



UNIVERSIDAD DE CÓRDOBA



**ESCUELA POLITÉCNICA SUPERIOR**

**Grado en Ingeniería Informática – Rama de Computación**

**Septiembre 2016**

Aplicación para la gestión de los recursos humanos empresariales (CRM) utilizando la tecnología WPF

---

MANUAL TÉCNICO

**Autor: Luis Flores Méndez**

DIRECTORES: D. ENRIQUE GARCÍA SALCINES | D. FRANCISCO MÉNDEZ SERRANO



Don ENRIQUE GARCÍA SALCINES, Profesor Ayudante del  
Departamento de Informática y Análisis Numérico de la Universidad de  
Córdoba

y

Don FRANCISCO MÉNDEZ SERRANO, Director técnico en sistemas y  
desarrollo de la empresa Tecnologías y Soluciones en la Nube, S.L.U.

CERTIFICAN

Que el presente proyecto fin de carrera titulado: **Aplicación para la gestión de los recursos humanos empresariales (CRM) utilizando la tecnología WPF**, ha sido realizado bajo nuestra dirección por Luis Flores Méndez, cumpliendo, a nuestro juicio, con los requisitos exigidos en este tipo de proyectos.



**Fdo: Enrique García Salcines      Fdo: Francisco Méndez Serrano**

**Córdoba, Septiembre de 2016**



ALUMNO AUTOR DEL PROYECTO

Fdo. Luis Flores Méndez

Córdoba, Septiembre 2016



# Contenido

---

Contenido .....	1
1 Introducción.....	7
1.1 Introducción .....	7
1.2 Evolución del CRM.....	7
1.3 Módulos del CRM.....	9
1.4 Tipos de CRM .....	10
1.4.1 Operacional.....	10
1.4.2 Analítico .....	10
1.4.3 Colaborativo.....	11
2 Identificación del problema .....	12
2.1 Introducción .....	12
2.2 Identificación del problema real .....	12
2.3 Identificación del problema técnico.....	13
2.3.1 Funcionamiento .....	13
2.3.2 Entorno .....	14
2.3.3 Vida esperada.....	15
2.3.4 Ciclo de mantenimiento.....	15
2.3.5 Competencia .....	15
2.3.6 Aspecto externo.....	16
2.3.7 Estandarización .....	16
2.3.8 Calidad y fiabilidad .....	16
2.3.9 Programa de tareas.....	16
2.3.10 Pruebas .....	18
2.3.11 Seguridad .....	18
3 Objetivos.....	19
4 Antecedentes.....	20
4.1 AlephCRM en WinForm.....	20
4.2 Microsoft Dynamics CRM.....	21
4.3 Justificación del proyecto.....	22
5 Restricciones .....	23
5.1 Introducción .....	23
5.2 Factores dato.....	23
5.3 Factores estratégicos .....	23
6 Recursos.....	24

6.1	Introducción .....	24
6.2	Recursos necesarios para el desarrollo .....	24
6.2.1	Recursos humanos .....	24
6.2.2	Recursos hardware .....	24
6.2.3	Recursos software .....	25
6.3	Recursos necesarios para la explotación .....	25
6.3.1	Recursos hardware .....	25
6.3.2	Recursos software .....	25
7	Especificación de requisitos .....	29
7.1	Introducción .....	29
7.2	Descripción general del problema .....	29
7.3	Descripción funcional .....	30
7.3.1	Identificación de los casos de uso .....	30
7.4	Descripción de la información .....	42
7.4.1	Análisis de las clases .....	43
7.4.2	Análisis de las asociaciones .....	49
7.4.3	Diagrama de clases .....	51
8	Especificación de la interfaz .....	52
8.1	Introducción .....	52
8.2	Características de la interfaz .....	52
9	Diseño de datos .....	57
9.1	Introducción .....	57
9.2	Definición de las tablas .....	57
9.3	Esquema relacional .....	59
10	Diseño arquitectónico .....	63
10.1	Introducción .....	63
10.2	Diagrama de despliegue .....	63
11	Diseño procedimental .....	65
11.1	Introducción .....	65
11.2	Diagramas de actividad .....	65
11.2.1	Funcionalidad principal del sistema .....	66
11.2.2	Autenticarse .....	67
11.2.3	Fichar .....	68
11.2.4	Gestionar CRM .....	69
11.2.5	Gestionar Recursos .....	71



11.2.6	Gestionar Proyectos.....	73
11.2.7	Gestionar Vistas .....	75
11.2.8	Cerrar sesión .....	75
11.2.9	Generar nuevo elemento.....	76
12	Diseño de la interfaz.....	77
12.1	Introducción .....	77
12.2	Ventana Principal .....	77
12.3	Ventanas secundarias .....	79
12.3.1	Editores .....	79
12.3.2	Mensajes.....	80
12.3.3	Información.....	80
12.4	Ventana de Inicio de sesión .....	81
13	Pruebas.....	85
13.1	Introducción .....	85
13.2	Enfoque estructural.....	85
13.3	Enfoque funcional .....	86
13.3.1	Primera Fase: Acceso al sistema .....	86
13.3.2	Segunda Fase: Interfaz.....	86
13.3.3	Tercera Fase: Operaciones.....	87
14	Conclusiones y futuras mejoras .....	91
14.1	Introducción .....	91
14.2	Conclusiones .....	91
14.3	Futuras mejoras .....	92
14.4	Conclusiones personales.....	92
	Bibliografía .....	93
14.5	Introducción .....	93
14.6	Referencias bibliográficas – Manual técnico.....	93
	Índice de tablas.....	95
	Índice de figuras.....	96
	Anexo I – Windows Presentation Foundation .....	97
	Introducción.....	97
	WPF.....	97



# SECCIÓN I

# PRESENTACIÓN



# 1 Introducción

---

## 1.1 Introducción

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.

## 1.2 Evolución del CRM

El CRM no siempre ha sido un software tan completo e íntegro como se le conoce hoy en día. Hay que remontarse a los años 80 para conocer el inicio de este tipo de software empresarial. A lo largo de los años se han ido desarrollando diferentes soluciones software para la empresa, que han ido evolucionando hasta lo que es el CRM actual.

Robert y Kate Kestnbaum fueron pioneros en realizar marketing con la información de clientes recolectada en bases de datos, conocido como 'database marketing', que era simplemente un slogan para definir la práctica de la creación de grupos de servicio al cliente para hablar individualmente a todos los clientes de una empresa. Esta información se utilizaba más tarde para ayudar a personalizar las relaciones con nuevos clientes potenciales, siendo muy útil a la vez para mantener un servicio abierto y adaptado a las necesidades de los clientes más importantes.

En 1986 se crean los primeros gestores de contactos, creados por ACT!, y en 1989 Jon Ferrara crea Goldmine, desarrollando el primer SFA (Sales Force Automation), que estaba orientado a las pymes. Estas aplicaciones incluían la Administración de Contactos (Contact Management), la Administración de Oportunidades de ventas (Opportunity Management) y los Pronósticos de Facturación (Revenue Forecasting). La proliferación de los ordenadores personales y la llegada de la arquitectura cliente/servidor prepararon el terreno para un crecimiento exponencial en el desarrollo de software.

Al principio de la década de los 90 Brock Control Systems ayudó a impulsar la evolución del software de gestión de contactos hacia los SFA. Los SFA tomaron muchas características del marketing basado en las bases de datos, las automatizaron y las combinaron con la gestión de contactos.

A lo largo de esta década las compañías comenzaron a mejorar lo que era el CRM dando a los clientes incentivos, regalos y otras ventajas por la lealtad a la empresa. Fue el comienzo de los actuales programas de viajeros, puntos en tarjetas de crédito y otros tipos de recursos propios de los CRM que están basados en el análisis de los patrones de actividad y compras de los consumidores.

En 1995 se le dio finalmente el nombre de CRM al software que aunaba los SFA y la gestión de contactos, después de haberse llamado ECM (Enterprise Customer Management) y CIS (Customer Information System). En la segunda mitad de esta década, los desarrolladores de los ERP (Enterprise Resource Planning) entraron en el mercado de los CRM esperando dominar la industria usando influencia, tamaño y poder adquiridos con los ERP. Hubo una gran competencia y esto empujó a los desarrolladores a introducir nuevas funcionalidades, así como generar más marketing, ofertas y aplicaciones para los CRM. Finalmente, apareció Salesforce, el primer gran desarrollador de SaaS (Software as a Service).

Durante el principio del nuevo siglo surgió lo que hoy se conoce como CRM, aunque la industria del CRM sufrió bastante a causa de la “burbuja puntocom”, al igual que la mayoría de las empresas desarrolladoras de software. Se denomina “burbuja puntocom” al período de crecimiento económico de las empresas vinculadas a internet, alrededor de los años 1997 y 2001 [1]. Los desarrolladores de los e-CRM fueron los más afectados debido a su desconfianza hacia las tecnologías “puntocom”.

El CRM se convirtió en una forma de actualizar constantemente el conocimiento sobre las necesidades y comportamientos de los clientes, consiguiendo información sobre sus motivaciones y reacciones. A lo largo de la mitad de esta primera década la interoperabilidad entre el software más antiguo se convirtió en algo importante. También entraron al mercado de los CRM Microsoft con Dynamics CRM y Oracle compró Siebel, además de un gran número de otras desarrolladoras.

Los desarrolladores de CRM pudieron superar las dificultades que surgieron para soportar la ingente cantidad de información gracias a Internet, ya que les dio la posibilidad de colocar las bases de datos en otras localizaciones diferentes a la empresa. En 2007 creó Force.com, que revolucionó la industria del CRM introduciendo el CRM basado en la nube y sometió las críticas de que las aplicaciones basadas en la nube no se podían personalizar.

A finales de la primera década de los 2000 la idea general del CRM consistía en un sistema que abarcara todas las relaciones empresariales, como ya sugirió al principio de siglo Paul Greenberg en su libro “CRM at the Speed of Light”. Durante esta etapa y hasta el día de hoy, los CRM basados en la nube y en SaaS continuaron añadiendo más características como el servicio al consumidor y el CRM social. Los tres sectores de negocio que se apoyan más en los CRM son los servicios financieros, corporaciones de alta tecnología y la industria de las telecomunicaciones. [2] [3]

## 1.3 Módulos del CRM

Como la gran mayoría de las soluciones software que se han desarrollado hasta ahora, los CRM constan de varios módulos que realizan diferentes tareas dentro del mismo y le dan la funcionalidad completa a éste. Existen algunos módulos que están incorporados en todos los CRM, los esenciales, y otros módulos (o funcionalidades) que pueden o no estar implementados, o bien no están incluidas en el paquete que se ha contratado (como ocurre con los CRM *on demand*). Otras veces, dos o más módulos pueden estar aunados en uno solo a la hora del diseño y la implementación del sistema debido a las similitudes entre ellos.

- **Servicios al cliente:** Es el módulo principal de cualquier CRM, es imprescindible. Este módulo se encarga de proporcionar información de clientes, resolver incidencias con las ofertas de productos o servicios de la empresa (llamado gestión de casos) y de las relaciones con los clientes a través de diversos medios de comunicación, como los centros de llamada o el correo electrónico. Se encarga de la fidelización de clientes mediante la utilización de varias estrategias para incrementar la satisfacción de los mismos, como ofrecer ayuda mediante soporte técnico.
- **Clientes:** Se podrían incluir en el anterior y trata las bases de datos donde está acumulada toda la información de los clientes, como su contacto, su categoría, estados de consultas, sus compras, etc.
- **Marketing:** En este módulo se incluyen las herramientas de segmentación de clientes para gestionar las campañas de marketing y planificar los presupuestos. Cuando la campaña se ha realizado permite analizar los datos que se han recopilado, los comentarios de los clientes sobre la campaña y su reacción. Con esta información permite saber a la empresa quienes son sus clientes: qué desean, cómo desean y cuándo desean realizar negocios con la empresa, haciendo posible la personalización de la experiencia para fidelizar a los clientes. La automatización de marketing incrementa el rendimiento de los gastos en mercadeo y permite valorar la eficacia y eficiencia de las campañas.
- **Ventas:** Es otro de los módulos más importantes y el principal impulsor de las iniciativas CRM. Ofrece una amplia gama de herramientas que ayudan a la administración de las actividades de ventas, como pueden ser: gestión de contactos, gestión de oportunidades, planificación y análisis de ventas, gestión de propuestas y citas de los clientes, captura de clientes potenciales, que es una de las características principales de un CRM, etc. Suelen llevar también la gestión de inventario y pedidos de la empresa.
- **Planificación:** Este módulo es fundamental para que la comunicación interna de la empresa funcione y sea efectiva. Aquí se incluyen la agenda, las tareas, las reuniones, así como toda la actividad que realicen los empleados para estar al tanto de todos los movimientos.
- **Finanzas:** Gestiona todos los datos de facturación, contabilidad, ingresos, presupuestos, pedidos de compra... se puede incluir dentro del módulo de compras o ventas, ya que es el encargado de tener al día las cuentas de la organización. [4] [5] [6]

## 1.4 Tipos de CRM

A lo largo de la historia del CRM han ido surgiendo diversas variaciones de éste, con funcionalidades que difieren de la idea esencial de un CRM, haciendo que incluso ahora no haya un consenso sobre cuál es la definición real de un software CRM. Todas las variaciones de software CRM que hay en el mercado se pueden dividir esencialmente en tres tipos, existiendo algunos que están compuestos por dos tipos o incluso los tres, dependiendo de las funcionalidades que posean esas soluciones software. Algunos proveen las funcionalidades por módulos y permiten tener un CRM de un solo tipo o varios, siendo las soluciones más flexibles, conocidas también como CRM *on demand*.

### 1.4.1 Operacional

Este tipo de CRM se encarga fundamentalmente de la gestión de las funciones de marketing, ventas y servicios al cliente, es decir, los procesos de negocios. Para ello se divide en dos partes:

- **Back Office:** Es la parte del CRM que no tiene interacción directa con el cliente, y se centra en las funciones de contabilidad y finanzas. Los procesos y procedimientos de esta parte del CRM son creados y gestionados por la empresa y los posibles cambios que puedan experimentar son debidos a la influencia indirecta que tiene el cliente sobre esta parte del CRM. La finalidad de estos cambios es para mejorar el servicio que se le ofrece al cliente y que éste llegue a estar a la altura de sus expectativas y necesidades.
- **Front Office:** Dentro de este apartado se pueden introducir todos los procesos, servicios o personal que tiene relación directa con el cliente, tales como los *call centers*, vendedores a comisión, establecimientos de venta al público, etc. Esta interacción debe servir para establecer cierta confianza con el cliente y conseguir una fidelización de éste. Esta parte es, además, la encargada de las campañas de marketing y de la gestión de las ventas.

Toda la información que se recauda de las campañas de marketing y de la relación con los clientes ayuda a conocer mejor sus necesidades, y así poder ofrecerles un producto o servicio adecuado. Para poder llevar a cabo la producción de estos nuevos productos o servicios habría que modificar procesos internos de la empresa (Back office), y una vez conseguidos, estos conllevarán nuevas campañas de marketing y captación y fidelización de nuevos clientes (Front office). Como puede observarse, ambas partes están fuertemente relacionadas. [7] [8]

### 1.4.2 Analítico

Se basa en el modelo de negocio *Business Intelligence*, que utiliza el almacenamiento, proceso, modelación, análisis y explotación de la información disponible sobre el cliente con el fin de mejorar el servicio al cliente. El almacenamiento de la información se realiza en un almacén de datos perteneciente a la empresa que se conoce como Data Warehouse. En él se guardan los datos de los clientes, número de ventas, costes de las campañas de marketing, etc. Esta información es analizada y explotada mediante el data mining para poder reconocer patrones tales como el comportamiento de compra de los clientes base, así como el impacto general que ha tenido una campaña de marketing y si ésta ha sido rentable o no. Esto hace que la información almacenada en el Data



Warehouse se transforme en conocimiento, con el que se puede mejorar el servicio a los clientes, enfocar las campañas de otras formas o segmentar el mercado, haciendo campañas diferentes sobre el mismo producto para diferentes tipos de clientes. [8] [9]

Un CRM analítico debe ser capaz de incorporar alarmas para el seguimiento y optimización de una campaña de marketing y de proponer alternativas, además de facilitar informes detallados sobre varias materias, pero principalmente sobre estas tres:

- **Impacto de la comunicación:** Debe reflejarse la información referente a la comunicación de la empresa con los clientes, como puede ser quiénes han recibido y abierto su e-mail, qué clientes de un segmento en particular han accedido a la zona personalizada de la web, cuántas veces y cuándo, qué cartas han sido devueltas por correo, etc.
- **Impacto de la promoción:** Debe contener la información sobre el impacto que ha tenido una campaña de marketing sobre el público objetivo, como por ejemplo los índices de respuesta y el cambio en el comportamiento del cliente, tanto si lo ha habido o no.
- **Índice de satisfacción:** Debe incluir información sobre las reclamaciones, dudas y sugerencias de los clientes hacia la empresa, redención de los vales descuento, satisfacción del servicio en los fines de semana o festivos, etc.

En definitiva, un CRM analítico debe poder analizar y explotar la información sobre los clientes que posee la empresa en sus bases de datos para transformarlos en conocimiento y utilizar éste para conocer mejor a sus clientes y poder así ofrecerles soluciones comerciales más adecuadas a sus necesidades. [7] [8]

### 1.4.3 Colaborativo

El CRM Colaborativo se centra en la interacción entre la empresa y el cliente a través de diferentes canales de comunicación, haciéndola más cómoda para ambas partes. Algunos de estos canales de comunicación son el e-mail, el teléfono o el chat, incluyendo los *call center*, en los que un IVR (Interactive Voice Response) atiende la llamada y, mediante una interfaz con voz, recoge información que le da el cliente. Con ella accede a la base de datos de la empresa para almacenarla o extraer otra información referente a este cliente. También se incluyen en estos canales de comunicación a los posibles proveedores y distribuidores de los servicios o productos de la empresa, que también pueden recoger el *feedback* de los clientes en sus establecimientos. [8] [9]

La principal ventaja que tiene este tipo de CRM es que todos estos tipos de canales que utiliza son accesibles desde casi cualquier dispositivo móvil, bien sea una tableta, un portátil o un smartphone. La empresa puede ofrecer a los clientes servicios o productos que se adecuen a sus necesidades aprovechando estos canales que proporcionan las nuevas tecnologías, estableciendo así un vínculo con los clientes y mejorando su relación con la empresa. [7]

Otra de las principales funcionalidades del CRM colaborativo es establecer esta comunicación multicanal entre todos los departamentos de la empresa. Así se produce una retroalimentación de información dentro de la propia empresa que ayudará a encontrar los posibles fallos en los diferentes procesos internos, analizarlos y modificarlos para evitar que se produzcan de nuevo y mejorar la productividad. [7]

## 2 Identificación del problema

---

### 2.1 Introducción

En este capítulo, se pretende obtener un entendimiento completo del dominio y del entorno del problema que se está considerando. Se deberán esclarecer todas las dificultades que presente el problema a tratar, alcanzando la mejor solución posible para el problema que se plantea.

### 2.2 Identificación del problema real

Se desea implementar la aplicación AlephCRM con la tecnología WPF (Windows Presentation Foundation). Esta aplicación ya está implementada por la empresa Tecnologías y Soluciones en la Nube, S.L.U. (de ahora en adelante Tecnube) con la tecnología WinForm. Ambas tecnologías han sido desarrolladas por Microsoft en la plataforma .NET y se utilizan para desarrollar implementar aplicaciones de escritorio para Windows.

La razón fundamental de este cambio es actualizar la aplicación a una tecnología más reciente como es WPF, ya que Microsoft tiene en modo mantenimiento a WinForm, corrigiendo únicamente bugs que se vayan encontrando, sin agregar nuevas características importantes [10]. Esto podría indicar que en un futuro no muy lejano WinForm deje de recibir soporte por parte de Microsoft, habiendo sido la tecnología que se ha estado usando para desarrollar las aplicaciones de escritorio en Windows desde las primeras versiones de .NET.

La mayor ventaja de WPF frente a WinForm es que separa completamente el componente gráfico del código, haciendo posible cambiar completamente la interfaz de la aplicación y que ésta haga exactamente lo mismo. Además, utiliza gráficos vectoriales, haciendo que los controles puedan ser escalados sin perder calidad o que se genere pixelización, aumentando así la accesibilidad.

En el mercado del desarrollo software existen diversas empresas y equipos que se dedican a fabricar diferentes componentes con funcionalidades completas para ser utilizados a la hora de desarrollar. Estos componentes son fáciles de personalizar siguiendo sencillos pasos, dejando a la aplicación con un acabado muy profesional. La justificación del uso de estos componentes es que sería muy costoso para cualquier empresa desarrollarlos desde cero, ya que requieren mucho tiempo de diseño y programación. En este caso se utilizarán los componentes de DevExpress [11], con los que la empresa lleva trabajando desde hace años.

En la implementación de esta nueva versión del software AlephCRM se desarrollarán algunos de los módulos del software original, que supondrán los básicos de un CRM consistente. El usuario final podrá realizar diferentes tareas de varios ámbitos:

- **Información:** Acceder a la información de los sujetos que conforman la aplicación, tanto si son clientes como si son otros usuarios de la aplicación (administradores, trabajadores, supervisores, etc). La información estará restringida según el rol que tenga el usuario dentro de la empresa.
- **Proyectos:** Programar proyectos y asignar tareas a los diferentes trabajadores, añadiéndolas a los calendarios de los trabajadores.

- **Asistencia:** Registrar las horas de trabajo realizadas durante la jornada mediante un sistema de fichajes de entrada y salida.
- **Autenticación:** Entrar en el sistema con unas credenciales propias y únicas que le identifiquen en la empresa.
- **Apariencia:** Cambiar la apariencia de la aplicación para obtener una experiencia más personalizada.

## 2.3 Identificación del problema técnico

Una vez detallada la definición del problema real con todos sus condicionantes, pasaremos a abordar el mismo de una forma técnica. Para llevar a cabo la identificación del problema técnico se utilizará una técnica de ingeniería denominada PDS (Product Design Specification) [Pressman, 1994]. Esta técnica permite realizar un análisis de los principales condicionantes técnicos del problema, contempla las siguientes cuestiones:

### 2.3.1 Funcionamiento

Dependiendo del rol que tenga asignado respecto a su puesto en la empresa, habrá algunas funcionalidades que estén capadas para el usuario en particular. Ahora se determinarán las funciones que podrá realizar el usuario con el rol más alto (administrador), pues este rol es el que no tendrá ninguna capada. Es decir, todas las funcionalidades a las que pueden acceder todos los usuarios son accesibles para el administrador, mientras que no todos los usuarios pueden acceder a todas las funcionalidades del administrador. Así que, una vez abierta la aplicación, el usuario podrá realizar las siguientes operaciones:

- **Acceso:** Autenticarse en el sistema con sus credenciales de usuario personalizadas de la empresa usando el correo corporativo. Esto le permitirá tener toda la información del trabajo accesible en todo momento, sincronizando el calendario y recibiendo los correos electrónicos en el mismo sitio.
- **Consulta de clientes:** Podrá visualizar las tablas con información referente a los clientes, tanto su contacto como el producto o servicio que tienen contratado. En estas tablas podrá filtrar por cualquiera de las columnas, además de tener un cajón de búsqueda. Haciendo doble click se accederá a la ficha del cliente, donde podrá modificar la información de éste.
- **Control de clientes:** Se podrán registrar las llamadas y correos electrónicos que se intercambien con los clientes, introduciendo información referente a ellas, como el día y la hora en la que se hicieron y el tema que se trató en estas conversaciones.
- **Consulta de trabajadores:** El sistema le mostrará una tabla con la información de los trabajadores de la empresa. Se podrá filtrar por cualquiera de sus valores además de poder buscar directamente un valor en concreto. Cuando haga doble click en uno de los trabajadores, podrá acceder a su ficha personal y modificar los datos de éste. También se podrá acceder a los registros de fichajes del control de presencia, para comprobar si cumplen con el número de horas establecidos.
- **Gestión de proyectos:** Crear proyectos, dividiéndolos en tareas y pudiendo asignarlos a diferentes trabajadores. Estas tareas y proyectos tendrán fechas límite que estarán integradas en el calendario del trabajador.

- **Control de presencia:** El usuario podrá registrar cuándo comienza y termina de trabajar mediante una interfaz simple. Este registro almacenará información referente al usuario y al equipo que está utilizando para trabajar.
- **Generación de informes:** El sistema permitirá generar informes a través de las diferentes tablas, pudiendo mostrar análisis de esta información.
- **Cambio de tema:** Personalizar la apariencia de la aplicación de forma que sea más acorde a los gustos del usuario. Es una funcionalidad meramente estética pero que los usuarios siempre agradecen.

## 2.3.2 Entorno

Para la consideración del entorno en que se encuentra nuestro sistema, consideraremos la división de éste en tres áreas que abarcan los aspectos principales a tener en cuenta: el entorno **hardware**, el entorno **software** y el entorno de **usuario**.

### 2.3.2.1 Hardware

La elección del hardware en el que se ejecutará la aplicación no presenta demasiada complejidad, puesto que no necesita demasiados recursos para lograr unos resultados óptimos. Deberá tener una tarjeta de red para la conexión con la base de datos y poder ejecutar el Framework 4.5, cuyos requisitos mínimos son:

- Procesador a 1 GHz o más rápido de 32 bits (x86) o de 64 bits (x64)
- 512 MB de RAM
- 4.5 GB de espacio disponible en el disco duro (32 bits y 64 bits)
- Tarjeta gráfica DirectX9 con controlador WDDM 1.0 o superior [12]

### 2.3.2.2 Software

El software necesario para poder utilizar la aplicación será un sistema operativo igual o superior a Windows Vista (7, 8, 8.1 y 10) con el Framework .NET 4.5 o superior instalado.

