diciembre de 2016, su permanencia ha sido de 6 años. Entonces al filtrar por cualquier mes o meses comprendidos entre las fechas anteriores, esta persona contará como que su permanencia ha sido de 6 años, tanto si se filtra por abril de 2010 como por junio de 2015, o cualquier otro mes o meses comprendidos entre las fechas que ha estado activo ininterrumpidamente.

Permanencia media en años

7,66

6.1.2 Permanencia media en años (mujeres):

Muestra la permanencia media en años de las mujeres activas durante el periodo de tiempo seleccionado.

Permanencia media en años (mujeres)

7,21

6.1.3 Permanencia media en años (hombres):

Muestra la permanencia media en años de los hombres activos durante el periodo de tiempo seleccionado.

Permanencia media en años (hombres)

8,38

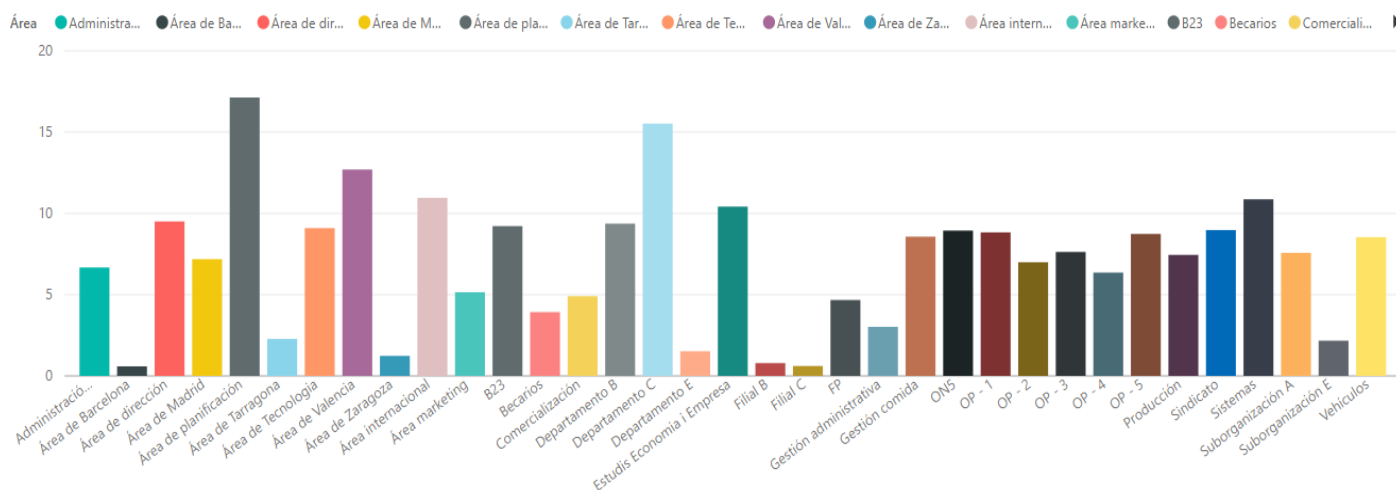


6.2 Permanencia por áreas

6.2.1 Permanencia media por área (en años):

Muestra la permanencia media en años de los empleados activos durante el periodo de tiempo seleccionado, dividido por las distintas áreas.

Permanencia media por área (en años)



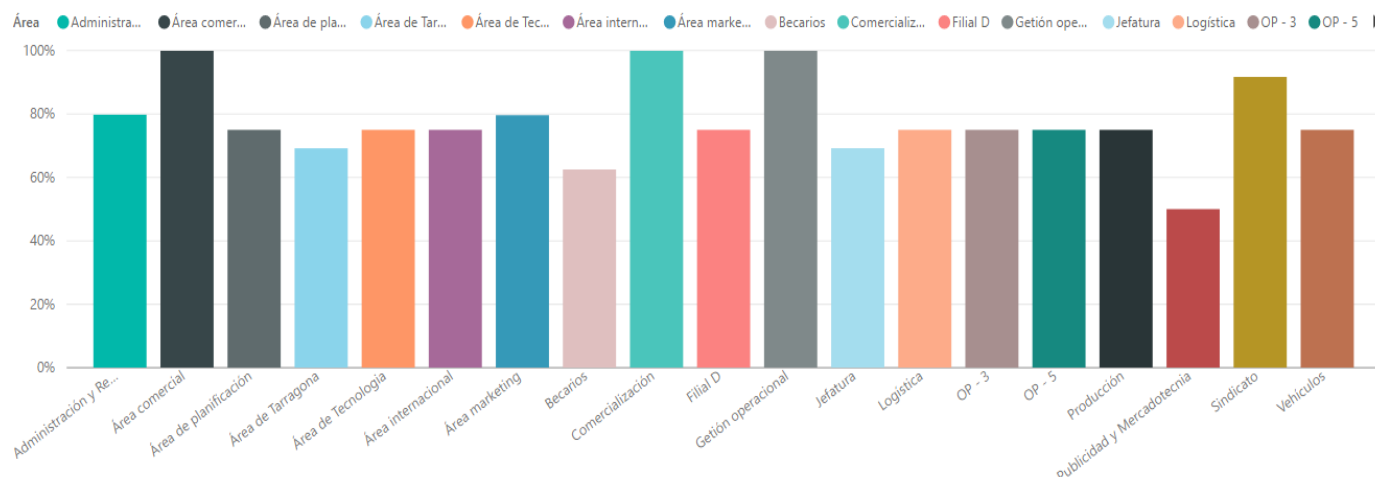
7. DESEMPEÑO / PEOPLE REVIEW

7.1 Desempeño por áreas

7.1.1 Desempeño medio por área

Muestra el desempeño medio de los empleados dividido por las distintas áreas tras aplicarse los filtros. Para hacer la media solo se tienen en cuenta aquellas personas que hayan realizado la evaluación de desempeño, es decir, que si una persona no ha realizado la evaluación de desempeño, su resultado sería 0%, pero esta persona no entraría en el cálculo de la media de desempeño del centro de coste.

