

DESCRIPCIÓN DEL ANTEPROYECTO

(Adapta-lo tamaño do documento según sexa necesario. Consulta-lo art. 6 do Regulamento de PFC da ESEI)

TÍTULO: Sistema de Gestión Administrativa para Industrias Areces

Introducción

En la actualidad, toda empresa necesita un software de gestión, pues éste la hace más competitiva en el mercado y más eficiente.

Las ventajas que presenta, tales como el uso racional de los recursos de la empresa, el aumento de la productividad, el ahorro que supone la sencillez y la rapidez de todas las operaciones de la empresa, hacen que cada vez más dueños de PYMES comprendan la necesidad competitiva de un software de gestión efectivo.

El software de gestión está constituido por los programas que facilitan la gestión o realización de ciertas tareas.

Se pretende así desarrollar una herramienta software destinada a una empresa dedicada a la fabricación y distribución de productos químicos de limpieza.

Este software debe administrar todos los aspectos relacionados con la fabricación y la distribución de los productos: gestión de pedidos, administración de clientes y proveedores, ventas, facturas, albaranes, etc.

Objetivos

El objetivo de este proyecto es proporcionar una aplicación de escritorio que se amolde a las necesidades de la empresa, facilitando la gestión de los elementos implicados en ella.

La aplicación contará con un sistema de gestión y validación de usuarios, diferenciándose entre usuario administrador y usuarios regulares.

Existirá un único usuario administrador, que podrá dar de alta nuevas cuentas de usuario y será el encargado de gestionar las cuentas ya existentes.

Todos los usuarios deberán tener un nombre de usuario y una contraseña que servirá para autenticarse en el sistema.

Los usuarios regulares, una vez autenticados en el sistema, podrán acceder a todas las funcionalidades de la herramienta, que quedan definidas a continuación:

- **Sección de proveedores.** Se podrán dar de alta nuevos proveedores, listar los proveedores actuales y controlar la emisión y gestión de pedidos a proveedores.
- **Sección de ventas.** Incluye confección de albaranes y facturación. Permitirá editar el formato de impresión de los mismos, de forma que la empresa pueda crear su propio formato de albaranes y facturas.
- **Sección de compras.** Recepción y control de facturas de proveedores.
- **Sección de pedidos.** Permite la gestión de clientes, pudiendo añadir, eliminar o modificar los datos de cada cliente. Controlará la admisión y gestión de las propuestas de pedidos.
- **Sección de fabricación.** Desde aquí se podrá llevar el control de las materias primas que hay en stock, así como gestionar las fórmulas de los diferentes productos que ofrece la empresa.

Con el desarrollo de este proyecto se pretende facilitar y agilizar la carga administrativa de la empresa, ofreciendo una interfaz de usuario amigable que permita usar de manera sencilla todas las funciones de la aplicación.

Descripción técnica

El proyecto consiste en la realización de una aplicación de escritorio, que funcionará de forma autónoma en el ordenador del cliente, basada en una arquitectura de tres capas.

La arquitectura de tres capas es un estilo de programación, cuyo objetivo primordial es la separación de la capa de presentación, la capa de negocio y la capa de datos. Se ha optado por esta arquitectura por las ventajas que ofrece esta descomposición en capas:

- Aplicaciones más robustas gracias al encapsulamiento, que permite aumentar la cohesión de los componentes del sistema.
- En caso de que sobrevenga algún cambio, sólo se modifica la capa involucrada sin tener que modificar el resto de capas.
- Mantenimiento y soporte más eficiente, pues es más sencillo cambiar un componente que modificar una aplicación monolítica.
- Mayor flexibilidad puesto que se pueden añadir nuevos módulos para dotar al sistema de nuevas funcionalidades.

