



UNIVERSIDAD
DE MÁLAGA

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS
DESARROLLO DE UNA APLICACIÓN DE CONTABILIDAD PARA
UNA EMPRESA DE ALIMENTACIÓN

Realizado por
CHRISTIAN PAREJA JENSEN

Dirigido por
JOSÉ MARÍA ÁLVAREZ PALOMO

Departamento
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

MÁLAGA, enero de 2013

UNIVERSIDAD DE MÁLAGA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Ingeniería Técnica en Informática de Sistemas

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente/a Dº/Dª. _____

Secretario/a Dº/Dª. _____

Vocal Dº/Dª. _____

para juzgar el proyecto Fin de Carrera titulado:

Desarrollo de una aplicación de contabilidad para una empresa de
alimentación

del alumno D. Christian Pareja Jensen

dirigido por D. José María Álvarez Palomo

ACORDÓ POR _____ OTORGAR LA CALIFICACIÓN DE
_____ Y PARA QUE CONSTE, SE EXTIENDE
FIRMADA POR LOS COMPARECIENTES DEL TRIBUNAL, LA PRESENTE
DILIGENCIA.

Málaga a ____ de _____ del 201__

El Secretario

El Presidente

El Vocal

Fdo: _____ Fdo: _____ Fdo: _____

Contenido

1. Introducción.....	1
1.1. Definición / Contexto	1
1.2. Objetivos Generales	1
1.3. Situación Actual	3
1.4. Soluciones	4
1.5. Recursos	5
2. Herramientas de Desarrollo	6
2.1. Lenguaje de Programación	6
2.2. Entorno de desarrollo	8
2.3. Operadores de consulta.....	9
2.4. Base de Datos.....	10
2.5. Otros programas	13
2.6. Métrica v3	14
2.6.1. Introducción a Métrica versión 3	14
2.6.2. Aportaciones de Métrica Versión 3.....	15
2.6.3. Procesos Principales de Métrica Versión 3	17
2.6.4. Planificación de Sistemas de Información.....	19
2.6.5. Desarrollo de Sistemas de Información	20
2.6.6. Mantenimiento de Sistemas de Información.....	29
3. Planificación del Sistema de Información (PSI)	31
3.1. Inicio del Plan de Sistemas de Información	31
4. Desarrollo del Sistema de Información	32
4.1. Estudio de Viabilidad del Sistema (EVS).....	32
4.2. Análisis del Sistema de Información (ASI).....	33
4.3. Diseño del Sistema de Información (DSI).....	37
4.4. Construcción del Sistema de Información (CSI).....	63
4.5. Implantación y Aceptación del Sistema (IAS)	68

5. Mantenimiento del Sistema de Información (MSI)	71
5.1. Sistema de Almacenamiento	71
5.2. Ubicación física para la copia	74
5.3. Software para la copia	75
5.4. Contenido de la copia	76
6. Conclusiones y Líneas Futuras	79
6.1. Líneas Futuras de Trabajo	79
6.2. Conclusiones	81
7. Manual de Usuario	83
8. Apéndice	119
9. Bibliografía	130

1. Introducción

1.1. Definición / Contexto

El proyecto que se va a desarrollar consiste en la realización de una aplicación de contabilidad para una empresa de alimentación. En concreto, esta empresa se dedica a la compra, elaboración y posterior venta de pescado ahumado.

Una aplicación de contabilidad o software contable está destinada a sistematizar y simplificar las tareas de contabilidad. El Software contable registra y procesa las transacciones históricas que se generan en una empresa o actividad productiva: las funciones de compras, ventas, cuentas por cobrar, cuentas por pagar, control de inventarios, balances, producción de artículos, nóminas, etc. Para ello hay que introducir la información requerida, como las pólizas contables, ingresos y gastos, y hacer que el programa realice los cálculos necesarios.

La empresa para la que se va a desarrollar este proyecto, la cual dispone de una plantilla de más de 200 personas, divididas en distintos departamentos, entre los que se encuentra, el departamento de contabilidad, para el cual se va a desarrollar esta aplicación. Además tiene aproximadamente unos 150 proveedores a quienes compra la materia prima que produce, y otros 100 clientes a los que vende los pescados ahumados ya elaborados. Este tamaño hace que la gestión de la contabilidad sea notablemente más complicada que en el caso de una pequeña empresa.

1.2. Objetivos Generales

Alcance Funcional del Proyecto:

- **Gestión del Plan Contable**

Engloba la gestión de cuentas y subcuentas, así como conceptos y distintos centros de costes. Además permitirá la elaboración de planes contables.

- **Apuntes y Diarios Contables**

El Apunte o Diario Contable es el libro donde se registran diariamente todas las transacciones económicas que realiza un comerciante o una empresa comercial o industrial. Se podrá realizar entradas, mantenimiento, edición o borrado de los apuntes.

- **Facturas Recibidas y Emitidas**

Reflejan toda la información de una operación de compraventa. La información fundamental que aparece en una Factura debe reflejar la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar.

- **Balances**

Son unos informes financieros que reflejan la situación del patrimonio de una empresa en un momento determinado. El estado de situación financiera se estructura a través de tres conceptos patrimoniales, el activo, el pasivo y el patrimonio neto, desarrollados cada uno de ellos en grupos de cuentas que representan los diferentes elementos patrimoniales.

- **Presupuestos**

Los Presupuestos son programas en los que se les asignan cifras a las actividades; implican una estimación de capital, de los costos, de los ingresos, y de las unidades o productos requeridos para lograr los objetivos.

- **Utilidades y Cierres**

El Cierre Contable es el proceso consistente en cerrar o cancelar las cuentas de resultados y llevarlas a las cuentas de balance respectivas.

- **Extractos**

Se trata de consultas a la base de datos de valores introducidos anteriormente. Estas se pueden realizar por fecha, cuentas, diario, cliente, centro de coste, proyectos, etc.

1.3. Situación Actual

Echando un vistazo a la forma de trabajar actual de la empresa, nos encontramos con un programa hecho a medida, similar al que nos han pedido que creemos. Vamos a ver en que consiste el antiguo, y si nos puede servir de ayuda o de base para construir el nuevo.

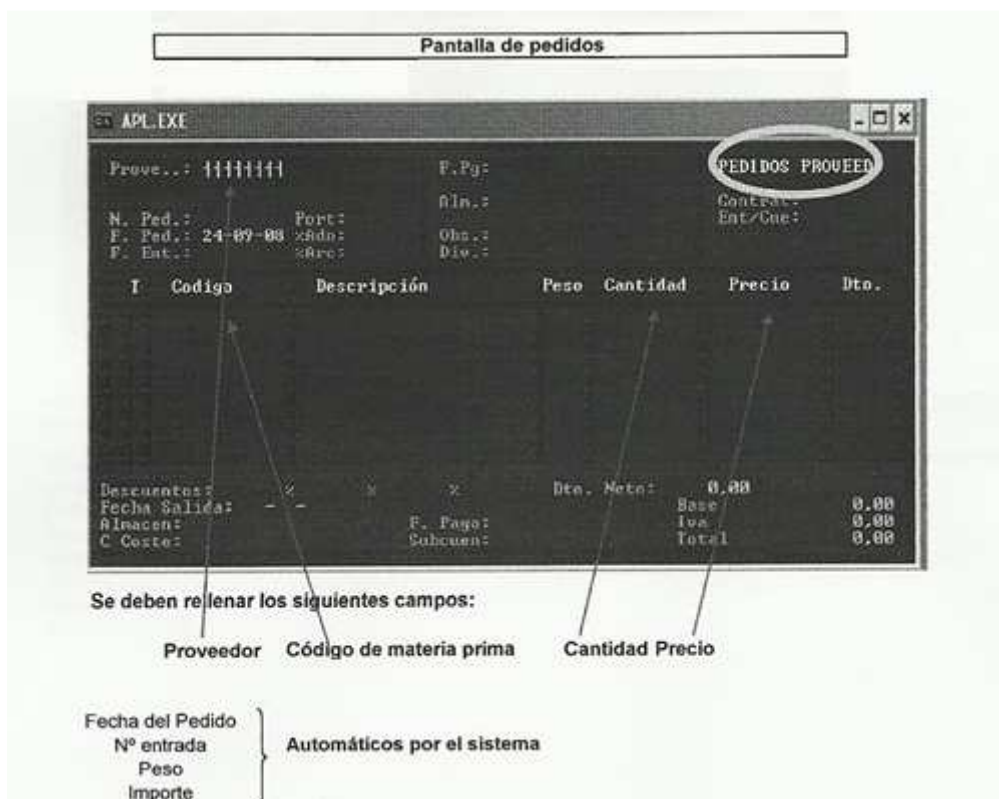


Como se puede ver en la imagen, se trata de un programa bastante antiguo, concretamente del año 1995, construido sobre MS-DOS, el cuál evidentemente a nivel visual es bastante limitado, pero que cumple perfectamente las expectativas que hay sobre él.

Las opciones son mayores que las que se le esperan al nuevo, ya que a lo largo de los años, al igual que se le han ido añadiendo nuevas funciones, otras quedaron totalmente obsoletas y en desuso.

Las virtudes del sistema actual son, ante todo, que el hecho de haber sido usado durante años por los trabajadores hace que estén muy habituados a él, y el cambio a otro nuevo pueda provocar momentos de confusión y necesidad de aprendizaje. Es por ello que el nuevo debe ser lo más similar y sencillo posible.

Y como ya hemos mencionado, los defectos son, a parte de las opciones extras innecesarias que ya no tienen uso, el hecho de estar programado bajo un lenguaje y un sistema anticuado.



1.4. Soluciones

Ante un caso como este, en el que se quiere cambiar el programa actual por uno más moderno, se abre un abanico de posibilidades. Por un lado, están los programas de contabilidad que hay ya en el mercado. Se trata de programas amplios, que ofrecen muchas posibilidades, pero que evidentemente no están hechos a medida.

Como ya dijimos, uno de los objetivos es facilitar al usuario el cambio desde el sistema actual al nuevo, por lo que estos tipos de aplicaciones fueron descartadas.

Y por otro lado, tenemos la posibilidad de crear un programa desde cero, hecho a medida, lo cual se ha decidido como la mejor opción. Dentro de esta posibilidad, existen infinidad de lenguajes y programas con los que hacerlos. Para elegir el más apropiado, compararemos las distintas opciones en el apartado 2. Herramientas de Desarrollo.

A su vez, también habría que ver las soluciones que se le dan a las necesidades de la empresa, pero en este caso no será necesario, ya que será el cliente directamente el que nos traslade esas necesidades y nos limitaremos a implementarlas.

1.5. Recursos

El entorno de desarrollo elegido inicialmente para este proyecto es el de Microsoft. Este entorno se usa ampliamente para el desarrollo de este tipo de aplicaciones y, además, es el entorno en el que están desarrolladas las actuales aplicaciones que maneja la empresa. Entre otras incluye las siguientes herramientas:

- **Microsoft Visual Studio**, un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.
- **Visual Basic .NET (VB.NET)**, un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET o Mono para ejecutarse.
- **.NET**, un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.
- SQL Server Express Edition. El **lenguaje de consulta estructurado** o **SQL** es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Profundizaremos más en estas herramientas, así como indicaremos los motivos por los cuales nos hemos decidido por ellos a continuación, en el apartado 2, Herramientas de Desarrollo.

Por otro lado, debemos mencionar que dentro de las herramientas no solo hablamos de los programas que usaremos para programarlo, sino también de la metodología en la que nos basaremos para ello. En nuestro caso, hemos optado por métrica v3 como la mejor opción. En el siguiente apartado, explicaremos cada una de ellas y los motivos por los cuales nos hemos decidido a escogerlas.

2. Herramientas de Desarrollo

2.1. Lenguaje de Programación

Para ver que programas vamos a utilizar para este proyecto debemos empezar desde el punto más bajo e ir subiendo niveles, eligiendo los lenguajes y programas más apropiados en cada momento.

El primer paso sería elegir el lenguaje de programación en el cual programar. Existen muchos y muy variados, sin embargo para el programa que pretendemos hacer las opciones razonables se reducen.

Recordemos que buscamos un programa específico para Windows en cualquiera de sus versiones, que se trata de hacerlos funcionar en unos 50 ordenadores distintos con sus características cada uno, pero que en su mayoría se tratan de computadoras lentas, ya que una empresa como esta no se puede permitir mantener modernizados constantemente todos sus terminales.

Por tanto, una de las prioridades será la rapidez en ordenadores “lentos”. Además toda la información irá conjunta en un servidor al cual todos accederán para sacar información de la base de datos, es decir, necesitamos un lenguaje de programación pensado para interactuar fácilmente con la base de datos.

Más cosas a tener en cuenta son los usuarios que no serán informáticos y usarán el programa como apoyo a sus trabajos de contabilidad, por lo que sería recomendable que el programa tenga un interfaz lo más sencillo y visual posible para que su adaptación sea rápida y fácil.

Con estas premisas, nos quedamos con dos opciones más que factibles, estas son, programar en .net o utilizar java. Veamos primero cada una de las opciones y sus ventajas e inconvenientes.



Java es un lenguaje de programación de alto nivel orientado a objetos, desarrollado por James Gosling en 1995. El lenguaje en sí mismo toma mucha de su sintaxis de C, Cobol y Visual Basic, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo, hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones, o como la misma plataforma las denomina, soluciones, permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

En sí la verdadera competencia exacta de .Net no es JAVA sino J2EE (Java 2 Enterprise Edition). J2EE es un grupo de especificaciones que forman lo que es denominado Java Application Server, la principal ventaja que J2EE esté basado en especificaciones es la libertad de elección sobre vendedores, esto es, los componentes escritos en Java son interoperables entre productos J2EE desarrollados por IBM, HP, Sun, BEA... A diferencia de .NET donde todo gira alrededor un solo vendedor: Microsoft.

Pero .NET ha tenido la oportunidad de aprender de los errores de J2EE. Su diseño está bien definido y estructurado, cumpliendo todas las exigencias de una buena ingeniería de software (comparado a J2EE).

Esta moderna plataforma fue diseñado con tres objetivos en mente: primero debería lograr aplicaciones Windows muchos más estables, aunque también debería proporcionar una aplicación con un mayor grado de seguridad. En segundo lugar, debería simplificar el desarrollo de aplicaciones y servicios web que no solo funcionen en plataformas tradicionales sino también en dispositivos móviles. Por último, el entorno fue diseñado para proporcionar un solo grupo de bibliotecas que pudieran trabajar con varios lenguajes (que actualmente suman 20).

Por tanto, como ambas nos abren un abanico de posibilidades parecidas, elegiremos .net como la solución más apropiada orientada al programa que pretendemos crear.

2.2. Entorno de desarrollo

La gran mayoría de programadores de VB.NET utilizan el entorno de desarrollo integrado Microsoft Visual Studio en alguna de sus versiones (desde el primer Visual Studio .NET hasta Visual Studio .NET 2010, que es la última versión de Visual Studio para la plataforma .NET), aunque existen otras alternativas, como SharpDevelop (que además es libre).

Al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET o Mono para ejecutarse.

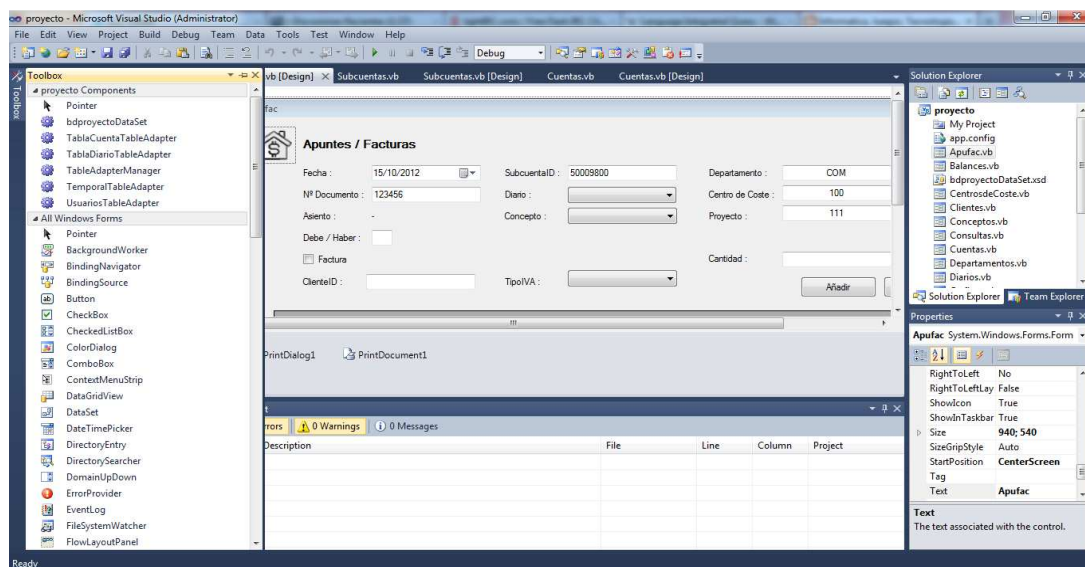
SharpDevelop es un entorno de desarrollo integrado libre para los lenguajes de programación C#, Visual Basic .NET, F#, Python, Ruby, Boo y C++.

Es usado típicamente por aquellos programadores de los citados lenguajes, que no desean o no pueden usar el entorno de desarrollo de Microsoft, Microsoft Visual Studio. Hay disponible un port para Mono/Gtk#, llamado MonoDevelop, el cual funciona en otros sistemas operativos.

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

2. Herramientas de Desarrollo

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

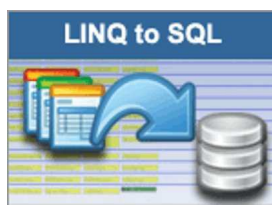


En este caso usaremos Microsoft Visual Studio como mejor opción para programar en .net directamente, para sacar el mayor partido posible a las opciones que ofrece el lenguaje de programación elegido anteriormente.

2.3. Operadores de consulta

Antes de llegar a la Base de Datos, tenemos que tratar un paso intermedio, el cual, por los programas que vamos a usar, no es necesario hacer nada, pero si es bueno saber que existe y en qué consiste.

Language Integrated Query (**LINQ**) es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET, inicialmente a los lenguajes Visual Basic .NET y C#. Muchos conceptos que LINQ ha introducido fueron originalmente probados en C, un proyecto de investigación de Microsoft.



LINQ define *operadores de consulta estándar* que permiten a lenguajes habilitados con LINQ filtrar, enumerar y crear proyecciones de varios tipos de colecciones usando la misma sintaxis. Tales colecciones pueden incluir vectores (arrays), clases enumerables, XML, conjuntos de datos desde bases de datos relacionales y orígenes de datos de terceros. El proyecto LINQ usa características de la versión 2.0 del .NET Framework, nuevos ensamblados relacionados con LINQ, y extensiones para los lenguajes C# y Visual Basic .NET.

El objetivo de crear LINQ es permitir que todo el código hecho en Visual Studio (incluidas las llamadas a bases de datos, datasets, XMLs) sean también orientados a objetos. Antes de LINQ, la manipulación de datos externos tenía un concepto más estructurado que orientado a objetos. Además LINQ trata de facilitar y estandarizar el acceso a dichos objetos.

Por tanto, gracias a esta unión podremos poner consultas directamente en el código que creamos en Visual Studio y la Base de Datos SQL, entenderá las mismas y devolverá los resultados que se le pidan.

2.4. Base de Datos

Y por último, necesitaremos una base de datos donde guardar toda la información que los usuarios quieran almacenar, por tanto tendremos que elegir la que más nos convenga para este proyecto.

Mientras que para llevar una contabilidad, una agenda de contactos, facturas... almacenar datos “alfanuméricas” sirve cualquier producto del mercado, el “motor” de un sistema de gestión documental, el “gestor” de la base de datos “documental” tiene unos requisitos mucho más exigentes. Hace ya muchos años que la continua evolución del hardware ha resuelto los problemas de capacidad y rendimiento que plantea guardar en un PC corriente un fichero con 40, 50 ó 300 mil registros. Los datos “alfanuméricos”, nombres, apellidos, números de teléfono, cuentas contables... ocupan unos cuantos bytes y, por muchos que tengamos, siempre estaremos hablando de kilobytes. Unos pocos “Megas” a lo sumo. ¿Qué son 3 ó 4 Mb en un PC que tiene un disco duro con 500Gb, o, lo que es lo mismo, unos 500.000 Mb? Es despreciable. Es un problema resuelto.

Pero si hablamos de gestión documental, la cosa cambia.

Ahora un “registro” de la base de datos, cada uno de los elementos que la componen, es un fichero. Puede ser un documento con texto de Word (20, 30, 140Kb...), PDF (50, 100, 200Kb), Excel (otro *tanto*)... pero puede que tengamos imágenes, logotipos, fotos, documentos escaneados con una buena resolución... con lo que ya el orden de magnitud empieza a ser de Mb, más que Kb. No es difícil que un usuario incorpore 2 ó 3 mil documentos al año, con lo que nos empezamos a mover (en una PYME de 10 usuarios) en cifras de varios Gb anuales. Esto ya no es tan fácil de gestionar. Y mucho menos sencillo es garantizar un rendimiento adecuado al introducir los nuevos documentos, y al consultar los existentes. Ya no me vale una base de datos tipo Access: necesitamos algo “serio”.

“Libres” o gratuitas hay varias: MySQL, PostgreSQL, Firebird, SQLite... las dos primeras son las más conocidas y, sin duda, son válidas para un proyecto, por complicado que sea.

“Propietarias” hay muchas más. Dejando de lado las menos “potentes”, dBASE (un clásico, pero muy superada), FileMaker, Interbase, Access... tenemos las 2 más conocidas, Oracle y SQL Server de Microsoft, además de IBM DB2, Informix, Progress... que es raro verlas en PYMES.

¿Cuál elegir? La respuesta es muy sencilla: todas sirven. Todas. Esto ya está inventado. Cualquiera de ellas es válida, si se configura y utiliza bien. La decisión no es en función de que sirva o no para mi necesidad. Seguro que te sirve. La decisión estará basada en otras características.

Finalmente, por la línea que estamos tomando, después de haber elegido como opción de entorno de programación el de Microsoft, y sabiendo de la existencia y utilidad de LINQ, parece lógica la elección de SQL Server de Microsoft. Veamos en que consiste:

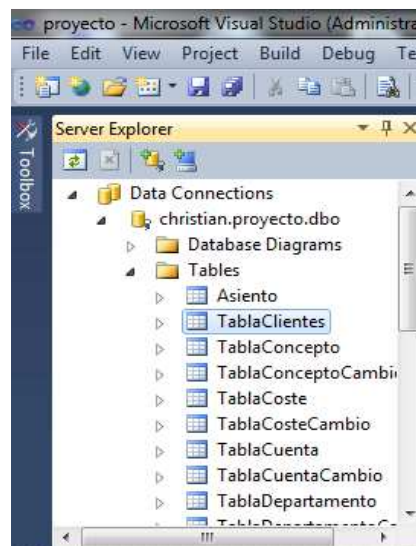
Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son *Oracle*, *PostgreSQL* o *MySQL*.

Características de Microsoft SQL Server:

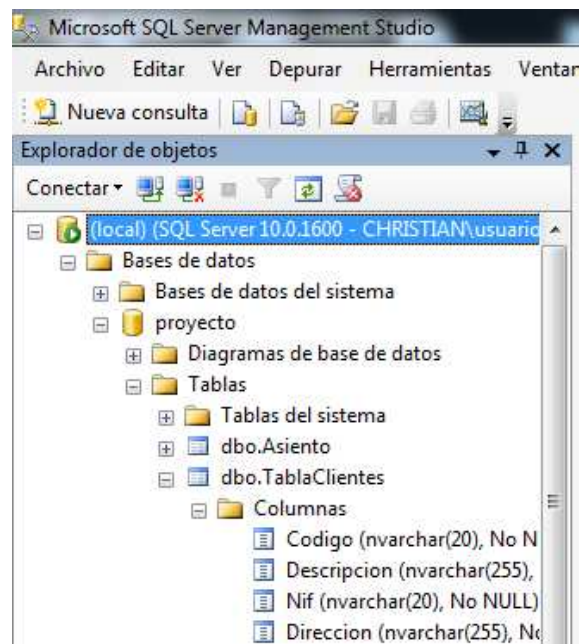
- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.

Para manejar SQL Server de Microsoft, necesitamos una herramienta, con la cual crear cada una de las tablas, con sus columnas, restricciones y enlaces. Además esta herramienta nos facilitará la posibilidad de introducir datos directamente, información que se tuviera previamente y que se pretenda agregar antes de poner el programa en funcionamiento. Esta herramienta, se puede encontrar por dos vías. La primera sería usar directamente **Visual Studio**. Como decidimos elegir el camino de Microsoft, los programas están fuertemente relacionados y adaptados unos a los otros por lo que directamente podemos manejar la Base de Datos desde nuestro entorno de programación. Aquí podemos ver una captura de como trae esta opción:



Microsoft SQL Server Management Studio Express es un entorno gratuito e integrado para obtener acceso, configurar, administrar y desarrollar todos los componentes de SQL Server, así como para combinar un amplio grupo de herramientas gráficas y enriquecidas editoras de scripts que proporcionan acceso a SQL Server para programadores y administradoras de todos los niveles.



2.5. Otros programas

Por último cabe mencionar también la utilización de editor de imágenes, para crear los menús, las imágenes de fondo y también todos los iconos. Para esto hemos alternado entre uno más sencillo, como es Microsoft Paint, y uno más complejo, que nos ofrece todas las posibilidades que queramos crear, Adobe Photoshop.

Photoshop es uno de los programas más famosos (o el que más) de Adobe, enfocado principalmente al tratamiento de imágenes digitales, pero que abarca desde la manipulación fotográfica hasta la pintura digital, pasando por el diseño web, edición de videos, etc.

La primera versión de esta aplicación vio la luz allá por el año 1990 y en exclusiva para Mac. Poco a poco fueron incluyendo nuevas características, como la inclusión del trabajo por capas, diferentes espacios de color, capas alfa vectoriales, inclusión de filtros de terceros, etc. hasta llegar al Photoshop actual.

El Photoshop de hoy es una herramienta tan potente y amplia que se ha convertido en una herramienta totalmente imprescindible para diseñadores gráficos, diseñadores web, ilustradores y dibujantes en general, fotógrafos, editores de video, creadores en 3D, prácticamente cualquier trabajo digital pasa tarde o temprano en alguna de sus fases por Photoshop



2.6. Métrica v3

2.6.1. Introducción a Métrica versión 3

La metodología MÉTRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.

- Facilitar la operación, mantenimiento y uso de los productos software obtenido. La nueva versión de MÉTRICA contempla el desarrollo de Sistemas de Información para las distintas tecnologías que actualmente están conviviendo y los aspectos de gestión que aseguran que un Proyecto cumple sus objetivos en términos de calidad, coste y plazos. Su punto de partida es la versión anterior de MÉTRICA de la cual se han conservado la adaptabilidad, flexibilidad y sencillez, así como la estructura de actividades y tareas, si bien las fases y módulos de MÉTRICA versión 2.1 han dado paso a la división en Procesos, más adecuada a la entrada-transformación-salida que se produce en cada una de las divisiones del ciclo de vida de un proyecto. Para cada tarea se detallan los participantes que intervienen, los productos de entrada y de salida así como las técnicas y prácticas a emplear para su obtención.

En la elaboración de MÉTRICA Versión 3 se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos. También se ha tenido en cuenta la experiencia de los usuarios de las versiones anteriores para solventar los problemas o deficiencias detectadas.

En una única estructura la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad. La automatización de las actividades propuestas en la estructura de MÉTRICA Versión 3, es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo disponibles en el mercado.

2.6.2. Aportaciones de Métrica Versión 3

En lo que se refiere a estándares se ha tenido en cuenta como referencia el Modelo de Ciclo de Vida de Desarrollo propuesto en la norma ISO 12.207 "Information technology – Software life cycle processes". Siguiendo este modelo se ha elaborado la estructura de MÉTRICA Versión 3 en la que se distinguen procesos principales (Planificación, Desarrollo y Mantenimiento) e interfaces (Gestión de Proyectos, Aseguramiento de la Calidad, Seguridad y Gestión de Proyectos) cuyo objetivo es dar soporte al proyecto en los aspectos organizativos.

Además de la norma ISO 12.207, entre los estándares de referencia hay que destacar las normas ISO/IEC TR 15.504/SPICE “Software Process Improvement and Assurance Standards Capability Determination”, UNE-EN-ISO 9001:2000 Sistemas de Gestión de la Calidad.Requisitos, UNE-EN-ISO 9000:2000 Sistemas de Gestión de la Calidad. Fundamentos y Vocabulario y el estándar IEEE 610.12-1.990 “Standard Glossary of Software Engineering Terminology”.

Igualmente se han tenido en cuenta otras metodologías como SSADM, Merise,Information Engineering, MAGERIT. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información promovida por el Consejo Superior de Informática y EUROMÉTODO. Referencias a estas últimas metodologías pueden localizarse en las páginas del CSAE: www.csae.map.es.

Se ha ampliado el enfoque de la Planificación de Sistemas de Información respecto a MÉTRICA Versión 2.1, incluyendo planificación estratégica y recogiendo las actividades de más alto nivel de la fase PSI de MÉTRICA Versión 2.1. Las actividades restantes de la antigua fase PSI se han incorporado al proceso de Desarrollo de la actual versión. Igualmente, aparece el proceso de Mantenimiento de Sistemas de Información que no estaba contemplado en MÉTRICA Versión 2.1.Se ha reforzado el ciclo de vida de las pruebas a través del plan de pruebas y se han mejorado los procedimientos de prueba. Se ha dado respuesta formal a problemáticas específicas de diseño con la incorporación de tecnologías tipo cliente/servidor, interfaces de usuario basadas en entornos gráficos, etc.

Los procesos principales se han enriquecido especificando el contenido, la forma y el momento en que se obtienen los productos, así como la relación entre los productos obtenidos en cada tarea, su reutilización en tareas posteriores y el producto final de cada actividad o proceso. También se han especificado los participantes de forma más precisa en cada tarea, reformando la participación de los usuarios, bien sea usuario final o de sistemas, de forma que dicha participación no se limite a labores informativas frente a las tareas de verificación, aumentando así su responsabilidad. Se ha diferenciado entre la aplicación de Técnicas, como conjunto de heurísticas y procedimientos apoyados en estándares que utilizan notaciones específicas en términos de sintaxis y semántica, y de Prácticas cuya utilización no conlleva reglas preestablecidas con la misma rigidez. Las nuevas técnicas están ampliamente soportadas por herramientas comerciales.

2.6.3. Procesos Principales de Métrica Versión 3

MÉTRICA Versión 3 tiene un enfoque orientado al proceso, ya que la tendencia general en los estándares se encamina en este sentido y por ello, como ya se ha dicho, se ha enmarcado dentro de la norma ISO 12.207, que se centra en la clasificación y definición de los procesos del ciclo de vida del software. Como punto de partida y atendiendo a dicha norma, MÉTRICA Versión 3 cubre el Proceso de Desarrollo y el Proceso de Mantenimiento de Sistemas de Información. MÉTRICA Versión 3 ha sido concebida para abarcar el desarrollo completo de Sistemas de Información sea cual sea su complejidad y magnitud, por lo cual su estructura responde a desarrollos máximos y deberá adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto.

La metodología descompone cada uno de los procesos en actividades, y éstas a su vez en tareas. Para cada tarea se describe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes. El orden asignado a las actividades no debe interpretarse como secuencia en su realización, ya que éstas pueden realizarse en orden diferente a su numeración o bien en paralelo, como se muestra en los gráficos de cada proceso. Sin embargo, no se dará por acabado un proceso hasta no haber finalizado todas las actividades del mismo determinadas al inicio del proyecto. Así los procesos de la estructura principal de MÉTRICA Versión 3 son los siguientes:

- PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN.
- DESARROLLO DE SISTEMAS DE INFORMACIÓN.
- MANTENIMIENTO DE SISTEMAS DE INFORMACIÓN.

