

Reingeniería del módulo de facturación del ERP de una fabrica

Andres Giovanni Vargas Peña

Resumen—Una empresa textil con una larga trayectoria a través de los años desea actualizar todo su programa de gestión de la fábrica, comercio, compres y ventas, clientes y administración interna. Para ello ha contratado a una empresa de software para desarrollar esta actualización. Desean que se tome como base la aplicación anterior y se le haga tanto un lavado de cara como una mejora en los procesos del software. Para ello, la empresa lleva los servicios al cloud y pasa el software a una aplicación web y se hace un proceso de ingeniería inversa para conocer cómo funciona la aplicación anterior y se hacen varias entrevistas con el personal para conocer el funcionamiento de la empresa y las respuestas deseadas para el usuario del software final. Yo Me encargo de realizar todo este proceso en el apartado de los albaranes y las facturas de una sección de la empresa.

Palabras Clave—ERP, Reingeniería, Facturación, Ingeniería de software, Telerik Reporting, Kendo UI, Visual Studio, MVC, Aplicación Web, Metodología ágil.

Abstract—A textile company with a long history over the years wants to update its entire program of factory management, trade, buy and sell, customers and internal administration. For this, he has hired a software company to develop this update. They want the previous application to be taken as a basis and both a facelift and an improvement in the software processes. For this, the company takes the services to the cloud and passes the software to a web application and a reverse engineering process is done to know how the previous application works and several interviews are made with the staff to know the operation of the company and the desired answers for the end user. I am in charge of carrying out all this process in the section of the delivery notes and the invoices of a section of the company.

Key Words—ERP, Re-engineering, Billing, Software Engineering, Telerik Reporting, Kendo UI, Visual Studio, MVC, Web Application, Agile Methodology.



1 INTRODUCCIÓN

Este proyecto tiene como finalidad desarrollar un el apartado de albaranes, facturación y devoluciones de los mismos para una empresa de fábrica textil. Dado que la empresa textil tiene ya un sistema que llevan usando desde hace unos 15 años se han visto desfasados y ven claramente que este sistema ya que el sistema anterior necesita cambios urgentes y ya no es posible seguirle dándole actualizaciones según las necesidades ni es posible tampoco mantenerlo a flote con tanto cambio que se ha aplicado desde su versión original. Dado que la empresa textil no cuenta con un departamento de desarrollo ni mantenimiento del software, han decidido contratar a la empresa Einatec Consulting para que les lleve a desarrollo la aplicación deseada.

- E-mail de contacte: andresjovanni.vargas@e-campus.uab.cat
- Menció realitzada: Enginyeria del Software.
- Treball tutoritzat per: LLuis Gesa (Departament de Ciències de la computació)
- Curs 2018/19

Dada esta situación Einatec consulting tomo los requerimientos básicos del cliente para comprender qué tipo de software debían desarrollar y como podían adecuarlo a las necesidades del cliente. Esto ha llevado a desarrollar un sistema ERP el cual integra (entre muchos otros módulos) un módulo de creación y facturación de albaranes Pasando por un extenso entendimiento de la aplicación anterior y los problemas que esta tenía (teniendo reuniones con los clientes y ellos mostrando cuales eran los fallos de esta) Hasta una modernización visual y funcional de TODO el código y vistas de la aplicación. Por suerte se contaba con el código fuente de la aplicación anterior y la BBDD fuente anterior de modo que se podía realizar todo un proceso de reingeniería para saber cómo este hecho antes, que estaba mal y tener una idea básica de cómo debía comportarse el programa. Así mismo, se debía hacer una actualización de los sistemas que usa el mismo programa para adecuarlos a estándares más modernos. El TFG aquí presente se encarga de tomar las necesidades, los requisitos y requerimientos del cliente para poder llevar a cabo este módulo de facturación.

2 OBJETIVOS

2.1 Objetivo principal:

El objetivo principal de todo este proyecto es analizar, diseñar, implementar, testear y actualizar tanto el front-end como el back-end de un software existente a una tecnología moderna que permita realizar las tareas de manera más sencilla y eliminar todos los errores de la aplicación anterior (sufre de muchos problemas de software y decisiones mal tomadas o procesos mal planteados).

2.2 Objetivos secundarios:

- Desarrollar habilidades de comunicación para con el cliente.
- Aprender nuevas tecnologías de desarrollo software tales como KendoUI, Telerik reporting, Razor MVC C# y Bootstrap.
- Realizar diferentes reportes en formato PDF para que el cliente pueda imprimir documentos físicos.
- Comprender como funciona un sistema de facturación de una empresa.
- Mejorar las habilidades de creación de aplicaciones web utilizando el esquema MVC.
- Comprender el funcionamiento del Entity Framework para modificar, consultar y almacena datos en una base de datos SQL Server.

3 ESTADO DEL ARTE

Los sistemas ERP han tenido una demanda y evolución bastante elevada durante los últimos años dado que estamos viviendo la "Era de la información" donde es requerido tener controlados todos los componentes de la empresa y los diferentes datos que esta pueda generar. Para ello, las empresas han estado buscando una herramienta que les permita acceder a esta era y no quedarse sin una ventaja estratégica para el negocio. Ahí es donde entran los sistemas ERP.



Fig. 1. Imagen representativa de un ERP [3]

Primero que nada, es necesario comprender que es un ERP y por qué su implementación en el sector empresarial está siendo fundamental para todas las organizaciones.