La gestión y almacenamiento de la base de datos se hará utilizando Microsoft Azure, a través del Portal de Azure. Éste permite restringir el acceso a ésta y configurarla de una forma bastante sencilla. Además, permitirá utilizar la autenticación mediante el Azure Active Directory, pudiendo utilizar cuentas corporativas de la empresa cliente. Los usuarios finales no requerirán de ningún tipo de conocimiento sobre Azure para la utilización de la aplicación, ya que será gestionado por Tecnube.

### 2.3.2.3 Usuario

Es una aplicación de gestión CRM bastante sencilla, con un funcionamiento y diseño intuitivos. El usuario no necesitará más que los conocimientos elementales de informática. La interfaz será muy similar a los productos de ofimática a los que estará acostumbrado, por lo que no debería suponerle ninguna dificultad el manejo óptimo de ésta.

### 2.3.3 Vida esperada

La accesibilidad y usabilidad hacen que la herramienta sea bastante útil debido a su fácil manejo, además de poseer un esquema de datos robusto que proporciona fiabilidad frente a la gran competencia en el mercado. Debido a estas razones se piensa que la vida esperada será larga, aunque servirá como punto de partida para la inclusión de nuevas funcionalidades que le permitirán posicionarse mejor en el mercado y aumentar más aún su esperanza de vida. Además, más adelante, se podrá implementar para dispositivos móviles de una manera sencilla, por lo que aumentará aún más su accesibilidad.

### 2.3.4 Ciclo de mantenimiento

El mantenimiento de esta aplicación se basa en la adaptación a los estándares y nuevas funcionalidades que puedan ir surgiendo con la actualización de la tecnología WPF por parte de Microsoft y de la implementación de nuevos controles o modificaciones en los ya existentes de DevExpress.

Además, se tendrá que tener en cuenta cualquier posible fallo en la conexión con la base de datos y el poder ofrecer servicio a todos los usuarios en todo momento, siendo estos los trabajadores de la empresa cliente. Hay que ofrecer soporte técnico en este ámbito y saber gestionar estos problemas, identificando su raíz, bien sea por parte de la plataforma de Azure o bien sea por un error de diseño o programación en el desarrollo de la aplicación.

También hay que ofrecer un soporte técnico para para los posibles errores o *bugs* que puedan aparecer durante la utilización de la aplicación que no se hayan encontrado en las pruebas realizadas en la fase final del desarrollo, ya que esto siempre puede ocurrir y hay que ser de ofrecer una solución rápida y eficaz. Esta solución puede ser momentánea o en forma de actualización de la aplicación corrigiendo dicho fallo a través de código.

En un futuro se irán introduciendo nuevas funcionalidades que son útiles para la gestión empresarial y se investigarán posibles mejoras en la navegación y uso de la aplicación, así como su rendimiento.

### 2.3.5 Competencia

Existen numerosas aplicaciones dedicadas a la gestión CRM, todas con sus propias características singulares que las hacen diferenciarse en algo del resto, pero siempre con unas funcionalidades básicas que todas comparten. Los principales competidores que se pueden encontrar en el mercado son Salesforce, con un gran la cuota más alta en el mercado, Microsoft Dynamics (véase Antecedentes) o SugarCRM como alternativa con código abierto. Estas soluciones ofrecen sus servicios bajo demanda (Salesforce), como productos completos e independientes (Microsoft Dynamics) o como paquetes de funcionalidades y características (SugarCRM). En el caso de AlpehCRM se ofrecerá un único paquete de funcionalidades, pudiendo en un futuro cambiar el modelo de negocio a un CRM bajo demanda, haciendo los módulos independientemente operativos.

Bien es cierto que alcanzar las cuotas de mercado que manejan estos grandes en la industria del CRM es bastante difícil y el acabado final es más consistente, hay algún público objetivo al que se puede enfocar este software. Este público serían las pequeñas empresas que buscan una solución más

simple y barata, pero que les ofrezca funcionalidades similares a las grandes marcas y con un acabado profesional, además de un trato más personal y personalizado. En este aspecto se podría incluso superar a la competencia, pues este tipo de empresas no suelen tener demasiado presupuesto dedicado a la gestión de la relación con los clientes, aun siendo una parte esencial del negocio empresarial.

### 2.3.6 Aspecto externo

La interfaz del sistema será similar a las aplicaciones del paquete de Microsoft Office para ofimática. Tendrá un sistema de pestañas que separará las categorías de las funciones de los diferentes módulos, así como un menú común a toda la aplicación. Todo será siguiendo las normativas y pautas de usabilidad existentes, de forma que sea lo más intuitiva posible. De esta forma se consigue un entorno familiar para el usuario, haciendo que no encuentre dificultad alguna durante el manejo de la aplicación.

Con respecto a la generación de informes se basará en el concepto WYSISYG (What You See Is What You Get, Lo Que Ves Es Lo Que Obtienes en inglés). Esto quiere decir que el informe que se genere a partir de la tabla que esté seleccionada en la aplicación contendrá exactamente la tabla en su estado actual, con los filtros y búsquedas realizadas (si las hubiera). Aunque también se dará la posibilidad de personalizarlos, los que generará la aplicación de forma automática serán como se ha descrito.

### 2.3.7 Estandarización

La aplicación se regirá por los estándares de diseño de Microsoft para las aplicaciones en los sistemas Windows de escritorio. Así mismo, se seguirán los estándares de desarrollo para esta plataforma, utilizando para ello el entorno de desarrollo Visual Studio 2015. Este entorno de desarrollo está preparado para utilizar las tecnologías desarrolladas por Microsoft, como son WinForms, WPF y UWP (Universal Windows Platform). Estas tecnologías pueden usar en su código base C# o Visual Basics. En este caso se utilizará C# para describir las funcionalidades, y XAML para describir la interfaz gráfica de la aplicación, característica exclusiva de WPF y UWP.

### 2.3.8 Calidad y fiabilidad

Debido a la importancia que tienen factores como la calidad y la fiabilidad en la construcción de software, se realizará un plan de pruebas unitarias y de integración que verifique y valide la funcionalidad y el rendimiento de la aplicación. Se pondrán a prueba tanto las funcionalidades como la integridad y consistencia de la base de datos.

### 2.3.9 Programa de tareas

La realización de este proyecto se ha programado en las siguientes fases. Estas se realizarán en orden cronológico, aunque la secuencialidad de todas ellas no es totalmente estricta.

### 2.3.9.1 Fase 1 - Fase de preparación

- Estudio de la tecnología WPF y su entorno de desarrollo, Visual Studio Enterprise 2015.
- Estudio del lenguaje C#, XAML y del Framework .NET 4.5.
- Estudio de los controles de DevExpress y su funcionamiento.
- Estudio y asimilación de la metodología de trabajo de la empresa Tecnube, usando Azure, Sharepoint y Exchange.
- Estudio y manejo de las bases de datos desarrolladas en SQL Server Azure de Microsoft.
- Estudio de la integración de los diferentes componentes en una aplicación profesional y de ámbito empresarial.

### 2.3.9.2 Fase 2 - Análisis y especificación del sistema.

- **Descripción detallada del problema.** En esta descripción se indicarán los objetivos, requisitos y restricciones que están presentes en el sistema, de una forma ordenada y que permita su posterior validación. Descripción detallada del dominio de la información en la que se presentarán sin ambigüedades los objetos que forman parte del dominio del problema y las relaciones entre ellos.
- **Descripción del dominio de la función** en la que se presentará la funcionalidad del sistema que satisface los objetivos, requisitos y restricciones. Validación del análisis hecho en la que se comprobará la funcionalidad y estructura de la información con respecto a la descripción del problema.

### 2.3.9.3 Fase 3 - Desarrollo del software

Desarrollo de la versión inicial de la interfaz de usuario. Implementación de la lógica del programa de forma que se alcance toda la funcionalidad requerida. Finalización de la aplicación desarrollada, con la interfaz de usuario final.

### 2.3.9.4 Fase 4 - Pruebas del software

Realización del plan de pruebas unitarias. Realización del plan de pruebas de integración del sistema. Realización de las pruebas contenidas en los planes de pruebas.

Estudio de la respuesta del sistema a las pruebas realizadas y detección de posibles incidencias. Corrección de incidencias encontradas.

### 2.3.9.5 Fase 5 – Documentación

Todas las etapas indicadas anteriormente, se documentarán debidamente, de forma que, a la conclusión del proyecto, se disponga de una memoria del mismo que incluirá un Manual Técnico, un Manual de Usuario y un Manual de Código. La memoria y la documentación del código se realizarán a la vez que las demás fases de desarrollo, dejando el desarrollo del manual de usuario una vez finalizado el software.

## 2.3.10 Pruebas

El software será sometido a diversas pruebas que se pueden clasificar en dos categorías:

- **Pruebas de Unidad o de Caja blanca:** Este tipo de pruebas se realizará durante la codificación de la aplicación, en la que irán surgiendo diversos errores y se corregirán mediante la aplicación de este tipo de pruebas.
- **Pruebas Funcional o de Caja negra:** Este tipo de pruebas se centran en el estudio de la especificación del software, análisis de las funciones que debe realizar, de las entradas y de las salidas.

## 2.3.11 Seguridad

La seguridad de la aplicación se plantea desde el punto de vista de la protección de datos de la aplicación y de los usuarios, es decir, garantizar que los recursos de la aplicación se utilizan de manera correcta y la información de los usuarios sólo es accesible a las personas que tienen permiso. Para ello se utilizan las siguientes medidas de seguridad:

- **Autenticación con credenciales empresariales:** Cada usuario deberá introducir sus credenciales únicas de la empresa para poder hacer uso de la aplicación. Esto se conseguirá utilizando una autenticación mediante la API de Azure Active Directory de Microsoft. Esta API hace que las credenciales que se utilicen se almacenen en el Administrador de credenciales de Windows, y pudiendo eliminarlas o modificarlas a través de las opciones de Internet Explorer, por lo que no hace falta encriptarlas en ningún momento, ya que el propio Windows se encarga de ello.
- **Restricción de datos:** No todos los usuarios podrán acceder a todas las funcionalidades de la aplicación y tampoco podrán acceder a todos sus datos. Dependiendo del rol que se les haya asignado en la base de datos (dependerá de su puesto de trabajo), podrán consultar unos datos u otros. Así bien, los administradores podrán consultar los datos y fichajes de entrada y salida de todos los trabajadores, mientras que si es un mero trabajador sólo podrá acceder a sus propios datos. Los datos de los clientes, sin embargo, estarán disponibles para todos los roles, puesto que todos necesitarán esta información para trabajar.
- **Encriptación de credenciales:** Se encriptarán las credenciales con las que se accede a la base de datos y al portal de Azure para la autenticación de los usuarios, de forma que nadie pueda encontrarlas haciendo un análisis de código o de ajustes de la aplicación y usarlas para acceder de forma fraudulenta.



## 3 Objetivos

---

De acuerdo a la identificación real y técnica del problema, que se ha realizado en el capítulo anterior, en este capítulo se expondrán todos los objetivos que se pretenden alcanzar para la consecución del proyecto. El objetivo principal del proyecto es la creación de una solución software CRM para la empresa desarrollado en la tecnología WPF de Microsoft, siendo esta solución software una actualización de tecnología de la aplicación AlephCRM, ya desarrollada por Tecnube anteriormente con la tecnología Windows Forms de Microsoft. Sin embargo, esta aplicación no contendrá todos los módulos y funcionalidades de su predecesora en esta primera etapa de desarrollo. Para lograr este objetivo final, se definen a continuación una serie de objetivos:

- **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.

## 4 Antecedentes

---

### 4.1 AlephCRM en WinForm

Esta es la aplicación que ha sido desarrollada por la empresa anteriormente, que consta de diversos módulos para la gestión de la empresa. Tiene un sistema de autenticación y conexión a la base de datos previa al inicio del software de gestión en sí. En esta primera ventana de autenticación se puede seleccionar la base de datos a la que se va a conectar (proporcionada por Tecnube) y el nombre y contraseña del usuario que accede. De esta forma se pueden alojar servicios para varias empresas con un único software para cada una de ellas.

Una vez en la aplicación se puede observar una interfaz con la línea de diseño del paquete Office de Microsoft. En la aplicación se puede acceder a la información que se tiene almacenada sobre los trabajadores, proveedores, acreedores, usuarios web, contactos, productos, etc.

Como CRM permite acceder a los teléfonos y correos de los clientes, así como registrar el contacto que se tiene con las empresas o comerciales, bien sean llamadas, correos o reuniones. Se puede mantener un seguimiento de estas comunicaciones clasificándolas como activas, pendientes o terminadas.

Como gestor de recursos permite generar tareas y asignarlas a los trabajadores. Cada trabajador puede consultar las tareas que tiene asignadas y las fichas de estas tareas. También se pueden registrar las horas de trabajo de los trabajadores y realizar análisis. Cuenta con una funcionalidad que permite introducir las tareas, proyectos u horas asignadas en el calendario personal del trabajador, similar al calendario de Outlook o Gmail.

Como gestor de proyectos se pueden consultar los proyectos actuales, así como crearlos o eliminarlos, dividirlos en tareas más básicas y asignar éstas a los trabajadores. Permite generar expedientes para el seguimiento de los proyectos, que se subdividen en referencias, las cuales muestran el estado de un proyecto para un determinado cliente.

Respecto a la facturación, el sistema puede generar documentos de factura o abono de venta. Permite registrar y gestionar los recibos, los cobros y otros aspectos administrativos y económicos. Se puede tener un balance de saldos y un control sobre los contratos que se tienen o se han tenido con los clientes, pudiendo clasificarlos como activos o bloqueados.

Como gestor de almacén, simplemente se tiene un listado de los artículos que se tienen. Se pueden clasificar por tipos o por subtipos (comercializados o propios).

En todas estas funcionalidades siempre se encuentran tablas con numerosos datos en ellas. La aplicación tiene la funcionalidad de poder buscar y agrupar por cualquier valor de las columnas y filas de estas tablas. Estas búsquedas y clasificaciones se podrán exportar a Excel, RTF o PDF, así como a un archivo de configuración que después se podrá cargar en el sistema para que se muestre de nuevo esa clasificación que se generó. También se pueden mostrar las diferentes ventanas que se han ido abriendo como una agrupación de pestañas o ventanas flotantes, así como presentarlas dividiendo la pantalla en dos para una mejor visualización para una comparación.

Por último, brinda la posibilidad al usuario de utilizar un editor de hojas de cálculo, un editor de textos, un visor de PDF, un visor de XML y un editor de informes, en el que se pueden diseñar diferentes tipos de informes con plantillas predeterminadas y crear nuevas plantillas. Estos informes se pueden exportar a diversos tipos de archivos, como pueden ser PDF, Excel, Word o XML, entre otros

## 4.2 Microsoft Dynamics CRM

Este software, que es ahora uno de los más populares y solicitados en el mercado, es un producto que ha ido evolucionando a lo largo de más de una década, desarrollando nuevas funcionalidades y refinando las ya incluidas. [12]

En un principio fue llamado Microsoft Business Solutions Customer Relationship Management 1.0, cambiado a CRM 1.0 por razones obvias. Fue lanzado en enero de 2003, y no fue hasta 2005, con la salida de la versión 3.0, cuando se produjo el cambio de nombre a Dynamics CRM. En esta versión se añadieron funcionalidades para la gestión del marketing.

En la versión 4.0, en 2007, se añadió la compatibilidad con SQL 2008 (tecnología de Microsoft) y la posibilidad de que los clientes se pudieran conectar al software a través de un solo server, usando una arquitectura multitenant, entre otras características interesantes. En la versión lanzada en 2010 se introdujeron los paneles de control, que permiten al usuario tener una vista resumida de la información que tienen almacenada, y el concepto de soluciones y capas de soluciones, que permitía a los partners desarrollar paquetes personalizados para los clientes como si se desarrollara una app.

En 2013 se rediseñó por completo la apariencia de la aplicación, permitiendo además acceder a ella desde cualquier navegador web o PC que pudiera utilizar Outlook, además de ser compatible con dispositivos iOS, Android y Windows Phone 8. La última versión hasta el momento, lanzada en 2014, tiene incluido el Excel y OneNote incrustado en él. De esta última versión destacaremos algunas características señaladas a continuación. [12]

Microsoft Dynamics CRM se vende como un software con varias soluciones de interacción con los clientes, integradas unas con otras ofreciendo una experiencia completa del CRM. Estas soluciones son:

- **CRM de ventas:** Adapta a la empresa a las nuevas tecnologías y a los nuevos clientes que están conectados a las redes sociales y realizan sus propias investigaciones. Tiene las siguientes funcionalidades:
  - **Gestión de oportunidades**
  - **Venta en medios sociales**
  - **Planificación y gestión**
  - **Colaboración en contenido**
  - **Ventas móviles**
  - **Inteligencia de ventas**

- **CRM de Servicios:** Ofrecen una solución de servicio de atención al cliente que aumenta el seguimiento y la fidelidad del mismo, usando los canales de autoservicio, asistidos y de servicio postventa. Sus funciones son:
  - **Omnicanal.**
  - **Portal**
  - **Capacitación de agentes**
  - **Servicio in situ**
  - **Conocimiento**
  - **Inteligencia de servicios**
  
- **CRM de Marketing:** Ofrece herramientas para la captación de clientes, permitiendo planificar, ejecutar y medir campañas en distintos canales. Las funciones que ofrece son:
  - **Gestión de los recursos de marketing**
  - **Gestión de clientes potenciales**
  - **Colaboración de ventas**
  - **Marketing social**
  - **Inteligencia de marketing**
  
- **CRM Social:** Proporciona herramientas que permiten escuchar, analizar y mejorar las interacciones con los clientes en las redes sociales. Ofrece las siguientes funciones:
  - **Acceso a opiniones de usuario**
  - **Análisis sociales**
  - **Interacción social**
  - **CRM de redes sociales**
  
- **CRM para Pymes:** Se trata de la versión Online de la aplicación. Es una solución móvil basada en la tecnología en la nube de Microsoft para ayudar a las pequeñas y medianas empresas a administrar contactos, clientes potenciales e impulsar las ventas. Permite una integración completa con el paquete de Office 365. [13]

### 4.3 Justificación del proyecto

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.

## 5 Restricciones

---

### 5.1 Introducción

En este capítulo se expondrán todas las restricciones, o factores limitativos, existentes en el ámbito del diseño y que condicionan la elección de una u otra alternativa. Estos factores limitativos, según su tipo, se pueden clasificar en dos grupos: factores dato y factores estratégicos.

### 5.2 Factores dato

Los factores dato son aquellos que no pueden ser modificados durante el transcurso del proyecto, se consideran los siguientes:

- La aplicación debe tener una interfaz y un funcionamiento sencillos, de manera que se facilite al máximo la navegación y el manejo del programa a los usuarios finales.
- Restricción de acceso. Se deberá controlar quién accede a la aplicación y en función de eso mostrar unas funcionalidades u otras. Así mismo se deberá controlar el acceso a la información en la base de datos, realizado con los permisos a los diferentes roles.
- Deberá permitir al menos el control de la información de los clientes (módulo CRM) y el control de presencia de los trabajadores.
- Se debe permitir una autenticación con credenciales únicas y cerrar sesión al usuario en la misma aplicación.
- El gestor de base será Microsoft SQL Server, usando el lenguaje T-SQL (Transact SQL).

### 5.3 Factores estratégicos

Los factores estratégicos son factores modificables por el ingeniero que dan lugar a diferentes alternativas en función de las decisiones que se hayan tomado. En este proyecto se han considerado los siguientes factores estratégicos:

- Se intentará incluir los 6 módulos anteriormente descritos, con todas sus funcionalidades. Deberán estar integrados correctamente y ser lo más independientes posibles entre sí, de forma que puedan ser reutilizados en un futuro.
- Se usarán como lenguaje de desarrollo XAML para la interfaz y C# para el código funcional, ya que son los lenguajes que se utilizan para desarrollar en WPF.

- El usuario podrá exportar las tablas y generar informes en diversos formatos de archivos, como son .docx (Word), .xlsx (Excel), .xml (XML) y .pdf (PDF).
- Se podrán modificar los datos almacenados de los sujetos (trabajadores y clientes), así como añadir y eliminarlos desde la propia aplicación.

## 6 Recursos

---

### 6.1 Introducción

En este capítulo se expondrán claramente los recursos, tanto humanos como materiales, necesarios para este proyecto. El análisis de los recursos necesarios se abordará desde dos puntos de vista:

- Recursos necesarios para el desarrollo.
- Recursos necesarios para la explotación.

### 6.2 Recursos necesarios para el desarrollo

En este apartado se describen los recursos que serán necesarios para la realización de este proyecto.

#### 6.2.1 Recursos humanos

Autor del proyecto:

- D. Luis Flores Méndez. Alumno del Grado en Ingeniería Informática de la Universidad de Córdoba.

Directores del proyecto:

- D. Francisco Méndez Serrano. Director técnico en sistemas y desarrollo de la empresa Tecnologías y Soluciones en la Nube, S.L.U.
- D. Enrique García Salcines. Profesor Asociado del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba.

#### 6.2.2 Recursos hardware

Ordenador personal PC:

- Procesador: Intel Core i7 4770K
- Gráfica: Gygabyte NVidia GeForce GTX 750 Ti 2GB GDDR5
- RAM: 8GB DDR3
- Almacenamiento: 2TB HD
- Placa Base: Gygabyte H97-D3H
- Sistema Operativo: Windows 10 Education

Tablet-PC Microsoft Surface Pro 3:

- Procesador: Intel Core i5 4300U
- Gráfica: Intel HD 4400
- RAM: 4 GB DDR3
- Almacenamiento: 128 GB SSD
- Sistema Operativo: Windows 10 Pro

Máquina Remota (recursos otorgados):

- RAM: 8 GB DDR3
- Procesador: 1 socket con 2 núcleos
- Almacenamiento: 120 GB disco duro
- Sistema Operativo: Windows 10 Education

### 6.2.3 Recursos software

- Sistema Operativo: Windows 10
- Entorno de desarrollo: Visual Studio 2015
- Módulos (Controls) de DevExpress para Visual Studio (versión 16.1.4)
- Lenguajes de programación: C# y XAML
- Paquete Microsoft Office 365 Empresa (Todas las versiones de 2016)
- Microsoft Visio 2016
- Microsoft Visual Studio 2015
- Herramienta de prototipado Pencil Project

## 6.3 Recursos necesarios para la explotación

En este apartado se describen los recursos que serán necesarios para el correcto funcionamiento del sistema software una vez haya concluido el desarrollo del mismo.

### 6.3.1 Recursos hardware

Serían los descritos anteriormente en el apartado de la Identificación del problema técnico, pues necesita al menos el Framework 4.5 para funcionar:

- Procesador a 1 GHz o más rápido de 32 bits (x86) o de 64 bits (x64)
- 512 MB de RAM
- 4.5 GB de espacio disponible en el disco duro (32 bits y 64 bits)
- Tarjeta gráfica DirectX9 con controlador WDDM 1.0 o superior [12]

### 6.3.2 Recursos software

- Sistema Operativo Windows Vista o superior con el Framework 4.5 instalado.





# SECCIÓN II

# ANÁLISIS



## 7 Especificación de requisitos

---

### 7.1 Introducción

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.

En la ERS se van a definir los datos y toda aquella información que el software manejará y transformará. También se definirá de forma gráfica y literal los procesos encargados de la transformación de la información y cómo esta información será mostrada al usuario.

La especificación de requisitos contempla los siguientes aspectos:

- **Descripción general del problema:** Definición del problema a tratar, especificando cuales son los objetivos generales del mismo de acuerdo a la información obtenida en el análisis del sistema.
- **Descripción funcional:** Se describirá cuáles serán las funciones necesarias a implementar para lograr un manejo de los datos acorde con los requisitos del software.
- **Descripción de la información:** Descripción detallada del problema, especificando cuales son los datos y la información a tratar, los flujos de datos y la interfaz de usuario.

### 7.2 Descripción general del problema

El proyecto que se pretende abordar es el diseño y desarrollo de una aplicación CRM que permita el control y consulta de la información de los clientes de una empresa, así como la de los trabajadores de la empresa. Como se expuso en el capítulo 3, se definieron los siguientes objetivos para este proyecto:

- La creación de un módulo principal de la aplicación, que gestionará las cuestiones generales de ésta. Se encargará del acceso a la información de los sujetos de la base de datos, así como agregar nuevos o eliminar los ya existentes. Se encargará también de controlar el rol del sujeto trabajador que está conectado a la aplicación en ese momento. Los formularios que utilizará para introducir nuevos sujetos serán los mismos que se usarán para modificar la información de estos en sus respectivos módulos.
- Creación de un módulo CRM. El usuario podrá consultar la información de los clientes, modificarla y añadir entradas con información sobre conversaciones con estos que se hayan producido en reuniones, por e-mail o por teléfono.
- Creación de un módulo de Recursos. El usuario podrá registrar cuándo comienza y cuando termina de trabajar, llevando así un control de las horas de trabajo realizadas. Aquí también

podrá acceder al historial de fichajes propio y, si su rol lo permite, ver el historial del resto de trabajadores para comprobar si existen anomalías.

- Creación de un módulo de Proyectos. Este módulo permitirá al usuario crear y gestionar proyectos, dividiéndolos en tareas y asignando estas a los trabajadores. Se podrán introducir en los calendarios de éstos, poniendo fechas límite y alertas.
- Creación de un módulo de Autenticación. Controlará el acceso a la aplicación, proporcionando así seguridad a la información que ésta maneja. Una vez que el usuario se ha autenticado correctamente con sus credenciales empresariales, se lanzará la aplicación completa.
- Dar la posibilidad al usuario de modificar la apariencia de la aplicación, cambiando sus colores, y la forma en la que se muestran las tablas de información.
- Desarrollar una interfaz gráfica sencilla, intuitiva y fácil de utilizar para todo tipo de usuarios, incluidos aquellos sin conocimientos informáticos. Esta interfaz será similar a la que se puede encontrar en las últimas versiones del paquete Office de Microsoft, haciendo así un entorno más familiar para el usuario.

