

▪ Trabajo Fin de Grado ▪

*Ingeniería del Software*

Desarrollo y programación de aplicaciones para  
Microsoft Dynamics 365 con SDK .NET

---

*Alberto García Benito*

Noviembre 2018

*Director: José Miguel Blanco Arbe*



## AGRADECIMIENTOS

*Me gustaría agradecer a todas las personas que han estado a mi lado durante estos 4 años universitarios y que han contribuido de alguna forma a que haya podido finalizar este Trabajo Fin de Grado.*

*Gracias a mi familia por el apoyo incondicional que me han mostrado en todo momento al tomar la difícil decisión de volver a la universidad.*

*Gracias a mi director de proyecto, José Miguel Blanco Arbe, por su confianza, paciencia y perseverancia para que terminará el TFG en noviembre 2018.*

*Gracias a mis compañeros de Ibermática que hicieron que mi estancia en la empresa fuera muy amena y productiva.*

*Gracias a mis compañeros de clase por aguantarme durante estos años, en especial, a Jon Arzelus y Alain Collazo compañeros de grupo en mis dos primeros años, a Mikel San Sebastián que tomó el relevo en tercero, a Héctor López, Joseba Carral, Begoña Martínez...*

*Gracias a todos los profesores de la facultad de informática con los que he tenido un trato muy cercano y ameno.*

*Y por supuesto, no me quiero olvidar de mis amigos de toda la vida del barrio de Anaka ni de los Cork-ianos aunque haya tenido que “sufrir” alguna bromilla por volver a estudiar “a mi edad” pero siempre con mucho humor.*



## Resumen

El presente proyecto se ha llevado a cabo en Ibermática S.A., una de las principales compañías de servicios en Tecnologías de la Información del mercado español. Así, en esta memoria se describe el proceso empleado para desarrollar nuevas funcionalidades personalizadas para la plataforma Microsoft Dynamics CRM Online para su integración posterior con la plataforma de soporte BPM VDoc.

El resultado ha sido la creación de dos aplicaciones totalmente diferenciadas, por una parte, una aplicación de Windows que permite la creación automática de listas de marketing y la sincronización de los contactos entre plataformas y, por la otra, un plug-in que desencadena el flujo de registro de ventas desde el CRM.

Para alcanzar estos resultados hay que conocer las plataformas a integrar, analizar las necesidades de los interesados, plantear una solución y realizar pruebas software para garantizar que las aplicaciones cumplen las especificaciones funcionales.

La implementación se ha realizado empleando una gran variedad de tecnologías del framework .NET de Microsoft, destacando el kit de desarrollo software (SDK) para Microsoft Dynamics 365 que permite acceder a los datos y metadatos del CRM.

## Abstract

The present project has been carried out in Ibermática S.A., one of the main services companies in Information Technologies of the Spanish market. Thus, this report describes the process used to develop new customized features for the Microsoft Dynamics CRM Online platform for its integration with the BPM VDoc support platform.

The result has been the creation of two totally differentiated applications, on the one hand, a Windows application that allows the automatic creation of marketing lists and the synchronization of contacts between platforms and, on the other, a plug-in that triggers the Sales record flow from the CRM.

In order to achieve these results, it was necessary to know the platforms to be integrated, analyze the needs of the interested parties, propose a solution and perform software tests to guarantee that the functional specifications are met.

The implementation has been made using a variety of Microsoft .NET framework technologies, highlighting the software development kit (SDK) for Microsoft Dynamics 365 that allows access to CRM data and metadata.

# Laburpena

Proiektu hau Ibermática S.A.-n burutu da, Espainiako merkatuaren Informazio-Teknologiaren zerbitzu konpainia nagusietako bat. Horrela, memoria honek Microsoft Dynamics CRM Online plataformarako pertsonalizatutako ezaugarri berriak garatzeko prozesua azaltzen du, BPM VDoc plataformarekin integratzeko.

Erabat desberdinak diren bi aplikazioak sortzea izan da emaitza, alde batetik, Windows aplikazioa, marketing-zerrendak automatikoki sortzea eta plataformen arteko kontaktuak sinkronizatzea ahalbidetzen duena eta, bestetik, plugin-a CRM-ren salmenten erregistroa sortzen duena.

Emaitza horiek lortzeko, integratu behar diren plataformak ezagutu izan behar dira, interesatuen beharrak aztertu, irtenbide bat proposatu eta software-probak egin funtzionaltasun zehaztapenak betetzen direla bermatzeko.

Inplementazioa Microsoft-eko .NET framework teknologia barietate handia erabilia egin da baina, batez ere, Microsoft Dynamics 365 softwarearen garapen kit-a (SDK) azpimarratu behar da CRM datu eta metadatuak lortzeko gai delako.

# Índice

---

RESUMEN .....	5
ÍNDICE .....	7
LISTA DE FIGURAS Y TABLAS.....	9
<b>1. INTRODUCCIÓN .....</b>	<b>11</b>
<b>2. ANTECEDENTES .....</b>	<b>13</b>
2.1 PRESENTACIÓN DE IBERMÁTICA S.A.....	14
2.2 MICROSOFT DYNAMICS CRM ONLINE.....	17
2.3 PLATAFORMA BPM VDOC (ON-PREMISES) .....	19
2.4 SITUACIÓN DE PARTIDA .....	19
<b>3. OBJETIVOS.....</b>	<b>21</b>
3.1 OBJETIVOS GENERALES.....	22
3.2 OBJETIVOS ESPECÍFICOS.....	23
3.3 ALCANCE.....	24
3.3.1 REQUISITOS .....	24
3.3.2 EXCLUSIONES DEL ALCANCE .....	25
3.4 WBS/EDT.....	26
<b>4. ANÁLISIS DE REQUISITOS.....</b>	<b>27</b>
4.1 REQUISITOS FUNCIONALES .....	29
4.1.1 REQUISITOS FUNCIONALES DE LA APLICACIÓN CRM-MARKETING .....	29
4.1.2 REQUISITOS FUNCIONALES DEL PLUG-IN DE VENTAS .....	30
4.2 REQUISITOS NO FUNCIONALES.....	30
4.2.1 REQUISITOS NO FUNCIONALES DE LA APLICACIÓN CRM-MARKETING .....	30
4.2.2 REQUISITOS NO FUNCIONALES DEL PLUG-IN DE VENTAS .....	30
4.3 DIAGRAMA DE CASOS DE USO .....	31
4.4 FLUJO DE EVENTOS.....	32
<b>5. DISEÑO .....</b>	<b>35</b>
5.1 ARQUITECTURA EN TRES CAPAS .....	36
5.1.1 CAPA DE PRESENTACIÓN .....	37
5.1.2 CAPA DE NEGOCIO.....	37
5.1.3 CAPA DE DATOS .....	37
5.2 MODELO DE DATOS DE MICROSOFT DYNAMICS CRM ONLINE .....	38
5.2.1 MODELO DE DATOS DE LA APLICACIÓN CRM-MARKETING .....	38
5.2.2 MODELO DE DATOS DEL PLUG-IN DE VENTAS.....	39
5.3 DISEÑO DE LAS BASES DE DATOS.....	40
5.3.1 BASE DE DATOS DE LA APLICACIÓN CRM-MARKETING.....	40
5.3.2 BASE DE DATOS DEL PLUG-IN DE VENTAS .....	41
5.4 DISEÑO GRÁFICO DE LAS INTERFACES.....	42
<b>6. TECNOLOGÍAS Y ENTORNO DE DESARROLLO .....</b>	<b>45</b>
6.1 ¿QUÉ ES .NET? .....	46
6.2 TECNOLOGÍAS .....	47
6.2.1 FORMULARIOS DE WINDOWS .....	47
6.2.2 C# .....	47
6.2.3 WCF (WINDOWS COMMUNICATION FOUNDATION).....	48
6.2.4 TRANSACT-SQL .....	48

6.3	HERRAMIENTAS SOFTWARE .....	48
6.3.1	<i>MICROSOFT VISUAL STUDIO 2015</i> .....	48
6.3.2	<i>NUGET</i> .....	49
6.3.3	<i>KIT DE DESARROLLO SOFTWARE (SDK) PARA MICROSOFT DYNAMICS 365</i> .....	49
6.3.4	<i>MICROSOFT SQL SERVER 2014</i> .....	50
<b>7.</b>	<b>APLICACIÓN CRM-MARKETING</b> .....	<b>51</b>
7.1	SERVICIOS WEB DE MICROSOFT DYNAMICS 365 .....	52
7.2	CONEXIÓN AL SERVIDOR CRM .....	53
7.2.1	<i>PUNTO DE PARTIDA</i> .....	55
7.2.2	<i>IBERMÁTICA PROXY</i> .....	57
7.3	CONEXIÓN A LA BASE DE DATOS .....	58
7.4	CONFIGURACIÓN DEL SERVIDOR CRM .....	58
7.5	CONFIDENCIALIDAD DE LA CONTRASEÑA DE USUARIO .....	59
7.6	FUNCIONALIDAD “CREAR LISTA DE MARKETING” .....	60
7.6.1	<i>CRITERIOS PARA LA DEFINICIÓN DE UNA LISTA DE MARKETING</i> .....	60
7.6.2	<i>CREACIÓN DE LISTAS DE MARKETING (CÓDIGO)</i> .....	61
7.6.3	<i>CREAR DE LISTAS DE MARKETING (APLICACIÓN CRM-MARKETING)</i> .....	63
7.7	FUNCIONALIDAD “SINCRONIZAR CONTACTOS CRM” .....	65
<b>8.</b>	<b>PLUG-IN DE VENTAS</b> .....	<b>67</b>
8.1	PLUG-INS EN MICROSOFT DYNAMICS 365 .....	68
8.2	PLUG-IN DE VENTAS EN MICROSOFT DYNAMICS CRM ONLINE .....	70
8.2.1	<i>PROYECTO ‘PLUG-INS’</i> .....	71
8.2.2	<i>SITIO WEB ‘CRM2SQLWS’</i> .....	73
8.3	REGISTRO DEL PLUG-IN DE VENTAS .....	75
8.4	PRUEBA DE FUNCIONAMIENTO DEL PLUG-IN DE VENTAS .....	76
<b>9.</b>	<b>PRUEBAS</b> .....	<b>79</b>
9.1	PRUEBAS DE SISTEMA .....	80
9.2	PRUEBA FINAL DE ACEPTACIÓN .....	82
<b>10.</b>	<b>GESTIÓN</b> .....	<b>85</b>
10.1	GESTIÓN DEL ALCANCE .....	86
10.2	GESTIÓN DEL CRONOGRAMA .....	87
10.3	GESTIÓN DE COSTES .....	88
10.4	GESTIÓN DE LA CALIDAD .....	89
10.4.1	<i>ASEGURAMIENTO Y CONTROL DE LA CALIDAD</i> .....	89
10.4.2	<i>CRITERIOS DE CALIDAD</i> .....	89
10.5	GESTIÓN DE RIESGOS .....	90
10.5.1	<i>IDENTIFICACIÓN DE RIESGOS Y RELEVANCIA</i> .....	90
10.5.2	<i>PLAN DE ACCIÓN</i> .....	91
10.6	GESTIÓN DE LOS INTERESADOS .....	92
<b>11.</b>	<b>CONCLUSIONES</b> .....	<b>93</b>
11.1	PROPUESTAS DE MEJORA Y LÍNEAS DE FUTURO .....	95
11.2	LECCIONES APRENDIDAS .....	96
11.2.1	<i>DISPONIBILIDAD DE LAS FUENTES DE INFORMACIÓN</i> .....	96
11.2.2	<i>SEGURIDAD DE LOS SERVICIOS WEB</i> .....	97
11.2.3	<i>DOCUMENTACIÓN DEL CÓDIGO</i> .....	97
11.2.4	<i>KIT DE DESARROLLO SOFTWARE (SDK) PARA MICROSOFT DYNAMICS 365</i> .....	98
	<b>Bibliografía y Referencias</b> .....	<b>99</b>
	<b>Anexo A: Guía para el Registro del Plug-in de Ventas</b> .....	<b>101</b>
	<b>Anexo B: Pruebas aplicación CRM-Marketing</b> .....	<b>105</b>
	<b>Anexo C: Certificado del Producto</b> .....	<b>112</b>



# Lista de Figuras y Tablas

---

## Figuras

FIGURA 2.1: ENTRADA PRINCIPAL A IBERMÁTICA S.A. ....	14
FIGURA 2.2: EVOLUCIÓN HISTÓRICA DE IBERMÁTICA S.A. <sup>2</sup> .....	15
FIGURA 2.3: ESTRUCTURA ORGANIZATIVA DE IBERMÁTICA S.A. <sup>3</sup> .....	16
FIGURA 2.4: MENU PRINCIPAL DE MICROSOFT DYNAMICS CRM ONLINE .....	17
FIGURA 2.5: ENTIDADES CRM IB INDUSTRIA.....	18
FIGURA 2.6: CAMPOS DE LA ENTIDAD CUENTA .....	18
FIGURA 3.1: EDT DEL PROYECTO.....	26
FIGURA 4.1: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN CRM-MARKETING .....	31
FIGURA 5.1: MODELO DE PROGRAMACIÓN EN TRES CAPAS .....	36
FIGURA 5.2: MODELO DE DATOS APLICACIÓN CRM-MARKETING .....	39
FIGURA 5.3: MODELO DE DATOS PLUG-IN DE VENTAS .....	40
FIGURA 5.4: ESQUEMA BD APLICACIÓN CRM-MARKETING .....	40
FIGURA 5.5: ESQUEMA BD PLUG-IN DE VENTAS .....	41
FIGURA 5.6: DISEÑO VENTANA PRINCIPAL APLICACIÓN CRM-MARKETING .....	43
FIGURA 5.7: DISEÑO VENTANA “USER CRM CREDENTIALS” .....	44
FIGURA 7.1: ARCHIVOS DEL PROGRAMA QUICKSTART .....	55
FIGURA 7.2: EJECUCIÓN PROGRAMA QUICKSTART (PASO 1).....	55
FIGURA 7.3: EJECUCIÓN PROGRAMA QUICKSTART (PASO 2).....	55
FIGURA 7.4: INSTANCIA DEL PROXY DEL SERVICIO DE ORGANIZACIÓN .....	56
FIGURA 7.5: CLASE CONFIGURACIÓN DEL SERVIDOR DEL CRM .....	56
FIGURA 7.6: CREDENCIALES DE WINDOWS DEL CRM.....	57
FIGURA 7.7: CONFIGURACIÓN DEL PROXY DE IBERMÁTICA.....	57
FIGURA 7.8: CADENA DE CONEXIÓN A LA BASE DE DATOS .....	58
FIGURA 7.9: ARCHIVO CREDENTIALS.XML: CONFIGURACIONES SERVIDORES CRM .....	58
FIGURA 7.10: UBICACIÓN DEL ARCHIVO CREDENTIALS.XML.....	59
FIGURA 7.11: FUNCIÓN READCONFIGURATIONS().....	59
FIGURA 7.12: FUNCIÓN CONVERTTOSECURESTRING().....	59
FIGURA 7.13: FUNCIÓN COUNTREPEATEDLMS() .....	61
FIGURA 7.14: MÉTODO CREATELM().....	62
FIGURA 7.15: APP: RESULTADO FUNCIONALIDAD “CREAR LISTAS DE MARKETING” .....	63
FIGURA 7.16: CRM: INFORMACIÓN GENERAL DE LISTA_CAD_CON .....	64
FIGURA 7.17: CRM: VISTA DE CONTACTOS (1) .....	64
FIGURA 7.18: MÉTODO UPDATECRMCONTACT().....	65
FIGURA 7.19: APP: RESULTADO FUNCIONALIDAD “SINCRONIZAR CONTACTOS CRM” .....	66
FIGURA 7.20: CRM: VISTA DE CONTACTOS (2) .....	66
FIGURA 8.1: ACCESO AL CONTEXTO DE EJECUCIÓN DEL PLUG-IN.....	68
FIGURA 8.2: ACCESO AL SERVICIO DE ORGANIZACIÓN.....	68
FIGURA 8.3: ACCESO AL SERVICIO DE SEGUIMIENTO (TRAZAS).....	69
FIGURA 8.4: CONFIGURACIÓN DEL SISTEMA DEL CRM.....	69
FIGURA 8.5: ARCHIVOS DEL PROYECTO ‘PLUG-INS’ .....	70
FIGURA 8.6: ARCHIVOS DEL SITIO WEB ‘CRM2SQLWS’ .....	71
FIGURA 8.7: MÉTODO EXECUTE DE LA INTERFAZ IPLUGIN .....	72
FIGURA 8.8: EVENTO DEL CONTEXTO Y PARÁMETROS DE ENTRADA .....	72
FIGURA 8.9: REFERENCIA AL SERVICIO DE ORGANIZACIÓN .....	72
FIGURA 8.10: PROCEDIMIENTO GETOPPORTUNITYATTRIBUTES().....	72
FIGURA 8.11: CONFIGURAR CONEXIÓN CON EL SERVICIO WCF .....	73