El enfoque del Proceso de Planificación de Sistemas de Información, al no estar dentro del ámbito de la norma ISO 12.207 de Procesos del Ciclo de Vida de Software, se ha determinado a partir del estudio de los últimos avances en este campo, la alta competitividad y el cambio a que están sometidas las organizaciones. El entorno de alta competitividad y cambio en el que actualmente se encuentran las organizaciones, hace cada vez más crítico el requerimiento de disponer de los sistemas y las tecnologías de la información con flexibilidad para adaptarse a las nuevas exigencias, con la velocidad que demanda dicho entorno.

La existencia de tecnología de reciente aparición, permite disponer de sistemas que apoyan la toma de decisiones a partir de grandes volúmenes de información procedentes de los sistemas de gestión e integrados en una plataforma corporativa. MÉTRICA Versión 3 ayuda en la planificación de sistemas de información facilitando una visión general necesaria para posibilitar dicha integración y un modelo de información global de la organización. En cuanto al Proceso de Desarrollo de Sistemas de Información, para facilitar la comprensión y dada su amplitud y complejidad se ha subdividido en cinco procesos:

- ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS).
- ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI).
- DISEÑO DEL SISTEMA DE INFORMACIÓN (DSI).
- CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI).
- IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA (IAS).

La necesidad de acortar el ciclo de desarrollo de los sistemas de información ha orientado a muchas organizaciones a la elección de productos software del mercado cuya adaptación a sus requerimientos suponía un esfuerzo bastante inferior al de un desarrollo a medida, por no hablar de los costes de mantenimiento. Esta decisión, que es estratégica en muchas ocasiones para una organización, debe tomarse con las debidas precauciones, y es una realidad que está cambiando el escenario del desarrollo del software. Otra consecuencia de lo anterior es la práctica, cada vez más habitual en las organizaciones, de la contratación de servicios externos en relación con los sistemas y tecnologías de la información y las comunicaciones, llevando a la necesidad de una buena gestión y control de dichos servicios externos y del riesgo implícito en todo ello, para que sus resultados supongan un beneficio para la organización. MÉTRICA Versión 3 facilita la toma de decisión y la realización de todas las tareas que comprende el desarrollo de un sistema de información.

Desde el enfoque de la norma ISO 12.207, el **Proceso de Mantenimiento de Sistemas de Información** comprende actividades y tareas de modificación o retirada de todos los componentes de un sistema de información (hardware, software, software de base, operaciones manuales, redes, etc.). Este marco de actuación no es el objetivo de MÉTRICA Versión 3, ya que esta metodología está dirigida principalmente al proceso de desarrollo del software.

Por lo tanto, MÉTRICA Versión 3 refleja los aspectos del Mantenimiento, correctivo y evolutivo, que tienen relación con el Proceso de Desarrollo.

2.6.4. Planificación de Sistemas de Información

El objetivo de un Plan de Sistemas de Información es proporcionar un marco estratégico de referencia para los Sistemas de Información de un determinado ámbito de la Organización. El resultado del Plan de Sistemas debe, por tanto, orientar las actuaciones en materia de desarrollo de Sistemas de Información con el objetivo básico de apoyar la estrategia corporativa, elaborando una arquitectura de información y un plan de proyectos informáticos para dar apoyo a los objetivos estratégicos.

Por este motivo es necesario un proceso como el de Planificación de Sistemas de Información, en el que participen, por un lado los responsables de los procesos de la organización con una visión estratégica y por otro, los profesionales de SI capaces de enriquecer dicha visión con la aportación de ventajas competitivas por medio de los sistemas y tecnologías de la información y comunicaciones.

Como productos finales de este proceso se obtienen los siguientes, que podrán constituir la entrada para el siguiente proceso de Estudio de Viabilidad del Sistema:

- Catálogo de requisitos de PSI que surge del estudio de la situación actual en el caso de que sea significativo dicho estudio, del diagnóstico que se haya llevado a cabo y de las necesidades de información de los procesos de la organización afectados por el plan de sistemas.
- Arquitectura de información que se compone a su vez de los siguientes productos:
 - Modelo de información.
 - Modelo de sistemas de información.
 - Arquitectura tecnológica.
 - Plan de proyectos.
 - Plan de mantenimiento del PSI.

Un Plan de Sistemas de Información proporcionará un marco de referencia en materia de Sistemas de Información. En ocasiones podrá servir de palanca de cambio para los procesos de la Organización, pero su objetivo estará siempre diferenciado del de un análisis de dichos procesos por sí mismos. Dicho en otras palabras, no se debe confundir el resultado que se persigue con un Plan de Sistemas de Información, con el de una mejora o reingeniería de procesos, ya que los objetivos en ambos casos no son los mismos, aunque el medio para conseguirlos tenga puntos en común (estudio de los procesos y alineamiento con los objetivos estratégicos).

Este nuevo enfoque de alineamiento de los sistemas de información con la estrategia de la organización, la implicación directa de la alta dirección y la propuesta de solución presenta como ventajas:

- La implicación de la alta dirección facilita que se pueda desarrollar con los recursos necesarios y el calendario establecido.
- La perspectiva horizontal de los procesos dentro de la Organización facilita que se atienda a intereses globales y no particulares de unidades organizativas que puedan desvirtuar los objetivos del Plan. Para mantener la visión general que apoye los objetivos estratégicos, el enfoque de un Plan de Sistemas de Información debe orientarse al estudio por procesos.
- La prioridad del desarrollo de los sistemas de información de la organización por objetivos estratégicos.
- La propuesta de Arquitectura de Información que se hace en el plan es más estratégica que tecnológica. El modelo de sistemas de información de la propuesta no es teórico y se contemplan los sistemas de información actuales que se mantendrán.

2.6.5. Desarrollo de Sistemas de Información

El proceso de Desarrollo de MÉTRICA Versión 3 contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema, cubriendo desde el análisis de requisitos hasta la instalación del software. Además de las tareas relativas al análisis, incluye dos partes en el diseño de sistemas: arquitectónico y detallado. También cubre las pruebas unitarias y de integración del sistema, aunque siguiendo la norma ISO 12.207 no propone ninguna técnica específica y destaca la importancia de la evolución de los requisitos. Este proceso es, sin duda, el más importante de los identificados en el ciclo de vida de un sistema y se relaciona con todos los demás.

Las actividades y tareas propuestas por la norma se encuentran más en la línea de un desarrollo clásico, separando datos y procesos, que en la de un enfoque orientado a objetos. En MÉTRICA Versión 3 se han abordado los dos tipos de desarrollo: estructurado y orientado a objeto, por lo que ha sido necesario establecer actividades específicas a realizar en alguno de los procesos cuando se utiliza la tecnología de orientación a objetos. Para este último caso se ha analizado alguna de las propuestas de otras metodologías orientadas a objetos y se han tenido en cuenta la mayoría de las técnicas que contempla UML 1.2 (Unified Modeling Language).

El desarrollo en MÉTRICA Versión 3 lo constituyen los procesos:

- ESTUDIO DE VIABILIDAD DEL SISTEMA (EVS).
- ANÁLISIS DEL SISTEMA DE INFORMACIÓN (ASI).
- DISEÑO DEL SISTEMA DE INFORMACIÓN (DSI).
- CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI).
- IMPLANTACIÓN Y ACEPTACIÓN DEL SISTEMA (IAS).

2.6.5.1. Estudio de Viabilidad del Sistema (EVS)

El propósito de este proceso es analizar un conjunto concreto de necesidades, con la idea de proponer una solución a corto plazo. Los criterios con los que se hace esta propuesta no serán estratégicos sino tácticos y relacionados con aspectos económicos, técnicos, legales y operativos.

Los resultados del Estudio de Viabilidad del Sistema constituirán la base para tomar la decisión de seguir adelante o abandonar. Si se decide seguir adelante pueden surgir uno o varios proyectos que afecten a uno o varios sistemas de información. Dichos sistemas se desarrollarán según el resultado obtenido en el estudio de viabilidad y teniendo en cuenta la cartera de proyectos para la estrategia de implantación del sistema global.

Se ha considerado que este proceso es obligatorio, aunque el nivel de profundidad con el que se lleve a cabo dependerá de cada caso. La conveniencia de la realización del estudio de la situación actual depende del valor añadido previsto para la especificación de requisitos y para el planteamiento de alternativas de solución.

En las alternativas se considerarán soluciones "a medida", soluciones basadas en la adquisición de productos software del mercado o soluciones mixtas. Para valorar las alternativas planteadas y determinar una única solución, se estudiará el impacto en la organización de cada una de ellas, la inversión y los riesgos asociados.

El resultado final de este proceso son los productos relacionados con la solución que se propone para cubrir la necesidad concreta que se planteó en el proceso, y que depende de si la solución conlleva desarrollo a medida o no:

- Contexto del sistema (con la definición de las interfaces en función de la solución).
- Impacto en la organización.
- Coste/beneficio de la solución.
- Valoración de riesgos de la solución.
- Enfoque del plan de trabajo de la solución.
- Planificación de la solución.
- Solución propuesta:
 - Descripción de la solución.
 - Modelo de descomposición en subsistemas.
 - Matriz de procesos/localización geográfica.
 - Matriz datos/localización geográfica. Entorno tecnológico y comunicaciones.
 - Estrategia de implantación global del sistema.
 - Descripción de los procesos manuales.

Si la alternativa incluye desarrollo:

- Modelo abstracto de datos/Modelo de procesos.
- Modelo de negocio/Modelo de dominio.

Si la alternativa incluye un producto software estándar de mercado:

- Descripción del producto.
- Evolución del producto.
- Costes ocasionados por el producto.
- Estándares del producto.
- Descripción de adaptación si es necesaria.

Si en la organización se ha realizado con anterioridad un Plan de Sistemas de Información que afecte al sistema objeto de este estudio, se dispondrá de un conjunto de productos que proporcionarán información a tener en cuenta en todo el proceso.

2.6.5.2. Análisis del Sistema de Información (ASI)

El propósito de este proceso es conseguir la especificación detallada del sistema de información, a través de un catálogo de requisitos y una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema de información y que serán la entrada para el proceso de Diseño del Sistema de Información.

Como ya se ha dicho MÉTRICA Versión 3 cubre tanto desarrollos estructurados como orientados a objetos, y las actividades de ambas aproximaciones están integradas en una estructura común aunque presenta alguna actividad exclusiva para cada tipo de desarrollo.

En primer lugar se describe inicialmente el sistema de información, a partir de los productos generados en el proceso Estudio de Viabilidad del Sistema (EVS). Se delimita su alcance, se genera un catálogo de requisitos generales y se describe el sistema mediante unos modelos iniciales de alto nivel.

Se recogen de forma detallada los requisitos funcionales que el sistema de información debe cubrir, catalogándolos, lo que permite hacer la traza a lo largo de los procesos de desarrollo. Además, se identifican los requisitos no funcionales del sistema, es decir, las facilidades que ha de proporcionar el sistema, y las restricciones a que estará sometido, en cuanto a rendimiento, frecuencia de tratamiento, seguridad, etc.

Para facilitar el análisis del sistema se identifican los subsistemas de análisis, y se elaboran los modelos de Casos de Uso y de Clases, en desarrollos orientados a objetos, y de Datos y Procesos en desarrollos estructurados. Se ha incorporado una actividad específica para la definición de Interfaces de Usuario al tiempo que se van obteniendo y depurando los requisitos y los anteriores modelos. Se especificarán todas las interfaces entre el sistema y el usuario, como formatos de pantallas, diálogos, formatos de informes y formularios de entrada.

Finalizados los modelos, se realiza un análisis de consistencia, mediante una verificación y validación, lo que puede forzar la modificación de algunos de los modelos obtenidos.

Una vez realizado dicho análisis de consistencia se elabora el producto *Especificación de Requisitos Software*, que constituye un punto de referencia en el desarrollo del software y la línea base de referencia para las peticiones de cambio sobre los requisitos inicialmente especificados.

En este proceso se inicia también la especificación del Plan de Pruebas, que se completará en el proceso Diseño del Sistema de Información (DSI).

Los productos resultantes del Análisis del Sistema de Información, dependen del tipo de desarrollo de que se trate y se detallan a continuación especificando los que son distintos, según los dos tipos de desarrollo a los que da respuesta MÉTRICA Versión 3:

- Descripción general del entorno tecnológico.
- Glosario de términos.
- Catálogo de normas.
- Catálogo de requisitos.
- Especificación de interfaz de usuario.

Además, en **Análisis Estructurado**:

- Plan de migración y carga inicial de datos.
- Contexto del sistema.
- Matriz de procesos/localización geográfica.
- Descripción de interfaz con otros sistemas.
- Modelo de procesos.
- Modelo lógico de datos normalizado.

Además, en **Análisis Orientado a Objetos**:

- Descripción de subsistemas de análisis.
- Descripción de interfaces entre subsistemas.
- Modelo de clases de análisis.
- Comportamiento de clases de análisis.
- Análisis de la realización de los casos de uso.

En este proceso es muy importante la participación de los usuarios, a través de técnicas interactivas, como diseño de diálogos y prototipos, que permiten al usuario familiarizarse con el nuevo sistema y colaborar en la construcción y perfeccionamiento del mismo.

2.6.5.3. Diseño del Sistema de Información (DSI)

El propósito del Diseño del Sistema de Información (DSI) es obtener la definición de la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información. A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la especificación técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando proceda.

El diseño de la arquitectura del sistema dependerá en gran medida de las características de la instalación, de modo que se ha de tener en cuenta una participación activa de los responsables de Sistemas y Explotación de las Organizaciones para las que se desarrolla el sistema de información.

Este proceso consta de un primer bloque de actividades, que se realizan en paralelo, y cuyo objetivo es obtener el diseño de detalle del sistema de información que comprende la partición física del sistema de información, independiente de un entorno tecnológico concreto, la organización en subsistemas de diseño, la especificación del entorno tecnológico sobre el que se despliegan dichos subsistemas y la definición de los requisitos de operación, administración del sistema, seguridad y control de acceso.

En el caso de diseño orientado a objetos, conviene señalar que se ha contemplado que el diseño de la persistencia se lleva a cabo sobre bases de datos relacionales.

De este primer bloque de actividades se obtienen los siguientes productos:

- Catálogo de requisitos (se completa).
- Catálogo de excepciones.
- Catálogo de normas para el diseño y construcción.
- Diseño de la arquitectura del sistema.
- Entorno tecnológico del sistema.
- Procedimientos de operación y administración del sistema.
- Procedimientos de seguridad y control de acceso.
- Diseño detallado de los subsistemas de soporte.
- Modelo físico de datos optimizado.
- Asignación de esquemas físicos de datos a nodos.

Además, en ***Diseño Estructurado***:

- Diseño de la arquitectura modular.
- Diseño de interfaz de usuario.

Además, en ***Diseño Orientado a Objetos***:

- Diseño de la realización de casos de uso.
- Modelo de clases de diseño.
- Comportamiento de clases de diseño.
- Diseño de interfaz de usuario.

Al igual que en el proceso de Análisis del Sistema de Información (ASI), antes de proceder a la especificación de los componentes, se realiza una verificación y validación, con objeto de analizar la consistencia entre los distintos modelos y formalizar la aceptación del diseño de la arquitectura del sistema por parte de los usuarios de Explotación y Sistemas.

Un segundo bloque de actividades complementa el diseño del sistema de información, en el que se generan todas las especificaciones necesarias para la construcción del sistema de información:

- Las especificaciones de construcción de los componentes del sistema (módulos o clases, según el caso) y de las estructuras de datos.
- Los procedimientos de migración y sus componentes asociados.
- La definición y revisión del plan de pruebas, y el diseño de las verificaciones de los niveles de prueba establecidos.
- El catálogo de excepciones que permite establecer un conjunto de verificaciones relacionadas con el propio diseño o con la arquitectura del sistema.
- La especificación de los requisitos de implantación.

2.6.5.4. Construcción del Sistema de Información (CSI)

La construcción del Sistema de Información (CSI) tiene como objetivo final la construcción y prueba de los distintos componentes del sistema de información, a partir del conjunto de especificaciones lógicas y físicas del mismo, obtenido en el Proceso de Diseño del Sistema de Información (DSI). Se desarrollan los procedimientos de operación y seguridad y se elaboran los manuales de usuario final y de explotación, estos últimos cuando proceda.

Para conseguir dicho objetivo, se recoge la información relativa al producto del diseño Especificaciones de construcción del sistema de información, se prepara el entorno de construcción, se genera el código de cada uno de los componentes del sistema de información y se van realizando, a medida que se vaya finalizando la construcción, las pruebas unitarias de cada uno de ellos y las de integración entre subsistemas.

Si fuera necesario realizar una migración de datos, es en este proceso donde se lleva a cabo la construcción de los componentes de migración y procedimientos de migración y carga inicial de datos.

Como resultado de dicho proceso se obtiene:

- Resultado de las pruebas unitarias.
- Evaluación del resultado de las pruebas de integración.
- Evaluación del resultado de las pruebas del sistema.
- Producto software:
 - Código fuente de los componentes.
 - Procedimientos de operación y administración del sistema.
 - Procedimientos de seguridad y control de acceso.
 - Manuales de usuario.
 - Especificación de la formación a usuarios finales.
 - Código fuente de los componentes de migración y carga inicial de datos.
 - Procedimientos de migración y carga inicial de datos.
 - Evaluación del resultado de las pruebas de migración y carga inicial de datos.

2.6.5.5. Implantación y Aceptación del Sistema (IAS)

Este proceso tiene como objetivo principal, la entrega y aceptación del sistema en su totalidad, que puede comprender varios sistemas de información desarrollados de manera independiente, según se haya establecido en el proceso de Estudio de Viabilidad del Sistema (EVS), y un segundo objetivo que es llevar a cabo las actividades oportunas para el paso a producción del sistema.

Se establece el plan de implantación, una vez revisada la estrategia de implantación y se detalla el equipo que lo realizará.

Para el inicio de este proceso se toman como punto de partida los componentes del sistema probados de formas unitarias e integradas en el proceso Construcción del Sistema de Información (CSI), así como la documentación asociada. El Sistema se someterá a las Pruebas de Implantación con la participación del usuario de operación cuya responsabilidad, entre otros aspectos, es comprobar el comportamiento del sistema bajo las condiciones más extremas.

También se someterá a las Pruebas de Aceptación cuya ejecución es responsabilidad del usuario final.

En este proceso se elabora el plan de mantenimiento del sistema de forma que el responsable del mantenimiento conozca el sistema antes de que éste pase a producción. También se establece el acuerdo de nivel de servicio requerido una vez que se inicie la producción. El acuerdo de nivel de servicio hace referencia a servicios de gestión de operaciones, de soporte a usuarios y al nivel con el que se prestarán dichos servicios.

Como resultado de este proceso se obtienen los siguientes productos:

- Plan de implantación del sistema en su totalidad.
- Equipo de implantación que realizará la implantación.
- Plan de formación del equipo de implantación (esquema, materiales, recursos necesarios, planificación y especificación de la formación de usuarios finales).
- Evaluación de las pruebas de implantación del sistema por parte del usuario de operación.
- Evaluación de las pruebas de aceptación del sistema por parte del usuario final.
- Plan de mantenimiento previo al paso a producción.
- Acuerdo de nivel de servicio del sistema.
- Sistema en producción.

2.6.6. Mantenimiento de Sistemas de Información

El objetivo de este proceso es la obtención de una nueva versión de un sistema de información desarrollado con MÉTRICA, a partir de las peticiones de mantenimiento que los usuarios realizan con motivo de un problema detectado en el sistema o por la necesidad de una mejora del mismo.

Como consecuencia de esto, sólo se considerarán en MÉTRICA Versión 3 los tipos de Mantenimiento Correctivo y Evolutivo. Se excluyen los tipos de Mantenimiento Adaptativo y Perfectivo, que abarcan actividades tales como la migración y la retirada de software que precisarían el desarrollo de un tipo de metodología específica para resolver su cometido.

Ante una petición de cambio de un sistema de información ya en producción, se realiza un registro de las peticiones, se diagnostica el tipo de mantenimiento y se decide si se le da respuesta o no, en función del plan de mantenimiento asociado al sistema afectado por la petición, y se establece con qué prioridad.

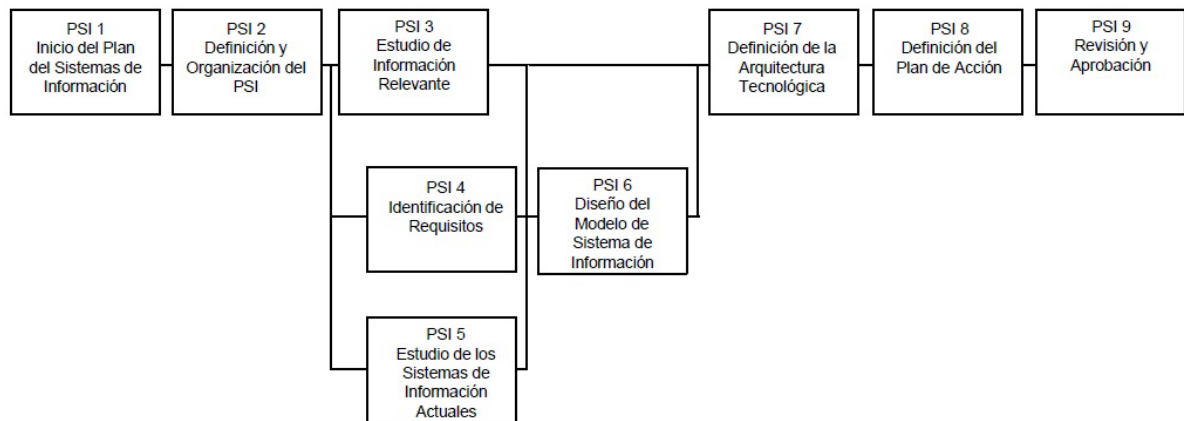
La definición de la solución al problema o necesidad planteada por el usuario que realiza el responsable de mantenimiento, incluye un estudio del impacto, la valoración del esfuerzo y coste, las actividades y tareas del proceso de desarrollo a realizar y el plan de pruebas de regresión.

Los productos que se obtienen en este proceso son los siguientes:

- Catálogo de peticiones de cambio.
- Resultado del estudio de la petición.
- Propuesta de solución.
- Análisis de impacto de los cambios.
- Plan de acción para la modificación.
- Plan de pruebas de regresión.
- Evaluación del cambio.
- Evaluación del resultado de las pruebas de regresión.

3. Planificación del Sistema de Información (PSI)

A continuación se incluye un gráfico que representa la secuencia de actividades del proceso PSI.



3.1. Inicio del Plan de Sistemas de Información

El objetivo de esta actividad es determinar la necesidad del Plan de Sistemas de Información y llevar a cabo el arranque formal del mismo, con el apoyo del nivel más alto de la organización. Como resultado, se obtiene una descripción general del Plan de Sistemas de Información que proporciona una definición inicial del mismo, identificando los objetivos estratégicos a los que apoya, así como el ámbito general de la organización al que afecta, lo que permite implicar a las direcciones de las áreas afectadas por el Plan de Sistemas de Información.

Además, se identifican los factores críticos de éxito y los participantes en el Plan de Sistemas de Información, nombrando a los máximos responsables.

En nuestro caso, estamos ante una empresa de alimentación con sus respectivos departamentos de compra y venta de productos, por lo que el departamento de contabilidad se convierte en una parte imprescindible de la empresa. Recordamos que se trata de una empresa con más de 200 trabajadores y varias sedes lo que obliga a una gran organización contable para una mayor eficiencia.

Como es la empresa la que pone los requisitos, pasaremos directamente a la fase de Desarrollo del Sistema de Información de nuestra aplicación.

4. Desarrollo del Sistema de Información

4.1. Estudio de Viabilidad del Sistema (EVS)



Todo proyecto que se vaya a realizar debe tener previamente un estudio completo de la viabilidad del mismo. En primer lugar se hará el estudio desde el punto de vista económico. Tomamos como base los sueldos medio que cobran un analista y un programador, que sería las personas encargadas de llevar a cabo este tipo de proyecto.

Analista – 2000€

Programador – 1200€

A partir de estos sueldos, calculamos el sueldo medio por hora basándonos en una jornada laboral de 160 horas mensuales.

Analista – 12,5 €/ Hora

Programador – 7,5 €/ Hora

Análisis del coste del proyecto:

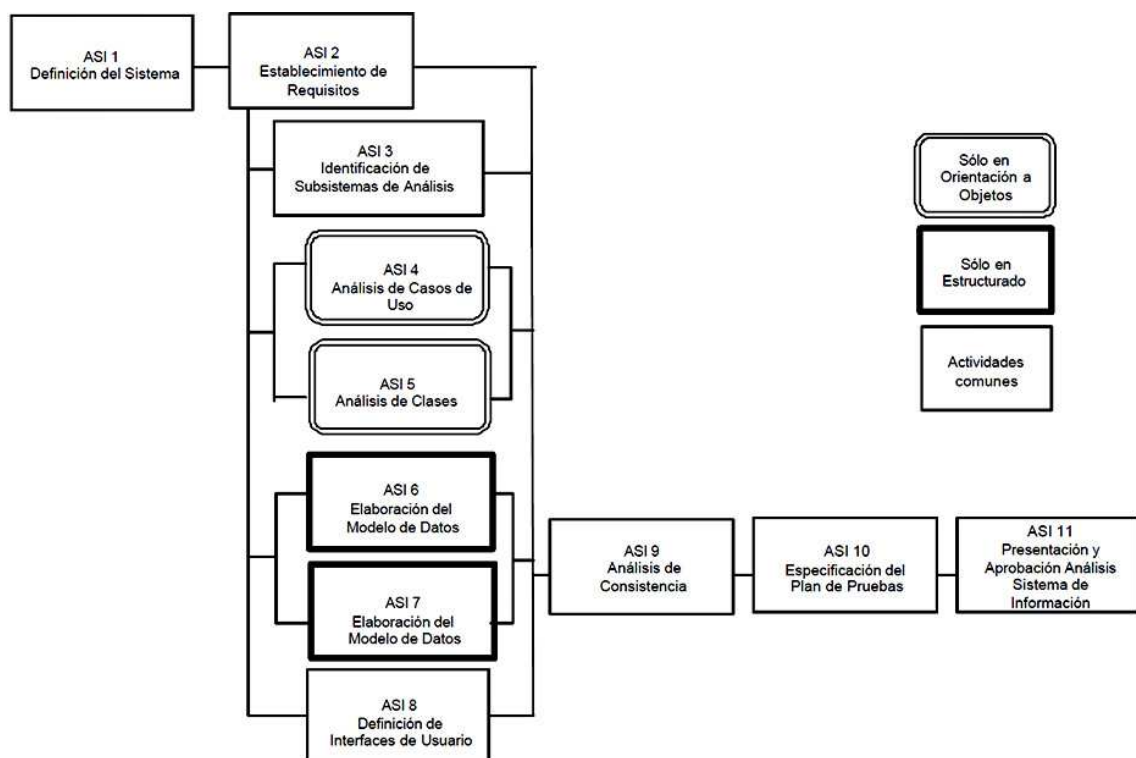
FASE	HORAS	PUESTO	CUANTÍA (€)
Entrevista	20	Analista	250
Análisis Especificaciones	60	Analista	750
Identificación Requisitos	25	Analista	312,5
Casos de Uso	10	Analista	125
Clases de Dominios	10	Analista	125
Fase de Programación	150	Programador	1125
Testeo	20	Programador	150
Documentación	150	Analista	1875
Total	445		4712,5

Tabla resumen de la duración y el coste del proyecto

4.2. Análisis del Sistema de Información (ASI)

En esta parte se establecen los requisitos que se van a necesitar, se hace un esquema de las clases que serán necesarias y la relación que habrá entre ellas. Además, se elabora la estructura que va a tener la base de datos y sus distintas tablas. Y por último, también se elige el interfaz que tendrá el programa así como sus distintas opciones.

Todo esto será una fase preliminar, que puede sufrir cambios a la hora de ser diseñado, porque se encuentren mejores formas de elaborarlo, o por exigencias del cliente.



En nuestro caso, a continuación veremos un esquema de las distintas partes que debemos abordar y, dentro de cada una de ellas, que funciones se deben crear.

En primer lugar tenemos unos esquemas de requisitos que deberá cumplir en el programa. Estos esquemas sirven como guía inicial para saber la distribución que vamos a hacer.

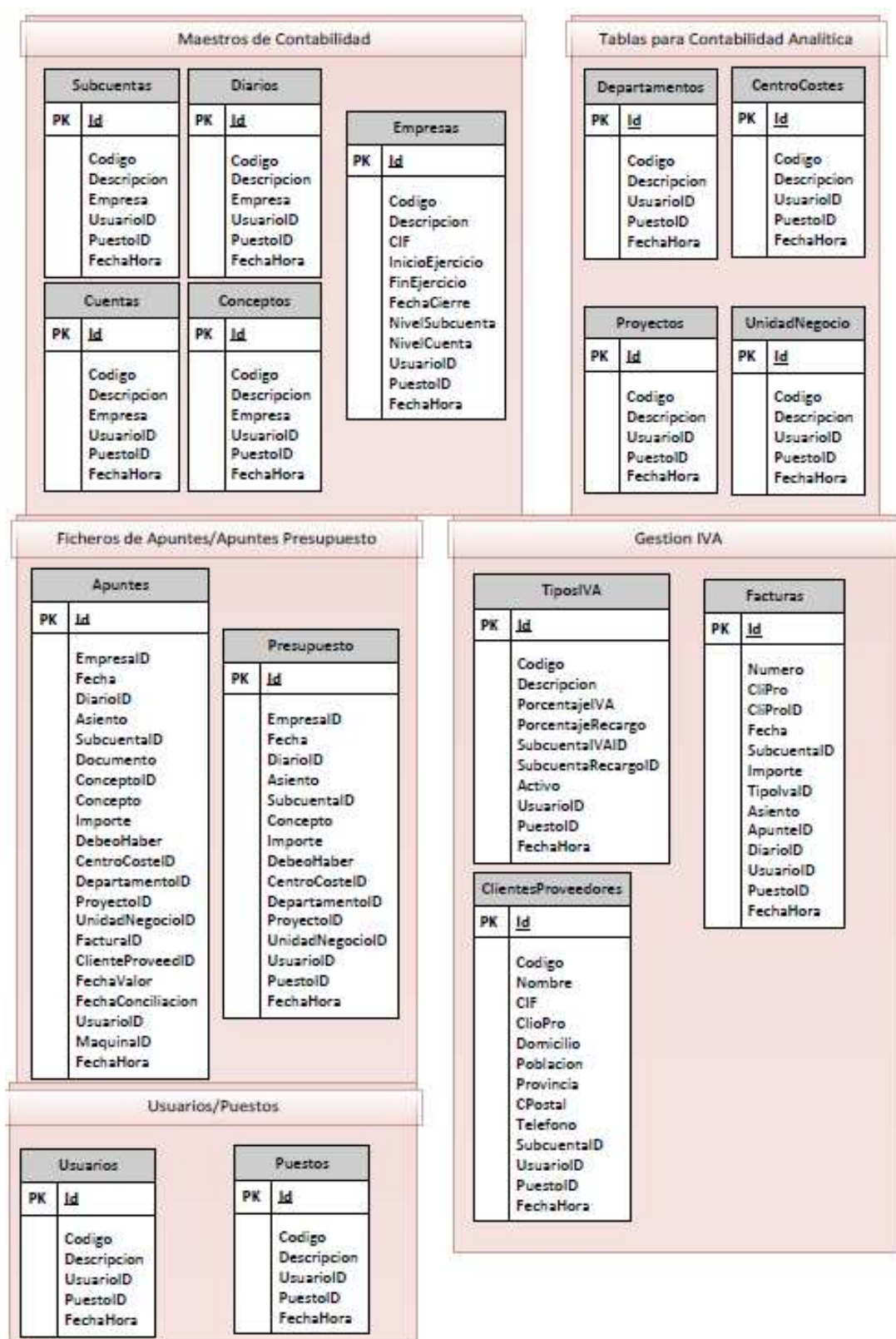
Apuntes y Facturas	<ul style="list-style-type: none">- Entrada de Apuntes.- Edición de Apuntes.- Eliminación de Apuntes.- Entrada de Facturas.- Edición de Facturas.- Eliminación de Facturas.
Maestros de Contabilidad	<ul style="list-style-type: none">- Creación de Cuentas, Subcuentas, Diarios y Conceptos.- Modificación de Cuentas, Subcuentas, Diarios y Conceptos.- Borrado de Cuentas, Subcuentas, Diarios y Conceptos.- Detallado de Cuentas, Subcuentas, Diarios y Conceptos.- Listado de Cuentas, Subcuentas, Diarios y Conceptos.
Contabilidad Analítica	<ul style="list-style-type: none">- Creación de Departamentos, Centros de Coste y Proyectos.- Modificación de Departamentos, Centros de Coste y Proyectos.- Borrado de Departamentos, Centros de Coste y Proyectos.- Detallado de Departamentos, Centros de Coste y Proyectos.- Listado de Departamentos, Centros de Coste y Proyectos.
Balances y Presupuestos	<ul style="list-style-type: none">- Balance de Sumas y Saldos- Balance de Situación- Balance de Pérdidas y Ganancias- Balances Oficiales- Presupuesto de Ventas.- Presupuesto de Inventarios.- Presupuesto de Producción.- Presupuesto de Compras.- Presupuesto de Mano de Obra Directa.- Presupuesto de Gastos de Producción.- Presupuesto de Gastos de Administración y Gastos de Venta.- Presupuesto de Caja.- Estados Financieros Presupuestados.