## 7.3 Descripción funcional

En este apartado se procederá a realizar una descripción funcional del sistema que se va a desarrollar. Para llevar a cabo esta tarea se utilizarán los casos de uso que proporciona UML.

Un caso de uso es la descripción de un conjunto de secuencias de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra y en la que los actores obtienen resultados observables.

Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Con los casos de uso denotaremos los comportamientos esenciales del sistema.

### 7.3.1 Identificación de los casos de uso

Para realizar el análisis del sistema, utilizaremos casos de uso mediante dos tipos distintos de representaciones. Se realizará una representación gráfica de los mismos mediante diagramas de casos de uso y se utilizará una representación tabular para explicar cada uno de los casos de uso.

El diagrama de casos de uso representa la forma en que un usuario (denominado actor dentro de la nomenclatura de UML) opera con el sistema, además de la forma, tipo y orden en que los elementos interactúan (operaciones o casos de uso).

Mediante representaciones tabulares se explicarán algunos aspectos de los casos de uso que nos ayudarán a comprender en qué consisten, cómo se comportará el sistema, etc.

Aunque UML no propone ningún formato específico para representar los casos de uso de forma tabular, utilizaremos el formato que se indica a continuación para lograr este objetivo.

	Nombre del Caso de Uso
Objetivo	Explicación del beneficio obtenido por el actor con el caso
Acción	Evento que produce la ejecución del Caso de Uso. Un caso de uso siempre es iniciado por una acción de un actor, explicar qué acción del actor principal provocará la ejecución del caso.
Tipo	Se distinguirán dos tipos de casos de uso: - <b>Secundarios:</b> son casos de uso generales que se refinarán posteriormente en casos de uso primarios. - <b>Primarios:</b> son casos de uso finales. Su funcionamiento se detallará durante el diseño procedimental (capítulo 12).
Precondiciones	Condiciones que deben cumplirse para que el Caso de Uso inicie su ejecución.
Descripción	Descripción informal del caso de uso.

**Tabla 7.1:** Plantilla para definir los Casos de Uso

Antes de mostrar los principales casos de uso que se presentarán en el sistema software, se van a identificar a los actores que van a interactuar con la aplicación CRM.

### USUARIO

Representa a cualquier usuario que acceda a la aplicación CRM.



**Figura 7.1:** Identificación del actor USUARIO

### ADMINISTRADOR

Representa a los usuarios que utilizan la aplicación como cualquier trabajador, pero que además pueden controlar la presencia de todos ellos y añadir y eliminar sujetos en la base de datos, sean clientes o trabajadores.



**Figura 7.2:** Identificación del actor ADMINISTRADOR

## TRABAJADOR

Representa a los usuarios que utilizan la aplicación como un trabajador de la empresa. Ficha su entrada y salida del trabajo, puede consultar su propio historial y la información disponible de los clientes, así como registrar la interacción con ellos e indicar si una tarea asignada está terminada.



Figura 7.3: Identificación del actor TRABAJADOR

Entre los actores identificados se da una relación de generalización/especialización. Los actores Trabajador y Administrador son generalizados en el actor Usuario.

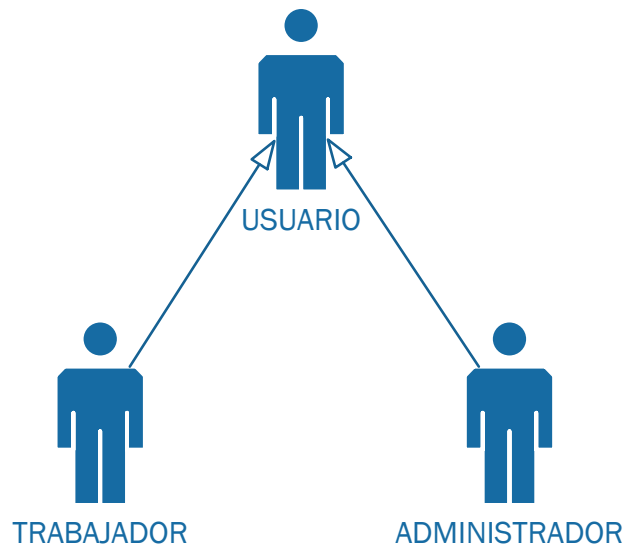


Figura 7.4: Relación entre los actores del sistema

A continuación, se muestran los principales casos de uso que se presentarán en el sistema software que se va a desarrollar. Se van a distinguir entre los casos de uso de trazo grueso, que tratarán las funcionalidades del sistema a un nivel más general, y los casos de uso de trazo fino, en los que se especificarán cada uno de los casos de uso de trazo grueso para una mejor comprensión de su funcionalidad total.

### 7.3.1.1 Casos de uso trazo grueso

Aquí se tratarán las funcionalidades más generales del sistema, en las que el actor será del tipo Usuario, pues ambos tipos de usuarios pueden realizar estas acciones.

Se han dividido en dos los casos de uso de trazo grueso, pues el usuario debe acceder al sistema antes de poder utilizar el resto de la aplicación siempre.

#### CU 0.a: Acceder al Sistema



Figura 7.5: CU 0.a - Acceder al sistema

CU 0.a - Acceder al Sistema	
Objetivo	Acceder a las funciones de la aplicación
Acción	El usuario abre la aplicación
Tipo	Primario
Precondiciones	La aplicación debe estar instalada en el dispositivo
Descripción	El usuario introduce sus credenciales de empresa y, si es correcto, se lanza la aplicación. En caso contrario, se muestra un error y la aplicación pide de nuevo introducir las credenciales.

Tabla 7.2: CU 0.a - Acceder al sistema

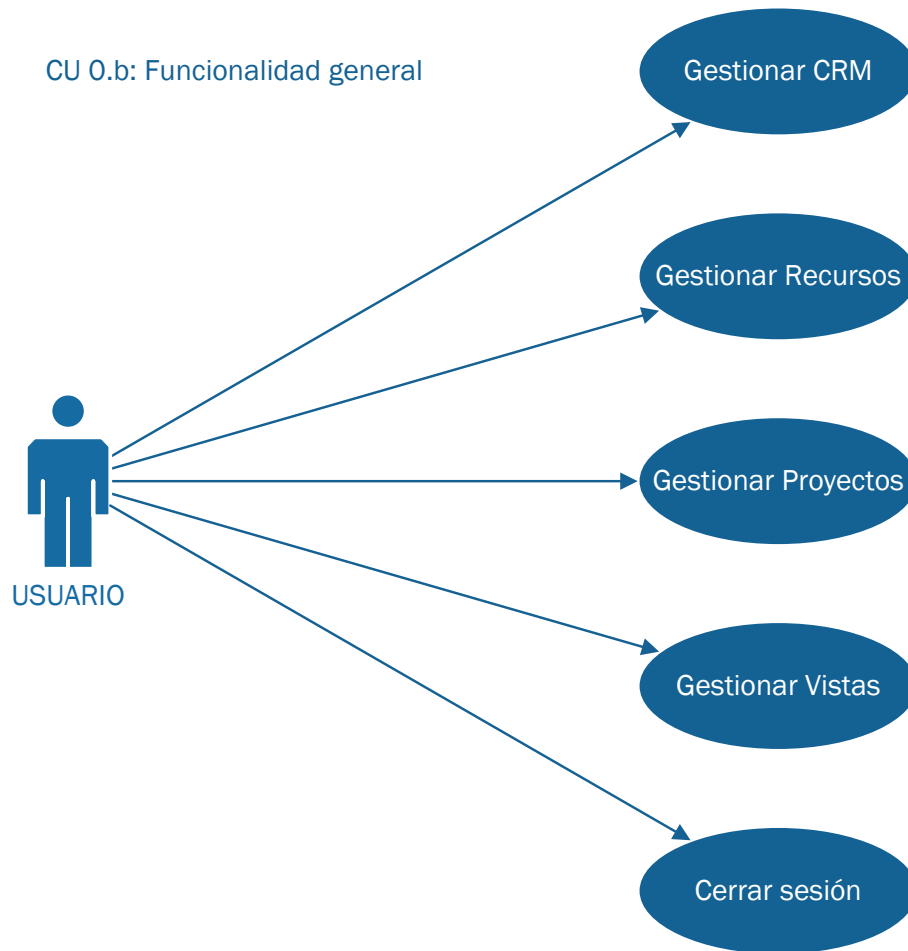


Figura 7.6: CU 0.b – Funcionalidad general

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



CU 2 - Gestionar Recursos	
Objetivo	Mostrar las funcionalidades de gestión de recursos de la aplicación
Acción	El usuario accede a la pestaña de Recursos de la aplicación
Tipo	Secundario
Precondiciones	Estar identificado en el sistema
Descripción	Se muestran en la barra superior las diferentes funcionalidades de gestión de recursos de la aplicación

**Tabla 7.5:** CU 2 - Gestionar Recursos

CU 3 - Gestionar Proyectos	
Objetivo	Mostrar las funcionalidades de gestión de proyectos de la aplicación
Acción	El usuario accede a la pestaña de Proyectos de la aplicación
Tipo	Secundario
Precondiciones	Estar identificado en el sistema
Descripción	Se muestran en la barra superior las diferentes funcionalidades de gestión de proyectos de la aplicación

**Tabla 7.6:** CU 3 - Gestionar Proyectos

CU 4 - Gestionar Vistas	
Objetivo	Mostrar las funcionalidades de gestión de vistas de la aplicación
Acción	El usuario accede a la pestaña de Proyectos de la aplicación
Tipo	Secundario
Precondiciones	Estar identificado en el sistema
Descripción	Se muestran en la barra superior las diferentes funcionalidades de gestión de vistas de la aplicación

**Tabla 7.7:** CU 4 - Gestionar Vistas

CU 5 - Cerrar sesión	
Objetivo	Desconectar al usuario de la aplicación y cerrarla
Acción	El usuario pulsa el botón de Cerrar sesión
Tipo	Primario
Precondiciones	Estar identificado en el sistema
Descripción	Se desconecta al usuario de la aplicación, se elimina la cookie del dispositivo y se cierra la aplicación

**Tabla 7.8:** CU 5 - Cerrar sesión

### 7.3.1.2 Casos de uso trazo fino



Figura 7.7: CU 1 - Gestionar CRM

CU 1.1 – Consultar clientes	
Objetivo	Consultar la información de los clientes en la base de datos
Acción	El usuario pulsa el botón de Clientes
Tipo	Secundario
Actor	Trabajador y Administrador
Precondiciones	Tener la pestaña de CRM activa en la aplicación
Descripción	Se muestra la tabla con la lista de los clientes junto a la información que se tiene de cada uno de ellos

Tabla 7.9: CU 1.1 Consultar clientes

CU 1.2 - Registrar interacción	
Objetivo	Almacenar en la base de datos una conversación que se ha tenido con un cliente
Acción	El usuario pulsa el botón de Registrar interacción
Tipo	Primario
Actor	Trabajador y Administrador
Precondiciones	Tener la pestaña de CRM activa en la aplicación
Descripción	El usuario rellena el formulario de una interacción con el cliente y se almacena la interacción en la base de datos

Tabla 7.10: 1.2 - Registrar interacción

CU 1.3 - Añadir cliente	
Objetivo	Introducir un cliente nuevo en la base de datos
Acción	El usuario pulsa el botón de Nuevo cliente
Tipo	Primario
Actor	Administrador
Precondiciones	Tener la pestaña de CRM activa en la aplicación
Descripción	El usuario rellena el formulario de un nuevo cliente y éste se registra en la base de datos

Tabla 7.11: CU 1.3 - Añadir cliente

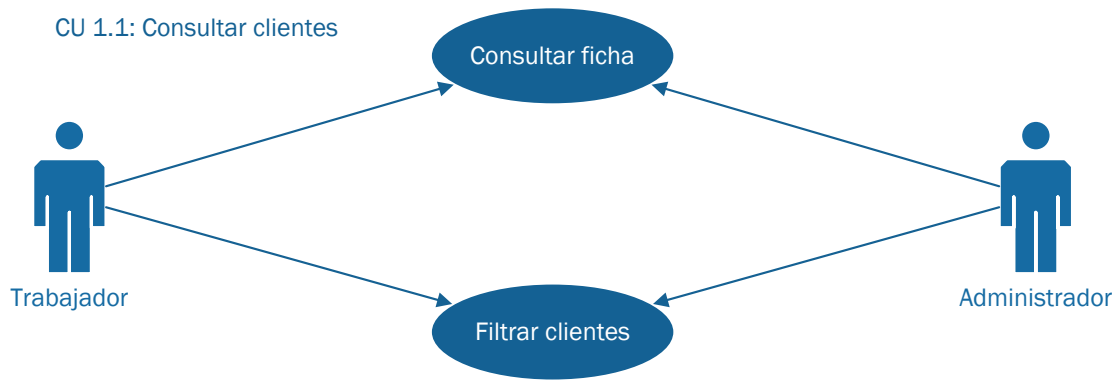


Figura 7.8: CU 1.1 - Consultar clientes

CU 1.1.1 - Consultar ficha	
Objetivo	Consultar la información de un cliente en particular
Acción	El usuario hace doble click en un cliente de la tabla
Tipo	Secundario
Actor	Trabajador y Administrador
Precondiciones	Tener abierta una tabla de información de clientes
Descripción	Se muestra la ficha de un determinado cliente al usuario, para poder ver más información sobre éste

Tabla 7.12: CU 1.1.1 - Consultar ficha

CU 1.1.2 - Filtrar clientes	
Objetivo	Encontrar un cliente o clientes en particular
Acción	El usuario busca en la caja de búsqueda o filtrando por columnas
Tipo	Primario
Actor	Trabajador y Administrador
Precondiciones	Tener abierta una tabla de información de clientes
Descripción	Se muestra un cliente o una lista de clientes que coincidan con la búsqueda o el filtro que haya establecido el usuario

Figura 7.9: CU 1.1.2 - Filtrar clientes

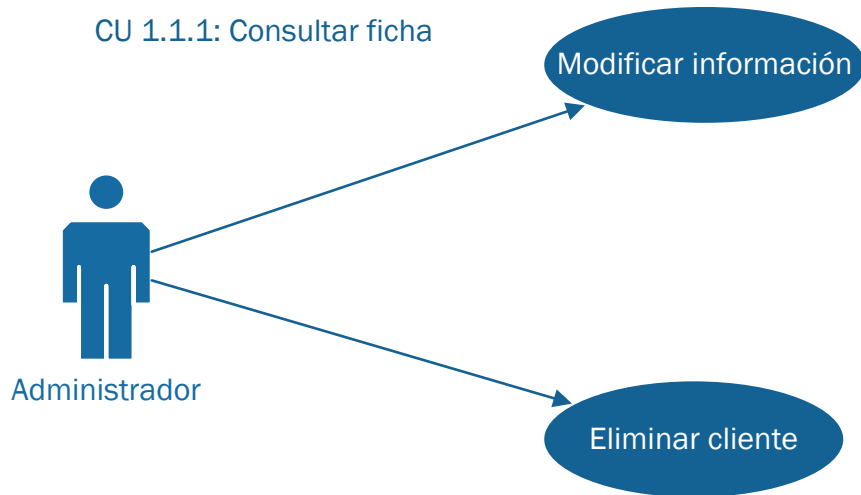


Figura 7.10: CU 1.1.1 - Consultar ficha

CU 1.1.1.1 - Modificar información	
Objetivo	Modificar la información de un cliente
Acción	El usuario tiene que modificar la información en el formulario de la ficha y pulsar guardar
Tipo	Primario
Actor	Administrador
Precondiciones	Tener abierta una ficha de un cliente
Descripción	Modifica la información del cliente de la ficha que está activa si se pulsa en Guardar en la ventana

Tabla 7.13: CU 1.1.1.1 - Modificar información

CU 1.1.1.2 - Eliminar cliente	
Objetivo	Eliminar un cliente de la base de datos
Acción	El usuario pulsa el botón Eliminar en la ficha del cliente
Tipo	Primario
Actor	Administrador
Precondiciones	Tener abierta una ficha de un cliente
Descripción	Elimina el cliente de la base de datos de la ficha activa

Tabla 7.14: CU 1.1.1.2 - Eliminar cliente

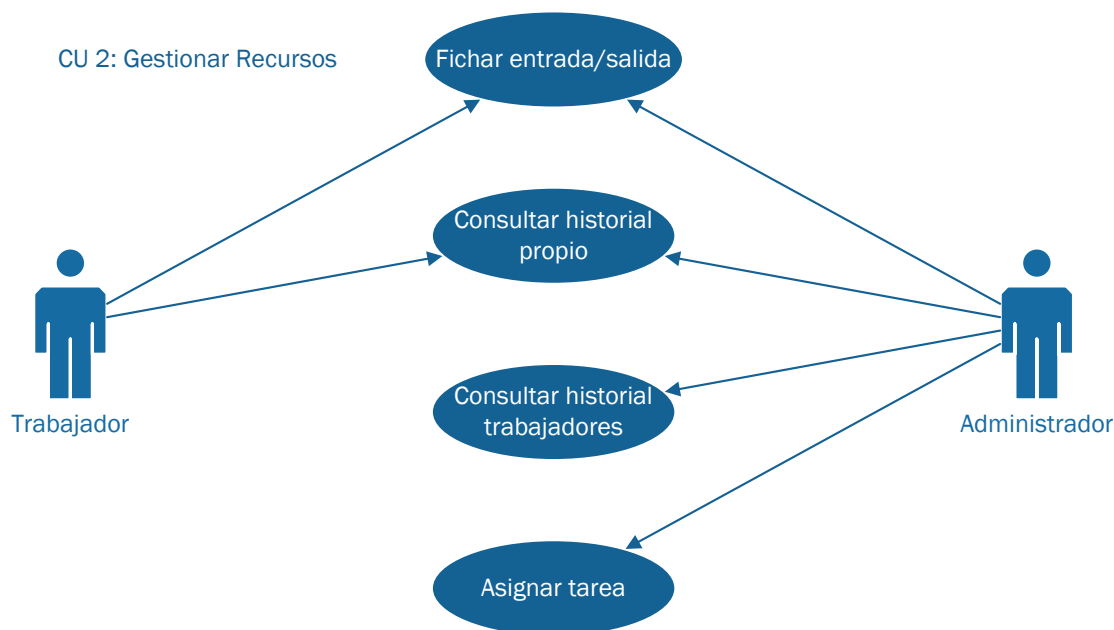


Figura 7.11: CU 2 - Gestionar Recursos

CU 2.1 - Fichar entrada/salida	
Objetivo	Fichar la entrada o salida de la jornada laboral del usuario
Acción	El usuario pulsa el botón Fichar en la barra superior
Tipo	Primario
Actor	Trabajador y Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Fichar
Descripción	Ficha la hora en la que empieza a trabajar el usuario si es la primera vez que le da, si no, fichará la hora a la que termina de trabajar

Tabla 7.15: CU 2.1 - Fichar entrada/salida

CU 2.2 - Consultar historial propio	
Objetivo	Mostrar el historial de fichajes del usuario
Acción	El usuario pulsa el botón Mostrar historial en la barra superior
Tipo	Primario
Actor	Trabajador y Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Mostrar historial
Descripción	Muestra el historial de fichajes del usuario que está conectado a la aplicación en ese momento

Tabla 7.16: CU 2.2 Consultar historial propio

CU 2.3 - Consultar historial trabajadores	
Objetivo	Mostrar el historial de fichajes de los trabajadores de la empresa
Acción	El usuario pulsa el botón Mostrar historial trabajadores en la barra superior
Tipo	Primario
Actor	Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Mostrar historial
Descripción	Muestra el historial de fichajes de los usuarios trabajadores de la empresa. Dando doble click en uno de ellos muestra la ficha del seleccionado

Tabla 7.17: CU 2.3 - Consultar historial trabajadores

CU 2.4 - Asignar tarea	
Objetivo	Asignar una tarea existente o nueva a un usuario en particular
Acción	El usuario pulsa el botón Asignar tarea en la barra superior
Tipo	Primario
Actor	Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Asignar tarea
Descripción	Asigna una tarea ya creada o nueva a un usuario del sistema, pudiendo ponerle fechas límite y enlazarlo en el calendario de éste

Tabla 7.18: CU 2.4 Asignar tarea

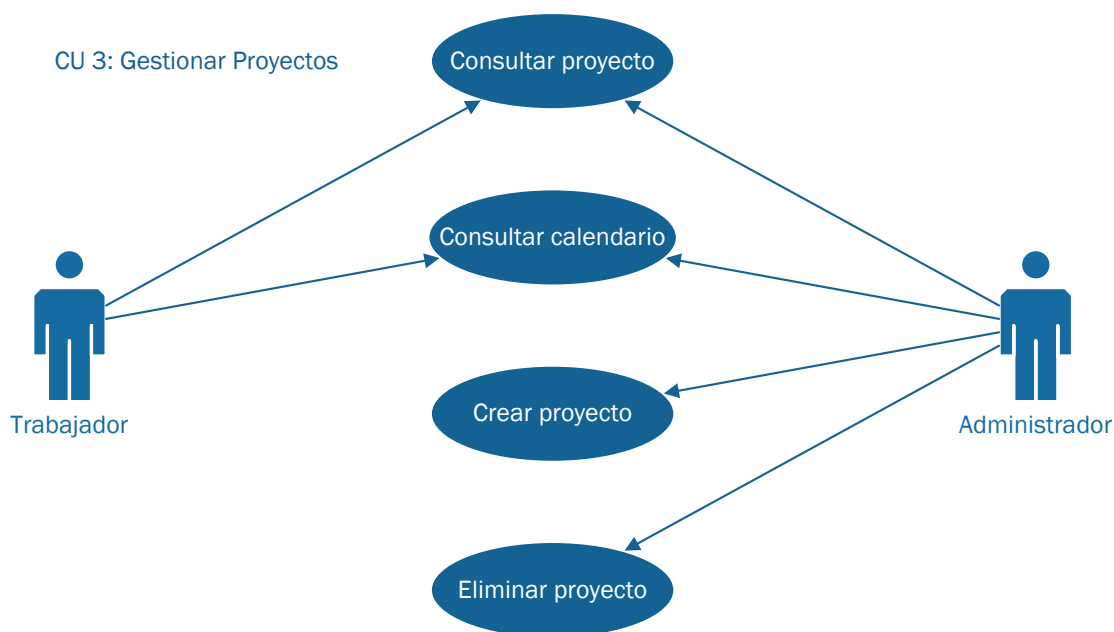


Figura 7.12: CU 3 - Gestionar Proyectos

CU 3.1 - Consultar proyecto	
Objetivo	Consultar la información de un proyecto en marcha
Acción	El usuario pulsa el botón Consultar proyecto tarea en la barra superior
Tipo	Primario
Actor	Trabajador y Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Consultar proyecto
Descripción	Permite consultar la información y estado de un proyecto activo de la empresa en ese momento, pudiendo buscar uno en concreto

Tabla 7.19: CU 3.1 - Consultar proyecto

CU 3.2 - Consultar calendario	
Objetivo	Consultar el calendario del usuario activo
Acción	El usuario pulsa el botón Consultar calendario tarea en la barra superior
Tipo	Primario
Actor	Trabajador y Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Consultar calendario
Descripción	Permite consultar el calendario del usuario que está conectado actualmente en la aplicación, pudiendo ver las fechas límite y alertas

**Tabla 7.20:** CU 3.2 - Consultar calendario

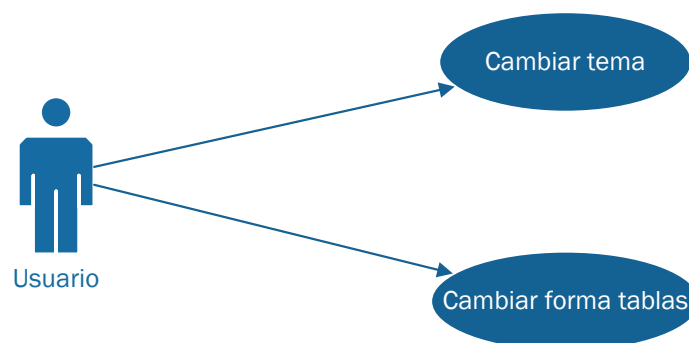
CU 3.3 - Crear proyecto	
Objetivo	Crear un nuevo proyecto empresarial
Acción	El usuario pulsa el botón Crear proyecto tarea en la barra superior
Tipo	Primario
Actor	Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Crear proyecto
Descripción	Genera un nuevo proyecto en la base de datos, dando la posibilidad de dividirlo en tareas que luego se pueden asignar a los trabajadores

**Tabla 7.21:** CU 3.3 - Crear proyecto

CU 3.4 - Eliminar proyecto	
Objetivo	Eliminar un proyecto en la base de datos
Acción	El usuario pulsa el botón Eliminar proyecto tarea en la barra superior
Tipo	Primario
Actor	Administrador
Precondiciones	Tener activa la pestaña en la que esté el botón Eliminar proyecto
Descripción	Elimina el proyecto seleccionado de la base de datos

**Tabla 7.22:** CU 3.4 Eliminar proyecto

#### CU 4: Gestionar Vistas



**Figura 7.13:** CU 4 - Gestionar Vistas

CU 4.1 - Cambiar tema	
Objetivo	Cambiar el tema de la aplicación
Acción	El usuario elige el tema que desea del selector de temas
Tipo	Primario
Actor	Usuario
Precondiciones	Tener activa la pestaña en la que esté el selector de temas
Descripción	Modifica los colores de la aplicación en función del tema seleccionado

**Tabla 7.23:** CU 4.1 - Cambiar tema

CU 4.2 - Cambiar forma tablas	
Objetivo	Cambiar cómo se muestran las tablas
Acción	El usuario elige la forma en la que se muestran las tablas en la barra superior
Tipo	Primario
Actor	Usuario
Precondiciones	Tener activa la pestaña en la que estén las opciones de tablas
Descripción	Modifica la forma en la que se muestran las tablas al usuario

**Tabla 7.24:** CU 4.2 - Cambiar forma tablas

## 7.4 Descripción de la información

En este apartado se procederá a describir la información manipulada por el sistema. En primer lugar, se analizarán las clases y las asociaciones entre las mismas, que han sido extraídas del estudio del problema. Por último, se pasará a obtener el modelo conceptual.