FIGURA 8.12: CONTRATO DE SERVICIO DEL SERVICIO WCF .....	74
FIGURA 8.13: CONTRATO DE DATOS DEL SERVICIO WCF .....	74
FIGURA 8.14: FUNCIÓN REGISTRAROP(1) .....	74
FIGURA 8.15: FUNCIÓN REGISTRAROP(2) .....	75
FIGURA 8.16: CRM: INFORMACIÓN GENERAL ENTIDAD OPORTUNIDAD .....	77
FIGURA 8.17: CRM: INFORMACIÓN SOBRE ENTIDAD IMPORTES .....	77
FIGURA 8.18: BD: TABLA OPORTUNIDAD .....	77
FIGURA 8.19: BD: TABLA IMPORTES .....	78
FIGURA 9.1: APP: VISTA "USER CRM CREDENTIALS" .....	82
FIGURA 9.2: CREDENCIALES DE WINDOWS .....	82
FIGURA 9.3: APP: SELECCIÓN DEL SERVIDOR DE CRM .....	83
FIGURA 9.4: APP: SELECCIÓN DE LA LISTA DE MAILING .....	83
FIGURA 9.5: APP: CONFIRMACIÓN DE LA LISTA DE MARKETING .....	83
FIGURA 9.6: APP: ARCHIVO LOG DE LA LISTA DE MARKETING .....	84
FIGURA 10.1: DIAGRAMA DE GANTT .....	87

## Tablas

TABLA 4.1: FLUJO DE EVENTOS DE "INICIALIZAR APLICACIÓN" .....	32
TABLA 4.2: FLUJO DE EVENTOS DE "CREAR LM" .....	33
TABLA 4.3: FLUJO DE EVENTOS DE "ACTUALIZAR CONTACTOS" .....	33
TABLA 4.4: FLUJO DE EVENTOS DE "AÑADIR CREDENCIALES" .....	34
TABLA 9.1: PRUEBA ESCENARIO 4 CASO 1 .....	81
TABLA 10.1: TABLA DE DEDICACIONES DE TIEMPO .....	88

# 1

---

## 1. *Introducción*

El proyecto que se presenta ha sido realizado por el alumno Alberto García Benito entre el 4 de junio y el 21 de septiembre del año 2018 en la empresa IBERMÁTICA S.A., Paseo de Mikeletegi 5, San Sebastián, como Trabajo de Fin de Grado de ingeniería informática, especialidad ingeniería del software, para la Facultad de Informática de San Sebastián - UPV/EHU. Este proyecto ha contado con la colaboración de Xabier Philippe Cantero Berrouet, ingeniero informático en el departamento de ingeniería de IBERMÁTICA S.A. y ha sido supervisado por José Miguel Blanco Arbe, profesor del departamento de Lenguajes y Sistemas Informáticos de la Facultad de Informática de San Sebastián, como director del Trabajo de Fin de Grado por parte de la UPV/EHU.

El presente proyecto pretende solucionar algunos de los problemas que se producen al gestionar información proveniente de los clientes con diferentes aplicaciones informáticas. En concreto, se quiere desarrollar un software que posibilite el intercambio de información de forma automática entre la herramienta software para la gestión de clientes, **Microsoft Dynamics CRM Online**, y la plataforma de soporte **BPM VDoc On-premises** para el control de las incidencias de los productos vendidos a esos mismos clientes. Éstas son dos de las herramientas informáticas más utilizadas en el departamento comercial de Ibermática.

La principal novedad será el uso por primera vez en la empresa del **kit de desarrollo de software** (SDK) de Microsoft Dynamics 365 (online y on-premises) para la implementación de estas aplicaciones informáticas. Este kit proporciona un conjunto de bibliotecas y herramientas

que permite a los desarrolladores la creación de aplicaciones para Microsoft Dynamics CRM Online.

La estructura de este documento, la memoria del proyecto, se divide en los siguientes capítulos:

- **Introducción:** Se realiza una breve explicación del proyecto, se indica la estructura de la memoria del proyecto y se presenta el contenido de los diferentes capítulos que contiene.
- **Antecedentes:** Elementos significativos del contexto en el que se realiza el proyecto. Incluye una presentación de la organización y negocio de Ibermática, una introducción a la plataforma Microsoft Dynamics CRM Online y una descripción de la situación de partida.
- **Objetivos:** Se exponen los objetivos generales y específicos, se define el alcance del proyecto junto con sus exclusiones y se representa la estructura de descomposición del trabajo (EDT).
- **Análisis de Requisitos:** Se realiza una reflexión de los requerimientos y expectativas de los interesados para obtener los requisitos funcionales y no funcionales. También se incluye el diagrama de casos de uso junto con el flujo de eventos para cada caso.
- **Diseño:** Se va a explicar la arquitectura de programación en tres capas como modelo desarrollo del software, el modelo de datos del CRM centrándonos en los datos de las entidades a las que acceden las aplicaciones y el modelo de las bases de datos. Finalmente, se incluye el diseño gráfico de las interfaces.
- **Herramientas y Entorno de Desarrollo:** Las herramientas utilizadas y el entorno de desarrollo elegido para implementar los productos.
- **Aplicación CRM-Marketing:** Se explica el proceso seguido para la implementación de la aplicación, los servicios web necesarios para conectarse al CRM, el punto de partida y las funcionalidades implementadas. Se incluyen también fragmentos de código significativos que reflejan características relevantes de la aplicación.
- **Plug-in de Ventas:** Se realiza una introducción sobre cómo implementar plug-ins para Dynamics 365, se explican los dos proyectos que forman parte de la solución adoptada para el plug-in de ventas, el proceso de registro del plug-in y una prueba de funcionamiento. Se incluyen también fragmentos de código significativos que reflejan características relevantes del plug-in.
- **Pruebas:** Se muestran las pruebas realizadas para testear el correcto funcionamiento de la aplicación CRM-Marketing. Se realizan pruebas de sistema para validar que el software cumple los requisitos funcionales y pruebas de aceptación, donde el cliente prueba el software para confirmar que cumple con sus expectativas.
- **Gestión del Proyecto:** Se relata la gestión realizada de las principales áreas de conocimiento del proyecto como son la gestión del alcance, gestión del cronograma, gestión de costos, gestión de la calidad, gestión de riesgos y gestión de los interesados.
- **Conclusiones:** Se realiza una revisión reflexiva de los resultados obtenidos en el proyecto y se compara con los objetivos inicialmente establecidos. Asimismo, se incluyen propuestas futuras de mejora para las aplicaciones realizadas y algunas lecciones aprendidas tras la realización del proyecto.
- **Bibliografía:** Listado de las diferentes fuentes bibliográficas utilizadas en la elaboración y desarrollo del proyecto.

# 2

---

---

## 2. *Antecedentes*

El conocimiento del contexto en el que se sitúa y se realiza el proyecto es un apartado importante para poder tener una correcta perspectiva de las motivaciones que dieron lugar al mismo y posteriormente, tener una mejor comprensión del porqué de algunas de las decisiones y soluciones adoptadas durante su implementación.

El capítulo comienza haciendo una pequeña presentación de la compañía IBERMÁTICA S.A. que incluye una descripción de su negocio, actividades y su estructura organizativa dentro de su contexto histórico.

En segundo lugar, se realiza una introducción de las dos herramientas informáticas que se quieren integrar: Microsoft Dynamics CRM online y BPM VDoc On-premises. Esta parte se centrará especialmente en explicar la estructura y las funcionalidades del CRM (Customer Relationship Management).

Finalmente, se realizará una descripción de la situación inicial del proyecto, es decir, el punto de partida junto con sus condicionantes y motivaciones para su realización.

## 2.1 Presentación de Ibermática S.A.

---

**Ibermática, S.A.** [1] es una compañía de servicios en Tecnologías de la Información implantada desde los años 70 en el mercado español.



*Figura 2.1: Entrada principal a Ibermática S.A.*

La actividad empresarial de Ibermática se desarrolla principalmente en las siguientes áreas de negocio:

- Consultoría TIC
- Equipamientos e infraestructuras
- Integración de sistemas de información
- Outsourcing
- Implantación de soluciones integradas de gestión empresarial

Ibermática se identifica como una compañía con personal altamente cualificado y con gran experiencia y conocimiento dentro del mundo de las Tecnologías de la Información y la Comunicación (TIC), capaz de asesorar tanto a empresas como a instituciones u organizaciones y de ofrecer soluciones integrales para mejorar su competitividad y desarrollo corporativo y social.

Según Ibermática<sup>1</sup>, “su misión es aportar valor a nuestros clientes, mercado y sociedad, prestando atención a sus retos y necesidades particulares para identificar y desarrollar las mejores oportunidades de avance tecnológico y transformación digital desde el conocimiento, el compromiso y la cercanía”.

---

<sup>1</sup> Sitio web de Ibermática: <http://ibermatica.com/quienes-somos/>

Desde su creación en 1973, Ibermática se ha caracterizado por su capacidad constante para evolucionar y así, poder adaptarse a los cambios continuos producidos en el mercado de las Tecnologías de la Información y la Comunicación, siendo su apuesta por el desarrollo, la innovación y la competitividad sus principales ventajas competitivas respecto a sus competidores.

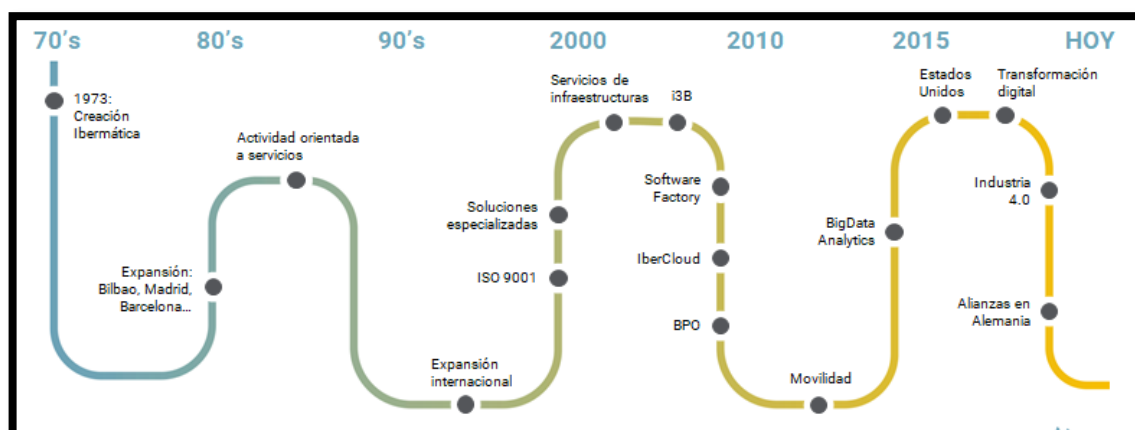


Figura 2.2: Evolución histórica de Ibermática S.A.<sup>2</sup>

Ibermática ha experimentado un continuo crecimiento que ha dado lugar a la apertura progresiva de sedes tanto en el mercado español como en el mercado internacional, sobre todo, en Latinoamérica. Como resultado, Ibermática se ha situado entre una de las cinco principales compañías de servicios en Tecnología de la Información del mercado español, tiene una plantilla superior a 3.200 empleados y presenta un volumen de negocio de 235 millones de euros.

Actualmente, el modelo de negocio de Ibermática se centra en los siguientes sectores de actividad:

- Industria (18%)
- Administración Pública (13%)
- Telecomunicaciones y Media (9%)
- Utilities (8%)
- Servicios Financieros y Seguros (16%)
- Salud (5%)
- Servicios (31%)

<sup>2</sup> Ibermática. Presentación corporativa 2018

## Estructura Organizativa

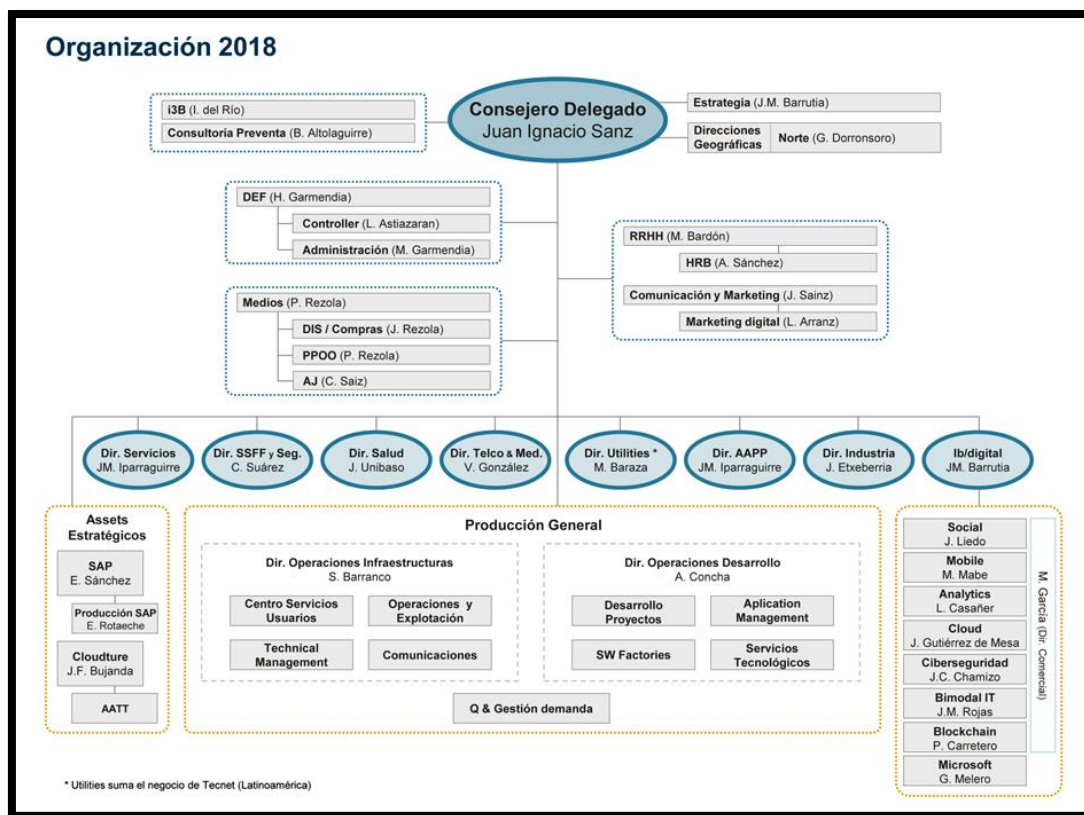


Figura 2.3: Estructura organizativa de Ibermática S.A.<sup>3</sup>

## Grupo Ibermática<sup>4</sup>

El Grupo de Ibermática consta de las siguientes filiales:

- **i3B (Instituto Ibermática de Innovación):** Centro de investigación aplicada con la finalidad de aportar al mercado soluciones innovadoras basadas en las Tecnologías de la Información.
- **Tecnet:** Filial de Ibermática en Latinoamérica, líder en la integración de sistemas para la automatización, el control y las comunicaciones de los procesos de negocio.
- **Sqédio:** Mayor proveedor en Portugal de sistemas y software para ingeniería y desarrollo de productos en 3D.
- **Ibermática Social:** Centro Especial de Empleo orientado a la integración social de las personas con capacidades diferentes, a partir de su incorporación al mercado laboral.