La **capa de presentación** se encarga de presentar el sistema al usuario, comunica la información y recoge las peticiones del usuario. Para la implementación de esta capa se empleará el propio Visual Studio 2008 C# Express Edition, entorno de desarrollo que integra herramientas para la creación de interfaces de usuario. Un ejemplo de las herramientas que Microsoft ofrece para el desarrollo de interfaces de usuario son Windows Forms (API gráfica incluida con .NET Framework) y Windows Presentation Foundation (tecnología que permite el desarrollo de interfaces en Windows y que ofrece una amplia infraestructura y potencia gráfica).

En la **capa de negocio** es donde se reciben las peticiones del usuario y se envían las respuestas tras ser procesadas. Se comunica con la capa de presentación para recibir peticiones y mostrar resultados, y con la capa de datos, para almacenar o recuperar datos a través del sistema gestor de base de datos. Se ha escogido C# como lenguaje de programación debido a la experiencia previa de la que dispongo en el desarrollo de aplicaciones con esta tecnología y con el entorno Visual Studio.

La **capa de datos** es la encargada de acceder a los mismos. Está formada por un sistema gestor de bases de datos, que se encarga del almacenamiento de datos y recibe solicitudes de recuperación o almacenamiento de los mismos desde la capa de negocio. Para la aplicación se usará el sistema gestor de bases de datos SQL Server, pues satisface los requisitos de uso de la aplicación en el entorno de producción. Además, el entorno de desarrollo Visual Studio facilita el diseño e integración de bases de datos SQL Server.