Para el modelado de la información y de las funciones del sistema en el desarrollo del proceso software se utilizará UML (Unified Modeling Language) como notación, ya que se ha convertido en un estándar actualmente.

Del estudio del problema y con el fin de alcanzar los objetivos marcados se ha establecido una serie de supuestos que deberá cumplir el sistema que se va a desarrollar.

- **Supuesto 1:** Se desea mantener información de los clientes que tiene la empresa, almacenando todos los datos necesarios para identificar a cada uno de ellos inequívocamente.
- **Supuesto 2:** De cada cliente se almacenará como mínimo el nombre, la razón social, un número de documento de identidad, un número de teléfono y un e-mail de contacto.
- **Supuesto 3:** Cada interacción con un cliente se almacenará de forma que quede enlazada con dicho cliente y el trabajador o trabajadores que la realizaron. Cada cliente puede tener muchas interacciones con diversos trabajadores, y viceversa.



- **Supuesto 4:** Se desea almacenar información de todos los trabajadores de la empresa, con los datos suficientes para identificarlos inequívocamente.
- **Supuesto 5:** De cada trabajador se almacenarán el nombre y apellidos, la razón social, el puesto de trabajo, el NIF, el número de teléfono y el correo electrónico de empresa como mínimo.
- **Supuesto 6:** Los fichajes de entrada y salida se almacenarán como una sola entrada en la base de datos, calculándose en ésta las horas trabajadas en una sesión. Estos fichajes quedarán enlazados con el usuario que los genera.
- **Supuesto 7:** De cada fichaje de entrada y salida se almacenará la hora de entrada, la hora de salida, la diferencia entre éstas, el usuario que las registró y la información del dispositivo en la que se ha generado como mínimo, pudiendo introducir una breve descripción.
- **Supuesto 8:** Las tareas pueden ser asignadas a uno o varios usuarios. No tienen por qué pertenecer a un proyecto.
- **Supuesto 9:** Cada tarea estará compuesta por un nombre, una descripción, una fecha límite, los trabajadores asignados y el proyecto al que está asignado. Estos tres últimos datos en el caso en el que existieran para dicha tarea.
- **Supuesto 10:** Se desea guardar información sobre los proyectos de la empresa, pudiendo ser estos divididos en tareas asignables a los trabajadores.
- **Supuesto 11:** De cada proyecto se almacenará el cliente destinatario (si lo hubiera), el nombre, la razón, una descripción y la fecha de inicio y final estimado del mismo. También quedará enlazado a las posibles tareas en las que pueda estar dividido.
- **Supuesto 12:** Se guardará de una sesión a otra el tema seleccionado por el usuario, de forma que no tenga que estar cambiándolo cada vez que inicie la aplicación.

### 7.4.1 Análisis de las clases

En este apartado se realizará un análisis detallado de las clases extraídas en del análisis del problema. Se identificarán y definirán claramente cada una de ellas, las responsabilidades que tienen asociadas, así como los atributos y operaciones necesarias para cumplir esas responsabilidades. Puesto que el problema objeto del análisis ha sido la creación de un nuevo módulo para una herramienta ya existente, se detallarán las clases creadas y aquellas existentes que hayan sufrido alguna modificación.

Para realizar este análisis se presentará una representación gráfica de cada clase y una explicación más detallada sobre los distintos elementos que la componen.

Las clases que se han identificado en el análisis del sistema son aquellas que se van a utilizar para dar funcionalidad al sistema, puesto que en .NET las clases que se generan pueden ser clases instanciables o clases estáticas. Las clases instanciables son aquellas que poseen uno o varios constructores y un destructor, pudiendo crear objetos (instancias) de la propia clase. Las clases estáticas se utilizan como contenedores de conjuntos de métodos que sólo funcionan en parámetros de entrada, sin necesidad de obtener o establecer campos internos de instancia. Se podría decir que vienen a ser las bibliotecas de funciones. Por tanto, en este apartado se van a identificar las clases instanciables del sistema, puesto que las otras muestran métodos y algoritmos propios de la empresa.

#### 7.4.1.1 Clase Usuario

Esta clase representa a cualquier usuario de la aplicación, bien sea trabajador de la empresa o bien sea administrador también. Implementa todos los atributos y métodos necesarios para otorgar la funcionalidad deseada a un usuario.

##### Atributos

- **nombre:** Cadena que almacena el nombre del usuario.
- **apellidos:** Cadena que almacena los apellidos del usuario.
- **adm:** Indica si el usuario es administrador (1) o no (0).
- **id:** Valor numérico que identifica inequívocamente a un usuario.
- **razonSocial:** Cadena que almacena la razón social del usuario.
- **NIF:** Cadena que almacena el NIF del usuario.
- **telefono:** Valor numérico que contiene el número de teléfono del usuario.
- **correo:** Cadena que almacena el correo electrónico de empresa del usuario.
- **puesto:** Cadena que almacena el puesto de trabajo en la empresa del usuario.
- **foto:** Imagen que muestra una foto que identifica al usuario.
- **fechaAlta:** Fecha que representa la fecha en la que el usuario se contrató.

##### Métodos

- **Trabajador(nombre, apellidos, adm, id, ?razonSocial, NIF, teléfono, correo, puesto, ?foto, fechaAlta):** Es el constructor de la clase. Genera una instancia para representar al usuario conectado al sistema en este momento.
- **fichar():** Es la función que genera un fichaje de entrada/salida.
- **registrar():** Registra al usuario en la base de datos.
- **eliminar():** Elimina al usuario en cuestión de la base de datos.

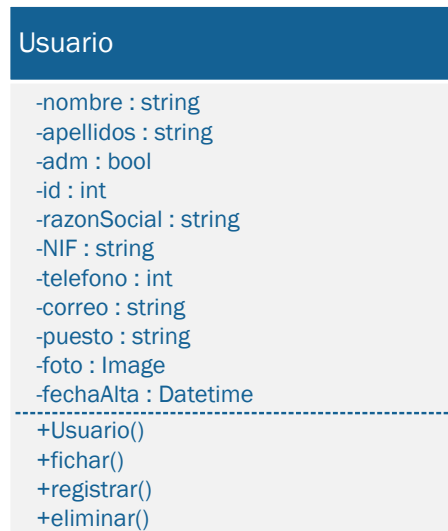


Figura 7.14: Clase Usuario

### 7.4.1.2 Clase Fichaje

Esta clase implementa los fichajes de entrada/salida de las sesiones de trabajo que pueden realizar los usuarios de la aplicación. Permite registrarlas en la base de datos.

#### Atributos

- **entrada:** Fecha y hora que indica el momento en el que el usuario comienza a trabajar.
- **salida:** Fecha y hora que indica el momento en el que el usuario termina de trabajar.
- **horas:** Valor numérico que indica el número de horas de trabajo que el usuario ha realizado en esta sesión.
- **idUser:** Valor numérico que identifica inequívocamente al usuario que realiza el fichaje.
- **info:** Cadena que almacena la información del dispositivo desde el que se ficha.
- **asunto:** Cadena que almacena información general sobre la sesión de trabajo.

#### Métodos

- **Fichaje(idUser):** Es el constructor de la clase. Genera una instancia que representa un fichaje de un usuario en el sistema. Se le pasa el id del usuario conectado actualmente.
- **registrar():** Registra el fichaje en la base de datos. Devuelve *true* si se ha realizado con éxito y *false* en caso contrario.



Figura 7.15: Clase Fichaje

### 7.4.1.3 Clase Cliente

Esta clase representa a los clientes que puede tener la empresa registrados en el sistema. Almacena toda la información sobre ellos y permite introducirlos o eliminarlos de la base de datos.

#### Atributos

- **nombre:** Cadena que almacena del nombre del cliente.
- **id:** Valor numérico que identifica inequívocamente a un cliente.
- **razonSocial:** Cadena que almacena la razón social del cliente.
- **CIF:** Cadena que almacena el CIF del cliente.
- **telefono:** Valor numérico que contiene el número de teléfono del cliente.
- **correo:** Cadena que almacena el correo electrónico de empresa del cliente.
- **logo:** Imagen que muestra el logotipo del cliente.
- **fechaAlta:** Fecha que representa la fecha en la que se hizo el contrato con el cliente.

#### Métodos

- **Cliente(nombre, id, ?razonSocial, CIF, teléfono, correo, ?foto, fechaAlta):** Es el constructor de la clase. Genera una instancia para representar a un cliente de la empresa.
- **registrar():** Registra al cliente en la base de datos.
- **eliminar():** Elimina al cliente de la base de datos.

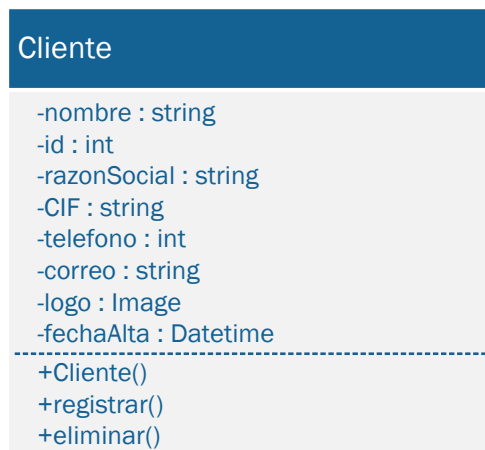


Figura 7.16: Clase Cliente

### 7.4.1.4 Clase Interaccion

#### Atributos

- **idCliente:** Valor numérico que identifica inequívocamente el cliente con el que se realiza la interacción.
- **idUsuario:** Valor numérico que identifica inequívocamente al usuario que genera la interacción.
- **id:** Valor numérico que identifica inequívocamente a una interacción.
- **fecha:** Fecha y hora en la que se realizó la interacción
- **asunto:** Cadena que contiene una descripción que explica en qué consistió la interacción.

## Métodos

- **Interaccion**(idCliente, idUsuario, ?participantes, fecha, ?asunto): Es el constructor de la clase. Genera una instancia para representar a una interacción del usuario con un cliente.
- **registrar()**: Registra la interacción en la base de datos.
- **eliminar()**: Elimina la interacción de la base de datos.

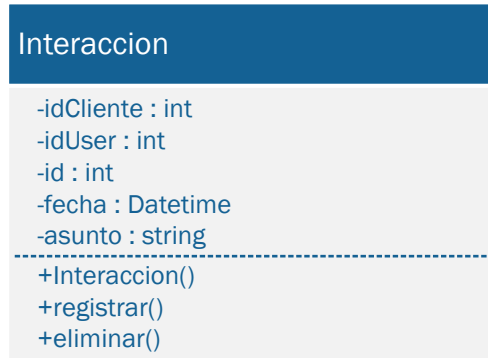


Figura 7.17: Clase Interaccion

### 7.4.1.5 Clase Proyecto

#### Atributos

- **nombre**: Cadena que almacena el nombre del proyecto.
- **id**: Valor numérico que identifica inequívocamente a un proyecto.
- **idCliente**: Valor numérico que identifica el cliente para el que se va a hacer el proyecto.
- **idUser**: Valor numérico que identifica el usuario que genera el proyecto.
- **participantes**: Vector de valores numéricos que identifican inequívocamente a los usuarios que participan en la interacción, si es que existen.
- **fechaInicio**: Fecha en la que se crea e inicia el proyecto.
- **fechaFinal**: Fecha que indica el final estimado del proyecto.
- **tareas**: Vector de valores numéricos que identifican las tareas en las que se divide el proyecto.

#### Métodos

- **Proyecto**(nombre, id, idCliente, idUser, ?participantes, fechaInicio, fechaFinal, ?tareas): Es el constructor de la clase. Genera una instancia para representar un proyecto de la empresa.
- **asignarTarea**(int id): Es la función que asigna una tarea al proyecto, introduciendo su id en el vector *tareas*.
- **registrar()**: Registra al proyecto en la base de datos.
- **eliminar()**: Elimina al proyecto de la base de datos.



Figura 7.18: Clase Proyecto

#### 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

Las clases del Framework .NET poseen unas variables llamadas Propiedades, que pueden ser llamadas como si fueran variables públicas. Las Propiedades se utilizan para proporcionar un acceso controlado a los detalles internos de una clase. Con ellas se generan los métodos *get* y *set* de las variables de las clases, pudiendo modificar el código de estas funciones para que hagan más que leer y escribir el valor de una variable interna. Todas las variables, exceptuando los id propios de cada clase y a la clase Fichaje, harán uso de las Propiedades, motivo por el que no se han descrito métodos *get* y *set* anteriormente para los atributos de las clases.

## 7.4.2 Análisis de las asociaciones

### 7.4.2.1 Asociación Usuario-Fichaje

Este tipo de asociación establece la relación entre un usuario y un fichaje de entrada/salida.

Un usuario puede poseer más de un fichaje, o incluso ninguno, y un fichaje sólo puede pertenecer a un usuario. Por lo tanto, la clase Usuario interviene con cardinalidad mínima y máxima 1, mientras que la clase Fichaje interviene con la cardinalidad mínima 0 y la máxima n.

### 7.4.2.2 Asociación Usuario-Interaccion

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

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

### 7.4.2.3 Asociación Usuario-Proyecto

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

Un usuario puede poseer más de un proyecto, o incluso ninguno, y un proyecto sólo puede ser creado por un usuario. Por lo tanto, la clase Usuario interviene con cardinalidad mínima y máxima 1, mientras que la clase Proyecto interviene con la cardinalidad mínima 0 y la máxima n.

### 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.

### 7.4.2.5 Asociación Cliente-Interaccion

Este tipo de asociación establece la relación entre un cliente y una interacción con un usuario.

Un cliente puede poseer más de una interacción, o incluso ninguna, y una interacción sólo puede pertenecer a un usuario. Por lo tanto, la clase Cliente interviene con cardinalidad mínima y máxima 1, mientras que la clase Interaccion interviene con la cardinalidad mínima 0 y la máxima n.

#### *7.4.2.6 Asociación Proyecto-Usuario*

Este tipo de asociación establece la relación entre un proyecto y sus participantes.

Un proyecto puede poseer uno o más de un participante, y un usuario puede participar en ninguno o muchos proyectos. Por lo tanto, la clase proyecto interviene con cardinalidad mínima 0 y máxima n, mientras que la clase Usuario interviene con la cardinalidad mínima 1 y la máxima n.

#### *7.4.2.7 Asociación Proyecto-Cliente*

Este tipo de asociación establece la relación entre un proyecto y el cliente para el que se realiza.

Un proyecto puede poseer un cliente, y un cliente puede pertenecer a ninguno, a uno o a más de un proyecto. Por lo tanto, la clase Proyecto interviene con cardinalidad mínima 0 y máxima n, mientras que la clase Cliente interviene con la cardinalidad mínima y máxima 1.

#### *7.4.2.8 Asociación Proyecto-Tarea*

Este tipo de asociación establece la relación entre un proyecto y las tareas en las que se puede dividir.

Un proyecto puede poseer ninguna, una o más tareas, y una tarea puede pertenecer a un proyecto. Por lo tanto, la clase Proyecto interviene con cardinalidad mínima 0 y máxima n, mientras que la clase Tarea interviene con la cardinalidad mínima y máxima 1.

#### *7.4.2.9 Asociación Tarea-Usuario*

Este tipo de asociación establece la relación entre una tarea y el usuario que la realiza.

Una tarea puede ser realizada por un usuario o varios, y un usuario puede realizar ninguna, una o más de una tarea. Por lo tanto, la clase Tarea interviene con cardinalidad mínima 0 y máxima n, mientras que la clase Usuario interviene con la cardinalidad mínima 1 y máxima n.

#### *7.4.2.10 Asociación Tarea-Tarea*

Este tipo de asociación establece la relación entre una tarea y las tareas en las que se puede dividir.

Una tarea puede poseer ninguna, una o más tareas, y una tarea puede pertenecer ninguna o a una tarea. Por lo tanto, la clase Tarea interviene con cardinalidad mínima 0 y máxima n en ambos extremos.



### 7.4.3 Diagrama de clases

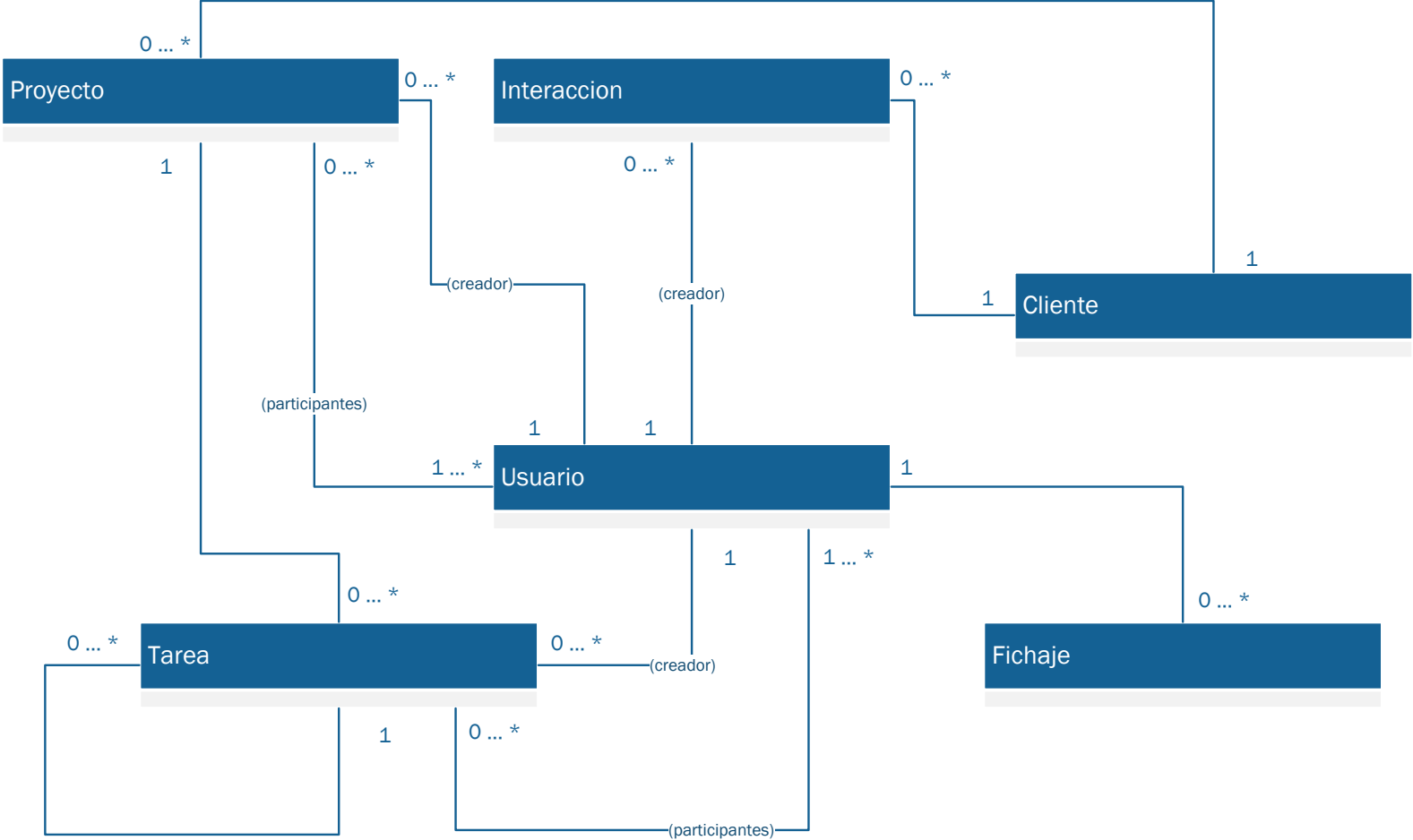


Figura 7.20: Diagrama de Clases

## 8 Especificación de la interfaz

---

### 8.1 Introducción

En este capítulo se realizará una descripción de las características que deberá poseer la interfaz de la aplicación AlephCRM.

### 8.2 Características de la interfaz

La interfaz seguirá la línea de diseño de las aplicaciones profesionales del paquete Office de Microsoft. Estas aplicaciones se caracterizan por tener una barra en la parte superior, en la que se dividen en categorías las diferentes funcionalidades que tienen. Véase como ejemplo Microsoft Word o Microsoft Excel. Las características de la interfaz se podrían resumir en:

- 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.
- Las ventanas de los editores de informes deberán tener una interfaz sencilla, similar al programa Word de Microsoft, pero sólo con las funciones fundamentales. La opción de exportación del archivo deberá estar bien señalizada.
- Los editores de información deberán ser comprender un formulario para modificar los datos esenciales de los sujetos, no pudiendo modificar sus números de identificación o sus roles desde la aplicación. Cada elemento del formulario deberá admitir de entrada el tipo de dato que coincida con la propiedad que lo almacena en la base de datos (char, int, datetime, etc).
- Los botones de la barra superior deberán tener un nombre lo suficientemente descriptivo como para saber qué función realiza. Por si con esto no fuera suficiente, deberán mostrar un diálogo que explique brevemente su funcionalidad.
- La opción de cambiar de tema de la aplicación deberá estar disponible en todo momento, y se debe guardar de una sesión a otra. Los temas deben ser completos y modificar de forma consistente la apariencia del programa.
- Con el usuario conectado, se debe mostrar en todo momento el nombre del usuario completo, junto con su rol. Debe tener acceso en cualquier momento a la opción de cerrar sesión.

- A la hora de crear un sujeto nuevo se rellenará un formulario similar al de edición de los datos de los sujetos. Los campos de los formularios variarán en función de si se trata de un cliente o un trabajador, pero serán visualmente similares para no confundir al usuario.
- Cuando se esté visualizando una tabla, el usuario deberá poder buscar en ella en cualquier momento, al igual que filtrar por algún valor de las columnas u ordenarlas de diferentes formas.
- El formulario de inicio de sesión aparecerá nada más arrancar la aplicación. Contará con dos cajas de texto, una para introducir el e-mail y otra para introducir la contraseña. La caja de texto de la contraseña deberá ocultar ésta mientras se escribe. Una vez iniciada la sesión correctamente, se lanzará la interfaz de la ventana principal de la aplicación.
- El tamaño de la ventana principal de la aplicación podrá ser modificado por el usuario a su antojo, estando en modo ventana independiente. No ocurrirá lo mismo con las ventanas de los editores.
- Los mensajes de error aparecerán cuando algo no funcione en la aplicación y remitirán a la ventana de información para ponerse en contacto con el servicio técnico.



# SECCIÓN III

# DISEÑO



## 9 Diseño de datos

---

### 9.1 Introducción

En este capítulo se definirá la estructura de datos que usará el sistema. Los datos de la aplicación estarán almacenados en una base de datos relacional, de esta forma se representarán las relaciones y restricciones lógicas que surgen en el desarrollo del sistema.

### 9.2 Definición de las tablas

En este apartado se definirá el esquema lógico de la Base de Datos que empleará el sistema. Como modelo lógico se ha elegido el modelo relacional, el cual se irá obteniendo a partir del modelo de clases obtenido en el capítulo 7, extrayendo de este modelo de clases aquellos objetos que deban ser persistentes, esto es, aquellos objetos cuya información deba ser almacenada. No se considerarán persistentes aquellos objetos que sean creados por el sistema para realizar una operación interna.

En la definición de las tablas se utilizará la siguiente notación:

- **Clave principal:** el identificador principal de cada tabla se indicará con un subrayado simple
- **Clave foránea:** los atributos que sean referencias a claves de otras tablas se indicarán en negrita.
- **Clave alterna:** los atributos que pueden identificar por si solos a una tabla, pero que no forman parte de la clave principal se indicarán mediante un doble subrayado.
- **Otros atributos:** el resto de atributos de las tablas, los que no forman parte de ninguna de las claves indicadas anteriormente, se indicarán sin ningún formato especial.

Pueden existir atributos que sean claves principales y foráneas, por lo que se representarán mediante subrayado simple y negrita.

#### 9.2.1.1 Tabla USUARIOS

La estructura de la tabla USUARIOS es la siguiente:

USUARIOS(Usuarioid, Nombre, Apellidos, Admin, RazonSocial, NIF, Telefono, Correo, Puesto, Foto, FechaAlta)

### 9.2.1.2 Tabla CLIENTES

La estructura de la tabla CLIENTES es la siguiente:

CLIENTES(Clienteld, Nombre, RazonSocial, CIF, Telefono, Correo, Logo, FechaAlta, **UsuarioAlta**)

### 9.2.1.3 Tabla FICHAJES

La estructura de la tabla FICHAJES es la siguiente:

FICHAJES(Fichajeld, Entrada, Salida, Horas, Informacion, Asunto, **UsuarioAlta**)

### 9.2.1.4 Tabla INTERACCIONES

La estructura de la tabla INTERACCIONES es la siguiente:

INTERACCIONES(InteraccionId, Asunto, FechaAlta, **Cliente**, **UsuarioAlta**)

### 9.2.1.5 Tabla PROYECTOS

La estructura de la tabla PROYECTOS es la siguiente:

PROYECTOS(Proyectold, Nombre, Inicio, Final, Terminado, FechaAlta, **Cliente**, **UsuarioAlta**)

### 9.2.1.6 Tabla TAREAS

La estructura de la tabla TAREAS es la siguiente:

TAREAS(Tareald, Nombre, Inicio, Final, Terminado, FechaAlta, **TareaSuperior**, **Cliente**, **Proyecto**, **UsuarioAlta**)

### 9.2.1.7 Tabla PARTIPROY

Esta tabla representa la relación de usuarios participantes de un proyecto. La estructura de la tabla PARTIPROY es la siguiente:

PARTIPROY(Id, **Proyecto**, **Usuario**)

### 9.2.1.8 Tabla PARTITAREAS

Esta tabla representa la relación de usuarios participantes de una tarea. La estructura de la tabla PARTITAREAS es la siguiente:

PARTITAREA(Id, **Tarea**, **Usuario**)



## 9.3 Esquema relacional

A continuación, se definirán los dominios y las restricciones de los atributos de cada tabla, así como las opciones de borrado y actualización entre las tablas relacionadas.