Desempeño medio por área



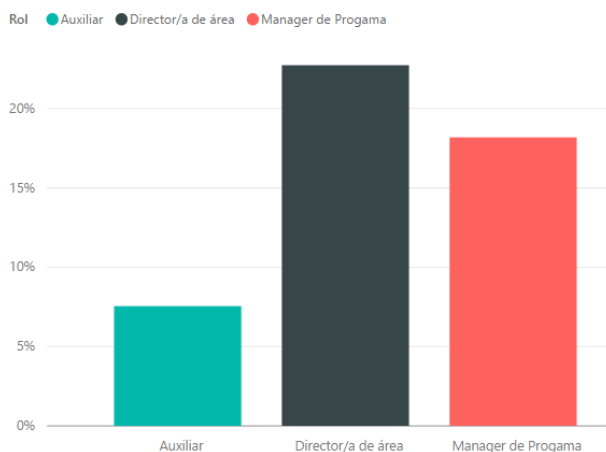
7.2 Porcentaje de alto potencial

7.2.1 Porcentaje de alto potencial por rol (Potencial: "A - Excepcional")

Muestra el porcentaje de personas que tienen alto potencial dividido por los distintos roles tras aplicarse los filtros. Por alto potencial se entiende aquellas personas que tengan como potencial el valor más alto, es decir, "A - Excepcional". En el gráfico mostrado abajo tan solo aparecen 3 roles, ya que éstos son los únicos roles que tienen personas asociadas con alto potencial, los demás roles no aparecen porque no tienen a nadie con potencial "A - Excepcional".

El cálculo del porcentaje es el cociente entre el número de personas del rol con alto potencial entre el número de personas que tienen asociado dicho rol.

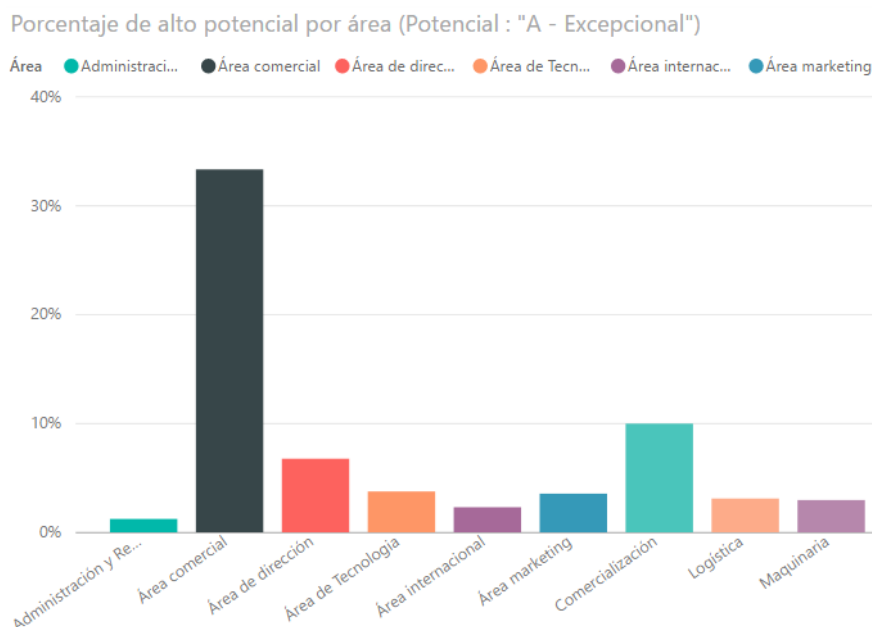
Porcentaje de alto potencial por rol (Potencial: "A - Excepcional")



7.2.2 Porcentaje de alto potencial por área (Potencial: "A - Excepcional")

Muestra el porcentaje de personas que tienen alto potencial dividido por los distintos centros de coste tras aplicarse los filtros. Por alto potencial se entiende aquellas personas que tengan como potencial el valor más alto, es decir, "A - Excepcional". En el gráfico mostrado abajo tan solo aparecen los centros de coste que tengan alguna persona con alto potencial.