Cientes	<ul style="list-style-type: none">- Creación de Clientes.- Búsqueda de Clientes.- Modificación de Clientes.- Borrado de Clientes.
Usuarios y Privilegios	<ul style="list-style-type: none">- Creación de Usuarios.- Creación de Privilegios.
Extractos	<ul style="list-style-type: none">- Extractos por Diarios.- Extractos por Subcuentas.- Extractos por Fechas.- Extractos por Centros de Coste.- Extractos por Asientos.- Extractos por Documentos.- Extractos por Conceptos.- Extractos por Cuentas.- Extractos por Departamentos.- Extractos por Centros de Coste.- Extractos por Proyectos.

Por otro lado, se estudió, comparándolo con lo que ya tenían, qué datos iban a ser necesarios guardar. En la siguiente imagen se puede ver un esquema previo de cómo se iban a guardar, y qué datos eran necesarios.

Como decimos, se trata de una versión previa, que luego sufrió modificaciones, ya que entre otras cosas, se optó por eliminar el campo Unidad de Negocio, o se duplicaron las tablas de Maestros de Contabilidad y Contabilidad Analítica, para tener un historial de los códigos.

Además, como se comentará más adelante, se creó la tabla Asiento, para poder meter facturas desde distintos ordenadores a la vez, sin que se repita el mismo código.



4.3. Diseño del Sistema de Información (DSI)

En este apartado procederemos a la programación del código. Para ello crearemos, en primer lugar, una pantalla de inicio donde presentar el programa y donde el usuario podrá logear su cuenta creada previamente. Para poder hacer esta consulta necesitaremos una clase donde hacer consultas a la base de datos, la cual será Main.vb, que explicaremos más adelante.

Una vez llegados a este punto ya podremos dedicarnos a crear un menú principal y cada una de las opciones del mismo. Para nuestro caso, crearemos unos Maestros de Contabilidad, que serán Cuentas, Subcuentas, Diarios y Conceptos, y unos Maestros de Contabilidad Analítica, que son Departamentos, Centros de Coste y Proyectos.

Por otro lado veremos cómo se crean las opciones de añadir apuntes y facturas, así como poder consultarlas, modificarlas o borrarlas. Más adelante, una vez podamos crear facturas, veremos las opciones de Balances y Presupuestos de las mismas.

Ya solo nos quedan dos clases más. Por un lado, poder guardar información de los clientes y por otro, crear usuarios y asignarle privilegios para poder usar este programa.

Explicaremos cada una de las clases en el orden que tendrán posteriormente en el menú principal de nuestro programa.

Principal.vb

Este primer Windows Form, será una ventana de bienvenida, para logear al usuario en el mismo. El objetivo es que ponga su nombre de Usuario y contraseña, comprobarlo en la base de datos de que esté anteriormente añadido, y darle paso al resto del programa, con los privilegios acorde a los que se les haya decidido dar.



Lo primero, consultar Usuario y Contraseña en la Tabla Usuarios. Será un simple “select” y posteriormente, un “if” la consulta es mayor que cero, el número de casos que coincide.

```
comm.CommandText = "select * from Usuarios where Nombre= " & "'" & TextBox1.Text & "' AND Contraseña=" & "'" & TextBox2.Text & "'"
```

```
Dim myReader As DataSet = New DataSet  
datadp.Fill(myReader)
```

```
If myReader.Tables(0).Rows.Count > 0 Then
```

Justo después de esto, guardaremos el usuario y el rango de privilegios que tiene. El usuario lo usaremos en múltiples casos, como iremos viendo, como cada vez que agreguemos algo a la base de datos, para saber quién fue. Y los privilegios como veremos a continuación, para controlar que puede hacer y que no, un usuario en concreto.

```
usuario = myReader.Tables(0).Rows(0)("Nombre")  
priv = myReader.Tables(0).Rows(0)("Privilegios")
```

Aquí vemos como sacamos de la tabla Privilegios cada uno de ellos, y los guardaremos en distintas variables. Para más información sobre esto, véase Menuprv.vb, donde se detalla más en profundidad.

```
TPrivilegio = myReader2.Tables(0).Rows(0)("Privilegio")  
TPrivAñAp = myReader2.Tables(0).Rows(0)("PrivAñAp")  
TPrivConAp = myReader2.Tables(0).Rows(0)("PrivConAp")  
TPrivBalances = myReader2.Tables(0).Rows(0)("PrivBalances")  
...
```

Main.vb

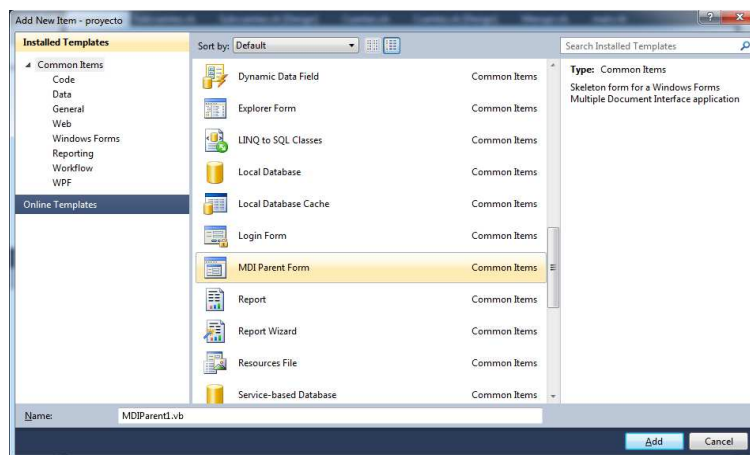
Se trata de una clase que usaremos para abrir una nueva conexión a la base de datos, cada vez que queramos consultar información de la misma. Debemos recordar que la conexión entre el programa y la base de datos no está ligada siempre, ya que supondría un tráfico de datos enorme, sino que cada vez que se lanza una consulta, se crea una nueva conexión, se recoge los datos que se busquen, se devuelven y se vuelve a cerrar la conexión. Como se trata de un código muy repetido en cada opción del programa, hemos optado por crearlo en una clase externa y así poder hacer una llamada al código, cada vez que sea necesario.

```
Public Sub main()  
    cnx.ConnectionString = "Data Source=CHRISTIAN;Initial  
    Catalog=proyecto;Integrated Security=True"  
    cnx.Open()  
    comm.Connection = cnx  
    datadp = New SqlClient.SqlDataAdapter(comm)  
End Sub
```

Menupr.vb

En este caso, vamos a crear el menú principal del programa. Se ha decidido por la opción de tener un menú permanente en el lado izquierdo y una zona de trabajo en el lado derecho, donde abrir tantas pestañas como se quiera. Había muchas posibilidades de diseño en este caso. Pero al tratarse de un programa hecho a medida para una empresa, cumpliremos los requisitos que nos pongan. Una alternativa hubiera sido la zona derecha hacerla mediante pestañas, teniendo en cuenta que Visual Studio no trae la librería necesaria de serie, pero si añadiéndole una, concretamente **IE WebControls**.

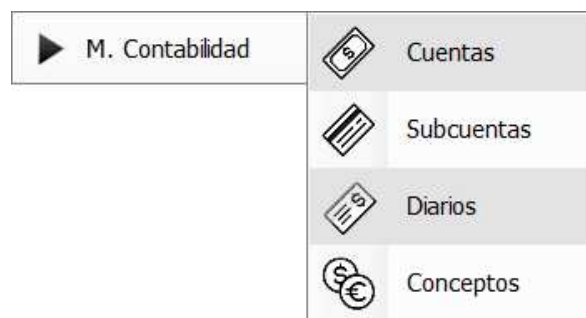
Para crear este diseño que hemos comentado usaremos un MDI Parent Form. Se trata de un diseño de padres e hijos, en la que la ventana principal y la zona del menú serán el padre, y cada ventana que abramos en el lado derecho serán los hijos.



Para la zona del menú, hemos creado un botón por cada opción que queramos usar posteriormente. Además hemos incluido iconos característicos que relacionen cada opción con su función.



Dentro de la tercera opción, los Maestros de Contabilidad, hemos agrupado cuatro opciones similares, para dejar un menú lo más claro y sencillo posible. Estas cuatro opciones son Cuentas, Subcuentas, Diarios y Conceptos. Simplemente al pulsar el botón se desplegará y permitirá abrir en nuestra zona de trabajo alguna de las cuatro opciones o incluso las cuatro.



Una vez visto el diseño visual del menú, vamos a profundizar en aquellos aspectos del código a destacar. Lo primero que cabe destacar es que no todas las opciones estarán disponibles a todos los usuarios. Como ya veremos en Usuarios.vb crearemos distintos privilegios, por lo que debemos conseguir que en el menú se desactiven las opciones que no nos interesen mostrar. Para ello, en la pantalla anterior, Principal.vb, como ya vimos, guardábamos esos privilegios para ahora hacer uso de ellos. Haremos un simple If, y según sea True o False, habilitaremos o deshabilitaremos las opciones.

```
Private Sub Menu_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    If Principal.TPrivAñAp = "False" Then
        ApuntesFacturasToolStripMenuItem.Enabled = False
    End If

    If Principal.TPrivConAp = "False" Then
        ConsultasToolStripMenuItem.Enabled = False
    End If

    ...

End Sub
```

Por otro lado, para llamar a los “hijos”, como dijimos anteriormente que llamamos a las ventanas que vayamos abriendo, simplemente deberemos hacer un Show() y colocarlo donde queramos. Sería algo así:

```
Private Sub CuentasToolStripMenuItem1_Click(sender As System.Object, e As
System.EventArgs) Handles CuentasToolStripMenuItem1.Click
    Cuentas.MdiParent = Me
    Cuentas.Show()
    Cuentas.BringToFront()
End Sub
```

Ya solo nos queda por mencionar la opción de salir, la cual haremos fácilmente con el comando Close(), indicando que esto será cuando se pulse el último botón del menú.

```
Private Sub SalirToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SalirToolStripMenuItem.Click
    Close()
End Sub
```

Apufac.vb

Este **Windows Form** será el que nos permita introducir nuevos apuntes y facturas en la base de datos. Para ello crearemos una ventana dotándola de una serie de menús desplegables, y campos para rellenar, así una casilla marcapable. También llevará una tabla, como ahora veremos, en la que mostrar los datos que vayamos introduciendo. En primer lugar vamos a ver cómo quedaría visualmente esta ventana y qué necesidades tendremos.

Apufac

Apuntes / Facturas

Fecha : 07/06/2012

Nº Documento : 123456

Asiento : 000000

Debe / Haber : ☐

☐ Factura

ClienteID :

SubcuentaID : 50009800

Diario :

Concepto :

Departamento : 111

Centro de Coste : 111

Proyecto : 100

Cantidad :

TipoIVA :

Total :

Usuario :

Como podemos ver, los campos que hemos puesto son, en primer lugar un **DateTimePicker**, para añadir la fecha del apunte/factura. Después viene el número de documento, en el cual lo único que controlamos es que sea numérico, mediante el comando `IsNumeric`. El campo Asiento es un valor que sale automáticamente como el siguiente número al último valor de asiento existente. Este lo conseguimos de la siguiente manera:

```
Dim Nuevo As String
Nuevo = Asiento.Text + 1
main()

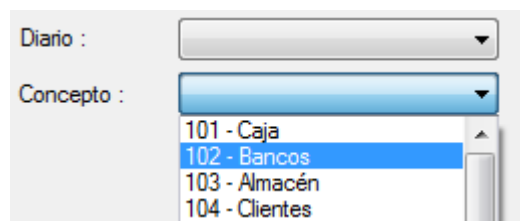
comm.CommandText = "update Asiento set Asiento=" & "'" & Nuevo & "' where
Asiento=" & "'" & Asiento.Text & "'"

comm.ExecuteNonQuery()
```

El campo Debe / Haber, nos permite escribir una D mayúscula o una H mayúscula también según necesitemos. Simplemente, mediante un `if` controlamos que esto ocurra.

```
If Not (Debehaber.Text = "D") And Not (Debehaber.Text = "H") Then
    MessageBox.Show("Código incorrecto. Solo se acepta 'H' o 'D'")
Else
```

Para el código de Subcuenta solo tendremos que controlar que ese código exista en la base de datos, ya que las condiciones se controlarán allí como veremos posteriormente. Luego tenemos dos **ComboBox**, que son menús desplegables para elegir entre distintas opciones. Las hemos configurado de modo que no se puedan escribir en ellos y que no pueda ser nulo, sino que siempre se tenga que rellenar con algún valor. Vemos una imagen de cómo quedaría el menú desplegado con algunos valores:



Los tres siguientes campos, a petición del cliente no son obligatorios y simplemente controlaremos que existan en la base de datos. En sus respectivas clases, se controlarán las peticiones del cliente en cuando a longitud y forma del código. Una vez llegados a este punto, los dos últimos campos son los que diferencian a los apuntes de las facturas. Por ello pondremos un **CheckBox** para marcar cuando sea una factura. Mientras tanto los dos campos, tanto cliente como IVA, estarán inaccesibles. Este evento lo controlaremos fácilmente del siguiente modo:

```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
        TextBox4.Enabled = True
        ComboBox3.Enabled = True
    Else
        TextBox4.Enabled = False
        ComboBox3.Enabled = False
    End If
End Sub
```

Cuando estén accesibles serán dos campos normales, como los anteriores que hemos descrito. Para completar todo esto, pondremos dos botones. El primero, el botón **Añadir**, se trata de agregar los datos en una tabla intermedia antes de pasarlos a nuestra base de datos. Para ello, controlaremos una serie de condiciones, como son que el código de subcuenta, centro de coste, departamento y proyectos existan. Algo parecido a esto:


```
'Subcuenta
comm.CommandText = "select * from TablaSubcuenta where CodigoSubcuenta= "
& "'" & Subcuenta.Text & "'"
Dim myReader As DataSet = New DataSet
datadp.Fill(myReader)

If myReader.Tables(0).Rows.Count = 0 Then
    MessageBox.Show("Código de Subcuenta no existente.")
Else
```

Además controlaremos los campos obligatorios que no sean nulos y que la cantidad no sea nula. Una vez hecho esto, usaremos introducimos en la tabla Temporal los datos. Estos mismos pasos serán posibles tantas veces como se quiera.

Consultas.vb

Para crear este formulario, nos basaremos en una consulta grande a la base de datos, acotada por una serie de condiciones.

Consultas

Fecha Inicio : 09/01/2012
Fecha Fin : 09/12/2012

SubcuentaID : 47200012
Diario : C1 - Diario de cobros
Concepto : 102 - Bancos

Departamento : MAR - Marketing
Centro de Coste : 100 - Málaga
Proyecto : 112 - Navidad 2012

Mostrar Modificar Borrar

Asiento	Fecha	NumDocumento	Subcuenta	DH	Total	Diario	Concepto	Departamento	Centrocoste	Proyecto
1040	22/10/2012	64000001	47200012	D	23,30	C1	102	MAR	100	112
1041	22/10/2012	64000002	47200012	D	23,30	C1	102	MAR	100	112
1042	22/10/2012	64000002	47200012	D	23,30	C1	102	MAR	100	112

Usuario : CHRISTIAN

Crearemos una ventana como la anterior donde cumplir todos los datos exigidos. Para ello, en primer lugar rellenaremos los cinco ComboBox con lo que ya tenemos almacenado. Además lo haremos de la forma que salga “Código – Descripción” para facilitar el trabajo al usuario. Esto lo conseguiremos de la siguiente manera:

```
Dim selectedRowCount As Integer = myReader.Tables(0).Rows.Count

ComboBox2.Items.Add("")

For i = 0 To selectedRowCount - 1
    Diarios = myReader.Tables(0).Rows(i)("CodigoDiario") + " - " +
    myReader.Tables(0).Rows(i)("DescripcionDiario")
    ComboBox2.Items.Add(Diarios)
Next i
```

La fila de Add(""), la usamos para añadir un campo vacío, por si después de indicar uno, queremos volver a dejarlo en blanco. Una vez hecho esto, haremos que al pulsar el botón **mostrar**, realice la consulta, añadiendo un AND por cada combo box señalizado.

```
cadena = "select Asiento, Fecha, NumDocumento, Subcuenta, DH, Total,
Diario, Concepto, Departamento, Centrocoste, Proyecto, cliente, IVA from
TablaFacturas where Fecha >= " & "'" & Fecha1.Value & "' AND Fecha <= "
& "'" & Fecha2.Value & "'"

If Not (ComboBox2.Text = String.Empty) Then
    temporal=" AND Diario = " & "'" & ComboBox2.Text.Substring(0, 2)""
    nueva = cadena + temporal
    cadena = nueva
End If

...
cadena = cadena + "order by Asiento"
```

Para **borrar**, lanzaremos un mensaje indicando el número de filas señaladas, así como el asiento de cada uno de ellos para que se confirme la eliminación. Aquí lo más llamativo es como capturar cada Asiento de cada fila seleccionada:

```
sb.Append(DataGridView1.SelectedRows(i).Cells(0).Value.ToString())
```

Para terminar, en caso de querer **modificar** un apunte / factura, abriremos una nueva ventana, que explicaremos su construcción, a continuación, en ApufacM.vb.

ApufacM.vb

Sabemos que queremos poder modificar apuntes o facturas que ya hayamos agregado previamente. La clave está en que en la clase anterior señalábamos una fila para modificar, entonces ahora, debemos mostrar una ventana con esos datos, con la posibilidad de modificarlos. Para ello crearemos los mismos campos que teníamos para crearlos y cargaremos la información que ya tengamos de ese asiento anteriormente.

Esto lo conseguiremos, primero cargando del siguiente modo, todos los posibles valores que tendrán los campos Diario y Concepto:

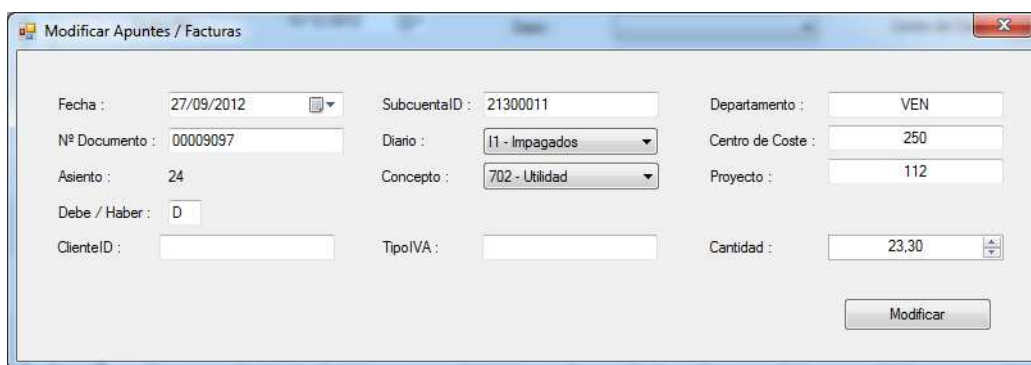
```
Dim selectedRowCount As Integer = myReader.Tables(0).Rows.Count

For i = 0 To selectedRowCount - 1
    Diarios = myReader.Tables(0).Rows(i)("CodigoDiario") + " - " +
    myReader.Tables(0).Rows(i)("DescripcionDiario")
    Combodiario.Items.Add(Diarios)
Next i
```

Tras esto copiaremos la información campo a campo, recordando que el campo asiento no podrá ser modificado.

```
...
Subcuenta.Text = myReader3.Tables(0).Rows(0)("Subcuenta")

If Not (myReader3.Tables(0).Rows(0)("Departamento") = String.Empty) Then
    Departamento.Text = myReader3.Tables(0).Rows(0)("Departamento")
End If
...
```



Una última cosa que hemos añadido y que no mencionamos a la hora de crear las facturas fue obligar al campo DebeHaber, a que siempre que se escriba una D o una H, salga automáticamente en mayúsculas, para ahorrar al usuario tiempo y posibles errores.

Esto se consigue usando el procedimiento TextChanged, para que cada vez que haya un campo en ese campo, revise si se trata de una de esas dos letras.

```
Private Sub Debehaber_TextChanged(sender As System.Object, e As
System.EventArgs) Handles Debehaber.TextChanged
    If (Debehaber.Text = "h") Then
        Debehaber.Text = "H"
    ElseIf (Debehaber.Text = "d") Then
        Debehaber.Text = "D"
    End If
End Sub
```

Cuentas.vb

Se divide en siete partes, las cuales son Agregar, Borrar, Detallar, Listar, Modificar, Load e Imprimir.

Código	Descripción
1	FINANCIACIÓN BÁSICA
10	CAPITAL
100	Capital social
101	Fondo social
102	Capital
103	Socios por desembolsos no exigidos
1030	Socios por desembolsos no exigidos, capital social
1034	Socios por desembolsos no exigidos, capital pendiente de inscripción
104	Socios por aportaciones no dinerarias pendientes

En la parte de **Agregar** pondremos una serie de condiciones a cumplir y la parte principal será la de insertar mostrando un mensaje previamente. Esto lo conseguiremos de la siguiente manera:

```
If MsgBox("¿ Insertar: " + TextBox1.Text + " - " + TextBox2.Text + "?",  
vbOKCancel, "Confirmación") = vbOK Then  
  
comm.CommandText = "insert into TablaCuenta values ('" + TextBox1.Text +  
"', '" + TextBox2.Text + "')"   
  
comm.ExecuteNonQuery()
```

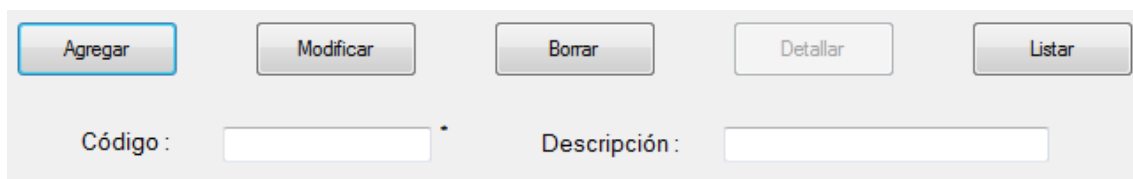
En **Borrar**, aparte de usar la instrucción Delete, tendremos que hacer un insert para guardar en la tabla TablaCuentaCambio, que se ha producido un cambio:

```
comm.CommandText = "insert into TablaCuentaCambio values ('" +  
TextBox1.Text + "', '" + usuario + "', '" + Fecha + "', 'Borrado')"  
comm.ExecuteNonQuery()  
Menupr.Estado.Text = "Borrado:Codigo(" + TextBox1.Text + ")"
```

Las funciones **Listar** y **Detallar** son de parecida implementación, aunque mostrarán datos diferentes. Hay que recordar que la opción detallar solo está disponible para unos usuarios en concreto, lo cual lo hemos conseguido de la siguiente manera:

```
If Principal.TPrivMoCo = "False" Then
    Borrar.Enabled = False
    Modificar.Enabled = False
End If
If Principal.TPrivDeCo = "False" Then
    Detallar.Enabled = False
End If
```

Se ve que se trata de un sencillo código de Enabled, con True o False según nos interese. Aquí una foto con el resultado final:



Una cosa que hemos tenido en cuenta es, a la hora de mostrar los datos, que se muestren de la manera más organizada y clara posible. Para ello, rellenaremos la columna que más nos interesa para completar la tabla. Estos cambios también surtirán efecto a la hora de imprimirse.

	Código	Descripción
▶	1	FINANCIACIÓN BÁSICA
	10	CAPITAL
	100	Capital social
	101	Fondo social
	102	Capital
	103	Socios por desembolsos no exigidos
	1030	Socios por desembolsos no exigidos, capital social
	1034	Socios por desembolsos no exigidos, capital pendiente de inscripción
	104	Socios por aportaciones no dinerarias pendientes

En la foto anterior vemos como la columna Descripción queda alargada más allá de la descripción de mayor longitud, quedando uniforme a la tabla. Para conseguir esto en Visual Studio, tan solo tendremos que añadir al comando **AutoSizeMode** del DataGridView, la forma que queremos que tenga. En este caso será de relleno, por lo que usaremos la palabra Fill, indicando en número de columna a rellenar. Hay que recordar que se comienza a contar por la columna cero, por lo que en el anterior ejemplo será la columna uno, y en el que veremos a continuación, la columna CuentaCambio es la cuatro.

```
DataGridView1.Columns(4).AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill
```

Y a continuación, el ejemplo con varias columnas:

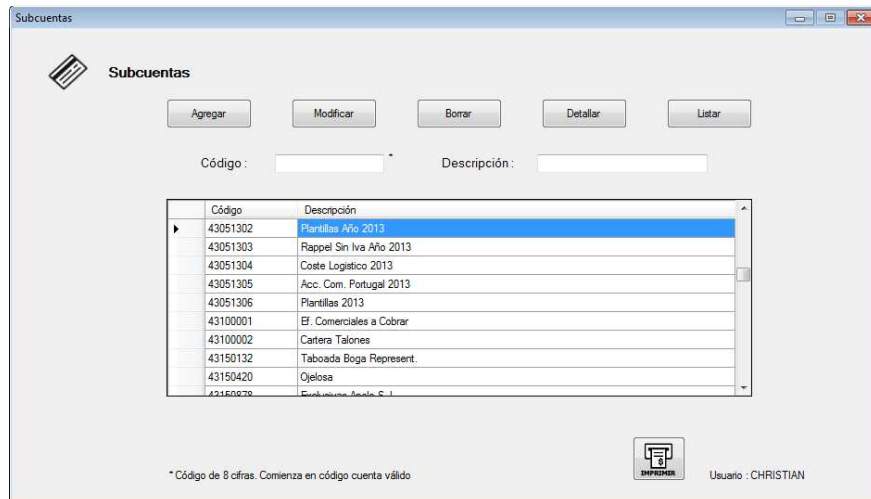
	CodigoCuenta	DescripcionCuenta	UsuarioCuenta	FechaCuenta	CuentaCambio
►	10		CHRISTIAN	15/10/2012	Borrado
	10	Nueva Cuenta	CHRISTIAN	15/10/2012	Insertado
	10	Nueva Cuenta 2	CHRISTIAN	15/10/2012	Modificado
	10		CHRISTIAN	15/10/2012	Borrado
	10	CAPITAL	CHRISTIAN	15/10/2012	Insertado

Y solo nos queda tratar los apartados de Load e Imprimir. En el primero tan solo nos preocuparemos de que según los privilegios que tenga el usuario, pueda acceder a unas opciones u otras, como ya vimos antes el caso de Detallar, y añadiremos que realice la opción de Listar nada más abrir la pestaña. Y el segundo apartado, como se trata de una opción común en varias pestañas, será tratado individualmente más adelante.

```
If Principal.TPrivAñCo = "False" Then  
    Agregar.Enabled = False  
    Listar.Enabled = False  
Else  
    Listar_Click(Nothing, Nothing)  
End If  
If Principal.TPrivMoCo = "False" Then  
    Borrar.Enabled = False  
    Modificar.Enabled = False  
End If  
If Principal.TPrivDeCo = "False" Then  
    Detallar.Enabled = False  
End If
```

Subcuenta.vb

En este caso nos encontramos un código similar al anterior de Cuentas.vb, teniendo en cuenta la diferencia de código entre uno y otro que pasamos a detallar.



Mientras en cuentas nos encontrábamos un código numérico de hasta cuatro cifras, ahora tenemos un código de ocho cifras fijas. Además, el principio del código debe ser un código de cuenta existente anteriormente, rellenado de ceros hasta las cuatro cifras. Con esto queremos decir que si el código de cuenta es un número de 2 cifras, el nuevo código subcuenta serán esas dos cifras, más dos ceros para rellenar esas cuatro cifras y más otras cuatro cifras que indique el código completo de ocho cifras. Ejemplo:

Código de Cuenta: 213

Código de Subcuenta: 213 (código cuenta existente)

0 (relleno de cuatro cifras con ceros)

5685 (cuatro cifras que completan una subcuenta)

21305685 (nuevo código de subcuenta)

Ahora tenemos que ver cómo controlar, tanto que exista el código de cuentas, como que no exista el de subcuentas y sobre todo, que tenga un formato válido. Para ello como vemos en el siguiente trozo de código, dividiremos la cadena en subcadenas, con el comando `Substring(x,y)`, el cual indica con “x” donde comienza una subcadena, y con “y” cuantos caracteres debe cortar. Una vez comprobado un caso, usaremos un `while` para seguir controlando, hasta encontrar una cifra que no sea cero por detrás.

```
Dim aux As String
Dim aux2 As String
Dim copia As String
copia = TextBox1.Text.Substring(0, 4)
aux = copia.Substring(0, 3)
aux2 = copia.Substring(3, 1)
While (copia.Length > 1) And (aux2 = 0)
    copia = aux
    aux = copia.Substring(0, copia.Length - 1)
    aux2 = copia.Substring(copia.Length - 1, 1)
End While
```

Diarios.vb

En esta pantalla tendremos que conseguir que se puedan hacer los movimientos básicos que requieren los diarios, es decir, poder agregar nuevos, borrarlos, modificarlos y listarlos. Además añadiremos una opción de administración mediante la cual, los usuarios de mayor rango puedan ver todos estos movimientos citados anteriormente. Para ello esta Windows Form quedará con el siguiente aspecto:

Código	Descripción
C1	Diario de cobros
C2	Adquisición intrac bienes
I1	Impagados
I2	Inversion sujeto pasivo
I3	Ingreso talones
N1	Nomina
P1	Diario de pagos
P2	Provisiones
R1	Diario de recibos

Analicemos parte por parte como hemos conseguido incluirle las condiciones requeridas por el cliente, así como que realice las tareas que debe hacer. Empezaremos por la parte principal, que será la de añadir. En ella controlaremos todas las condiciones, lo que facilitará al modificar o borrar ya que tendremos garantizado que los datos de la base de datos cumplen las condiciones.