Un ERP es un "Enterprise Resource Planning" por sus siglas en inglés o un sistema de planificación de recursos empresariales donde se mantiene monitoreada las distintas operaciones de la empresa desde el proceso de recolección de materiales hasta el mismo servicio postventa, realizando diferentes trabajos como los registros del movimiento de las materias primas, o el control de los clientes/proveedores y las compras y las ventas (entre muchas otras tareas que se pueden implementar en el sistema). Una de las grandes ventajas de los sistemas ERP es que permite mucho una implementación modular, de modo que se pueden poner en ejecución ciertos apartados necesarios para la organización (gestión interna de trabajadores, compras/ventas) y en un futuro agregar nuevos módulos de gestión o agregar nuevas funcionalidades a los módulos existentes (o realizar un proceso de reingeniería y mejorar su funcionamiento).

Las organizaciones tienen diferentes motivos para crear un sistema ERP, según el reporte de Panorama [4]:

- El 49% de las empresas reemplaza un ERP que se encuentra desfasado.
- El 16% lo hace para reemplazar un sistema hecho para ellos.
- El 15% lo hace para reemplazar un software contable.
- El 20% o hace para reemplazar un sistema que no es un ERP o que como tal no es un sistema como tal (que se encuentre conectado entre sus módulos).

Y aunque son sistemas altamente necesarios para las organizaciones no representan un sobre coste para las mismas. Por lo general el coste general de un sistema ERP representa un 6.5% de sus gastos ANUALES realizar un proyecto ERP (según el informe de panorama [4], el 35% de las empresas usa entre el 1-3% de sus gastos anuales y el 20% usa entre el 3-5% de sus gastos). Otros datos interesantes que destacar de los sistemas ERP Son los siguientes:

- El 81% de las organizaciones están implementando un ERP o tienen un sistema implementado
- El 14% de las organizaciones está en un proceso de selección del software.
- El 5% está en proceso de mejorar el software.
- El 16% La razón más común para implementar un sistema ERP es para reemplazar algún ERP viejo o de un software antiguo (como es nuestro caso).

Dentro de los ERP existen diferentes variedades de software (o aplicaciones web) creados para uso genérico o realizar una personalización de este. Aquí podemos ver

que softwares son comúnmente usados:

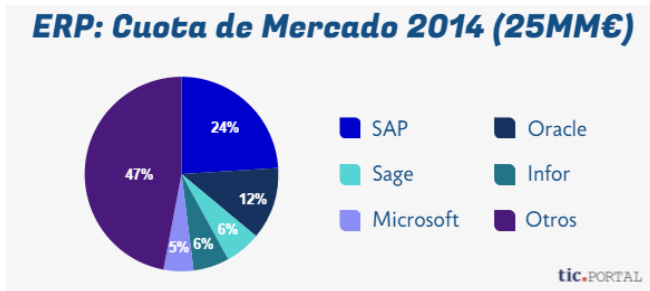


Fig. 2. Gráfico que presenta los softwares usados (2014) [5]

Estos sistemas pueden ser tanto modificables por el usuario o el proveedor te da todas las soluciones en pequeños módulos que te dan. ¿Cuál es el mejor? Depende de las necesidades de la empresa y de que tan específico sea lo que se quiera hacer (para eso hay soluciones de ERP a medida).

4 METODOLOGÍA

Para este proyecto dadas las necesidades de la empresa, cliente y el TFG, se ha decidido optar por la metodología del DESARROLLO ÁGIL.

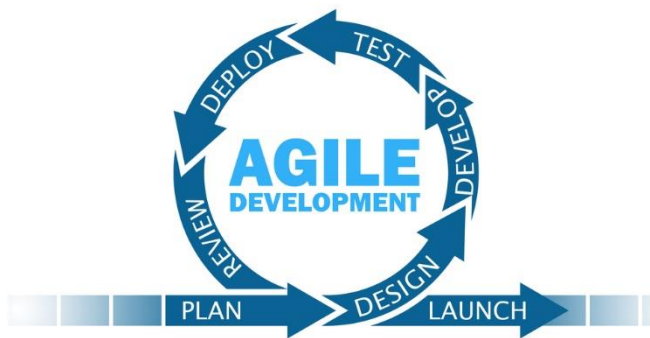


Fig. 3. Diagrama representativo del ciclo de vida del desarrollo ágil [6]

El desarrollo ágil es una serie de procesos y best practices que responde a las necesidades de dar soluciones prontas a los clientes, o desarrollar software en poco tiempo, dado que la metodología más usual para el momento de su creación (a mediados de 1990) era cascada, que estaba basado en métodos demasiado rigurosos y que alargaban los tiempos de desarrollo.

AGILE METHODS AND PRACTICES

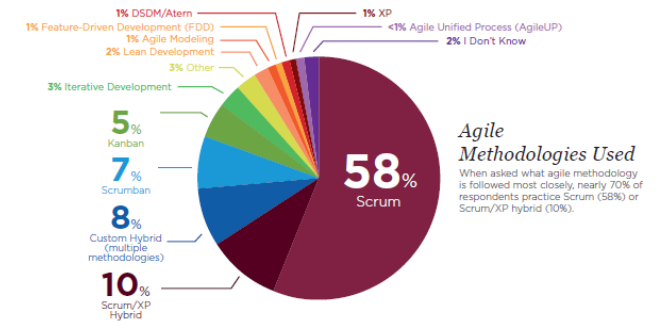


Fig. 4. Uso de la metodología ágil (2016) [7]

El desarrollo ágil enfoca su ciclo de vida en los “sprints” que son pequeñas iteraciones en los cuales se van entregando pequeñas versiones del software completadas. Es importante que el software que se quiere desarrollar pueda ser modular para que se pueda ir entregando en diferentes iteraciones.

