



**Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza**



Proyecto Fin de Carrera

Optimización en los procesos de reporte de información de actividad

^{Autor}
Cristian Garrido Méndez

Director
Diego Martín Aranda
Ponente
Francisco José López Pellicer

Ingeniería en Informática
Curso 2013/2014

AGRADECIMIENTOS

A mis padres y hermana por su apoyo y comprensión durante todos estos Duros años de aprendizaje.

A Francisco José López Pellicer por su infatigable ayuda durante el proyecto y en la redacción de este documento.

A Diego Martín y Víctor Vidaller por haberme dado la oportunidad de integrarme en un entorno empresarial y de aprender una tecnología puntera.

A Patricia Moreno por demostrarme que la gente eficiente y trabajadora existe y que no es una leyenda; y a María José Esteban por sus consejos sobre Sharepoint.

A David Iruzubieta por haber sido un gran compañero de prácticas y no prácticas durante la carrera.

Al doctor TorreIglesias y a su ayudante de campo por esas tardes de desconexión previas a exámenes.

Y por último, a Raquel Sanz por su apoyo en los peores momentos de esta etapa que está a punto de finalizar.

GRACIAS A TODOS!!

RESUMEN GENERAL

El presente documento conforma la memoria del proyecto fin de carrera (PFC) desarrollado durante el curso 2013-2014, en él se resume todo el trabajo elaborado en el proyecto consistente en el diseño e implementación de una aplicación contenida en una Intranet empresarial. Ésta planifica, ordena y centraliza toda la información de los distintos proyectos en curso gestionados en una consultora informática.

Para el desarrollo de la aplicación, se ha utilizado Sharepoint 2010 así como otras herramientas como Sharepoint Designer 2010 o Visual Studio 2010, las cuales han permitido realizar las modificaciones más avanzadas.

Índice general

1. Introducción	1
1.1. Contexto Profesional	1
1.1.1. Hiberus Tecnología	1
1.2. Contexto Tecnológico	2
1.2.1. Microsoft Sharepoint 2010	2
1.2.2. Tecnologías .NET	2
1.2.3. Microsoft SQL Server 2008 R2	3
1.2.4. Microsoft Reporting Services 2008 R2	3
1.2.5. JavaScript	3
1.3. Objetivos y motivación del proyecto	3
1.4. Metodología	4
1.5. Estructura del documento	5
2. Trabajo realizado	7
2.1. Análisis de requisitos	7
2.1.1. Obra en curso	7
2.1.2. Ingresos y gastos	9
2.1.3. Informes	10
2.1.4. Procedimientos internos	13
2.1.5. Migración SEDA	15
2.2. Arquitectura general del sistema	15
2.3. Diseño	17
2.3.1. Obra en curso	18
2.3.2. Ingresos y gastos	18
2.3.3. Informes	19
2.3.4. Procedimientos internos	20
2.3.5. Migración SEDA	23
2.4. Implementación	24
2.4.1. Obra en curso	24
2.4.2. Ingresos y gastos	25
2.4.3. Informes	25
2.5. Verificación y validación	26

2.5.1. Pruebas unitarias	26
2.5.2. Pruebas de integración	27
2.6. Resultado Final	27
3. Conclusiones	31
3.1. Valoración técnica	31
3.2. Continuidad del proyecto	31
3.3. Valoración personal	32
Bibliografía	32
A. Diagramas Casos de Uso	37
B. Fórmulas obra en curso	41
C. Diagramas de flujo en procedimientos	43
D. Tipos de dato en procedimientos	47
D.1. Asignación de recursos	47
D.2. Desasignación de recursos	50
D.3. Baja Voluntaria	52
D.4. Cambio contractual	52
D.5. Vacaciones	54
D.6. Gastos	55
E. Alertas de Procedimientos	57
E.1. Asignación de recursos	57
E.2. Desasignación de recursos	59
E.3. Baja voluntaria	61
E.4. Cambio contractual	62
E.5. Vacaciones	63
E.6. Gastos	64

Índice de figuras

1.1. Interacción entre sistemas antes y despues del proyecto	4
1.2. Metodología iterativa e incremental	5
2.1. Ejemplo informe rentabilidad	11
2.2. Arquitectura lógica	17
2.3. Mapa de navegación de la aplicación	17
2.4. Diagrama que muestra el comportamiento de la obra en curso	19
2.5. Relaciones entre asignación, desasignación y baja voluntaria	22
2.6. Captura de pantalla de la biblioteca Obra en Curso	28
2.7. Captura de pantalla de la lista de permisos	28
2.8. Captura de pantalla de la página de generación de informes de rentabilidad	29
A.1. Diagrama de casos de uso de la obra en curso	37
A.2. Diagrama de casos de uso de la obra en curso	38
A.3. Diagrama de casos de uso de los informes	38
A.4. Diagrama de casos de uso de los procedimientos internos	39
A.5. Diagrama de casos de uso de la migración de SEDA	40
C.1. Diagrama de flujo del procedimiento de desasignación de recursos	43
C.2. Diagrama de flujo del procedimiento de asignación de recursos	44
C.3. Diagrama de flujo del procedimiento de baja voluntaria	45
C.4. Diagrama de flujo del procedimiento de cambio contractual	45
C.5. Diagrama de flujo del procedimiento de cambio solicitud de vacaciones	46
C.6. Diagrama de flujo del procedimiento de solicitud de gastos	46

Índice de tablas

2.1. Requisitos obra en curso	9
2.2. Requisitos ingresos y gastos	9
2.3. Requisitos de informes	12
2.4. Documentos que participan en los procedimientos	14
2.5. Requisitos procedimientos internos	15
2.6. Requisitos migración	16
D.1. Campos y tipo de <i>Ficha de solicitud de asignación de recursos</i>	48
D.2. Campos y tipo de <i>Ficha de solicitud de incorporación</i>	49
D.3. Campos y tipo de <i>Ficha del perfil del puesto de trabajo</i>	49
D.4. Campos y tipo de <i>Alta de usuario</i>	49
D.5. Campos y tipo de <i>Ficha de entrevista</i>	50
D.6. Campos y tipo de <i>Ficha de solicitud de desasignación de recursos</i>	51
D.7. Campos y tipo de <i>Ficha de desvinculación</i>	51
D.8. Campos y tipo de <i>Baja de usuario</i>	52
D.9. Campos y tipo de <i>Carta de cese</i>	52
D.10. Campos y tipo de <i>Ficha de cambios contractuales</i>	54
D.11. Campos y tipo de <i>Ficha de solicitud de vacaciones</i>	55
D.12. Campos y tipo de <i>Ficha de solicitud de gastos</i>	56
E.1. Estados del procedimiento de asignación de recursos	57
E.2. Transiciones y efectos del procedimiento de asignación de recursos	59
E.3. Estados del procedimiento de desasignación de recursos	60
E.4. Transiciones y efectos del procedimiento de desasignación de recursos	61
E.5. Estados del procedimiento de baja voluntaria	61
E.6. Transiciones y efectos del procedimiento de baja voluntaria	62
E.7. Estados del procedimiento de cambio contractual	62
E.8. Transiciones y efectos del procedimiento de cambio contractual	63
E.9. Estados del procedimiento de vacaciones	63
E.10. Transiciones y efectos del procedimiento de vacaciones	64
E.11. Estados del procedimiento de gastos	64
E.12. Transiciones y efectos del procedimiento de gastos	65

Capítulo 1

Introducción

El presente documento es una descripción del proyecto de desarrollo de una aplicación de intranet para la gestión económica y comercial de una entidad empresarial.

1.1. Contexto Profesional

En este apartado se realiza una breve descripción de la compañía y del problema a solucionar.

1.1.1. Hiberus Tecnología

*Hiberus Tecnología*¹ es una compañía especializada en la consultoría de negocio y la prestación de servicios tecnológicos y outsourcing. Es la empresa de tecnología líder del Valle del Ebro, referente en el mercado Español y en pleno proceso de expansión en el mercado latinoamericano. Hiberus se integra en uno de los principales grupos empresariales del sector de las TIC (Tecnologías de la Información y Comunicación) en España.

El objetivo de Hiberus es ayudar a empresas y organizaciones a que los sistemas de información les ayuden a mejorar el funcionamiento de su negocio y obtener más rentabilidad. El slogan *Crecemos Contigo* refleja el esfuerzo por implicar a los objetivos de negocio de los clientes en un nuevo modelo de relación basado en la implicación con los resultados, el pago por uso y la cooperación.

Hiberus nace de la unión de las empresas Iritec, Comex Integración e Ibercentro Media Consulting en Noviembre de 2011. Los grupos de comunicación regionales *Heraldo*² y *La Información*³ se incorporaron además como socios de referencia.