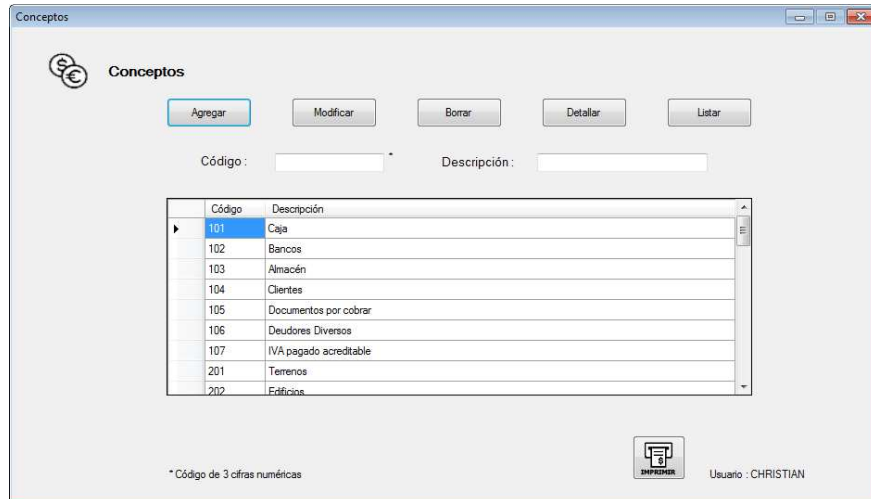

```
If TextBox1.Text.Length = 2 Then
    Dim aux As String
    Dim aux2 As String
    Dim copia As String
    copia = TextBox1.Text
    aux = copia.Substring(0, 1)
    aux2 = copia.Substring(1, 1)
    If aux >= "A" And aux <= "Z" And IsNumeric(aux2) Then
```

Los códigos de diario son un código formado por una letra y un número. En ese trozo de código vemos como controlamos por separado que se cumplen ambas cosas. Gracias a las funciones que traen de serie este programa, Visual Studio, podemos hacer uso de ellos y hacerlo realmente sencillo usando el comando Substring para separar la cadena y comparar cada parte con letras y números. El resto son tareas sencillas, como son comprobar que no sea nulo, mediante el comando empty o asegurar que no exista ya en la base de datos, sacando los datos que ya existan y viendo si el número de filas buscadas son cero.

```
comm.CommandText = "select * from TablaDiario where CodigoDiario= " & ""
& TextBox1.Text & ""
```

```
If myReader.Tables(0).Rows.Count = 0 Then
```

Conceptos.vb



La creación de esta ventana es muy similar a las anteriores. Veremos los pequeños cambios que la diferencian. En primer lugar, como ya sabemos se trata de un código de tres cifras numéricas y una descripción ligada a él. Para controlar esto, como ya hemos hecho en anteriores casos, usaremos los comandos "length" para la longitud y "IsNumeric" para que sean numéricas.

```
If TextBox1.Text.Length = 3 Then
    If IsNumeric(TextBox1.Text) Then
```

Una vez hecho, lo importante será en primer lugar asegurarse que no exista previamente, luego lanzar un mensaje de confirmación. Para los mensajes de confirmación usamos el comando “vbOKCancel”. Con esto automáticamente en el mensaje de confirmación tendremos los botones **Aceptar** y **Cancelar**.

```
If MsgBox("¿ Insertar: " + TextBox1.Text + " - " + TextBox2.Text + "?",
vbOKCancel, "Confirmación") = vbOK Then
comm.CommandText = "insert into TablaConcepto values ('" + TextBox1.Text +
"', '" + TextBox2.Text + "'"")
comm.ExecuteNonQuery()
```

```
comm.CommandText = "insert into TablaConceptoCambio values ('" +
TextBox1.Text + "', '" + TextBox2.Text + "', '" + usuario + "', '" + Fecha +
"', 'Insertado')"
comm.ExecuteNonQuery()
```

Departamentos.vb

Veamos las diferencias principales entre la elaboración de este Windows Form y los demás maestros de contabilidad. Su diferencia radica en que en este caso se trata de un código de tres letras mayúsculas, con lo cual deberemos controlar que no se agrega ningún otro código no válido. Para ello lo hemos hecho de la siguiente manera:

```
Temp = TextBox1.Text
iLen = Temp.Length
If iLen > 0 Then
    For iCtr = 1 To iLen
        Character = Mid(Temp, iCtr, 1)
        If Not Character Like "[A-Z]" Then
            Valido = False
        End If
    Next
    If Valido Then
        . . .
    Else
        MessageBox.Show("Código no correcto.")
    End If
End If
```

Como se puede ver, se trata de un bucle desde uno hasta la longitud de la cadena, que previamente hemos controlado que es de tres, comprobando que esta entre la letra A y la letra Z. Por tanto, de camino controlamos que solo incluyan mayúsculas. Algo que se podría añadir es que además de A-Z permitiera otros símbolos como, por ejemplo, la letra Ñ. Para esto solo hubiéramos tenido que añadir una condición “Or” y añadir más letras o símbolos permitidos.

	Código	Descripción
►	COM	Compras
	CON	Contabilidad
	MAR	Marketing
	VDI	Ventas Dinamarca
	VEN	Ventas
	VES	Ventas España
	VFR	Ventas Francia
	VNO	Ventas Noruega

Para las demás opciones de modificar, borrar o detallar, haremos un bucle parecido, haciendo las consultas igual que las hicimos en las anteriores ventanas.

CentrosdeCoste.vb

No profundizaremos demasiado en Centros de Coste y en Proyectos ya que se tratan de situaciones similares a las que ya vimos en Concepto.vb. Necesitamos una ventana para crear códigos de tres cifras numéricas, y luego poder trabajar con ellas.

```
If MsgBox("; Insertar: " + TextBox1.Text + " - " + TextBox2.Text + "?",  
vbOKCancel) = vbOK Then  
    comm.CommandText = "insert into TablaCoste values ('" +  
        TextBox1.Text + "','" + TextBox2.Text + "'"")"  
    comm.ExecuteNonQuery()  
  
    cnx.Close()  
    main()  
  
    comm.CommandText = "insert into TablaCosteCambio values ('" +  
        TextBox1.Text + "','" + TextBox2.Text + "','" + usuario + "','" +  
        Fecha + "','Insertado')"  
    comm.ExecuteNonQuery()  
  
    cnx.Close()  
    Menupr.Estado.Text = "Insertado: Código(" + TextBox1.Text + ")"  
    Listar_Click(Nothing, Nothing)  
End If
```

Primero mandamos un mensaje de confirmación, y si es positivo, insertamos en las dos tablas, la de códigos y la de cambios. Modificamos nuestro campo de Estado a "Insertado: Código ()" y actualizamos la lista de códigos.

Proyectos.vb

Al igual que en el anterior, crearemos las cinco funciones, Agregar, Modificar, Borrar, Detallar y Listar, para códigos numéricos de tres cifras. (Véase Concepto.vb)



Balances.vb

En esta ventana obtendremos el saldo o diferencia entre debe / haber. Para ello necesitamos cinco columnas. Tendremos la primera de código, dividido en 4 niveles distintos igual que antes, la segunda con la descripción, que variará de la tabla cuenta o subcuentas según el nivel. Luego una columna con las sumas de los debe, otra con las sumas de los haber y otra con la diferencia total entre ambas. Es fácil entender que la complejidad de esta ventana radica en cómo hacer la consulta. Veamos el resultado final:

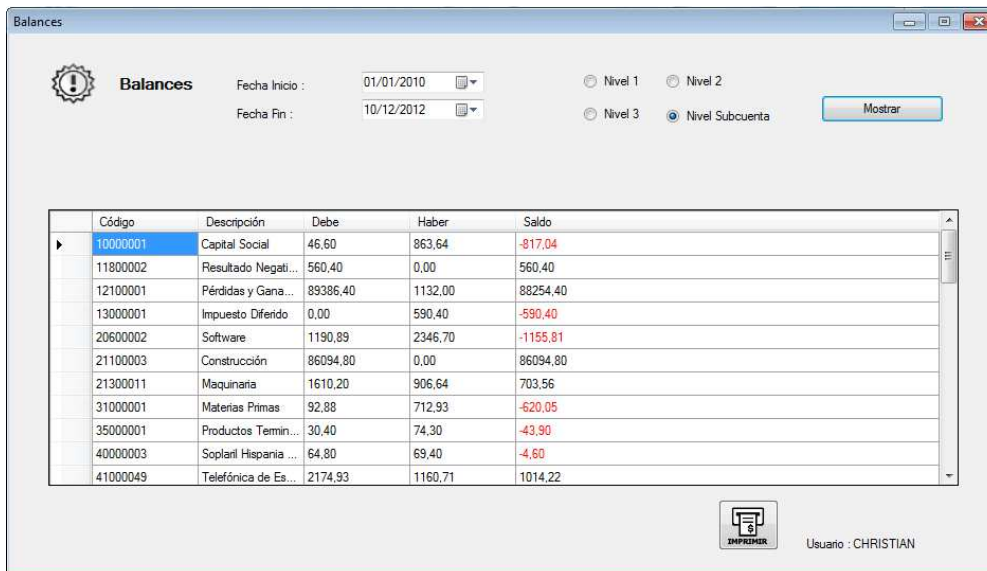
```
If (Nivel1.Checked = True) Then

    main()
    comm.CommandText = "select round(subcuenta/10000000,0) Codigo,
sum(case when DH='D' then total else 0 end) as debe, sum(case when
DH='H' then total else 0 end) as haber,sum(case when DH='D' then
total else total*-1 end) as suma from TablaFacturas where Fecha >=
" & "'" & Fecha1.Value & "' AND Fecha <= " & "'" & Fecha2.Value &
"' group by round(subcuenta/10000000,0)"

    Dim myReader As SqlClient.SqlDataReader
    myReader = comm.ExecuteReader()
    Dim DT As New DataTable
    DT.Load(myReader)
    DataGridView1.DataSource = DT
    DataGridView1.Columns(3).AutoSizeMode =
        DataGridViewAutoSizeColumnMode.Fill
    cnx.Close()
EndIf
```

Este es el caso del primer nivel, que variará de los otros en la cifra por la que divide, en este caso es 10000000, y será de una cifra menos cada vez, hasta llegar al 4º nivel que directamente no dividirá. Después usamos el comando case para diferenciar lo que es un Debe y un Haber.

Este comando, es una instrucción de decisión múltiple, donde el compilador prueba o busca el valor contenido en una variable contra una lista de constantes, cuando el computador encuentra el valor de igualdad entre variable y constante, entonces ejecuta el grupo de instrucciones asociados a dicha constante, si no encuentra el valor de igualdad entre variable y constante, entonces ejecuta un grupo de instrucciones asociados a un default, aunque este último es opcional.



The screenshot shows a window titled 'Balances' with a search icon and the title 'Balances'. It includes date selection fields for 'Fecha Inicio' (01/01/2010) and 'Fecha Fin' (10/12/2012). There are radio buttons for 'Nivel 1', 'Nivel 2', 'Nivel 3', and 'Nivel Subcuenta' (selected). A 'Mostrar' button is present. Below is a table with columns: Código, Descripción, Debe, Haber, and Saldo. The table lists various accounts with their respective debit, credit, and balance values. At the bottom right, there is a printer icon and the text 'Usuario : CHRISTIAN'.

	Código	Descripción	Debe	Haber	Saldo
▶	10000001	Capital Social	46,60	863,64	-817,04
	11800002	Resultado Negati...	560,40	0,00	560,40
	12100001	Pérdidas y Gana...	89386,40	1132,00	88254,40
	13000001	Impuesto Diferido	0,00	590,40	-590,40
	20600002	Software	1190,89	2346,70	-1155,81
	21100003	Construcción	86094,80	0,00	86094,80
	21300011	Maquinaria	1610,20	906,64	703,56
	31000001	Materias Primas	92,88	712,93	-620,05
	35000001	Productos Termin...	30,40	74,30	-43,90
	40000003	Soplantl Hispania ...	64,80	69,40	-4,60
	41000049	Telefónica de Es...	2174,93	1160,71	1014,22

Además nos quedan dos detalles más en la consulta. Primero un acotarlo a las fechas que se indique, teniendo en cuenta que el día marcado será a las 00.00, con lo cual si por ejemplo quisiéramos consultar solo los balances de facturas realizadas en el día de hoy, tendríamos que indicar como fecha de inicio, el mismo día de hoy y como fecha final, el día de mañana.

Lo último a reseñar será realizar el group by que comentábamos, los 4 distintos niveles requeridos que se realizarán de la misma manera que hicimos para las distintas columnas anteriormente.

Presupuestos.vb

A continuación veremos cómo hemos construido esta nueva opción de presupuestos. La idea es agrupar de 4 maneras distintas los apuntes/facturas sumando las cantidades totales, independientemente si son debe o haber y mostrar lo que presupuestamos que gastaremos o ingresaremos por ello. Hay que saber que los presupuestos van una vez por mes y por código de subcuenta, con lo cual debemos controlar dos cosas.

4. Desarrollo del Sistema de Información

Código	Descripción	Debe	Haber	Saldo	DebePre	HaberPre	SaldoPre	Diferencia
10000001	Capital Social	46,60	863,64	-817,04	50,00	605,00	-555,00	-262,04
11800002	Resultado Negati...	560,40	0,00	560,40	151,80	0,00	151,80	408,60
12100001	Pérdidas y Gana...	89386,40	1132,00	88254,40	25050,50	33989,00	-8938,50	97192,90
13000001	Impuesto Diferido	0,00	590,40	-590,40	0,00	80,00	-80,00	-510,40
20600002	Software	1190,89	2346,70	-1155,81				-1155,81
21100001	Construcciones				375,00	0,00	375,00	-375,00
21100003	Construcción	86094,80	0,00	86094,80	250,00	250,00	0,00	86094,80
21300005	Maquinarias				250,00	250,00	0,00	0,00
21300011	Maquinaria	1610,20	906,64	703,56	250,00	250,00	0,00	703,56
28130003	Amort. Maquinaria				560,00	780,00	-220,00	220,00
31000001	Materias Primas	92,88	712,93	-620,05	1000,00	2300,00	-1300,00	679,95

Por un lado, que el usuario solo pueda elegir meses/años y no días, y por otro que al meterlo nosotros en la base de datos, siempre metamos el mismo día del mes, por comodidad en nuestro caso, el día 1 de cada mes.

```
Fecha3.Format = DateTimePickerFormat.Custom
Fecha3.CustomFormat = "MM/yyyy"
```

```
Fecha3.Value = New Date (Fecha3.Value.Year, Fecha3.Value.Month, 1)
```

Otro detalle será de los debes y haberes, que al ser campos numéricos (Numericupdown) tendremos que computar el valor que irán escritos con comas, y cambiarlo por puntos, que es el formato que admite nuestra base de datos.

```
Debe.Value.ToString.Replace(",", ".")
```

Vamos con la parte más compleja en cuanto a SQL se refiere. Para mayor claridad lo hemos dividido en varias partes, las cuales iremos desmembrando paso a paso. Lo primero es crearnos una idea mental de que es lo que queremos conseguir.

Código	Descripción	Debe	Haber	Saldo	DebePre	HaberPre	SaldoPre	Diferencia						
Común	TablaSubcuentas	-----	TablaFacturas	-----	+	-	-----	TablaPresupuestos	-----	+	-	-----	+	-

Tenemos que conseguir unir nueve campos de tres tablas distintas. La unión de estas tablas lo conseguimos mediante el campo Código de subcuenta. Para explicarlo lo más detalladamente, lo mejor es empezar por el final e ir volviendo hasta el principio poco a poco.

En la parte 5 vemos como unimos con LEFT JOIN la TablaSubcuenta. El motivo de usar LEFT JOIN es para sólo quedarnos con la descripción de las que necesitemos y no de todas. En la parte 4, vemos cómo hacemos la consulta de presupuestos. Esta parte es sencilla. Se trata de agrupar los presupuestos por código, que estén dentro del intervalo de fechas pedido, sacando una suma de los debes, otra de los haberes y la diferencia entre ambas.

En la parte 3, hacemos algo parecido a lo que vimos en balances. Sacamos de la Tabla Facturas un sumatorio de los debes y otro de los haberes usando el comando case. Además, sacamos también una columna de diferencia entre ambas, todo ello sin olvidar el intervalo de fechas y agruparlo todo mediante group by.

Las partes 3 y 4, como podemos ver, se agrupan mediante FULL JOIN porque necesitamos sacar todo lo que hemos sacado, tanto de una parte como de la otra, ya que habrá casos de facturas sin presupuestos y de presupuestos sin facturas.

Una vez hecha la unión de la tercera y cuarta parte, en la segunda vemos cómo hacer el select de todos los datos que necesitamos. Una de las cosas fundamentales y que comentamos antes, es la de casos de facturas sin presupuestos y viceversa. Nuestra última columna es la diferencia de las dos diferencias o saldos. Esta columna no podría hacerse si alguno de los dos lados son nulos, es por ello que usamos la cláusula ISNULL.

Ya sólo nos queda la primera parte, la más sencilla, que es mostrar cada una de las nueve columnas.

```
comm.CommandText = "SELECT T1.Código Código, T2.DescripcionSubcuenta
(PARTE 1) Descripción, T1.Debe, T1.Haber, T1.Saldo, T1.DebePre, T1.HaberPre,
T1.SaldoPre, T1.Diferencia from

(select ISNULL(t1.Codigo,t2.CodigoPre) Código, t1.Debe, t1.Haber,
(PARTE 2) t1.Saldo, t2.DebePre, t2.HaberPre, t2.SaldoPre, ISNULL(Saldo,0) +
ISNULL((SaldoPre*-1),0) Diferencia from

(select subcuenta Codigo, sum(case when DH='D' then total else 0 end) as
Debe, sum(case when DH='H' then total else 0 end) as Haber,sum(case when
(PARTE 3) DH='D' then total else total*-1 end) as Saldo from TablaFacturas where
Fecha >= " & "'" & Fecha1.Value & "'" AND Fecha <= " & "'" & Fecha2.Value
& "'" group by subcuenta) t1

FULL JOIN

(select CodigoPre, sum(DebePre) DebePre, sum(HaberPre) HaberPre,
(PARTE 4) sum(DebePre+(HaberPre*-1)) SaldoPre from TablaPresupuestos where
FechaPre >= " & "'" & Fecha1.Value & "'" AND FechaPre <= " & "'" &
Fecha2.Value & "'" group by CodigoPre) t2 on t2.CodigoPre=t1.Codigo) T1

LEFT JOIN

(PARTE 5) TablaSubcuenta T2 ON T2.CodigoSubcuenta=T1.Código"
```

Importante ahora saber que, como ya dijimos, presupuestos tiene cuatro niveles. Hemos visto el último, el de subcuenta. Para los demás será bastante similar, modificando dos cosas. La primera es "round (subcuenta/100000,0)" en cada una de las veces que consultemos los códigos de subcuenta. El número de ceros irá en función del nivel. Y la otra es que la descripción ya no será de subcuenta, sino de cuentas, con lo cual, en la parte 5, cambiaremos de tabla de consulta.

Y para terminar la parte de Presupuestos, queremos darle color a los números negativos, en especial los de la columna diferencia. Es bastante sencillo y lo hacemos con la siguiente condición:

```
If Row.Cells("Diferencia").Value < 0 Then  
    DataGridView1.Item (8, Row.Index).Style.ForeColor = Color.Red  
End If
```

Cientes.vb

Codigo	Descripción	Nif	Dirección	Postal	Población	País
0001	Venay España	10000000	C/Hernest Hemin...	29004	Málaga	España
0002	Carrefour	28425280	P.I.Los Olivos c/ ...	28906	Madrid	España

Aquí trabajaremos con todo lo relacionado con clientes. Al igual que en otras opciones, podremos crear, modificar, buscar y borrar clientes. En primer lugar, veamos cómo crearlos. Hay que tener en cuenta que tanto el Código como el Nif son únicos y ligados, por lo que deberemos controlar ambos de la siguiente forma:

```
If Not (Codigo.Text = String.Empty) And (Codigo.Text.Length = 4) And  
IsNumeric(Codigo.Text) Then
```

```
comm.CommandText = "select * from TablaClientes where Codigo= " &  
"" & Codigo.Text & ""
```

```
If Not (Nif.Text = String.Empty) And (Nif.Text.Length < 20) And  
Valida(Nif.Text) Then
```

```
comm.CommandText = "select * from TablaClientes where Nif= "  
& "" & Nif.Text & ""
```


Tras esto controlaremos cada uno de los campos, por un lado que no estén vacíos y por el otro la longitud que hemos programado para cada uno de ellos en la base de datos. En casos excepcionales algunos de ellos deberán ser solo numéricos, lo cual es sencillo mediante IsNumeric.

```
If Not (Descripcion.Text = String.Empty) And (Descripcion.Text.Length < 255) Then
If Not (Direccion.Text = String.Empty) And (Direccion.Text.Length < 255) Then
If Not (Postal.Text = String.Empty) And (Postal.Text.Length < 10)
And IsNumeric(Postal.Text) Then
If Not (Poblacion.Text = String.Empty) And (Poblacion.Text.Length < 30) Then
If Not (Pais.Text = String.Empty) And (Pais.Text.Length < 30) Then
If Not (Telefono1.Text = String.Empty) And (Telefono1.Text.Length < 30) And
IsNumeric(Telefono1.Text) Then
If Not (Telefono2.Text = String.Empty) And (Telefono2.Text.Length < 30) And
IsNumeric(Telefono2.Text) Then
If Not (Correo.Text = String.Empty) And (Correo.Text.Length < 255) Then
If Not (Web.Text = String.Empty) And (Web.Text.Length < 255) Then
```

En la opción de rellenar, debemos controlar dos campos. El de código y el de Nif. Si en alguno de ellos hay escrito un código, debemos buscar ese código y rellenar cada uno de los campos que tenemos en la parte superior. Primero controlaremos campo código y luego en Nif. Por lo tanto, si escribimos algo en el campo código obviaremos el de Nif y viceversa. Y si escribimos en ambas, solo buscaremos por códigos. En caso de que no exista, lanzaremos mensaje de error y no haremos nada más. Aquí se puede ver como copiamos los campos para mostrarlos en la pantalla.

```
If myReader.Tables(0).Rows.Count > 0 Then

    Nif.Text = myReader.Tables(0).Rows(0)("Nif")
    Descripcion.Text = myReader.Tables(0).Rows(0)("Descripcion")
    Direccion.Text = myReader.Tables(0).Rows(0)("Direccion")
    Postal.Text = myReader.Tables(0).Rows(0)("Postal")
    ...
Else
    MessageBox.Show("Código no encontrado.")
End If
```

La opción de modificar se realizara mediante el código, ya que aunque pide que se rellene ambos campos, será con ese con el que buscará en la base de datos, lo anteriormente introducido, y modificará todos los campos por los nuevos. Para modificar datos se usan las cláusulas Update y Set. Además le agregaremos un mensaje previo de confirmación para evitar problemas.

```
If MsgBox("¿ ModificarCodigo: " + Codigo.Text + " con Nif: " + Nif.Text +
"?", vbOKCancel, "Confirmación") = vbOK Then

    comm.CommandText = "update TablaClientes set Descripcion=" & "'" &
Descripcion.Text & "', Direccion=" & "'" & Direccion.Text & "',
Postal=" & "'" & Postal.Text & "',Poblacion=" & "'" &
Poblacion.Text & "'",
    ...
    where codigo=" & "'" & Codigo.Text & "'"

    comm.ExecuteNonQuery()
    Menupr.Estado.Text = "Insertado"
    Listar_Click(Nothing, Nothing)
End If
```

Por último ver una pequeña función que hemos creado y utilizado en varios puntos de este Windows Form. Esta función Válida (String), controla que una cadena de caracteres tenga en cada uno de sus caracteres o un número, o una letra, tanto mayúscula como minúscula. La usaremos para controlar el Nif.

```
Public Function Valida (TestString As String) As Boolean

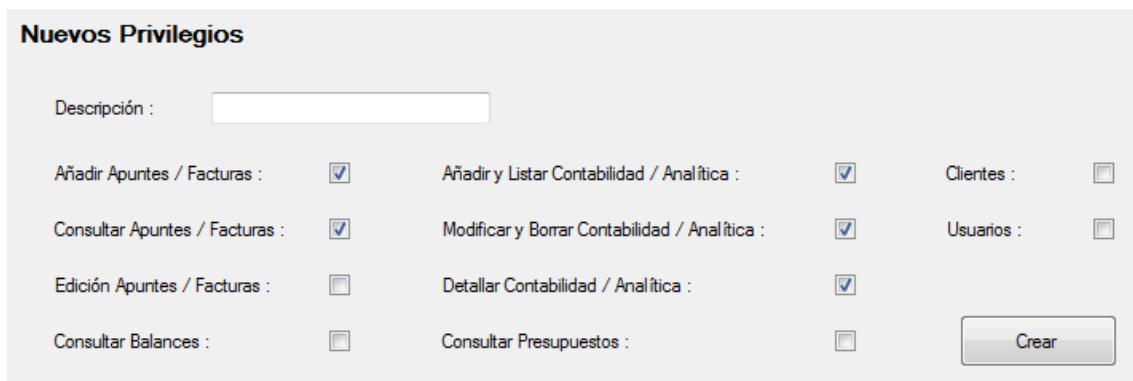
    Dim sTemp As String
    Dim iLen As Integer
    Dim iCtr As Integer
    Dim sChar As String
    Valida = False

    sTemp = TestString
    iLen = Len(sTemp)
    If iLen > 0 Then
        For iCtr = 1 To iLen
            sChar = Mid(sTemp, iCtr, 1)
            If Not sChar Like "[0-9A-Za-z]" Then Exit Function
        Next
        Valida = True
    End If

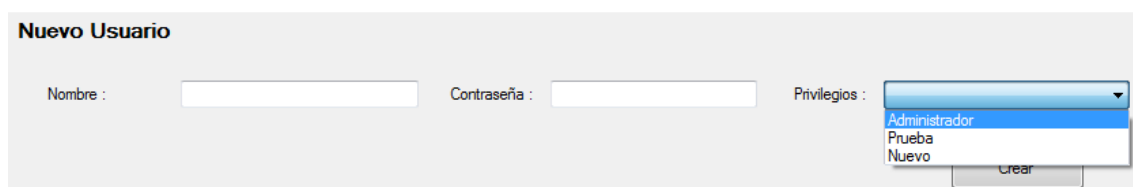
End Function
```

Usuarios.vb

La ventana de usuarios tiene dos partes principales. La de arriba para crear usuarios y la de abajo para crear tipos de privilegios para esos usuarios. Veamos primero la parte inferior. En ella vemos una serie de CheckBox para marcar las opciones que queremos darle y un campo Descripción para darle un nombre. No hemos puesto ningún tipo de requisito entre unos CheckBox y otros y será quien los cree, él que tome esa decisión.



Ahora en la parte superior, tenemos 3 campos. Nombre, Contraseña y Privilegios. En Nombre y Contraseña pondremos sus respectivas condiciones, que son, en nombre, la longitud y en contraseña, tipo numérica.



Para los privilegios usaremos un ComboBox, desplegando todos los privilegios creados anteriormente. Ya vimos cómo se crean distintos tipos de privilegios. Para cargarlos en el desplegable, lo haremos del siguiente modo:

```
For Each row As DataRow In DT.Rows
    ComboBox1.Items.Add(row("Privilegio"))
Next
```

Al seleccionar un tipo de privilegio, programaremos que los indique en la parte inferior, marcándolos o desmarcándolos, según estén previamente creados.

```
If myReader2.Tables(0).Rows.Count > 0 Then

    MPrivilegio = myReader2.Tables(0).Rows(0)("Privilegio")
    MPrivAñAp = myReader2.Tables(0).Rows(0)("PrivAñAp")
    ...

    If MPrivAñAp = True Then
        CheckBox1.Checked = True
    Else
        CheckBox1.Checked = False
    End If
    ...
End If
```

4.4. Construcción del Sistema de Información (CSI)

Una vez terminada la parte de diseño, en la que programamos todo el código, necesitamos una base de datos donde guardar toda la información. Ya vimos en el apartado 4.2 un esquema previo de las necesidades que íbamos a tener, y tras el diseño ya tenemos una idea definitiva de las tablas que harán falta.

Además comentaremos los distintos tipos de valores, y para que usemos cada uno, así como la diferencia entre varios de ellos.

Las tablas de **Maestros de Contabilidad** son ocho, dos por cada una de las opciones. Como ya vimos anteriormente, las cuatro partes de esta contabilidad son Cuentas, Subcuentas, Diarios y Conceptos. Para cada una de ellas tenemos dos tablas, la primera para guardar el código y la descripción, y la segunda a modo de historial guardando además el usuario, la fecha y que tipo de cambio sufrió ese código. En esta segunda tabla podrán estar repetidos los códigos, cosa que en el primero no ocurrirán. Usaremos nvarchar y date para estas tablas.

TablaCuenta CodigoCuenta DescripcionCuenta	TablaCuentaCambio CodigoCuenta DescripcionCuenta UsuarioCuenta FechaCuenta CuentaCambio	TablaSubcuenta CodigoSubcuenta DescripcionSubcuenta	TablaSubcuentaCambio CodigoSubcuenta DescripcionSubcuenta UsuarioSubcuenta FechaSubcuenta SubcuentaCambio
TablaDiario CodigoDiario DescripcionDiario	TablaDiarioCambio CodigoDiario DescripcionDiario UsuarioDiario FechaDiario DiarioCambio	TablaConcepto CodigoConcepto DescripcionConcepto	TablaConceptoCambio CodigoConcepto DescripcionConcepto UsuarioConcepto FechaConcepto ConceptoCambio

Una vez visto las tablas de Maestros de Contabilidad, ahora veremos las relativas a **Contabilidad Analítica**, las cuales tienen un funcionamiento similar. Tendremos dos tablas por cada función del programa. Una para guardar códigos y sus descripciones, y la otra para cambios en esta. Como ya vimos anteriormente, la Contabilidad Analítica engloba a Departamentos, Centros de Coste y Proyectos.

TablaCoste CodigoCoste DescripcionCoste	TablaCosteCambio CodigoCoste DescripcionCoste UsuarioCoste FechaCoste CosteCambio	TablaDepartamento CodigoDepartamento DescripcionDepartamento	TablaDepartamentoCambio CodigoDepartamento DescripcionDepartamento UsuarioDepartamento FechaDepartamento DepartamentoCambio
TablaProyecto CodigoProyecto DescripcionProyecto	TablaProyectoCambio CodigoProyecto DescripcionProyecto UsuarioProyecto FechaProyecto ProyectoCambio		

Una vez guardados los distintos tipos de códigos, ahora veremos que tablas necesitaremos para guardar nuestros apuntes / facturas. Como ya vimos en el apartado anterior de diseño del programa, las facturas se guardarán en dos partes, en primer lugar en una tabla **Temporal**, y una vez hayamos agregado tantas facturas como queramos, podremos confirmar y agregarlas a nuestra tabla **Facturas**. Estas dos tablas tendrán los mismos campos. Un campo Asiento, de tipo Integer, donde guardaremos el código único de la factura.

Este código lo tendremos guardado en la tabla **Asiento**, e irá aumentando cada vez que abramos una nueva. Esto se podría haber hecho de otra manera, y es cogiendo el código de la última factura que haya, y sumándole una, pero de este modo, tendríamos el problema de que como el programa se usará en varios ordenadores a la vez, varios tendrían en mismo número, y causaría problema. Es por ello que hemos decidido que la mejor opción es una tabla donde cada ordenador consulte el asiento actual.

Siguiendo con las tablas anteriores, tendremos el campo Fecha, para guardar la fecha de la factura, de tipo date. Después de esto, tendremos una serie de campo, todos ellos de tipo nvarchar, pero cada uno con la longitud adecuada. Serán el número del Documento, si se trata de Debe o Haber, y los campos de Maestros de Contabilidad y Contabilidad Analítica. Además si se trata de una factura, y no de un apunte, guardaremos los campos Cliente e Iva.

A continuación tendremos el campo Total, donde va la cantidad de la factura, por tanto será de tipo Decimal, con dos decimales. Y por último, guardaremos el usuario que creo la factura, para futuras consultas.

TablaFacturas	Temporal	Asiento
Asiento	Asiento	Asiento
Fecha	Fecha	
NumDocumento	NumDocumento	
DH	DH	
Subcuenta	Subcuenta	
Diario	Diario	
Concepto	Concepto	
Departamento	Departamento	
Centrocoste	Centrocoste	
Proyecto	Proyecto	
Cliente	Cliente	
IVA	IVA	
Total	Total	
Usuario	Usuario	

Las dos siguientes tablas están relacionadas con la clase Usuarios, y será donde guardaremos por un lado los usuarios, y por otros los tipos de privilegios existentes. En la tabla **Usuarios** solo guardaremos tres valores. El nombre y la contraseña, y el nombre del tipo de privilegios.