En el ciclo de vida del desarrollo ágil ocurren 4 actividades principales:

- Planificar: Se hacen reuniones con los clientes para saber que se quiere desarrollar, cual es el estado actual del proyecto y saber que es lo próximo a hacer. En este punto es donde se hace la toma de requerimientos a implementar.
- Desarrollo: Con los requerimientos existentes se procede a implementar en el programa, acorde a los requerimientos se desarrollan las actividades para la implementación.
- Testeo: Una vez se ha implementado el software se procede a realizar un testeo profundo de lo implementado, asegurándose que lo implementado tenga la menor cantidad de fallos posibles y a ser posible solucionarlos.
- Feedback: una vez que se ha comprobado que el software funciona como debería, se procede a hacer una reunión más con el cliente y/o usuario final, dado que nos tendrá que decir si los procesos y la implementación es tal y como la desea o si desea hacer algún cambio en lo entregado. En caso de que todo sea correcto se pasa a una nueva implementación.

Es importante destacar que aplicar la metodología de desarrollo ágil tiene como requisito CASI OBLIGATORIO que el cliente disponga de una buena cantidad de tiempo para estar implicado en el desarrollo, dado que se necesita un *feedback* constante de parte del usuario final/cliente. Algo que destacar del desarrollo ágil es que deja en un segundo plano la documentación, reduciéndola a documentos netamente necesarios y además poco extensos para reducir los tiempos de diseño y dar más importancia al desarrollo.

Aunque esta metodología da buenos resultados, no es la solución total para todos los proyectos, solo es totalmente

recomendable para aquellos en los cuales se tenga muy poco tiempo para el desarrollo y se tenga un alto presupuesto, dado que los desarrolladores deben ser conocedores de las herramientas a usar para que todo el desarrollo sea pronto, y todos sabemos que los desarrolladores experimentados son un coste bastante elevado para el desarrollo.

5 PLANIFICACIÓN Y DESARROLLO

Para desarrollar este TFG fue necesario pedir permiso a la empresa Einatec Consulting para presentar esta parte de un proyecto que se está desarrollando para un cliente. Partiendo de esto y las necesidades de la empresa/cliente se decidió que la forma en la cual se desarrollaría el TFG sería con la metodología ágil, en la cual cada semana se tendría una reunión con el cliente para mostrarle el estado del desarrollo para resolver dudas o realizar cambios y en caso de que todo este correcto, se procedería a pedir requerimientos para la próxima pantalla a desarrollar. Por lo general, el desarrollo constaba de 5 pasos:

5.1 PASOS A REALIZAR

- Reunión con el cliente.
- Análisis de código legado.
- Replanteamiento según el código legado y cliente.
- Implementación de código.
- Testing de la implementación.
- *Feedback* del cliente.
- Corrección según cliente.
- *Feedback* y terminación.

5.1.1 Reunión con cliente:

Lo primero a realizar siempre son reuniones con el cliente para solicitarle parte de lo que se desea implementar (en nuestro caso exportar). En este punto se hace la primera toma de requerimientos de lo que se desea hacer (restricciones, comportamientos, vistas, cambios, añadidos). Se suele mostrar también como es el comportamiento en la aplicación antigua y nos exponen que problemas tiene y que se desea conservar/modificar.

```
-para las facturas internacionales hay 4 reportes por hacer :
-Carta de instrucciones.
-factura para el cliente.
-factura para lanitex.
-factura de aduana.
```

Fig. 5. Fragmento de uno de los requisitos tomados en cliente (primero se escribía una versión rápida en cliente y luego una versión más adecuada en el documento de requerimientos de manera más detallada).

5.1.2 Análisis de código legado:

Ya con la reunión tomada se procede a leer el código legado, entender que fue implementado, como fue implementado, porque fue implementado y, además, comprender la estructura de la base de datos anterior, entender como estaban enlazadas las anteriores tablas y com-

prender como cargaban los datos, en qué momento hacían x cambios a los elementos de la BBDD, y buscar relaciones en la BBDD anterior. Para lograr todo esto contábamos con el código original y la base de datos anterior, además de un pequeño entorno de pruebas con una BBDD replicada para probar el comportamiento en caso de que quisiéramos replicar comportamientos.

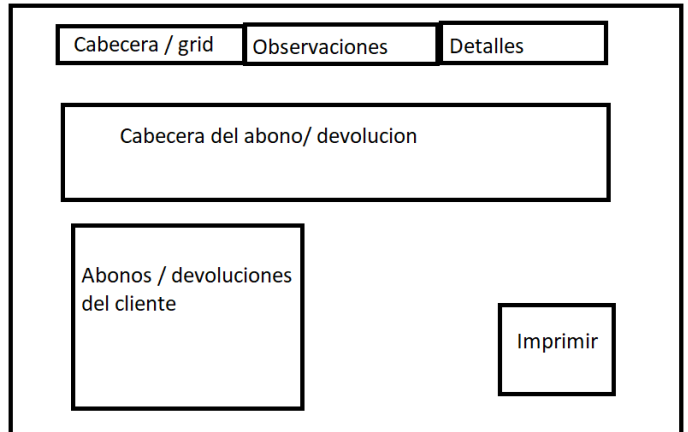


Fig. 6. Ejemplo de esquema de visualización de una de las pantallas a modificar del programa anterior (devoluciones).