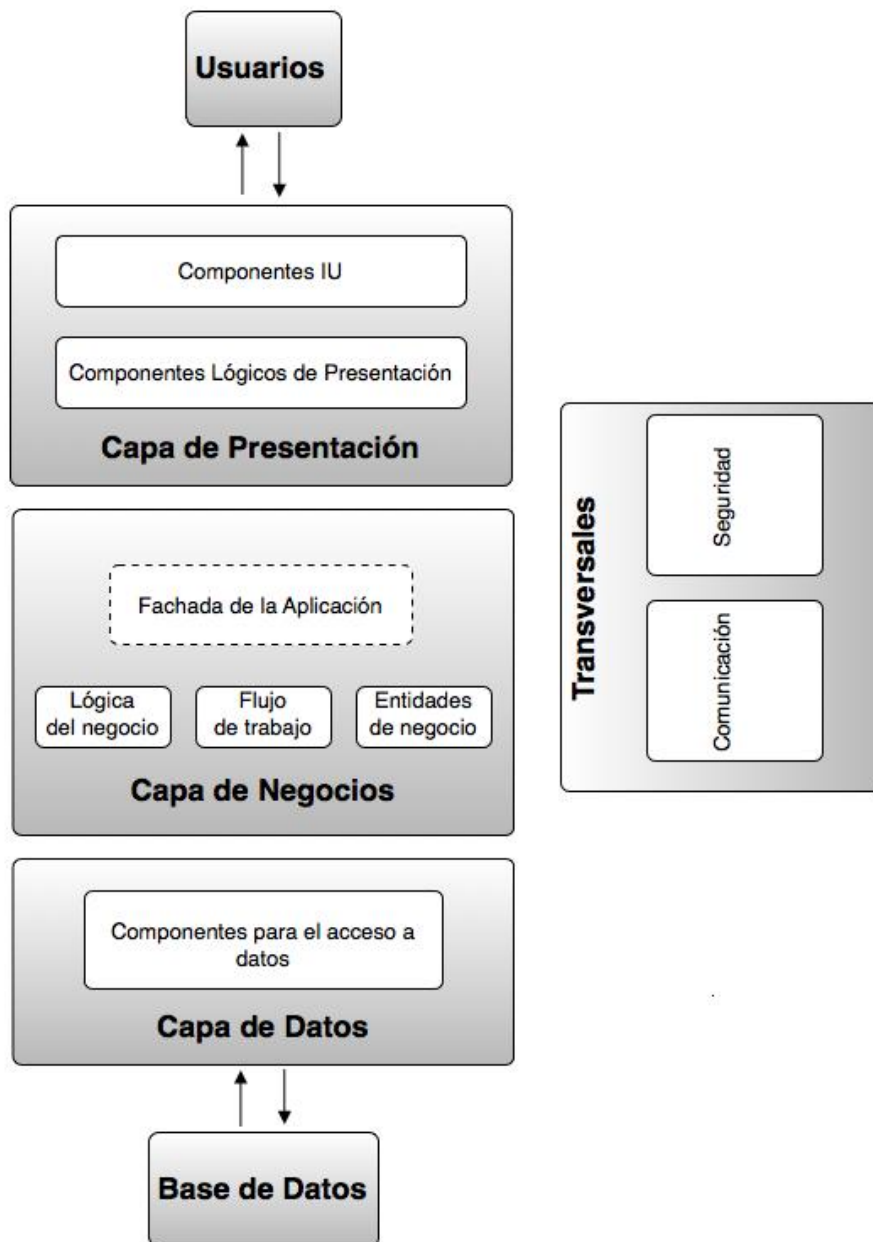


Figura 1: Arquitectura de tres capas que implementará el sistema.

Esquema de funcionamiento de la aplicación:



Figura 2: Diagrama del sistema en el entorno de producción.

Proceso de desarrollo a utilizar en el proyecto

El proyecto se realizará dentro del marco de desarrollo de software conocido como Proceso Unificado de Desarrollo de Software.

Se ha elegido este modelo de desarrollo debido a su versatilidad. Puede ser utilizado para construir una gran cantidad de tipos de sistemas de software y usado en diferentes tipos de organizaciones. Afronta en una fase temprana los mayores riesgos del diseño, evitando así retrasos importantes en el desarrollo. El Proceso Unificado es un marco de trabajo que además puede ser adaptado a proyectos u organizaciones específicas.

El Proceso Unificado se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

Iterativo e incremental

El Proceso Unificado es un marco de desarrollo compuesto de cuatro fases:

- Inicio
- Elaboración
- Construcción
- Transición

Cada una de ellas es, a su vez, dividida en una serie de iteraciones que ofrecen como resultado un incremento del producto desarrollado, que añade o mejora las funcionalidades del sistema en desarrollo.

Durante cada una de estas iteraciones se realizarán a su vez las actividades definidas en el ciclo de vida clásico: requisitos, análisis, diseño, implementación, prueba e implantación.

Aunque todas las iteraciones suelen incluir trabajo en casi todas estas actividades, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

Dirigido por los casos de uso

Un sistema software se crea para servir a sus usuarios por lo que, para construir un sistema exitoso, se debe conocer qué es lo que quieren y necesitan.

En el Proceso Unificado, los casos de uso se utilizan para capturar los requisitos funcionales y para definir los objetivos de las iteraciones. En cada una, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso.

Centrado en la arquitectura

El concepto de arquitectura del software involucra los aspectos estáticos y dinámicos más significativos del sistema, y actúa como vista del diseño, dando una perspectiva completa y describiendo los elementos más importantes

Puesto que la arquitectura y los casos de uso están relacionados, por una parte, los casos de uso deben, cuando son realizados, acomodarse en la arquitectura, y ésta debe ser lo bastante flexible para realizar todos los casos de uso.

Enfocado en los riesgos

Para disminuir la posibilidad de fallo en las iteraciones, se deben llevar a cabo sucesivos análisis de riesgos durante todo el desarrollo.

Por supuesto, los riesgos principales deben ser identificados en una etapa temprana del ciclo de vida, y además, los resultados de cada iteración deben seleccionarse en un orden que asegure que estos son considerados primero.