¹<http://www.hiberus.com>

²<http://www.grupoheraldo.com>

³<http://www.lainformacion.es/>

1.2. Contexto Tecnológico

A continuación, se detallan las herramientas informáticas utilizadas durante el desarrollo del proyecto realizado.

1.2.1. Microsoft Sharepoint 2010

*Microsoft SharePoint 2010*⁴ es una plataforma de colaboración empresarial para empresas e Internet, donde todas las personas involucradas en el ecosistema de una empresa es decir, socios, empleados, clientes y colaboradores puedan compartir contenidos que forman parte del proceso de negocio de la empresa, este contenido pueden ser documentos, hojas de cálculo, reportes, gráficos, ideas, conocimientos, experiencias, contactos, fotos, videos, etc.

El principal objetivo de SharePoint es aumentar la productividad y el acceso a la información oportuna, y todos sabemos que el aumento de la productividad se traduce en mayor producción o disminución de costes y por lo tanto en ganancias para la empresa.

Por otro lado sabemos también que la información es la base para la toma de decisiones por lo que SharePoint puede ayudar a la empresa en este proceso de toma de decisiones ofreciendo más información, de mayor calidad y de manera más oportuna, acerca de las distintas áreas de la empresa y por otra parte no hace falta decir que decisiones acertadas se traducen en ganancias y nuevas oportunidades de negocio.

1.2.2. Tecnologías .NET

*Microsoft.NET*⁵ es el conjunto de tecnologías en las que Microsoft ha trabajado con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

*ASP.NET*⁶ es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Su finalidad es la de facilitar la creación de sitios web dinámicos.

⁴<http://office.microsoft.com/es-es/sharepoint/>

⁵<http://www.microsoft.com/net>

⁶<http://asp.net-tutorials.com>

C# [1] es un lenguaje de programación simple pero eficaz, diseñado para escribir aplicaciones empresariales. Es una evolución de los lenguajes C y C++. Utiliza muchas de las características de C++ en las áreas de instrucciones, expresiones y operadores y presenta considerables mejoras e innovaciones en áreas como seguridad de tipos, control de versiones, eventos y recolección de elementos no utilizados (liberación de memoria).

1.2.3. Microsoft SQL Server 2008 R2

*Microsoft SQL Server 2008 R2*⁷ es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o DB2(IBM).

1.2.4. Microsoft Reporting Services 2008 R2

*Microsoft Reporting Services 2008 R2*⁸ es una plataforma de informes basada en servidor que proporciona la funcionalidad completa de generación de informes para una gran variedad de orígenes de datos. Incluye un conjunto completo de herramientas para que crear, administrar y entregar informes, y las API que permiten a los desarrolladores integrar o ampliar el procesamiento de datos e informes en aplicaciones personalizadas. Los informes pueden ser interactivos, tabulares, gráficos o de forma libre

1.2.5. JavaScript

JavaScript es un lenguaje de programación que permite desarrollar acciones en páginas web. No requiere compilación ya que se ejecuta desde el cliente, los navegadores son los encargados de interpretar estos códigos.

1.3. Objetivos y motivación del proyecto

El objetivo principal de este PFC es el desarrollo de una serie de aplicaciones para control interno de *Hiberus*, a través de las cuales se pueda agilizar y automatizar la integración de la información de distintos sistemas de gestión internos para el control y seguimiento del personal y de los proyectos en curso. Actualmente *Hiberus* cuenta con distintos sistemas de gestión:

- *Rosseta*: Orientado a Intranet, es una herramienta de comunicación interna para

⁷<http://www.microsoft.com/es-es/download/details.aspx?id=30438>

⁸<http://technet.microsoft.com/es-es/library/ms159106.aspx>

todos los empleados que ofrece una eficiente vía de acceso a la gestión, colaboración e información.

- *SEDA*: Sistema de gestión Comercial y Clientes, es una utilidad corporativa de registro, gestión, seguimiento y análisis unificado de la información.
- *ADN*: Sistema de Gestión y Control de Tareas y Recursos, es una aplicación corporativa de registro, gestión, seguimiento y análisis de todas las actividades/proyectos/servicios desarrolladas en la empresa.

La finalidad de este proyecto es la integración de parte de ellos en una única aplicación haciendo uso de las posibilidades y potencialidad que ofrece Sharepoint. Esta unión, reflejará una mejora de la gestión y automatización de los procedimientos. La figura 1.1 muestra la interacción entre los sistemas antes y después del desarrollo de este proyecto.

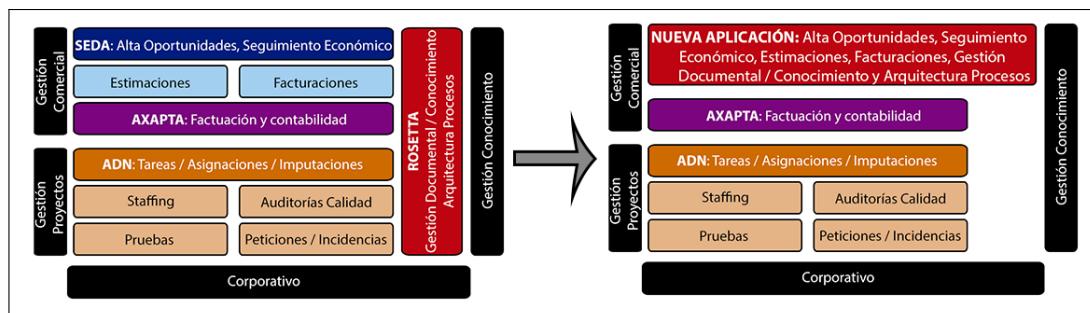


Figura 1.1: Interacción entre sistemas antes y después del proyecto

A nivel personal, se plantean otra serie de objetivos orientados a la consecución del título de Ingeniero Superior en Informática:

- Conseguir una mejor visión de la estructura de una empresa.
- Mejora de la capacidad de autoaprendizaje.
- Integración en un grupo de trabajo.
- Búsqueda de soluciones óptimas ante un problema real.

1.4. Metodología

Para la realización de este proyecto se ha utilizado una metodología de desarrollo iterativo e incremental [2] (*véase Figura 1.2*). Esta metodología de trabajo se caracteriza por dividir el proyecto en bloques o tareas. En cada bloque se repite un proceso de trabajo

similar para proporcionar un resultado completo sobre el producto final. Las etapas han seguido una metodología en cascada en la que cada fase no puede comenzar hasta que su fase previa haya terminado. Las fases de cada tarea han sido las siguientes: Análisis de requisitos, diseño, implementación, pruebas y mantenimiento.

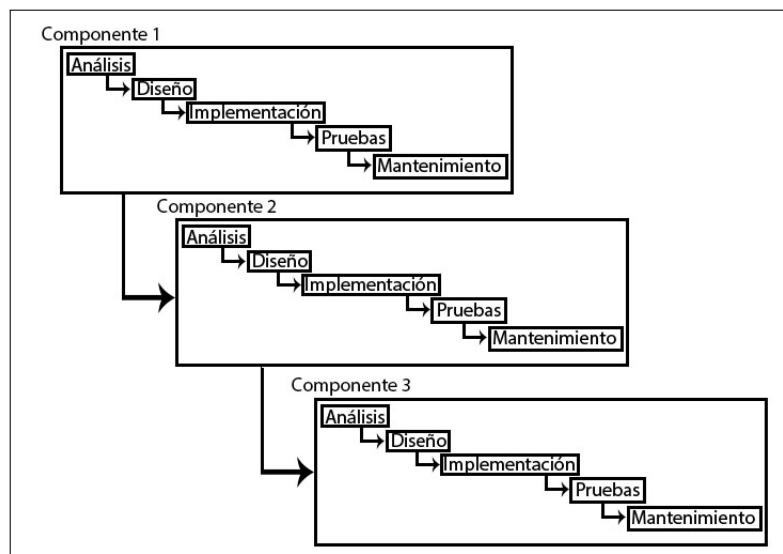


Figura 1.2: Metodología iterativa e incremental

1.5. Estructura del documento

A continuación, se detallan las siguientes secciones que forman parte de este documento:

- *Capítulo 1*: Es el apartado en el que nos encontramos y que sirve de introducción al resto de la memoria.
- *Capítulo 2*: Esta sección describe el trabajo realizado para la consecución del proyecto. Se especifica la arquitectura utilizada y las diferentes etapas de todo proyecto software: análisis, diseño, implementación, verificación y validación.
- *Capítulo 3*: Muestra una conclusión general sobre el desarrollo del proyecto, su continuidad y valoraciones técnicas y personales.

Capítulo 2

Trabajo realizado

En este apartado de la memoria, se especifica todo el trabajo desarrollado durante el transcurso del proyecto.