En la tabla **Privilegio**, guardaremos, en primer lugar, el nombre del tipo de privilegios. A partir de ahí, tendremos diez campos, donde guardaremos True o False, según lo queramos activo o no.

Los distintos privilegios son:

- **PrivAñAp:** Gracias a esto podremos acceder a la primera pestaña del Menú Principal.
- **PrivConAp:** En este caso, nos dará acceso a la siguiente pestaña, en la que podremos consultar, modificar y borrar Apuntes y/o Facturas introducidas previamente.
- **PrivBalances:** Permite ver los Balances dentro del intervalo de fecha que indiquemos y por niveles.
- **PrivPresupuestos:** Permite ver los Balances, los Presupuestos y la diferencia entre ambos dentro del intervalo de fecha que indiquemos y por niveles.
- **PrivAñCo:** Permite añadir nuevos códigos en los Maestros de Contabilidad (Cuentas, Subcuentas, Diarios y Conceptos) y en la Contabilidad Analítica (Departamentos, Centros de Coste y Proyectos). Además podremos listar los códigos que haya actualmente en la Base de Datos.
- **PrivMoCo:** Permite modificar y borrar códigos en los Maestros de Contabilidad (Cuentas, Subcuentas, Diarios y Conceptos) y en la Contabilidad Analítica (Departamentos, Centros de Coste y Proyectos).
- **PrivDeCo:** Permite consultar el historial de cualquier código de Maestros de Contabilidad y en la Contabilidad Analítica.
- **PrivClientes:** Crear, modificar y borrar Clientes.

- **PrivEdAp:** Permite dentro de la ventana de Consultas, borrar Facturas. Además, también estará activa la opción Modificar para abrir una nueva ventana donde cambiarlas.
- **PrivUsuarios:** Acceso a esta misma pantalla para crear nuevos Usuarios y Privilegios

Usuarios	TablaPrivilegio
Nombre	Privilegio
Contraseña	PrivAñAp
Privilegios	PrivConAp
	PrivBalances
	PrivPresupuestos
	PrivAñCo
	PrivMoCo
	PrivDeCo
	PrivClientes
	PrivGraficos
	PrivUsuarios

La Tabla **Clientes** será como una agenda, donde guardaremos información de distintos Clientes. Lo primero será asignarle un Código y el Nif, ambos campos únicos que diferencian unos Clientes de otros. Además tendremos una serie de campos de tipo nvarchar donde guardar toda la información de un Cliente, la Descripción, Dirección, Código Postal, Teléfonos, etc. Además, al igual que en otras tablas, añadiremos Usuario y Fecha de cuando se creó.

TablaClientes
Codigo
Descripcion
Nif
Direccion
Postal
Poblacion
Pais
Telefono1
Telefono2
Correo
Web
Usuario
Fecha

Nos queda la parte de Balances y Presupuestos. Para Balances no necesitaremos ninguna tabla, ya que haremos las consultas directamente de la tabla Facturas, pero para los Presupuestos si tendremos que guardar unos valores. Los Presupuestos se guardan una vez por mes y por código, por eso, para facilitar guardarlos, cambiaremos el tipo dd/mm/aaaa por 01/mm/aaaa. Como ya hemos dicho, en la tabla Presupuestos, guardamos un código y una fecha, además de dos cantidades de tipo decimal (18,2). Estas serán el debe y el haber. Y por último, usuario y fecha de cambio, como en otras tablas.

TablaPresupuestos	
CodigoPre	
FechaPre	
DebePre	
HaberPre	
UsuarioPre	
FechaCambioPre	

Tipos de datos

- **Integer**

Contiene enteros de 32 bits con signo (4 bytes) que se sitúan en el intervalo entre -2.147.483.648 y 2.147.483.647.

- **Decimal**

Contiene valores de 128 bits con signo (16 bytes) que representan números enteros de 96 bits (12 bytes) que se ajustan a una potencia variable de 10. El factor de ajuste especifica el número de dígitos a la derecha del separador decimal; comprendidos entre 0 y 28. Con 28 posiciones decimales, el valor más grande es +/-7.9228162514264337593543950335 y el valor más pequeño distinto de cero es +/-0.000000000000000000000000000001 (+/-1E-28).

- **Date**

Contiene valores IEEE de 64 bits (8 bytes) que representan fechas comprendidas entre el 1 de enero del año 0001 y el 31 de diciembre de 9999, y horas comprendidas entre las 0:00:00 (medianoche) y las 11:59:59. Cada incremento representa 100 nanosegundos de tiempo transcurrido desde el comienzo del 1 de enero del año 1 del calendario gregoriano. El valor máximo representa 100 nanosegundos antes del comienzo del 1 de enero del año 10000.

- **nvarchar [(n | max)]**

Datos de cadena Unicode de longitud variable. n define la longitud de la cadena y puede ser un valor entre 1 y 4.000. max indica que el tamaño máximo de almacenamiento es de $2^{31}-1$ bytes (2 GB). El tamaño de almacenamiento, en bytes, es dos veces la longitud real de los datos especificados + 2 bytes. Los sinónimos ISO de nvarchar son national char varying y national character varying.

- **varchar [(n | max)]**

Datos de cadena no Unicode de longitud variable. n define la longitud de la cadena y puede ser un valor entre 1 y 8.000. max indica que el tamaño máximo de almacenamiento es de $2^{31}-1$ bytes (2 GB). El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes. Los sinónimos ISO de varchar son char varying o character varying.

4.5. Implantación y Aceptación del Sistema (IAS)

Este proceso consiste en dar validez al programa y cada una de las opciones. Como su elaboración se ha hecho bajo la supervisión de una empresa, es la misma la que ha ido aceptando cada uno de los pasos que se iban dando, así como la que iba dando los requisitos que debía tener el mismo.

En esta parte de Métrica v3, por tanto, lo único que se ha ido haciendo es, a la vez que se iban implementando nuevas opciones, mostrársela al cliente para que diera el visto bueno si cumplía sus exigencias.

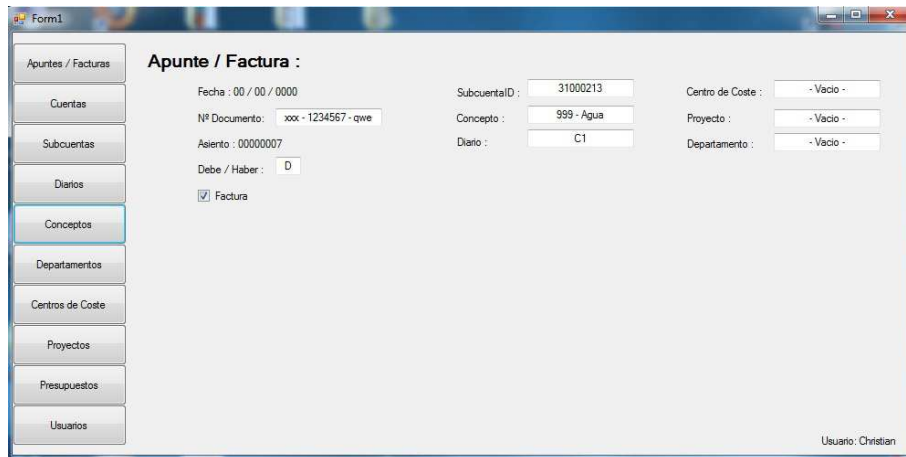
Fases de la aplicación

Durante el tiempo en el que se ha construido el programa, como ya hemos dicho, se han hecho una serie de entrevistas con el cliente, para ir dándole forma, y haciendo posibles modificaciones.

La primera entrevista, consistió en hacer un esquema de las opciones que eran necesarias, así como de la información que iba a ser necesaria guardar. Este parte ya la hemos desarrollado anteriormente en el apartado 4.2. Análisis del Sistema de Información.

Tras esto, tomamos las decisiones de que programa íbamos a usar, el lenguaje que era el más apropiado, y también la metodología más interesante para este tipo de proyectos. Hemos visto en el apartado 2. Herramientas de desarrollo, que finalmente nos decidimos por .net dentro del entorno de Visual Studio como la mejor opción.

Una vez hecho esto, elaboramos un primer interfaz, aún sin base de datos, para que el cliente pudiera observar visualmente como iba a quedar la aplicación.



Vemos una imagen de esta fase preliminar, en la que tenemos ya un menú de opciones, pero aun sin la zona derecha elaborada, sobre la que finalmente se decidió trabajar a base de ventanas.

A partir de entonces, todas las consultas se hacían por medio de emails y teléfonos, ya que se trataban de consultas pequeñas, sobre restricciones o pequeñas modificaciones, que se iban haciendo. Durante este periodo, una vez más se hizo un encuentro para ver la evolución general y aceptar el camino que se estaba tomando.

Una vez terminado, se hizo una importante serie de pruebas. Para ello se crearon un gran número de ejemplo para dar uso al programa y ver que todo funcionaba correctamente. Para ello, se introdujeron un gran número de códigos en cada una de las opciones, así como, se crearon unas 1500 facturas, mezclando todas las posibilidades posibles, sobre todo para ver en los apartados de Balances y Presupuestos, que todo funcionaba correctamente.

4. Desarrollo del Sistema de Información

A continuación vemos listados de estos ejemplos que se agregaron:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Asiento	Fecha	NumDocumento	DH	Subcuenta	Diario	Concepto	Departamento	Centrocoste	Proyecto	Cliente	IVA	Total	Usuario		
2	2	2012-09-27	00009082	D	20600002	N1	802	COM	100	112		4	1190,89	Christian		
3	3	2012-09-27	00009083	H	20600002	N1	802		100	112			2323,40	Christian		
4	4	2012-09-27	00009084	H	20600002	N1	802		100	112			23,30	Christian		
5	5	2012-09-27	00009085	D	47200012	C1	102	COM	100	112		0	91,87	Christian		
6	8	2012-09-27	00009086	H	47200012	C1	102	COM	100	112		0	2,02	Christian		
7	9	2012-09-27	00009087	D	47200012	C1	102	COM	100	112		0	1,20	Christian		
8	11	2012-09-27	00009088	D	43640076	N1	102	VES	100	112		8	3,20	Christian		
9	12	2012-09-27	00009089	D	43640076	N1	102	VES	100			8	30,40	Christian		
10	14	2012-09-27	00009090	H	43640076	N1	102	VES	100			8	11,11	Christian		
11	18	2012-09-27	00009091	H	10000001	P2	102		100				12,32	Christian		
12	19	2012-09-27	00009092	H	10000001	P2	102		100				45,34	Christian		
13	20	2012-09-27	00009093	H	10000001	P2	102		100				112,10	Christian		
14	21	2012-09-27	00009094	H	62200008	N1	102		100				24,10	Christian		
15	22	2012-09-27	00009095	D	62200008	N1	102		100				99,30	Christian		
16	23	2012-09-27	00009096	D	21300011	N1	306	COM					675,30	Christian		
17	24	2012-09-27	00009097	D	21300011	V1	202	VEN	250	112		18	23,30	Christian		
18	25	2012-09-27	00009098	H	60000001	V1	202	VEN	250	112		18	23,30	Christian		
19	26	2012-09-27	00009099	H	31000001	V1	202	VEN	250	112		18	222,11	Christian		
20	27	2012-09-27	00009100	D	31000001	V1	202	VEN	250	112		18	23,22	Christian		
21	28	2012-09-27	00009101	D	70000001	V1	202	VEN	250	112		18	23,30	Christian		
22	29	2012-09-27	00009102	H	70000001	V1	202	VEN	250	112		18	23,30	Christian		
23	30	2012-09-27	00009103	H	62200008	V1	202	VEN	250	112		18	11,20	Christian		
24	31	2012-09-27	00009104	H	62200008	V1	202	VEN	250	112		18	0,75	Christian		
25	33	2012-09-27	00009105	D	62200008	V1	202	VEN	250	112		18	9,10	Christian		

TablaCuenta TablaSubcuenta TablaFacturas TablaPresupuestos TablaConcepto TablaDiario TablaPrivilegio

Listo 100%

Una vez terminada la fase de testeo, y viendo que todos los posibles errores estaban solucionados, se daba por finalizado el proyecto, listo para ser utilizado.

5. Mantenimiento del Sistema de Información (MSI)

La parte de Mantenimiento del Sistema consiste en una vez implantado el programa en la empresa que se va a usar, de qué manera se mantendrá el mismo.

Como la construcción del programa no lo ha hecho un trabajador propio de la empresa sino que se hace de forma externa, no se tendrá al programador original del mismo para controlar los posibles problemas que puedan surgir, o los cambios que se pretendan hacer en un futuro, por lo que si no tienen a alguien que se pueda hacer cargo, tendrán que contratar a alguien que se encargue de ello, basándose en los manuales que quedan de la aplicación.

En principio, el programa no requiere un gran mantenimiento, y a parte de lo ya comentado, como puedan ser cambios o errores, lo único que si sería necesario sería hacer copia de los datos para salvaguardar la información. A continuación veremos en qué consiste la seguridad de datos, los tipos de almacenamiento y que tipos de archivos habría que guardar, entre otras cosas.

5.1. Sistema de Almacenamiento

¿Qué es un sistema de discos duros en RAID?

La palabra RAID es la abreviación de la frase inglesa "Redundant Array of Independent Disk" o conjunto redundante de discos duros independientes que viene a ser como una matriz de discos duros interconectados entre sí y cuya peculiaridad es que se comportan como un único disco es decir, la información se multiplica en cada disco, se graba la misma información en cada uno de ellos, de esta forma si existiese un error físico o mal funcionamiento en uno de ellos el sistema podría continuar funcionando.

Un *Raid* es un elemento de almacenamiento de datos e información mucho más seguro que el típico que podemos encontrar en un ordenador de sobremesa.

Por norma general un sistema RAID suele encontrarse en un servidor empresarial contando con más de un disco duro para que el sistema RAID realice copias de seguridad automatizadas, a diferencia de un ordenador de sobremesa o estación de trabajo que cuenta con un único disco duro.

Copias automatizadas

El sistema RAID duplica, triplica o cuadruplica la información recibe, disponiendo de varios discos duros que contienen la misma información, de esta forma si un disco duro falla, el sistema RAID continuará trabajando con todos los discos duros restantes, sin tener que parar el sistema de servidor de ficheros.

Esta función hace que la empresa no sufra un parón técnico.

El sistema de discos duros RAID suele avisar al usuario administrador o encargado del mantenimiento del servidor de los fallos ocurridos en el proceso de grabación de datos para que actúe en consecuencia. En muchas de estas ocasiones el fallo del disco duro podrá solucionarse reemplazándolo por otro y activando la opción de forzado de "mirror" o espejo, pero en ocasiones la pérdida de datos puede ser total y no afectar única y exclusivamente a un disco duro por ejemplo en los casos de incendio o inundación o si entrase un hacker al sistema y borrara deliberadamente los datos.

El mercado ofrece, además, otras opciones para el alojamiento de las unidades RAID, a saber:

NAS: Los dispositivos NAS (Network-Attached Storage, almacenamiento conectado a la red) facilitan la configuración de las escasas versiones RAID que ofrecen, así como el mapeo de los usuarios de la red corporativa en la matriz RAID de almacenamiento conectado.

eSATA: Las cajas de almacenamiento eSATA (external Serial Advanced Technology Attachment), separadas recientemente de las matrices NAS, constituyen carcasas diseñadas para establecer una conexión con el puerto eSATA del ordenador. Suelen ofrecer distintos niveles RAID y otras funcionalidades, tales como la sustitución "en caliente" (extracción de una unidad e introducción de otra de recambio mientras el sistema sigue ejecutándose), o la posibilidad de añadir una unidad con el sistema en ejecución.

Por último, la elección del nivel RAID se configura como un pulso entre la variable del rendimiento y la de la seguridad de los datos. En resumen, estas son las opciones más habituales:

RAID 0

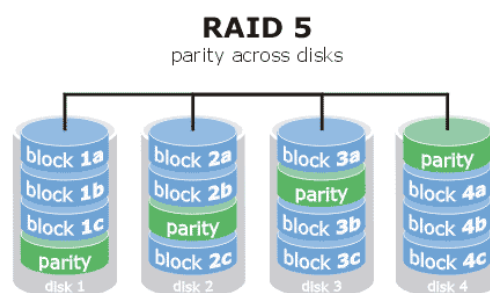
RAID 0 representa el nivel de mayor rendimiento disponible. Con RAID 0, se realiza una secuenciación de los datos de una unidad con respecto a la siguiente. Durante la lectura de datos, mientras la unidad en cuestión transfiere la información al usuario, una segunda unidad (o una tercera, una cuarta, etc.) puede procesar, paralelamente, el siguiente bloque de datos. Este sistema elimina prácticamente los tiempos de espera durante el procesamiento de datos. La capacidad de lectura es excelente, pero no existe ningún sistema de protección de los datos. Si se produce un fallo en una de las unidades, se pierden todos los datos, que pueden considerarse irrecuperables.

RAID 1

RAID 1 replica las unidades. En efecto, se produce una réplica exacta en otra unidad de todos los datos de lectura o escritura que se procesan en una determinada unidad. La capacidad de lectura es adecuada: mientras una unidad procesa un sector, la segunda unidad lee el siguiente. Si se genera un fallo en una unidad, los datos están a salvo, ya que se han replicado en otra unidad. Aunque el nivel RAID 1 es más caro e implica la realización de réplicas de las unidades, ofrece un gran rendimiento.

RAID 5

RAID 5 constituye, probablemente, el nivel de mayor disponibilidad. Los sistemas de almacenamiento NAS y los dispositivos eSATA incorporan dicho nivel, que, además, suele ser compatible con el dispositivo RAID de la placa base y con los controladores RAID. RAID 5 garantiza la seguridad de los datos al dividirlos entre unidades, y gracias a la redundancia que ofrece la información de paridad, que permitiría restaurar los datos si una unidad fallara. El rendimiento de RAID 5 no es tan elevado como el que proporcionan RAID 1 o RAID 0, pero existen nuevos niveles que ofrecen solución a algunos de los problemas asociados a la capacidad de lectura.



RAID 10

RAID 10 es en realidad una combinación de RAID 1 y RAID 0. En este nivel, la información se divide entre las distintas unidades para proporcionar una gran velocidad de lectura de los datos de las unidades. Las unidades con que trabajan en RAID 0 se replican a efectos de la seguridad de los datos. Dado que esta configuración requiere, al menos, cuatro unidades (dos para la matriz de nivel 0 y otras dos para el duplicado), su coste resulta más elevado. Por otra parte, no todas las carcassas diseñadas para los sistemas de las pymes tienen capacidad para albergar más de cuatro unidades.

Conclusión

Por todo ello, podemos llegar a la conclusión de que la mejor manera, o la que más se acerca a lo que puede necesitar nuestra empresa sería un Raid 5, usando varios discos duros, en este caso pondríamos 4. Con ello nos aseguramos una velocidad considerable a la vez que una seguridad de que no perderemos los datos.

5.2. Ubicación física para la copia

En primer lugar debemos hablar de la ubicación que tendrá dentro de la empresa la Base de Datos. Dichos datos como hemos dicho irán en un Raid 5 para beneficiar la seguridad de los mismos. Además será aconsejable ubicarlos en una sala aparte, en el cual podamos montar el servidor, con los pcs, discos duros y demás aparatos centrales que sean necesarios. Esto favorecerá el poder cerrar la sala con llave y solo el personal autorizado acceder a él. Esta sala para mayor rendimiento y durabilidad deberá estar a una temperatura fría, además, dependiendo de la importancia de los mismos, se le podría añadir cámaras de seguridad.

Una vez tenido esto en cuenta pasaremos a nuestra copia que como ya dijimos se tratará de 2 discos duros externos. El personal encargado realizara copias diarias (una al comenzar el día y otra al acabar la jornada matinal) y dichas copias se llevarán fuera del edificio ya que así estaremos seguros ante posibles incendios, robos, etc.

Otra buena opción, aunque más costosa y que podría tener fallos de seguridad, sería tener la copia alojada en un servidor externo. Este caso dependerá de lo necesario de los datos ya que si se quieren usar desde distintos sitios, esta opción será la más interesante. Como ya hemos dicho esta opción sufre de problemas de seguridad varios (ataques online, virus, etc.).



5.3. Software para la copia

Primero tendremos en cuenta nuestro RAID. En nuestro caso dijimos que nos quedaríamos con un RAID 5. Estos traen su software de copia de seguridad incluido por lo que no tendríamos que molestarnos en buscar software.

Luego valoraremos las 2 opciones mayoritarias que hemos estado mirando. En primer lugar estaría la opción online. Los servicios en línea, tales como Xdrive, que los usuarios puedan subir sus archivos a un servidor para almacenamiento seguro. Aunque puede ser conveniente tener los datos disponibles siempre que sea una conexión a Internet está disponible, hay algunas limitaciones.

Los servicios que generalmente cobran una tarifa mensual con respecto a la cantidad de espacio de almacenamiento necesario. En Xdrive, por ejemplo, los gastos de almacenamiento de 5GB de 10 € por mes.

Y por otro lado tenemos la copia en discos duros diariamente, la cual será realizada fácilmente con cualquier programa de software libre y se realizara basándose en uno de los tipos de copias que veremos a continuación.

5.4. Contenido de la copia

Decidir qué se va a incluir en la copia de seguridad es un proceso más complejo de lo que parece a priori.

Si copiamos muchos datos redundantes agotamos la capacidad de almacenamiento disponible rápidamente. Si no realizamos una copia de seguridad de los suficientes datos, podría perderse información crítica.

La clave está en guardar copias de seguridad sólo de aquello que se ha modificado.

Archivos a copiar

Sólo copiar los ficheros que se hayan modificado.

Depósito del sistema de ficheros

Copiar el sistema de ficheros que tienen los ficheros copiados. Esto normalmente implica desmontar el sistema de ficheros y hacer funcionar un programa como un depósito. Esto es también conocido como copia de seguridad particionada en bruto. Este tipo de copia de seguridad tiene la posibilidad de hacer funcionar una copia más rápida que la simple copia de ficheros.

Control de Cambios

Algunos sistemas de ficheros poseen un bit de archivo para cada fichero este nos indica si recientemente ha sido modificado. Algunos softwares de copia de seguridad miran la fecha del fichero y la comparan con la última copia de seguridad, para así determinar si el archivo se ha modificado.

Incremental a nivel de bloque

Un sistema más sofisticado de copia de seguridad de ficheros es el basado en solamente copiar los bloques físicos del fichero que han sufrido algún cambio. Esto requiere un alto nivel de integración entre el sistema de ficheros y el software de la copia de seguridad.

Incremental o diferencial binaria

Son tecnologías de backup que se desarrollan en la década de 2000. El método es similar a la Incremental a nivel de bloque, pero basada en reflejar las variaciones binarias que sufren los ficheros respecto al anterior backup. Mientras las tecnologías a nivel de bloque trabajan con unidades de cambio relativamente grandes (bloques de 8Ks, 4Ks, 1K) las tecnologías a nivel de byte trabajan con la unidad mínima capaz de ahorrar espacio para reflejar un cambio. Otra diferencia importante es que son independientes del sistema de ficheros. Actualmente son las tecnologías que consiguen la máxima compresión relativa de la información y ofrecen así una ventaja importante en las copias de seguridad a través de la Internet.

Versionando el sistema de ficheros

El versionado del sistema de ficheros se mantiene atento a los cambios del fichero y crea estos cambios accesibles al usuario. Esta es una forma de copia de seguridad que está integrada al ambiente informático.

Otro factor a tener en cuenta será el tipo de copia de seguridad. A continuación valoraremos los distintos tipos.

Tipos de copia de seguridad:

- **Desestructurado**

Un almacén desestructurado podría ser simplemente una pila de disquetes o CD-R con una mínima información sobre qué ha sido copiado y cuándo. Ésta es la forma más fácil de implementar, pero ofrece pocas garantías de recuperación de datos.

- **Completa + Incremental**

Un almacén completo-incremental propone hacer más factible el almacenamiento de varias copias de la misma fuente de datos. En primer lugar se realiza la copia de seguridad completa del sistema. Más tarde se realiza una copia de seguridad incremental, es decir, sólo con los ficheros que se hayan modificado desde la última copia de seguridad. Recuperar y restaurar un sistema completamente a un cierto punto en el tiempo requiere localizar una copia de seguridad completa y todas las incrementales posteriores realizadas hasta el instante que se desea restaurar. Los inconvenientes son tener que tratar con grandes series de copias incrementales y contar con un gran espacio de almacenaje.

- **Espejo + Diferencial**

Un almacén de tipo espejo + diferencial inversa es similar al almacén completo-incremental. La diferencia está en que en vez de hacer una copia completa seguida de series incrementales, este modelo ofrece un espejo que refleja el estado del sistema a partir de la última copia y un historial de copias diferenciales. Una ventaja de este modelo es que solo requiere una copia de seguridad completa inicial. Cada copia diferencial es inmediatamente añadida al espejo y los ficheros que son remplazados son movidos a una copia incremental inversa. Una copia diferencial puede sustituir a otra copia diferencial más antigua sobre la misma copia total.

- **Protección continua de datos**

Este modelo va un paso más allá y en lugar de realizar copias de seguridad periódicas, el sistema inmediatamente registra cada cambio en el sistema anfitrión. Este sistema reduce al mínimo la cantidad de información perdida en caso de desastre.

- **Sintética**

Esta tecnología permite crear una nueva imagen de copia de respaldo a partir de copias de respaldo anteriormente completas y posteriores incrementales. Es de gran utilidad sobre todo en redes de almacenamiento (SAN) ya que no es necesaria la participación del host/nodo final, quitándole mucha carga de proceso.

6. Conclusiones y Líneas Futuras

6.1. Líneas Futuras de Trabajo

Para tratar las posibles líneas futuras de ampliación, tenemos que recordar que se trata de un proyecto hecho a medida para una empresa, lo que supone que tendremos varios posibles casos en los que podríamos ampliar y seguir trabajando con el programa.

Lo primero a tener en cuenta sería si la empresa amplía sus pretensiones respecto al programa, es decir, que sea la propia empresa la que pida **nuevas funciones**. Estas podrían ser por ejemplo, temas de recursos humanos, tesorería, control de nóminas, etc. Estas opciones serán sencillas de añadir tal y como está planteado ya que se trataría de poner una pestaña que abriera una nueva ventana donde introducir datos o donde consultar mediante tablas lo que se busque conseguir.

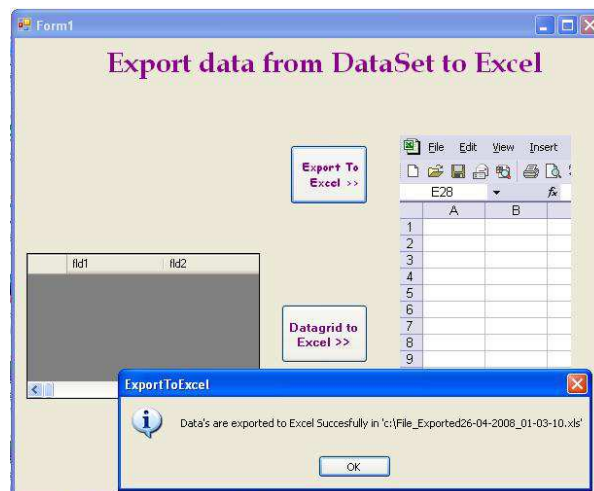
Además podría ser necesario nuevas tablas dentro de la Base de Datos, lo que sería también bastante sencillo. Sí que tendríamos que controlar si las nuevas operaciones y condiciones influyen en las ya introducidas anteriormente, que puede perfectamente mirar la persona que vaya a hacer los cambios en el manual sin tener que ser el mismo programador original. Estas opciones como ya hemos dicho serían como las opciones originales, totalmente a voluntad del cliente.

Otra vía de trabajo con respecto al proyecto, sería a nivel utilitario o **gráfico**. Una vez más mencionamos que el proyecto está hecho a medida del cliente, el cual quería un interfaz muy sencillo para los que fueran a usar el programa, personas que no tienen estudios informáticos, supieran rápidamente las opciones y cómo usar el programa. Por ello se optó por colocar botones claros para usar cada opción. Quizás hoy en día, se podría haber tendido a una opción en la que se manipulara los datos directamente desde la tabla que muestra los datos. Desde allí se podría haber introducido los datos o borrado, sin necesidad de colocar botones externos.

Esta sería ser una de las cosas que se podría cambiar en un futuro para esta u otra empresa. Es una opción relativamente sencilla de introducir puesto que cada pestaña lleva a una ventana independiente de las demás, con lo cual los cambios no influyen de unos a otros. Además para evitar que se pudiera modificar los datos directamente de las tablas se bloquearon, con lo cual sólo sería necesario volver a activarlas.

La última línea futura de trabajo sería en realidad la **unión** de las dos anteriores. Dado el caso de querer usar el programa en otra empresa, se podría adaptar tanto modificando las opciones ya existente como añadiendo nuevas. Desde el punto de vista del programador se recomendaría añadir sin quitar opciones, ya que toda opción que se tenga aunque no se use, siempre es aconsejable tenerla en caso de necesitarse. Por ello, si para introducir Apuntes, por ejemplo, si quieren guardar más datos, se añadirían nuevos campos sin borrar los que no se vayan a usar. Llegados el caso de tener muchos datos a mostrar, o más concretamente, muchas campos distintos, se podría limitar las opciones a la hora de mostrar en la tabla, pero no eliminando campos.

Una vez visto estas líneas futuras, podemos plantear otras cosas en las que trabajar. Hemos visto como el programa trae una opción para imprimir los datos mostrados por pantalla, en este caso las tablas. Esta opción por tanto permite además pasar los datos a formato .pdf, lo cual puede resultar muy útil para presentar informes en papel. Es por ello, que quizás sería interesante añadir otras opciones como pueden ser, pasar los datos a formato .xls, es decir, volcar la información a un archivo **Excel**, donde poder continuar trabajando con ellos.



Por otro lado, siempre a necesidad del cliente también, se podría acortar los pasos para enviar información y en vez de pasarlos a .pdf y luego enviarlos por email, que el programa trajera una opción de escribiendo un email, enviarlos directamente.

También existe otra posibilidad, y es la contraria a la vista anteriormente. Si tenemos información almacenada en un archivo Excel por ejemplo y queremos pasar estos datos a nuestra base de datos, crear una nueva ventana que lea y añada estos datos sin la necesidad de un informático que lo haga directamente en nuestra BD.

Como se puede ver, este programa queda totalmente abierto a posibles cambios y/o ampliaciones. Dependerá de las exigencias del cliente.

6.2. Conclusiones

Este proyecto me ha supuesto un gran reto ya que era la primera vez que me enfrentaba con algo así, y aunque había hecho pequeños programas, ninguno de esta escala. Ya había hecho practicas trabajando con este tipo de programas relacionados con la contabilidad, es por esto que me animé a realizarlo. Uno de los problemas de la carrera es que por su limitado tiempo no es posible ver programas más en profundidad y encontrarse con situaciones más parecidas a las que me he encontrado aquí, y las que se pueden ver en el día a día de un programador.

La dificultad en sí, no radica en el tamaño del mismo, ya que, en ese caso, solo se trataría de dedicarle más tiempo, sino de programar cosas que no había hecho antes. En este punto, ha sido clave el lenguaje de programación Visual Basic .net, y el programa Visual Studio, ya que nunca anteriormente había trabajado con ellos.

Pese a esto, no han supuesto un gran esfuerzo adaptarme, ya que tienen una gran similitud a otros que había visto antes. Es por ello que, una vez terminado, me alegro de la elección que hice. Creo que por esto mi formación previa durante la carrera se adecua perfectamente a lo que necesitaba para su realización.

La parte principal ha sido el lenguaje de programación y el entorno para programar. Sin duda alguna ha sido una alegría para mí la sencillez para crear el interfaz gráfico del programa, y en cuanto al lenguaje, bastante similar a lo visto anteriormente, por tanto, he podido programar una gran parte simplemente con los conocimientos que ya tenía, y cuando necesitaba hacer uso de nuevos recursos no he tenido problemas para buscarlos ya que entendía perfectamente códigos con solo verlos por encima. Espero seguir trabajando con este programa o si no con entornos parecidos.

