



UNIVERSIDAD DE CÓRDOBA

ESCUELA POLITÉCNICA
SUPERIOR DE CÓRDOBA
Universidad de Córdoba



Proyectos

Evaluación técnica de un Trabajo Fin de Grado

Autor del Manual técnico

Luis Flores Méndez

Autor de este documento

Juan José Méndez Torrero

4º Grado en Ingeniería Informática.

12 de enero de 2019

Índice

1. Introducción	5
2. Análisis de las partes del proyecto	6
2.1. Introducción	6
2.2. Identificación del problema	6
2.3. Objetivos	7
2.4. Antecedentes	8
2.5. Restricciones	8
2.6. Recursos	8
2.7. Especificación de requisitos	9
2.8. Especificación del Sistema	13
2.9. Diseño y pruebas	14
2.10. Conclusiones	14
2.11. Bibliografía	15
2.12. Anexos	15
3. Análisis del diseño del documento	16
4. Conclusiones personales	17

Índice de figuras

1.	Introducción	6
2.	Objetivos	7
3.	Antecedentes	8
4.	Introducción apartado Especificación de requisitos	9
5.	Diagrama UML del usuario de la aplicación.	10
6.	Tablas informativas de la Funcionalidad General.	11
7.	Descripción de la clase Tarea.	12
8.	Explicación de la relación entre la clase Usuario y la clase Tarea. . . .	12
9.	Explicación del diseño de la aplicación.	13
10.	diagrama de despliegue de la conexión Aplicación-Servidor	14
11.	Bibliografía del Manual Técnico.	15

1. Introducción

En el presente documento se hará una evaluación técnica a un Trabajo Fin de Grado cuyo objetivo es crear un CRM (Customer Relationship Management). El documento estará dividido según las partes que todo manual técnico de un proyecto de ingeniería debería de tener, que son las siguientes: *Introducción, Identificación del problema, Objetivos, Antecedentes, Restricciones, Recursos, Especificación de requisitos y de interfaz, Diseño y Pruebas, Conclusiones, Bibliografía y Anexos*.

Para toda explicación se adjuntará una Figura en la que se podrá observar un ejemplo de ésta. Finalmente, habrá un apartado para las conclusiones propias del proyecto en general.

2. Análisis de las partes del proyecto

En esta sección realizaremos un análisis parte por parte a todo el proyecto.

2.1. Introducción

Como ya sabemos, las introducciones han de ser siempre concisas y muy explicativas, siendo correcto el refrán *Lo bueno, si breve, dos veces bueno*, pues bien, el autor de este manual técnico nos introduce el problema de manera que el lector se pueda hacer una idea del por qué de la creación de la aplicación. Además, el autor explica, de forma general, una solución a dicho problema introductorio. En la Figura 1 se puede observar la introducción del manual técnico.

En el mundo empresarial actual es necesario tener una buena relación con los clientes, ya sea en la venta o en la postventa, pues son los que van a mantener y hacer crecer a la empresa en concreto. El desarrollo de los sistemas de información, las telecomunicaciones y la tecnología en general ha permitido facilitar el mantenimiento y mejora de estas relaciones, así como una mejor gestión de los recursos y de la producción de la empresa.

Aunque hay diversos dispositivos y aplicaciones que tienen diversas funcionalidades para ayudar en el proceso de desarrollo de una empresa, hay un tipo de aplicación que es esencial en cualquier empresa y que sirve de base para la informatización íntegra de la gestión de los clientes de la empresa, son los llamados CRM (Customer Relationship Management).

Los CRM han ido evolucionando desde su aparición hasta el punto en el que no sólo ayudan a tener un mejor control sobre la relación con los clientes, si no que se encargan de otros aspectos importantes de la empresa, como la gestión de reuniones, proyectos y su subdivisión en tareas (junto con su asignación), control de inventarios y gestión del calendario para todos los empleados.