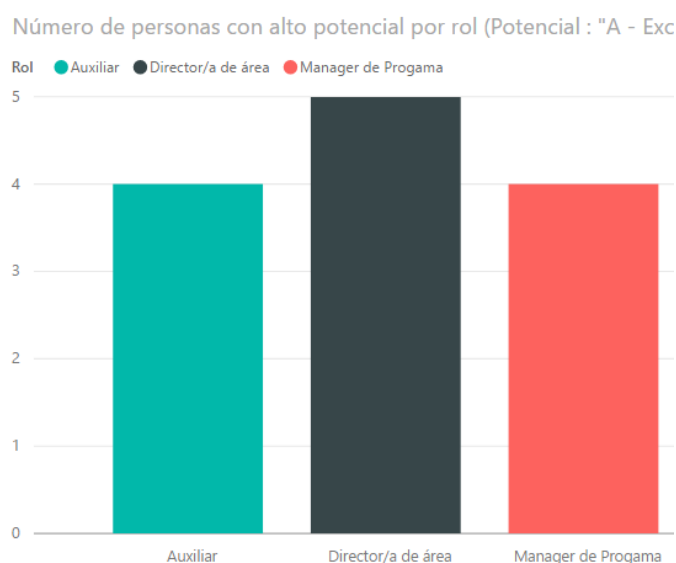
El cálculo del porcentaje es el cociente entre el número de personas que pertenecen al centro de coste con alto potencial entre el número de personas total del centro de coste.



7.3 Número de personas con alto potencial

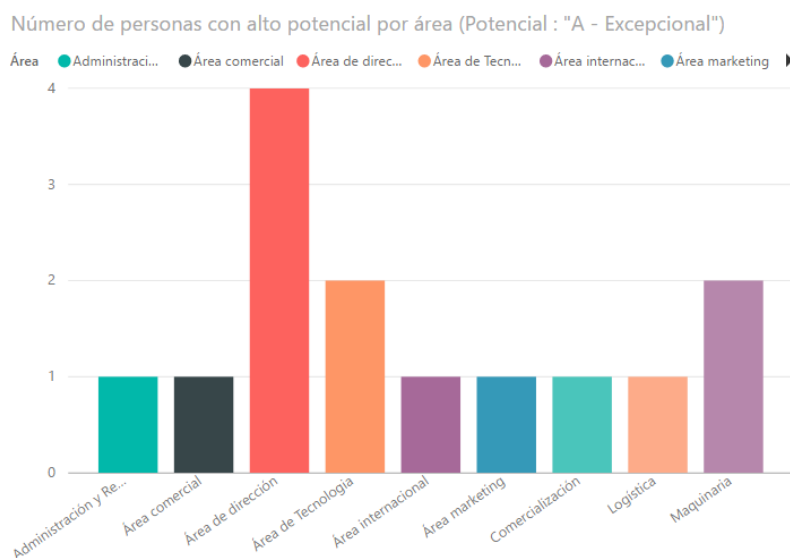
7.3.1 Número de personas con alto potencial por rol (Potencial: "A - Excepcional"):

Muestra el número de personas que tienen alto potencial ("A - Excepcional") dividido por los distintos roles tras aplicarse los filtros. Tan solo aparecen los roles que tengan asociadas personas que tienen alto potencial.



7.3.2 Número de personas con alto potencial por área (Potencial: "A - Excepcional"):

Muestra el número de personas que tienen alto potencial ("A - Excepcional") dividido por los distintos centros de coste tras aplicarse los filtros. Tan solo aparecen los centros de coste que tengan asociadas personas que tienen alto potencial.



8. FORMACIÓN

8.1 Tarjetas específicas

8.1.1 Personas que participan en formación:

Muestra el número total de personas que han participado en formación tras aplicarse los filtros. Es un resultado anual.

Personas que participan en formación

36

8.1.2 Porcentaje de personas que participan en formación:

Muestra el porcentaje de personas que han participado en formación durante el año tras aplicarse los filtros.

Porcentaje de personas que participan en formación

3,63%

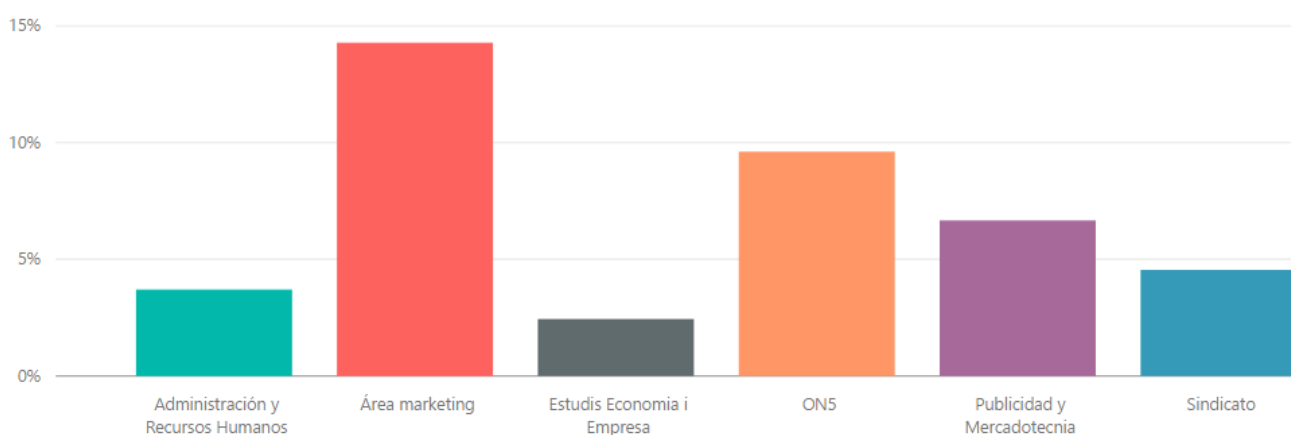
8.2 Formación por área

8.2.1 Porcentaje de participación en formación por área:

Este gráfico muestra los porcentajes de personas que han participado en formación a lo largo de un año dividido por las distintas áreas.

