

1 INTRODUCCIÓN

El Proyecto Fin de Carrera es el paso que culmina la etapa de estudios de Ingeniero de Telecomunicación. En el caso del presente documento hay que destacar que se ha desarrollado de forma externa al contexto académico puesto que ha sido convenido con una empresa real de distribución de material eléctrico.

Veremos a continuación qué objetivos cubre este Proyecto y cuál ha sido la motivación para su desarrollo.

1.1 MOTIVACIÓN Y OBJETIVOS

El Presente Proyecto Fin de Carrera tiene como objetivo la optimización del proceso de preparación de pedidos –comúnmente conocido como *picking*- en el contexto de una empresa de distribución de productos y materiales eléctricos.

La preparación de pedidos es un proceso del almacén logístico cuyo fin es recolectar una serie de productos disponibles en el almacén y reagruparlos en un sitio específico antes de ser expedidos hacia el cliente. Esta actividad es sin duda una de las actividades de mayor importancia de un almacén ya que puede afectar a la productividad de toda la cadena logística y en muchos casos llegar a ser un cuello de botella. Es por ello que se convierte en uno de los procesos clave en el momento de reducción de costes que es realmente el objetivo que persigue este proyecto.

La empresa donde se ha desarrollado este proyecto tenía unas necesidades reales de optimizar su proceso de *picking* puesto que eran conscientes de que se podía mejorar de alguna forma su forma de trabajo actual, siendo ésta la motivación de donde surge el proyecto.

1.1.1 RESUMEN DE LA SOLUCIÓN DESARROLLADA EN EL PROYECTO

Cada almacén suele ser un caso particular y con grandes diferencias respecto otros. Es por tanto fundamental estudiar en primer lugar la forma de trabajo actual de los operarios y usarla como base para proponer mejoras y ayudas a su mecánica de trabajo. No se busca reemplazar el papel del operario por una tecnología o proceso automatizado sino proporcionarle herramientas que hagan más eficaz su labor y por tanto ésta sea menos propensa a cometer errores. La opinión del operario es fundamental para el buen curso del proyecto ya que implantar una interfaz excesivamente automatizada puede provocar rechazo.

Una vez vista la forma de trabajo actual del operador se propondrá implantar en el proceso de *picking* un uso normalizado de una tecnología de autoidentificación. Tras compararlo con otras tecnologías existentes como los botones de memoria de contacto y etiquetas RFIDs se decide utilizar el código de barras, fundamentalmente por su bajo costo. Además esto permite reutilizar los códigos de barras ya existentes en los productos del almacén simplemente catalogándolos en el ERP de la Empresa. Aquellos productos donde este código de barras sea inexistente o no se pueda utilizar serán reetiquetados con un nuevo código interno a la Empresa.

Se propone el uso por parte de los operarios de un ordenador de mano equipado con lector de código de barras y comunicaciones inalámbricas. Se desarrollará una aplicación en Visual Basic.NET que se ejecutará en dicho terminal bajo la plataforma

.NET CF y que proporcionará la verificación de los productos y cantidades escaneadas respecto del pedido que el operador pretende servir.

La aplicación se comunicará de forma inalámbrica con un servicio web ASP.NET que también se desarrollará en este proyecto. El servicio web hará de intermediario entre el servidor ERP de la empresa y los dispositivos de mano. Del primero tomará los datos necesarios de los pedidos a servir para enviárselos a los dispositivos y una vez finalizadas las lecturas se devolverán estas al ERP generando un documento de venta.

Destacar que la aplicación también funcionará parcialmente de forma desconectada ya que en el almacén no se puede garantizar cobertura inalámbrica total. Una vez descargado el pedido a servir en el dispositivo, el operario puede realizar el *picking* con normalidad aunque no tenga cobertura ya que la aplicación se vale de su base de datos local SQL CE. Una vez de regreso a la zona cobertura el operador podrá cargar los datos leídos y finalizar la operación en el ERP.

Además para facilitar lo máximo posible a los operarios el uso del dispositivo de mano y minimizar los errores se han codificado la mayoría de comandos que deben usar como códigos de barras. De esta forma el operador para dar una orden de “cargar un pedido” o “descargar un pedido” escanearía el código de barras correspondiente de los que lleva siempre consigo en lugar de tener que mirar la pantalla y activar una opción. De esta forma el proceso de *picking* resulta mucho más fluido.

La aplicación además permite descargar en el dispositivo múltiples pedidos e incluso llevar una operación sin pedido o *ad-hoc*, todas ellas simultáneamente. El operario puede entonces dirigirse a las ubicaciones de cada producto e ir marcando en el dispositivo a qué pedido corresponde.

Desde el punto de vista técnico destacar que la aplicación internamente se ha orientado a comandos usando el patrón de diseño correspondiente. De esta forma se han podido componer comandos complejos en base a otros más simples. Esto también ha simplificado la escritura de algunas baterías de pruebas de comportamiento.

Finalmente comentar que todos los métodos y software expuestos en este documento se han llevado a cabo con éxito en la empresa y siguen funcionando y evolucionando positivamente a fecha de hoy.