5.1.3 Replanteamiento según el código legado y cliente:

Analizando el código anterior y la estructura de base de datos se procedía a replantear la BBDD modificando las tablas, extrayendo los datos en otras tablas para que las tablas no tengan demasiados datos o estén sobre cargados de información innecesaria. Una modificación de la BBDD original según se necesite para mejorar el proceso en general, o al menos hacer los datos más entendibles. Ya con la BBDD modificada para que los datos sean más adaptables a estándares modernos, Se procede a reanalizar la vista de la aplicación anterior y adaptarla a una vista HTML, usando diferentes herramientas visuales y de código como, el propio HTML, razor MVC c#, KendoUI...

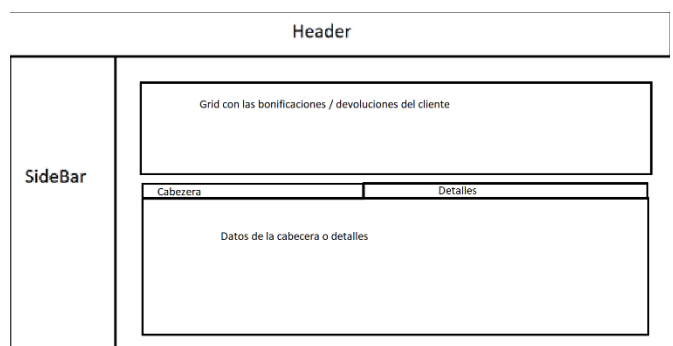


Fig. 7. Ejemplo de esquema de visualización de una de las pantallas a implementar ya con la remodelación hecha (devoluciones).

De modo que ahora las vistas son más agradables visualmente con iconos y colores más representativos en cada botón o input y por último se procede a modificar

los procesos realizados, de modo que ahora se cargan datos de diferentes maneras, o se añaden o modifican funcionalidades según las peticiones del cliente, ETC, vamos un proceso completo de reingeniería.

5.1.4 Implementación de código:

Teniendo ya todo esquematizado de cómo se va a planear, que cambios hay que hacer a la BBDD, como debería verse visualmente las paginas, y como deberían realizarse los procesos ahora. Se procede a plasmar todas estas funcionalidades y cambios en el código, de modo que si todo quedo bien esquematizado y los procesos son correctos no suele generar mucho problema a la hora de implementarlo. Muchas veces toca leer documentación de las herramientas para lograr los resultados deseados o si pueden realizarse X funcionalidad más rápido (véase todo el tema del responsive de Bootstrap, funcionalidades varias de KendoUI, aprender de 0 el funcionamiento de Telerik reporting y buscar posibles soluciones cada que aparecían nuevos obstáculos).

```
<!-- Obtener los abonos por el codigo del cliente de devolucion (invoicecustomer)
<script>
function getPaymentsByClientCode() {
    var code = $("#txtDevolClient").val();
    emptyFields();
    $("#txtDevolClient").val(code);
    getDataDevolts(code);
    $("#gridDetails").data("kendoGrid").setDataSource(dataSourceDetails);

    $("#botonAdd").show();
    $("#botonSave").hide();
    $("#botonDelete").hide();
}
</script>
```

Fig. 8. Fragmento del código implementado en las devoluciones (obtener las devoluciones según el código del cliente).

5.1.5 Testing de la implementación:

Ya con el código implementado y las vistas creadas se procede a realizar un proceso de testing en el cual por lo general consiste en comprobar que los datos requeridos son los indicados en los campos, que se estén enviando los datos debidos a las diferentes APIS de la aplicación, comprobar los comportamientos de la vista cuando ocurran X acciones (si ocurre x respuesta, se cargan x datos, si se da clic a X botón o elemento los menús cambian...) además de pruebas de inserción y modificación de datos en la BBDD para comprobar que todo se está enviando como debe y actualizando los valores correctos en los momentos adecuados y modificando las diferentes dependencias.

5.1.6 Feedback del cliente:

Una vez se ha testeado el código, se procede a mostrar los resultados al cliente, este se encargaría de comprobar el comportamiento de la herramienta mostrándole los pasos

a realizar según las diferentes funcionalidades, comprobando que los datos cargados o mostrados sean los correctos, analizando que, en caso de cálculo, se realicen las fórmulas pertinentes y demás. Además, se encargaría de revisar la nueva vista de la aplicación, de modo que indi-

cara si hay algún elemento que no le guste como se ve, o comprobar el texto cargado, ETC. En este punto el cliente puede destacar alguna funcionalidad faltante, agregar alguna funcionalidad que desee o pedir corrección de algo que no ha sido implementado correctamente, esto puede añadir nuevos requerimientos que deben ser agregados a la lista de requerimientos.

5.1.7 Corrección según cliente:

En caso de que el cliente haya dado nuevas indicaciones sobre el funcionamiento se procede a retomar los pasos 5.3, 5.4 y 5.5 dadas las nuevas indicaciones para proceder a mostrarlo al cliente de nuevo, este punto suele ser más corto dado que pueden ser pequeñas variaciones en el código como por ej, que se muestren X elementos que tengan Y característica o cambios visuales y algún pequeño comportamiento rápido. Pocas veces han sido las que han añadido cambios grandes o ha habido algunos errores grandes.