Porcentaje de participación en formación por área

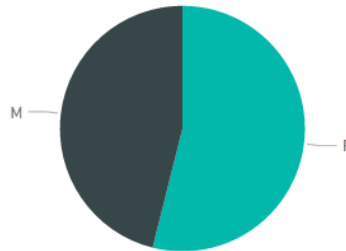
Área ● Administración y Recursos Humanos ● Área marketing ● Estudios Economía i Empresa ● ON5 ● Publicidad y Mercadotecnia ● Sindicato



8.2.2 Porcentaje de participación en formación por género:

Muestra el porcentaje de personas que han participado en formación a lo largo de un año dividido por género.

Porcentaje de participación en formación por género



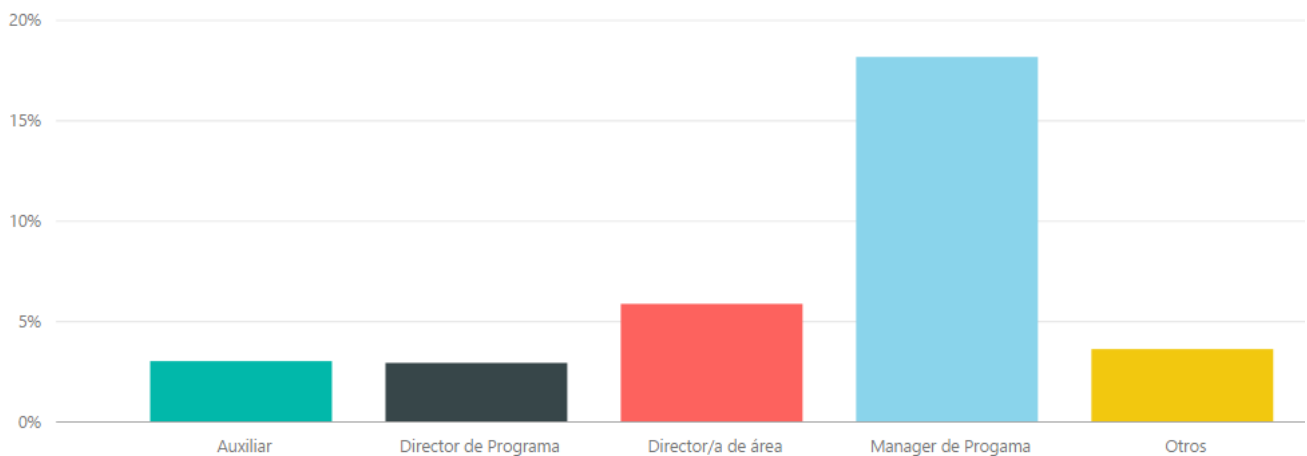
8.3 Formación por rol

8.3.1 Porcentaje de participación en formación por rol:

Este gráfico muestra los porcentajes de personas que han participado en formación a lo largo de un año dividido por los distintos roles. Tan solo aparecen los roles que tengan personas que han participado en formación

Porcentaje de participación en formación por rol

Rol ● Auxiliar ● Director de Programa ● Director/a de área ● Manager de Progama ● Otros



9. DIMENSIONAMIENTO Y PERSONAL NO RESIDENTE

9.1 Tarjetas específicas

9.1.1 Plantilla fija:

Muestra el total de la plantilla fija tras aplicarse los distintos filtros.

Plantilla fija

663

9.1.2 Plantilla temporal:

Muestra el total de plantilla temporal tras aplicarse los distintos filtros. Se entiende por temporal aquellas personas que tienen asociadas el subtipo "Temporal".

Plantilla temporal

40

9.1.3 Porcentaje de plantilla fija:

Muestra el porcentaje de plantilla fija, el cual se obtiene al dividir la plantilla fija entre el número total de personas.

Porcentaje de plantilla fija

94,31%

9.1.4 Porcentaje de plantilla temporal:

Muestra el porcentaje de plantilla temporal, el cual se obtiene al dividir la plantilla temporal entre el número total de personas.

Porcentaje de plantilla temporal

5,69%



9.1.5 Porcentaje de empleados extranjeros:

Muestra el porcentaje de plantilla que ha estado activa durante el periodo seleccionado cuya nacionalidad no sea la española, tras aplicarse los filtros.

Porcentaje de empleados extranjeros

8,62%

9.1.6 Porcentaje de empleados extranjeros pertenecientes a la UE:

Muestra el porcentaje de plantilla que ha estado activa durante el periodo seleccionado cuya nacionalidad no sea la española, pero si sea de la Unión Europea, tras aplicarse los filtros.

Porcentaje de empleados extranjeros pertenecientes a la UE

2,78%

9.1.7 Personas incorporadas:

Muestra el número de personas que se han incorporado durante el periodo seleccionado tras aplicarse los filtros.