1.2 MATERIAL Y MÉTODOS

Se elige un terminal portátil con lector de código de barras y con comunicaciones inalámbricas WiFi de la marca Datalogic ADC modelo Skorpion X3 y muy orientado al tipo de aplicaciones como la que se va a desarrollar. El terminal tiene la posibilidad de adquirirse con sistema operativo Windows CE 6.0 o Windows Mobile 6.5, siendo este último el elegido para la realización de la aplicación.

La plataforma sobre la que se programa en este entorno es el .NET Compact Framework, una versión reducida del .NET Framework de escritorio y orientada a dispositivos móviles inteligentes. Esta librería de tiempo de ejecución es proporcionada por Microsoft de forma gratuita y sobre ella se puede programar empleando lenguajes como C# o Visual Basic.NET

El entorno de desarrollo recomendado para implementar aplicaciones en Windows Mobile es Visual Studio 2008, ed. Professional, siendo esta el elegido para llevar a cabo el proyecto. Hay que destacar que esta edición incluye un emulador de Windows Mobile con el que se puede probar y depurar las aplicaciones desarrolladas para este sistema operativo.

La empresa Datalogic proporciona un SDK para el terminal Skorpion que también será utilizado en el desarrollo de la aplicación.

Respecto la parte servidora, se programará un servicio web en la plataforma ASP.NET utilizando también el Microsoft Visual Studio 2008 Professional.

1.2.1 HARDWARE

Se ha utilizado un equipo portátil para el diseño, programación y depuración de la aplicación. Por otro lado, se ha empleado el terminal portátil Datalogic Skorpion ya mencionado.

Las características de ambos equipos son las siguientes:

- Equipo portátil:
Portátil Asus F55V, procesador Core i3, 4GB de RAM
- Terminal Portátil:
Datalogic Skorpion X3, escáner láser de alto rendimiento, Radio 802.11 b/g, Bluetooth

1.2.2 SOFTWARE

El sistema operativo empleado en el equipo portátil es el Microsoft Windows 7 Home Premium con Service Pack 1 instalado.

El sistema operativo del terminal Datalogic es Windows Mobile 6.5. Éste incluye la plataforma Microsoft .NET Compact Framework 3.5, en adelante .NETCF.

En cuanto a los programas y utilidades empleadas en el equipo portátil, se detallan a continuación.

1.2.2.1 Microsoft Visual Studio Professional 2008

Visual Studio es un entorno de desarrollo integrado para sistemas Windows que soporta una variedad de lenguajes de programación tales como C#, Visual Basic.NET o Visual C++.

Es necesario utilizar la edición 'Professional' ya que incluye todo lo necesario para el desarrollo y depuración de aplicaciones en dispositivos de mano inteligentes, así como servicios web en la plataforma ASP.NET lo cual será necesario para la parte del servidor.

1.2.2.2 Datalogic C/C++ SDK para Windows Mobile

Es un SDK proporcionado por Datalogic y que incluye las API para acceder a funcionalidades específicas del terminal portátil y que no vienen incluidas de forma estándar en .NET CF, como por ejemplo la funcionalidad Bluetooth o Wifi. Puede ser descargado de la siguiente dirección:

http://www.adc.datalogic.com/Skorpio_prd_prd61_7_sw.html

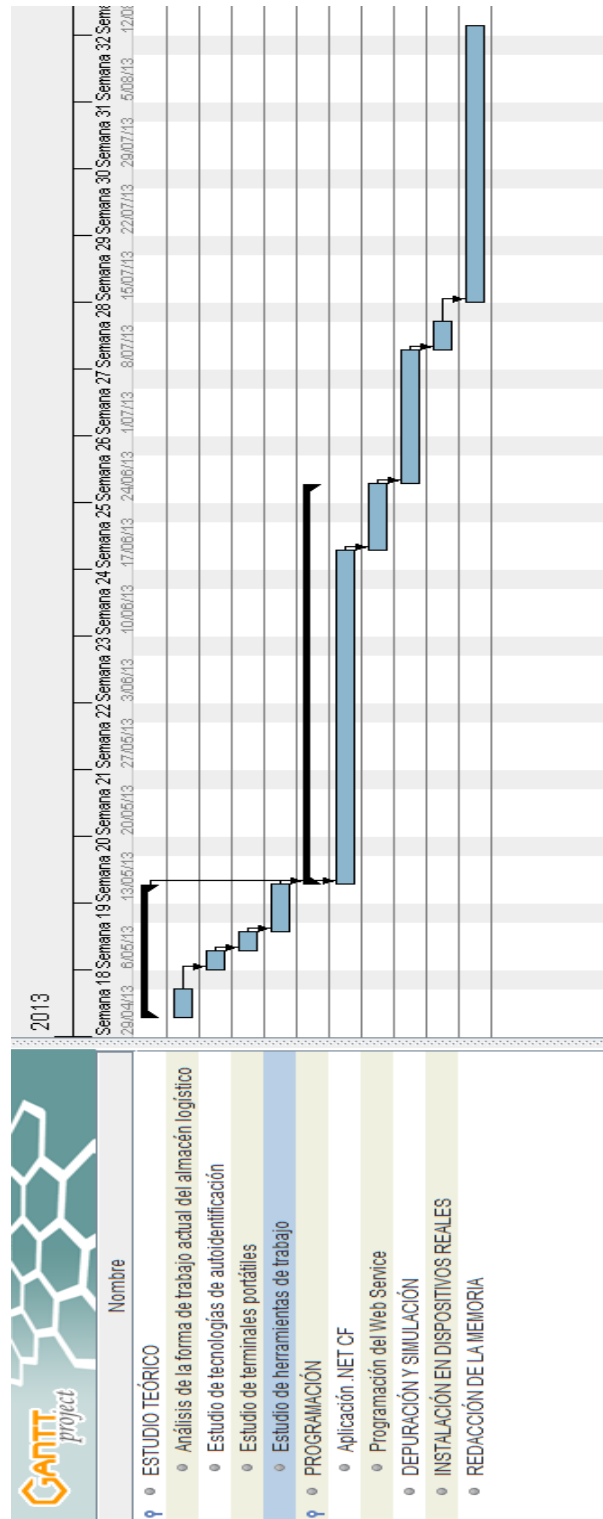
1.2.2.3 GanttProject

Es una herramienta de código abierto para realizar diagramas de Gantt como el que aparece en esta memoria. Puede descargarse de la siguiente dirección:

<http://ganttproject.sourceforge.net/>

1.3 ETAPAS DE DESARROLLO Y TIEMPOS

Detallaremos a continuación las etapas por las que ha pasado el desarrollo del presente proyecto. Para ello hacemos uso del diagrama de Gantt de la figura 1-1, que indica la extensión en el tiempo y el solapamiento de las diferentes etapas.



1-1 Diagrama de Tiempos

1.3.1 ANÁLISIS DE LA FORMA DE TRABAJO ACTUAL DEL ALMACÉN LOGÍSTICO

Este apartado implica un pequeño estudio de campo junto a los operadores del almacén para recopilar las tareas que realiza y sus métodos y concluir cómo pueden mejorarse

1.3.2 ESTUDIO DE TECNOLOGÍAS DE AUTOIDENTIFICACIÓN

Este apartado conlleva la selección de la tecnología de autoidentificación que mejor se adapte a los propósitos del presente proyecto y de ver la mejor forma de aplicarla a las necesidades de la empresa

1.3.3 ESTUDIO DE TERMINALES PORTÁTILES

Se tratará de evaluar terminales de mano que cumplan las condiciones necesarias para el desarrollo del proyecto.

1.3.4 ESTUDIO DE HERRAMIENTAS DE TRABAJO

Aunque se trata de herramientas conocidas por el autor como el Microsoft Visual Studio, se incide en la parte que incluye el desarrollo para dispositivos inteligentes ya que será el grueso de desarrollo de programación.

1.3.5 APLICACIÓN .NET CF

Esta parte sería ya el desarrollo en lenguaje Visual Basic .NET de la aplicación

1.3.6 PROGRAMACIÓN DEL SERVICIO WEB

La aplicación del dispositivo de mano cuenta con una parte servidora en forma de servicio web que se desarrollará en este punto

1.3.7 DEPURACIÓN Y SIMULACIÓN

A lo largo de esta etapa se monitoriza el funcionamiento de la aplicación en todos sus niveles, se comprueba que funciona como es de esperar y se hacen cambios si es necesario. Esta etapa incluye la depuración de la aplicación en un emulador.

1.3.8 INSTALACIÓN EN DISPOSITIVOS REALES

Instalación real en dispositivos de la empresa y ajuste fino de la aplicación según las incidencias encontradas.

1.3.9 REDACCIÓN DE LA MEMORIA

Finalmente se redacta la presente memoria del proyecto.

1.4 ESTRUCTURA DEL DOCUMENTO

En este apartado se define cual es la estructura que el lector se encontrará en la presente memoria:

- En la sección 2 se hará un repaso sobre el estado del arte de las tecnologías implicadas en este proyecto. Esto incluye las tecnologías de autoidentificación, los ordenadores de mano industriales y el framework de programación .NET Compact Framework.
- La sección 3 constituye el grueso del documento ya que es donde se detalla el desarrollo del presente proyecto. Esto incluye un pequeño estudio de campo sobre la forma de trabajo del operador para estudiar las deficiencias encontradas. Seguidamente se darán los parámetros para aplicar la tecnología de autoidentificación escogida: el código de barras. Por último se detalla el desarrollo de la aplicación .NET CF incluyendo diagramas de clase, base de datos y diseño del servicio web.
- Las conclusiones del proyecto y futuras líneas de avance se comentarán en la sección 4.
- En el último apartado, el 5, encontramos la bibliografía.
- Como anexo al documento se encuentra principalmente el código de la aplicación .NET CF y del servicio web