Figura 1: Introducción

Además, el autor de este proyecto habla sobre la historia de la creación de los CRM, siendo esto algo esencial para realizar una buena introducción para que el lector se entere de por qué se crearon los CRM. Junto con esto, el autor explica los diferentes tipos de CRM que existen y las diferencias existentes entre éstos.

2.2. Identificación del problema

Ahora, en esta sección, el autor comienza a realizar una definición tanto técnica como funcional del problema además de las especificaciones necesarias para hacer frente a este problema.

El autor acierta a la hora de crear las distintas secciones de este apartado, ya que ha diferenciado muy bien lo que sería el problema real del problema técnico. Además, el usuario también redacta el ciclo del mantenimiento de la aplicación junto

con la vida esperada de ésta.

Por último, el autor realiza un resumen de lo que será la definición de los objetivos, recursos necesarios, especificaciones de requisitos, diseños y pruebas.

2.3. Objetivos

En esta sección, el autor comienza explicando que, una vez establecidos los problemas reales y los técnicos, han surgido varios problemas que han llegado a convertirse en objetivos del proyecto. Además, éstos son explicados de manera clara y concisa como se puede observar en la Figura 2.

- **Módulo Sistema General:** Es el módulo principal de la aplicación. Gestiona las cuestiones generales de la aplicación y es el encargado de proporcionar acceso a la información de todos los sujetos contenidos en la aplicación, así como la modificación de esta información y la agregación y eliminación de sujetos. Estos sujetos se dividirán en clientes y trabajadores de la empresa. Estos últimos son los que tendrán distintos roles asignados en la base de datos y en base a estos podrán acceder a unas funcionalidades o a otras.
- **Módulo de Proyectos:** Permite al usuario gestionar los proyectos y las tareas de la empresa. Esto incluye la creación y eliminación de proyectos, dividirlos en tareas y asignar éstas a los trabajadores. Junto con las asignaciones se pueden introducir fechas y horas de trabajo, añadiéndolas automáticamente a los calendarios de los trabajadores para que estos puedan recibir alertas y estar al tanto de las *deadlines*.
- **Módulo de CRM:** Es el módulo que se encarga de realizar la función de CRM como tal. Otorga las funcionalidades de almacenar, eliminar y consultar información sobre los clientes que tiene la empresa y registrar las conversaciones que se mantienen con los clientes (bien por teléfono o bien por e-mail).
- **Módulo de Recursos:** Es el encargado de registrar las horas de trabajo que realizan los trabajadores de la empresa y del registro y asignación de tareas, así como de mostrar toda la información referente a estos registros.
- **Módulo de Autenticación:** Este módulo proporcionará el control de acceso y conectividad a la base de datos y a la aplicación. En este módulo se integrará la API de Azure Active Directory y se controlarán los datos de conexión a la base de datos.
- **Vistas:** Controla la forma en la que se ven las diferentes tablas que se van mostrando en la aplicación con forme se va navegando y haciendo uso de ella, además del cambio de apariencia de la aplicación.

Figura 2: Objetivos

Como vemos, el autor ha sido capaz de transmitir los objetivos del proyecto para que una persona no experta pueda entender claramente y sin confusión los objetivos.

2.4. Antecedentes

En los antecedentes, el autor explica cómo ha surgido la idea de esta aplicación, siendo la razón la necesidad del cliente de cambiar la tecnología usada por su aplicación. En este caso, el autor explica el funcionamiento de la aplicación anterior del cliente junto con la herramienta usada para la creación de la nueva.

El autor de este proyecto termina esta sección justificando el por qué ha usado la herramienta utilizada en vez de otras. En definitiva, el autor realiza una buena descripción de los antecedentes ya que el lector se puede enterar de cómo ha surgido la idea de este proyecto. En la Figura 3 se puede apreciar la justificación del uso de estas herramientas.

La empresa Tecnube, como ya se ha dicho anteriormente, desea actualizar su aplicación a la tecnología WPF, ya que la actual, que usa WinForms, podría quedarse obsoleta en un futuro no muy lejano y podría no ser capaz de introducir funcionalidades nuevas con la misma facilidad con la que se podría conseguir con WPF.