5.1.8 Feedback y terminación:

Para este punto se suele crear un ciclo entre el punto 5.6 y 5.7 en caso de que haya algún elemento que falte se haya olvidado o hayan surgido nuevos errores. Si todo ha ido de manera correcta se procede a dar por acabado esta parte de la implementación y se retoma el punto 5.1 para adquirir nuevos requerimientos para la próxima funcionalidad a implementar.

Como bien podemos ver, el proceso sigue la metodología de desarrollo ágil que hemos planteado (análisis-implementación-Testing-Feedback) lo cual permite pequeños sprints con el cliente (de 1 o 2 semanas como mucho) y permite desarrollar de manera más rápida la aplicación.

5.2 Herramientas y tecnologías:

Este desarrollo incluye varias herramientas que usa la empresa que promueve este TFG. Son herramientas que facilitan el trabajo en equipo, la interpretación del código y ayudan a desarrollar de manera más fácil y sencilla muchos elementos tales como los reportes y desarrollo de grids [8]. Aquí listaremos las herramientas que más soporte y ayuda nos han dado durante el desarrollo:

1. Kendo UI [9]: es una herramienta que permite el desarrollo ágil de elementos de formularios y vistas, tales como dropdowns, tablas con múltiples funciones, herramientas de listado de elementos rápida, campos de fechas rápida y un largo ETC, esta herramienta es muy potente y da mucha ayuda al desarrollo.
2. Telerik Reporting [10]: Es una herramienta de reportes que proporciona mucha ayuda y facilidad para generar vistas de reportes tales como facturas, detalles de compras, generación rápida pdf a partir de datos. Tiene un excelente asistente para la creación de estas vistas.
3. Visual Studio [11]: Visual estudio es un entorno de desarrollo que da muchas herramientas para faci-

litar el trabajo, entre sus muchas herramientas se ha utilizado la integración con git^[12] para tener un repositorio común para todos los involucrados en el desarrollo y tener un control de versiones, un asistente a la codificación que te ayuda a encontrar y resolver errores y un entorno visual agradable.

Tanto Kendo Ui como Telerik reporting pertenecen a la empresa *Progress Software Corporation* y consta de un excelente foro de ayuda y asistencia y una extensa API excelentemente documentada.

5.3 Requerimientos del sistema:

Los requerimientos del sistema se iban adquiriendo mediante cada iteración de las reuniones con los clientes dada la metodología escogida, al inicio del proyecto se tenía una idea general de todo el sistema, pero en cada iteración se van agregando los requisitos específicos de esa iteración. En el siguiente listado nombramos los requisitos de manera rápida:

1. El programa deberá tener una página de gestión de albaranes (creación, edición, vista y borrado). Que, además, muestre los detalles de los albaranes.
2. En la página de albaranes, el sistema deberá contar con un método de búsqueda de albaranes acorde al código del cliente y el número del pedido.
3. El sistema deberá poder generar diferentes reportes de los albaranes según una selección.
4. En la ventana de gestión de albaranes, habrá un formulario donde se muestren todos los campos y datos de los albaranes.
5. El sistema deberá, cuando se cargue la página de gestión de albaranes, cargar en el grid de albaranes todos los albaranes que estén activos y pendientes de facturar.
6. El sistema deberá, automáticamente, cargar los datos de un pedido y plasmarlo en sus respectivos campos de un albarán.
7. El sistema deberá, cuando se cargue un pedido, cargar en el modal de las piezas a facturar, las piezas que estén disponibles a facturar de ese pedido y todos los pedidos que compartan los datos de la cabecera del pedido.
8. El sistema tiene que, a partir de un formulario completo de albarán, poder crear un albarán y registrarlo en la base de datos.
9. El sistema deberá, mediante un albarán existente y seleccionado, tener la posibilidad de editar sus campos de albarán.
10. El sistema deberá tener un sistema de "borrado" de albaranes.
11. El sistema deberá generar diferentes reportes de los albaranes según se seleccionen.
12. El sistema deberá contar con una ventana de gestión de las facturas de exportación.
13. El sistema deberá mostrar los datos de una factura automáticamente según el código del cliente o el código de la factura.
14. El Sistema cuenta con una pantalla que permite realizar la facturación de los albaranes de los clientes de exportación.
15. El sistema debe permitir, a partir de una factura de exportación existente poder modificar los campos de bonificación y cargos extra de la factura de exportación.
16. El sistema deberá permitir "eliminar" una factura de exportación (poner su campo "IsActive" a false).
17. El sistema debe generar los diferentes reportes (documentos a imprimir) de los albaranes de exportación.
18. El sistema debe contar con una ventana de facturación de albaranes nacionales.
19. El sistema deberá cargar los albaranes nacionales que estén dentro de un rango de fechas Escritas en los campos de fecha desde, fecha hasta en la ventana de facturas nacionales
20. El sistema deberá, mediante un botón, facturar los albaranes nacionales que estén en el grid. Y que, además, concuerde la fecha de facturación con el campo de texto de fecha de facturación.
21. El sistema deberá mostrar un mensaje indicando que albaranes han sido facturados y, además, generar un reporte con todos los albaranes facturados (imprimir la factura).
22. El sistema deberá generar diferentes reportes de los diferentes facturas y facturas nacionales según el botón que se use en la ventana de facturas nacionales.
23. El sistema contara con una ventana de gestión de abonos de los clientes mostrando los diferentes abonos creados y la lista de detalles de estos abonos.
24. Dado un código de cliente activo, se cargará las bonificaciones/devoluciones de ese cliente.
25. Dado un código de Abono, cargara los datos del cliente y los detalles de ese abono.
26. En la creación de un abono, se podrán introducir líneas del abono, estas líneas pueden contener piezas concretas de algún albarán o algún elemento que cumpla con las características ingresada y se encuentre en algún albarán.
27. Crear un abono a partir de los datos de las líneas ingresadas en el abono.
28. Modificar los datos de un abono creado, llámese agregar más líneas o la fecha del abono.
29. Dado un abono seleccionado, se podrá "eliminar" este abono de la BBDD.