<sup>3</sup> Imagen obtenida de la Intranet de Ibermática

<sup>4</sup> Sitio web de Ibermática: <http://ibermatica.com/quienes-somos/empresas-del-grupo/>



## 2.2 Microsoft Dynamics CRM Online

Microsoft Dynamics CRM Online es una herramienta software desplegada en la nube para la gestión de las relaciones con los clientes y que ha sido desarrollada por Microsoft Corporation. El CRM<sup>5</sup> (**C**ustomer **R**elationship **M**anagement) almacena una gran cantidad de información sobre los clientes en una sola base de datos y permite realizar un seguimiento de todas las interacciones entre la empresa y sus clientes.

El objetivo final de uso del CRM es la optimización de la rentabilidad de las relaciones de la empresa con sus clientes. Alguno de los beneficios que se obtienen utilizando el CRM son:

- Mejorar los procesos ventas, marketing y servicio al cliente.
- Mejorar la satisfacción de los clientes.
- Fomentar la fidelidad de los clientes.
- Aumentar la base de clientes.
- Incrementar el volumen de ventas.

Microsoft Dynamics CRM Online consta de tres módulos principales: ventas, marketing y servicio al cliente. El módulo de configuración, que permite cambiar las características del CRM, solamente está disponible para cuentas de administrador o para cuentas de usuarios a los que se les haya otorgado los permisos pertinentes.

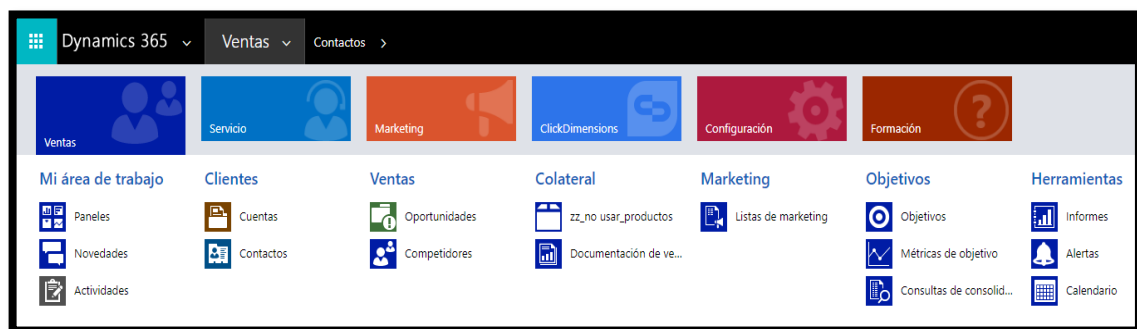


Figura 2.4: Menu principal de Microsoft Dynamics CRM Online

Desde el módulo de ventas de CRM Online se puede realizar un seguimiento a las cuentas (empresas) y sus contactos, administrar clientes potenciales, crear oportunidades, crear ofertas, generar pedidos, generar facturas, realizar un seguimiento a sus competidores, administrar la documentación de ventas, etc.

Desde el módulo de marketing de CRM Online se pueden crear listas de marketing (con contactos o clientes potenciales), crear y editar campañas, crear campañas exprés, añadir activos a una campaña, agregar actividades a una campaña y la realización del seguimiento de respuestas de una campaña.

---

<sup>5</sup> CRM (Customer Relationship Management): Este término se usará durante toda la memoria.

En Ibermática, una vez la venta de un producto se ha completado el servicio de atención al cliente se realiza a través de la plataforma de soporte VDoc.

La organización de los datos del CRM se corresponde con la estructura de una base de datos relacional en la que a cada tabla le corresponde una entidad del CRM. Las entidades están formadas por campos que son los atributos que caracterizan dicha entidad.

Los campos de las entidades pueden tener diferentes tipos de datos con formatos específicos del CRM como pueden ser *texto*, *fecha*, *número entero*, *búsqueda*, *dos opciones*, *conjunto de opciones*, *divisa*, *GUID*, etc. El GUID (Global Unique Identifier) es un string de 32 caracteres alfanuméricos que se usa como clave principal.

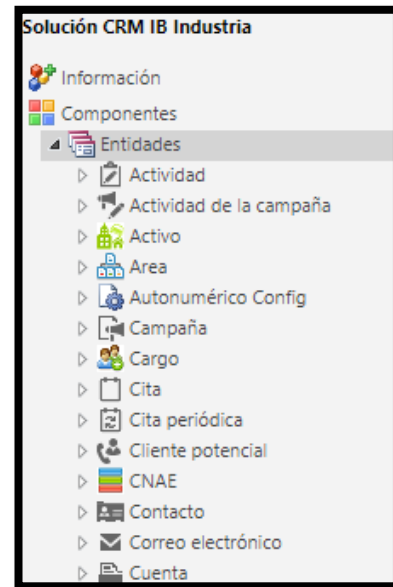


Figura 2.5: Entidades CRM IB Industria

Cada campo cuenta con una clave principal que se origina automáticamente al crear un nuevo registro y que permite identificar unívocamente ese registro dentro de una entidad. Así, el CRM permite mantener relaciones de tipo 1:N, N:1 y N:N entre las distintas entidades a través de sus claves principales.

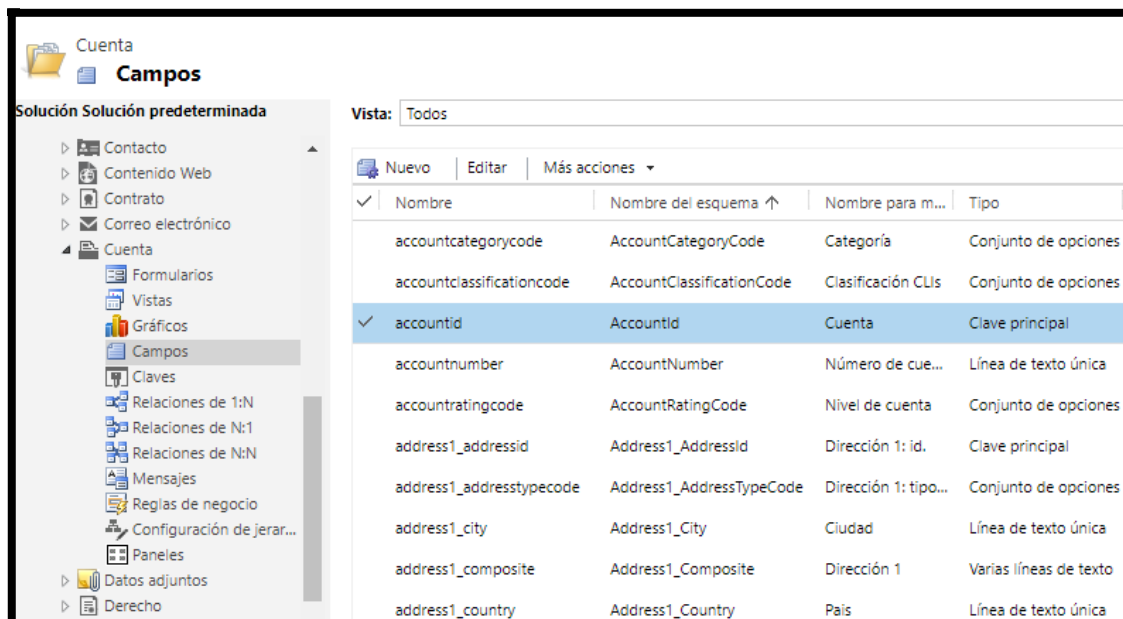


Figura 2.6: Campos de la entidad cuenta

## 2.3 Plataforma BPM VDoc (On-premises)

---

La plataforma de soporte BPM VDoc se utiliza para la gestión del servicio de atención a los clientes y permite llevar un seguimiento de todas las incidencias generadas sobre los productos vendidos.

## 2.4 Situación de partida

---

Microsoft Dynamics CRM Online y la plataforma de soporte BPM VDoc (On-premises) son dos de las herramientas software más utilizadas en el departamento comercial de Ibermática, para la gestión de las relaciones con sus clientes.

Sin embargo, ambas herramientas se utilizan en fases diferentes de las relaciones con los clientes. El CRM de Microsoft Dynamics se utiliza para llevar un control de las cuentas (empresas) y sus contactos, realizar el seguimiento de todo el proceso de venta de los productos y para gestionar las campañas de marketing. En cambio, la plataforma VDoc se utiliza una vez la venta del producto ya se ha realizado para gestionar todas las incidencias relacionadas con los productos vendidos.

De este modo, ambas herramientas comparten información sobre los clientes pero son funcionalmente independientes y no están intercomunicadas, es decir, no existe ningún flujo de datos automático entre ellas.

Por ejemplo, los trabajadores de Ibermática tienen que añadir manualmente en la plataforma de soporte VDoc información proveniente del CRM cada vez que se cierra una oportunidad como ganada, es decir, cuando se registra una venta de un producto. Por otra parte, se quieren utilizar los contactos de los clientes ya existentes en la plataforma VDoc para la creación automática de listas de marketing en el CRM y, a su vez, suprimir los procesos de importación y exportación que incluyen la carga y descarga de ficheros con información relevante sobre los clientes.

De este modo, la motivación principal de este proyecto es dar solución a los problemas aquí planteados mediante el desarrollo de aplicaciones personalizadas para

Microsoft Dynamics CRM Online para su posterior integración con la plataforma de soporte VDoc.

# 3

---

---

## **3. *Objetivos***

En este capítulo, se declaran los objetivos generales explicando que beneficios se pretenden obtener tras la finalización del proyecto. Después se enumeran los objetivos específicos que tenemos que cumplir para poder alcanzar dichos objetivos,

En la parte final, se define el alcance del proyecto incluyendo los requisitos y las exclusiones. En el último apartado se muestra la EDT (Estructura de Descomposición del Trabajo).

### 3.1 Objetivos generales

---

El objetivo general que motiva este proyecto es implementar un software que permita el intercambio de información entre la herramienta software para la gestión de clientes, Microsoft Dynamics CRM Online, y la plataforma de soporte BPM VDoc (on-premises) para la gestión de las incidencias de los productos vendidos a los clientes de Ibermática, para una integración futura entre ambas aplicaciones.

Los principales beneficios que se pretenden alcanzar sobre la gestión de la información que se comparte entre ambas plataformas son los siguientes:

- **Sincronización de la información entre el CRM y VDoc:** Se pretende asegurar que la información compartida existente sobre los clientes sea la misma y que se encuentre correctamente actualizada en ambas plataformas. Se quieren evitar los posibles errores que se pueden producir al tener que copiar información o en los procesos de importación / exportación de datos entre ambas aplicaciones.
- **Eliminación de tareas para gestión de la información compartida entre el CRM y VDoc:** Se pretende evitar tener que volver a introducir información duplicada que ya se encuentra almacenada en alguna de las dos plataformas. De esta forma, se previene tener que copiar información de una aplicación a otra o tener que exportar y luego importar datos entre ambas. Además, al eliminar estas tareas administrativas los trabajadores se liberan de carga de trabajo no productiva.
- **Automatización de los flujos de información entre el CRM y VDoc:** Se pretende automatizar el intercambio de información para algunas de las actividades, de forma que se programen eventos que desencadenen flujos de datos entre ambas plataformas. El beneficio obtenido es doble, por una parte, se garantiza el intercambio de la información de forma inmediata, casi en tiempo real y sin errores, y por otra parte, se eliminan tareas administrativas como la introducción manual de información.
- **Reducción de los riesgos derivados del acceso a información sensible sobre clientes:** La información sobre los clientes es de un alto nivel estratégico para Ibermática, de esta forma, se quiere restringir el acceso a este tipo información. Así, excepto para ciertos usuarios con los permisos pertinentes, el acceso a la información de los clientes se quiere reducir a la realización de consultas para evitar posibles fugas de información de los archivos de descarga con datos de clientes.

## 3.2 Objetivos específicos

---

En este punto, es necesario advertir que el presente proyecto solamente abarca una parte del proceso de integración entre ambas herramientas software. Las funcionalidades que se pretenden desarrollar en una primera etapa son las siguientes:

- **Creación automática de listas de marketing:** Se pretende la creación automática de listas de marketing en el CRM para realizar campañas de mailing con datos de los clientes provenientes de la plataforma VDoc.
- **Sincronización de los contactos entre el VDoc y CRM:** Se requiere una nueva funcionalidad para poder actualizar de forma sencilla la información en el CRM sobre los contactos (clientes) provenientes de la plataforma VDoc.
- **Desencadenamiento automático del flujo de registro de ventas en VDoc:** Se pretende que cuando se realice una venta de un producto en el CRM se produzca un evento que dé lugar a la creación automática de un registro de ventas en la plataforma VDoc con los datos de esa misma venta provenientes del CRM.

### 3.3 Alcance

---

La definición del alcance viene determinada por los requisitos y expectativas que nos han expresado los interesados y que, en el apartado anterior, se han resumido como objetivos específicos del proyecto.

De este modo, el alcance de este proyecto es la creación de aplicación(es) informática(s) para Microsoft Dynamics CRM Online que implementen las siguientes funcionalidades:

1. Creación automática de listas de marketing en el CRM.
2. Sincronización de contactos entre el VDoc y el CRM.
3. Desencadenamiento del flujo de registro de ventas en el VDoc desde el CRM.