2.1. Análisis de requisitos

En esta sección se describe el comportamiento que debe tener cada uno de los componentes que forman la aplicación de este proyecto fin de carrera. En el anexo A se pueden consultar los diagramas de casos de uso de cada uno de los componentes de la aplicación.

2.1.1. Obra en curso

La obra en curso es un informe de rentabilidad que crea mensualmente *Hiberus*, en él aparecen reflejados diversos datos (fijos y variables por mes) ligados a los proyectos que maneja la compañía. A continuación, se detallan los datos fijos que se manejan de cada proyecto: código, cliente, proyecto, importe total, costes derivados, importe para proyecto, total facturado, pendiente facturar, estado, DIM 1, DIM 2, DIM 3, DIM 4, DIM 6 y proyecto cerrado.

Cada proyecto está identificado por 6 dimensiones: DIM 1 (o dirección general), DIM 2 (o unidad de gestión), DIM 3 (o responsable comercial), DIM 4 (o tipo de servicio), código y DIM 6 (o delegación). Proyecto cerrado nos indica si en realidad es un proyecto o una asistencia técnica.

Los datos que varían cada mes por proyecto son los siguientes: horas totales, horas periodo, avance total, avance periodo, incurrido total, incurrido periodo, coste derivado, facturado mes y estado facturas.

Toda esta información se extrae de la base de datos *SIIIC* (base de datos de SEDA) a través de la ejecución de un procedimiento almacenado cuyos parámetros de entrada son:

fecha de inicio, fecha de fin y el identificador de la unidad de gestión (0 para extraer de todas).

El problema que se plantea es la necesidad de extraer todos estos datos en una tabla Excel (.xlsx) calculando nuevos datos mensuales por proyecto usando como fuente los datos descritos anteriormente. Esta tabla se podrá aceptar (consolidar toda la información en base de datos) o rechazar (volver a generar). Los nuevos datos calculados mensualmente son los siguientes:

- *Horas*: Horas acumuladas hasta el periodo actual.
- *Incurrido*: Importe trabajado hasta la fecha actual.
- *Facturado*: Facturado acumulado hasta el mes actual.
- *Costes Derivados*: Costes totales del proyecto hasta el periodo actual.
- *Obra Producida Periodo*: Indicador que evalúa el trabajo interno realizado durante cada periodo.
- *Obra Producida Acumulada*: Indicador que evalúa el trabajo interno realizado en el año en curso.
- *Generado Periodo*: Indicador que valora el trabajo realizado durante cada periodo.
- *Generado Acumulado*: Indicador que valora el trabajo realizado durante todos los meses del año actual.
- *Obra En Curso*: Indicador que valora el trabajo realizado frente a la facturación.
- *Euro/Hora Periodo*: Indicador que indica el valor de cada hora respecto a la obra producida del último mes.
- *Euro/Hora Acumulado*: Indicador que indica el valor de cada hora respecto a la obra producida desde el inicio del curso.
- *Tipo Hora*: Distingue cinco tipos de horas (Inversión, productivas, externas, comerciales y de garantía).

La tabla 2.1 detalla los requisitos de este componente de la aplicación.

2. Trabajo realizado

2.1 Análisis de requisitos

ID	Descripción
R1	La aplicación guardará backups en base de datos de la obra en curso consolidada mensualmente
R2	La aplicación generará automáticamente la tabla .xlsx el día 6 de cada mes
R3	La aplicación permitirá consolidar el documento excel deseado en base de datos
R4	La aplicación permitirá volver a generar los datos de la tabla excel, siempre y cuando dicha tabla no esté ya consolidada en base de datos
R5	La aplicación mostrará errores en el caso de que los hubiera, indicando la columna de qué proyecto da el error en la tabla .xlsx
R6	Cada documento .xlsx tendrá cinco estados: pendiente, consolidado, rechazado, en proceso y con errores
R7	Los cálculos entre columnas deberán aparecer reflejados en el documento (Ej: =H2+J2/K2)
R8	Cada columna de la tabla deberá tener filtros con el fin de ordenar las distintas filas por el criterio deseado
R9	La aplicación deberá estar orientada a Sharepoint y a su arquitectura

Tabla 2.1: Requisitos obra en curso

2.1.2. Ingresos y gastos

Mensualmente, la administración de *Hiberus* actualiza un documento Excel en el que van apareciendo reflejados los diferentes ingresos y gastos (tanto directos como indirectos) de cada proyecto que desarrolla la empresa. Algunas de las columnas más representativas de dicho documento son: mes, cliente SEDA, epígrafe, varios (es el código del proyecto), medio, submedio, cliente/proveedor, importe, descripción, tipo de ingreso (directo, indirecto o ninguno) y tipo de gasto (directo, indirecto o ninguno).

El único objetivo de esta parte es que exista una forma de consolidar este archivo en base de datos como con el componente anterior.

La tabla 2.2 detalla los requisitos de este componente de la aplicación.

ID	Descripción
R1	La aplicación permitirá consolidar una hoja de ingresos y gastos en base de datos
R2	La aplicación avisará de posibles fallos en la hoja en el caso de que los hubiera
R3	La aplicación deberá estar orientada a Sharepoint y a su arquitectura

Tabla 2.2: Requisitos ingresos y gastos

2.1.3. Informes

Mensualmente, se realizan dos tipos distintos de informes: de rentabilidad y de horas. Ambos utilizan los datos de las hojas Excel citadas en los dos apartados anteriores y se realizan mediante el uso de tablas dinámicas en Excel.

2.1.3.1. Informe de rentabilidad

Este informe se divide en otros dos subinformes:

- *Ingreso-Gasto (Rentabilidad)*: Muestra la cantidad de gastos e ingresos de cada medio y submedio durante los meses transcurridos del año en curso junto a la obra e interno producidos por cada uno.
- *Generado-Gasto*: Muestra la cantidad de generado y de gastos de cada medio y submedio durante los meses transcurridos del año actual.

Ambos subinformes poseen unos filtros fijos comunes: tipo de usuario, costes (directos o indirectos), meses, medios y submedios. El filtro tipo de usuario modifica la vista del informe de tal forma que se pueden ver medios contenidos dentro de submedios y viceversa.

Existen cinco filtros optativos que en función de su valor aumentan el número de niveles de cada fila, por defecto es dos: medio y submedio (véase figura 2.1). Estos filtros poseen los siguientes valores: cliente SEDA, cliente/proveedor, epígrafe, varios (o código de proyecto) o descripción.

2.1.3.2. Informe por horas

Representa la cantidad de horas de cada tipo (productivas, de inversión, comerciales y de garantía) invertidas por cada medio/ unidad de gestión de la compañía en cada uno de los meses transcurridos del año actual. El informe permite filtrar las columnas por meses y las filas por unidad de gestión.

El objetivo principal de la aplicación es que genere dinámicamente los distintos informes. En el caso del informe de rentabilidad, la aplicación permitirá al departamento de operaciones asignar un listado de medios y submedios visibles por cada usuario. También debe existir la posibilidad de reportar los detalles de un informe de rentabilidad al departamento de operaciones para su posterior revisión. Destacar que en Excel, al pulsar sobre cualquier celda de un informe, ésta nos redirige a la tabla fuente de datos filtrada de tal forma que muestra únicamente los detalles de la celda.

2. Trabajo realizado

2.1 Análisis de requisitos

Informe de Rentabilidad con dos niveles, ningún filtro optativo aplicado									
		Enero							
Medio	Submedio	Comercial	Garantía	Productivas	Inversión	Total	Total
M1	Total	10	50	0	25	85	85
	S1

Total		10	50	0	25	85	85

Informe de Rentabilidad con 3 niveles, filtro 'Epígrafe' aplicado									
		Enero							
Medio	Submedio	Epígrafe	Comercial	Garantía	Productivas	Inversión	Total	...	Total
M1	Total		10	50	0	25	85	...	85
	S1	Total
		E1
	
Total			10	50	0	25	85	...	85

Figura 2.1: Ejemplo informe rentabilidad

Otra de las tareas a realizar será mostrar ambos informes de forma gráfica mediante un gráfico de barras en el que se muestren las cantidades por meses, así como una tabla debajo del gráfico que detalle las cantidades en forma numérica.

La tabla 2.3 detalla los requisitos de este componente de la aplicación.