Trabajar con la Base de Datos, me ha resultado realmente sencillo y cómodo, ya que se trataban de consultas parecidas a las que ya había aprendido previamente. Además, la conectividad de Visual Studio con Sql Server hace muy fácil trabajar con ellos y ver los resultados rápidamente.

En cuanto a Métrica-3, debo decir que nunca había visto lo que era y lo veo como una mecánica muy útil para trabajar en este tipo de proyecto. Sí que es verdad, que en mi caso no he necesitado seguir cada uno de los pasos, pero sí me ha servido de guía a seguir. Para mí, las partes principales en las que he trabajado han sido la planificación previa del Sistema de Información, y luego el Diseño y Construcción del mismo.

Una vez terminado el proyecto, creo que ha supuesto un importante paso para mí, tanto en experiencia personal como en conocimientos adquiridos. Sin duda alguna he aprendido mucho, no sólo en comandos y en código de programación, también en cómo organizarse para llevar a cabo este tipo de proyectos. Creo que de volver a realizarlo, seguiría otros pasos, por lo que en mis futuros trabajos y programas ya tengo esta experiencia inicial para facilitarme las cosas. Además, el hecho de haber sido un proyecto muy relacionado con la contabilidad y la empresa, me ha hecho darme cuenta de que es una parte de la Informática que me atrae.

7. Manual de Usuario

LECCIÓN 1: INTRODUCCIÓN AL PROGRAMA	85
Entorno de Trabajo.	
Menús del Programa.	
LECCIÓN 2: APUNTES / FACTURAS	86
Creación de Apuntes.	
Creación de Facturas.	
Confirmación de Apuntes / Facturas.	
LECCIÓN 3: CONSULTAS	90
Criterios de Consultas.	
Modificación de Apuntes / Facturas.	
Borrado de Apuntes / Facturas.	
LECCIÓN 4: CUENTAS	93
Creación de Cuentas.	
Modificación de Cuentas.	
Borrado de Cuentas.	
Detallado de Cuentas.	
Listado de Cuentas.	
LECCIÓN 5: SUBCUENTAS	96
Creación de Subcuentas.	
Modificación de Subcuentas.	
Borrado de Subcuentas.	
Detallado de Subcuentas.	
Listado de Subcuentas.	
LECCIÓN 6: DIARIOS	99
Creación de Diarios.	
Modificación de Diarios.	
Borrado de Diarios.	
Detallado de Diarios.	
Listado de Diarios.	



LECCIÓN 7: CONCEPTOS	101
Creación de Conceptos.	
Modificación de Conceptos.	
Borrado de Conceptos.	
Detallado de Conceptos.	
Listado de Conceptos.	
LECCIÓN 8: DEPARTAMENTOS	103
Creación de Departamentos.	
Modificación de Departamentos.	
Borrado de Departamentos.	
Detallado de Departamentos.	
Listado de Departamentos.	
LECCIÓN 9: CENTROS DE COSTE	106
Creación de Centros de Coste.	
Modificación de Centros de Coste.	
Borrado de Centros de coste.	
Detallado de Centros de Coste.	
Listado de Centros de Coste.	
LECCIÓN 10: PROYECTOS	108
Creación de Proyectos.	
Modificación de Proyectos.	
Borrado de Proyectos.	
Detallado de Proyectos.	
Listado de Proyectos.	
LECCIÓN 11: BALANCES	110
Consulta de Balances.	
LECCIÓN 12: PRESUPUESTOS	111
Agregar Presupuestos.	
Consulta de Presupuestos.	
LECCIÓN 13: CLIENTES	114
Creación de Clientes.	
Búsqueda de Clientes.	
Modificación de Clientes.	
Borrado de Clientes.	
LECCIÓN 14: USUARIOS	116
Creación de Usuarios.	
Creación de Privilegios.	

LECCIÓN 1: INTRODUCCIÓN AL PROGRAMA

Entorno de Trabajo.

Como puede observar, el **programa** es una típica aplicación de Windows y su funcionamiento es similar al de cualquier otra aplicación de ese sistema.

A continuación veremos una descripción de las partes más importantes que componen el entorno de la aplicación con las que nos iremos familiarizando para iniciamos en el manejo de la aplicación.

La primera pantalla será la que dé acceso al mismo al usuario. Para ello deberá introducir su usuario y contraseña previamente creados. Para más información de cómo crear usuario y darle privilegios, ver la Lección 15: Usuarios.



En la parte izquierda de la aplicación se encuentra la **barra de menús**. Desde esta barra podremos acceder a todas las opciones o comandos de la aplicación. En el siguiente capítulo describiremos con más detalle los menús del programa.

La parte central de la aplicación es el **área de trabajo**. En ella se mostrarán todas las ventanas y cuadros de diálogo para la introducción de datos. En esta zona también aparecerán los listados e informes que realicemos por pantalla.

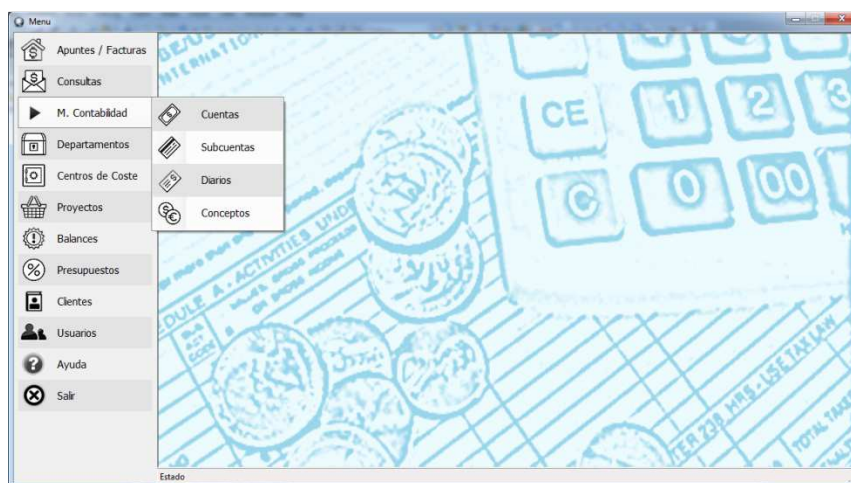
Esta parte central tiene la característica de permitir tantas ventanas como opciones tiene el programa, y poder ver dos a la vez, si bien para mayor facilidad de trabajo, se permitirá maximizar y minimizar cada una de ellas cuando se quiera.

Por último, en la parte inferior del entorno está la **barra de estado**. En ella podemos ver algunos pasos que hagamos como crear o borrar información.

Menú del Programa.

En la zona izquierda de nuestro programa tenemos el menú de opciones disponibles en todo momento, aunque dependiendo de los privilegios del usuario, tendremos acceso a unas opciones u otras. Cada vez que se cliquee una opción, se abrirá la ventana correspondiente, quedando las otras que ya haya abiertas en un segundo plano, pudiendo retomarlas más tarde tal y como la dejamos.

Para saber las funciones de cada una de las opciones, tan solo hay que consultar su apartado correspondiente de este manual.



La opción de **Ayuda** abre un fichero con este manual, para tener a mano en cualquier momento una guía de cómo usar el programa.

LECCIÓN 2: APUNTES / FACTURAS

- Apunte:

Se denomina Asiento Contable o Apunte Contable a cada una de las anotaciones o registros que se hacen en un libro o diario de contabilidad, que se realizan con la finalidad de registrar un hecho económico que provoca una modificación en el patrimonio de una empresa lo que conlleva a un movimiento en las cuentas de una empresa, usando una partida del Debe y otra del Haber.

- Factura:

La Factura, factura de compra o factura comercial es un documento mercantil que refleja toda la información de una operación de compraventa. La información fundamental que aparece en una factura debe reflejar la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar.

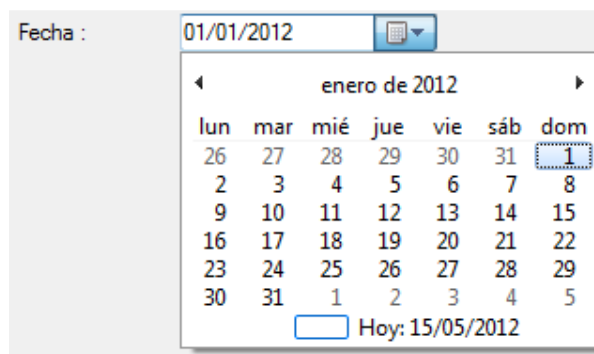
Creación de Apuntes.

Para poder empezar a estudiar las opciones de la aplicación, lo primero que debemos hacer, es crear una nueva entrada en la base de datos de apuntes y facturas. Para meter un nuevo apunte tendremos que rellenar una serie de campos que veremos a continuación, los cuales tienen una serie de condiciones. Como ya hemos visto tendremos que ir a la pestaña de Apuntes / Facturas y nos encontraremos una subventana como esta:

The screenshot shows a software window titled 'Apufac' with a sub-header 'Apuntes / Facturas'. The form contains the following fields and controls:

- Fecha :** 22/02/2012 (with a calendar icon)
- Nº Documento :** 123456
- Asiento :** 000000
- Debe / Haber :** (empty)
- Factura :** (checked checkbox)
- ClienteID :** (empty)
- SubcuentaID :** 50009800
- Diario :** (dropdown menu)
- Concepto :** (dropdown menu)
- Departamento :** 111
- Centro de Coste :** 111
- Proyecto :** 100
- Cantidad :** (empty)
- TipoIVA :** (dropdown menu)
- Añadir** and **Finalizar** buttons
- Total :** (empty)
- Usuario :** (empty)

El primer campo que nos encontramos es el de **Fecha**, el cual viene por defecto la fecha actual. Esta la podremos modificar pulsando sobre la flecha, y se abrirá un menú desplegable para elegir la fecha que nos interese.



Fecha : 01/01/2012

enero de 2012

lun	mar	mié	jue	vie	sáb	dom
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Hoy: 15/05/2012

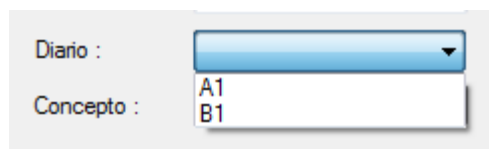
El siguiente campo es el número de **Documento**, donde habrá que introducir el número del apunte o factura. Este número será un código de 8 cifras numéricas.

El campo **Asiento**, es un número automático para enumerar los apuntes que se vayan introduciendo.

La casilla **Debe / Haber** será para indicar si se trata de un debe o un haber. Este y otros términos serán ampliados en el apartado Apéndice. En esta casilla tendremos dos posibilidades. Escribir una D mayúscula en caso de Debe y H mayúscula en caso de Haber. Este campo será de obligado relleno.

En el apartado **SubcuentaID**, deberemos introducir un código de subcuenta válido, es decir, un código de ocho cifras, ya introducido en la pestaña de Subcuentas. Si no existiera anteriormente ese código, el programa mostrará un mensaje de error, indicando que este ha sido el problema.

Para elegir los dos siguientes campos, **Diario** y **Concepto**, tendremos dos botones desplegados entre los que elegir. Estos códigos serán los códigos que previamente se hayan introducido en sus pestañas correspondientes. Por tanto mientras que no se hayan metido en la base de datos al menos un código de Diario y un código de Concepto válidos, no se podrán crear nuevas entradas de apuntes.



Diario :

Concepto :

Hasta aquí, todas las opciones que hemos visto son campos obligatorios y si no son rellenados, darán mensajes de error, indicando cual es el motivo.

El campo **Departamento**, es un campo opcional, para poder organizar los apuntes/facturas por departamentos. Para ello podremos rellenar este campo con un código de tres letras. Este código debe estar previamente introducido en la base de datos, mediante su pestaña correspondiente.

El siguiente campo es también opcional y es para indicar el **Centro de Coste**. En este caso el código es de tres cifras y debe haber sido creado previamente en su pestaña correspondiente.

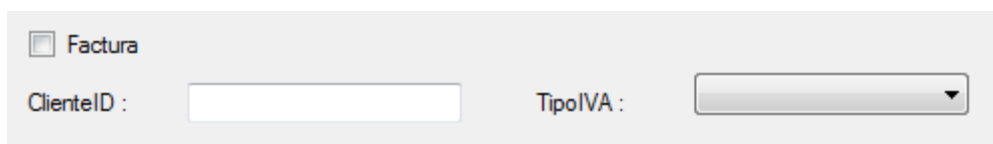
El campo **Proyecto** será similar a los dos anteriores. En él podremos introducir un código de tres cifras para indicar a que proyecto corresponde. Este código debe haber sido previamente aceptado en el apartado de Proyecto. Este campo también es voluntario y sí puede dejar vacío.

Por último, en la casilla **Cantidad**, introduciremos la cantidad final del apunte que estemos introduciendo. Esta cantidad irá con coma en caso de tener decimales.

Una vez hayamos introducido todos los valores anteriores y rellenado aquellos que son de obligado cumplimiento, pulsaremos el botón **Añadir**. Si alguno de los datos que hemos introducido es erróneo, saltará un mensaje de error indicando en cuál hemos cometido el error y por qué. En caso contrario, los datos pasaran a la tabla que vemos en la parte inferior y podremos seguir introduciendo más apuntes o facturas.

Creación de Facturas.

Las facturas, a diferencia de los apuntes, tendrán dos campos más, estos son ClienteID e IVA. Para poder introducir una nueva factura deberemos seguir los mismos pasos previos que en la creación de un apunte. Una vez hecho esto, marcaremos la casilla **Factura**, y de este modo se volverán accesibles los dos campos necesarios que deberemos rellenar.

El formulario muestra un grupo de controles. En la parte superior izquierda hay un checkbox etiquetado como 'Factura'. Debajo de él, a la izquierda, se encuentra el campo 'ClienteID' con un texto de entrada vacío. A la derecha de este campo, se encuentra el campo 'TipoIVA' con un menú desplegable que muestra una flecha hacia abajo.

Una vez activo, completaremos el campo de **ClienteID** con un código de cliente válido. Para ello, el código será creado previamente, y cumplirá el requisito de 4 cifras numéricas. Para ver cómo crear nuevos clientes ver la Lección 13: Clientes.

En la pestaña **IVA** elegiremos el IVA que le corresponde a esta factura entre los que hayamos previamente creado. Siempre que creamos una factura deberemos elegir un tipo de IVA, no pudiéndola dejar vacía.

Tras hacer estos dos pasos podremos hacer el mismo último proceso que en la creación de apuntes. Para ello, pulsaremos el botón **Añadir** y si no hemos cometido ningún error, la factura pasará directamente a la tabla inferior.

Confirmación de Apuntes / Facturas.

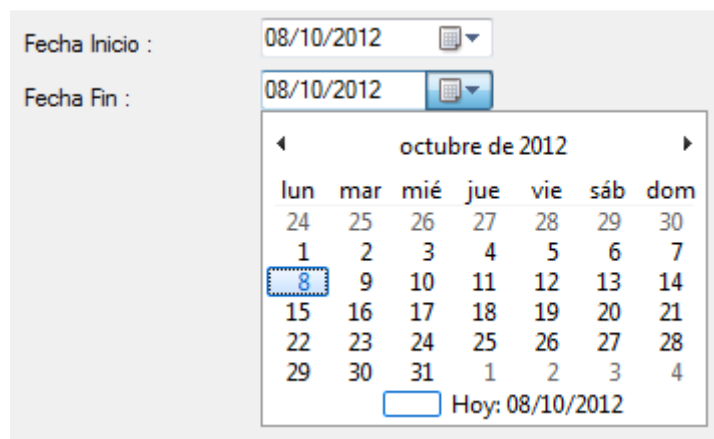
La confirmación de apuntes y facturas consiste en dar validez a los apuntes o facturas que se hayan introducido previamente y salga por la tabla. Por ello previamente podremos ir añadiendo tantas como se quiera y finalmente pulsar el botón **Finalizar**. Esta confirmación está sujeta a la condición de que la cantidad total entre debe y haber sea cero. Recordemos que un debe suma positivamente y un haber negativamente, por lo que para que nos permita finalizar, en la casilla Total, se recomienda que la diferencia entre ambas sea cero, en caso contrario nos mostrará un error, indicando el motivo. No obstante, podremos continuar pese a esto, y agregarlas aunque no sea cero.

	Asiento	Fecha	NumDocumento	DH	Subcuenta	Diario	Concepto	Departamento	Ce
▶	43	03/10/2012	123456	D	50009800	A1	001	COM	10
	44	03/10/2012	123456	H	50009800	B1	001	COM	10

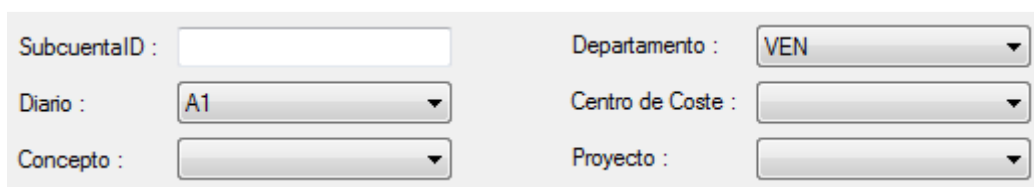
Criterios de Consultas

En esta ventana podremos consultar apuntes y/o facturas que hayamos introducido en el sistema anteriormente. Para ello tenemos una serie de opciones con las que restringir las búsquedas que saldrán por la tabla. Veamos cuales son:

En primer lugar tenemos dos campos **fechas**, inicio y final. Tenemos dos opciones para elegir las fechas, o bien podemos escribirla directamente, respetando el formato dd/mm/aaaa, o bien podemos utilizar la flecha final para abrir un calendario desplegable y seleccionar las fechas que nos interese. Un dato a tener en cuenta es que la fecha inicio es inclusive, pero la final y por tanto si queremos consultar las facturas de hoy debemos poner como fecha inicio el día de hoy y fecha final el día de mañana. Estos dos campos son obligatorios de uso.



A partir de aquí tenemos 6 campos optativos para usar en caso de querer restringir nuestra **búsqueda**, los cuales son Subcuenta, Diario, Concepto, Departamento, Centro de Coste y Proyecto.



Como se puede ver, excepto el campo Subcuenta, los demás son menús desplegables para elegir el criterio a buscar. Recordemos que no es necesario marcar cada opción, ni siquiera alguna de ellas. En Subcuenta, pondremos igualmente algún código existente. Una vez elegido lo que queramos buscar pulsaremos el botón **Mostrar**.

Además, como en las demás opciones podremos imprimir todo lo que se muestre por pantalla, pulsando el botón **Imprimir**, que hay en la parte inferior.

Modificación de Apuntes / Facturas.

Para modificar una factura, primero consultaremos tal como hemos hecho en el apartado anterior. Una vez que la factura que queremos modificar salga en la tabla, pinchamos al principio de esa fila y pulsamos el botón **Modificar**. Si pulsamos más de una fila, saldrá un mensaje de error, ya que no podemos modificar más de una factura a la vez.

Si lo hemos hecho correctamente tendremos una ventana con todos los campos igual que cuando creamos la factura (Ver Creación Apuntes / Facturas en la Lección 2). Podremos modificar todos los campos excepto el de asiento. Una vez hecho los cambios pertinentes, pulsaremos el botón **Modificar** para terminar.

Borrado de Apuntes / Facturas.

Para borrar Apuntes / Facturas, en primer lugar haremos una consulta para encontrar las que queramos eliminar. Una vez hecho, marcaremos en la tabla las filas que van a ser eliminadas. Para seleccionarlasy, pincharemos al principio de cada fila.

Si queremos eliminar más de una fila a la vez, debemos tener pulsado el botón del teclado Control, mientras cliqueamos las filas correspondientes. Tras esto pulsamos el botón **Borrar**, y nos saldrá un mensaje como el siguiente:

Asiento	Fecha	Num	Diario	Concepto	Departamento	Centrocoste	Proyecto
614	03/10/2012	89898					
615	03/10/2012	89898					
616	03/10/2012	89898					
617	03/10/2012	89898					
618	03/10/2012	89898					
619	03/10/2012	89898					
620	03/10/2012	89898					
621	03/10/2012	89898989	41000049	H	70.05	S1	208
622	03/10/2012	89898989	41000049	H	70.05	S1	208

LECCIÓN 4: CUENTAS

La Cuenta es el elemento básico y central en la contabilidad y en los servicios de pagos. La cuenta contable es la representación valorada en unidades monetarias, de cada uno de los elementos que componen el patrimonio de una empresa (bienes, derechos y obligaciones) y del resultado de la misma (ingresos y gastos), y permite el seguimiento de la evolución de los elementos en el tiempo. Por lo tanto, hay tantas cuentas como elementos patrimoniales tenga la empresa. En consecuencia, el conjunto de cuentas de una empresa supone una representación completa del patrimonio y del resultado (beneficios o pérdidas) de la empresa.

Creación de Cuentas.

Las cuentas contables se clasifican por niveles (grupos, subgrupos, cuentas y subcuentas).

Los grupos constan de un dígito (nivel 1), los subgrupos de primer nivel de dos dígitos (nivel 2), los subgrupos de tercer nivel de tres dígitos (nivel 3) siendo a partir de dicho nivel donde se suelen encuadrar las cuentas, esto es, con más de cuatro dígitos, si bien es muy habitual la utilización de 8 dígitos para una cuenta en particular, por ejemplo la cuenta de un cliente o de un proveedor o banco.

Cuentas

Código : * Descripción :

Código	Descripción
1	FINANCIACIÓN BÁSICA
10	CAPITAL
100	Capital social
101	Fondo social
102	Capital
103	Socios por desembolsos no exigidos
1030	Socios por desembolsos no exigidos, capital social
1034	Socios por desembolsos no exigidos, capital pendiente de inscripción
104	Socios por aportaciones no dinerarias pendientes

* Códigos dados por el PGC 2007

Usuario : CHRISTIAN

Para crear nuevas cuentas tendremos que tener en cuenta lo anteriormente comentado, es decir, para crear subgrupo de dos dígitos, primero tendremos que crear el grupo de un dígito. Por ejemplo:

- Para la creación del código “100 – Capital social”, previamente hemos creado el código “10 – CAPITAL” y a su vez, antes de crear este, hemos creado el grupo de un dígito “1 – FINANCIACIÓN BÁSICA”.

Una vez aclarado los niveles de cuentas, para crear uno nuevo, debemos introducir un código de entre uno y cuatro cifras numéricas en su respectivo campo, el de Código y añadirle una Descripción. Una vez hecho esto, pulsaremos el botón **Agregar**, y si hemos respetado los criterios anteriormente mencionado, obtendremos un mensaje de confirmación.

Modificación de Cuentas.

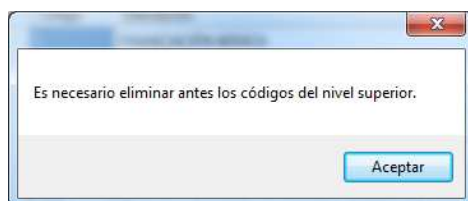
La modificación de cuentas consiste en cambiar la descripción de un código, por otra descripción. No están permitidos los cambios de códigos, ya que eso influiría en apuntes y/o facturas que ya haya en la base de datos anteriormente. Para cambiar una descripción, en el campo código pondremos el código a cambiar y en el de descripción, la nueva. Una vez rellenados pulsamos la tecla **Modificar**, y nos mostrará un mensaje como el siguiente:



Ahora solo tendremos que pulsar la tecla aceptar en caso de que estemos de acuerdo, y el cambio se habrá realizado.

Borrado de Cuentas.

Antes de ver cómo es posible borrar códigos de cuenta, debemos recordar que los códigos de cuentas van por niveles y por tanto para poder borrarlos, al igual que para crearlos, debemos respetar estos criterios. Para más información véase “Creación de cuentas” dentro de esta misma Lección.



Además, tampoco podremos borrar códigos que ya se hayan usado en algún apunte o factura anteriormente. En caso de saltarnos cualquiera de estas condiciones, nos saldrá el respectivo mensaje, indicándonos en que nos hemos equivocado. Para borrar un código de cuentas, simplemente debemos rellenar el campo código con la cadena numérica de una a cuatro cifras que la represente y pulsar el botón Borrar. Una vez hecho esto saldrá un mensaje de confirmación, para acabar con el borrado.

Detallado de Cuentas.

La utilidad de la opción Detallar es ver los cambios de un código a lo largo de un historial. Es muy útil para un administrador, para saber quién agregó o borró un código y cuando fue. Para ello simplemente en el campo código introducimos un código válido. Recordemos que en cuentas, los códigos son entre 1-4 cifras numéricas. No importa si ese código existe actualmente en la lista de cuentas. Una vez esto, pulsamos el botón detallar y nos mostrará cinco campos, el código, que será invariable, la descripción, que en varias situaciones ni siquiera será necesaria que aparezca, el usuario que hizo los cambios, la fecha cuando se hicieron y el motivo. Este último campo indicará Insertado, Borrado o Modificado, según el caso.

	CodigoCuenta	DescripcionCuenta	UsuarioCuenta	FechaCuenta	CuentaCambio
►	10		CHRISTIAN	15/10/2012	Borrado
	10	Nueva Cuenta	CHRISTIAN	15/10/2012	Insertado
	10	Nueva Cuenta 2	CHRISTIAN	15/10/2012	Modificado
	10		CHRISTIAN	15/10/2012	Borrado
	10	CAPITAL	CHRISTIAN	15/10/2012	Insertado

Listado de Cuentas.

Esta opción es una opción por defecto, ya que automáticamente se listarán los códigos en caso de agregar, borrar o incluso modificar. Se usará principalmente para volver de la opción detallado y simplemente deberemos pulsar el botón **Listar** sin tener que rellenar ningún campo extra.

LECCIÓN 5: SUBCUENTAS

Forma parte del cuadro de cuentas del Plan General de Contabilidad y constituye el mayor detalle de las cuentas de cada grupo, de forma que cada grupo se divide en subgrupos y éstos en cuentas principales, las cuales se desagregan en las subcuentas, que se identifican con códigos de cuatro cifras.

A la hora de introducir apuntes, nosotros trabajaremos con subcuentas de 8 dígitos. Antes de poder introducir cualquier apunte o factura, deberemos crear todas las subcuentas que intervengan en él.

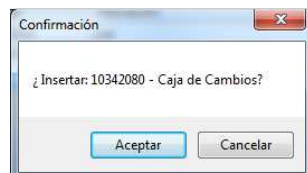
Código	Descripción
43051302	Plantillas Año 2013
43051303	Rappel Sin Iva Año 2013
43051304	Coste Logístico 2013
43051305	Acc. Com. Portugal 2013
43051306	Plantillas 2013
43100001	Ef. Comerciales a Cobrar
43100002	Cartera Talones
43150132	Taboada Boga Represent.
43150420	Ojelosa
43150878	Evaluación Anual S. 1

En este capítulo aprenderemos a dar de alta las subcuentas, a modificarlas, a darlas de baja, etc...

Creación de Subcuentas.

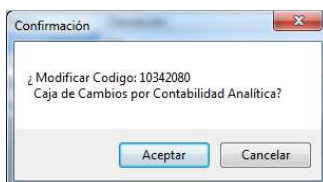
La función de **Agregar** permite agregar nuevos códigos de Subcuentas. Estos códigos deben cumplir que no sea vacío, que sea un código de 8 cifras exactamente, las cuales todas deben ser de tipo numérico. Estas ocho cifras se dividen en dos partes, una primera corresponde a un código de cuentas existente previamente completados con ceros, es decir, que si el código de cuentas del cual queremos que dependa tiene una longitud de dos, rellenaremos con dos ceros hasta completar las cuatro primeras cifras. La segunda parte, que también serán cuatro cifras, completarán el código y será lo que distinga un código de subcuenta de otro.

Por tanto, para agregar un nuevo código de subcuentas escribiremos en la casilla código las 8 cifras y en el de descripción, escribiremos una breve descripción del mismo. Esta descripción será obligatoria. Una vez rellenados ambos campos, pulsaremos el botón Agregar y tendremos que pasar un mensaje de confirmación.



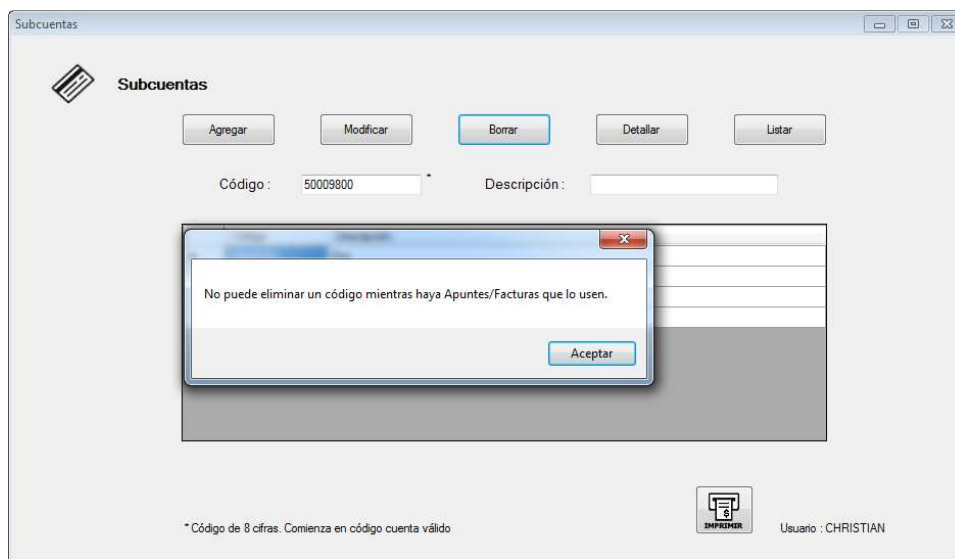
Modificación de Subcuentas.

La modificación de un código de subcuenta consiste en cambiar su descripción ya que los códigos no se pueden modificar. Esto es porque es posible que apuntes y facturas dependan de él y podrían quedarse sin código al que pertenecer. Por eso, veremos cómo cambiar la descripción de un código. Para ello en el campo código metemos una cifra de código existente anteriormente, si no saldrá un mensaje de error. Y en la descripción ponemos la nueva que le queremos dar. Al pulsar el botón **Modificar**, nos saldrá un mensaje indicándonos el código que queremos modificar, la descripción antigua y la nueva, listo para confirmar que ese es el cambio que queremos hacer.



Borrado de Subcuentas.

Para borrar un código de subcuentas, escribiremos en el campo correspondiente, el código de ocho cifras que deseamos borrar. Este código debe existir previamente en la aplicación, si no recibiremos un mensaje advirtiéndonoslo. En el campo descripción no será necesario poner nada en esta ocasión. Al pulsar el botón **Borrar**, nos mostrará un mensaje indicándonos cuál es el código y la descripción que vamos a eliminar a modo de confirmación. Para que este código pueda ser eliminado no puede existir ningún apunte o factura con este código ya que si no perderíamos información, por lo que tendremos que ir previamente a la opción de borrado de apuntes/facturas y eliminarlas (Ver Lección 3).



Detallado de Subcuentas.

El detallado de una subcuenta consiste en mostrar todos los cambios que se han realizado en un código determinado. Esta opción es accesible según los privilegios que tenga el usuario sobre la aplicación. En caso de tener dichos privilegios, al ingresar un código de subcuenta válido en el campo correspondiente, podremos pulsar el botón **Detallar**, y nos mostrará en la tabla cada vez que se ha creado, borrado o modificado dicho código, añadiendo el usuario que lo ha hecho y la fecha en la que se realizó dicho cambio.