En general, el autor de este proyecto ha tenido libertad absoluta para desarrollar las soluciones que implementen estas funcionalidades, sin embargo, también es cierto que los interesados han impuesto algunas limitaciones y restricciones. De este modo, la implementación de esta(s) aplicación(es) informática(s) se debe realizar utilizando el kit de desarrollo de software (SDK) de Microsoft Dynamics 365 y el framework .NET de Microsoft como entorno de desarrollo.

#### 3.3.1 *Requisitos*

El producto final debe incluir además los siguientes requerimientos:

- **Conexión a una base de datos:** El intercambio de datos entre el CRM de Microsoft Dynamics 365 y la plataforma de soporte BPM VDoc se debe realizar a través de una base de datos que almacene la información de los clientes.
- **Sistema de mensajes de información:** Las aplicaciones deben proveer al usuario la información necesaria para facilitar el uso de la aplicación, recopilar información sobre los resultados obtenidos e informar sobre los errores producidos al ejecutar la aplicación si se diera el caso.
- **Control de acceso a las aplicaciones:** Se debe restringir el acceso a las aplicaciones únicamente a los usuarios autorizados con el objeto de garantizar la confidencialidad, la disponibilidad e integridad de la información.
- **Simplicidad y facilidad de uso de las aplicaciones:** Se debe optar por un diseño de las aplicaciones simple, claro y fácil de usar, donde se da mayor relevancia a las funcionalidades de la aplicación y en menor medida al aspecto visual de las mismas.
- **Idioma:** Las aplicaciones deberán desarrollarse íntegramente en inglés incluyendo la información que se presente al usuario.



### 3.3.2 *Exclusiones del alcance*

Los siguientes puntos se han excluido del alcance del proyecto:

- El estudio de alternativas a las tecnologías utilizadas para la realización de este proyecto puesto que están impuestas por los interesados.
- El desarrollo del módulo que conecte la(s) base(s) de dato(s) donde se almacena la información compartida sobre los clientes con la plataforma de soporte VDoc.
- El software se implementa en un entorno de desarrollo de Microsoft Dynamics CRM Online que es una réplica exacta del que utilizan los trabajadores comerciales de Ibermática, de este modo, queda fuera del alcance del proyecto la implantación de las aplicaciones en un entorno de trabajo real.

### 3.4 WBS/EDT

---

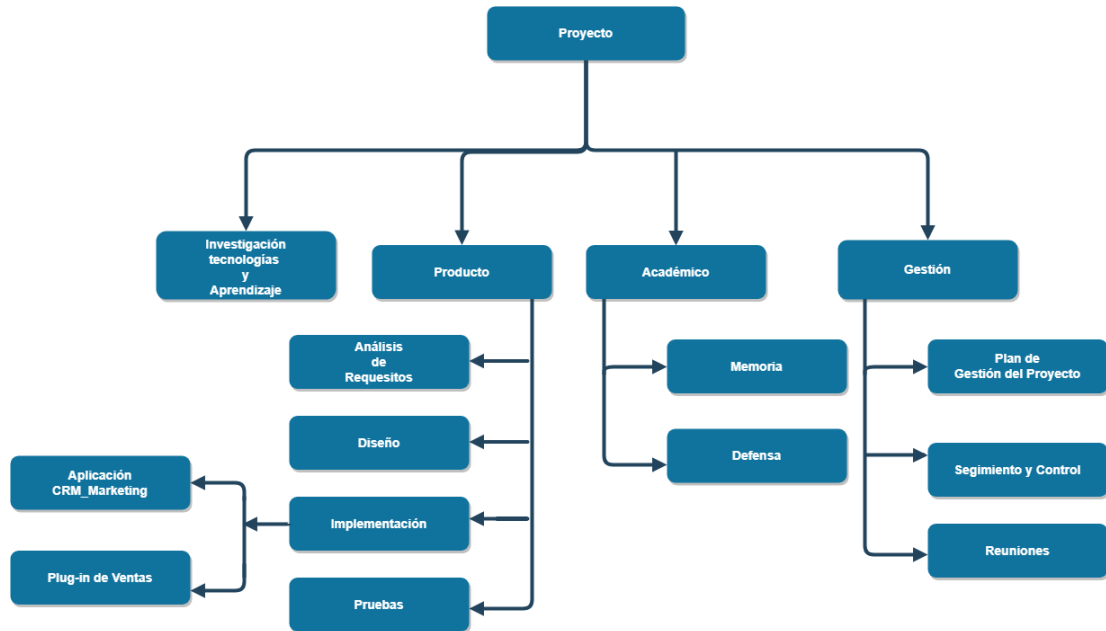


Figura 3.1: EDT del proyecto

# 4

---

---

## 4. *Análisis de Requisitos*

La idea inicial para este proyecto era intentar desarrollar las tres funcionalidades expuestas en el alcance del capítulo anterior en una sola aplicación informática. Sin embargo, rápidamente se desechó esta opción tras analizar para cada funcionalidad el tipo de información, la forma de acceder a la información y el flujo de información entre el CRM y VDoc.

El tipo de información a la que acceden las dos primeras funcionalidades, véase *creación automática de listas de marketing en el CRM* y *sincronización de contactos entre el VDoc y el CRM*, son datos sobre los contactos de clientes de la plataforma VDoc y en el caso de la última funcionalidad, véase *desencadenamiento del flujo de registro de ventas en el VDoc desde el CRM*, por el contrario se accede a la información básica sobre la venta de un producto en el CRM.

Por otra parte, para la funcionalidad *creación automática de listas de marketing en el CRM*, se precisa la interacción humana con la aplicación, en cambio, para la última funcionalidad, *desencadenamiento del flujo de registro de ventas en el VDoc desde el CRM*, es preferible que el flujo de ventas se desencadene automáticamente sin intervención humana.

En el caso de la funcionalidad restante no había una preferencia definida.

Finalmente, mientras que para las dos primeras funcionalidades el flujo de información sobre los clientes proviene de la plataforma de soporte VDoc y tiene que actualizar la información en Microsoft Dynamics CRM Online, es decir, la información se transmite desde la plataforma VDoc al CRM, en el caso de la última funcionalidad el sentido del flujo de información es el contrario, es decir, tiene origen en el CRM y tiene que actualizar la información en la plataforma VDoc. Recuérdese que la intercomunicación entre el CRM y la plataforma VDoc se realiza por medio de una base de datos y que la conexión de la base de datos con la plataforma VDoc queda fuera del alcance de este proyecto.

De este modo, teniendo en cuenta estos tres factores: tipo, forma de acceso y flujo de información se decidió incluir las dos primeras funcionalidades, véase *creación automática de listas de marketing en el CRM y sincronización de contactos entre el VDoc y el CRM*, en una misma aplicación informática debido a que utilizan información común sobre los clientes, el sentido del flujo de la información es el mismo y la forma de acceder a la información por el usuario no es determinante para la segunda funcionalidad. Además, como era necesaria la interacción humana con la aplicación informática y se realizaba para el ámbito local de Ibermática en San Sebastián se decidió que la mejor opción era crear una **aplicación de Windows** para ambas funcionalidades. A esta aplicación se la denominó **aplicación CRM-Marketing**.

En el caso de la última funcionalidad, véase *desencadenamiento del flujo de registro de ventas en el VDoc desde el CRM*, se decidió que la mejor opción era añadir un **plug-in** a Microsoft Dynamics CRM Online de Ibermática que desencadenará el flujo de ventas automáticamente sin interacción humana desde el CRM hacia la plataforma VDoc. A este plug-in se le denominó **plug-in de ventas**.

Hay que tener en cuenta que la toma de las decisiones para el diseño de las aplicaciones se compaginó con la implementación de las aplicaciones puesto que, era necesario adaptarse a las posibilidades de las tecnologías utilizadas sobre las que no se tenía un conocimiento previo.

Además, las tecnologías utilizadas en el desarrollo de las aplicaciones, sobre todo el trabajar con el SDK de Microsoft Dynamics 365, supeditaron en buena parte la definición requerimientos funcionales y no funcionales de las aplicaciones.

Finalmente, fruto del análisis realizado de los requisitos para ambas aplicaciones y teniendo en cuenta los condicionantes de las tecnologías utilizadas se definieron los **requisitos funcionales** y **no funcionales** tanto para la aplicación CRM-Marketing como para el plug-in de ventas.

## 4.1 Requisitos funcionales

---

Los requisitos funcionales definen las funciones o servicios que el sistema debe proporcionar, describen las características que debe poseer, la forma de reaccionar ante determinados inputs y cómo comportarse ante situaciones particulares, es decir, describen lo que el sistema debería hacer.

### 4.1.1 *Requisitos funcionales de la aplicación CRM-Marketing*

- La aplicación deberá permitir el acceso a las funcionalidades de la aplicación solamente a los usuarios autorizados que se hayan identificado y autenticado (nombre usuario y contraseña).
- La aplicación deberá permitir al usuario acceder a cualquiera de las organizaciones (instancias del CRM) a las que pertenezca.
- La aplicación deberá permitir al usuario seleccionar una lista concreta del conjunto de listas de mailing disponibles.
- La aplicación deberá alertar al usuario si se produce un error en la conexión con el servidor del CRM.
- La aplicación deberá alertar al usuario si se produce un error en la conexión con la base de datos.
- La aplicación permitirá al usuario crear una lista de marketing en el CRM a partir de la lista de mailing seleccionada (funcionalidad uno).
- La aplicación permitirá al usuario sincronizar los clientes entre la plataforma VDoc y el CRM (funcionalidad dos).
- La aplicación deberá mostrar por pantalla mensajes informativos que ayuden al uso de la aplicación, mensajes que muestren un resumen del resultado obtenido al ejecutar las funcionalidades y mensajes de error para facilitar identificar el error acaecido.
- La aplicación pondrá a disposición del usuario la opción de crear un archivo de log con la información que se ha mostrado por pantalla.
- El lenguaje que utilizará la aplicación para interactuar con el usuario será el inglés, incluyendo el texto mostrado por pantalla y el escrito en el archivo de log.

#### 4.1.2 *Requisitos funcionales del plug-in de ventas*

- El plug-in deberá ejecutarse cuando una oportunidad (entidad) en el CRM se cierre como ganada.
- El plug-in al ejecutarse deberá recopilar los datos de la venta realizada de las secciones *General* e *Importes* del panel principal de oportunidades del CRM.
- El plug-in deberá transmitir toda la información recopilada a una base de datos por medio de un servicio WCF.

## 4.2 **Requisitos no funcionales**

---

Los requisitos no funcionales definen los criterios que se utilizan para valorar las funciones o servicios del sistema y establecen restricciones y/o marcan objetivos de rendimiento, disponibilidad, diseño, seguridad, etc.

#### 4.2.1 *Requisitos no funcionales de la aplicación CRM-Marketing*

- La aplicación será una aplicación ejecutable en el Sistema Operativo de Windows.
- La interacción con la interfaz gráfica deberá ser lo más simple, clara y fácil de usar posible. El funcionamiento de la interfaz deberá dirigir al usuario en el proceso ejecución de las funcionalidades para disminuir la probabilidad de cometer errores.
- La aplicación se desarrollará con Visual Studio 2015, utilizando el kit de desarrollo de Microsoft Dynamics 365 y se programará en el lenguaje C#.

#### 4.2.2 *Requisitos no funcionales del plug-in de ventas*

- El plug-in deberá registrarse en un entorno aislado (sandbox).
- La aplicación se desarrollará con Visual Studio 2015, utilizando el kit de desarrollo de Microsoft Dynamics 365 y se programará en el lenguaje C#.

### 4.3 Diagrama de casos de uso

Un caso de uso describe la interacción entre los usuarios (actores) y el sistema. La aplicación CRM-Marketing cuenta con los siguientes casos de uso:

- **Inicializar Aplicación:** Este caso de uso incluye desde que el usuario ejecuta el archivo ejecutable de la aplicación hasta que se conecta al servidor del CRM y a la base de datos.
- **Crear LM:** Se corresponde con la primera funcionalidad que consiste en crear una lista de marketing en el CRM.
- **Actualizar Contactos:** Se corresponde con la segunda funcionalidad que consiste en la sincronización de los contactos de clientes entre el VDoc y el CRM
- **Añadir Credenciales:** Define el proceso a seguir para almacenar las credenciales del usuario en el sistema operativo de Windows y añadir una configuración del servidor del CRM en el archivo Credentials.xml.

En este apartado se va mostrar el diagrama de casos de uso para la aplicación CRM-Marketing.

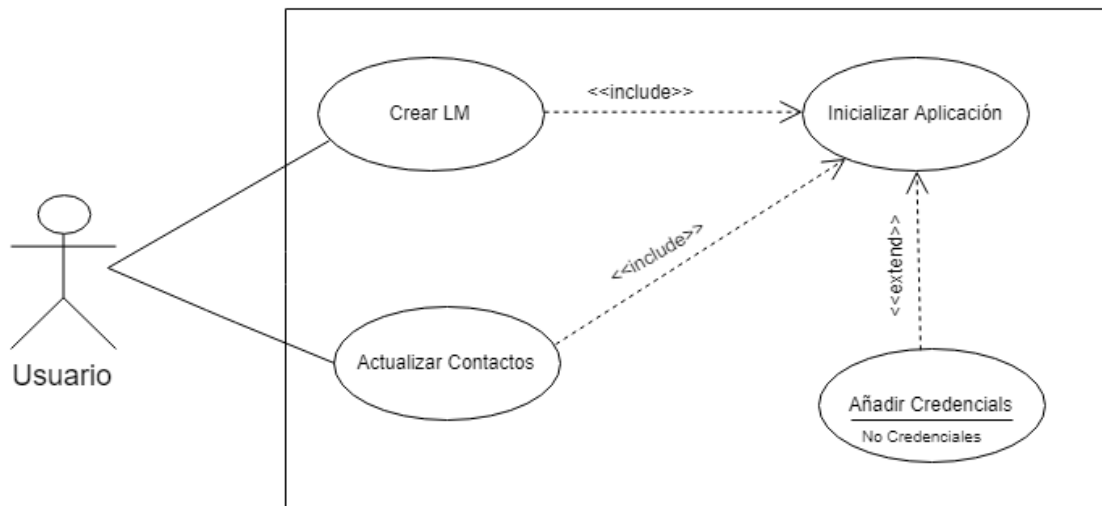


Figura 4.1: Diagrama de casos de uso de la aplicación CRM-Marketing

El caso del plug-in de ventas es un poco especial porque se compone de una sola funcionalidad y no hay interacción con el usuario por lo que no se ha realizado ningún diagrama de casos de uso ni los flujos de eventos correspondientes.

## 4.4 Flujo de eventos

El flujo de eventos de un caso de uso describe el comportamiento del sistema y muestra los pasos que los actores y el sistema van realizando sucesivamente en el contexto del requisito funcional capturado para dicho caso. El flujo de eventos consta del flujo principal y los flujos alternativos. El flujo principal describe el comportamiento ideal del sistema, aquí los actores y el sistema siguen el camino normal diseñado para cumplir una función del sistema. El flujo alternativo describe desviaciones del flujo principal ante situaciones de error o comportamientos inesperados del sistema.