ID	Descripción
Requisitos generales para los informes	
R1	La aplicación deberá estar orientada a Sharepoint y a su arquitectura
R2	Existirán dos tipos de informe: rentabilidad y horas
R3	La aplicación permitirá generar un gráfico del tipo de informe elegido
R4	Tanto los informes como las gráficas se podrán exportar a formato .xls o .pdf
R5	Existirá la posibilidad de ver los detalles de cualquier dato de un informe
Requisitos dedicados del informe por horas	
R6	El informe por horas contendrá los filtros: medio y mes (desde enero hasta el mes actual). Ambos filtros serán de elección multiple.
R7	El informe por horas representará la cantidad de horas de cada tipo (ordenadas por mes) de los medios y meses seleccionados
R8	En el informe por horas se mostrarán cifras totales, tanto por filas como por columnas
R9	El gráfico del informe por horas estará formado por un diagrama de barras en el que aparecerá la cantidad de cada tipo de horas por mes y una tabla en la que podremos ver los mismos datos de forma numérica
Requisitos dedicados del informe de rentabilidad	

R10	Se podrán reportar los detalles de un informe de rentabilidad para que sea revisado por el Departamento de Operaciones
R11	Los tipos de reporte son: OC, generado, ingresos y gastos, varios y error - tipo proyecto
R12	Al realizar un reporte se introducirá un comentario y se indicará el tipo de reporte
R13	El reporte deberá contener un archivo excel en el que aparezcan reflejados los datos reportados por el usuario
R14	El Departamento de Operaciones podrá añadir un comentario a un reporte dado
R15	Un reporte podrá tener uno de los siguientes estados: solicitado, desestimado, aceptado, pendiente modificación admin, pendiente modificación operaciones, subsanado o cancelado por el usuario
R16	La aplicación tendrá la opción de administrar los medios y submedios que cada usuario podrá seleccionar
R17	El informe de rentabilidad contendrá los siguientes filtros: tipo de informe, tipo de usuario, costes, medio, submedio, mes (desde enero hasta el mes actual), mostrar obra, mostrar interno y cinco filtros optativos (servirán para desglosar cada medio/submedio en función de su: cliente SEDA, cliente/proveedor, epígrafe, código de proyecto o descripción)
R18	El filtro tipo de informe permitirá elegir entre dos subinformes: rentabilidad(ingresos-gasto) o generado-gasto. En función del tipo seleccionado cada mes del informe se dividirá en ingresos y gastos o en generado y gastos
R19	El filtro tipo de usuario tendrá los siguientes valores: MAC y Responsable. En función de la selección, los medios se desglosarán en submedios (MAC) o los submedios se dividirán en medios (Responsable)
R20	El filtro costes tendrá dos valores: todos y directos. En función de la selección, se utilizarán en el informe gastos directos o indirectos y directos
R21	El filtro mostrar obra nos permitirá mostrar la obra como un mes más si el tipo de subinforme elegido es 'rentabilidad'
R22	El filtro mostrar interno nos permitirá mostrar el interno como un mes más si el tipo de subinforme elegido es 'rentabilidad'
R23	Los valores de los filtros medio y submedio serán dedicados en función del usuario. El departamento de Operaciones tiene que poder asignar a cada usuario los medios y submedios que un usuario puede visualizar
R24	En los dos subinformes se mostrarán cifras totales, tanto por filas como por columnas
R25	El gráfico del informe de rentabilidad estará formado por un diagrama de barras en el que aparecerá la cantidad de ingresos-gastos/generado-gasto (dependiendo del tipo de subinforme elegido) por mes y una tabla en la que podremos ver los mismos datos de forma numérica

Tabla 2.3: Requisitos de informes

2.1.4. Procedimientos internos

Hiberus posee un conjunto de procedimientos internos para la gestión de sus empleados. De este conjunto, se ha decidido que se implanten en una arquitectura Sharepoint: asignación de recursos, desasignación de recursos, baja voluntaria, cambios contractuales, vacaciones y gastos.

2.1.4.1. Asignación de recursos

El proceso se inicia cuando un responsable de área detecta la necesidad de incorporación de un nuevo recurso a su equipo de trabajo. Inicialmente, se buscan reasignaciones internas que cumplan con el perfil del puesto y en caso de no haber, se buscan externas. Una vez elegidos uno o varios candidatos por el solicitante, el Departamento de Operaciones decidirá cuál de ellos será el elegido. Para finalizar, se da de alta al usuario en el sistema. Una solicitud de este tipo puede tener los siguientes estados: pendiente de evaluación, pendiente de información, evaluada negativamente, evaluada positivamente, asignación interna, asignación externa, pendiente de validar por el solicitante, validado, seleccionado, en proceso de tramitación y incorporado.

2.1.4.2. Desasignación de recursos

El proceso se inicia cuando un responsable detecta la necesidad de desasignar a un recurso de su unidad de gestión. Desde el Departamento de Operaciones se intentará reasignar el recurso en otro centro de trabajo. Si no ha sido posible la reasignación, se procederá a la desvinculación del recurso y a la baja de éste en el sistema. Una solicitud de este tipo puede tener los siguientes estados: pendiente de evaluación, pendiente de información, evaluada negativamente, evaluada positivamente, en proceso de reasignación, reasignado, desasignado pendiente de desvinculación, pendiente de evaluar condiciones, aceptado, rechazado y desvinculado.

2.1.4.3. Baja voluntaria

El proceso se inicia cuando cualquier recurso de la empresa desea desvincularse de ésta. El Departamento de Operaciones informa al responsable del recurso de la desvinculación y se da de baja al recurso en el sistema. Una solicitud de este tipo puede tener los siguientes estados: pendiente de evaluación, pendiente de información, evaluada positivamente y desvinculado.

2.1.4.4. Cambio contractual

El proceso se inicia cuando un responsable tiene la necesidad de realizar una revisión contractual de un recurso de su equipo de trabajo. El CEO de la compañía evaluará la situación y comunicará su decisión al solicitante y a los departamentos de Operaciones y Recursos Humanos. Una solicitud de este tipo puede tener los siguientes estados: pendiente

de evaluación, pendiente de información, evaluada negativamente, evaluada positivamente, rechazado, aceptado y modificado.

2.1.4.5. Vacaciones

El proceso se inicia cuando cualquier recurso desea disfrutar de días de vacaciones, o por motivos personales, ajenos a la empresa, deba ausentarse del puesto de trabajo. La solicitud la valorará el responsable del recurso en las fechas en las que se deseen las vacaciones. Una solicitud de este tipo puede tener los siguientes estados: pendiente de evaluación, evaluada negativamente, evaluada positivamente, concedida y denegada.

2.1.4.6. Gastos

El proceso se inicia cuando cualquier recurso realiza un pago adicional causado por desplazamiento debido a un proyecto o visita a clientes. El solicitante facilitará los comprobantes de los gastos a su responsable. En el caso de que el responsable acepte la solicitud, el propio recurso remitirá los comprobantes al Departamento de Administración desde donde se introducirán dichos gastos en la hoja de ingresos y gastos. Una solicitud de este tipo puede tener los siguientes estados: pendiente de evaluación, pendiente de información, evaluada negativamente, evaluada positivamente y cobrado.

En cada uno de los procedimientos participan un conjunto de documentos internos (véase tabla 2.4).

Procedimiento	Archivo	Campos
Asignación de recursos	Ficha de solicitud de asignación de recursos	44
	Ficha de solicitud de incorporación	13
	Ficha del perfil del puesto de trabajo	6
	Alta de usuario	9
	Ficha de entrevista	16
Desasignación de recursos	Solicitud de desasignación de recursos	17
	Ficha de desvinculación	18
	Baja de usuario	5
Baja de usuario	Carta de cese	5
	Baja de usuario	5
Cambio contractual	Ficha de cambios contractuales	78
Gastos	Ficha solicitud de gastos	28
Vacaciones	Ficha solicitud de vacaciones	13

Tabla 2.4: Documentos que participan en los procedimientos

En la tabla 2.5 podemos ver un resumen de los requisitos de este componente del

sistema.

ID	Descripción
R1	Cualquier usuario podrá visualizar su listado de solicitudes de vacaciones y crear nuevas
R2	Cualquier usuario podrá generar una solicitud de baja voluntaria
R3	Cualquier usuario podrá ver su listado de solicitudes de gastos y crear nuevas
R4	Cualquier responsable de área podrá ver su listado de solicitudes de cambios contractuales y generar nuevas
R5	Cualquier responsable de área podrá ver su listado de solicitudes de asignación y crear nuevas
R6	Cualquier responsable de área podrá ver su listado de solicitudes de desasignación y crear nuevas
R7	El departamento de operaciones podrá ver todas las solicitudes existentes y realizar los cambios oportunos
R8	En cada uno de los listados, se podrá observar como mínimo de cada solicitud: estado, fecha de la última modificación, persona que modificó por última vez la solicitud.
R9	La aplicación deberá estar orientada a Sharepoint y a su arquitectura

Tabla 2.5: Requisitos procedimientos internos

2.1.5. Migración SEDA

SEDA es el actual sistema de gestión comercial y clientes que utiliza la compañía. El objetivo es realizar una migración parcial de la aplicación. Las partes a migrar son: gestión de ofertas, gestión de acciones, gestión de clientes, facturaciones, actividad comercial y seguimiento económico.