En el documento de requerimientos se explica de manera más detallada todos los elementos mencionados aquí, incluyendo sub-requerimientos de estos e indicando restricciones, dependencias, etc.

Requerimiento Nº	REQ1		
Descripción	El programa deberá tener una página de gestión de albaranes (creación, edición, vista y borrado). Que, además, muestre los detalles de los albaranes.		
Tipo de requerimiento	Funcional	X	No funcional:
Restricciones			
Requerimiento Padre:			

Fig. 9. Ejemplo de requerimiento del documento. El código del requerimiento encaja con el número mostrado en este documento.

6 RESULTADOS:

El Proyecto se ha logrado desarrollar gracias a diferentes técnicas y tecnologías aprendidas durante los años de carrera. Estos conocimientos han servido mucho como base para poder entender todas las tecnologías y herramientas presentadas por la empresa para satisfacer las necesidades del cliente. Siendo más concretos, se ha podido completar todos los requerimientos para este TFG y dar un producto apropiado al cliente. El proyecto desarrollado ha logrado completarse hasta los límites requeridos por este TFG y avanzado bastante más respecto a los hitos establecidos en este TFG.

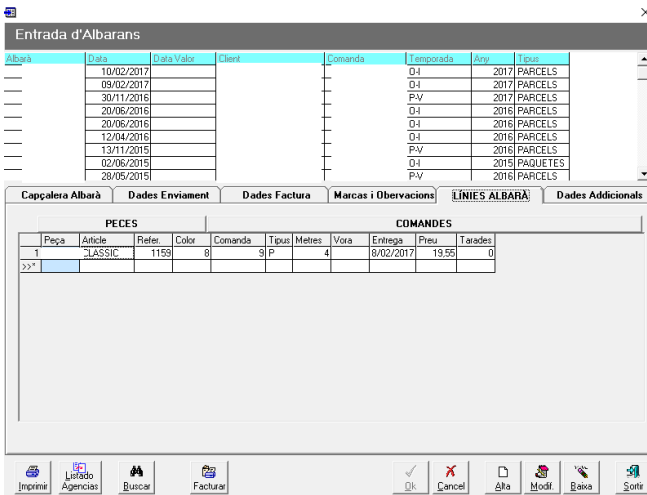


Fig. 10. Vista antigua de la ventana de creación de albaranes.

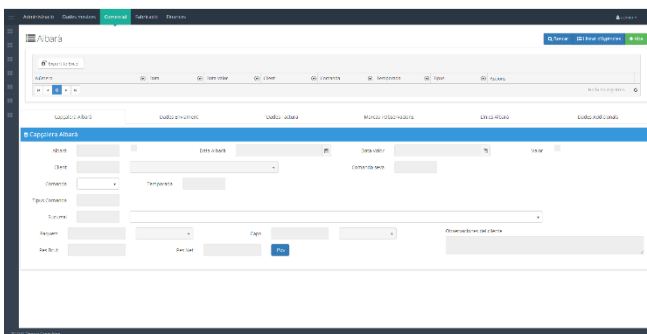


Fig. 11. Vista actual de la ventana de creación de albaranes.

Se ha logrado darle el lavado de cara que la aplicación estaba necesitando y un nuevo entorno para trabajar, aunque manteniendo su esencia para que el cliente se sienta a gusto con el nuevo entorno y siga en un ambiente conocido para ellos, corrigiendo muchos errores de la aplicación anterior como crasheos sin motivo alguno además de añadirle funcionalidades nuevas a la aplicación o eliminando pasos innecesarios para agilizar la creación y el trabajo en el mismo.

También se ha ido reestructurando la BBDD anterior para que trabaje de manera más eficiente y dejando más claro la procedencia de los datos, dado que hoy en día como está hecha la BBDD anterior es todo un caos que permite hacer cosas imposibles (como repetir posibles códigos) dando como resultado final una BBDD basada en la anterior, conservando algunos enlaces y creando tablas nuevas y agregando o quitando (muchos) campos que ahora ya no son necesarios o pueden ser vistos como una tabla nueva. Además, al actualizar el tipo de BBDD que usa también se optimiza las búsquedas y los tiempos de respuesta.

7 CONCLUSIONES

Este proyecto de TFG me ha servido para mi crecimiento como profesional. No solo he aprendido a como es el entorno de trabajo y desarrollo en la vida real si no que, además, me ha enseñado un montón de conceptos de cómo funcionan los sistemas facturación de muchas empresas y como se estructura y forma un ERP desde 0.