Además, WPF permite separar el diseño de la interfaz del código de la aplicación, permitiendo dividir el trabajo entre dos equipos diferentes, para luego juntarlos y tener la aplicación completa. Así, se podría cambiar la interfaz entera sin modificar el código que da las funcionalidades de la aplicación. También se siguen actualizando las diferentes APIs y desarrollando otras nuevas, tanto por parte de Microsoft como de terceros, que abren el abanico de posibles nuevas funcionalidades que se pueden introducir en la solución CRM.

Figura 3: Antecedentes

2.5. Restricciones

En esta sección, el autor se debe de encargar de explicar todas y cada una de las restricciones que tendrá el cliente cuando utilice su aplicación, habiendo que definir los factores dato y estratégicos. Los primeros no pueden ser modificados mientras que los segundos sí. En este caso, el autor ha realizado una explicación de la definición de estos factores y seguidamente ha establecido cuáles son.

En este caso, el autor acierta redactando los factores ya que, por ejemplo, la restricción de acceso del cliente ha de ser un factor dato ya que no puede ser modificada durante el transcurso del proyecto. Para los factores estratégicos, también ha acertado ya que, por ejemplo, que el usuario pueda descargarse unas tablas.

2.6. Recursos

En esta sección, el autor del proyecto ha de explicar tanto los recursos que ha utilizar por su aplicación, como los recursos utilizados para su creación. En este caso, el

autor ha dividido los recursos en dos partes, los recursos para el desarrollo y otra para los recursos para la utilización, con lo que en este caso ha vuelto a acertar el autor a la hora de definir los recursos.

Un ejemplo es que en los recursos para la utilización, redacta claramente las especificaciones mínimas necesarias para poder usar la aplicación. Otro ejemplo, en los recursos para el desarrollo, es que explica cuáles son los recursos humanos que han sido necesarios para el desarrollo de la aplicación.

2.7. Especificación de requisitos

En esta sección, la labor del autor es identificar todas y cada una de las especificaciones de su proyecto desde el punto de vista técnico.

En el caso de nuestro documento, el autor en la introducción de este apartado, dice: Figura 4.

En este capítulo se especificará técnicamente “qué debe hacer” el sistema. La especificación de requisitos (ERS) se apoya en los datos obtenidos durante el análisis. Esta es una de las fases más importantes en el desarrollo de un producto software ya que, en gran medida, determinará la calidad del producto final.

Figura 4: Introducción apartado Especificación de requisitos

Además, el autor identifica cómo interactúa el cliente con la aplicación de una manera muy intuitiva, como se puede ver en la Figura 5. Como vemos, está creando los diagramas con un programa llamado UML, el cuál hace que crear estos diagramas sea muy fácil y que el resultado final sea fácil de entender para el lector.