Personas incorporadas

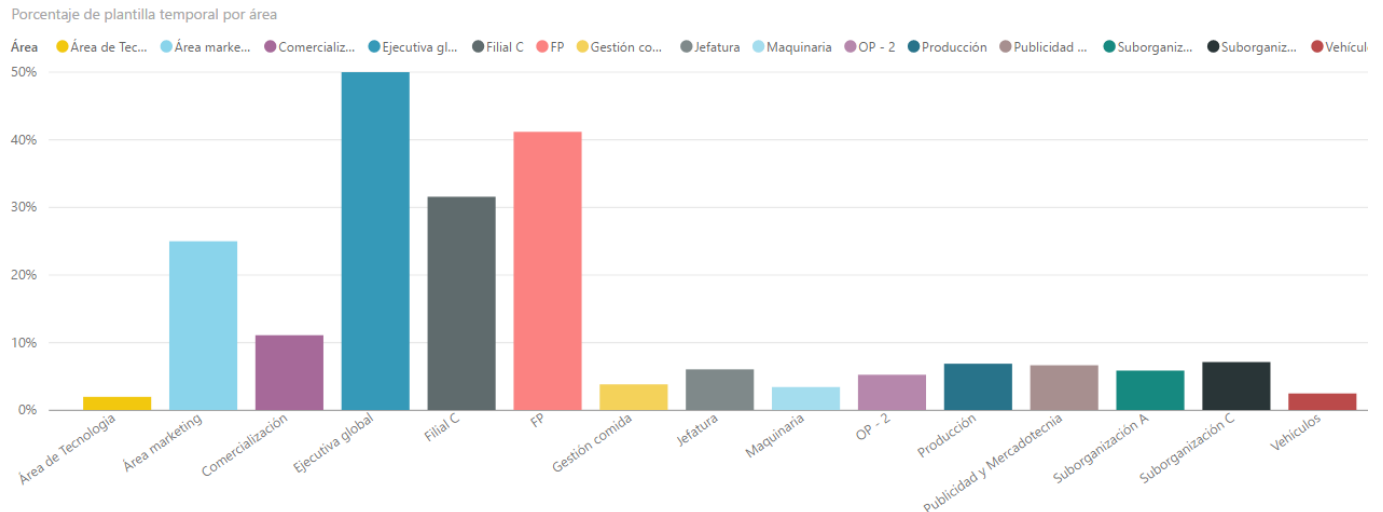
27



9.2 Porcentaje de plantilla fija y temporal

9.2.1 Porcentaje de plantilla temporal por área

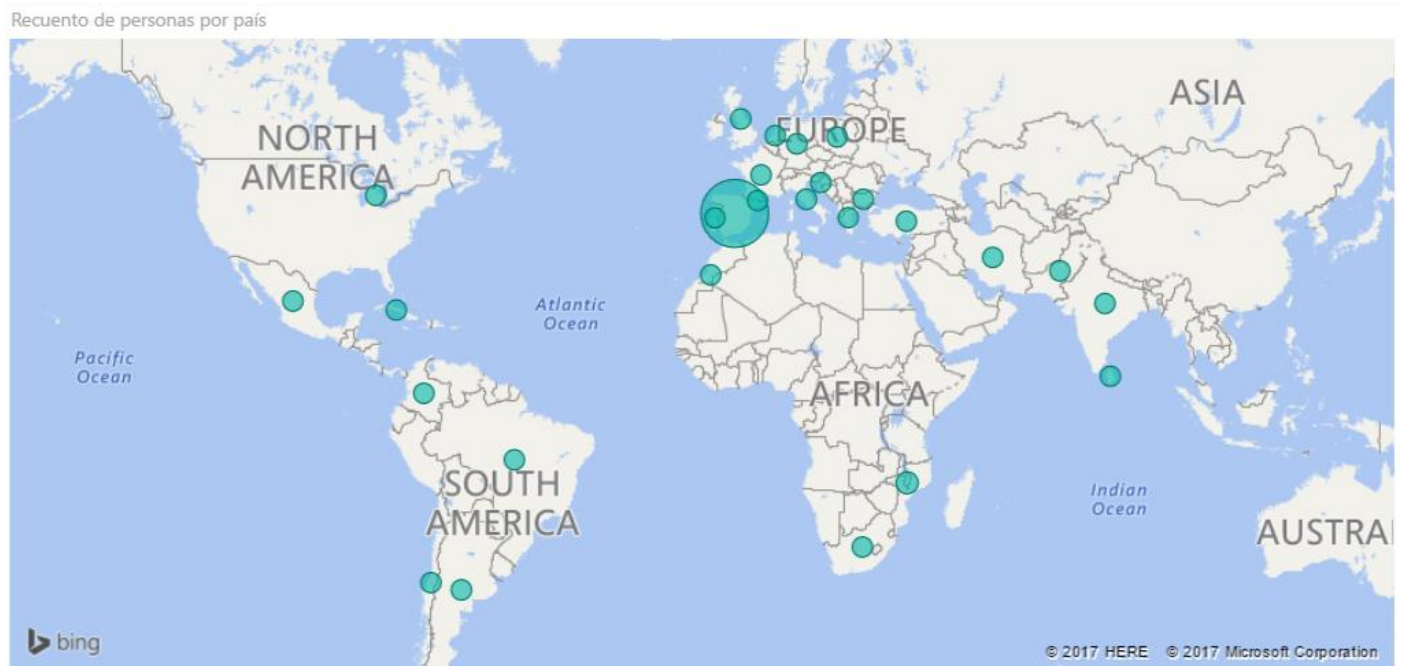
Muestra el porcentaje de plantilla temporal dividido entre las distintas áreas tras aplicarse los distintos filtros.