El conjunto de tablas que constituyen la base de datos que se usará en el sistema será el siguiente:

USUARIOS				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Usuarioid	int	Principal	No	-
Nombre	nvarchar(30)	-	No	-
Apellidos	nvarchar(60)	-	No	-
Admin	bit	-	Sí	0
RazonSocial	nvarchar(80)	-	Sí	-
NIF	nvarchar(16)	-	No	-
Telefono	nvarchar(15)	-	Sí	-
Correo	nvarchar(60)	-	No	-
Puesto	nvarchar(60)	-	Sí	-
Foto	varbinary(MAX)	-	Sí	-
FechaAlta	datetime	-	No	getdate()

Tabla 9.1: Tabla USUARIOS

CLIENTES				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Clienteld	int	Principal	No	-
Nombre	nvarchar(30)	-	No	-
RazonSocial	nvarchar(80)	-	Sí	-
CIF	nvarchar(16)	-	No	-
Telefono	nvarchar(15)	-	Sí	-
Correo	nvarchar(60)	-	No	-
Logo	varbinary(MAX)	-	Sí	-
FechaAlta	datetime	-	No	getdate()
UsuarioAlta	int	Foránea	No	-

Tabla 9.2: Tabla CLIENTES

FICHAJES				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Fichajeld	int	Principal	No	-
Entrada	datetime	-	No	-
Salida	datetime	-	Sí	-
Horas	float	-	Sí	0
Informacion	nvarchar(150)	-	No	-
Asunto	nvarchar(40)	-	Sí	-
UsuarioAlta	int	Foránea	No	-

Tabla 9.3: Tabla FICHAJES

INTERACCIONES				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
InteraccionId	int	Principal	No	-
Asunto	nvarchar(150)	-	Sí	-
FechaAlta	datetime	-	No	getdate()
Cliente	int	Foránea	No	-
UsuarioAlta	int	Foránea	No	-

Tabla 9.4: Tabla INTERACCIONES

PROYECTOS				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Proyectold	int	Principal	No	-
Nombre	nvarchar(50)	-	No	-
Cliente	int	Foránea	No	-
Inicio	datetime	-	Sí	-
Final	datetime	-	Sí	-
Terminado	bit	-	Sí	0
FechaAlta	datetime	-	No	getdate()
UsuarioAlta	int	Foránea	No	-

Tabla 9.5: Tabla PROYECTOS

TAREAS				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Tareald	int	Principal	No	-
Nombre	nvarchar(15)	-	No	-
Inicio	datetime	-	Sí	-
Final	datetime	-	Sí	-
Terminada	bit	-	Sí	0
FechaAlta	datetime	-	No	getdate()
UsuarioAlta	int	Foránea	No	-
TareaSuperior	int	Foránea	Sí	-
Cliente	int	Foránea	Sí	-
Proyecto	int	Foránea	Sí	-

Tabla 9.6: Tabla TAREAS

PARTIPROY				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Id	int	Principal	No	-
Proyecto	int	Foránea	No	-
Usuario	int	Foránea	No	-

**Tabla 9.7:** Tabla PARTIPROY

PARTITAREAS				
Atributo	Tipo	Clave	Permitir valor nulo	Valor predeterminado
Id	int	Principal	No	-
Tarea	int	Foránea	No	-
Usuario	int	Foránea	No	-

**Tabla 9.8:** Tabla PARTITAREAS

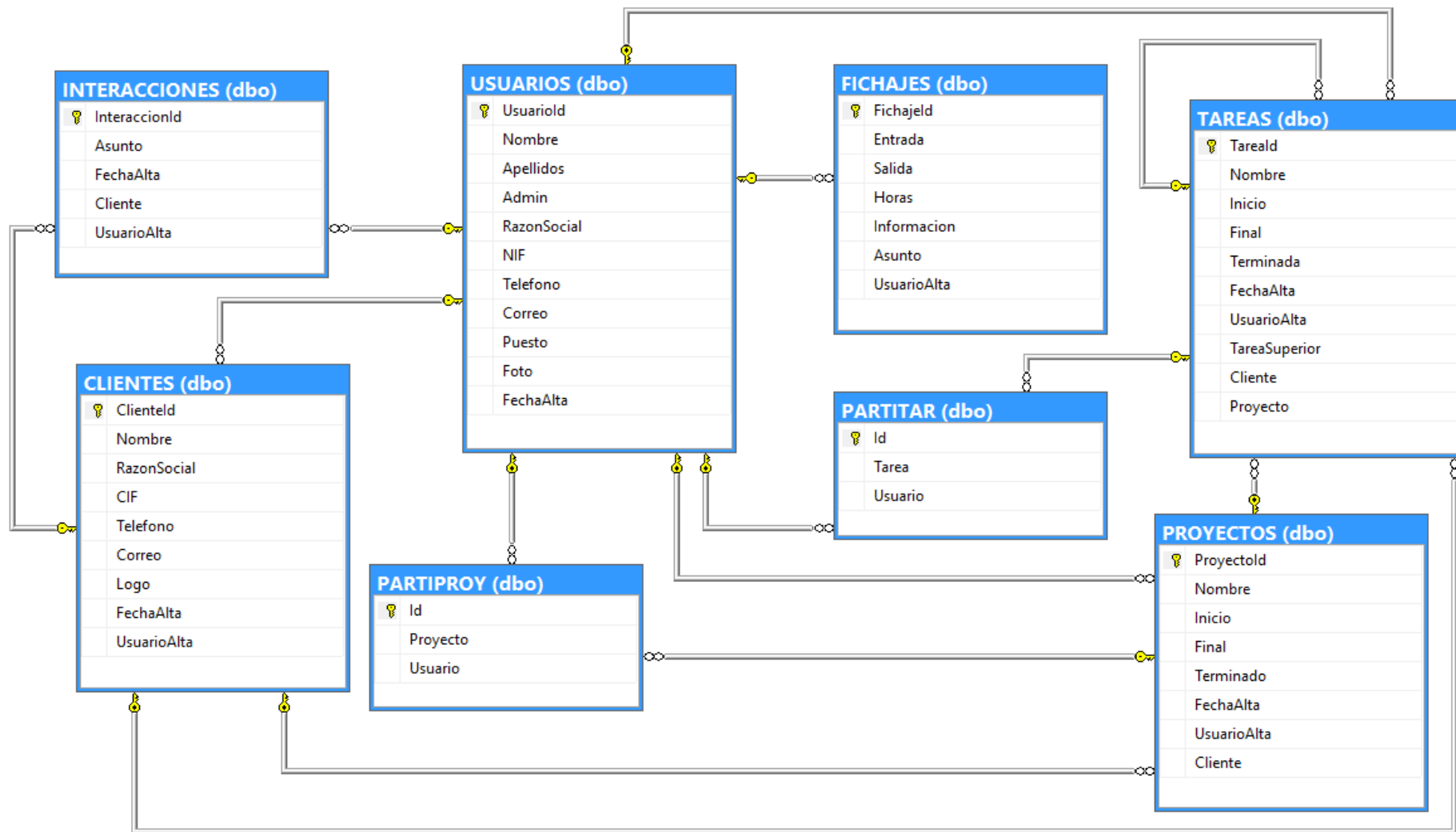


Figura 9.1: Diagrama Esquema Relacional

## 10 Diseño arquitectónico

---

### 10.1 Introducción

En este capítulo se definirá la arquitectura del sistema. Para ello se utilizarán los diagramas de despliegue, que constituyen una de las técnicas que nos ofrece UML para realizar el diseño arquitectónico. Representan la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Muestran la vista de despliegue estática de una arquitectura y se relacionan con los componentes ya que, por lo común, los nodos contienen uno o más componentes.

### 10.2 Diagrama de despliegue

Este sistema presenta claramente una arquitectura Cliente-Servidor. En este caso, el servidor se compone por una base de datos y un Motor de base de datos. Ambos están contenidos en Microsoft Azure, junto con la autenticación mediante Active Directory. El cliente es el responsable de la presentación, la entrada de datos, la gestión de los módulos de la aplicación y la validación sencilla de entradas. Los componentes principales de la arquitectura del sistema se basan en los siguientes conceptos:

- **Azure:** Es una colección de servicios integrados en la nube de Microsoft. Todos estos servicios están integrados unos con otros, manejándose desde el Portal de Azure en cualquier navegador. Se encarga de administrar los servicios que utiliza la aplicación.
  - **Servidor de base de datos (SQL Server):** Para almacenar los datos persistentes de la aplicación se dispondrá de una base de datos compuesta por una serie de tablas. El servidor de bases de datos es el encargado de procesar todas las peticiones sobre la base de datos. Se utilizará SQL Server como servidor de base de datos.
  - **Motor de base de datos:** Se encarga de la conexión con la base de datos desde el cliente y de almacenar, procesar y proteger los datos.
  - **Active Directory:** Se trata de un administrador de identidad y acceso a la nube de Azure. Se encarga de validar a los usuarios para que puedan acceder a la aplicación. Sólo se validarán correctamente si están autorizados desde el Portal de Azure.
- **Aplicación cliente:** Se trata de una instancia de la aplicación de escritorio Windows AlephCRM, que estará siendo utilizada por un usuario de la empresa. En ella realizarán diferentes gestiones y procedimientos, en los cuales se harán llamadas a la base de datos, bien para almacenar datos o bien para leerlos. También se encarga de realizar la llamada al Active Directory para la identificación de los usuarios usando las credenciales de la empresa.

La arquitectura del sistema está representada por el diagrama de la figura 10.1, en el que se puede observar cómo interactúan los elementos explicados.

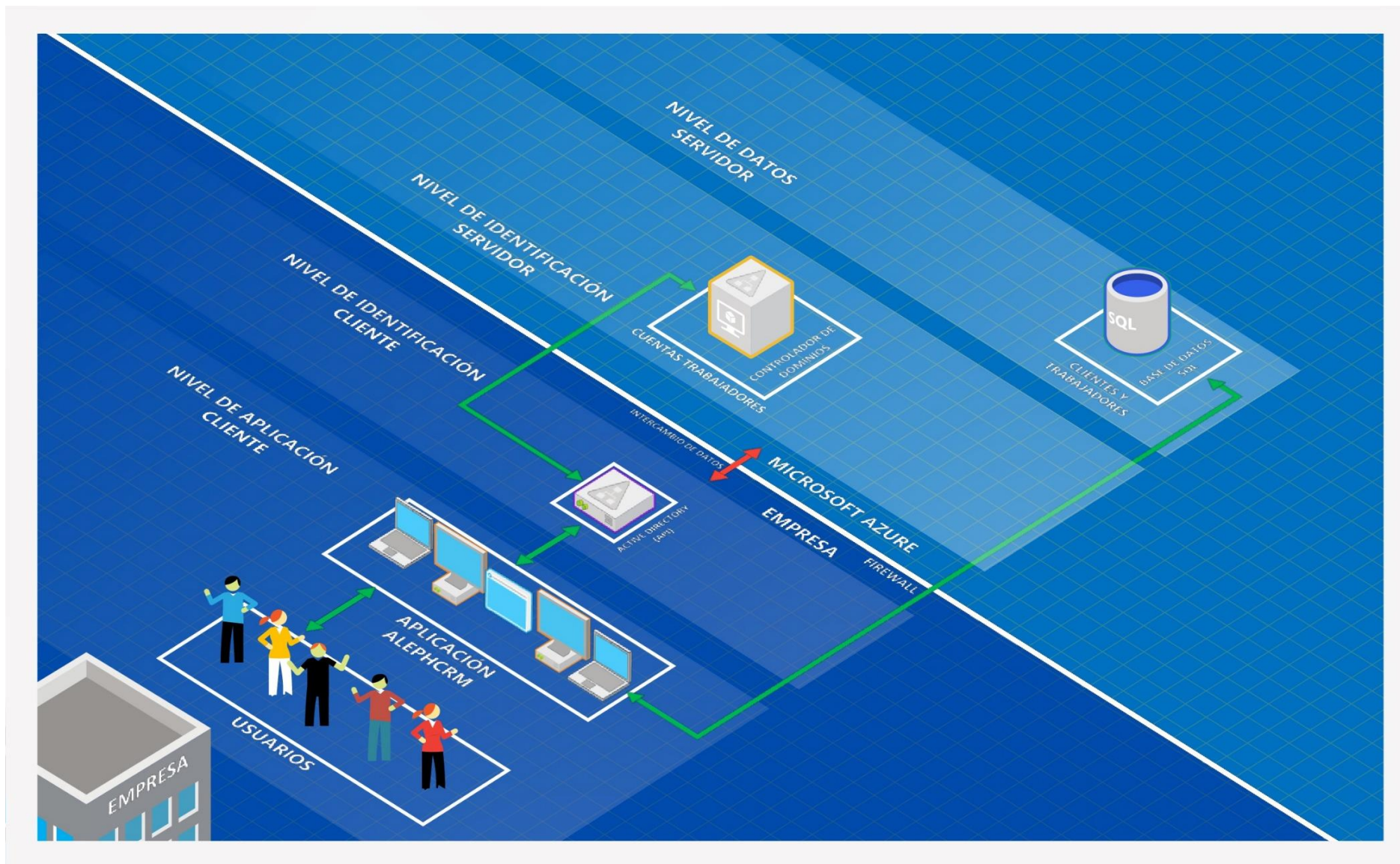


Figura 10.1: Diagrama de despliegue

# 11 Diseño procedimental

---

## 11.1 Introducción

En este capítulo se procederá a describir el conjunto de procesos que realiza cada una de las funciones del sistema. Para realizar esta tarea se hará uso de los diagramas de actividad que ofrece la notación UML.

Los diagramas de actividad se centran fundamentalmente en mostrar el flujo de control entre diferentes actividades. Se suelen utilizar para modelar los aspectos dinámicos de un sistema. La mayoría de las veces esto implica modelar pasos secuenciales (y posiblemente también concurrentes) de un proceso computacional. Cada diagrama de actividad consta de una serie de operaciones elementales, así como de un punto de inicio, un punto de parada y el número de bifurcaciones que sean necesarias.

## 11.2 Diagramas de actividad

A continuación, se mostrarán los diagramas de actividad correspondientes al sistema. La organización de estos diagramas de actividad se realizará de manera descendente, de tal modo que se mostrará primero el diagrama correspondiente a la actividad principal, que se irá desglosando paulatinamente hasta llegar a cada función.

### 11.2.1 Funcionalidad principal del sistema

La siguiente figura muestra el diagrama de actividad principal del sistema. En él que podemos observar las diferentes opciones disponibles cuando el usuario inicia la aplicación. Al principio deberá autenticarse para que se cargue la interfaz y el resto de las opciones del sistema. El nodo final indica que se cierra la aplicación.

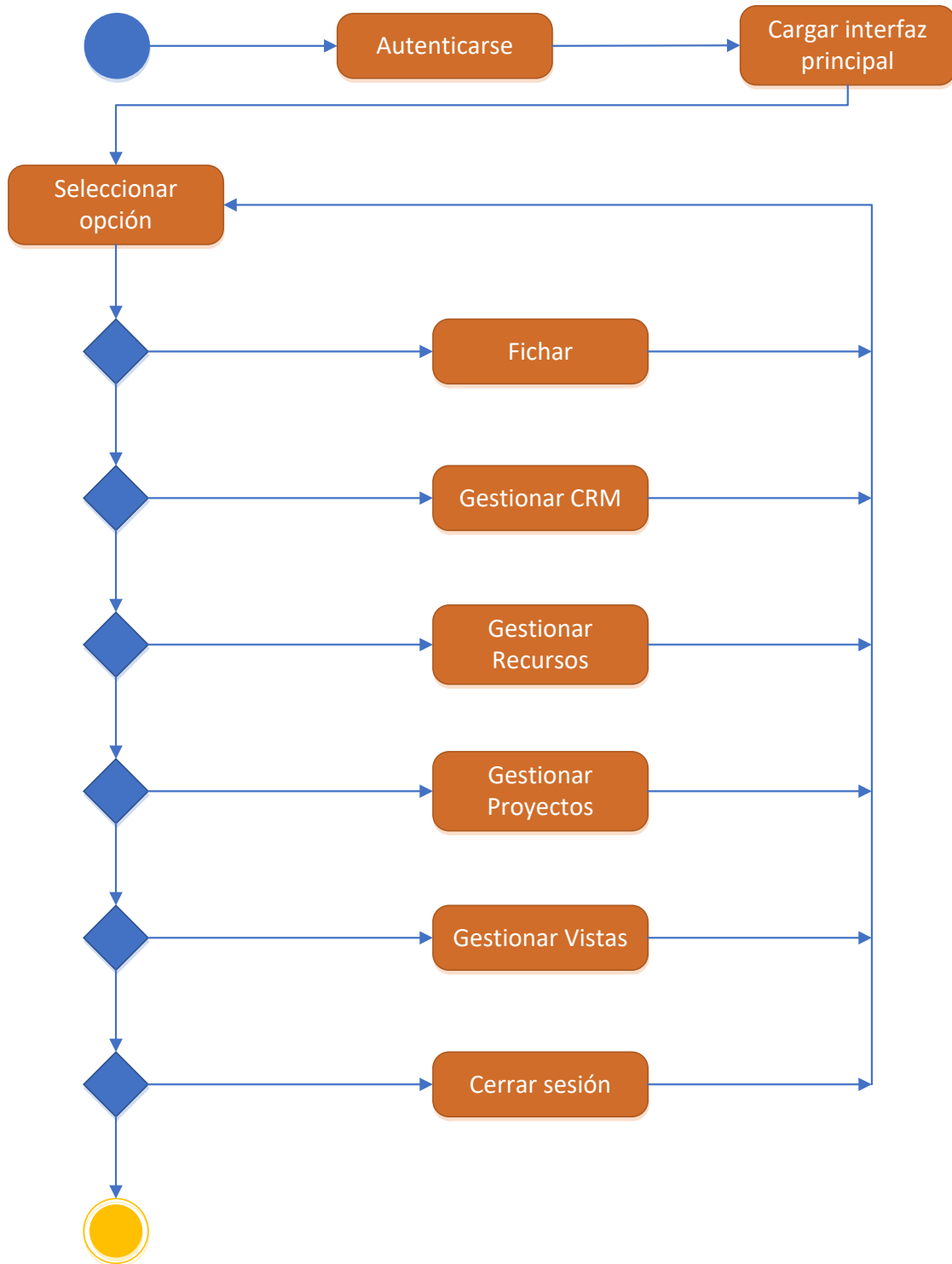


Figura 11.1: Diagrama de actividad - Funcionalidad principal



### 11.2.2 Autenticarse

En esta figura se muestran los pasos que ha de realizar el usuario para llegar a identificarse y así seguir a la siguiente opción en el diagrama de la funcionalidad principal, Cargar interfaz. En el caso en el que haya alguna cookie almacenada, se ingresará de forma automática con la última cuenta con la que se entró al sistema. Si la última vez se cerró sesión, la cuenta quedará almacenada en Windows. Si es la primera vez que se inicia sesión en la aplicación, se han borrado todas las cuentas del Administrador de credenciales de Windows o quiere introducir una cuenta nueva, se le pedirá al usuario introducir las credenciales completas.

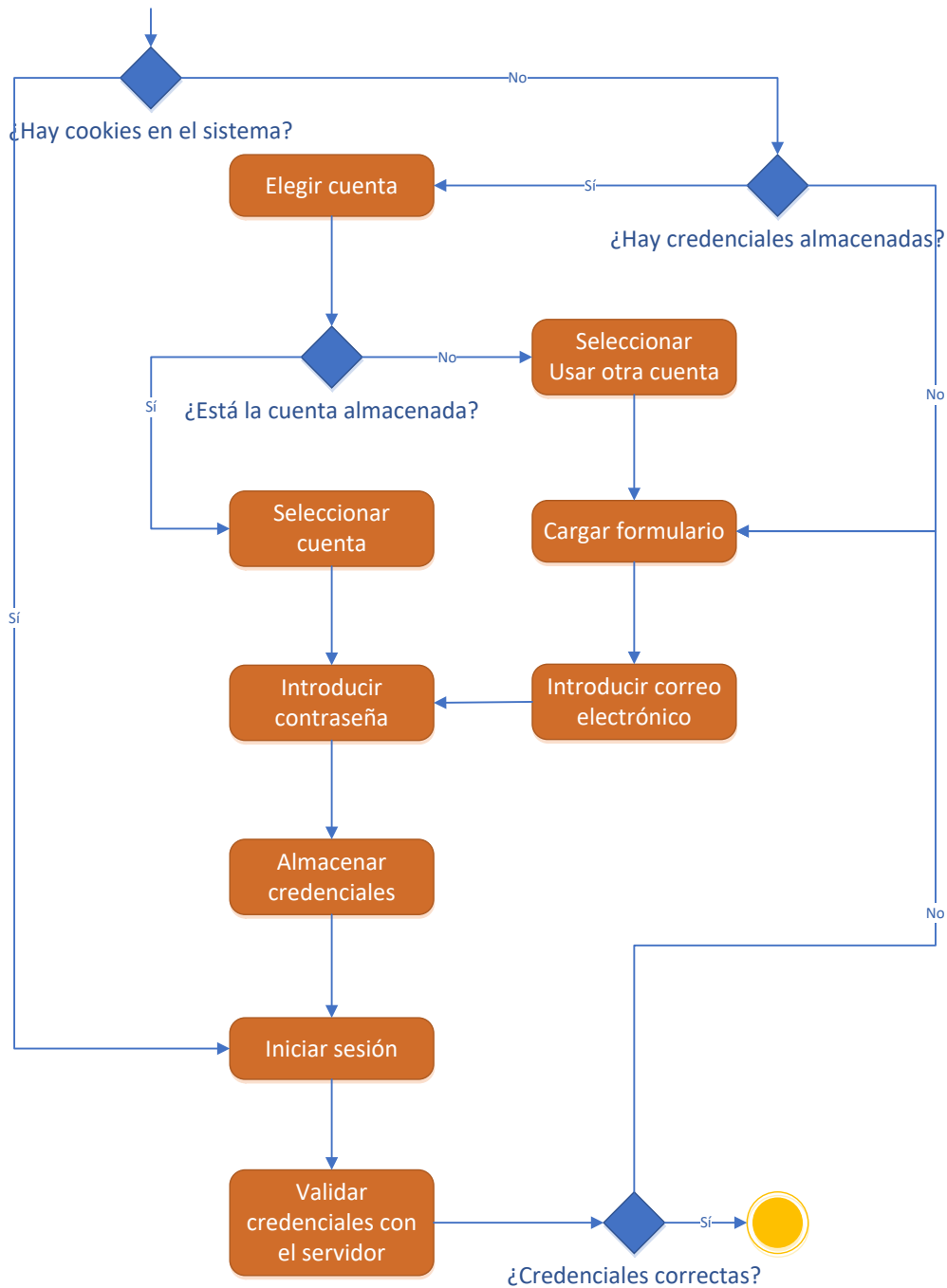


Figura 11.2: Diagrama de actividad - Autenticarse

### 11.2.3 Fichar

Esta opción estará disponible nada más iniciar el sistema, ya que es lo primero que debe hacer un usuario cuando abre la aplicación. Si a la hora de fichar hay algún problema, se notificará un mensaje de error y el usuario deberá ponerse en contacto con un superior o con servicio técnico.

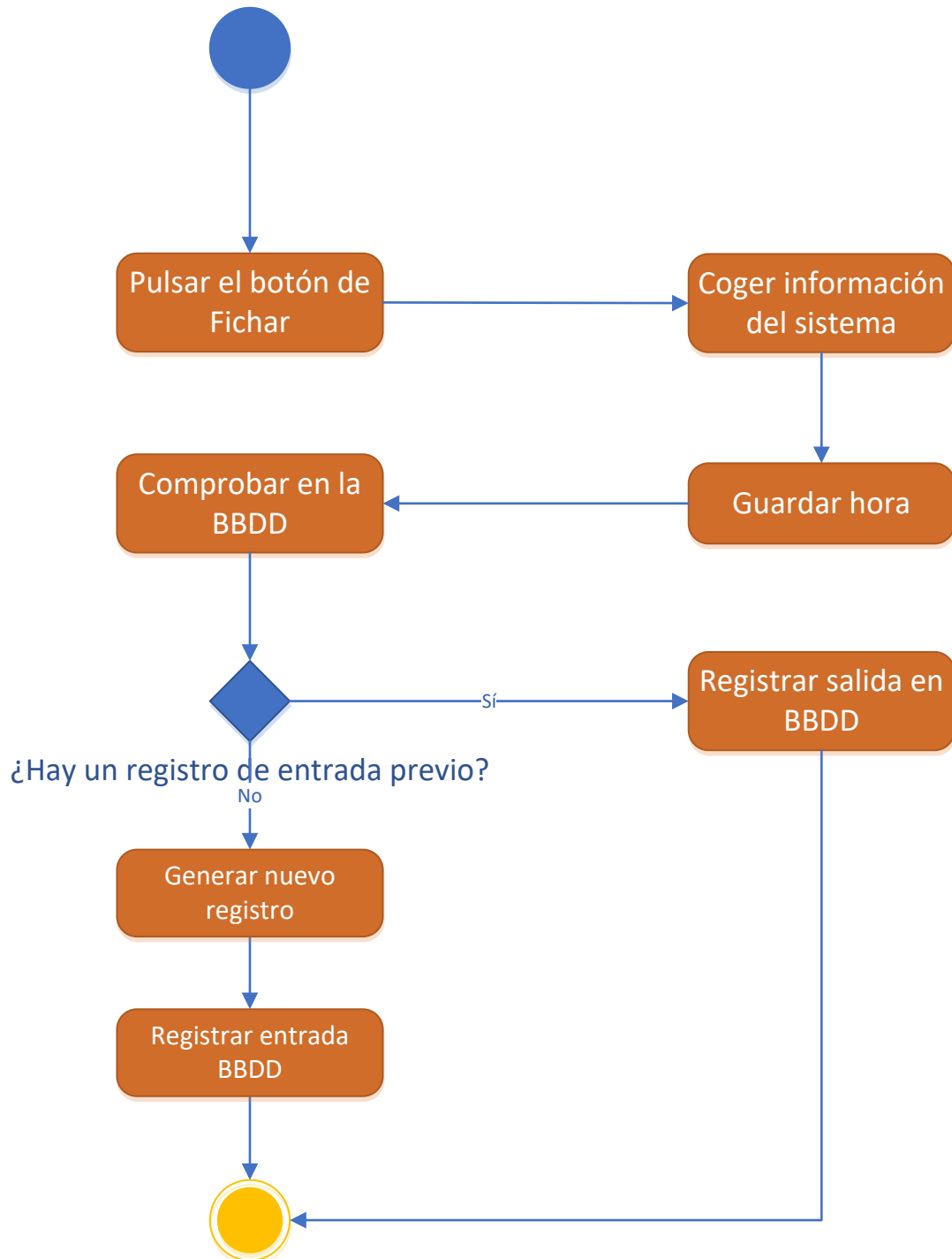


Figura 11.3: Diagrama de actividad - Fichar

## 11.2.4 Gestionar CRM

En la gestión CRM el usuario podrá consultar la lista de los clientes de la empresa y registrar una interacción que haya tenido con ellos. Si el usuario es administrador, podrá además añadir un nuevo cliente.