Conociendo las herramientas que Telerik ofrece he podido lograr nuevas funcionalidades y comprender que no todo lo que se va a implementar en un desarrollo de software tiene que ser creado desde el absoluto 0, si ya existe una herramienta eficaz que sabes que da resultado y cumple con creces las necesidades actuales, es preciso hacer uso de esta y además nutrirse del cómo funciona y que puedes aprender de ella para futuros desarrollos.

Cabe destacar también que el desarrollo ágil no se puede implementar en todos los proyectos. En este proyecto en particular teníamos la fortuna de contar con la cercanía del cliente y su disposición de tiempo para lograr sacar el desarrollo adelante, aunque en más de 1 ocasión era común notarse que los tiempos para todos nos aprietan y que tanto el cliente como nosotros los desarrolladores tenemos que disponer de un amplio tiempo para que estos proyectos desarrollados con esta metodología puedan ser llevados a cabo.

Ha sido de total importancia los conocimientos adquiridos durante la carrera, al menos para comprender los esquemas básicos del trabajo, el cómo pueden funcionar las herramientas, que tipo de procesos deben ser usados para desarrollar un proyecto de software y mucho más.

Lo último que quiero nombrar en estas conclusiones es lo vital que es para el desarrollo que un software se

encuentre debidamente documentado o cuente con algún tipo de soporte, dado que nosotros al únicamente tener un código fuente y una descripción de cuáles son los resultados que da el programa, no es suficiente para poder rehacer un programa ya hecho, hasta cierto punto he llegado yo a pensar que era mejor tirar todo lo que hacía el programa antes y rehacerlo según los requisitos del cliente, pero como esto no es algo habitual en las empresas y además es un coste muy elevado, puedo entender que se haya usado todo lo anterior como fuente, pero entender todo el código sin ningún tipo de documentación es una pesadilla.

8 LÍNEAS FUTURAS

El proyecto como tal es una pequeña parte de un gran sistema ERP que hay que desarrollar, hay módulos a futuro que hay que terminar de implementar y seguramente haya que volver a esta parte del módulo a realizar algún pequeño cambio ya sea visual o añadir nuevas funcionalidades y posiblemente, corregir alguna funcionalidad implementada.

Actualmente el proyecto ha sido acabado en codificación y se ha llegado a lograr mucho más a partir de lo desarrollado aquí (por ejemplo, se ha hecho un módulo de finanzas que contempla las facturas aquí hechas y permite realizar otras funcionalidades e integración con otro sistema para las finanzas).

Afortunadamente al ser un programa modular y escalable (como casi todo ERP), se puede acabar esta parte del módulo y ponerla en producción para que el cliente pueda empezar a darle uso y familiarizarse con el programa final.

AGRADECIMIENTOS

Primero que todo quiero agradecer a mi padre, Edgar Vargas Villegas, por haberme inculcado el camino del estudio y enseñarme que lo más valioso en la vida siempre será la salud y el conocimiento, desde que uno cuente con esas 2 cosas todo lo demás ira llegando.

También quiero hacer especial mención a mi madre, Maria Peña, que gracias a sus esfuerzos durante unos difíciles años aquí en España logro hacerse con la nacionalidad española y gracias a ello puedo estar yo aquí escribiendo este trabajo.

A mi compañera, amiga, y actual pareja, Alejandra Rocha Ronderos, por haberme enseñado el camino necesario para poder optar a inscribirme a esta universidad y, de no ser por ella, no habría tomado el paso para tomar la decisión de arriesgarme a la aventura en el extranjero.

A mi tutor, Lluís Gesa bote, por todo el apoyo que mostro durante el desarrollo de este TFG. Creo que no le hago honor a la atención que el da como tutor de un proyecto, me parece que es alguien que se preocupa por los mismos y este tipo de profesores es difícil de encontrar.