En la siguiente imagen se puede observar un ejemplo del ciclo de vida del Proceso Unificado:

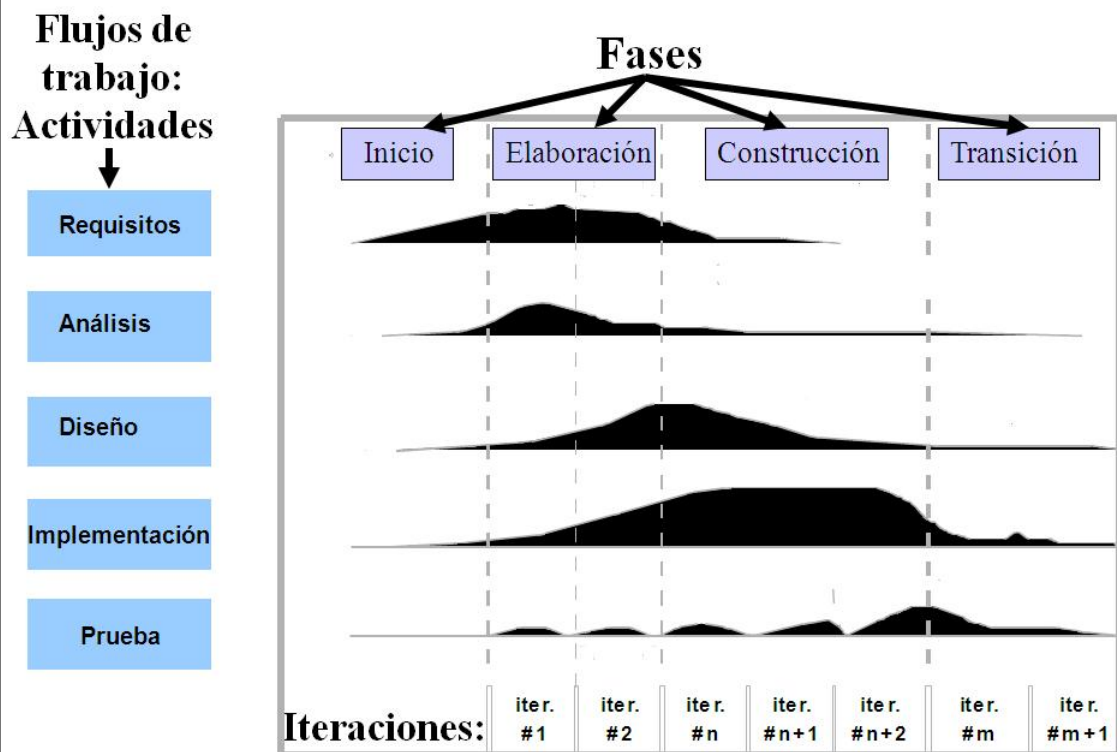


Figura 3: Representación de las fases y actividades del Proceso Unificado

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones abarcan más los flujos de trabajo de requisitos, modelo de negocios, análisis, diseño y una parte de implementación.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega al usuario.

Dentro de los flujos de trabajo podemos distinguir entre:

- Captura de requisitos: Se identifican los requisitos del sistema y se construye un modelo del mismo, a partir de las funcionalidades capturadas en los casos de uso.
- Análisis: Se documentan y especifican detalladamente los requisitos identificados anteriormente. Al concluir se tendrá una especificación de lo que debe hacer el sistema, sin entrar en detalles internos.
- Diseño: Se selecciona la aproximación básica para resolver cada funcionalidad. Se especifican las herramientas a usar en la etapa de codificación.
- Implementación: se trata de la fase de programación, en ella se implementará el código fuente, realizando pruebas y ensayos para corregir errores. En función del lenguaje o lenguajes utilizados se crearán las bibliotecas necesarias y demás componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso más rápido y eficiente.
- Prueba: una vez se tenga una versión de la aplicación se realizan una serie de pruebas para comprobar el correcto funcionamiento del sistema en conjunto.