La tabla 2.6 detalla los requisitos de este componente de la aplicación.

2.2. Arquitectura general del sistema

La arquitectura de la aplicación es similar a la utilizada por Sharepoint 2010 porque el sistema ha sido diseñado bajo este entorno empresarial (*véase Figura 2.2*).

A continuación se detallan los diferentes componentes de la aplicación:

- *Home*: En toda aplicación Sharepoint debe existir un sitio de nivel superior del que desciendan el resto de sitios de la aplicación, Home es el sitio principal de nuestra aplicación.

ID	Descripción
R1	Se podrá visualizar un listado de ofertas
R2	Se podrán crear nuevas ofertas
R3	Una oferta podrá ser modificada desde el listado de ofertas
R4	Se podrá visualizar un listado de acciones
R5	Se podrán crear nuevas acciones
R6	Una acción podrá ser modificada desde el listado de acciones
R7	Se podrá visualizar un listado de clientes
R8	Se podrán crear nuevos clientes
R9	Un cliente podrá ser modificado desde el listado de clientes
R10	Deberá existir la posibilidad de realizar el seguimiento económico de los proyectos de la compañía
R11	El usuario podrá generar facturas agrupadas y desglosadas
R12	La aplicación permitirá realizar la actividad comercial mediante la generación de cuatro tipos distintos de informe: contratos firmados, oportunidades, ofertas presentadas y facturación de contratos
R13	Este componente de la aplicación deberá poseer una interfaz lo más similar posible a la que posee SEDA

Tabla 2.6: Requisitos migración

- *Informes*: En este sitio existirán las diferentes estructuras necesarias para la gestión de la obra en curso, ingresos y gastos e informes.
- *Administración*: Será el sitio desde donde se podrán asignar los permisos necesarios para la generación de informes de rentabilidad.
- *SEDA*: Este sitio contendrá la mayoría de las opciones de SEDA.
- *Facturación*: Desde este sitio se podrán generar facturas y realizar el seguimiento económico de proyectos.
- *Procedimientos*: Sitio que gestionará los diferentes procedimientos internos de Hibebus.

Respecto a las bases de datos, se pueden distinguir tres:

- *BDADN*: Base de datos del sistema ADN.
- *SIIC*: Base de datos del sistema SEDA.
- *BDApuntes*: Base de datos de la nueva aplicación.

2. Trabajo realizado

2.3 Diseño

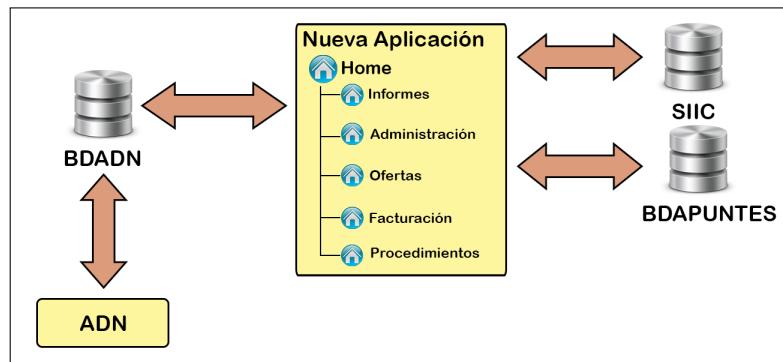


Figura 2.2: Arquitectura lógica

2.3. Diseño

En esta sección se describe el modelado del sistema en base a los requerimientos especificados en el apartado anterior. La figura 2.3 representa el mapa de navegación que seguirá la aplicación.

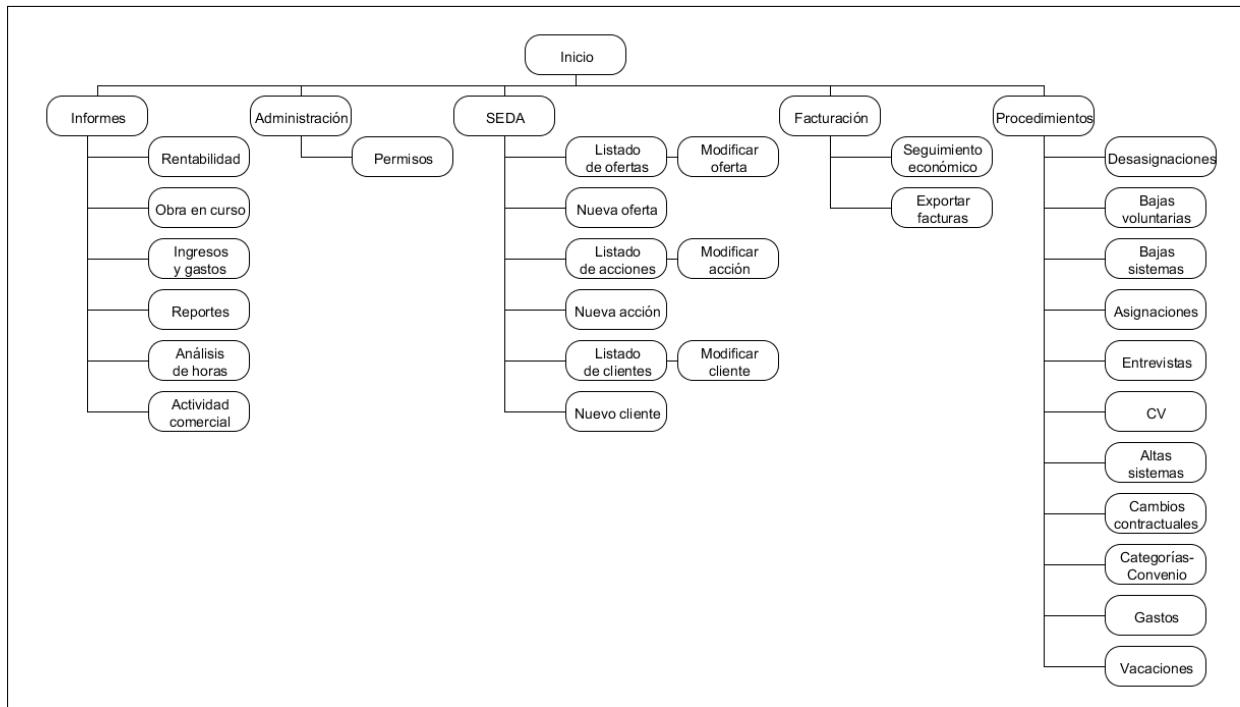


Figura 2.3: Mapa de navegación de la aplicación

2.3.1. Obra en curso

El primer paso será la generación de fórmulas que calculen los diferentes datos mensuales de la obra. En el anexo B se pueden ver de forma matemática los diferentes cálculos.

Como el usuario podrá revisar, rechazar y consolidar una obra, se creará una biblioteca de documentos en Sharepoint donde almacenar las diferentes obras mensuales. Esta biblioteca, tendrá una columna llamada estado, de tipo elección, que al cambiar de valor ejecutará un evento de actualización del ítem dado. Este evento consolidará la obra en base de datos o la volverá a generar en función del estado. Para facilitar la revisión de una obra en los diferentes estados por los que haya pasado, se habilitará el historial de versiones. Otra columna a destacar de esta biblioteca es InfoError que indicará al usuario si hay algún tipo de error en la obra, indicando la columna y el proyecto dónde se ha encontrado el error. En la figura 2.4 se puede ver un diagrama que muestra los estados que tendrá la obra en curso de un mes dado.

La base de datos *BDApунtes* tendrá una tabla llamada *HojaRentabilidad* en la que se insertarán los datos de rentabilidad de cada proyecto por periodo. Al consolidar una obra, se borrarán todos los registros que posea la tabla en ese momento, se insertarán los nuevos y al finalizar, se creará una copia exacta de la tabla como backup de los datos de rentabilidad del periodo.

Por último, uno de los objetivos era que automáticamente se generará la obra el sexto día de cada mes. Para ello, se creará una aplicación de escritorio alojada en el servidor que genere la obra en curso del mes anterior al actual y la aloje en la biblioteca de documentos. Usando el programador de tareas de Windows Server, indicaremos que se ejecute la aplicación el sexto día de cada mes.

2.3.2. Ingresos y gastos

Al ser la hoja de ingresos y gastos un documento, se ha decidido utilizar una biblioteca de documentos que vaya almacenando mensualmente los documentos. Como los documentos que sean subidos a esta biblioteca ya han sido revisados, se ha decidido utilizar un evento que se ejecute al añadir un nuevo ítem en la biblioteca. Este evento, enviará los datos a una tabla en *BDApунtes* llamada *tabAxapta* que tiene las mismas columnas que el documento. Ante la posibilidad de existir algún fallo en el documento, se creará una columna llamada InfoError, de tipo línea de texto, que indicará si existe algún error o si los datos se han guardado correctamente en *BDApунtes*. Antes de guardar los datos, se borrarán todos los registros existentes en *tabAxapta*.