BIBLIOGRAFÍA

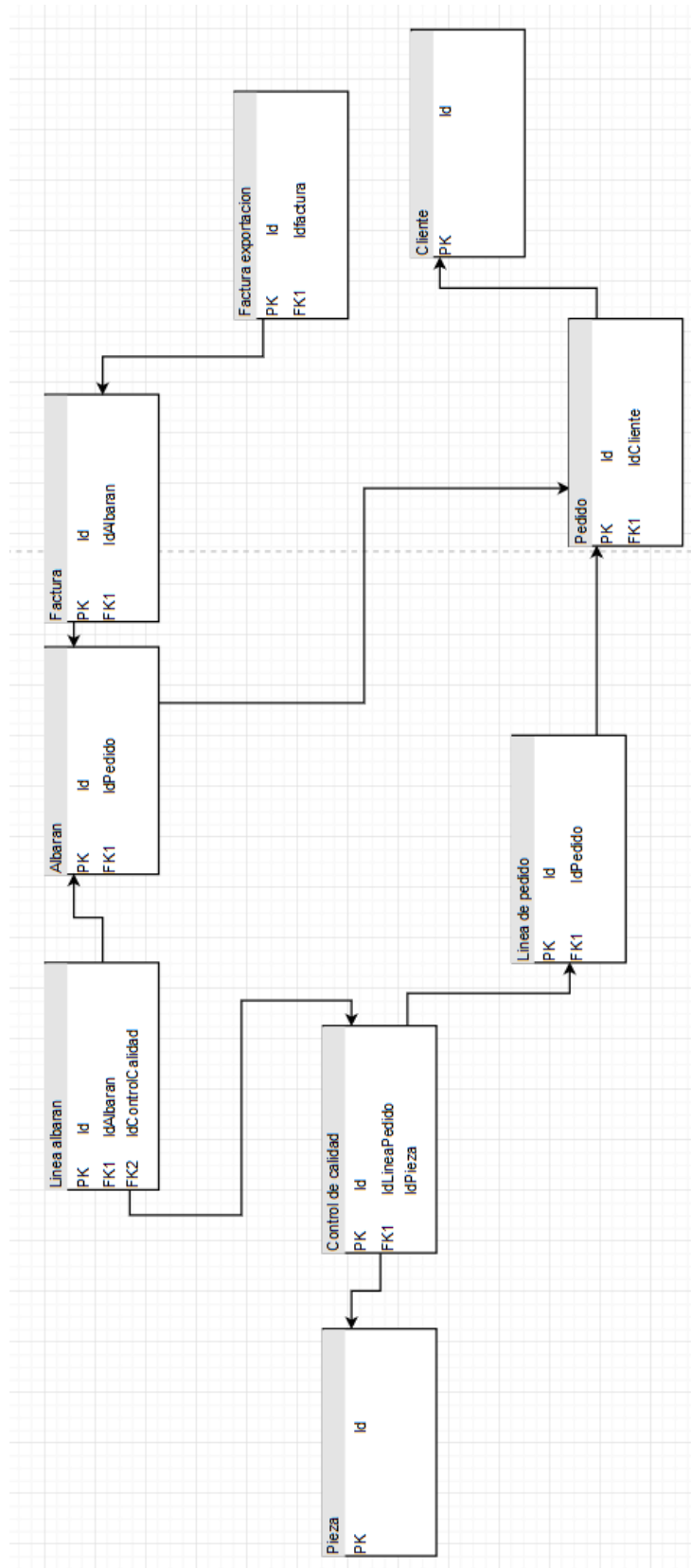
- [1] "Que es un ERP ", Por Einatec Consulting. Disponible en: <https://einatec.com/que-es-un-erp/>.
- [2] "Que es un sistema ERP" por Tic Portal. Disponible en: <https://www.ticportal.es/temas/enterprise-resource-planning/que-es-sistema-erp>.
- [3] "What is a ERP" Por Wally Tonra. Disponible en: <https://www.xtuple.com/blog/wtonra/what-is-erp>.
- [4] "2017 Report on ERP systems & Enterprise software", por Panorama Consulting solutions. <https://go.panorama-consulting.com/rs/603-UJX-107/images/2017-ERP-Report.pdf>.
- [5] "Programas ERP: Empezar el proceso de selección de software" por Tic Portal. Disponible en: <https://www.ticportal.es/temas/enterprise-resource-planning/programas-enterprise-resource-planning-erp>.
- [6] "What is Agile software Development" por Scott D'Ambra Disponible en: <https://www.clear.com/what-is-agile-software-development.html>.
- [7] "State of Agile: Still Scrumming After All These years" por Lee Cunningham. Disponible en: <https://blog.versionone.com/state-of-agile-report-scrum-methodology/>.
- [8] Luis_calvo, "Grids", MDN web docs, 27 sept. 2016, Disponible en : https://developer.mozilla.org/es/docs/Learn/CSS/CSS_layout/Grids.
- [9] Administrador, "Kendo-UI", Telerik, Fecha desconocida, Disponible en: <https://www.telerik.com/kendo-ui>.
- [10] Administrador, "Telerik Reporting", Telerik, Fecha desconocida, Disponible en: <https://www.telerik.com/products/reporting.aspx>.
- [11] Administrador, "VisualStudio ", Microsoft, Fecha desconocida, Disponible en: <https://visualstudio.microsoft.com/es/>.
- [12] Administrador, "Git", Git, Fecha Desconocida, Disponible en: <https://git-scm.com/>.

APÈNDIX

A1. PAGINA ACTUAL DE ALBARANES

The screenshot displays the Albarà web application interface. At the top, there is a navigation bar with the following menu items: Administració, Dades mestres, Comercial (highlighted), Fabricació, and Finances. The user is logged in as 'admin'. Below the navigation bar, there is a search bar with the text 'Capçalera Albarà' and a search icon. To the right of the search bar, there are several filter buttons: 'Export to Excel', 'Número', 'Data', 'Data Valor', 'Client', 'Comanda', 'Temporada', 'Tipus', and 'Accions'. A dropdown menu is open, showing 'No hi ha registres.' Below the search bar, there is a form for entering data. The form is divided into several sections: 'Dades Enviament' (with fields for 'Albarà', 'Client', 'Comanda', 'Tipus Comanda', and 'Sucursal'), 'Dades Factura' (with fields for 'Data Albarà', 'Data Valor', and 'Comanda seva'), 'Marcas i Observacions', 'Línies Albarà', and 'Dades Addicionals'. The 'Dades Enviament' section includes a 'Temporada' dropdown and a 'Pes' button. The 'Dades Factura' section includes a 'Data Valor' dropdown and a 'Pes' button. The 'Dades Addicionals' section includes a 'Pes Net' field and a 'Pes' button. The 'Línies Albarà' section includes a 'Valor' field. The 'Dades Addicionals' section includes a 'Observacions del client' field. At the bottom right, there is a footer with the text '2019 © Enracc Consulting'.

A2. FRAGMENTO DE LA BBDD DONDE SE TRABAJO



A3. DIAGRAMA DE TIEMPO ESPERADO