9.3 Mapa de nacionalidades

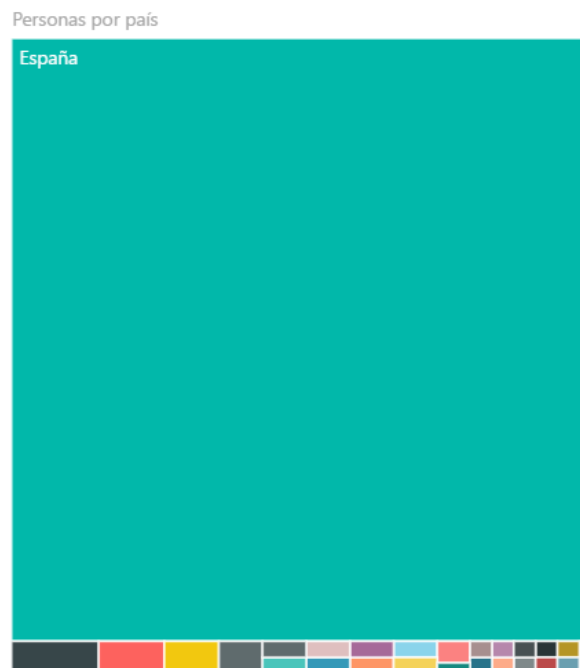
9.3.1 Recuento de personas por país:

Muestra las nacionalidades de las personas que han estado activas durante el periodo seleccionado tras aplicarse los filtros. Contra más personas pertenecen a una nacionalidad, más grande se ve el punto sobre el país en cuestión.



9.3.2 Personas por país:

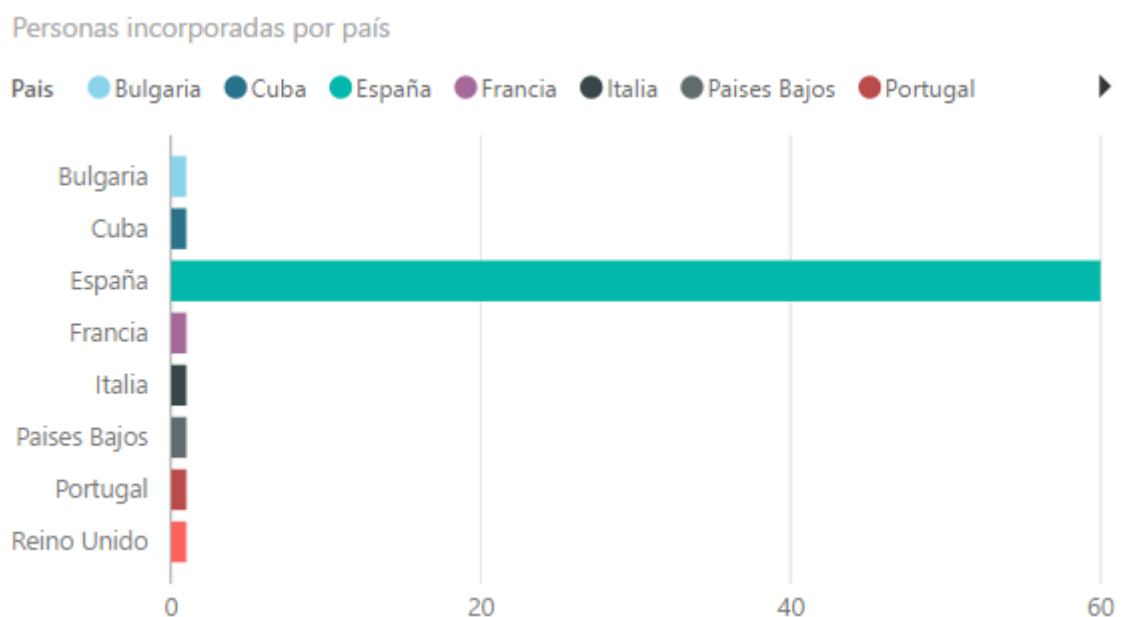
Muestra como están divididas las nacionalidades de las personas que han estado activas a lo largo del periodo seleccionado en un Treemap.



9.4 Personal no residente

9.4.1 Personas incorporadas por país:

Muestra el número de personas que se han incorporado durante el periodo seleccionado, divididos entre las distintas nacionalidades de estas personas.



9.4.2 Personas incorporadas por país:

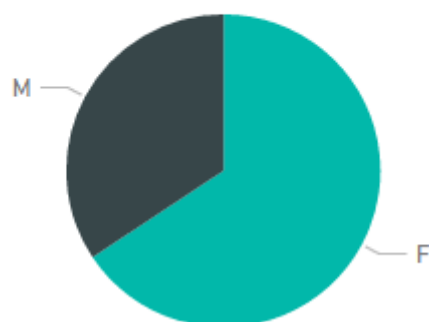
Muestra como están divididas las nacionalidades de las personas que se han incorporado a lo largo del periodo seleccionado en un Treemap.