A continuación se mostrarán los flujos de eventos para los cuatro casos de uso de la aplicación CRM-Marketing.

Nombre	INICIALIZAR APLICACIÓN
Actores	Usuario.
Descripción	Se arranca la aplicación y se produce la conexión al servidor del CRM y BD.
Precondición	Las credenciales del usuario están almacenadas en el SO de Windows. Las configuraciones de los servidores CRM están guardadas en el archivo Credentials.xml.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario hace doble click en el archivo ejecutable para arrancar la aplicación.</li> <li>2. La aplicación (sistema) muestra la pantalla principal.</li> <li>3. La aplicación identifica y autentica al usuario, valida las configuraciones de los servidores y carga todas las organizaciones (instancias del CRM) a las que pertenece el usuario.</li> <li>4. El usuario selecciona una organización (instancia del CRM).</li> <li>5. El usuario pulsa el botón [ConnectCRM] para conectarse al servidor del CRM de la organización seleccionada.</li> <li>6. La aplicación se conecta al servidor de CRM de la organización seleccionada, muestra un Led verde para indicar que la conexión es correcta y habilita el botón [ConnectDB].</li> <li>7. El usuario pulsa el botón [ConnectDB] para conectarse a la base de datos.</li> <li>8. La aplicación se conecta a la Base de Datos y carga un listado de Listas de Mailing. Además muestra un Led verde para indicar que la conexión es correcta y habilita el botón [Run] y [Update Contacts].</li> </ol>
Flujo alternativo	<ol style="list-style-type: none"> <li>3. A) Si la aplicación no puede identificar / autentificar al usuario o la configuración es incorrecta se mostrará un mensaje de error y se habilitará el botón [Add Credentials]. EXTEND &gt; AÑADIR CREDENCIALES</li> <li>6. A) Si se produce un error al conectarse al servidor del CRM, la aplicación mostrará un Led rojo para indicar que la conexión es incorrecta y un mensaje de error.</li> <li>8. A) Si se produce un error al conectarse a la base de datos, la aplicación mostrará un Led rojo para indicar que la conexión es incorrecta y un mensaje de error.</li> </ol>
Postcondición	Ninguna.

Tabla 4.1: Flujo de eventos de "Inicializar Aplicación"



<b>Nombre</b>	<b>CREAR LM</b>
<b>Actores</b>	Usuario.
<b>Descripción</b>	Crear una lista de marketing en el CRM.
<b>Precondición</b>	La aplicación se ha arrancado correctamente y el usuario se ha conectado al servidor del CRM y base de datos.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. INCLUYE INICIALIZAR APLICACIÓN.</li> <li>2. El usuario selecciona la Lista de Mailing que desea crear en el CRM.</li> <li>3. El usuario marca la opción <i>Create a Log File</i> si desea generar un archivo Log de texto.</li> <li>4. El usuario pulsa el botón de [Run] para crear la Lista de Marketing en el CRM.</li> <li>5. La aplicación pide confirmación para la creación de la Lista de Marketing mediante un cuadro de diálogo.</li> <li>6. El usuario pulsa OK y confirma la creación de la lista de marketing.</li> <li>7. La aplicación crea la lista de marketing y muestra por pantalla el resultado obtenido al ejecutar la funcionalidad.</li> <li>8. Si el usuario ha seleccionado la opción <i>Create a Log File</i> también creará el archivo Log y mostrará la ubicación de la carpeta que contiene el archivo.</li> </ol>
<b>Flujo alternativo</b>	6. A) Si el usuario NO confirma la creación de la lista de marketing (pulsa el botón [Cancel]) la lista de marketing NO se creará.
<b>Postcondición</b>	Se crea una lista de marketing en el CRM con los datos de la lista de mailing (VDoc) seleccionada. Si se ha seleccionado la opción correspondiente también se crea un archivo de log con información sobre la Lista de Marketing creada.

Tabla 4.2: Flujo de eventos de “Crear LM”

<b>Nombre</b>	<b>ACTUALIZAR CONTACTOS</b>
<b>Actores</b>	Usuario.
<b>Descripción</b>	Actualizar los contactos del CRM.
<b>Precondición</b>	La aplicación se ha arrancado correctamente y el usuario se ha conectado al servidor del CRM y base de datos.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. INCLUYE INICIALIZAR APLICACIÓN.</li> <li>2. El usuario marca la opción <i>Create a Log File</i> si desea generar un archivo Log de texto.</li> <li>3. El usuario pulsa el botón de [Update Contacts] para actualizar los contactos del CRM.</li> <li>4. La aplicación pide confirmación para la actualización de los contactos mediante un cuadro de diálogo.</li> <li>5. El usuario pulsa OK y confirma la actualización de los contactos en el CRM.</li> <li>6. La aplicación actualiza los contactos y muestra por pantalla el resultado obtenido al ejecutar la funcionalidad.</li> <li>7. Si el usuario ha seleccionado la opción <i>Create a Log File</i> también creará el archivo Log y mostrará la ubicación de la carpeta que contiene el archivo.</li> </ol>
<b>Flujo alternativo</b>	5. A) Si el usuario NO confirma la actualización de los contactos (pulsa el botón [Cancel]) los contactos NO se actualizan en el CRM.
<b>Postcondición</b>	Se actualizan los contactos en el CRM con los datos de la base de datos (VDoc). Si se ha seleccionado la opción <i>Create a Log File</i> también se crea un archivo de log con información sobre los contactos actualizados.

Tabla 4.3: Flujo de eventos de “Actualizar Contactos”

<b>Nombre</b>	<b>AÑADIR CREDENCIALES</b>
<b>Actores</b>	Usuario.
<b>Descripción</b>	Almacenar las credenciales del usuario en el SO de Windows y guardar las configuraciones de los servidores CRM en el archivo Credentials.xml.
<b>Precondición</b>	Se ha producido un error al identificar y/o autenticar las credenciales del usuario bien porque no están almacenadas en el SO de Windows o porque algún valor es erróneo.
<b>Flujo principal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón [Add Credentials].</li> <li>2. La aplicación (sistema) muestra la ventana “CRM user credentials”.</li> <li>3. El usuario introduce su nombre de usuario (username) y la contraseña (password) y pulsa OK.</li> <li>4. La aplicación identifica y autentica al usuario, almacena las credenciales en Windows, guarda las configuraciones de los servidores CRM a los que pertenece en el archivo Credentials.xml.</li> <li>5. La ventana “CRM user credentials” se cierra, se oculta el botón [Add Credentials] y se muestra un mensaje confirmando que las credenciales del usuario se han almacenado correctamente.</li> </ol>
<b>Flujo alternativo</b>	<ol style="list-style-type: none"> <li>4. A) Si la aplicación no puede identificar / autenticar al usuario se vuelve a la ventana principal y se muestra un mensaje (de error) para indicar que la verificación de las credenciales ha sido errónea y que no se han almacenado en el SO de Windows.</li> </ol>
<b>Postcondición</b>	Se almacenan las credenciales del usuario en el SO de Windows como una credencial genérica y se crea (actualiza) el archivo Credentials.xml en el disco local donde se almacenan las configuraciones de los servidores del CRM a los que pertenece el usuario.

Tabla 4.4: Flujo de eventos de “Añadir Credenciales”

# 5

---

---

## **5. *Diseño***

En este capítulo, se va a explicar la arquitectura de programación en tres capas como modelo de desarrollo del software utilizado para implementar la aplicación CRM-Marketing.

Después, se mostrará el modelo de datos del CRM centrándonos en los datos de las entidades a las que acceden las aplicaciones. Una vez conocido el modelo del dominio diseñaremos las bases de datos. Finalmente, se incluye el diseño gráfico de las interfaces.

## 5.1 Arquitectura en tres Capas

---

La aplicación CRM-Marketing se decidió implementar siguiendo el modelo de desarrollo software de programación por capas. [2]

“El estilo arquitectural en n capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan.”<sup>6</sup>

Al referirse a la arquitectura de una aplicación, es muy importante distinguir la diferencia entre los conceptos de capas (layers) y de niveles (tiers). El término “capa” hace referencia a la separación lógica de los componentes y de las funcionalidades mientras que el término “nivel” se refiere a la distribución física de las capas lógicas en diferentes servidores o lugares. En nuestro caso, nos estamos refiriendo siempre a la separación lógica de las capas de la aplicación.

La principal ventaja de este modelo de desarrollo software es el aislamiento entre capas que permite realizar modificaciones en los componentes software de una capa sin que sea vea afectado el código del resto de las capas.

El diseño más comúnmente utilizado y el que se adoptó para desarrollar la aplicación CRM-Marketing es la **programación en tres capas**:

- Capa de presentación
- Capa de negocio
- Capa de datos



Figura 5.1: Modelo de programación en tres capas

---

<sup>6</sup>César de la Torre Llorente, Unai Zorrilla Castro, Miguel Ángel Barros, Javier Calvario Nelson. Guía de arquitectura en N capas orientadas al dominio con Net 4.0 impreso en España.

Los componentes software de una capa se comunican solamente con los componentes de otra capa anexa, sea inferior y superior, por ejemplo, la capa de presentación se comunica directamente con la capa de negocio pero no con la capa de datos.

### 5.1.1 *Capa de presentación*

La capa de presentación, también conocida como interfaz gráfica, es la encargada de interactuar con el usuario y responsable de la presentación visual de la aplicación. Es la capa que utiliza el usuario para comunicarse con la aplicación, donde se le presenta la información, recibe los parámetros de entrada del usuario y le presenta los resultados obtenidos. Mediante la capa de presentación se separa la interacción del usuario respecto a la lógica de negocio.

En nuestra solución para la aplicación CRM-Marketing, la capa de presentación se compone de las vistas que se van a presentar al usuario final y que se desarrollan mediante Formularios de Windows.

### 5.1.2 *Capa de negocio*

La capa de negocio, también conocida lógica de negocio, es una capa intermedia que implementa las reglas de negocios que debe cumplir la aplicación. Esta capa se comunica con la capa presentación, para recibir las solicitudes del usuario y devolver los resultados, y con la capa de negocio, para enviar las solicitudes de almacenamiento o recuperación de información.

### 5.1.3 *Capa de datos*

La capa de datos, la última capa del modelo de programación en 3 capas, es la encargada de acceder a los datos. Esta es la capa donde se implementan las conexiones al servidor y a la base de datos, y la que recibe las solicitudes de la capa de negocios para la recuperación y almacenamiento de la información. Está compuesta por uno o más gestores de bases de datos y actúa como puente entre la capa lógica de negocio y el proveedor de datos.

En el presente proyecto, la capa de datos deberá acceder, por una parte, a una base de datos de SQL Server que está alojada en un servidor local de la compañía y por otra parte, a la base de datos del CRM de Microsoft Dynamics 365 que se encuentra alojada en un servidor externo y que es solamente accesible a través de los servicios web que proporciona el CRM.

## 5.2 Modelo de datos de Microsoft Dynamics CRM Online

---

En el capítulo 2.2 donde se hace una pequeña introducción a Microsoft Dynamics CRM Online, ya se comenta que el modelo de datos del CRM sigue un patrón similar al de una base de datos relacional. Así, a cada entidad del CRM le correspondería una tabla y a cada atributo de una entidad le correspondería un campo de la tabla. Las entidades mantienen relaciones de integridad del tipo 1:N, N:1 y N:N. Los atributos de las entidades, por su parte, tienen tipos de datos específicos del CRM.

En este momento, es necesario puntualizar que no se tiene acceso directo al modelo de datos del CRM y que la representación de este modelo en parte o en su conjunto se debe realizar aplicando **ingeniería inversa**, es decir, analizando las relaciones entre entidades y sus atributos. Así, debido a la gran cantidad de entidades que tiene el CRM, alrededor de 100, incluso la solución “CRM IB Industria” que es la solución específica para Ibermática y que consta de 32 entidades, resulta inviable mostrar el modelo completo de datos del CRM.

De este modo, nos centraremos en la representación de la parte del modelo de datos del CRM cuyas entidades intervienen en la **aplicación CRM-Marketing** y/o en el **plug-in de ventas**.

### 5.2.1 Modelo de datos de la aplicación CRM-Marketing

La aplicación CRM-Marketing accede a los datos de las siguientes entidades del CRM:

- **Lista de marketing:** es un listado de contactos o clientes potenciales filtrados siguiendo criterios comerciales y que es la base para ejecutar campañas de marketing.
- **Contacto:** son los clientes de nuestra compañía. Incluye toda la información sobre una persona, como el email, los números de teléfono, la dirección postal e información de tipo personal.
- **Cuenta:** representa una compañía con la que la unidad de negocio tiene una relación comercial. Incluye información sobre la compañía, información de contacto, categoría, tipo de relación, etc.

En la siguiente figura 5.2 se muestra el modelo de datos de la aplicación:

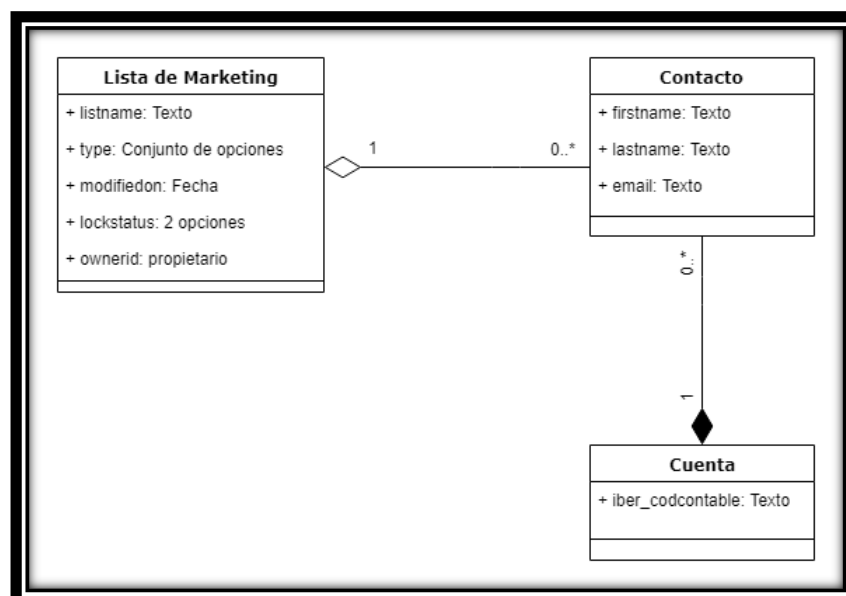


Figura 5.2: Modelo de datos aplicación CRM-Marketing

### 5.2.2 Modelo de datos del plug-in de ventas

El plug-in de ventas accede a los datos de las siguientes entidades del CRM:

- **Oportunidad:** se refiere a una oportunidad de negocio, es decir, una posible venta a clientes nuevos o ya existentes.
- **Importes:** son los productos o servicios que se quieren vender al cliente, incluye los datos del producto, cantidad, precio e ingresos.
- **Cuenta:** es la compañía o empresa interesada en el producto o servicio que está ofreciendo Ibermática.
- **Usuario:** se refiere al usuario del sistema que ha creado la oportunidad. El CRM asigna automática a este usuario como propietario de la oportunidad.
- **Área:** se refiere al área de negocio correspondiente al producto que se está ofertando al cliente. Las áreas de negocio más importantes de Ibermática son: ING (Ingeniería), ERP (Enterprise Resource Planning) y MES (Manufacturing Execution Systems).