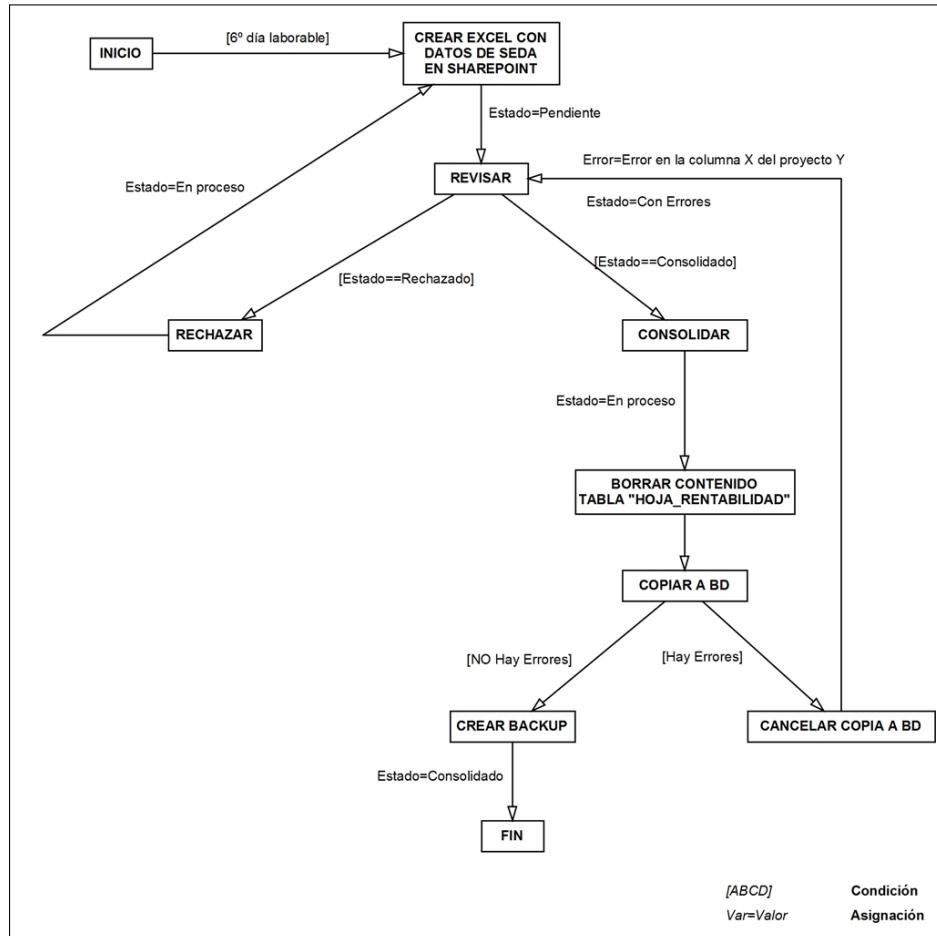


Figura 2.4: Diagrama que muestra el comportamiento de la obra en curso

2.3.3. Informes

Al tomar los datos de *tabHojaRentabilidad* y *tabAxapta* los dos tipos de informe, se ha decidido crear una vista llamada *HojaTotal* que combine ambas tablas en una sola con la estructura de la segunda. Para realizar la creación de los informes se utilizará Microsoft SQL Reporting Services tomando como fuente de datos la vista anteriormente citada. Se deberá activar el servicio de reporting en *BDApuntes*, ya que por defecto viene desactivado.

En el informe de rentabilidad, el principal problema que aparece es que puede tener varios niveles en función de cuantos filtros optativos se apliquen. No se puede crear un informe de siete niveles y que dependiendo de los seleccionados se oculten o muestren las filas/columnas necesarias ya que las columnas al ocultarlas dejan un espacio en blanco en mitad de la tabla. Por esta razón se crearán 6 informes, uno por cada número de filtros optativos seleccionados [0, 5]. A parte, se crearán otros dos informes extra, uno para vi-

sualizar los detalles de una celda dada de los anteriores y otro que muestre el gráfico de barras junto a la tabla de cantidades.

Para gestionar los permisos de los filtros medio y submedio, se creará una lista Sharepoint con las siguientes columnas destacables:

- *Persona*: Será de tipo persona e indicará el usuario al que se le asignan los permisos.
- *Medios*: Será de tipo datos externos y obtendrá los datos de la tabla de medios existente en SIIC.
- *Submedios*: Será de tipo datos externos y obtendrá los datos de la tabla de submedios existente en SIIC.

En cuanto a la funcionalidad de reportes de un informe detallado, se generará una lista de reportes en la que cada ítem tendrá como adjunto un archivo Excel con los mismos datos que el usuario haya reportado.

En cuanto al informe por horas, se crearán tres tipos de informe: general, detallado y gráfico.

Para mostrar ambos tipos de informe en el sitio web, se crearán dos páginas ASP.NET que ofrecerán las funcionalidades de cada uno de los informes por separado.

2.3.4. Procedimientos internos

En este apartado se va a describir el comportamiento que debe tener cada uno de los procedimientos por separado, excepto, los procedimientos de asignación, desasignación y baja voluntaria ya que están muy ligados entre sí.

En el anexo C se muestran los diagramas de flujo de cada uno de los procedimientos existentes.

En el anexo D se pueden encontrar una serie de tablas que muestran el tipo de dato de cada uno de los campos de los diferentes archivos que participan en cada procedimientos.

En el anexo E se muestran las alertas y efectos surgidos de los diferentes cambios de estado de cada tipo de solicitud.

La generación de un código alfanumérico para distinguir a las diferentes solicitudes es un problema que aparece en todos los procedimientos. Por defecto, cada ítem de una lista/biblioteca tiene un campo llamado ID que es único y autoincremental. A priori, se podría utilizar este campo para generar el código como campo de tipo calculado, pero no se puede de forma directa porque el ID es asignado una vez el ítem ha sido ya creado. Para solucionar este inconveniente:

- Se hará uso de una columna intermedia, de tipo línea de texto y oculta, llamada Identificador que por defecto tendrá el valor *Generando*.
- Se creará un workflow que hará la siguiente asignación: *Item.Identificador:= Item.ID*
- Este workflow se ejecutará al crear un nuevo ítem en la lista.
- Código será un campo de tipo calculado que usará la columna intermedia, si ésta tiene el valor *Generando* sabremos que aún podemos generar el código.

2.3.4.1. Asignación de recursos / Desasignación de recursos / Baja voluntaria

Tras revisar los documentos que participan en el proceso, se ha decidido que el único que seguirá existiendo será el Curriculum Vitae (CV). Los campos del resto de documentos serán columnas de listas/bibliotecas que serán cumplimentadas mediante el uso de formularios. Se crearán las siguientes listas en base a las relaciones que se muestran en la figura 2.5:

- *Lista de bajas voluntarias*: En el desplegable de un ítem, aparecerá una nueva acción que servirá para informar de la baja a sistemas. Esta acción nos mandará a un nuevo formulario de creación de un ítem de la lista bajas de sistemas, a este formulario se le pasará como parámetro el ID del ítem de baja voluntaria/desasignación. De esta forma, se creará una baja de sistemas que estará referenciada con su origen mediante una columna de búsqueda.
- *Biblioteca de CV*: Para facilitar la creación de entrevistas desde un CV, se creará una acción en la lista desplegable de cada ítem llamada entrevistar.
- *Lista de entrevistas*: Como una entrevista puede ser asignada a una solicitud de asignación que permita asignar una entrevista a una o varias solicitudes de asignación existentes. Se creará un evento que refuerce la entrevista con las solicitudes de asignación seleccionadas. La lista poseerá una columna de tipo verificación que indicará la disponibilidad de ésta con el fin de que no pueda ser asignada a ninguna otra solicitud de asignación.
- *Lista de solicitudes de asignación*: Tendrá seis columnas de tipo búsqueda para gestionar entrevistas: candidatas externas, candidatas internas, validadas externas,

validadas internas, seleccionada externa y seleccionada interna. Tendrá una acción en el despegable de un ítem que permita crear una solicitud de alta de sistemas.