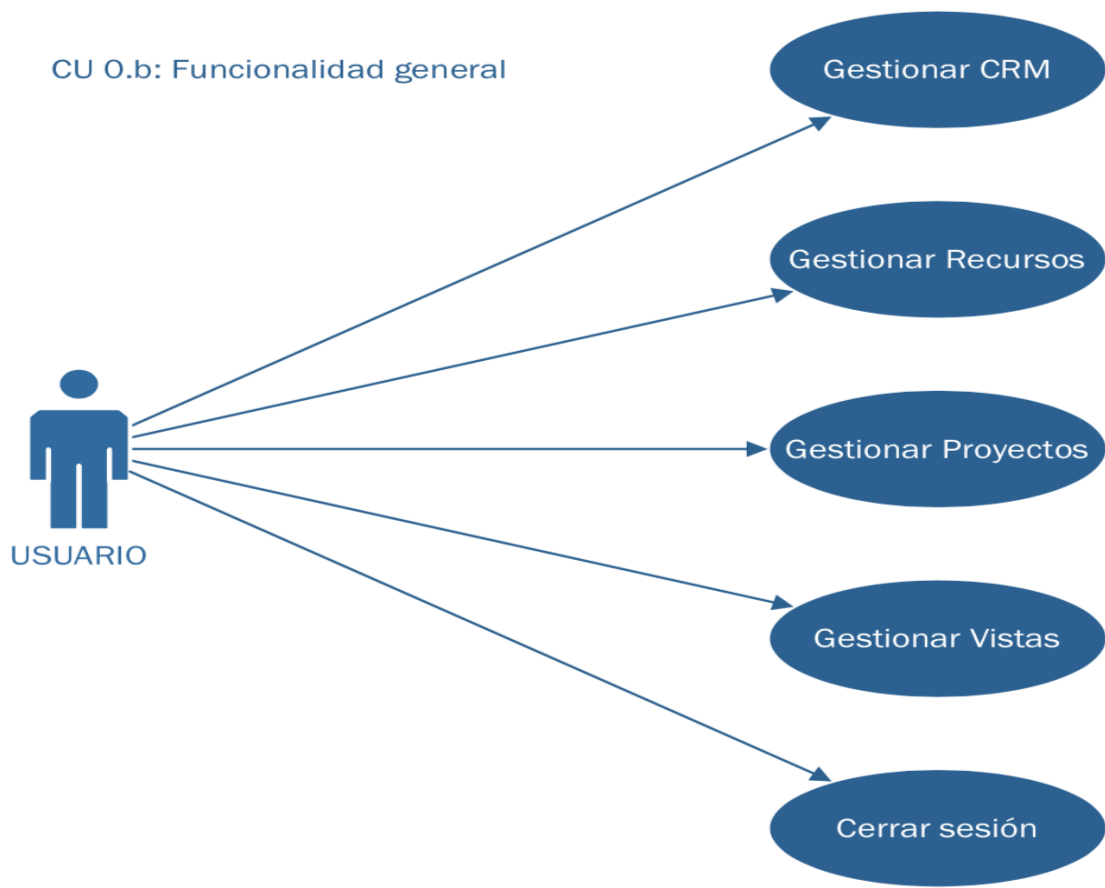


Figura 5: Diagrama UML del usuario de la aplicación.

Junto con esto, podemos observar, por cada especificación, una tabla informativa en la que explica qué hacen y qué necesitan. Figura 6.

	CU 0.a - Funcionalidad general
Objetivo	Mostrar la ventana principal de la aplicación
Acción	El se identifica en la aplicación
Tipo	Primario
Precondiciones	El usuario debe haberse autenticado correctamente en la aplicación
Descripción	Se muestra la aplicación y sus funcionalidades, pudiendo realizar acciones de los diversos módulos de la aplicación.

Tabla 7.3: CU 0.b – Funcionalidad general

	CU 1 - Gestionar CRM
Objetivo	Mostrar las funcionalidades de CRM de la aplicación
Acción	El usuario accede a la pestaña de CRM de la aplicación
Tipo	Secundario
Precondiciones	Estar identificado en el sistema
Descripción	Se muestran en la barra superior las diferentes funcionalidades de CRM de la aplicación

Tabla 7.4: CU 1 - Gestionar CRM

Figura 6: Tablas informativas de la Funcionalidad General.

El autor después de haber declarado la descripción general del problema, comienza a describir la información relativa al proyecto, como qué clases ha creado junto con los parámetros pertenecientes a éstas. En la Figura 7 podremos ver

7.4.1.6 Clase Tarea

Atributos

- **nombre:** Cadena que almacena el nombre de la tarea.
- **id:** Valor numérico que identifica inequívocamente a una tarea.
- **idUser:** Valor numérico que identifica al usuario al que está asignada una tarea.
- **fecha:** Fecha que indica la fecha límite de la tarea.
- **tareas:** Vector de valores numéricos que identifican las tareas en las que se divide la tarea, ya que una tarea de un proyecto se puede dividir en tareas más simples.