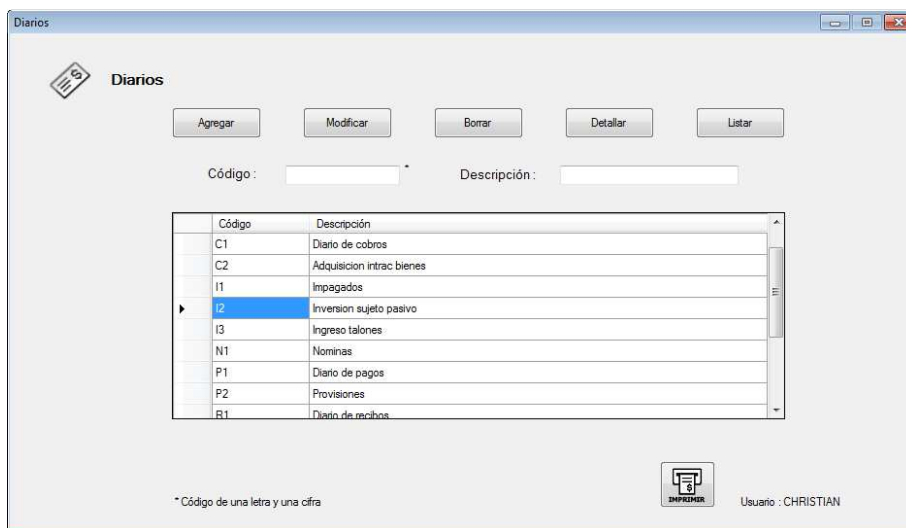
	CodigoSubcuenta	DescripcionSubcuenta	UsuarioSubcuenta	FechaSubcuenta	SubcuentaCambio
▶	10342080	Caja de Cambios	CHRISTIAN	03/10/2012	Insertado
	10342080	Contabilidad Analítica	CHRISTIAN	03/10/2012	Modificado
	10342080		CHRISTIAN	03/10/2012	Borrado
	10342080	Subcuenta Uno	CHRISTIAN	03/10/2012	Insertado

Listado de Subcuentas.

Esta opción viene por defecto al abrir la ventana de subcuentas y se actualizará tras agregar o borrar un código. Si después de hacer una consulta de subcuentas queremos listar todos los códigos, pulsaremos el botón **Listar** y veremos todos los códigos que tenemos en ese momento de subcuentas.

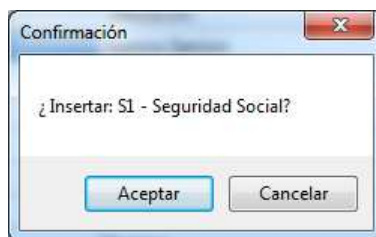
LECCIÓN 6: DIARIO

Es la clasificación de cada uno de los apuntes contables que nos permite agrupar por tipo de operación como, por ejemplo, compras interiores, intracomunitarias, inversiones, etc. El diario nos facilita el obtener los datos necesarios por actividad.



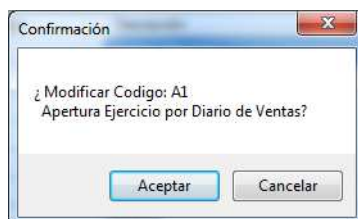
Creación de Diarios.

La función de **Agregar** permite agregar nuevos códigos de Diarios. Para agregar un nuevo código de Diario, tendremos que poner en el campo Código uno válido, que en este caso será el formado por una letra mayúscula y un número. Todo lo que no sea esto dará su respectivo mensaje de error, indicando en que nos hemos equivocado. A su vez también tendremos que asignarle una descripción a esa nueva entrada. Como el código es único y representativo, si no estaba ya anteriormente en la base de datos, se introducirá correctamente tras pasar una pantalla de confirmación.



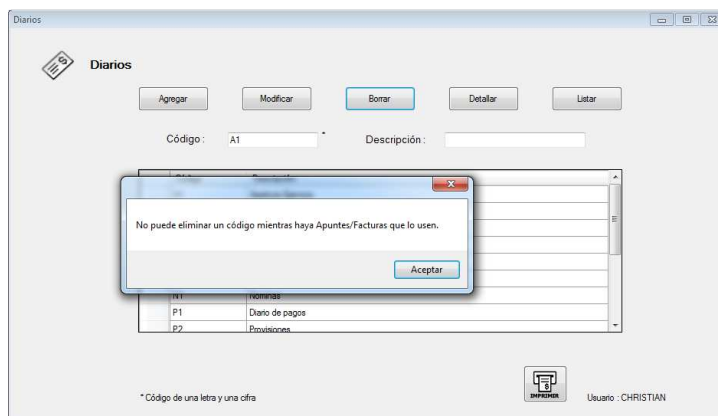
Modificación de Diarios.

La modificación de un código de diario consiste en cambiar su descripción ya que los códigos no se pueden modificar. Esto es porque es posible que apuntes y facturas dependan de él y podrían quedarse sin código al que pertenecer. Por eso, veremos cómo cambiar la descripción de un código. Para ello en el campo código metemos una cadena de código existente anteriormente, si no saldrá un mensaje de error. Recordemos que para diario, tendremos que poner una letra seguido de un número. Y en la descripción ponemos la nueva que le queremos dar. Al pulsar el botón **Modificar**, nos saldrá un mensaje indicándonos el código que queremos modificar, la descripción antigua y la nueva, listo para confirmar que ese es el cambio que queremos hacer.



Borrado de Diarios.

Para borrar un código de Diarios, escribiremos en el campo correspondiente el código que deseemos borrar. Recordar que un código diario consiste en una letra seguido de un número. Este código debe existir previamente en la aplicación, si no recibiremos un mensaje advirtiéndonoslo. En el campo descripción no será necesario poner nada en esta ocasión. Al pulsar el botón **Borrar**, nos mostrará un mensaje indicándonos cuál es el código y la descripción que vamos a eliminar a modo de confirmación. Para que este código pueda ser eliminado, no puede existir ningún apunte o factura con este código ya que si no perderíamos información, por lo que tendremos que ir previamente a la opción de borrado de apuntes/facturas y eliminarlas (Ver Lección 3).



Detallado de Diarios.

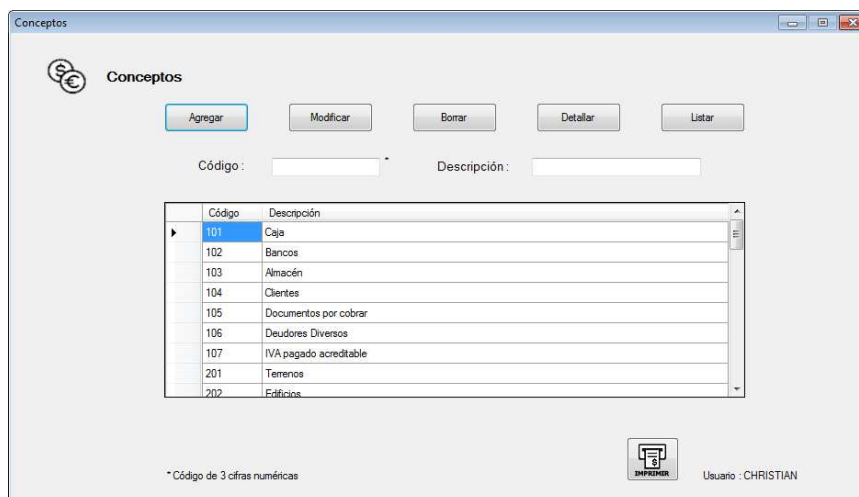
El detallado de un código de diario consiste en mostrar todos los cambios que se han realizado en un determinado código. Esta opción es accesible según los privilegios que tenga el usuario sobre la aplicación. En caso de tener dichos privilegios, al ingresar un código de diario válido en el campo correspondiente, podremos pulsar el botón **Detallar**, y nos mostrará en la tabla cada vez que se ha creado, borrado o modificado dicho código, añadiendo el usuario que lo ha hecho y la fecha en la que se realizó dicho cambio.

Listado de Diarios.

Esta opción viene por defecto al abrir la ventana de diarios y se actualizará tras agregar o borrar un código. Si después de hacer una consulta de diarios queremos listar todos los códigos, pulsaremos el botón **Listar** y veremos todos los códigos que tenemos en ese momento de diarios.

LECCIÓN 7: CONCEPTO

Los **Conceptos** son construcciones o **imágenes mentales**, por medio de las cuales comprendemos las experiencias que emergen de la interacción con nuestro entorno. Estas construcciones surgen por medio de la integración en clases o categorías, que agrupan nuestros nuevos conocimientos y nuestras nuevas experiencias, con los conocimientos y experiencias almacenados en la memoria.



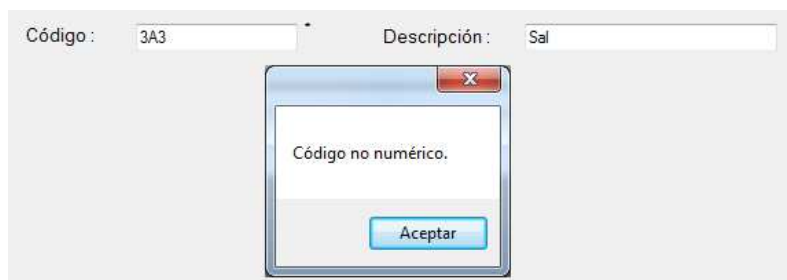
The screenshot shows a window titled 'Conceptos' with a currency icon (€) and a list of concepts. The window has a title bar with standard OS controls. Below the title bar, there is a currency icon and the word 'Conceptos'. Below this, there are five buttons: 'Agregar', 'Modificar', 'Borrar', 'Detallar', and 'Listar'. Below the buttons, there are two input fields: 'Código:' and 'Descripción:'. Below the input fields, there is a table with two columns: 'Código' and 'Descripción'. The table contains the following data:

Código	Descripción
101	Caja
102	Bancos
103	Almacén
104	Clientes
105	Documentos por cobrar
106	Deudores Diversos
107	IVA pagado acreditable
201	Terrenos
202	Edificios

At the bottom of the window, there is a footer area with a printer icon and the text 'Usuario : CHRISTIAN'. There is also a small note at the bottom left: '* Código de 3 cifras numéricas'.

Creación de Concepto.

Para la creación de nuevos códigos Concepto, debemos abrir la pestaña Concepto, introducir en el campo código un valor de 3 cifras numéricas y una descripción para dicho código. Una vez hecho esto pulsaremos el botón **Agregar** y pasaremos a la ventana de confirmación. Debemos recordar que al igual que en casos anteriores, nos mostrará un mensaje de error, siempre que cometamos alguna irregularidad. A continuación vemos un ejemplo de mensaje de error:



Modificación de Conceptos.

La modificación de un código de conceptos consiste en cambiar su descripción, ya que los códigos no se pueden modificar. Es un caso similar al visto anteriormente en Diarios, ya que aunque el código varía, como ya dijimos no existe la posibilidad de cambio de un código a otro en esta opción. Importante recordar que el máximo número de letras aceptadas es de 255 para cualquier descripción. Al pulsar el botón **Modificar**, nos saldrá un mensaje indicándonos el código que queremos modificar, la descripción antigua y la nueva, listo para confirmar que ese es el cambio que queremos hacer.

Borrado de Conceptos.

Para borrar un código de Conceptos, escribiremos en el campo correspondiente, el código de tres cifras, siempre las tres numéricas, que deseamos borrar. Este código debe existir previamente en la aplicación, si no recibiremos un mensaje advirtiéndonoslo. En el campo descripción no será necesario poner nada en esta ocasión. Al pulsar el botón **Borrar**, nos mostrará un mensaje indicándonos cuál es el código y la descripción que vamos a eliminar a modo de confirmación. Para que este código pueda ser eliminado, no puede existir ningún apunte o factura con este código ya que si no perderíamos información, por lo que tendremos que ir previamente a la opción de borrado de apuntes/facturas y eliminarlas (Ver Lección 3).

Detallado de Conceptos.

Esta función será idéntica a la explicada en la opción Detallado de Diarios dentro de la Lección 6. Tendremos un botón **Detallar**, el cual nos mostrará en una tabla cada vez que se ha creado, borrado o modificado un código, que como ya sabemos, debe ser de tres números. Además mostrará información del usuario que lo ha hecho y la fecha en la que se realizó dicho cambio.

Listado de Conceptos.

Esta opción viene por defecto al abrir la ventana de conceptos y se actualizará tras agregar o borrar un código. Si después de hacer una consulta de conceptos queremos listar todos los códigos, pulsaremos el botón **Listar** y veremos todos los códigos que tenemos en ese momento de conceptos.

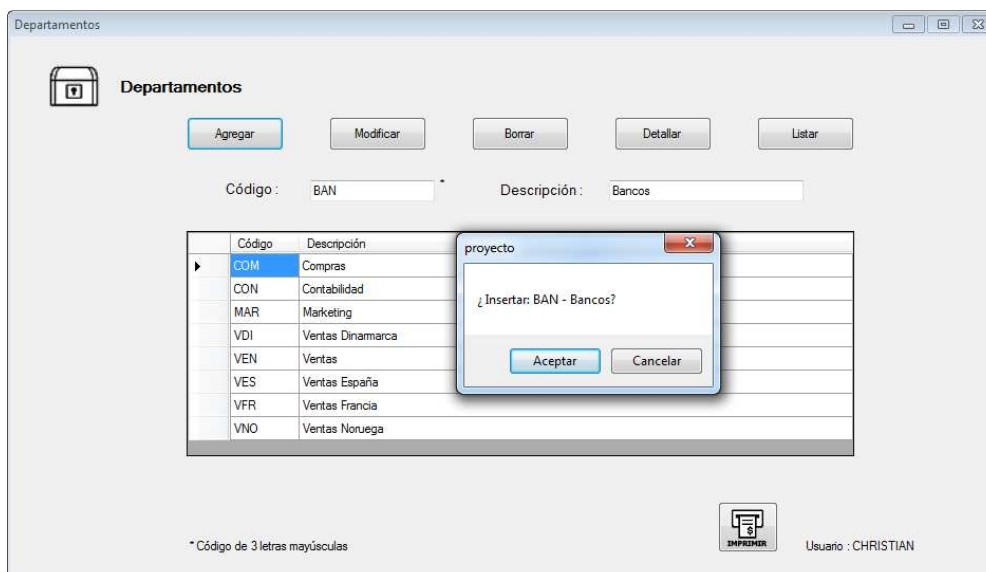
LECCIÓN 8: DEPARTAMENTO

La organización de la mayoría de las empresas se basa en la existencia de **Departamentos**. Éstos pueden ser creados según diferentes criterios:

- **Funcional**, cuando se establecen según las funciones que se vayan a realizar.
- **Territorial**, cuando se establecen según las áreas geográficas en las que actúa la empresa.
- **Por productos o servicios**, cuando se establecen según los distintos productos o servicios que ofrece la empresa.
- **Por procesos de fabricación**, cuando se establecen según las diferentes etapas o procesos de fabricación de los productos.

Creación de Departamentos.

La función de Agregar permite añadir nuevos códigos de Departamentos. Para agregar un nuevo código de Diario, tendremos que poner en el campo Código uno válido, que en este caso será el formado por tres letras mayúsculas. Todo lo que no sea esto dará su respectivo mensaje de error, indicando en que nos hemos equivocado. A su vez, también tendremos que asignarle una descripción a esa nueva entrada. Como el código es único y representativo, si no estaba ya anteriormente en la base de datos, se introducirá correctamente tras pasar una pantalla de confirmación.



Modificación de Departamentos.

La modificación de departamentos es sencillamente la misma operación que en otros apartados. (Véase Modificación de Diarios en la Lección 6). Volveremos a introducir en el campo Código, un código válido y existente, es decir un código de tres letras mayúsculas. Y en descripción la nueva, por la cual queremos cambiar la ya existente. Pulsaremos el botón modificar y confirmaremos la operación.

Borrado de Departamentos.

Para borrar un código de Departamentos, escribiremos en el campo correspondiente, el código que deseemos borrar. Este código debe existir previamente en la aplicación, si no recibiremos un mensaje advirtiéndonoslo. En el campo descripción no será necesario poner nada en esta ocasión. Al pulsar el botón **Borrar**, nos mostrará un mensaje indicándonos cuál es el código y la descripción que vamos a eliminar a modo de confirmación. Para que este código pueda ser eliminado no puede existir ningún apunte o factura con este código ya que si no perderíamos información, por lo que tendremos que ir previamente a la opción de borrado de apuntes/facturas y eliminarlas (Ver Lección 3).

Detallado de Departamentos.

Esta opción nos da la posibilidad de detallar el historial de los códigos de departamentos. Consiste en mostrar todos los cambios que se han realizado en un código determinado. Esta opción es accesible según los privilegios que tenga el usuario sobre la aplicación. En caso de tener dichos privilegios, al ingresar un código de departamento válido en el campo correspondiente, podremos pulsar el botón **Detallar**, y nos mostrará en la tabla cada vez que se ha creado, borrado o modificado dicho código, añadiendo el usuario que lo ha hecho y la fecha en la que se realizó dicho cambio.

Listado de Departamentos.

Esta opción viene por defecto al abrir cualquiera de las ventanas de maestros de contabilidad (cuentas, subcuentas, diarios y conceptos) y también cualquiera de las de contabilidad analítica (departamentos, centros de coste y proyectos).

Esto significa que la tabla se actualizará tras agregar, modificar o borrar un código. Si después de hacer una consulta de departamentos, queremos listar todos los códigos, pulsaremos el botón **Listar** y veremos todos los códigos que tenemos en ese momento de departamentos.

	Código	Descripción
▶	COM	Compras
	CON	Contabilidad
	MAR	Marketing
	VDI	Ventas Dinamarca
	VEN	Ventas
	VES	Ventas España
	VFR	Ventas Francia
	VNO	Ventas Noruega

LECCIÓN 9: CENTROS DE COSTE

Los Centros de Costes sirven a la administración para tener un mejor control presupuestal, al medir a cada departamento o línea de negocio, poder obtener un estado de resultados por centro de negocios y medir en forma aislada cada unidad de negocio

Creación de Centros de Coste.

La función de **Agregar** permite agregar nuevos códigos de Centros de Coste. Para agregar un nuevo código de Centros de Coste, tendremos que poner en el campo Código uno válido, que en este caso será el formado por tres cifras numéricas. Todo lo que no sea esto dará su respectivo mensaje de error, indicando en qué nos hemos equivocado. A su vez también tendremos que asignarle una descripción a esa nueva entrada. Como el código es único y representativo, si no estaba ya anteriormente en la base de datos se introducirá correctamente tras pasar una pantalla de confirmación.

Código	Descripción
100	Málaga
101	Málaga - Fábrica
102	Málaga - Oficina
120	Grenada
250	Madrid
300	Copenhague
500	Londres
600	Oslo

* Código de 3 cifras numéricas

Usuario : CHRISTIAN

Modificación de Centros de Coste.

Para modificar un código de Centro de Coste seguiremos los mismos pasos que con los de Conceptos (Véase Modificación de Conceptos en la Lección 7). Introducir en el campo código el que queramos modificar y en el de descripción la nueva.

Borrado de Centros de Coste.

Para borrar un código de Centro de Coste, escribiremos en el campo correspondiente, el código que deseemos borrar. Este código debe existir previamente en la aplicación, si no recibiremos un mensaje advirtiéndonoslo. En el campo descripción no será necesario poner nada en esta ocasión. Al pulsar el botón **Borrar**, nos mostrará un mensaje indicándonos cuál es el código y la descripción que vamos a eliminar a modo de confirmación. Para que este código pueda ser eliminado no puede existir ningún apunte o factura con este código ya que si no perderíamos información, por lo que tendremos que ir previamente a la opción de borrado de apuntes/facturas y eliminarlas (Ver Lección 3).

Detallado de Centros de Coste.

Esta opción nos dá la posibilidad de detallar el historial de los códigos de Centros de Coste. Consiste en mostrar todos los cambios que se han realizado en un código determinado. Esta opción es accesible según los privilegios que tenga el usuario sobre la aplicación. En caso de tener dichos privilegios, al ingresar un código de Centro de Coste válido en el campo correspondiente, podremos pulsar el botón **Detallar**, y nos mostrará en la tabla cada vez que se ha creado, borrado o modificado dicho código, añadiendo el usuario que lo ha hecho y la fecha en la que se realizó dicho cambio.

Listado de Centros de Coste.

Esta opción viene por defecto al abrir cualquiera de las ventanas de maestros de contabilidad (cuentas, subcuentas, diarios y conceptos) y también cualquiera de las de contabilidad analítica (departamentos, centros de coste y proyectos).

Esto significa que la tabla se actualizará tras agregar, modificar o borrar un código. Si después de hacer una consulta de Centro de Coste queremos listar todos los códigos, pulsaremos el botón **Listar** y veremos todos los códigos que tenemos en ese momento de Centro de Coste.

	Código	Descripción
▶	100	Málaga
	101	Málaga - Fábrica
	102	Málaga - Oficina
	120	Granada
	250	Madrid
	300	Copenhague
	500	Londres
	600	Oslo

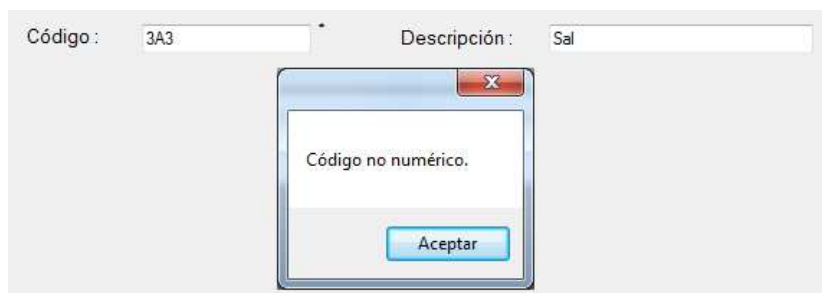
LECCIÓN 10: PROYECTOS

El **Proyecto** representa una subdivisión o información más detallada del motivo o concepto del apunte contable, por ejemplo compramos una mesa (inmovilizado material) y se afecta al proyecto reforma oficina producción.

De esta forma podremos filtrar la información contable por subcuentas y si queremos más detalle o subdivisión por proyectos, y de este modo ayudará a obtener una contabilidad analítica más detallada.

Creación de Proyectos.

Para la creación de nuevos códigos Proyectos debemos abrir la pestaña Proyectos, introducir en el campo código un valor de 3 cifras numéricas y una descripción para dicho código. Una vez hecho esto pulsaremos el botón **Agregar** y pasaremos la ventana de confirmación. Debemos recordar que al igual que en casos anteriores, nos mostrará un mensaje de error siempre que cometamos alguna irregularidad. A continuación vemos un ejemplo de mensaje de error:



Modificación de Proyectos.

La modificación de un código de Proyectos consiste en cambiar su descripción, ya que los códigos no se pueden modificar. Es un caso similar al visto anteriormente en Diarios, ya que aunque el código varía, como ya dijimos no existe la posibilidad de cambio de un código a otro en esta opción. Importante recordar que el máximo número de letras aceptadas es de 255 para cualquier descripción. Al pulsar el botón **Modificar**, nos saldrá un mensaje indicándonos el código que queremos modificar, la descripción antigua y la nueva, listo para confirmar que ese es el cambio que queremos hacer.

Borrado de Proyectos.

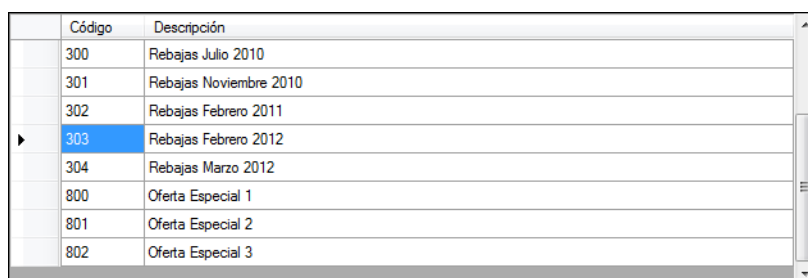
Para borrar un código de Proyectos, escribiremos en el campo correspondiente, el código de tres cifras, siempre las tres numéricas, que deseemos borrar. Este código debe existir previamente en la aplicación, si no recibiremos un mensaje advirtiéndonoslo. En el campo descripción no será necesario poner nada en esta ocasión. Al pulsar el botón **Borrar**, nos mostrará un mensaje indicándonos cuál es el código y la descripción que vamos a eliminar a modo de confirmación. Para que este código pueda ser eliminado, no puede existir ningún apunte o factura con este código ya que si no perderíamos información, por lo que tendremos que ir previamente a la opción de borrado de apuntes/facturas y eliminarlas (Ver Lección 3).

Detallado de Proyectos.

Esta función será idéntica a la explicada en la opción Detallado de Diarios dentro de la Lección 6. Tendremos un botón **Detallar**, el cual nos mostrará en una tabla cada vez que se ha creado, borrado o modificado un código, que como ya sabemos debe ser de tres números. Además mostrará información del usuario que lo ha hecho y la fecha en la que se realizó dicho cambio.

Listado de Proyectos.

Esta opción viene por defecto al abrir la ventana de Proyectos y se actualizará tras agregar o borrar un código. Si después de hacer una consulta de conceptos queremos listar todos los códigos, pulsaremos el botón **Listar** y veremos todos los códigos que tenemos en ese momento de Proyectos.



	Código	Descripción
	300	Rebajas Julio 2010
	301	Rebajas Noviembre 2010
	302	Rebajas Febrero 2011
▶	303	Rebajas Febrero 2012
	304	Rebajas Marzo 2012
	800	Oferta Especial 1
	801	Oferta Especial 2
	802	Oferta Especial 3

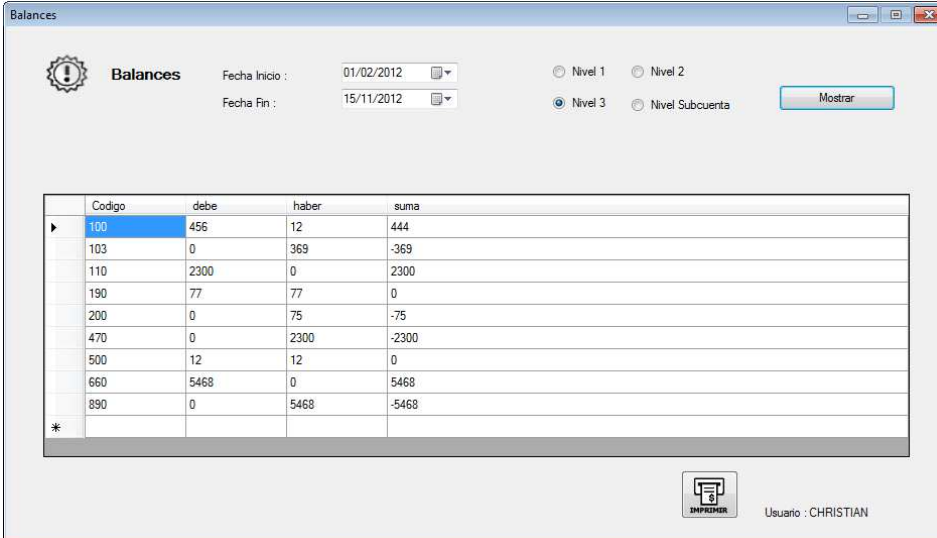
LECCIÓN 11: BALANCES

El estado de situación patrimonial, también llamado **Balance general** o **Balance de situación**, es un informe financiero o estado contable que refleja la situación del patrimonio de una empresa en un momento determinado.

El estado de situación financiera se estructura a través de tres conceptos patrimoniales, el activo, el pasivo y el patrimonio neto, desarrollados cada uno de ellos en grupos de cuentas que representan los diferentes elementos patrimoniales.

Consultas de Balances

Las consultas de balances nos mostrarán por medio de la tabla hasta cuatro distintas columnas. Estas son, en primer lugar, el código, que como veremos más abajo tendrá cuatro niveles distintos. En segundo y tercer lugar, la suma de los debe y haber por separado para ese nivel de subcuenta en concreto. Y en cuarto lugar, la resta entre la suma total de los debe y la suma total de los haber. Todo esto irá marcado por un intervalo de tiempo que marcaremos en los dos calendarios de la parte superior izquierda.



The screenshot shows a software window titled 'Balances'. It includes a search icon, the title 'Balances', and date selection fields for 'Fecha Inicio' (01/02/2012) and 'Fecha Fin' (15/11/2012). There are radio buttons for 'Nivel 1', 'Nivel 2', 'Nivel 3' (selected), and 'Nivel Subcuenta'. A 'Mostrar' button is present. Below is a table with columns: 'Codigo', 'debe', 'haber', and 'suma'. The table lists various account codes and their corresponding debit, credit, and balance values. A '*' symbol is at the bottom of the table. At the bottom right, there is a printer icon and the text 'Usuario : CHRISTIAN'.

Codigo	debe	haber	suma
100	456	12	444
103	0	369	-369
110	2300	0	2300
190	77	77	0
200	0	75	-75
470	0	2300	-2300
500	12	12	0
660	5468	0	5468
890	0	5468	-5468
*			

Los criterios de búsqueda serán cuatro niveles distintos:

- El **Nivel 1** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, quedándonos con su primera cifra únicamente, con lo cual tendremos un máximo de diez filas, desde 0 al 9.
- El **Nivel 2** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, quedándonos con sus dos primeras cifras, con lo cual tendremos un máximo de cien filas, desde 0 al 99.
- El **Nivel 3** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, quedándonos con sus tres primeras cifras, con lo cual tendremos un máximo de mil filas, desde 0 al 900.
- El **Nivel Subcuenta** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, con lo cual tendremos una fila por cada código de subcuenta existente, es decir, un máximo de cien millones de filas, desde el 0 al 99.999.999.

Como último detalle mencionar que volvemos a disponer con la opción de imprimir la tabla. Además de la tabla es posible borrar las filas que no nos interesen imprimir. Para borrar, simplemente pulsamos la fila que queramos eliminar y pulsamos la tecla de nuestro teclado “Supr”. Estas modificaciones no afectan para nada a los datos de la Base de Datos y si volvemos a consultar los mismos datos, veremos cómo vuelven a su estado inicial.

LECCIÓN 12: PRESUPUESTOS

Se le llama **Presupuesto** al cálculo anticipado de los ingresos y gastos de una actividad económica (personal, familiar, un negocio, una empresa, una oficina, un gobierno) durante un período, por lo general en forma anual. Es un plan de acción dirigido a cumplir una meta prevista, expresada en valores y términos financieros que, debe cumplirse en determinado tiempo y bajo ciertas condiciones previstas. Este concepto se aplica a cada centro de responsabilidad de la organización. El presupuesto es el instrumento de desarrollo anual de las empresas o instituciones cuyos planes y programas se formulan por término de un año.

Agregar Presupuestos

Para añadir un nuevo Presupuesto, deberemos rellenar los campos que tenemos junto al botón **Añadir**. Tenemos un campo para la subcuenta, la cual debe ser un código de ocho cifras, dos campos para el debe y el haber y un campo Fecha que será exclusivamente de meses y años. Aunque elijamos un día distinto del día 1, automáticamente el programa tomará el presupuesto como ese día, para que sea un presupuesto distinto por cada mes y subcuenta.

Subcuenta : Debe : Haber : Fecha Fin :

Consultas de Presupuestos

En esta opción podremos consultar los Presupuestos. Los Presupuestos constan de dos grupos de valores y una columna de diferencia entre ambas. Además será posible mirarlo por 4 distintos niveles. Las fechas, como ya mencionamos en otras lecciones, serán desde el comienzo de los días marcados, es decir, para consultar el día de hoy, marcaremos fecha inicio de hoy, y fecha final mañana. Esto lo conseguiremos fácilmente con los 2 calendarios desplegables de la parte superior izquierda de la ventana.