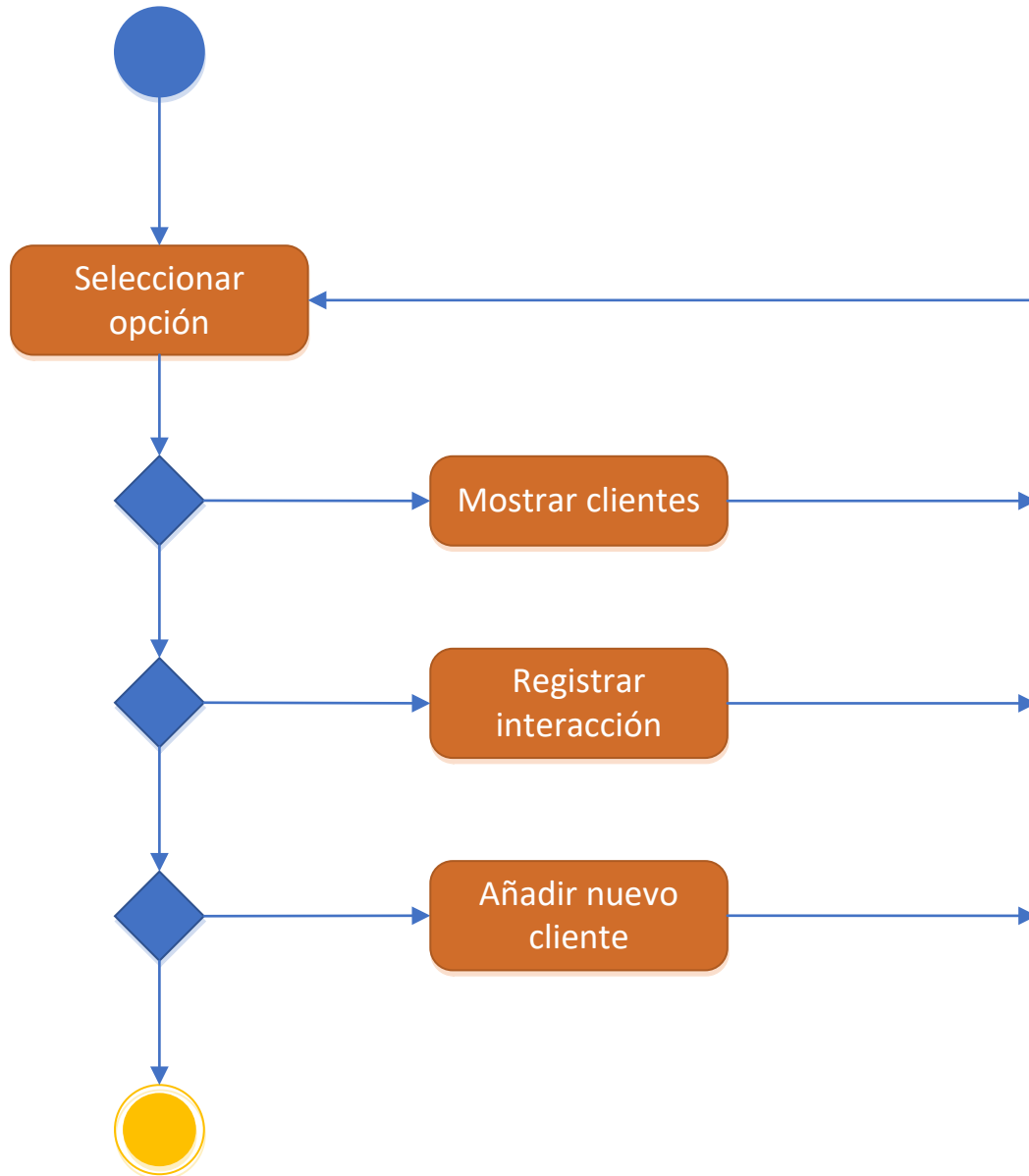


Figura 11.4: Diagrama de actividad - Gestionar CRM

### 11.2.4.1 *Mostrar clientes*

En esta actividad, el usuario puede consultar la ficha de un cliente determinado de la lista, filtrar la lista o buscar uno en particular. Si genera un informe, en éste aparecerá la tabla tal y como la está viendo en ese momento. Si es administrador, al consultar la ficha de un cliente, podrá modificarla o eliminarla.

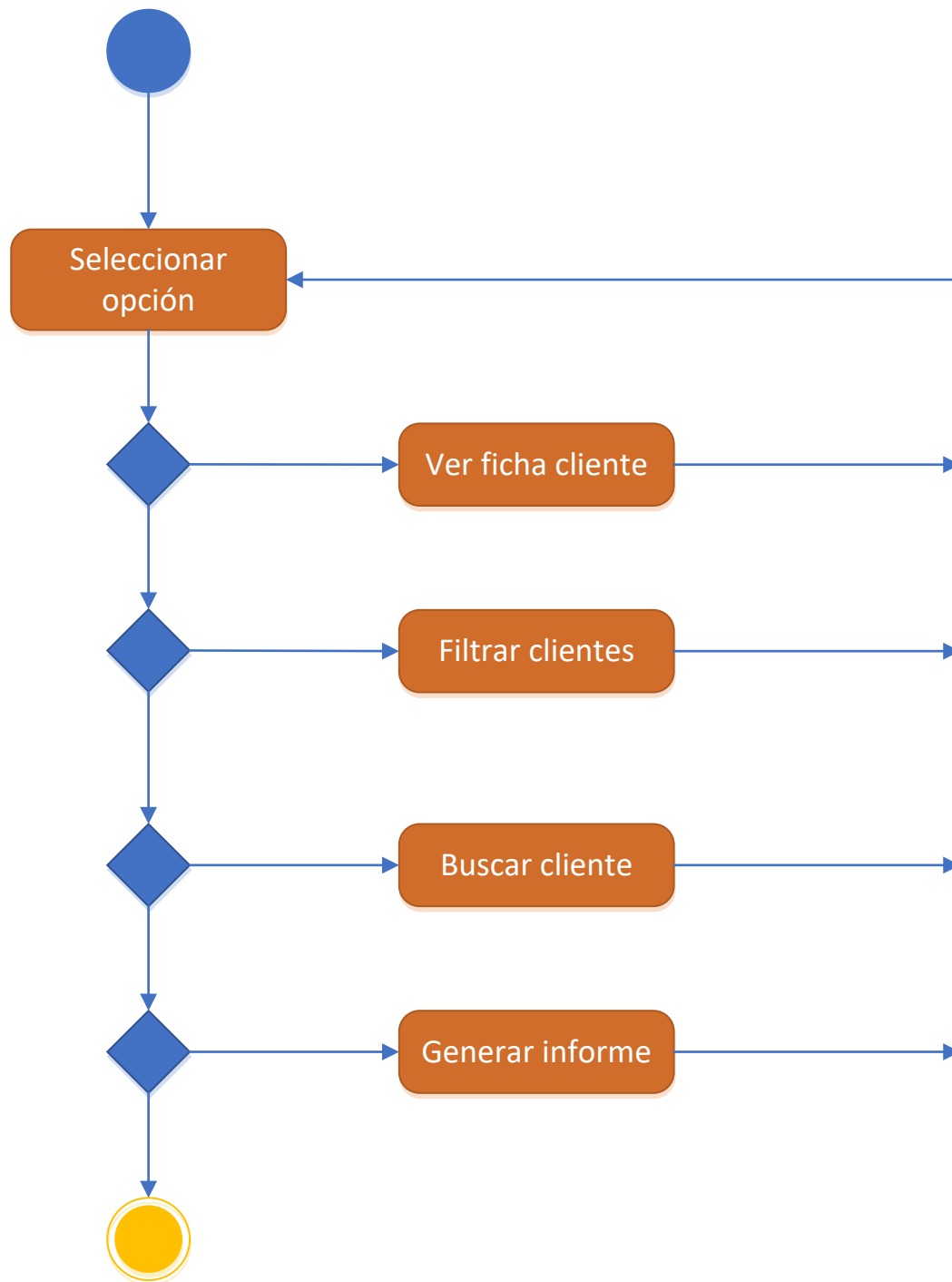


Figura 11.5: Diagrama actividad - Mostrar clientes

### 11.2.5 Gestionar Recursos

En la gestión de Recursos el usuario tendrá la posibilidad de ver su propio historial de fichajes y ver las tareas que tiene asignadas. Si el usuario es administrador, además podrá ver el historial de fichajes del resto de trabajadores, así como una lista completa de todos ellos.

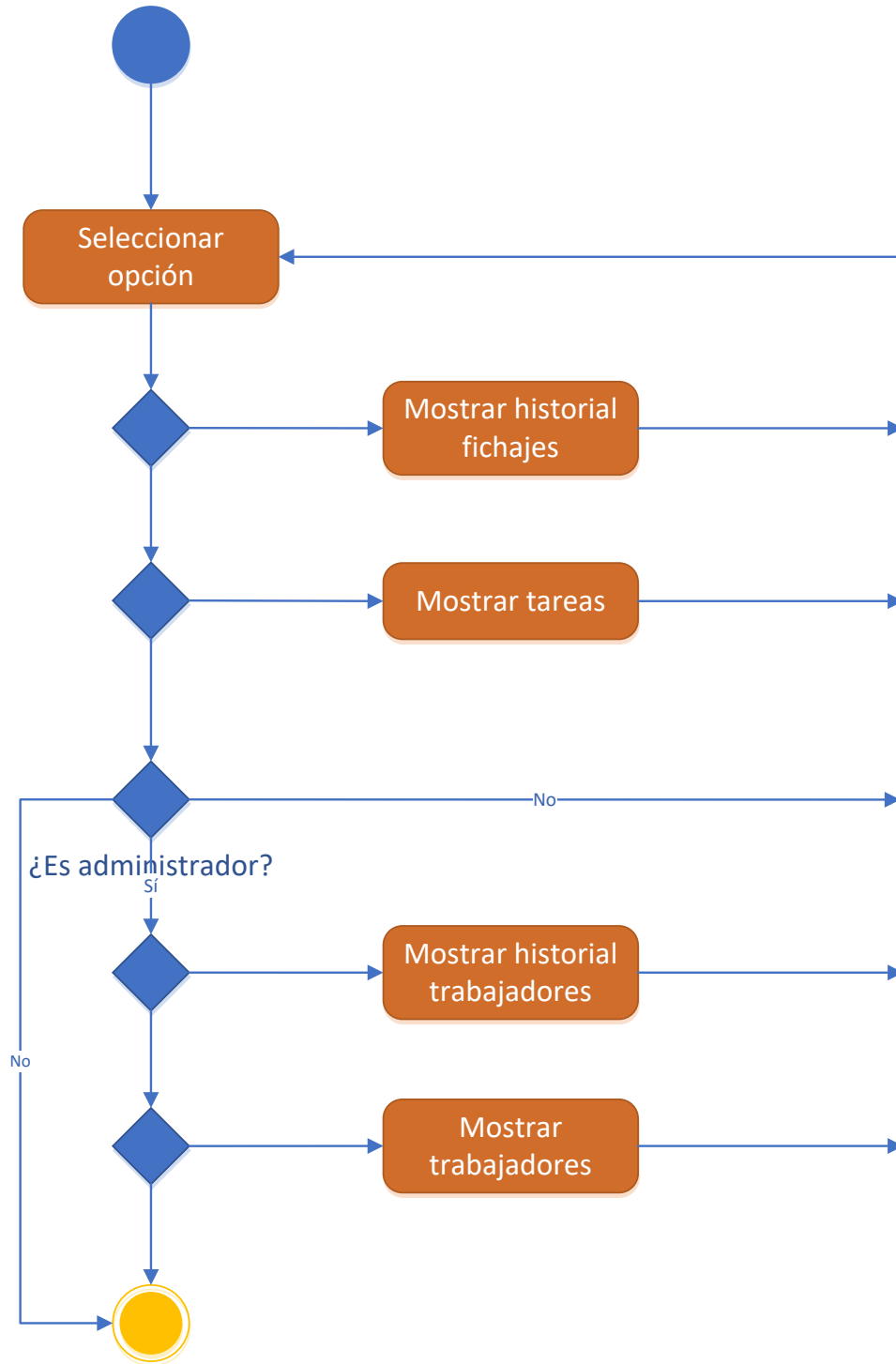


Figura 11.6: Diagrama de actividad - Gestionar Recursos

### 11.2.5.1 *Mostrar trabajadores*

En esta actividad, el usuario puede consultar la ficha de un trabajador determinado de la lista, filtrar la lista o buscar uno en particular. Si genera un informe, en éste aparecerá la tabla tal y como la está viendo en ese momento. Al consultar la ficha de un trabajador, podrá modificarla o eliminarla, ya que a esta actividad sólo pueden acceder los administradores.

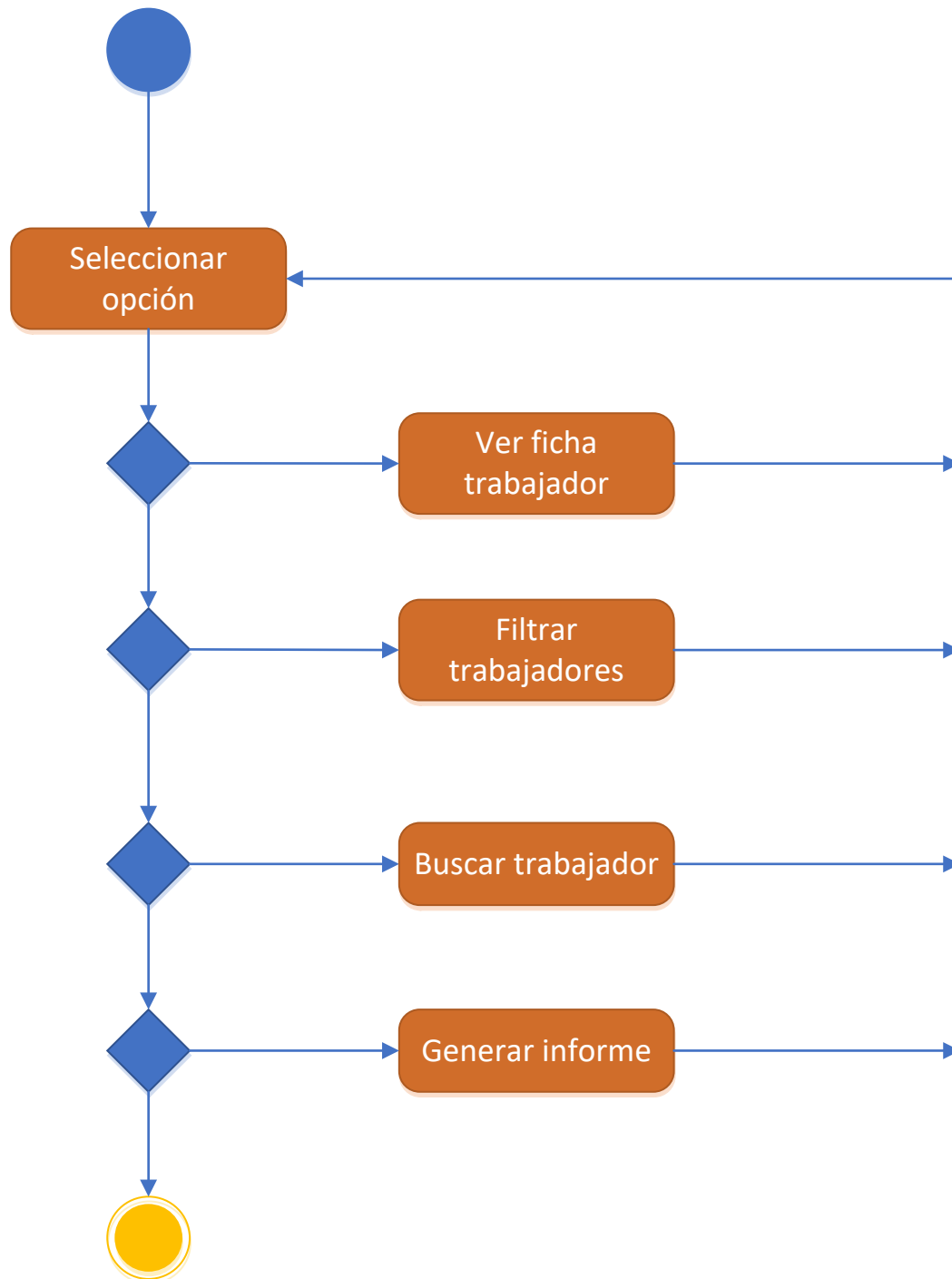


Figura 11.7: Diagrama actividad - Mostrar trabajadores

## 11.2.6 Gestionar Proyectos

En esta sección el usuario podrá consultar los proyectos en los que está participando (proyectos propios), en los que verá su estado actual, y ver su calendario, en el que se encontrarán las tareas y fechas límite que tendrá. Si el usuario es administrador, podrá además crear un proyecto y consultar todos los proyectos que están activos actualmente en la empresa, aunque no sea partícipe.

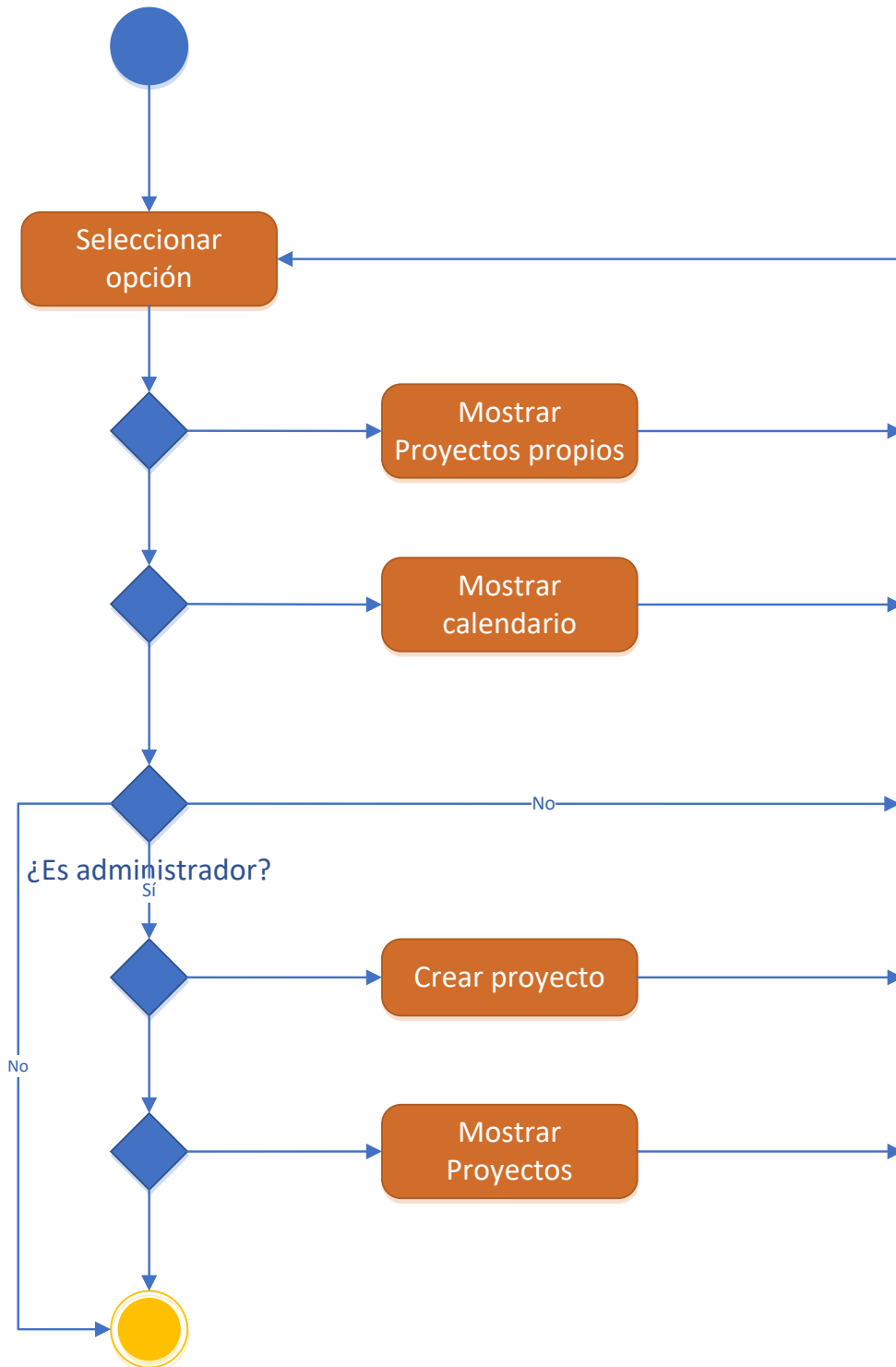


Figura 11.8: Diagrama de actividad - Gestionar Proyectos

### 11.2.6.1 *Mostrar Proyectos*

En esta actividad, el usuario puede consultar la ficha de un proyecto determinado de la lista, filtrar la lista o buscar uno en particular. Si genera un informe, en éste aparecerá la tabla tal y como la está viendo en ese momento. Al consultar la ficha de un proyecto, podrá modificarla o eliminarla, ya que a esta actividad sólo pueden acceder los administradores.

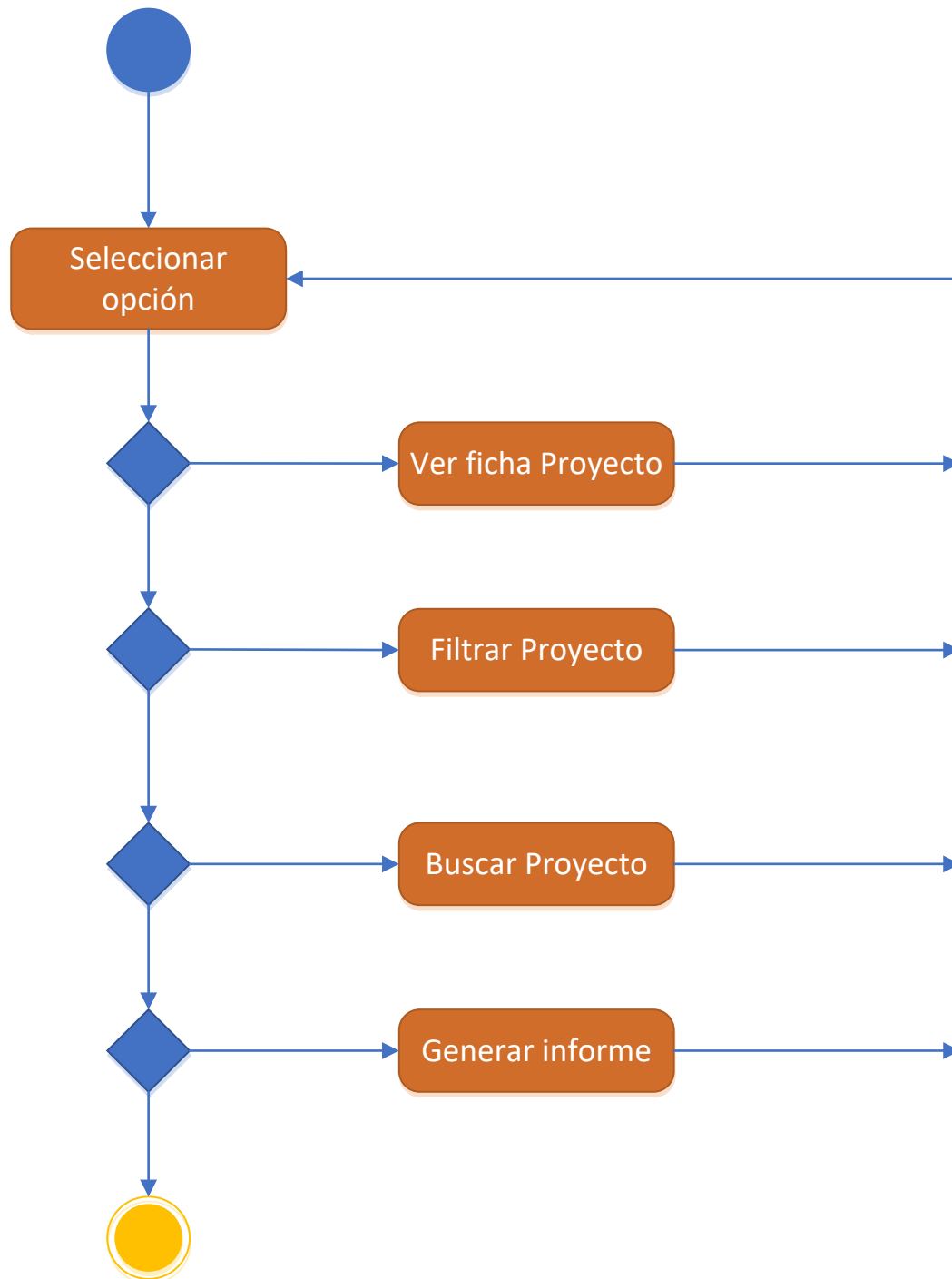


Figura 11.9: Diagrama de actividad - Mostrar Proyectos



### 11.2.7 Gestionar Vistas

En las vistas, el usuario podrá modificar la forma en la que se ven las tablas de la aplicación, así como los colores de la aplicación por completo seleccionando un tema en el selector de temas.