- *Lista de solicitudes de desasignación:* Tendrá una columna de búsqueda para relacionar una solicitud de desasignación con una de asignación para el caso en el que se haya procedido a una reasignación interna. También tendrá dos columnas llamadas *medio origen* y *medio destino* para dejar constancia del cambio y una acción en el despegable de un ítem que permita crear una solicitud de baja de sistemas.
- *Lista de bajas de sistemas:* Tendrá una columna de tipo búsqueda que relacione un ítem con una solicitud de desasignación.
- *Lista de altas de sistemas:* Tendrá una columna de tipo búsqueda que relacione un ítem con una solicitud de asignación.

Las columnas de búsqueda no aplican filtros, entonces una solicitud de asignación podrá validar de entre las candidatas alguna entrevista que posteriormente haya sido marcada como no disponible. Para solucionar este problema, se utilizará una característica implementada por Codeplex¹ que permitirá tener columnas de búsqueda que usen consultas para filtrar elementos².

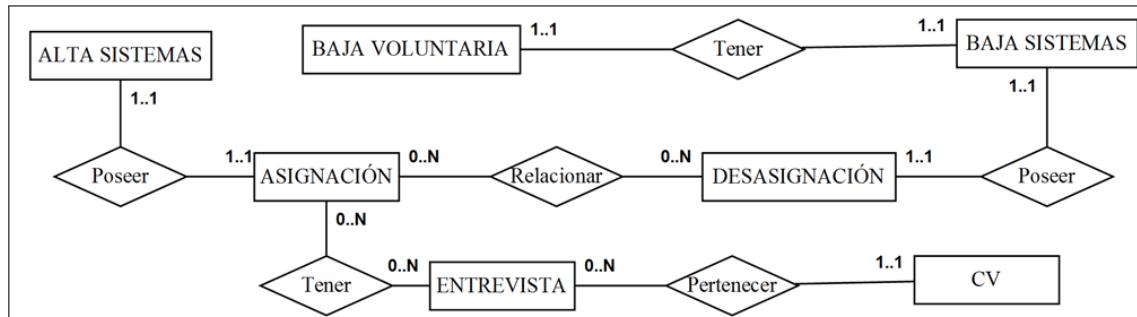


Figura 2.5: Relaciones entre asignación, desasignación y baja voluntaria

2.3.4.2. Cambios contractuales

Al eliminarse el único documento que participa, se ha decidido que se creará una lista de solicitudes de cambios contractuales en la que aparecerán los campos del documento como columnas de la lista. Este documento utiliza una hoja maestra oculta con salarios en función de la categoría seleccionada. Estos datos, son usados en algunas de las columnas calculadas de la lista de solicitudes de cambios contractuales. Esta información

¹<https://www.codeplex.com>

²<http://filteredlookup.codeplex.com/>

debe seguir existiendo, por lo que deberá existir una lista que almacene la información de categoría/convenio.

2.3.4.3. Vacaciones

Al eliminarse el único documento que participaba, se ha decidido que se creará una lista de solicitudes de vacaciones en la que aparecerán los campos del documento como columnas de la lista.

2.3.4.4. Gastos

Al eliminarse el único documento que participaba, se ha decidido que se creará una lista de solicitudes de gastos en la que aparecerán los campos del documento como columnas de la lista. Un gasto tiene recibos asociados con el fin de justificar dicho gasto, éstos aparecerán como archivos adjuntos de un gasto dado. En el documento aparecen sumatorios por categorías (gasolina, peaje...). Por defecto en Sharepoint, se puede activar la vista hoja de datos que muestra la lista con aspecto tabla excel y ofrece la posibilidad de mostrar los totales de cada una de las columnas.

2.3.5. Migración SEDA

Tras examinar las páginas relevantes para la migración de SEDA, se ha decidido convertirlas en controles de usuario que puedan ser llamados desde diferentes páginas ASP.NET en Sharepoint. Para ello, se modificará el código necesario para su correcto funcionamiento, prestando especial atención a fragmentos de código en JavaScript que pueden ser críticos en un entorno Sharepoint.

SEDA utiliza una biblioteca que actúa como capa intermedia entre la aplicación y las bases de datos *SIIC* y *BDADN*. Esta biblioteca será transformada en una dll a través de la cual se puedan utilizar los diferentes métodos implementados en ella.

Se intentará conservar una interfaz lo más similar posible a la actual con el fin de que sea lo más cercana al usuario.

2.4. Implementación

En esta sección se citan y explican algunas de las funciones más representativas de la aplicación. En todos los componentes se hace uso de la biblioteca ClosedXML³ creada por Codeplex para el tratamiento de archivos Excel. Se ha descartado la biblioteca oficial de Microsoft porque para usarla es obligatorio que en el servidor esté instalado Microsoft Office.

2.4.1. Obra en curso

La función más representativa de este componente es aquella en la que se sube un archivo Excel que contiene la obra en curso a una biblioteca de documentos. Esta función se ejecuta con privilegios de administrador para evitar problemas de permisos.

Para subir un archivo a una biblioteca de documentos, se debe convertir previamente el archivo a un array de bytes. Mediante el uso de HashTable se asignan los diferentes atributos que tendrá el ítem y por último, se agrega toda la información a la biblioteca. A continuación se muestra el fragmento de código de dicha función.

```
using Microsoft.SharePoint;
...
protected void UploadDocumentToSP(byte[] byt, String filename, int anyo, int mes){
    try {
        SPSecurity.RunWithElevatedPrivileges(delegate(){
            using (SPSite objSite = new SPSite(sp_site)){
                using (SPWeb objWeb = objSite.OpenWeb(sp_sitio_excel)){
                    objWeb.AllowUnsafeUpdates = true;
                    SPFolder mylibrary = objWeb.Folders[sp_biblio_produccion];

                    Hashtable properties = new Hashtable();
                    properties.Add("vti_title", Path.GetFileNameWithoutExtension(filename));
                    properties.Add("Estado", "Pendiente");
                    properties.Add("Año", anyo);
                    properties.Add("Mes", mes);

                    mylibrary.Files.Add(filename, byt, properties, true);
                    mylibrary.Update();
                    objWeb.AllowUnsafeUpdates = false;
                }
            });
        });
    }
    catch (Exception ex){
        LogError("UploadDocumentToSP:" + ex.Message + " - " + ex.StackTrace);
    }
}
```

³<https://closedxml.codeplex.com>

2.4.2. Ingresos y gastos

En este componente, la función más representativa es aquella perteneciente al evento que consolida los datos en BDApuntes al agregar un archivo a la biblioteca de ingresos y gastos. Para ello, primero se guarda un archivo temporal con el contenido de los ingresos y gastos. Una vez guardado, se procede a la lectura de este y conversión en un DataTable para su posterior paso a BD. Por último, se borra el archivo temporal del servidor y se actualiza el ítem.

```
using Microsoft.SharePoint;
...
public override void ItemAdded(SPItemEventProperties properties){
    base.ItemAdded(properties);
    EventFiringEnabled = false;
    try{
        if (properties.ListItem != null){
            SPListItem oItem = properties.ListItem;

            byte[] binFile = oItem.File.OpenBinary();
            System.IO.Directory.CreateDirectory("C:/Temporales");
            String ruta = "C:/Temporales/temporalAxapta.xlsx";
            FileStream fstream = File.Create(ruta);
            fstream.Write(binFile, 0, binFile.Length);

            XLWorkbook libro = new XLWorkbook(fstream);
            fstream.Close();
            File.Delete(ruta);
            Directory.Delete("C:/Temporales");
            //Debo parsear para pasar a la tabla
            IXLRange rango = libro.Worksheet(1).RangeUsed();
            DataTable tabla_Axapta = XLSXToTabla(libro, rango);

            //Se manda a BD
            bool hayError = false;
            String error = saveAxapta(tabla_Axapta, ref hayError);
            oItem["Info_Error"] = error;
            oItem.Update();
        }
    }
    catch (Exception ex){
        LogError("ItemAdded:" + ex.Message + " - " + ex.StackTrace);
    }
    EventFiringEnabled = true;
}
```