Planificación del trabajo y estimación temporal

La planificación temporal del proyecto será la siguiente:

Dedicación semanal prevista (en horas/semana): 30 Dedicación diaria prevista (en horas/día): 5	
Fase	Estimación temporal (en semanas)
Requisitos	
Estudio preliminar	1
Análisis	
Análisis	1
Diseño	
Diseño del sistema	1
Base de datos	
Diseño de la BD	1
Implementación de la BD	1
Implementación	
Codificación	4
Pruebas	5
Documentación	
Manual de usuario	2
Manual técnico	9
TOTAL PROYECTO	9

A continuación mostraremos el diagrama de Gantt que nos mostrará la planificación temporal de una forma más específica:

Nombre de tarea	Duración	Comienzo	Fin
<input checked="" type="checkbox"/> TOTAL	59 días	lun 02/08/10	mié 29/09/10
<input checked="" type="checkbox"/> Requisitos	2 días	lun 02/08/10	mar 03/08/10
Estudio Preliminar	2 días	lun 02/08/10	mar 03/08/10
<input checked="" type="checkbox"/> Análisis	5 días	mié 04/08/10	dom 08/08/10
Análisis	5 días	mié 04/08/10	dom 08/08/10
<input checked="" type="checkbox"/> Diseño	9 días	lun 09/08/10	mar 17/08/10
Diseño del sistema	7 días	lun 09/08/10	dom 15/08/10
Diseño de la BD	1 día	lun 16/08/10	lun 16/08/10
Implementación de la BD	1 día	mar 17/08/10	mar 17/08/10
<input checked="" type="checkbox"/> Implementación	35 días	mié 18/08/10	mar 21/09/10
Codificación	30 días	mié 18/08/10	jue 16/09/10
Pruebas	35 días	mié 18/08/10	mar 21/09/10
<input checked="" type="checkbox"/> Documentación	59 días	lun 02/08/10	mié 29/09/10
Manual de usuario	10 días	vie 17/09/10	dom 26/09/10
Manual técnico	59 días	lun 02/08/10	mié 29/09/10

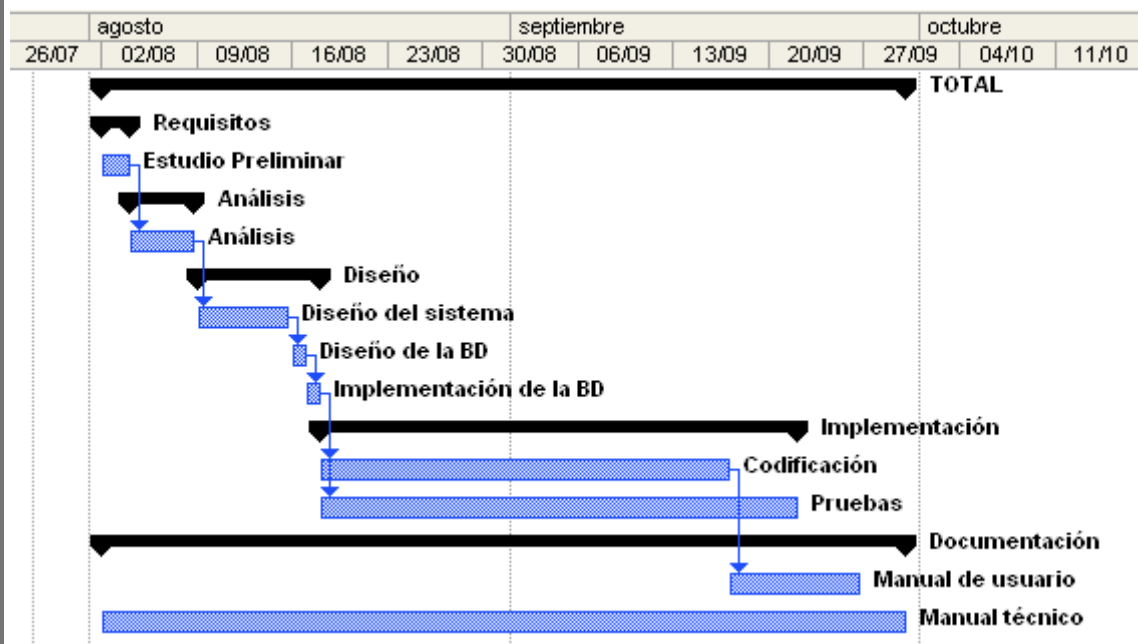


Figura 4: Diagrama de Gantt

Descripción de la documentación a entregar en el proyecto

La documentación del proyecto quedará recogida en un único volumen, que contendrá tres partes bien diferenciadas:

- Memoria: recogerá las consideraciones respecto al proceso de desarrollo del proyecto, comenzando con una introducción al problema a tratar, en la que se abordarán los objetivos y metodologías escogidas dentro del proyecto. También se expondrán reflexiones respecto al análisis y diseño de la aplicación. Finalmente se incluirán la bibliografía utilizada, las conclusiones y las posibles ampliaciones del proyecto.
- Manual técnico: describirá en profundidad la lógica de la aplicación. Incluirá el modelo de casos de uso de la herramienta y el modelo del dominio, que incluye las clases del proyecto.
- Manual de usuario: contendrá las indicaciones necesarias para la instalación y uso de la aplicación, incluyendo requisitos software y hardware, guía de instalación e instrucciones para el uso de la aplicación.

También se adjuntará un CD que incluirá la documentación del proyecto y el software que lo compone, así como el software necesario para la instalación y funcionamiento de la aplicación.

Medios materiales necesarios

Hardware.

- PC con conexión a Internet para búsqueda de información y codificación de la aplicación:
 - Intel Core 2 Duo 3,00 Ghz
 - 2 Gb de RAM
 - Disco Duro 250 GB
- Impresora de tinta para probar las funcionalidades que requieren la impresión de documentos.

Software.

- Sistema Operativo: Windows XP Home Edition Service Pack 2
- Sistema Gestor de Base de Datos: SQL Server.
- Entorno de desarrollo: Visual Studio 2008 C# Express Edition.
- Para la realización de la planificación temporal: Microsoft Office Project 2003.
- Para el proceso de análisis y modelado del proyecto: Visual Paradigm for UML 6.1 Standard Edition.
- Procesador de textos: Microsoft Office 2007.

Bibliografía

Bibliografía básica.

- LARMAN, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2ª edición. México: Prentice Hall. 2002
- DAWSON, Christian W.; QUETGLÁS, Gregorio Martín. *El Proyecto fin de carrera en ingeniería informática, una guía para el estudiante*. Madrid : Prentice Hall.2002
- MCCONNELL, Steve. *Desarrollo y gestión de proyectos informáticos*. Madrid: McGrawHill. 1996
- SHARP, John. *Visual C# 2008 Step by Step*. Estados Unidos: Microsoft Press. 2008
- HOTEK, Mike. *SQL Server 2008 Paso a Paso*. Madrid: Anaya. 2008
- HUDDLESTON, James; VRAT AGARWAL, Vidya. *Beggining C# 2008 Databases*. Estados Unidos: Apress. 2008
- CEBALLOS, Javier. *Enciclopedia de Microsoft: Visual C#*. Madrid: Ra-Ma. 2006
- JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. *El proceso unificado de desarrollo de software*. Madrid: Prentice Hall. 2000
- GUNDERLOY, Mike; JORDEN, Joseph L.; TSCHANZ, David W. *La Biblia de SQL Server 2005*. Madrid: Anaya. 2006
- LAUDON, Kenneth C.; LAUDON, Jane P. *Sistemas de información gerencial*. 8ª edición. México: Prentice Hall. 2004

Recursos Web.

- MEIER,J .D.; HOMER, Alex; HILL, David; TAYLOR, Jason; BANDOSE, Prashant; WALL, Lonie; BOUCHER Jr, Rob; BOGAWAT, Akshay. *Application Architecture guide 2.0*. Disponible en Web: <<http://msdn.microsoft.com/en-us/library/dd673617.aspx>>. [Consulta: 27 de julio de 2010]

Observaciones