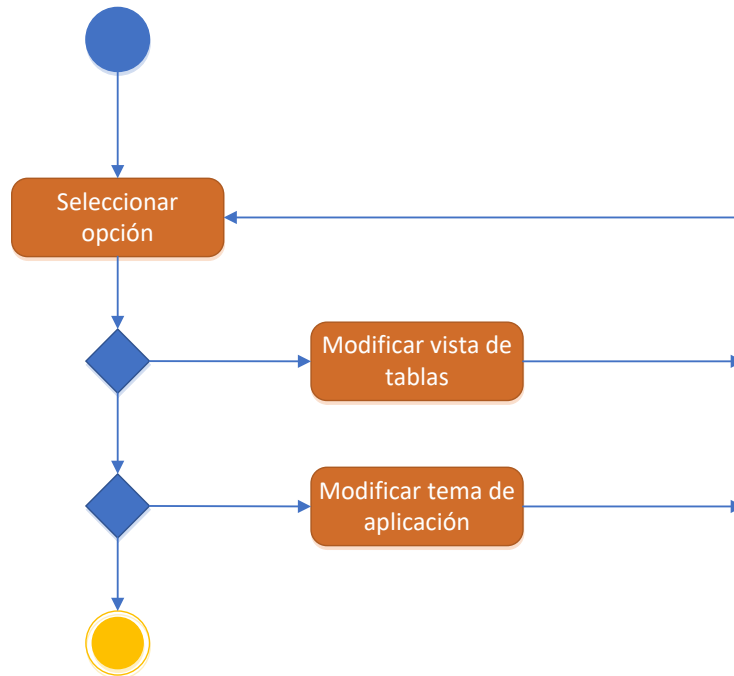


Figura 11.10: Diagrama de actividad - Gestionar vistas

### 11.2.8 Cerrar sesión

En este caso el usuario no tiene más que pulsar el botón de cerrar sesión, que estará disponible en todo momento. Se borrarán las cookies de inicio de sesión automático y se cerrará la aplicación.

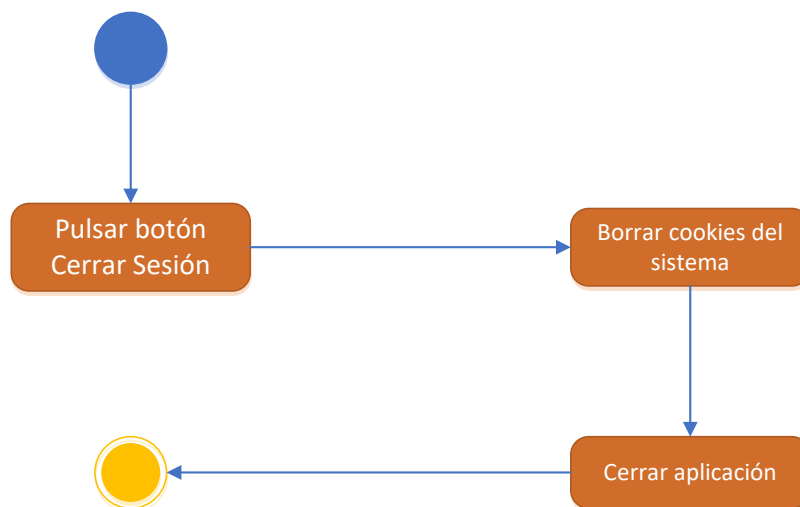


Figura 11.11: Diagrama de actividad - Cerrar sesión

### 11.2.9 Generar nuevo elemento

En este diagrama se mostrarán las actividades que se deben realizar para generar un nuevo elemento en la aplicación, contemplando un nuevo proyecto, con sus respectivas tareas, y un nuevo cliente. Estas actividades sólo podrán ser realizadas por un administrador. Se ha decidido realizar un diagrama general porque los procedimientos son prácticamente idénticos, variando únicamente entre ellos los formularios que se han de rellenar.

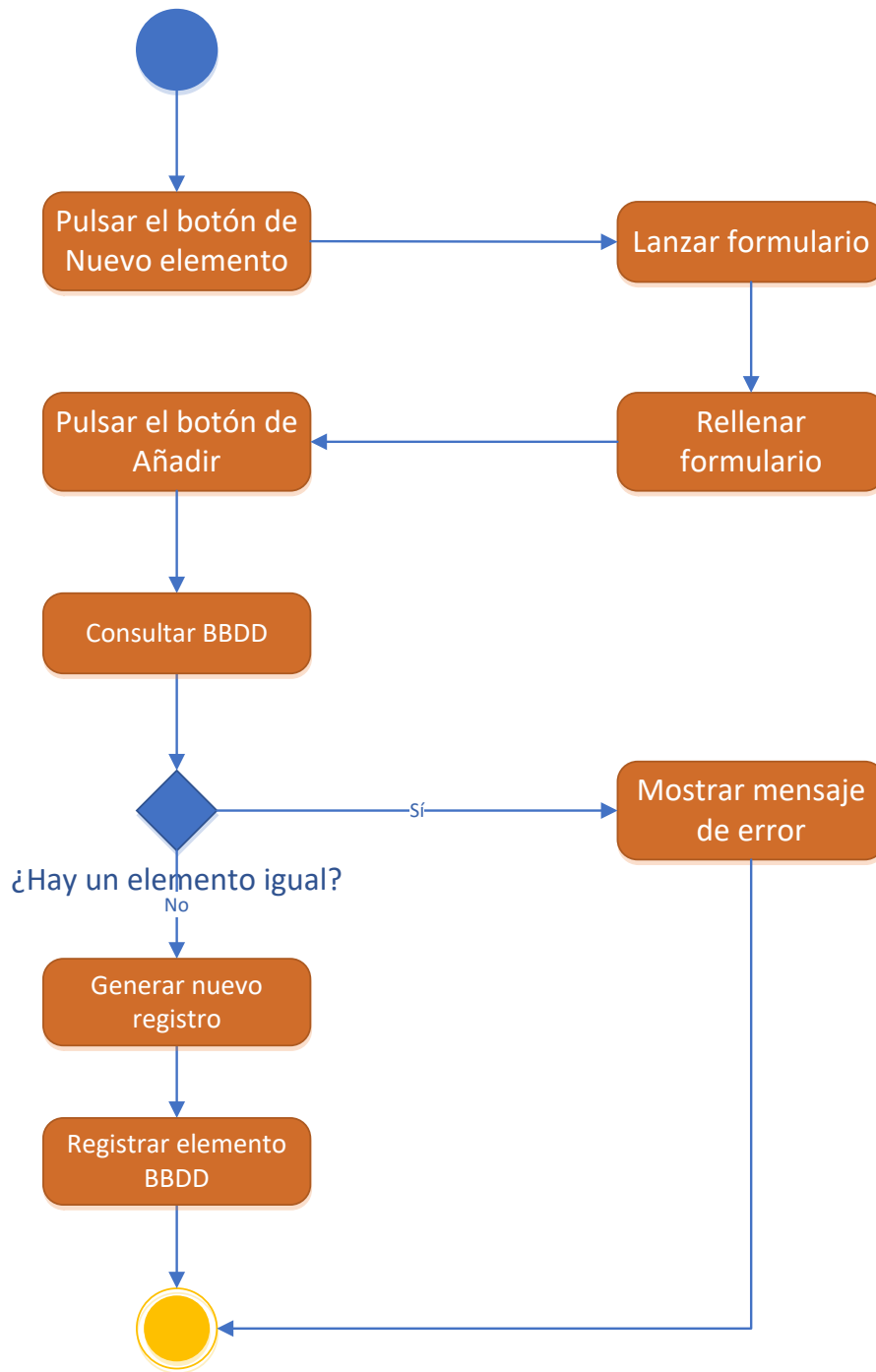


Figura 11.12: Diagrama actividad - Generar Nuevo elemento

## 12 Diseño de la interfaz

---

### 12.1 Introducción

En este capítulo se mostrará el diseño realizado para la interfaz de la aplicación a desarrollar. Con el objetivo de desarrollar una interfaz sencilla que posibilite la localización de la información deseada, el entendimiento claro de las funcionalidades ofrecidas y la navegación intuitiva por la aplicación, se ha buscado un diseño similar a las aplicaciones del paquete Office de Microsoft.

A continuación, se mostrarán y explicarán los diferentes elementos de la interfaz. La modificación de la información de los sujetos y la edición de informes se harán a través de ventanas emergentes, que también se mostrarán más adelante. Toda la aplicación tendrá una apariencia muy similar, por lo que se explicará de forma general los elementos comunes a todos los módulos. No se profundizará en el diseño de las funciones específicas de los módulos porque estos trabajan todos de forma similar y no modifican el aspecto general de la aplicación.

El diseño de la interfaz que se va a mostrar son meros prototipos, es decir, son ideas de diseño y distribución de los elementos que pueden variar en la versión final del sistema. Estos prototipos se han realizado con la herramienta de prototipado Pencil, que es de código abierto y está desarrollada por Evolus. Pencil está disponible para todas las plataformas y proporciona elementos para realizar prototipos de interfaces, como si se tratase de un dibujo en papel. También contiene plantillas para sistemas específicos, como los sistemas operativos móviles iOS y Android.

### 12.2 Ventana Principal

Como se puede comprobar en la figura 12.1, el diseño se compone de tres zonas bien diferenciadas:

- Una **barra superior** donde se encuentran las diferentes páginas, en las que en cada una de ellas están las funcionalidades de los módulos de la aplicación. Cada una de las pestañas contemplará una agrupación por categorías de las funcionalidades de la aplicación.
- Una **zona de trabajo** que ocupa la mayor parte de la interfaz, en la que se presentarán las tablas de información referentes a los sujetos de las bases de datos. También contendrá la caja de búsqueda y un sistema de pestañas para pasar de una tabla a otra.
- Una **barra de estado** en la que se mostrará el usuario que está conectado, su e-mail y su rol en el sistema.

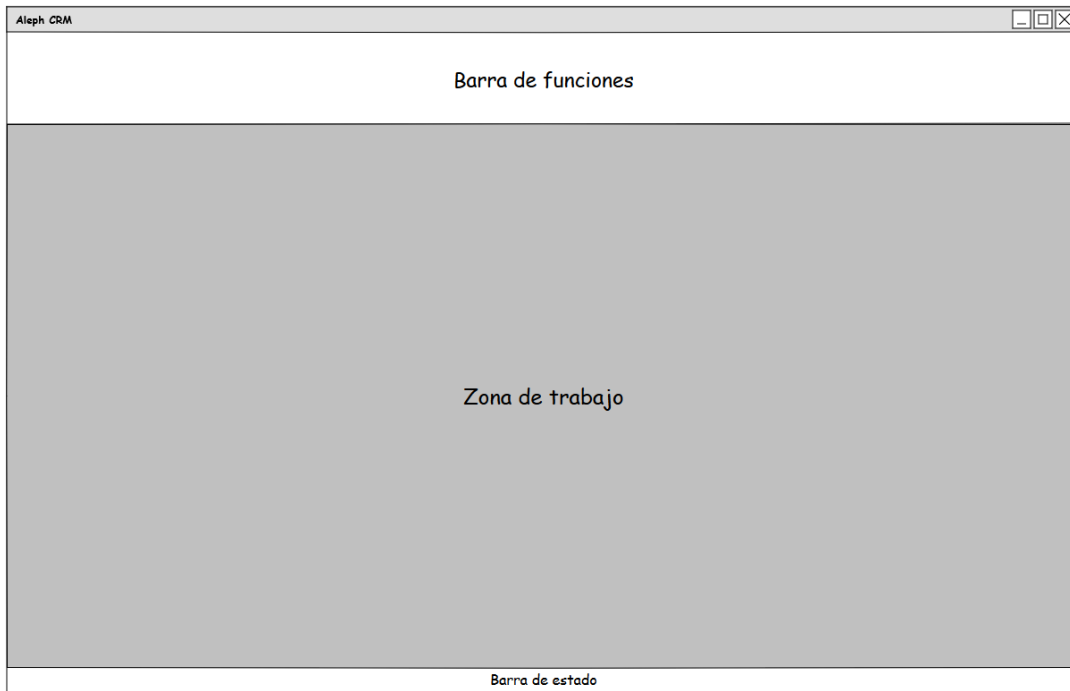


Figura 12.1: Ventana Principal - Vista base

En la zona de trabajo estarán las tablas, separadas por pestañas, que irían agrupándose conforme se van abriendo haciendo uso de los botones de la barra. En las páginas de la barra los botones se agruparán según categorías más concretas por funcionalidades relacionadas, para una visión más intuitiva de éstos. Justo encima de la tabla se encontrará la barra de búsqueda para filtrar los resultados de las tablas, y en la parte inferior de la ventana estará el nombre de usuario, seguido del e-mail y del rol, del usuario que está conectado en ese momento en la aplicación.

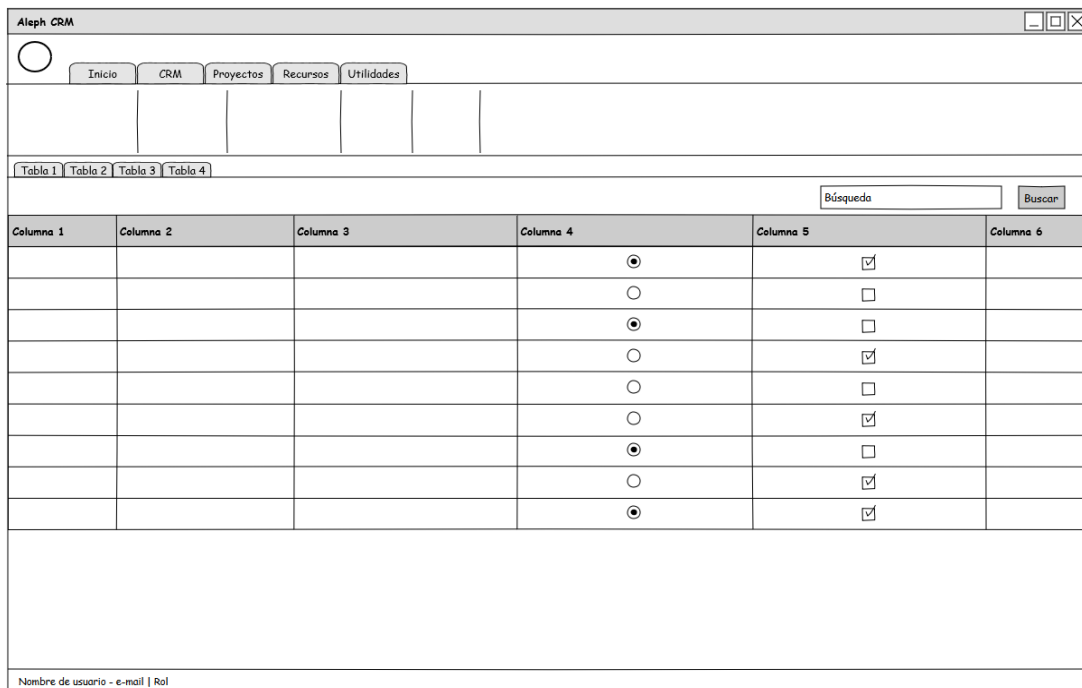


Figura 12.2: Ventana principal – Vista completa

## 12.3 Ventanas secundarias

Las ventanas secundarias son unas ventanas auxiliares que aparecerán en momentos determinados del uso de la aplicación. Estas ventanas se usarán para realizar ciertas funcionalidades del programa de forma que no intervengan en el trabajo actual que esté realizando el usuario. De esta forma no se interrumpirá en ningún momento la labor del usuario al no haber tediosos sistemas de pestañas múltiples o áreas de trabajo que se deben cerrar para continuar trabajando. Estas ventanas auxiliares se pueden dividir en tres categorías bastante diferenciadas, que son los editores, los mensajes y la información.

### 12.3.1 Editores

Los editores contendrán los formularios necesarios para consultar más detenidamente los datos de un sujeto de la aplicación. En ellos se podrán modificar los datos que se muestren. Aquí están incluidos los editores referentes a ingresar un nuevo sujeto y los de consultar/modificar la información de un sujeto.

Estos formularios variarán entre sí, pero serán muy similares en apariencia y funcionamiento. De la misma forma, estos dos tipos de editores tendrán campos diferentes a rellenar o modificar dependiendo de si se trata de la información de un cliente o de la información de un trabajador.

The screenshot shows a window titled "AlephCRM - Editor" with standard Windows window controls (minimize, maximize, close). The form contains the following fields and controls:

- Nombre:** A text input field.
- N.I.F.:** A text input field.
- Apellidos:** A text input field.
- Razón Social:** A text input field.
- Activo:** A checkbox that is checked.
- Modificable:** A checkbox that is checked.
- Usuario Alta:** A date input field with the format DD/MM/AAAA.
- Fecha Alta:** A date input field with the format DD/MM/AAAA.
- Sujeto:** A button above a large rectangular area containing a placeholder for a subject image, represented by a square with an 'X' and the text "108 x 108".
- Observaciones:** A large text area for notes.
- Guardar:** A button at the bottom right of the form.

Figura 12.3: Editores

## 12.3.2 Mensajes

Estas ventanas secundarias se usarán para mostrar advertencias o errores durante un uso normal de la aplicación. La apariencia de estas ventanas será siempre la misma, variando sólo el contenido del mensaje, que dependerá del error que se haya producido. El contenido del mensaje lo genera el propio código de la aplicación de forma automática cuando se produce una excepción o similares.

Junto al contenido del error se pondrán unas instrucciones para acceder a la ventana de información, indicando que se ponga en contacto con el servicio técnico para poder solventar dicho error.

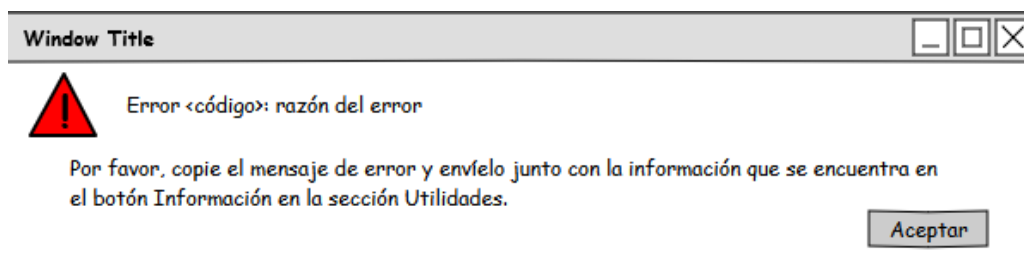


Figura 12.4: Mensaje de error

## 12.3.3 Información

Es una ventana que estará disponible en la pestaña Utilidades de la aplicación. En ella se mostrarán toda la información referente a la aplicación y la empresa Tecnube, incluyendo formas de contacto. También aparecerá información sobre el sistema que está ejecutando la aplicación (nombre del sistema, dirección IP, dirección MAC, sistema operativo, nombre de usuario, etc). Esta información, como se indicará en los mensajes de error y aquí mismo, se deberá adjuntar junto con el mensaje de error dado para así tener más información a la hora de solucionarlo.

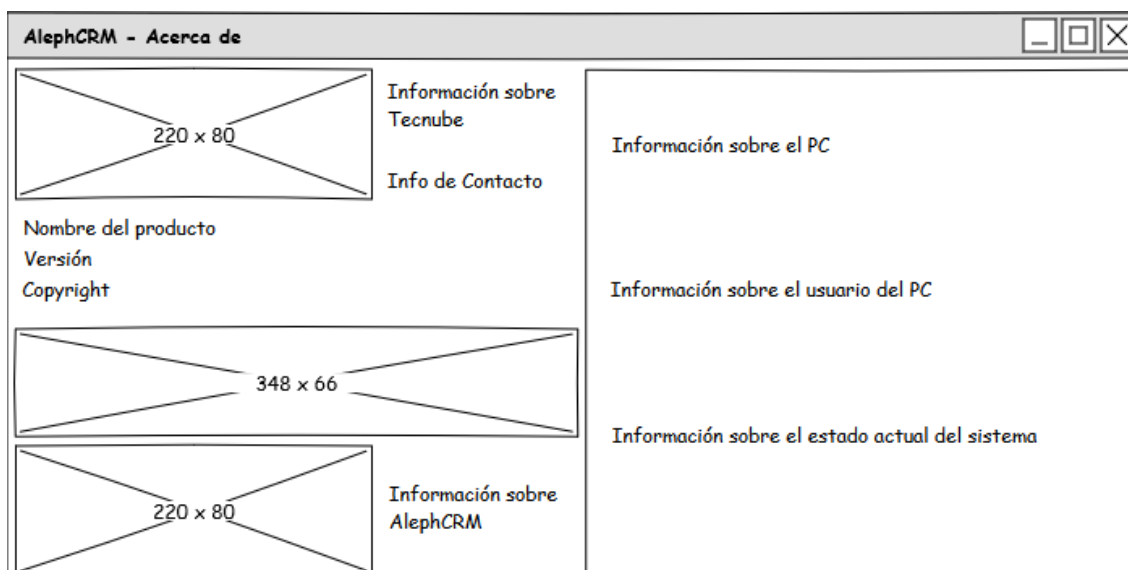
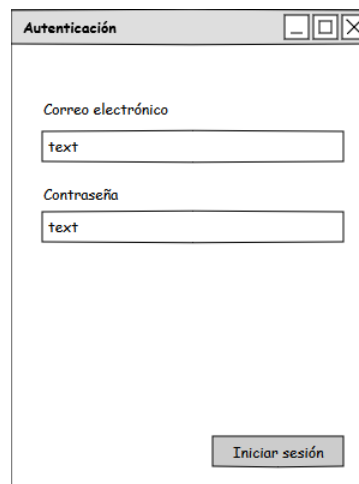


Figura 12.5: Acerca de - Información

## 12.4 Ventana de Inicio de sesión

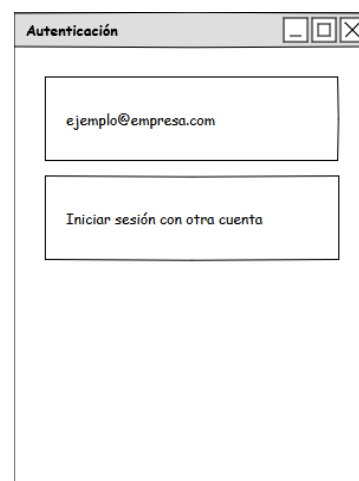
Esta ventana constará de dos cajas de texto, una para el e-mail del usuario y otra para su contraseña, que no la mostrará en ningún momento. Se iniciará sesión pulsando en el botón “Iniciar sesión” y, si el login es correcto, se lanzará la ventana principal de la aplicación. Si no es correcto el sistema se lo notificará y le dará la oportunidad de volver a introducir las credenciales.

El sistema almacena una cookie en el dispositivo que se esté utilizando, de forma que la próxima vez se iniciará sesión con las últimas credenciales que se han introducido. Si esta cookie expira, cuando el usuario inicie la aplicación le dará la oportunidad de acceder a ésta con una de las cuentas que ya ha introducido anteriormente, introduciendo únicamente la contraseña de nuevo. Si no desea autenticarse con una de esas cuentas, le dará la opción de introducir una nueva, como se puede apreciar en la figura 12.7.



The screenshot shows a window titled "Autenticación" with standard Windows window controls. It contains two text input fields. The first is labeled "Correo electrónico" and contains the text "text". The second is labeled "Contraseña" and also contains the text "text". Below the fields is a button labeled "Iniciar sesión".

Figura 12.6: Autenticación



The screenshot shows the same "Autenticación" window. The email input field now contains "ejemplo@empresa.com". Below it is a button labeled "Iniciar sesión con otra cuenta".

Figura 12.7: Autenticación – Cambio de cuenta





# SECCIÓN IV

## PRUEBAS



## 13 Pruebas

---

### 13.1 Introducción

En este capítulo se detallarán las pruebas realizadas al software desarrollado. El objetivo que se pretende conseguir mediante la realización de este conjunto de pruebas es valorar la calidad del software, y además nos permitirá detectar y corregir los errores que se produzcan.

Esta fase es muy importante en el desarrollo de un sistema software, puesto que de ella depende la garantía de calidad del producto. Las pruebas se han realizado desde un enfoque estructural (pruebas de la caja blanca) y desde un enfoque funcional (pruebas de la caja negra).

El enfoque estructural o prueba de la caja blanca consiste en probar la estructura interna (implementación) para elegir los casos de prueba. La prueba ideal consistiría en probar todos los caminos de ejecución a través de las instrucciones de código.

Por otro lado, el enfoque funcional o prueba de la caja negra se centra en el estudio de la especificación de cada una de las funciones, sus entradas y sus salidas, para obtener así los diferentes casos de prueba. La prueba ideal en este caso sería probar para cada función todas las posibles entradas y salidas.