Presupuestos

Fecha Inicio : 08/01/2012 Fecha Fin : 08/12/2012

Nivel 1 Nivel 2 Nivel 3 ☒ Nivel Subcuenta

Mostrar

Subcuenta : Debe : 0,00 Haber : 0,00 Fecha Fin : 12/2012

Añadir

Código	Descripción	Debe	Haber	Saldo	DebePre	HaberPre	SaldoPre	Diferencia
10000001	Capital Social	0,00	397,32	-397,32	50,00	605,00	-555,00	157,68
11800002	Resultado Negati...	280,20	0,00	280,20	151,80	0,00	151,80	128,40
12100001	Pérdidas y Gana...	22447,60	508,30	21939,30	25050,50	33989,00	-8938,50	30877,80
13000001	Impuesto Diferido	0,00	194,20	-194,20	0,00	80,00	-80,00	-114,20
20600002	Software	1190,89	2346,70	-1155,81				-1155,81
21100001	Construcciones				375,00	0,00	375,00	-375,00
21100003	Construcción	21732,60	0,00	21732,60	250,00	250,00	0,00	21732,60
21300005	Maquinarias				250,00	250,00	0,00	0,00
21300011	Maquinaria	997,10	342,62	654,48	250,00	250,00	0,00	654,48
28130003	Amort. Maquinaria				560,00	780,00	-220,00	220,00
31000001	Materias Primas	46,44	467,52	-421,08	1000,00	2300,00	-1300,00	878,92

IMPRIMIR

Usuario : CHRISTIAN

Los criterios de búsqueda serán cuatro niveles distintos:

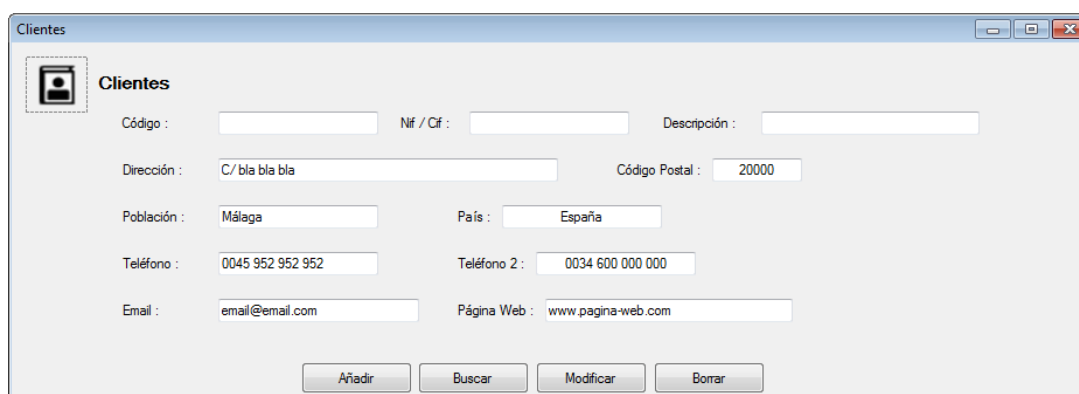
- El **Nivel 1** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, quedándonos con su primera cifra únicamente, con lo cual tendremos un máximo de diez filas, desde 0 al 9.
- El **Nivel 2** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, quedándonos con sus dos primeras cifras, con lo cual tendremos un máximo de cien filas, desde 0 al 99.
- El **Nivel 3** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, quedándonos con sus tres primeras cifras, con lo cual tendremos un máximo de mil filas, desde 0 al 900.
- El **Nivel Subcuenta** consiste en agrupar todos los apuntes/facturas por el código de subcuenta, con lo cual tendremos una fila por cada código de subcuenta existente, es decir, un máximo de cien millones de filas, desde el 0 al 99.999.999.

Una vez elegido el nivel, tenemos nueve columnas. La primera indica el código, que como ya hemos dicho, dependerá del nivel exigido. La segunda indica la descripción del código, que variará según el nivel. Las tres siguientes serán idénticas a las vistas en Balances (ver Lección 11: Balances). A continuación, tendremos otras tres, serán los presupuestos que hayamos metidos para ese código y esa fecha. Recordar que las fechas en presupuestos siempre son el día 1 de cada mes. Y la última es la diferencia entre la suma de los balances y la suma de los presupuestos.

Como último detalle, mencionar que volvemos a disponer con la opción de imprimir la tabla. Además de la tabla es posible borrar las filas que no nos interesen imprimir. Para borrar simplemente pulsamos la fila que queramos eliminar y pulsamos la tecla de nuestro teclado “Supr”. Estas modificaciones no afectan para nada a los datos de la Base de Datos y si volvemos a consultar los mismos datos, veremos cómo vuelven a su estado inicial.

LECCIÓN 13: CLIENTES

En economía, **cliente** es quien accede a un producto o servicio por medio de una transacción financiera (dinero) u otro medio de pago. Quien compra, es el comprador, y quien consume el consumidor. Normalmente, cliente, comprador y consumidor son la misma persona. El término opuesto al de "cliente" es el de "competidor".



The screenshot shows a window titled 'Clientes' with a form for adding or editing client information. The form includes the following fields:

- Código :** [Empty text box]
- Nif / Cif :** [Empty text box]
- Descripción :** [Empty text box]
- Dirección :** [Text box containing 'C/ bla bla bla']
- Código Postal :** [Text box containing '20000']
- Población :** [Text box containing 'Málaga']
- País :** [Text box containing 'España']
- Teléfono :** [Text box containing '0045 952 952 952']
- Teléfono 2 :** [Text box containing '0034 600 000 000']
- Email :** [Text box containing 'email@email.com']
- Página Web :** [Text box containing 'www.pagina-web.com']

At the bottom of the form are four buttons: **Añadir**, **Buscar**, **Modificar**, and **Borrar**.

Creación de Clientes.

Para crear nuevos clientes deberemos rellenar todos los campos que se ven en la pantalla. Y tras esto pulsar el botón **Añadir**. En caso de fallo en alguno de ellos, nos lanzará un mensaje advirtiéndonoslo. Veamos cuales son uno por uno:

Los dos primeros son los campos claves, **Código** y **Nif/Cif**. Son campos claves ya que ninguno de los dos puede ser repetido y cada empresa tendrá el suyo. En ambos casos el número máximo será de 20 caracteres y en el caso de Código, además, estos caracteres serán exclusivamente numéricos. En el código saldrá automáticamente uno nuevo posterior al último guardado en la base de datos, para empezar a crear el cliente con ese código.

En el campo **Descripción** introduciremos el nombre del cliente o empresa y tendremos un máximo de 255 caracteres. Como ya mencionamos anteriormente, en caso de excedernos recibiremos un mensaje advirtiéndonoslo. El campo **Dirección** será idéntico al anterior, para añadir la dirección física del cliente.

El campo **Postal** será más corto que los demás y solo admitirá 10 cifras numéricas, mientras que **Población** y **País** tendrán un máximo de 30 caracteres. Evidentemente todos estos datos en caso de error podrán ser modificados. Puede verse en el apartado Modificar clientes dentro de esta misma Lección.

Los dos números de **Teléfonos** tendrán un máximo de 30 cifras y podrán ser escritos del modo que se prefiera, por ejemplo escribiendo prefijo, espacio y luego el resto, o bien de tres en tres cifras.

Y los dos últimos campos serán el **Email** y la **Web** del cliente, los cuales se podrán escribir hasta 255 caracteres.

Ya solo nos quedará pulsar el botón **Añadir** y, si no hemos cometido ningún error, el nuevo cliente se agregará a nuestra Base de Datos.

Búsqueda de Clientes.

Para poder modificar un cliente, previamente podremos hacer uso de la opción **Rellenar**, en la que dado un código de clientes o su Nif, nos rellenará todos los campos. El criterio de búsqueda es primero por código y en caso de estar vacío, por Nif.

Modificación de Clientes.

Para modificar un cliente, la mejor opción es primero buscarlo, usando lo aprendido anteriormente en Buscar clientes. Una vez relleno todos los campos, tan solo deberemos cambiar aquellos datos que queramos, recordando que ni el código de Clientes ni el Nif son modificables. Además tenemos que tener en cuenta las limitaciones de cada dato, ya sea en forma de longitud o de formato. Para más información ver Creación de Clientes. Una vez terminado, pulsaremos el botón **Modificar**.

Borrado de Clientes.

La opción de **Borrar** será similar a las dos anteriores en cuanto a criterio de búsqueda, ya que tendrá prioridad el campo código y en caso de estar vacío, el campo Nif. Si existe anteriormente lo que buscamos, nos lanzará un mensaje de confirmación de borrado.

LECCIÓN 14: USUARIOS

Un **Usuario** es un individuo que utiliza una computadora, sistema operativo, servicio o cualquier sistema informático. Por lo general es una única persona.

Un usuario generalmente se identifica frente al sistema o servicio utilizando un nombre de usuario (nick) y a veces una contraseña, este tipo es llamado usuario registrado. Por lo general un usuario se asocia a una única cuenta de usuario, en cambio, una persona puede llegar a tener múltiples cuentas en un mismo sistema o servicio (si eso está permitido).

The screenshot shows a window titled 'Usuarios'. Inside, there's a section 'Nuevo Usuario' with fields for 'Nombre', 'Contraseña', and 'Privilegios' (set to 'Prueba'). A 'Crear' button is below. Below that is the 'Nuevos Privilegios' section with a 'Descripción' field and a grid of checkboxes for various permissions. At the bottom right, it says 'Usuario : CHRISTIAN'.

Nuevos Privilegios		
Descripción : <input type="text"/>		
Añadir Apuntes / Facturas : <input checked="" type="checkbox"/>	Añadir Listar Contabilidad / Analítica : <input checked="" type="checkbox"/>	Clientes : <input type="checkbox"/>
Consultar Apuntes / Facturas : <input checked="" type="checkbox"/>	Modificar y Borrar Contabilidad / Analítica : <input checked="" type="checkbox"/>	Usuarios : <input type="checkbox"/>
Edición Apuntes / Facturas : <input type="checkbox"/>	Detallar Contabilidad / Analítica : <input checked="" type="checkbox"/>	
Consultar Balances : <input type="checkbox"/>	Consultar Presupuestos : <input type="checkbox"/>	

Un usuario registrado accede a un servicio a través de un login, luego de su autenticación.

Creación de Usuarios.

En este apartado veremos cómo crear nuevos usuarios, para que puedan usar el programa. El usuario, con sus nuevos datos, logeará al principio del mismo, y según los privilegios que se le hayan dado, podrá acceder a unas opciones u otras.

This is a close-up of the 'Nuevo Usuario' form. It includes the 'Nombre', 'Contraseña', and 'Privilegios' fields, and the 'Crear' button.

En la parte superior de la ventana vemos las tres opciones existentes. El nombre, que deberá ser único, por tanto, en caso de intentar crear uno que ya exista, lanzará un mensaje de error advirtiéndonoslo. La contraseña que será de tipo numérica entre seis y diez cifras. Y los privilegios, en el cual tendremos que elegir entre los ya creados. Cómo crear un tipo de privilegios nuevo lo veremos en el siguiente apartado dentro de esta misma lección.

Al seleccionar un tipo de privilegio, automáticamente, en la parte inferior, se marcarán los campos a los que corresponde, para facilitar saber de qué tipo se trata. Una vez rellenado los tres campos obligatorios, pulsaremos el botón **Crear**, y si no hemos cometido ningún error, se creará el nuevo usuario.

Creación de Privilegios.

Dentro de la creación de Privilegios tenemos en primer lugar un campo para darle un nombre y luego una serie de opciones que podremos marcar para dar o quitar privilegios.

Como ya hemos dicho en el campo **Descripción** indicaremos un nombre nuevo y único para el privilegio, que como ya vimos en el apartado anterior, saldrá en el menú desplegable a la hora de crear nuevos usuarios.



Nuevos Privilegios

Descripción :

Añadir Apuntes / Facturas :	<input checked="" type="checkbox"/>	Añadir y Listar Contabilidad / Analítica :	<input checked="" type="checkbox"/>	Cientes :	<input type="checkbox"/>
Consultar Apuntes / Facturas :	<input checked="" type="checkbox"/>	Modificar y Borrar Contabilidad / Analítica :	<input checked="" type="checkbox"/>	Usuarios :	<input type="checkbox"/>
Edición Apuntes / Facturas :	<input type="checkbox"/>	Detallar Contabilidad / Analítica :	<input checked="" type="checkbox"/>		
Consultar Balances :	<input type="checkbox"/>	Consultar Presupuestos :	<input type="checkbox"/>		

A continuación, los distintos privilegios son:

- **Añadir Apuntes / Facturas:** Gracias a esto podremos acceder a la primera pestaña del Menú principal.
- **Consultar Apuntes / Facturas:** En este caso, nos dará acceso a la siguiente pestaña, en la que podremos consultar, modificar y borrar apuntes y/o facturas introducidas previamente.
- **Edición Apuntes / Facturas:** Permite dentro de la ventana de Consultas, borrar facturas. Además, también estará activa la opción Modificar para abrir una nueva ventana donde cambiarlas.
- **Añadir y Listar Contabilidad / Analítica:** Permite añadir nuevos códigos en los maestros de Contabilidad (Cuentas, Subcuentas, Diarios y Conceptos) y en la Contabilidad Analítica (Departamentos, Centros de Coste y Proyectos). Además podremos listar los códigos que haya actualmente en la base de datos.
- **Modificar y Borrar Contabilidad / Analítica:** Permite modificar y borrar códigos en los maestros de Contabilidad (Cuentas, Subcuentas, Diarios y Conceptos) y en la Contabilidad Analítica (Departamentos, Centros de Coste y Proyectos).
- **Detallar Contabilidad / Analítica:** Permite consultar el historial de cualquier código de maestros de Contabilidad y en la Contabilidad Analítica.
- **Consulta Balances:** Permite ver los balances dentro del intervalo de fecha que indiquemos y por niveles.
- **Consulta Presupuestos:** Permite ver los balances, los presupuestos y la diferencia entre ambos dentro del intervalo de fecha que indiquemos y por niveles.
- **Clientes:** Crear, modificar y borrar clientes.
- **Usuarios:** Acceso a esta misma pantalla para crear nuevos usuarios y privilegios

Una vez elegido los privilegios pulsaremos el botón **Crear** y se creará el nuevo privilegio y estará disponible a la hora de crear usuarios.

8. Apéndice

Contabilidad / Programa de contabilidad:

Es la ciencia social que se encarga de estudiar, medir y analizar el patrimonio de las organizaciones, empresas e individuos, con el fin de servir en la toma de decisiones y control, presentando la información, previamente registrada, de manera sistemática y útil para las distintas partes interesadas. Posee además una técnica que produce sistemáticamente y estructuradamente información valiosa cuantitativa (realización) y cualitativa (revelación), expresada en unidades monetarias acerca de las transacciones que efectúan las entidades económicas y de ciertos eventos económicos identificables y cuantificables que la afectan, con la finalidad de facilitarla a los diversos públicos interesados. La finalidad de la contabilidad es suministrar información en un momento dado y de los resultados obtenidos durante un período de tiempo, que resulta de utilidad a los usuarios en la toma de sus decisiones, tanto para el control de la gestión pasada, como para las estimaciones de los resultados futuros, dotando tales decisiones de racionalidad y eficiencia.

Por otro lado, llamamos **software contable** a los programas de contabilidad o paquetes contables, destinados a sistematizar y simplificar las tareas de contabilidad. El software contable registra y procesa las transacciones históricas que se generan en una empresa o actividad productiva: las funciones de compras, ventas, cuentas por cobrar, cuentas por pagar, control de inventarios, balances, producción de artículos, nóminas, etc. Para ello solo hay que ingresar la información requerida, como las pólizas contables, ingresos y egresos, y hacer que el programa realice los cálculos necesarios.

Estas funciones pueden ser desarrolladas internamente por la compañía o la organización que lo está utilizando o puede ser adquirido a un tercero, existiendo también una combinación de ambas alternativas, es decir, un paquete de software desarrollado por un tercero con modificaciones locales.

Apunte:

Se denomina asiento contable o apunte contable a cada una de las anotaciones o registros que se hacen en un libro o diario de contabilidad, que se realizan con la finalidad de registrar un hecho económico, que provoca una modificación en el patrimonio de una empresa, lo que conlleva a un movimiento en las cuentas de una empresa, usando una partida del Debe y otra del Haber.

Factura:

La factura, factura de compra o factura comercial es un documento mercantil que refleja toda la información de una operación de compraventa. La información fundamental que aparece en una factura debe reflejar la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar.

Además, en la factura deben aparecer los datos del expedidor y del destinatario, el detalle de los productos y servicios suministrados, los precios unitarios, los precios totales, los descuentos y los impuestos.

La factura se considera como el justificante fiscal de la entrega de un producto o de la provisión de un servicio, que afecta al obligado tributario emisor (el vendedor) y al obligado tributario receptor (el comprador). La factura es el único justificante fiscal, que da al receptor el derecho de deducción del impuesto (IVA). Esto no se aplica en los documentos sustitutivos de factura, recibos o tickets.

Maestro de Contabilidad:

Bajo ese epígrafe se registran todos los conceptos posibles o a utilizar en todos los documentos que se realizan en los distintos módulos.

El objetivo de estos conceptos y la digitación en pantalla es registrar todos los conceptos posibles para cada caso de realización de transacciones en todos los módulos, de forma que todas las transacciones estén enlazadas y definidas en base a lo que se trata.

Ejemplo de conceptos: Ingresos por venta, Cobros por misceláneos, Pago a suplidores, Pago energía eléctrica, Descuento en compra, Devolución de mercancía, etc.

A cada concepto se le asigna la cuenta contable en la que debe reflejarse, forma de automatizar, los balances del catálogo contable.

También son asociados a las entidades y documentos de los distintos módulos que aplica cada concepto.

Por lo tanto, en nuestro Maestro de Contabilidad o de Conceptos, se deberá de identificar, numerar y definir los conceptos y subpartidas de nuestro plan contable y contabilidad.

Cuentas:

La cuenta es el elemento básico y central en la contabilidad y en los servicios de pagos. La cuenta contable es la representación valorada en unidades monetarias, de cada uno de los elementos que componen el patrimonio de una empresa (bienes, derechos y obligaciones) y del resultado de la misma (ingresos y gastos). Permite el seguimiento de la evolución de los elementos en el tiempo. Por lo tanto hay tantas cuentas como elementos patrimoniales tenga la empresa. En consecuencia, el conjunto de cuentas de una empresa supone una representación completa del patrimonio y del resultado (beneficios o pérdidas) de la empresa.

Las transacciones económicas se recogen en la contabilidad a través de las variaciones en el valor de las distintas cuentas, facilitando el registro de las operaciones contables en los libros de contabilidad. Cada cuenta se configura por un título que hace referencia al elemento que representa, un código numérico que la identifica y un valor de la misma.

- Clasificación de las cuentas contables:

Las cuentas contables se clasifican por niveles (grupos, subgrupos, cuentas y subcuentas).

Los grupos constan de un dígito (nivel 1), los subgrupos de primer nivel, de dos dígitos (nivel 2), los subgrupos de tercer nivel, de tres dígitos (nivel 3) siendo a partir de dicho nivel donde se suelen encuadrar las cuentas, esto es, con más de cuatro dígitos, si bien es muy habitual la utilización de 8 dígitos para una cuenta en particular, por ejemplo la cuenta de un cliente o de un proveedor o banco.

El grupo es de nivel 1 (tiene un dígito), porque es el que encabeza y dicta las características que tienen que tener las cuentas que se encuentren clasificadas dentro de él.

1. Financiación Básica
2. Inmovilizado
3. Existencias
4. Acreedores y deudores por operaciones comerciales
5. Cuentas financieras
6. Compras y gastos
7. Ingresos y ventas
8. Gastos imputados al patrimonio neto
9. Ingresos imputados al patrimonio neto

El subgrupo es de nivel 2 (tiene dos dígitos), es la siguiente categoría después del grupo y se encuentra dentro de éste, adaptado a las características que define el grupo pero además hace una clasificación más detallada de las cuentas que deben ir en él. Ejemplo:

- 1 Financiación básica
 - 10 Capital
 - 11 Reservas
 - 12 Resultados pendientes de aplicación
 - 13 Subvenciones, donaciones y legados
 - 14 Provisiones
 - 15 Deudas a largo plazo especiales
 - 16 Deudas a largo plazo con partes vinculantes
 - 17 Deudas a largo plazo por préstamos recibidos y otros conceptos
 - 18 Pasivos por fianzas y garantías y otros conceptos a largo plazo
 - 19 Situaciones transitorias de financiación

Los subgrupos de segundo nivel son de nivel 3 (tiene tres dígitos) y le siguen las cuentas de 4 dígitos en adelante.

Ejemplo:

- 1 Financiación básica
 - 10 Capital
 - 100 Capital social
 - 101 Fondo social
 - 102 Capital
 - 103 Socios por desembolsos no exigidos
 - 1030 Socios por desembolsos no exigidos capital social

Subcuentas:

Forma parte del cuadro de cuentas del Plan General de Contabilidad y constituye el mayor detalle de las cuentas de cada grupo, de forma que cada grupo se divide en subgrupos y éstos en cuentas principales, las cuales se desagregan en las subcuentas, que se identifican con códigos de cuatro cifras.

Diarios:

Es una forma de clasificar cada uno de los apuntes contables que nos permite agrupar por tipo de operación como por ejemplo compras interiores, intracomunitarias, inversiones, etc... El diario nos facilita el obtener los datos necesarios por actividad.

Conceptos:

Los **conceptos** son construcciones o **imágenes mentales**, por medio de las cuales comprendemos las experiencias que emergen de la interacción con nuestro entorno. Estas construcciones surgen por medio de la integración en clases o categorías, que agrupan nuestros nuevos conocimientos y nuestras nuevas experiencias con los conocimientos y experiencias almacenados en la memoria.

Se considera una unidad cognitiva de significado; un contenido mental que a veces se define como una "unidad de conocimiento".

El concepto representa el motivo del asiento o apunte contable, por ejemplo pago Fra. 565 Martínez Lopez,S.A

Contabilidad Analítica:

La Contabilidad analítica también recibe el nombre de contabilidad de costes. Recopila y analiza la información interna sobre costes de la empresa. Suministra información a la contabilidad financiera, y simultáneamente utiliza la información que ésta recoge.

Se basa en datos históricos tanto para la determinación del coste real soportado como para la determinación del coste estándar y de las posibles desviaciones.

Cada empresa y cada situación deberán dotarse de su propia contabilidad analítica. El sistema de costes es específico de cada empresa, de cada producto y de cada momento en concreto.

La evolución de la contabilidad de costes ha llevado a la aparición de la contabilidad de gestión. AECA define la contabilidad de gestión como “una rama de la contabilidad que tiene por objeto la captación, medición y valoración de la circulación interna, así como su racionalización y control, con el fin de suministrar a la organización la información relevante para la toma de decisiones empresariales.” (Documentos AECA serie Contabilidad de Gestión. Documento nº1 pag.23)

Departamentos:

La organización de la mayoría de las empresas se basa en la existencia de **departamentos**. Éstos pueden ser creados según diferentes criterios:

- **Funcional**, cuando se establecen según las funciones que se vayan a realizar.
- **Territorial**, cuando se establecen según las áreas geográficas en las que actúa la empresa.
- **Por productos o servicios**, cuando se establecen según los distintos productos o servicios que ofrece la empresa.
- **Por procesos de fabricación**, cuando se establecen según las diferentes etapas o procesos de fabricación de los productos.

El criterio más utilizado para establecer los departamentos de la empresa es el funcional. Si analizamos la actividad de una empresa, podemos distinguir con claridad los siguientes grupos de **funciones**:

- **Dirección**: gestión de la empresa, planificación, toma de decisiones esenciales.
- **Compras**: aprovisionamiento, compras, almacenamiento de materias primas y de productos fabricados y gestión de los stocks.
- **Producción**: transformación de materias y productos, y fabricación.
- **Ventas**: organización y realización de las ventas, estudio de mercados, estrategias de comercialización y publicidad.
- **Financiación**: tesorería, gestión de los cobros y pagos, movimientos de dinero y relación con entidades bancarias y provisiones de fondos.
- **Recursos humanos**: puestos de trabajo, selección, contratación, formación, elaboración y pago de nóminas, y relación con los trabajadores.
- **Administración**: control de la documentación, archivo, registro y contabilidad de operaciones, declaración y pago de impuestos, elaboración de informes, propuestas de objetivos y relaciones con organismos y entidades.

Centros de Costes:

Los Centros de Costes sirven a la administración para tener un mejor control presupuestal, al medir a cada departamento o línea de negocio, poder obtener un estado de resultados por centro de negocios y medir de forma aislada cada unidad de negocio.

Proyectos:

El proyecto representa una subdivisión o información más detallada del motivo o concepto del apunte contable, por ejemplo compramos una mesa (inmovilizado material) y se afecta al proyecto reforma oficina producción.

De esta forma podremos filtrar la información contable por subcuentas y si queremos más detalle o subdivisión por proyectos, y de este modo ayudará a obtener una contabilidad analítica más detallada.

Balances:

El estado de situación patrimonial, también llamado balance general o balance de situación, es un informe financiero o estado contable que refleja la situación del patrimonio de una empresa en un momento determinado.

El estado de situación financiera se estructura a través de tres conceptos patrimoniales, el activo, el pasivo y el patrimonio neto, desarrollados cada uno de ellos en grupos de cuentas que representan los diferentes elementos patrimoniales.

El activo incluye todas aquellas cuentas que reflejan los valores de los que dispone la entidad. Todos los elementos del activo son susceptibles de traer dinero a la empresa en el futuro, bien sea mediante su uso, su venta o su cambio. Por el contrario, el pasivo muestra todas las obligaciones ciertas del ente y las contingencias que deben registrarse. Estas obligaciones son, naturalmente, económicas: préstamos, compras con pago diferido, etc.

El patrimonio neto es el activo menos el pasivo y representa los aportes de los propietarios o accionistas, más los resultados no distribuidos. El patrimonio neto o capital contable muestra también la capacidad que tiene la empresa de autofinanciarse.

La ecuación básica de la contabilidad relaciona estos tres conceptos:

$$\text{Patrimonio neto} = \text{Activo} - \text{Pasivo}$$

Que dicho de manera sencilla es:

$$\text{Lo que se posee} = \text{Lo que se tiene} - \text{Lo que se debe}$$

Presupuestos:

Se le llama **presupuesto** al cálculo anticipado de los ingresos y gastos de una actividad económica (personal, familiar, un negocio, una empresa, una oficina, un gobierno) durante un período, por lo general en forma anual. Es un plan de acción dirigido a cumplir una meta prevista, expresada en valores y términos financieros que, debe cumplirse en determinado tiempo y bajo ciertas condiciones previstas. Este concepto se aplica a cada centro de responsabilidad de la organización. El presupuesto es el instrumento de desarrollo anual de las empresas o instituciones cuyos planes y programas se formulan por término de un año.

Elaborar un presupuesto permite a las empresas, los gobiernos, las organizaciones privadas o las familias, establecer prioridades y evaluar la consecución de sus objetivos. Para alcanzar estos fines, puede ser necesario incurrir en déficit (que los gastos superen a los ingresos) o, por el contrario, puede ser posible ahorrar, en cuyo caso el presupuesto presentará un superávit (los ingresos superan a los gastos).

Asiento:

Un asiento se realiza cada vez que la empresa contabiliza una entrada contable relacionada con la actividad que realiza.

Un asiento es una anotación en el libro de contabilidad que refleja los movimientos económicos de una persona o institución.

En el sistema de contabilidad de partida doble, crear una entrada en un libro de diario implica dos apuntes (2 asientos) en dos cuentas distintas pero relacionadas entre sí, una en el debe y otra en el haber.

e-conomic es un sistema de contabilidad de partida doble y por tanto crea como mínimo 2 asientos por cada entrada que el usuario introduce en el sistema. Si la operación que estamos realizando tiene IVA entonces se crean 3 asientos:

- Una anotación en la cuenta de IVA
- Una anotación en la cuenta de compras
- Una anotación en la cuenta de proveedor

- **Tipos de asiento:**

Los tipos de asientos contables son los siguientes:

- Asiento de Apertura: A partir del inventario inicial o estado de situación inicial, el asiento de apertura realizado en el libro diario consiste en hacer un cargo a todas las cuentas del activo y un crédito a las del pasivo y patrimonio. Desde el libro diario se pasara las anotaciones al libro mayor, de esta manera quedarán abiertas las cuentas.
- Asiento de Gestión: Registrarán las operaciones que diariamente suceden en las empresas en el desarrollo de su actividad. Tras su registro en el libro diario, estas cuentas serán trasladadas a las cuentas correspondientes al libro mayor.
- Asiento de Ajuste: Estos asientos contables corrigen posibles diferencias entre datos inventariados y saldos contables.
- Asiento de Regulación: Estos asientos contables consisten en concentrar todas las cuentas de ingresos y gastos en el estado de pérdidas y ganancias para obtener el resultado del ejercicio.
- Asiento de Cierre: Con este asiento se finaliza el proceso contable y solo queda por presentar los libros correctamente elaborados.

IVA:

El impuesto sobre el valor agregado o impuesto sobre el valor añadido, es aplicado en muchos países, y generalizado en la Unión Europea. Generalmente se le conoce también por las siglas IVA.

El IVA es un impuesto indirecto sobre el consumo, es decir financiado por el consumidor final. Un impuesto indirecto es el impuesto que no es percibido por el fisco directamente del tributario. El IVA es percibido por el vendedor en el momento de toda transacción comercial (transferencia de bienes o prestación de servicios). Los vendedores intermediarios tienen el derecho a reembolsarse el IVA que han pagado a otros vendedores que los preceden en la cadena de comercialización (crédito fiscal), deduciéndolo del monto de IVA cobrado a sus clientes (débito fiscal), debiendo abonar el saldo al fisco.

Los consumidores finales tienen la obligación de pagar el IVA sin derecho a reembolso, lo que es controlado por el fisco obligando a la empresa a entregar justificantes de venta al consumidor final e integrar copias de éstas a la contabilidad de la empresa.

Debe/Haber:

- **Debe:**

Se llama Debe, al lado izquierdo de la cuenta o asiento, y Cargo a las anotaciones practicadas en el Debe.

En el Debe se anotarán los incrementos de inversiones, o las disminuciones de financiaciones.

- **Haber:**

Se denomina Haber, al lado derecho de la cuenta o asiento, y Abono a las anotaciones practicadas en el Haber.

En el Haber se anotarán las disminuciones de inversiones y los aumentos de financiaciones.

Ciente:

En economía, **cliente** es quien accede a un producto o servicio por medio de una transacción financiera (dinero) u otro medio de pago. Quien compra, es el comprador, y quien consume el consumidor. Normalmente, cliente, comprador y consumidor son la misma persona. El término opuesto al de "cliente" es el de "competidor".

Los **clientes activos** son aquellos que en la actualidad, están realizando compras o que lo hicieron dentro de un periodo corto de tiempo. En cambio, los **clientes inactivos** son aquellos que realizaron su última compra hace bastante tiempo atrás, por tanto, se puede deducir que se pasaron a la competencia, que están insatisfechos con el producto o servicio que recibieron o que ya no necesitan el producto. Esta clasificación es muy útil por dos razones:

- Porque permite identificar a los clientes que en la actualidad están realizando compras y que requieren una atención especial para retenerlos, ya que son los que en la actualidad le generan ingresos económicos a la empresa.

- Para identificar aquellos clientes que por alguna razón ya no le compran a la empresa, y que por tanto, requieren de actividades especiales que permitan identificar las causas de su alejamiento para luego intentar recuperarlos.

9. Bibliografía

- Métrica v3:
 - Métrica. Versión 3 / ; Ministerio de Administraciones Públicas
 - <http://administracionelectronica.gob.es>
 - <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema08.pdf>
 - www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node62.html

- Visual Studio:
 - Introducing .NET 4.0 with Visual Studio 2010 / Alex Mackey
 - Microsoft Visual Studio 2010 Unleashed / Mike Snell, Lars Powers
 - <http://msdn.microsoft.com/es-es/library/default.aspx>

- SQL Server
 - Microsoft SQL Server 2008: manual de referencia / Dusan Petkovic
 - http://www.helpdna.net/sqlserver_faq.htm
 - <http://msdn.microsoft.com/es-es/library/default.aspx>

- Contabilidad
 - Manual práctico de contabilidad de gestión / [Julio A. Ortega Seco, Ángel Alberola López]
 - www.economia.com.mx/la_cuenta_individual_y_sus_subcuentas.htm
 - <http://es.wikipedia.org>

- Recursos Contables:
 - www.plangeneralcontable.com/