Métodos

- **Tarea(nombre, id, idUser, ?tareas):** Es el constructor de la clase. Genera una instancia para representar al usuario conectado al sistema en este momento.
- **asignarTarea(int id):** Es la función que asigna una tarea a otra, introduciendo su id en el vector *tareas*.
- **registrar():** Registra la tarea en la base de datos.
- **eliminar():** Elimina la tarea de la base de datos.

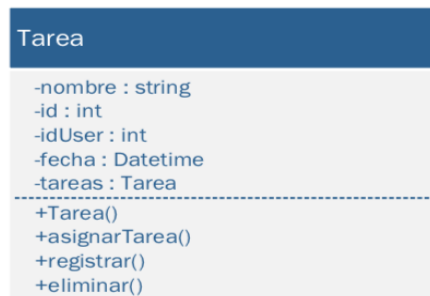


Figura 7.19: Clase Tarea

Figura 7: Descripción de la clase Tarea.

Por último, el autor declara la relación de cada una de las clases identificadas en la descripción de la información. Ver Figura 8.

7.4.2.4 Asociación Usuario-Tarea

Este tipo de asociación establece la relación entre un usuario y una tarea que él crea.

Un usuario puede poseer más de una tarea, o incluso ninguna, y una tarea sólo puede ser creada por un usuario. Por lo tanto, la clase Usuario interviene con cardinalidad mínima y máxima 1, mientras que la clase Tarea interviene con la cardinalidad mínima 0 y la máxima n.

Figura 8: Explicación de la relación entre la clase Usuario y la clase Tarea.

2.8. Especificación del Sistema

Esta sección, que el autor denomina *Especificación de la interfaz*, hay que explicar cómo se va a abordar el diseño del proyecto teniendo en cuenta la especificación de requisitos anteriormente descrita.

En nuestro caso, el autor realiza una introducción del diseño estético de la aplicación, junto con la explicación de cómo será el diseño. Ver Figura 9.

- Una **ventana principal** en la que se haría prácticamente todo el trabajo. Esta ventana deberá tener dos zonas diferenciadas: una barra dividida en pestañas por categorías, incluyendo en ellas las funcionalidades de los módulos de la aplicación y una zona central, que ocuparía el resto de la ventana, que será el área de trabajo.
- **Ventanas secundarias** auxiliares, que aparecerían en momentos determinados del uso de la aplicación. Estas ventanas se usarán para editar los datos de la aplicación, mostrar mensajes de error o información sobre ésta.

Figura 9: Explicación del diseño de la aplicación.

2.9. Diseño y pruebas

Hasta ahora, el autor ha descrito todas las necesidades y tareas que debe de tener la aplicación. Ahora, el autor comienza a describir unas tablas en las que explica la relación entre el servidor y su aplicación, añadiendo incluso un diagrama esquemático para que el lector pueda hacerse una idea de cómo se reciben los datos desde un servidor. Ver Figura 10.