9.4.3 Personas incorporadas por género:

Muestra el número de personas que se han incorporado durante el periodo seleccionado divididas por género. Al pasar el ratón por encima del gráfico aparecen los valores concretos.

Personas incorporadas por género



10. ROTACIÓN DE PLANTILLA

10.1 Tarjetas específicas

10.1.1 Recuento de bajas voluntarias:

Muestra el número de personas que han terminado su relación laboral con la organización durante el periodo seleccionado y cuyo motivo ha sido:

“Dimisión/baja voluntaria”

Recuento de bajas voluntarias

7

10.1.2 Recuento de bajas totales:

Muestra el número total de personas que han terminado su relación laboral con la organización durante el periodo seleccionado.

Recuento de bajas totales

23

10.1.3 Recuento de altas:

Muestra el número total de personas que se han incorporado a la organización durante el periodo seleccionado.

Recuento de altas

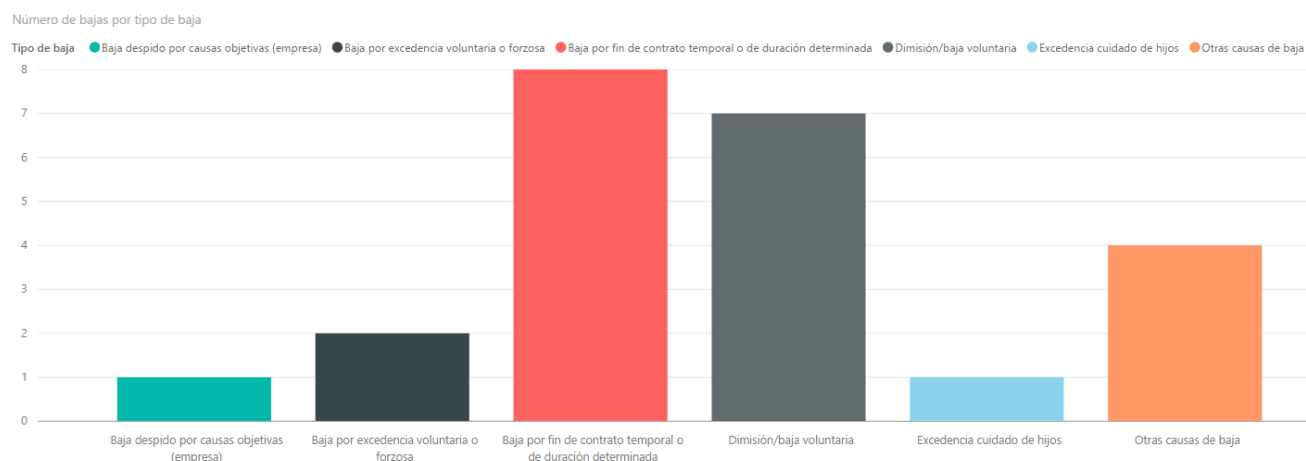
23



10.2 Recuento de bajas

10.2.1 Número de bajas por tipo de baja:

Muestra el número de personas que han terminado su relación laboral con la organización durante el periodo seleccionado, divididos por los distintos motivos posibles de baja. Aquellos motivos de baja que no tenga a nadie asociado durante el periodo seleccionado no se muestran en el gráfico.



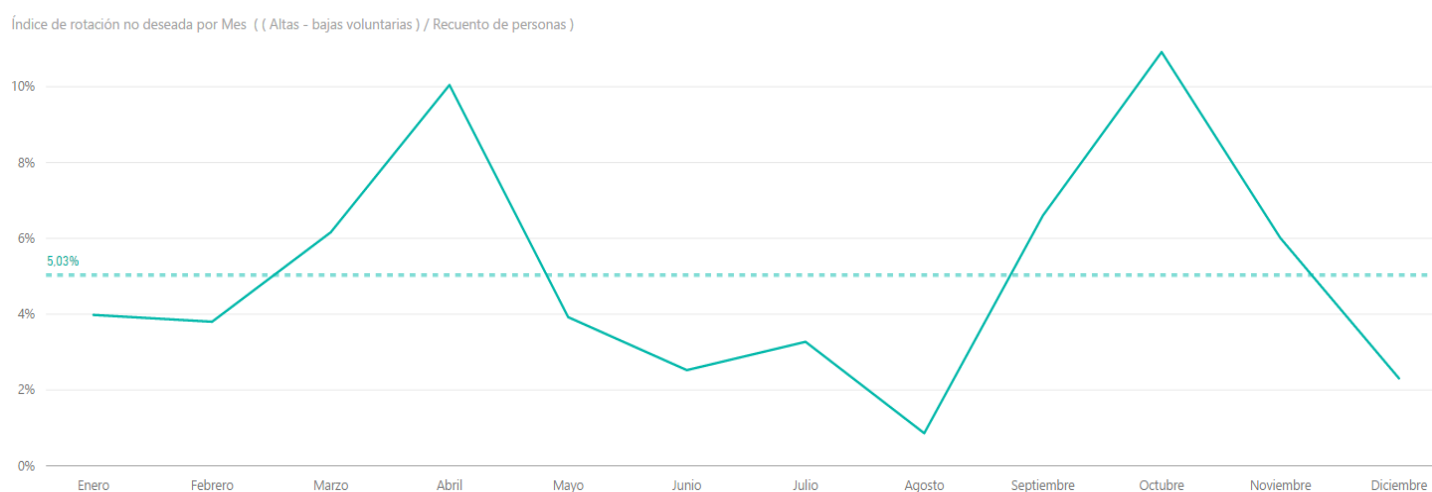
10.3 Índice de rotación no deseada (mensual)

10.3.1 Índice de rotación por mes:

Muestra la evolución del índice de rotación no deseada lo largo de un año. Además también muestra en una línea discontinua el valor medio del índice a lo largo del año (calculado como la media de los valores del índice para los distintos meses que se muestran), para así destacar aquellos meses que están o bien por encima o bien por debajo de esta media.