2.4.3. Informes

La función más destacable para la gestión de informes sería aquella que carga los medios y submedios de un usuario para el caso de un informe de rentabilidad. Para ello se busca en la lista de permisos al usuario y en el caso de aparecer, se rellenan los listados con el contenido de las columnas de la lista referentes a los medios y submedios visibles por el usuario. Se aplican privilegios de administrador para evitar problemas.

```
using Microsoft.SharePoint;
```

```

...
protected void LoadMediosSubmedios(){
    try{
        SPSecurity.RunWithElevatedPrivileges(delegate(){
            using (SPSite objSite = new SPSite(sp_site)){
                using (SPWeb objWeb = objSite.OpenWeb(sp_sitio_admin)){
                    objWeb.AllowUnsafeUpdates = true;
                    String nombre_usuario = SPContext.Current.Web.CurrentUser.LoginName;
                    SPList oList = objWeb.Lists[sp_lista_permisos];
                    SPQuery myQuery = new SPQuery();
                    myQuery.Query = "<Where>" + "<Eq>" + "<FieldRef Name=\"Persona\"/>" + "<Value
Type=\"Text\">" + nombre_usuario + "</Value>" + "</Eq>" + "</Where>";
                    SPListItemCollection myItems = oList.GetItems(myQuery);
                    if(myItems.Count==1){
                        SPListItem item = myItems[0];
                        String[] Medios = item["Medios"].ToString().Split(',');
                        foreach (String Medio in Medios)
                            if (Medio.Replace("#", "") != null && Medio.Replace("#", "") != "")
                                ListaMedios.Items.Add(Medio.Replace("#", ""));
                        String[] Submedios = item["Submedios"].ToString().Split(',');
                        foreach (String Submedio in Submedios)
                            if (Submedio.Replace("#", "") != null && Submedio.Replace("#", "") != "")
                                ListaSubmedios.Items.Add(Submedio.Replace("#", ""));
                    }
                    objWeb.AllowUnsafeUpdates = false;
                }
            });
        }
    catch (Exception ex){
        LogError("LoadMediosSubmedios: " + ex.Message + " - " + ex.StackTrace);
    }
}

```

2.5. Verificación y validación

En esta sección se van a explicar algunas de las pruebas realizadas más significativas en el transcurso del proyecto. Cabe destacar, que aquellas pruebas que en un principio no se han realizado satisfactoriamente, se ha hecho todo lo posible por encontrar los errores hasta lograr realizarlas con éxito.

2.5.1. Pruebas unitarias

Las pruebas unitarias se realizan sobre cada uno de los componentes del sistema por separado. Algunas de las más destacables se explican a continuación:

- Generar una obra en curso de enero a diciembre y comprobar que no hay error de overflow en el número de columnas
- Rechazar varias obras en curso a la vez con el fin de observar que no se sobrepasa ningún Timeout.
- Comprobación de los cálculos en la obra en curso y del uso de fórmulas excel.

- Consolidación correcta en BD para la obra en curso y para los ingresos y gastos.
- Creación de ofertas, acciones, clientes y su posterior modificación en SEDA (migrado).
- Comprobar los efectos en el informe de rentabilidad al asignar/desasignar permisos a un determinado usuario.
- Comprobar la generación correcta de un reporte de un informe de rentabilidad.

2.5.2. Pruebas de integración

Este tipo de pruebas se realizan una vez se han integrado los componentes en el sistema. Algunas de las más destacables se explican a continuación:

- Creación en SEDA (migrado) de ofertas, facturaciones e incurridos y la posterior comprobación de la existencia de estos datos en la obra en curso al volver a generarla.
- Eliminación de las tablas tabHojaRentabilidad y tabAxapta para posteriormente ver los resultados presentados en la generación de informes.

2.6. Resultado Final

En esta sección se muestran capturas de pantalla de la aplicación con el fin de poder apreciar parte del trabajo desarrollado.

En la figura 2.6 se puede observar una captura de pantalla de la biblioteca de documentos de obras en curso.

En la figura 2.7 se puede visualizar una captura de pantalla de la lista de permisos para los informes de rentabilidad.

En la figura 2.8 se puede apreciar una captura de pantalla de la generación de informes de rentabilidad.

2.6 Resultado Final

2. Trabajo realizado

Figura 2.6: Captura de pantalla de la biblioteca Obra en Curso

Figura 2.7: Captura de pantalla de la lista de permisos

2. Trabajo realizado

2.6 Resultado Final

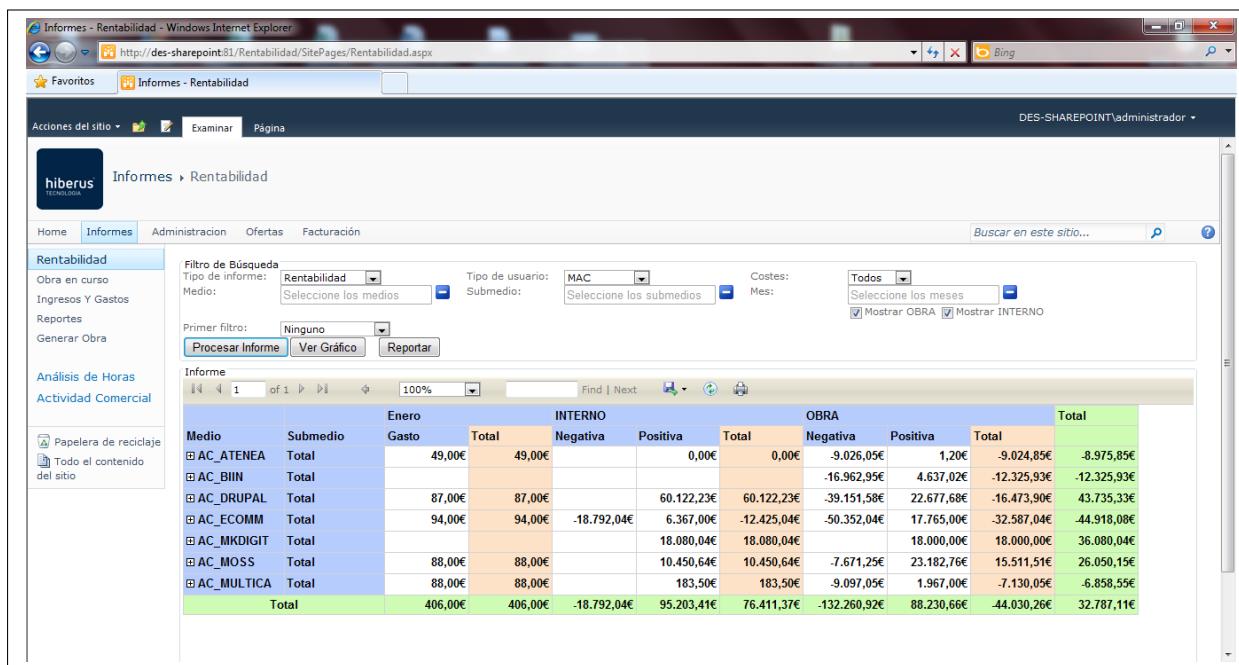


Figura 2.8: Captura de pantalla de la página de generación de informes de rentabilidad

Capítulo 3

Conclusiones

En este apartado se va a hacer una valoración, tanto personal como técnica, del proyecto y se van a establecer las líneas futuras de éste.

3.1. Valoración técnica

Como se ha descrito a lo largo de este documento, se han conseguido los objetivos propuestos al inicio de este trabajo de forma satisfactoria.

Se ha aprendido una tecnología específica como es Sharepoint, exprimiéndola al máximo posible con el fin de alcanzar una experiencia de usuario aceptable. Gracias a este aprendizaje, se han logrado automatizar procesos de la compañía que se realizaban periódicamente de forma manual, como pueden ser la obra en curso o la generación de informes.

Para cada uno de los problemas propuestos, se han estudiado diferentes soluciones con el objetivo de tener un gran abanico de posibilidades y elegir la solución más óptima en cada problema para la empresa.

3.2. Continuidad del proyecto

En el futuro se realizará la implementación de los procedimientos internos y la creación de grupos de usuario para gestionar de una manera más eficiente los permisos en determinadas zonas de la aplicación. Previamente, se deberá activar el uso de metadatos administrados en el servidor e insertar los términos necesarios ordenados por categorías.

Una vez puesta en producción la aplicación, se procederá a la integración de ADN en el nuevo sistema con el fin de tener un único sistema de gestión de la información en Hiberus.

Mientras se realicen las tareas anteriormente descritas, se irán arreglando fallos detectados por los usuarios y modificando el contenido de determinadas zonas en función de las exigencias de la dirección de la compañía.

3.3. Valoración personal

Gracias a la realización de este trabajo a través de Hiberus, se ha obtenido una visión empresarial y una experiencia en un entorno de trabajo empresarial que antes no se tenían.

Desde un enfoque académico, este proyecto me ha permitido consolidar conceptos aprendidos durante los últimos años en la universidad, destacando las asignaturas de *Sistemas de Información, Ingeniería del Software y Bases de Datos*.

En definitiva, este PFC me ha sido de gran utilidad para obtener conocimientos en una plataforma totalmente desconocida para y me ha permitido diseñar una herramienta corporativa que establece un canal de comunicación para los diferentes departamentos y servicios integrados en la empresa.

Bibliografía

- [1] *C# Language Specification*, Anders Hejlsberg, Scott Wiltamuth, and Peter Golde. 2003. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
- [2] Metodología iterativa incremental: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>
- [3] Google: <http://www.google.es>
- [4] Stackoverflow: <http://stackoverflow.com/>
- [5] Microsoft Developer Network: <http://msdn.microsoft.com/>