En la siguiente figura 5.3 se muestra el modelo de datos del plug-in de ventas:

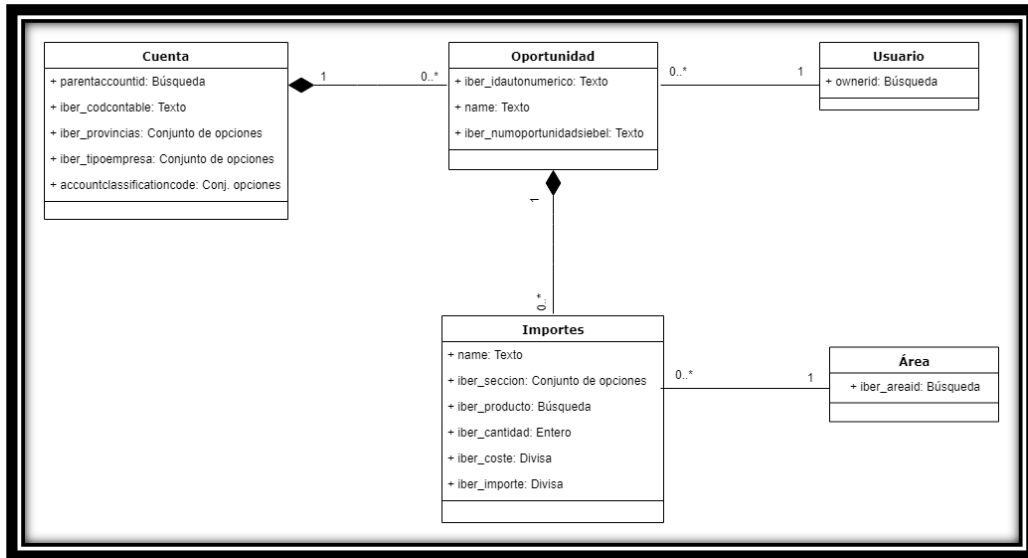


Figura 5.3: Modelo de datos plug-in de ventas

### 5.3 Diseño de las bases de datos

La información de los clientes que el CRM de Microsoft Dynamics 365 y la plataforma de soporte VDoc comparten se va a almacenar en bases de datos relacionales localizadas en servidores locales de Ibermática.

#### 5.3.1 Base de datos de la aplicación CRM-Marketing

El dominio de datos con el que trabaja la aplicación CRM-Marketing solamente consta de tres entidades del CRM y, por tanto, el diseño de la base de datos es simple. Además, los campos de la entidad *Lista de Marketing*: type, modifiedon, lockstatus y ownerid toman valores por defecto al crearse una lista de marketing en el CRM y no es necesario almacenarlos en la base de datos.

Así, el esquema de la base de datos para la aplicación CRM-Marketing solamente consta de dos tablas independientes entre sí, una para cada funcionalidad de la aplicación.

- Tabla LISTA\_MAIL: almacena los campos necesarios para crear una lista de marketing en el CRM.
- Tabla CONTACTOS: almacena los campos necesarios para actualizar los contactos en el CRM.

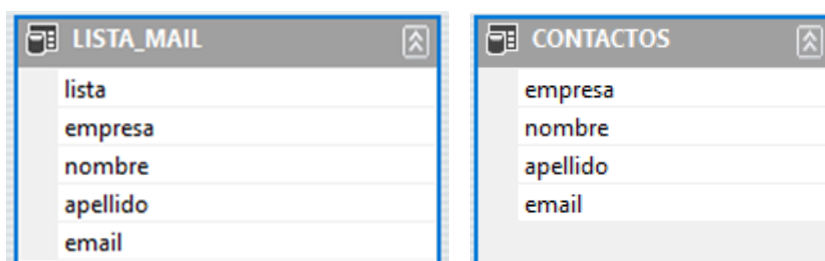


Figura 5.4: Esquema BD aplicación CRM-Marketing



Los datos para ambas tablas son los mismos menos el campo *lista*. El campo *lista* es el nombre de la lista de marketing a crear. Los campos *nombre*, *apellido* y *email* corresponden a los contactos. El campo *email* se utiliza como clave principal para los contactos en el CRM, no pueden existir dos contactos diferentes con el mismo email.

El campo **empresa** es un campo muy importante que se refiere al código SAP de las empresas clientes. Éste es el único código identificativo que es común tanto en el CRM como en VDoc. Es el campo *iber\_codcontable* de la entidad *cuenta* del CRM.

En el caso de la tabla de contactos, solamente se recogen algunos atributos para poder probar la funcionalidad correspondiente. En el futuro, se deberá hacer estudio para definir todos los campos necesarios.

### 5.3.2 Base de datos del plug-in de ventas

El dominio de datos con el que trabaja el plug-in es algo más amplio que el expuesto para la aplicación CRM-Marketing. Además, en este caso se van a utilizar todos los campos del modelo de datos del CRM para el plug-in de ventas.

De este modo, se van a crear dos tablas relacionadas entre sí: *Oportunidades* e *Importes*. La tabla *Oportunidades* almacena los datos de los campos provenientes de las entidades *Oportunidad*, *Cuentas* y *Usuario* y la tabla *Importes* almacena los datos de las entidades *Importes* y *Área*. La relación de integridad entre ambas tablas se establece entre los campos *numero* y *numoport* que representan al campo *iber\_idautonumerico* de la entidad *Oportunidad*.

En la siguiente figura 5.5 se muestra el esquema de la base de datos resultante para el plug-in de ventas.

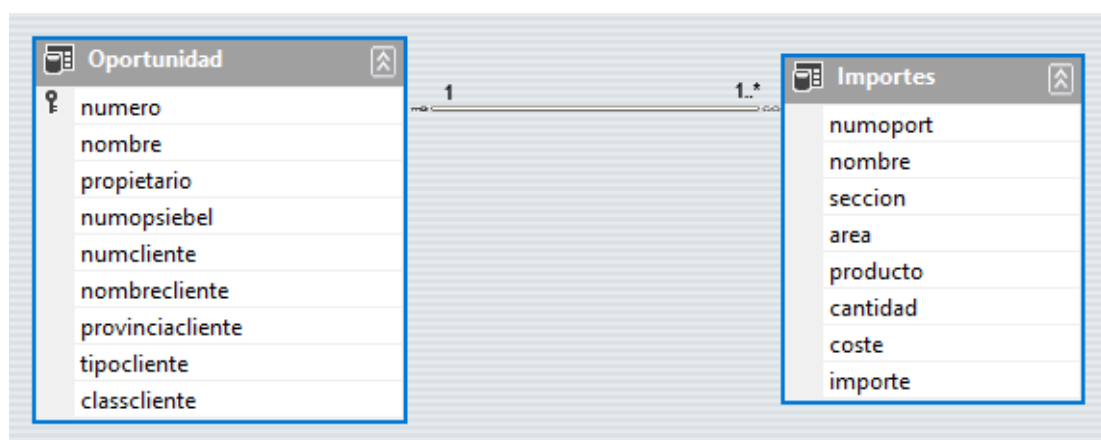


Figura 5.5: Esquema BD plug-in de ventas

## 5.4 Diseño gráfico de las interfaces

---

En este proyecto, se optó por una capa de presentación simple, clara y fácil de usar, donde la mayor relevancia se da a las funcionalidades de la aplicación y en menor medida al diseño gráfico de la misma.

La aplicación CRM-Marketing se compone de dos vistas que se implementan mediante formularios de Windows:

- MainMenu Form.
- UserLogonCredentials Form.

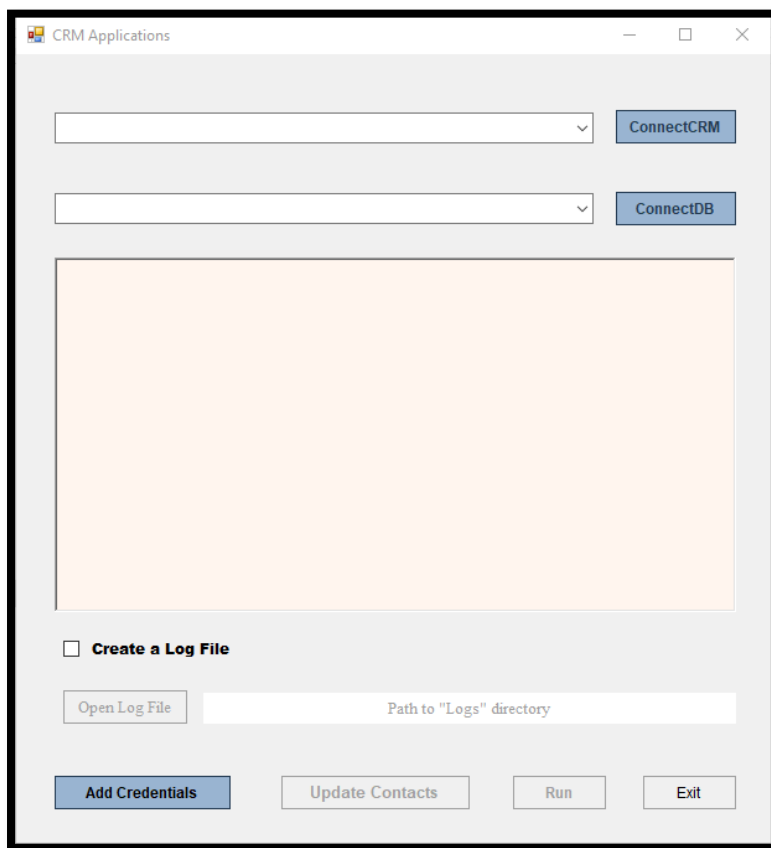
La ventana principal (MainMenu Form) es donde se va realizar la mayor parte de las interacciones con los usuarios y desde donde el usuario accede a la ejecución de las dos funcionalidades desarrolladas para la aplicación:

- Creación de listas de marketing (mailing).
- Sincronización de los contactos del VDoc y CRM.

El diseño y desarrollo de la vista del Menú Principal se empieza con un prototipo que se va refinando durante el transcurso de proyecto. Los elementos que conforman esta vista son:

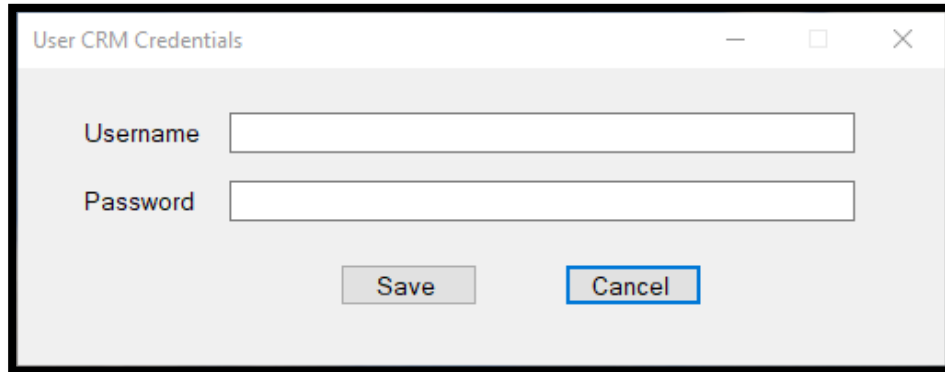
1. Lista desplegable de las soluciones de CRM disponibles al usuario.
2. Botón [ConnectCRM] de conexión al servidor del CRM.
3. Lista desplegable de listas de mailing.
4. Botón [ConnectDB] de conexión a la base de datos.
5. Panel de texto informativo para mostrar los resultados obtenidos.
6. Casilla de verificación para la creación del archivo Log.
7. Botón [OpenLogFile] para abrir Archivo Log.
8. Etiqueta con enlace al directorio temporal donde se almacenan los archivos Log.
9. Botón [Add Credentials] para añadir los credenciales del usuario.
10. Botón [Update Contacts] para actualizar los contactos.
11. Botón [Run] para crear la lista de marketing.
12. Botón [Exit] para abandonar la aplicación.
13. Leds verdes (2) y rojos (2) para indicar si las conexiones al servidor del CRM y la base de datos se han realizado correctamente.

En la figura 5.6, se muestra el resultado final de la ventana de inicio.



*Figura 5.6: Diseño ventana principal aplicación CRM-Marketing*

La segunda vista “User CRM Credentials” que corresponde al formulario *UserLogonCredentials* es similar a una vista típica de login. Sin embargo, esta vista se crea para implementar una nueva funcionalidad: almacenar las credenciales del usuario (nombre de usuario y contraseña) en el sistema operativo de Windows.



The image shows a Windows-style dialog box titled "User CRM Credentials". It has a light gray background and a white title bar with the text "User CRM Credentials" and standard window controls (minimize, maximize, close). The dialog contains two input fields: "Username" and "Password". Below the fields are two buttons: "Save" and "Cancel".

*Figura 5.7: Diseño ventana “User CRM Credentials”*

# 6

---

---

## ***6. Tecnologías y Entorno de Desarrollo***

Las aplicaciones se han implementado por medio del **framework .NET** [\[3\]](#) que es un framework desarrollado por Microsoft Corporation para mejorar el desarrollo y ejecución de aplicaciones Windows. El framework .NET surge como alternativa estratégica de Microsoft al panorama abierto por Java y otras plataformas de software libre. Su objetivo principal es facilitar el desarrollo de aplicaciones en entornos web.

En este capítulo primeramente se va a hacer una introducción al framework .NET de Microsoft. A continuación, se van a explicar las tecnologías relacionadas con dicho framework, luego, se detallarán las herramientas software utilizadas y, finalmente, se hablará del kit de desarrollo software (SDK) para Microsoft Dynamics 365.

## 6.1 ¿Qué es .NET?

---

Microsoft .NET [3] es un framework de código abierto, multi-plataforma y gratuito que permite el desarrollo y ejecución de aplicaciones. .NET ofrece un entorno gestionado de ejecución de aplicaciones, lenguajes de programación y compiladores, que permite crear muchos tipos de aplicaciones: desde las tradicionales aplicaciones de consola o aplicaciones de escritorio, pasando por aplicaciones y servicios web, hasta aplicaciones para dispositivos móviles, juegos o desarrollos para Internet.

Los objetivos<sup>7</sup> del framework .NET se pueden resumir en los siguientes puntos:

- Proporcionar soluciones prefabricadas para problemas comunes de desarrollo de software.
  - Desarrollo de interfaces.
  - Acceso a datos.
  - Controles, transacciones y gestión de memoria automáticas.
- Entorno común de desarrollo multilinguaje.
  - Posibilitar la integración de software desarrollado con distintos lenguajes de programación. El framework .NET permite utilizar cualquiera de los siguientes lenguajes: C #, C ++, Visual Basic, Jscript, Perl, Python, COBOL, etc.
- Crear un entorno visual para desarrollar e implantar rápidamente servicios y aplicaciones Web.
- Combinar un paradigma de programación fácil de usar con los protocolos abiertos de Internet.
  - Integración a través de estándares de Internet públicos.
  - Integración de aspectos de seguridad.