Las pruebas se han realizado a lo largo de la etapa de codificación del sistema y tras su finalización, llevadas a cabo por el autor del sistema y por personal ajeno al proceso de desarrollo. A continuación se describirán más detalladamente.

### 13.2 Enfoque estructural

Las pruebas de la caja blanca han sido realizadas por el autor de forma paralela a la etapa de codificación, es decir, conforme se han implementado las funciones o procedimientos se han hecho las pruebas necesarias para determinar su validez. En cada uno de los casos se ha comprobado, para todos los caminos de ejecución, la no existencia de bucles infinitos, el buen funcionamiento de las sentencias condicionales, valores de variables y parámetros, variables sin usar, etc.

Mediante la ejecución experimental de los distintos caminos que podía tomar la información dentro del código, haciendo uso de todos los posibles rangos de valores que pueden tomar las variables que utiliza la aplicación, se detectaron diversos errores en el funcionamiento de la aplicación. Estos errores han sido corregidos, evaluando nuevamente las partes de código comprometidas por los mismos y aquellas relacionadas.

Una vez terminadas las pruebas de caja blanca o enfoque estructural, se puede garantizar que se han recorrido al menos una vez todos los posibles caminos independientes que se forman mediante las instrucciones del código de la aplicación, además de corregir todos los errores que fueron apareciendo durante la realización de este tipo de pruebas.

## 13.3 Enfoque funcional

Las pruebas de la caja negra se desarrollaron posteriormente a la etapa de codificación y han sido estructuradas en tres fases diferentes para intentar detectar diferentes tipos de errores. A continuación, se detallan cada una de las fases:

### 13.3.1 Primera Fase: Acceso al sistema

Las actividades de esta etapa se refieren a hacer chequeos completos respecto al acceso a la aplicación y a los datos que ésta maneja. En los siguientes apartados se muestran las pruebas realizadas.

#### 13.3.1.1 *Autenticación Azure Active Directory*

Para realizar esta prueba se han utilizado tres cuentas distintas, dos de ellas con acceso permitido en el Active Directory y otra configurada para que no tenga acceso. Una de las que tiene acceso está almacenada en la base de datos como un trabajador y la otra como un administrador.

Se introdujeron las credenciales de estas tres cuentas, comprobando que la cuenta que no tiene acceso se detectaba como tal y la API mostraba un mensaje pidiendo unas credenciales correctas o con acceso al directorio. Con las otras dos cuentas se introdujeron mal las credenciales para comprobar que lanzaba el mensaje de error. Cuando se introdujeron bien se comprobó que la aplicación detectaba correctamente el rol de ambos usuarios.

#### 13.3.1.2 *Acceso a la base de datos*

En los ajustes de la aplicación se introdujeron mal los datos de conexión a la base de datos para que lanzara el diálogo de configuración de la aplicación, en el que se pueden modificar estos datos para poder conectarse a la base de datos. Se comprobó que con las credenciales correctas la conexión a esta era correcta y se cargaba el resto de la aplicación. Esta comprobación siempre se realiza después de la autenticación.

### 13.3.2 Segunda Fase: Interfaz

Esta fase de pruebas se revisan aspectos gráficos de la aplicación, para determinar si el despliegue de las páginas, botones e iconos es el correcto. Se realizaron las siguientes pruebas:

#### 13.3.2.1 *Localización e integridad de elementos*

Aquí se comprobó que el orden de las pestañas es el correcto y que en cada una de ellas los botones estaban correctamente colocados. Cada uno de los botones debe tener a su vez el icono correspondiente que indique de forma gráfica lo que realiza. También se comprobó que en la barra de estado aparecía el nombre, correo y rol del usuario conectado, cambiando de cuenta para ver si estos datos se veían modificados. Una última comprobación fue para ver si la ventana se podía minimizar, maximizar y cambiar el tamaño correctamente.

### 13.3.2.2 *Tamaño de los elementos*

Con estas sencillas pruebas se verificaron que los tamaños de los elementos de la interfaz eran los óptimos, tales como el tamaño de los botones, de los iconos y de las fuentes.

Otra parte de estas pruebas han consistido en comprobar si al pulsar un botón de la interfaz que cargara algo en la zona de trabajo, el elemento que se cargaba ocupaba toda el área disponible, de forma que aprovechara al máximo esta zona. También se comprobó que la disposición de los elementos en las fichas de los sujetos y tareas era la deseada.

### 13.3.2.3 *Presentación por roles*

Hay ciertas operaciones que están capadas para el rol del trabajador, por tanto, éstas no se muestran en la interfaz cuando un trabajador está autenticado en el sistema y sí cuando está conectado un administrador. Esta prueba se ha realizado utilizando las dos cuentas con acceso autorizado en el directorio de Azure.

## 13.3.3 Tercera Fase: Operaciones

Las actividades de esta etapa se refieren a hacer chequeos completos respecto de las funcionalidades y aplicaciones que ofrece la aplicación. En los siguientes apartados se muestran las pruebas realizadas:

### 13.3.3.1 *Pruebas de datos*

En estas pruebas se comprobó si se accedía correctamente a la base de datos a la hora de leer información de ella, abriendo todos los listados y fichas de los sujetos de la aplicación, comprobando que la información correspondiente a todos ellos era la correcta.

Así mismo, se ha comprobado que la introducción de un nuevo fichaje para indicar el inicio de la jornada laboral era correcta y que, al darle de nuevo, indicaba que se iba a introducir una salida. Respecto a la creación de nuevos clientes, tareas e interacciones se hicieron pruebas para determinar si la inserción era correcta en la base de datos y si mantenía correctamente las relaciones que se indicaban en los formularios, accediendo a los listados y fichas de los nuevos elementos introducidos de estos nuevos elementos. También se ha probado la modificación de campos de los elementos ya almacenados y que estos cambios se guarden en la base de datos.

Se han realizado con las dos cuentas diferentes que tienen acceso al sistema para verificar que las acciones disponibles para ambos roles son las correctas.

### 13.3.3.2 *Pruebas de informes*

Se ha comprobado que cuando se desea crear un informe se lanza un diálogo para introducir la ruta para guardar el archivo que se va a generar. Se ha revisado que los informes se generaban a partir de la tabla activa en ese momento en la interfaz, y que ésta mantenía los filtros y la presentación que se habían realizado en ella.



# SECCIÓN V

# CONCLUSIONES





## 14 Conclusiones y futuras mejoras

---

### 14.1 Introducción

En este capítulo se mencionarán las conclusiones obtenidas tras la realización del proyecto. También se mencionará las futuras mejoras que se podrían introducir en el presente software.

### 14.2 Conclusiones

Tras haber llevado a cabo todas las etapas que componen el desarrollo del sistema software, se han extraído las siguientes conclusiones en base a los objetivos propuestos inicialmente:

- El diseño de la interfaz se ha mantenido sencillo y estructurado, de forma que cualquier usuario sin conocimientos previos pueda hacer uso del software desde el minuto uno.
- Se ha mantenido un diseño con el estilo del paquete Office de Microsoft, dando un ambiente familiar al usuario final.
- Se ha conseguido integrar las funcionalidades que se deseaban desde un principio, introduciendo los módulos totalmente independientes entre sí.
- La estructuración del código del software permite la integración de otros módulos de forma bastante sencilla, pudiendo en un futuro desarrollar e introducir nuevas funcionalidades en el sistema rápidamente sin tener que rediseñar la aplicación por completo.
- La aplicación permite la autenticación con cuentas empresariales, consiguiendo así una personalización aún mejor a la hora de venderla a un cliente.
- Es posible consultar la información almacenada en la base de datos de forma rápida, y modificar ésta de forma intuitiva y sencilla.
- Para introducir nuevos elementos como clientes, interacciones, proyectos o tareas se utilizan los mismos editores que para modificar la información de los ya existentes, por lo que mantiene una coherencia e integridad respecto a la presentación de elementos.
- El sistema desarrollado es totalmente independiente de la base de datos a la que se conecte, pues permite cambiar los apartados para la cadena de conexión, pudiendo conectarse a otra base de datos diferente y seguir funcionando. Las tablas, sin embargo, deberán seguir la misma estructura siempre.
- Es totalmente independiente del Active Directory al que se conecte, pues da la oportunidad de configurar los ajustes de conexión a Azure, de forma que se podría conectar a un directorio diferente y seguir funcionando perfectamente. El único requisito es que los correos que se utilizan en el directorio se encuentren ligados inequívocamente a su usuario correspondiente en la base de datos a la que se conecte.

- Se ha construido una aplicación de gran utilidad para una empresa pequeña, que permite gestionar de manera eficiente y sencilla la información que ésta posee de sus clientes, de sus trabajadores y de los proyectos que tiene en marcha.

Por último, cabe señalar que, en vista de los resultados obtenidos, se han cumplido plenamente los objetivos marcados al comienzo de este proyecto.

### 14.3 Futuras mejoras

En el mercado actual, las aplicaciones software de ámbito empresarial evolucionan continuamente, añadiendo nuevas funcionalidades y características, haciéndolas más atractivas para los clientes potenciales y aquellos que ya tienen contratado algún servicio. Para prosperar es necesario seguir esta corriente, además de que no todos los módulos que están en la aplicación original han sido implementados en esta, por lo que las futuras mejoras podrían ser:

- Introducir el resto de los módulos de la aplicación AlephCRM que está desarrollada en Windows Forms, y así tener esta aplicación al completo y con muchas más funcionalidades.
- Hacer la aplicación multiplataforma, de forma que se pueda acceder a ella desde cualquier dispositivo. En un principio se implementaría en UWP, para tener una aplicación de la tienda de Windows, y que así esté disponible para Windows 10 en escritorio y en dispositivos móviles. Esto se debe a que el ecosistema de Windows 10 está enfocado al ámbito empresarial, aunque más adelante se desarrollaría para iPhone y Android.
- Adaptar la interfaz actual a dispositivos móviles o con pantalla táctil que ejecuten Windows en su versión de escritorio, como podrían ser las tablets o los últimos ultrabooks convertibles, para que sea más cómoda de utilizar sin necesidad de un teclado o un ratón.
- Incluir la internacionalización, de forma que pueda estar accesible a usuarios de diferentes lenguas. De esta forma la aplicación se podría vender a empresas extranjeras o que tengan equipos compuestos por personas de diferentes países.
- Centrar la aplicación completamente en la nube, de forma que los ajustes personales de un usuario se mantengan entre los dispositivos que éste usa.

### 14.4 Conclusiones personales

Como conclusiones personales destacar la adquisición de conocimientos en diversas áreas que abarca este proyecto, entre las que se pueden destacar:

- Estructuración y desarrollo de aplicaciones profesionales y de ámbito empresarial
- Conocimientos de la gestión empresarial
- Conocimientos en la plataforma .NET y de las tecnologías Windows Forms y WPF.
- Lenguaje de programación C# y lenguaje de marcado XAML
- Uso del entorno de desarrollo Visual Studio 2015 para el desarrollo y depuración de aplicaciones, así como la conexión de estas a las bases de datos
- Conocimientos de Transac-SQL

## Bibliografía

---

### 14.5 Introducción

En esta sección se encuentran las referencias bibliográficas que se han utilizado en el desarrollo del proyecto. En esta ocasión sólo se han utilizado referencias web, que estarán ordenadas por orden de aparición en este informe. Además, se han introducido referencias de sitios web que se utilizaron para estudiar y aprender a utilizar las tecnologías y lenguajes necesarios para la realización del programa en sí. El formato elegido es el de IEEE.

### 14.6 Referencias bibliográficas – Manual técnico

- [1] «Wikipedia - Burbuja puntocom,» [En línea]. Available: [https://es.wikipedia.org/wiki/Burbuja\\_punto\\_com](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](http://www.zazachat.com/kb/crm/articles/crm_history.aspx).
- [4] «Evaluando CRM - Módulos de un CRM,» [En línea]. Available: <http://www.evaluandocrm.com/modulos-de-un-crm/>.
- [5] «Suma CRM - Módulos,» [En línea]. Available: <https://www.sumacrm.com/soporte/modulos-crm>.
- [6] «Bluenote - Módulos,» [En línea]. Available: <https://www.bluenote-systems.eu/crm-features/modules-functions-crm-software.html>.
- [7] «Suma CRM,» [En línea]. Available: <https://www.sumacrm.com/soporte/tipos-de-crm-y-modelos>.
- [8] «GestionEGRP,» [En línea]. Available: <http://gestionegrp103.weebly.com/tipos-de-crm.html>.
- [9] “Wikipedia - CRM,» [Online]. Available: [https://en.wikipedia.org/wiki/Customer\\_relationship\\_management](https://en.wikipedia.org/wiki/Customer_relationship_management). [Accessed 15 05 2016].
- [10] «WinForm - Wikipedia,» [En línea]. Available: [https://en.wikipedia.org/wiki/Windows\\_Forms](https://en.wikipedia.org/wiki/Windows_Forms).
- [11] «DevExpress - Web,» [En línea]. Available: <https://www.devexpress.com/>.
- [12] «Support Microsoft - Framework 4.5,» [En línea]. Available: <https://msdn.microsoft.com/es-es/library/8z6watww%28v=vs.110%29.aspx?f=255&MSPPError=-2147217396>.
- [13] «Chorus - Dynamics CRM,» [En línea]. Available: <https://www.chorus.co/filters/news/microsoft-dynamics-crm-a-history/>.

- [14] «Microsoft - Dynamics CRM,» [En línea]. Available: <https://www.microsoft.com/es-es/dynamics/crm.aspx>.
- [15] «Wikipedia - Windows Instaler,» [En línea]. Available: [https://en.wikipedia.org/wiki/Windows\\_Installer](https://en.wikipedia.org/wiki/Windows_Installer).
- [16] «Wikipedia - ClickOnce,» [En línea]. Available: <https://en.wikipedia.org/wiki/ClickOnce>.
- [17] «Wikipedia - WPF,» [En línea]. Available: [https://en.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation).
- [18] «Wikipedia - CRM,» [En línea]. Available: [https://es.wikipedia.org/wiki/Customer\\_relationship\\_management](https://es.wikipedia.org/wiki/Customer_relationship_management).
- [19] «Stackoverflow,» [En línea]. Available: <http://stackoverflow.com/>.
- [20] DevExpress, «Documentación WPF de DevExpress,» DevExpress Team, [En línea]. Available: <https://documentation.devexpress.com/#WPF/CustomDocument7875>.
- [21] Microsoft, «Microsoft Developer Network,» Microsoft, [En línea]. Available: <https://msdn.microsoft.com/es-es/dn308572.aspx>.
- [22] DevExpress, «Tutoriales de DevExpress para WPF,» DevExpress Team, [En línea]. Available: <https://www.devexpress.com/Support/Start/#!platform=wpf>.
- [23] DevExpress, «Tickets y ejemplos de DevExpress,» DevExpress Team, [En línea]. Available: <https://www.devexpress.com/Support/Center/Question/List/1>.

## Índice de tablas

---

Tabla 7.1: Plantilla para definir los Casos de Uso.....	31
Tabla 7.2: CU 0.a - Acceder al sistema .....	33
Tabla 7.3: CU 0.b – Funcionalidad general.....	34
Tabla 7.4: CU 1 - Gestionar CRM .....	34
Tabla 7.5: CU 2 - Gestionar Recursos .....	35
Tabla 7.6: CU 3 - Gestionar Proyectos.....	35
Tabla 7.7: CU 4 - Gestionar Vistas .....	35
Tabla 7.8: CU 5 - Cerrar sesión .....	35
Tabla 7.9: CU 1.1 Consultar clientes.....	36
Tabla 7.10: 1.2 - Registrar interacción .....	36
Tabla 7.11: CU 1.3 - Añadir cliente .....	36
Tabla 7.12: CU 1.1.1 - Consultar ficha .....	37
Tabla 7.13: CU 1.1.1.1 - Modificar información .....	38
Tabla 7.14: CU 1.1.1.2 - Eliminar cliente .....	38
Tabla 7.15: CU 2.1 - Fichar entrada/salida .....	39
Tabla 7.16: CU 2.2 Consultar historial propio .....	39
Tabla 7.17: CU 2.3 - Consultar historial trabajadores .....	39
Tabla 7.18: CU 2.4 Asignar tarea .....	40
Tabla 7.19: CU 3.1 - Consultar proyecto .....	40
Tabla 7.20: CU 3.2 - Consultar calendario .....	41
Tabla 7.21: CU 3.3 - Crear proyecto .....	41
Tabla 7.22: CU 3.4 Eliminar proyecto .....	41
Tabla 7.23: CU 4.1 - Cambiar tema.....	42
Tabla 7.24: CU 4.2 - Cambiar forma tablas.....	42
Tabla 9.1: Tabla USUARIOS.....	59
Tabla 9.2: Tabla CLIENTES .....	59
Tabla 9.3: Tabla FICHAJES.....	59
Tabla 9.4: Tabla INTERACCIONES .....	60
Tabla 9.5: Tabla PROYECTOS .....	60
Tabla 9.6: Tabla TAREAS.....	60
Tabla 9.7: Tabla PARTIPROY .....	61
Tabla 9.8: Tabla PARTITAREAS .....	61

## Índice de figuras

---

Figura 7.1: Identificación del actor USUARIO.....	31
Figura 7.2: Identificación del actor ADMINISTRADOR.....	31
Figura 7.3: Identificación del actor TRABAJADOR .....	32
Figura 7.4: Relación entre los actores del sistema.....	32
Figura 7.5: CU 0.a - Acceder al sistema .....	33
Figura 7.6: CU 0.b – Funcionalidad general.....	34
Figura 7.7: CU 1 - Gestionar CRM.....	36
Figura 7.8: CU 1.1 - Consultar clientes .....	37
Figura 7.9: CU 1.1.2 - Filtrar clientes .....	37
Figura 7.10: CU 1.1.1 - Consultar ficha.....	38
Figura 7.11: CU 2 - Gestionar Recursos.....	39
Figura 7.12: CU 3 - Gestionar Proyectos.....	40
Figura 7.13: CU 4 - Gestionar Vistas .....	41
Figura 7.14: Clase Usuario.....	45
Figura 7.15: Clase Fichaje .....	45
Figura 7.16: Clase Cliente .....	46
Figura 7.17: Clase Interaccion .....	47
Figura 7.18: Clase Proyecto .....	48
Figura 7.19: Clase Tarea .....	48
Figura 7.20: Diagrama de Clases .....	51
Figura 9.1: Diagrama Esquema Relacional .....	62
Figura 10.1: Diagrama de despliegue .....	64
Figura 11.1: Diagrama de actividad - Funcionalidad principal .....	66
Figura 11.2: Diagrama de actividad - Autenticarse .....	67
Figura 11.3: Diagrama de actividad - Fichar.....	68
Figura 11.4: Diagrama de actividad - Gestionar CRM .....	69
Figura 11.5: Diagrama actividad - Mostrar clientes .....	70
Figura 11.6: Diagrama de actividad - Gestionar Recursos.....	71
Figura 11.7: Diagrama actividad - Mostrar trabajadores .....	72
Figura 11.8: Diagrama de actividad - Gestionar Proyectos .....	74
Figura 11.9: Diagrama de actividad - Mostrar Proyectos.....	74
Figura 11.10: Diagrama de actividad - Gestionar vistas.....	75
Figura 11.11: Diagrama de actividad - Cerrar sesión .....	75
Figura 11.12: Diagrama actividad - Generar Nuevo elemento.....	76
Figura 12.1: Ventana Principal - Vista base.....	78
Figura 12.2: Ventana principal – Vista completa .....	78
Figura 12.3: Editores .....	79
Figura 12.4: Mensaje de error.....	80
Figura 12.5: Acerca de - Información .....	80
Figura 12.6: Autenticación .....	81
Figura 12.7: Autenticación – Cambio de cuenta .....	81

## Anexo I – Windows Presentation Foundation

---

### Introducción

Hasta 2006, el desarrollo de aplicaciones de escritorio en Windows se realizaba utilizando Windows Forms, también conocido como WinForms. En este año se lanzó la versión 3.0 del Framework .NET de Microsoft, en el que se introdujo esta nueva tecnología de desarrollo. El despliegue de esta tecnología ofrecía tanto la versión standalone como la XAML Browsers Applications (XBAP), siendo el desarrollo muy similar entre ambas.

Las aplicaciones desarrolladas con XBAP se ejecutan en un navegador web. No fue hasta la versión 4.0 de .NET (y en las siguientes) que las aplicaciones XBAP podían ejecutarse como aplicaciones de confianza total, es decir, que pueden usar todos los recursos del ordenador. Además, estas aplicaciones se ejecutan a través de un proceso diferente al del navegador, aunque de la sensación de que se están ejecutando en el navegador. Al principio, las aplicaciones desarrolladas con XBAP sólo podían ejecutarse en Internet Explorer. Con la salida de la versión 4.0 del Framework .NET se pudieron ejecutar en Firefox, y así ha sido hasta la fecha.

Las aplicaciones standalone son las aplicaciones que se instalan en la propia máquina para poder ejecutarse. Tienen siempre a su disposición todos los recursos del ordenador, es decir, son consideradas aplicaciones de total confianza. Estas aplicaciones se instalan utilizando Windows Instaler (anteriormente conocido como Microsoft Instaler – MSI [14]) o ClickOnce (instalador web de Microsoft [15]). [16]

### WPF

El objetivo de Windows Presentation Foundation (WPF) era, y sigue siendo, proporcionar avances en el entorno de Windows que permitieran crear interfaces que incorporaran documentos, componentes multimedia, gráficos bidimensionales y tridimensionales, animaciones, características tipo web, etc. Se pueden resaltar tres aspectos fundamentales:

- Una tecnología común para interfaces de usuario Windows y explorador web
- Una plataforma unificada para interfaces de usuario modernas
- La posibilidad de desarrolladores y diseñadores de trabajar conjuntamente

La creación de aplicaciones XAML para un explorador utilizando WPF se realiza utilizando XBAP, del que se ha hablado anteriormente, que permite realizar dos aplicaciones, una para escritorio y otra para explorador web, con el mismo código.

Esta plataforma unificada consta de varias características que consiguen otorgar las herramientas necesarias para crear unas interfaces de usuario más completas, simples y atractivas visualmente. Estas características son:

- **Direct3D:** Es la tecnología que se utiliza para renderizar los gráficos en Windows. Esta característica permite mostrar gráficos más complejos y temas personalizados, pudiendo descargar algunos procesos gráficos a la GPU, ya que ésta está mejor preparada para el procesamiento gráfico que la CPU.

- **Servicios de medios:** Se pueden crear interfaces utilizando muchos medios, bastante comunes, como pueden ser imágenes vectorizadas y rasterizadas, audio y vídeo, pudiendo introducir también animaciones y elementos 2D y 3D gracias a sus sistemas de animación y renderización. Las capacidades 3D de WPD son un subconjunto de las características que ofrece Direct3D.
- **Plantillas:** Las características de un elemento de WPF se pueden modificar directamente para definirlo o bien se pueden modificar indirectamente utilizando plantillas o estilos. Estos son un conjunto de ajustes que se predefinen para un tipo de elemento en concreto, y luego se pueden usar en cualquier momento haciendo uso de *keys* o bien de forma predeterminada.
- **Animaciones:** Ofrece animaciones basadas en tiempo, y no en frames. Esto hace que la velocidad sea independiente del rendimiento del sistema. Estas animaciones pueden ser activadas por eventos externos.
- **Imágenes:** Puede acceder al código del Windows Imaging Component (WIC), permitiendo a los desarrolladores escribir codécs específicos para sus propios formatos de imagen.
- **Efectos:** Se utiliza la clase Effects para generar efectos rasterizados en los componentes visuales, pudiendo especificar shaders (o sombreadores) personalizados.
- **Documentos:** WPF soporta de forma nativa documentos paginados, pudiendo verlos por páginas o en modo scroll. También reorganiza el texto si la zona cambia de tamaño.
- **Texto:** Soporta de manera nativa la tecnología OpenType, lo que otorga a los desarrolladores una amplia gama de características de renderizado de texto, pudiendo acceder a diversos tipos de fuentes, fracciones, sustitución de los glifos, ligaduras, entre otras posibilidades.

WPF tiene un conjunto de servicios de datos incluido conocido como Data binding. Esto permite a los desarrolladores conectar y manipular datos en las aplicaciones. WPF ofrece algunos controles predefinidos para el manejo de datos como pueden ser los botones, el menú, las mallas (grids) y las listas (list box). El enlace de los datos no determina su presentación, pues se pueden modificar los controles predefinidos mediante las plantillas. Existen cuatro tipos de data binding:

- **Una vez:** Cuando el cliente ignora actualizaciones del servidor.
- **Una vía:** Cuando el cliente tiene sólo acceso en modo lectura al servidor.
- **Dos vías:** Cuando el cliente puede introducir y leer datos en el servidor.
- **Una vía a la fuente:** Cuando el cliente sólo puede introducir datos en el servidor.

WPF también otorga interoperabilidad con Windows Forms, permitiendo así usar controles y crear controles de usuario de esta tecnología. [16]

WPF introdujo un nuevo lenguaje basado en XML centrado en la interfaz, siguiendo el éxito de los lenguajes de marcado en el desarrollo web. Este lenguaje es eXtensible Application Markup Language, XAML en su acrónimo, y fue diseñado como un método más eficiente para el desarrollo de interfaces de usuario.

Con este lenguaje se consigue que los diseñadores y los desarrolladores trabajen conjuntamente, ya que separa las capas de modelo y vista, permitiendo a los diseñadores ser más efectivos durante el ciclo de desarrollo. Se trata también de un lenguaje declarativo, por lo que se puede programar el comportamiento y la integración de los elementos sin tener que utilizar programación procedural, aunque todos los elementos de WPF pueden ser codificados en cualquiera de los dos lenguajes de .NET, C# y VB.NET.