El índice de rotación no deseada de cada mes se calcula mediante la siguiente fórmula:

$$\frac{\text{Altas} - \text{Bajas voluntarias}}{\text{Recuento de personas}}$$



10.4 Índice de rotación no deseada (anual)

10.4.1 Índice de rotación por año:

Muestra la evolución del índice de rotación no deseada lo largo de los años. Además también muestra en una línea discontinua el valor medio del índice a lo largo los años, para así destacar aquellos años que están o bien por encima o bien por debajo de esta media. Solo se muestra información a partir del año 2016 porque es a partir de este año que empiezan a registrarse bajas con el motivo “51 Dimisión/baja voluntaria”.

El índice de rotación no deseada de cada año se calcula mediante la siguiente fórmula:

$$\frac{\text{Altas} - \text{Bajas voluntarias}}{\text{Recuento de personas}}$$



10.5 Índice de desvinculación no deseada (mensual)

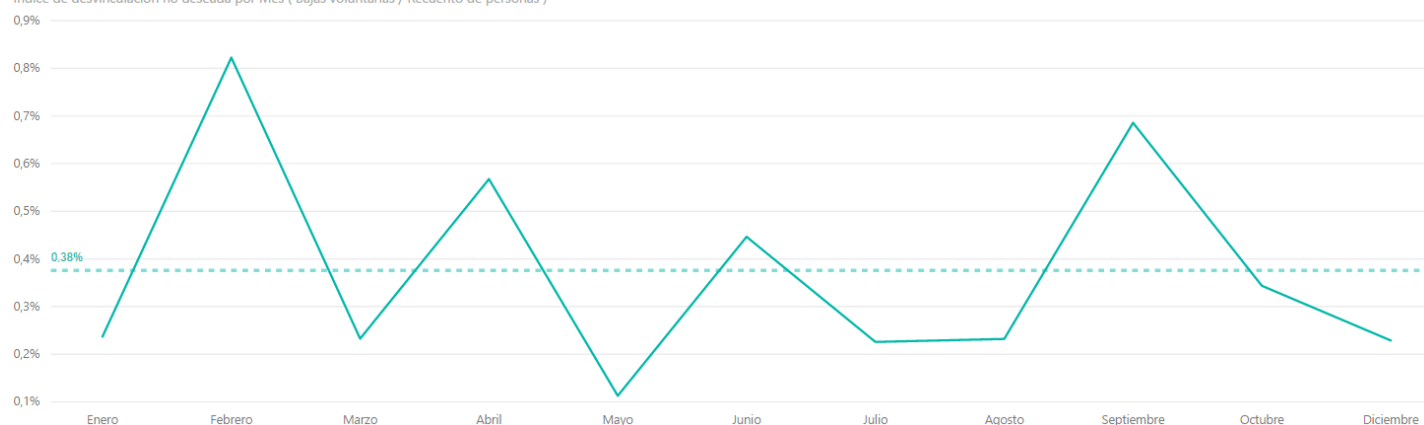
10.5.1 Índice de desvinculación no deseada por mes:

Muestra la evolución del índice de desvinculación no deseada lo largo de los meses de un año. Además también muestra en una línea discontinua el valor medio del índice a lo largo los meses (calculado como la media de los valores del índice para los distintos meses que se muestran), para así destacar aquellos años que están o bien por encima o bien por debajo de esta media. Solo se muestra información a partir del año 2016 porque es a partir de este año que empiezan a registrarse bajas con el motivo “51 Dimisión/baja voluntaria”.

El índice de desvinculación no deseada de cada mes se calcula mediante la siguiente fórmula:

$$\frac{\text{Bajas voluntarias}}{\text{Recuento de personas}}$$

Índice de desvinculación no deseada por Mes (Bajas voluntarias / Recuento de personas)



10.6 Índice de desvinculación no deseada (anual)

10.6.1 Índice de desvinculación no deseada por año:

Muestra la evolución del índice de desvinculación no deseada lo largo de los años. Además también muestra en una línea discontinua el valor medio del índice a lo largo los años, para así destacar aquellos años que están o bien por encima o bien por debajo de esta media. Solo se muestra información a partir del año 2016 porque es a partir de este año que empiezan a registrarse bajas con el motivo “51 Dimisión/baja voluntaria”.

El índice de desvinculación no deseada de cada mes se calcula mediante la siguiente fórmula:

$$\frac{\text{Bajas voluntarias}}{\text{Recuento de personas}}$$

Índice de desvinculación no deseada por año(Bajas voluntarias / Recuento de personas * 12)