Los principales componentes del framework .NET son:

- **Common Language Runtime (CLR):** Es el entorno común de ejecución de lenguajes, donde se ejecuta el código de las aplicaciones. Es el corazón del framework .NET y la implementación del CLI (Common Language Infrastructure) de Microsoft.
- **Class Library:** o biblioteca de clases unificada. Se divide en dos partes principales.
  - Base Class Library (BCL) o biblioteca de clases base: Es la biblioteca del entorno de ejecución (runtime) que controla las operaciones básicas que se usan en el desarrollo de las aplicaciones.
  - Framework Class Library (FCL) o biblioteca de clases del framework: Es la implementación de Microsoft .NET de las bibliotecas estándar tal y como se define en la infraestructura del lenguaje común (CLI).
- **El conjunto de lenguajes de programación,** junto con sus compiladores y “linkers”.

---

<sup>7</sup>Vadillo Zorita, José Ángel. Facultad de Informática de San Sebastián (UPV/EHU). Asignatura HADS 2017.

## 6.2 Tecnologías

---

Durante la implementación de las aplicaciones se ha empleado una gran variedad de las tecnologías que incluye el framework .NET.

### 6.2.1 Formularios de Windows

“Los Formularios Windows (Windows Forms) es el nombre dado a la interfaz de programación de aplicación gráfica (API) que se incluye como parte de Microsoft .NET Framework”<sup>8</sup>.

Los Formularios Windows se han utilizado para desarrollar la **capa de presentación** de la aplicación CRM-Marketing, es decir, la interfaz gráfica que permite al usuario interactuar con la aplicación. Sin embargo, como en el caso del plug-in de ventas no hay interacción humana no se ha utilizado esta tecnología.

### 6.2.2 C#

“C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft como parte de su plataforma .NET. Es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes”<sup>9</sup>.

C# es un lenguaje de programación que está optimizado para ser usado en el framework .NET. Así, C# es el lenguaje base para escribir aplicaciones en .NET y es el primer lenguaje en importancia para el CLR (Common Language Runtime).

Ibermática establece C# como lenguaje principal de programación para implementar aplicaciones que se vayan a desarrollar mediante el framework .NET. Por este motivo, C# ha sido el lenguaje de programación que se ha utilizado para la implementación de la **lógica de negocio** de las aplicaciones, tanto para la aplicación CRM-Marketing como para el plug-in de ventas.

---

<sup>8</sup>Wikipedia. Windows Forms: [https://es.wikipedia.org/wiki/Windows\\_Forms](https://es.wikipedia.org/wiki/Windows_Forms)

<sup>9</sup>Wikipedia. Windows C#: [https://es.wikipedia.org/wiki/C\\_Sharp](https://es.wikipedia.org/wiki/C_Sharp)

### 6.2.3 WCF (*Windows Communication Foundation*)

WCF (Windows Communication Foundation) [4] es un entorno de desarrollo y conjunto de APIs que permiten crear aplicaciones orientadas a servicios (SOA). A su vez, la arquitectura orientada a servicios (SOA) depende de estos servicios web para enviar y recibir datos.

WCF es una plataforma que posibilita una comunicación basada en mensajes que se envían entre los clientes y los servicios. Los clientes son las aplicaciones que inician la comunicación y los servicios son las aplicaciones que se mantienen a la espera y que responden a las comunicaciones de los clientes. Los mensajes se envían entre los puntos de conexión (endpoints) que es donde se define toda la información necesaria para el intercambio de mensajes.

En este proyecto se ha utilizado esta tecnología para desarrollar un servicio WCF para el plug-in de ventas.

### 6.2.4 *Transact-SQL*

Transact-SQL (T-SQL), extensión al SQL de Microsoft, es el lenguaje que se ha utilizado para interactuar con las bases de datos de las aplicaciones. Transact-SQL es fundamental para poder utilizar Microsoft SQL Server.

## 6.3 Herramientas software

---

Las dos principales herramientas software que se han utilizado para el desarrollo de este proyecto son:

- Microsoft Visual Studio
- Microsoft SQL Server

### 6.3.1 *Microsoft Visual Studio 2015*

“Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta múltiples lenguajes de programación como C#, C++, Visual Basic, Java, Python, Ruby, PHP o F#. Visual Studio permite a los desarrolladores crear aplicaciones de escritorio, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros”.<sup>10</sup>

---

<sup>10</sup>Wikipedia. Microsoft Visual Studio: [https://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://es.wikipedia.org/wiki/Microsoft_Visual_Studio)



Microsoft Visual Studio 2015 es la principal herramienta que se ha empleado para la implementación de las aplicaciones y, en concreto, de la capa de presentación (interfaces gráficas) y la lógica de negocio.

### 6.3.2 *NuGet*

NuGet es un gestor de paquetes y software libre diseñado para el framework de Microsoft .NET y que permite a los desarrolladores crear, consumir, compartir y reutilizar código público. De esta forma, permite agregar, actualizar y eliminar librerías y herramientas externas a los proyectos de Visual Studio.

### 6.3.3 *Kit de desarrollo software (SDK) para Microsoft Dynamics 365*

Este kit de desarrollo de software [5] proporciona un conjunto de bibliotecas y herramientas que permite a los desarrolladores la creación de aplicaciones para Microsoft Dynamics 365. El SDK contiene una visión del **conjunto de la arquitectura de Microsoft Dynamics 365**, el modelo de entidades, el modelo de seguridad, los servicios web y ejemplos de muestra de código.

El kit permite desarrollar múltiples tipos de aplicaciones tanto en el framework de Microsoft .NET como en otras plataformas. Las aplicaciones pueden ser clientes de Windows o no y también permite desarrollar otros elementos de software como plug-ins, actividades de flujo de trabajo o flujos de trabajo XAML personalizados que se ejecutan automáticamente sin necesidad de la interacción humana.

Microsoft Dynamics CRM Online no posibilita el acceso directo a la base de datos SQL y para la interacción con los datos es necesario utilizar los servicios web que se encuentran en el kit de desarrollo software para Microsoft Dynamics 365. Así, este SDK es necesario para poder acceder a los datos (registros de entidades) y metadatos (entidades, atributos, relaciones, etc.) del CRM y por tanto, imprescindible para poder implementar aplicaciones para Microsoft Dynamics 365.

Nota: A partir de la actualización de diciembre de 2016 para Dynamics 365 (en línea y local), la funcionalidad de Dynamics CRM ahora se incluye como parte de Dynamics 365.

#### 6.3.4 *Microsoft SQL Server 2014*

Microsoft SQL Server 2014 es el gestor de bases de datos que se ha utilizado en este proyecto. Esta herramienta se ha utilizado para el desarrollo de las bases de datos de las aplicaciones. Transact-SQL es el lenguaje empleado para interactuar con SQL server.

# 7

---

---

## ***7. Aplicación CRM-Marketing***

En este capítulo, se muestra el proceso seguido para la implementación de la aplicación CRM-Marketing. Empieza explicando los servicios web necesarios para conectarse a Microsoft Dynamics CRM Online. A continuación, el punto de partida para empezar a implementar la aplicación y las clases más importantes para realizar la conexión con el CRM.

Finalmente, se describe la implementación de las 2 funcionalidades de la aplicación CRM-Marketing incluyendo fragmentos de código significativos que reflejan características relevantes de la aplicación. Las dos funcionalidades son:

1. Crear Listas de Marketing en el CRM
2. Sincronización de Contactos en el CRM.

## 7.1 Servicios Web de Microsoft Dynamics 365

---

Microsoft Dynamics CRM Online no posibilita el acceso directo a la base de datos SQL. De este modo, para la interacción con los datos es necesario utilizar los servicios web del CRM y los REST y SOAP endpoints que se encuentran en el SDK de Microsoft Dynamics 365.

Los servicios web [6] que el SDK de Microsoft Dynamics 365 pone a nuestra disposición proporcionan las APIs necesarias para programar para Microsoft Dynamics 365 (online y on-premises). En este apartado, se presentarán los servicios del SDK, se hará un estudio básico de las alternativas disponibles y se justificará el servicio web seleccionado para desarrollar las aplicaciones.

1. **Web API:** implementa el protocolo OData (Open Data Protocol) que cumple la norma OASIS y consume REST APIs. La Web API está construida en estándares abiertos y no requiere usar librerías específicas de cliente para programar para Microsoft Dynamics 365.
2. **Organization Service:** es también conocido como “SOAP endpoint”. Es el servicio web más utilizado por los desarrolladores que trabajan con Microsoft Dynamics 365. Este servicio está optimizado para ser usado con .NET.
3. **Organization Data Service:** es también conocido como “ODATA” o “REST” endpoint. Es un servicio web obsoleto.
4. **Discovery Web Service:** Microsoft Dynamics 365 es una aplicación “multi-inquilino”. Eso significa que una sola instalación de Microsoft Dynamics 365 puede albergar varias instancias del CRM en varios servidores. Este servicio permite determinar las organizaciones a las que pertenece un usuario determinado.
5. **Deployment Web Service:** este servicio web se utiliza para crear, importar, o eliminar organizaciones así como aplicar ciertos ajustes en código.

El siguiente paso es analizar las cinco alternativas de servicios web disponibles con el SDK de Microsoft Dynamics 365 y decidir cuál es la mejor opción para desarrollar nuestras aplicaciones.

Desde el inicio se descartaron las opciones del Organization Data Service por ser un servicio obsoleto y del Deployment Web Service por ser un servicio muy específico que proporciona unas utilidades que no son necesarias para nuestras aplicaciones. Inicialmente, también se consideró que el Discovery Web Service ofrecía una funcionalidad muy limitada.

Por este motivo, el estudio más completo se dedicó a comparar las ventajas e inconvenientes de la Web API vs el Organization Service.

En general, se puede afirmar que la Web API proporciona una mejor compatibilidad en una gran variedad de lenguajes de programación, de plataformas y de dispositivos. Es una versión de servicio web más avanzado y no requiere el uso de librerías específicas para Microsoft Dynamics 365. Sin embargo, tiene el gran inconveniente de que este servicio no es muy indicado para programar código que se ejecuta sobre el servidor, como son los plug-ins o los montajes de flujos de trabajo.

Por otra parte, aunque el Organization Service es una versión de servicio de web más antigua está optimizado para su uso en el framework .NET que es la tecnología que se utiliza en Ibermática.

De este modo, la decisión final fue elegir el Organization Service como servicio web para desarrollar nuestras aplicaciones por las siguientes razones:

- Está optimizado para su uso en .NET que es el framework que se va a utilizar.
- La Web API es problemática si se quiere desarrollar un plug-in.
- Como es el servicio web más utilizado por los desarrolladores de Microsoft Dynamics 365 se prevé que será más fácil encontrar información.

Finalmente, también se va hacer uso del Discovery Web Service debido a que en Ibermática con la misma instalación del CRM de Microsoft Dynamics algunos usuarios tienen acceso a varias instancias del CRM. Las dos más significativas son:

1. **IBERMATICA S.A.:** Es la instancia del CRM que se utiliza en producción, es decir, la que está disponible para todos los trabajadores de Ibermática.
2. **Desarrollo Industria:** Esta instancia del CRM es la que se utiliza en el entorno de desarrollo (sandbox) y donde se van a implementar inicialmente las aplicaciones.

## 7.2 Conexión al servidor CRM

---

El primer reto que tenemos que afrontar para empezar a desarrollar las aplicaciones para Microsoft Dynamics 365 es cómo establecer la conexión con el CRM. La comunicación con Microsoft Dynamics CRM Online se produce principalmente gracias a la interacción con dos servicios web proporcionados por el Kit de desarrollo software (SDK) de Microsoft Dynamics 365: los llamados servicios web Discovery (**IDiscoveryService**) y Organization (**IOrganizationService**). [7]

Como ya se explicó en el apartado anterior, una sola instalación de Microsoft Dynamics CRM puede alojar múltiples organizaciones empresariales en varios servidores. Así, el servicio web Discovery (IDiscoveryService) se usa para obtener la(s) URL(s) que dan acceso

a las organizaciones a las que pertenece el usuario. A su vez, para acceder a los datos de una organización en particular a través del Servicio de Organización (IOrganizationService) tenemos que usar la URL especificada para esa organización.

La mayoría de las veces se trabaja con datos (registros de entidades) y metadatos (entidades, atributos, relaciones, etc.) de una organización, por tanto, el Servicio de Organización (IOrganizationService) es el servicio web primario con el que se va a trabajar, mientras que el servicio web Discovery (IDiscoveryService) es un servicio secundario que solamente se utiliza para obtener la URL correcta para el Servicio de Organización (IOrganizationService).

Por lo tanto, el proceso inicial de conexión podría dividirse en dos etapas:

1. Una aplicación cliente obtiene el nombre del servidor de CRM, el puerto (si corresponde), el protocolo (http o https) y las credenciales de usuario (nombre de usuario y contraseña) y envía la petición al IDiscoveryService. El servicio web IDiscoveryService determina las organizaciones a las que pertenece el usuario y envía la información sobre las organizaciones encontradas a la aplicación cliente incluyendo la(s) URL(s) para el servicio web IOrganizationService. La aplicación cliente elige una organización con la que interactuar, autentica las credenciales del usuario y obtiene la URL para el IOrganizationService correspondiente.
2. La aplicación cliente se conecta al IOrganizationService utilizando la URL que le ha proporcionado el IDiscoveryService. Ahora, el Servicio de Organización (IOrganizationService) ya puede administrar los datos y metadatos de la organización.

Después de la primera conexión, es muy recomendable almacenar en algún lugar la URL obtenida con el IOrganizationService para poder ser utilizada en posteriores ocasiones, ya que permite omitir la interacción con el servicio web IDiscoveryService.

Sin embargo, debemos tener en cuenta que la ubicación de los servidores y las organizaciones para una instalación concreta de Microsoft Dynamics CRM Online podría cambiar como parte de la administración del centro de datos o el balance de cargas. Por lo tanto, es posible que tengamos que actualizar la URL real del Servicio de Organización (IOrganizationService) cada cierto tiempo lo que implicaría tener que volver a utilizar el servicio web (IDiscoveryService) para volverla a obtener. [8]

### 7.2.1 Punto de partida

El SDK para Microsoft Dynamics 365 proporciona varios ejemplos de muestra. El ejemplo de inicio rápido (QuickStart) [9] desarrolla una aplicación de consola que muestra todos los pasos necesarios para conectarse al CRM y que sirve como **ejemplo base** para desarrollar posteriores aplicaciones de Microsoft Dynamics 365.