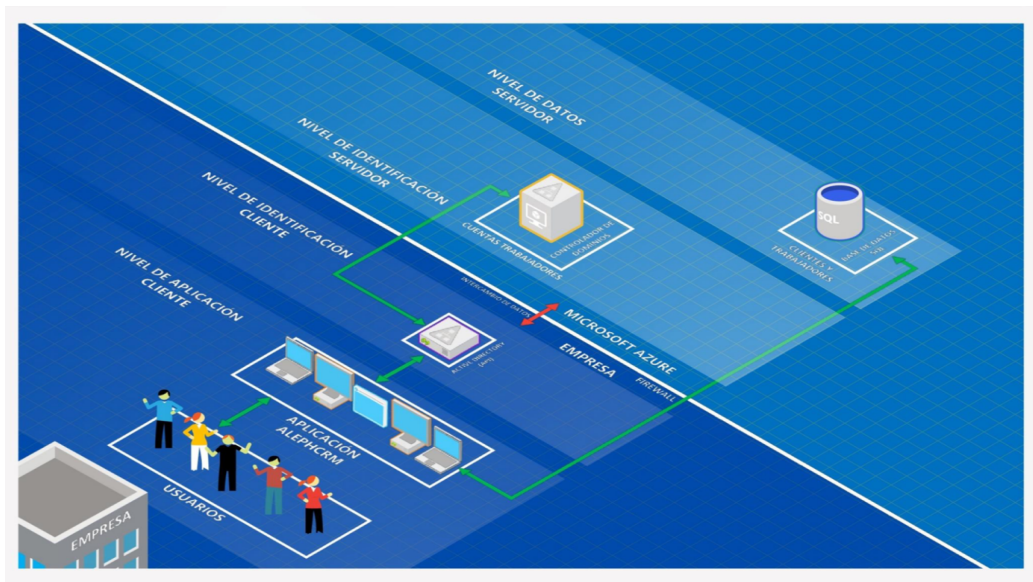


Figura 10: diagrama de despliegue de la conexión Aplicación-Servidor

El autor, esta sección de diseño ha querido dividirla en cinco partes:

1. Diseño de datos.
2. Diseño arquitectónico.
3. Diseño procedimental.
4. Diseño de la interfaz.

Finalmente, acaba con la sección de pruebas, en la que comienza a comprobar que la aplicación funcione correctamente y que el diseño sea el correcto.

2.10. Conclusiones

En este apartado, el autor debe de expresar las conclusiones finales propias con respecto a las pruebas realizadas, incluyendo algunas de las futuras mejoras que se podrían hacer para que el lector de dicho proyecto pueda empezar algo con ayuda de la aplicación descrita.

En nuestro caso, el autor del proyecto divide esta sección en tres partes, las cuales son las conclusiones generales de la aplicación, futuras mejoras y conclusiones personales.

2.11. Bibliografía

En esta sección el autor del proyecto a de listar todas las referencias bibliográficas utilizadas para la realización del proyecto que aparezcan en el Manual Técnico. En nuestro caso, el autor se decanta por una bibliografía numérica, en la que a lo largo de todo el texto podamos ir a la referencia de la misma. Ver Figura 11.

- [1] «Wikipedia - Burbuja puntocom,» [En línea]. Available: https://es.wikipedia.org/wiki/Burbuja_punto_com.
- [2] “CRM Switch,” [Online]. Available: <https://www.crmswitch.com/crm-industry/crm-industry-history/>.
- [3] «CRM Zazachat,» [En línea]. Available: http://www.zazachat.com/kb/crm/articles/crm_history.aspx.

Figura 11: Bibliografía del Manual Técnico.

2.12. Anexos

Aquí, el autor puede poner información adicional que tenga relación con le proyecto. En nuestro caso, el autor incluye información sobre *Windows Presentation Foundation (WPF)*

3. Análisis del diseño del documento

En esta sección intentaremos dar un análisis del Manual Técnico desde el punto de vista del diseño. Esta sección se dividirá en puntos a favor y en puntos en contra:

1. Puntos a favor:

- Portada sencilla, con los símbolos de su universidad y de su escuela. Además cuenta con la información personal, como nombre y apellidos, grado, especialización, director/es del proyecto y un título identificativo.
- Deja todas las portadas en las páginas impares, dejando las pares en blanco.
- No he encontrado faltas de ortografía.
- Incluye un anexo al final.

2. Puntos en contra:

- Empieza a enumerar el proyecto desde el índice de contenidos, no desde la portada del primer capítulo.

4. Conclusiones personales

En mi opinión personal el documento de manual técnico describe a la perfección todo lo necesario para poder realizar alguna futura mejora en él. Además, el autor divide correctamente cada una de las partes, incluso añadiendo un anexo al final. Las tablas de índices y figuras las ha escrito al final, no estando mal pero que a mi parecer debería de haber colocado al principio, ya que así se podría ir más rápidamente a cualquier tabla y figura.

Los programas usados para la creación del manual han sido acordes a lo que las explicaciones necesitaban. Como hemos comentado, para la explicación de las especificaciones ha utilizado un programa llamado UML, lo cual lo considero algo muy positivo.