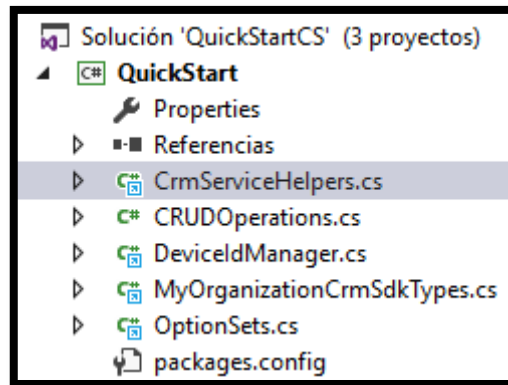


Figura 7.1: Archivos del programa QuickStart

La ejecución de la aplicación de consola implementada en la solución *QuickStart* sigue los pasos siguientes:

1. El usuario introduce el nombre del servidor del CRM, el puerto (si corresponde), el protocolo (http o https) y las credenciales de usuario (nombre de usuario y contraseña). El servicio web IDiscoveryService determina las organizaciones a las que pertenece el usuario y las muestra por pantalla.

```
Enter a CRM server name and port [crm.dynamics.com]: crm4.dynamics.com
Is this organization provisioned in Microsoft Office 365 (y/n) [y]: y

Enter Username: a.garcia.benito@ibermatica01.onmicrosoft.com
Enter Password: *****

List of organizations that you belong to:
(1) Ibermática DEMOS (ibermaticademos365)
(2) Desarrollo Industria (desindustria)
(3) IBERMATICA S.A (ibermatica01)

Specify an organization number (1-3) [1]:
```

Figura 7.2: Ejecución programa QuickStart (paso 1)

2. El usuario selecciona la organización a la que se quiere conectar, después el sistema autentica sus credenciales y se realiza el inicio de sesión en el CRM.

```
(1) Ibermática DEMOS (ibermaticademos365)
(2) Desarrollo Industria (desindustria)
(3) IBERMATICA S.A (ibermatica01)

Specify an organization number (1-3) [1]: 2
Logged on user is Alberto Garcia Benito.
Microsoft Dynamics CRM version 8.2.2.1659.
```

Figura 7.3: Ejecución programa QuickStart (paso 2)

La conexión a Microsoft Dynamics 365 se realiza por medio de una clase especial llamada **ServerConnection** [10] y que se encuentra en el fichero **CrmServiceHelpers.cs** de la solución *QuickStart*. El objetivo principal de esta clase es mostrar cómo conectarse a los servicios web de Microsoft Dynamics 365 para invocar los métodos web. El mayor problema de esta clase es que está adaptada solamente para trabajar como **aplicación de consola** y que es necesario **re-implementarla** si se quiere usar en otro tipo de aplicaciones.

```
// The connection to the Organization web service.
OrganizationServiceProxy serviceProxy = null;

try
{
    // Obtain the target organization's web address and client logon credentials
    // from the user by using a helper class.
    ServerConnection serverConnect = new ServerConnection();
    ServerConnection.Configuration config = serverConnect.GetServerConfiguration();

    // Establish an authenticated connection to the Organization web service.
    serviceProxy = new OrganizationServiceProxy(config.OrganizationUri, config.HomeRealmUri,
                                                config.Credentials, config.DeviceCredentials);
}
```

Figura 7.4: Instancia del proxy del servicio de organización

La clase **ServerConnection.Configuration** es una clase que sirve para almacenar toda la información necesaria para configurar una conexión al servidor de Microsoft Dynamics 365. Así, en base a la configuración del servidor del CRM obtenida se crea un proxy para el Servicio de Organización (IOrganizationService).

```
/// <summary>
/// Stores Microsoft Dynamics CRM server configuration information.
/// </summary>
public class Configuration
{
    public String ServerAddress;
    public String OrganizationName;
    public Uri DiscoveryUri;
    public Uri OrganizationUri;
    public Uri HomeRealmUri = null;
    public ClientCredentials DeviceCredentials = null;
    public ClientCredentials Credentials = null;
    public AuthenticationProviderType EndpointType;
    public String UserPrincipalName;
}
```

Figura 7.5: Clase configuración del servidor del CRM

La otra clase **CredentialManager** que se proporciona en el fichero *CrmServiceHelpers.cs* de la solución *QuickStart* es la clase encargada de la **administración segura de las credenciales de los usuarios**. Anteriormente, ya comentábamos que durante el proceso de conexión al CRM se produce la autenticación de las credenciales del usuario y solamente entonces el usuario final podía iniciar sesión.

Ahora, para cada configuración de servidor se almacenan las credenciales de usuario (nombre de usuario y contraseña) en el **Administrador de Credenciales de Windows** como una credencial genérica. El formato de denominación de las credenciales es `Microsoft_CRMSDK: <servidor>: <organización>: <idusuario>`.



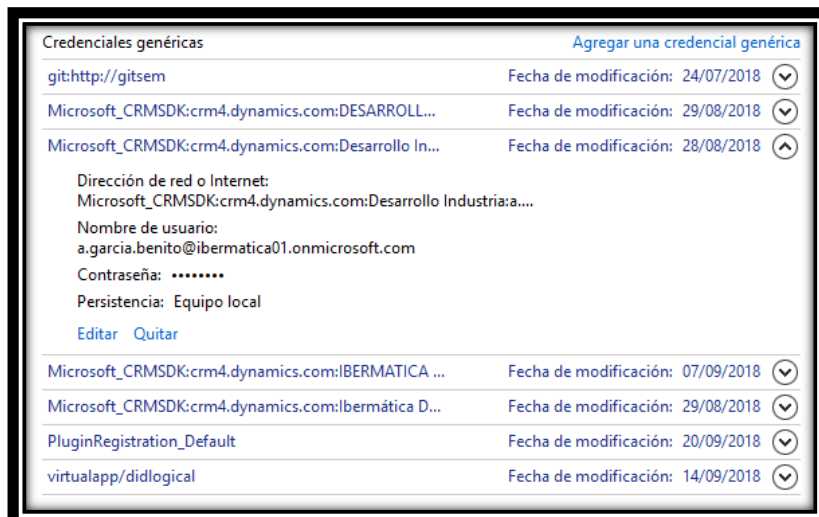


Figura 7.6: Credenciales de Windows del CRM

### 7.2.2 Ibermática proxy

La aplicación de consola realizaba la conexión al CRM y funcionaba perfectamente cuando se probó en un entorno aislado. Sin embargo, los servicios web utilizados en la aplicación CRM-Marketing tienen que pasar por el proxy de Ibermática y se produce el error (407) de autenticación del proxy.

```

Error en el servidor remoto: (407) Se requiere autenticación del proxy.
en System.Net.HttpWebRequest.GetResponse()
en System.ServiceModel.Description.MetadataExchangeClient.MetadataLocationRetriever.DownloadMetadata(TimeoutHelper timeoutHelper)
en System.ServiceModel.Description.MetadataExchangeClient.MetadataRetriever.Retrieve(TimeoutHelper timeoutHelper)
Press <Enter> to exit.

```

La solución adoptada fue añadir un fichero **App.config** a la aplicación y configurarlo de forma que permita a la aplicación trabajar sobre el proxy.

```

<system.net>
  <defaultProxy useDefaultCredentials="true" enabled="true">
    <proxy proxyaddress="http://[redacted]ibermatica[redacted]"/>
  </defaultProxy>
</system.net>

```

Figura 7.7: Configuración del proxy de Ibermática

De este modo, se utilizará la configuración del proxy actualmente existente en Internet Explorer (por defecto) y se usarán las credenciales de inicio de sesión actuales (login) para acceder a la web a través del proxy.

## 7.3 Conexión a la base de datos

---

La conexión a la base de datos de la aplicación CRM-Marketing se realiza mediante el objeto `SqlConnection` del proveedor de datos .NET Framework para SQL Server. A su vez, la cadena de conexión que se usa para abrir la base de datos se almacena en el archivo de configuración de la aplicación (**App.config**).

```
<connectionStrings>
  <add name="ClientesSEMConnectionString"
        connectionString="Data Source=spvdocdes;Initial Catalog=ClientesSEM;
                          Persist Security Info=True;User ID=***;Password=***"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

Figura 7.8: Cadena de conexión a la base de datos

Las razones para incluir la cadena de conexión en el archivo de configuración son:

- Evitar vulnerabilidades de seguridad derivadas de la inclusión de la cadena de conexión en el código de la aplicación.
- Facilitar el mantenimiento de código de la aplicación. Además, si la cadena de conexión se modifica en alguna ocasión ya no es necesario volver a compilar la aplicación.

## 7.4 Configuración del servidor CRM

---

La configuración de cada servidor de Microsoft Dynamics CRM al que tiene acceso el usuario final se almacena en el archivo **Credentials.xml**. Así, en este archivo se almacena toda la información necesaria para poder configurar una conexión al servidor del CRM según se recoge en la clase **ServerConnection.Configuration**.

```
<?xml version="1.0" encoding="utf-8"?>
<Configurations>
  <Configuration>
    <ServerAddress>crm4.dynamics.com</ServerAddress>
    <OrganizationName>IBERMATICA S.A.</OrganizationName>
    <DiscoveryUri>https://disco.crm4.dynamics.com/XRMServices/2011/Discovery.svc</DiscoveryUri>
    <OrganizationUri>https://ibermatica01.api.crm4.dynamics.com/XRMServices/2011/Organization.svc</OrganizationUri>
    <HomeRealmUri>
    </HomeRealmUri>
    <Credentials>
      <UserName>a.garcia.benito@ibermatica01.onmicrosoft.com</UserName>
    </Credentials>
    <EndpointType>OnlineFederation</EndpointType>
    <UserPrincipalName>
    </UserPrincipalName>
  </Configuration>
```

Figura 7.9: Archivo Credentials.xml: configuraciones servidores CRM

La ubicación en disco elegida para almacenar el archivo Credentials.xml es: C:\Users\<username>\AppData\Roaming\CrmServer\Credentials.xml que se encuentra oculta al usuario de la aplicación.

```
/// <summary>
/// Credentials file path.
/// </summary>
public static readonly string ServerCredentialsFile = Path.Combine(
    Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData), "CrmServer"),
    "Credentials.xml");
```

Figura 7.10: Ubicación del Archivo Credentials.xml

De este modo, una vez se ha establecido la conexión inicial al servidor del CRM ya no será necesario volver a llamar al servicio web IDiscoveryService. Así, en las sucesivas ocasiones en que se quiera a conectar al CRM la aplicación leerá la configuración del CRM guardada en el disco local desde el archivo Credentials.xml.

```
public Boolean ReadConfigurations()
{
    //...
    if (File.Exists(CrmServiceHelperConstants.ServerCredentialsFile))
    {
        XElement configurationsFromFile =
            XElement.Load(CrmServiceHelperConstants.ServerCredentialsFile);
        foreach (XElement config in configurationsFromFile.Nodes())
        {
            Configuration newConfig = new Configuration();
            var serverAddress = config.Element("ServerAddress");
            if (serverAddress != null)
                if (!String.IsNullOrEmpty(serverAddress.Value))
                    newConfig.ServerAddress = serverAddress.Value;
            var organizationName = config.Element("OrganizationName");
            if (organizationName != null)
                if (!String.IsNullOrEmpty(organizationName.Value))
                    newConfig.OrganizationName = organizationName.Value;
            //...
        }
    }
}
```

Figura 7.11: Función ReadConfigurations()

## 7.5 Confidencialidad de la contraseña de usuario

---

La protección de la contraseña del usuario es un elemento crítico para asegurar la seguridad de nuestra aplicación. Por este motivo, en todo momento la aplicación CRM-Marketing trata de forma segura la contraseña y se almacena encriptada.

```
private SecureString ConvertToSecureString(string secret)
{
    if (string.IsNullOrEmpty(secret))
        return null;

    SecureString securePassword = new SecureString();
    char[] passwordChars = secret.ToCharArray();
    foreach (char pwdChar in passwordChars)
    {
        securePassword.AppendChar(pwdChar);
    }
    securePassword.MakeReadOnly();
    return securePassword;
}
```

Figura 7.12: Función ConvertToSecureString()

## 7.6 Funcionalidad “Crear Lista de Marketing”

---

El Servicio de Organización (**IOrganizationService**) [11] contiene los métodos necesarios para realizar las **operaciones más comunes** sobre los datos (registros de entidades) y metadatos (entidades, atributos, relaciones, etc.) en Microsoft Dynamics 365. [12][13] A partir de ahora, se van a hacer uso de esos métodos para desarrollar las nuevas funcionalidades de la aplicación CRM-Marketing.

La creación de las **listas de marketing** [14] en Microsoft Dynamics CRM Online se realiza a partir de la información existente en la tabla LISTA\_MAIL de la base de datos de la aplicación. Dicha información tiene origen en la plataforma de soporte VDoc.

### 7.6.1 Criterios para la definición de una Lista de Marketing

La creación de las listas de marketing se tiene que implementar cumpliendo las siguientes condiciones:

1. Creación de la lista de marketing
  - Si la lista de marketing no existe en el CRM se crea una lista nueva (nombre). [14]
  - Si la lista de marketing ya existe en el CRM se actualiza la lista existente.
  - Si existe la lista de marketing está repetida en el CRM, la aplicación **NO** permite continuar su ejecución debido a que se está incumpliendo una regla de integridad (solamente puede existir una lista de marketing en el CRM con el mismo nombre).
2. Validación de la empresa del contacto
  - Si la empresa del contacto en la base de datos existe en el CRM la aplicación sigue su flujo normal de ejecución. [15]
  - Si la empresa del contacto en la base de datos no existe en el CRM la aplicación continua ejecutándose pero ese contacto no se añade al CRM. [15]  
Nota: Posteriormente, esta empresa deberá añadirse manualmente al CRM.
3. Verificación del email del contacto [16]
  - Si el email del contacto ya existe en el CRM entonces, se obtienen los datos del contacto a partir del CRM y se añade el contacto a la lista de marketing. [17]
  - Si el email del contacto no existe en el CRM entonces, se crea un contacto nuevo en el CRM con los datos provenientes de la base de datos, se asocia a su cuenta (empresa) correspondiente y se añade el contacto a la lista de marketing. [12][13] [17]

## 7.6.2 Creación de Listas de Marketing (código)

En este apartado, se van a explicar las partes más significativas del proceso de creación de una lista de marketing junto con fragmentos de código que ayuden a entender mejor el proceso.

El primer paso es verificar si la lista de marketing (base de datos) ya existe en Microsoft Dynamics CRM Online de Ibermática. La figura 7.13 muestra el método *CountRepeatedLMs* que devuelve el número de apariciones de la lista de marketing en el CRM. Si el valor devuelto por el método es mayor que 1 la aplicación aborta su ejecución y mostrará un mensaje de error, en caso contrario, se crea (actualiza) la lista de marketing.

```
/// <summary>
/// Counts number of appearances of target LM in CRM that matches the name of the target LM in DataBase.
/// Note: Expected results are 0 or 1. More than 1 is a violation of Primary Key[list name].
/// </summary>
/// <param name="Mailing_List">Selected Mailing List from DB to be import in CRM.</param>
/// <returns>Number of appearances of Marketing Lists in CRM with the same list name
/// [Primary Key] than the target Mailing List of the DataBase.</returns>
public static int CountRepeatedLMs(String Mailing_List)
{
    int count = 1;
    // returns Lists of Marketing in CRM that matches list name with target Mailing_List.
    EntityCollection lmquery = CRMConnection.QueryMailing_List(Mailing_List);

    371 // <returns>All Marketing Lists in CRM that matches the target Mailing_List</returns>
    372 private static EntityCollection QueryMailing_List(String Mailing_List)
    373 {
    374     // Create query using querybyattribute
    375     QueryByAttribute querybyexpression = new QueryByAttribute("list");
    376     querybyexpression.ColumnSet = new ColumnSet("listname", "listid");
    377     // Attribute to query
    378     querybyexpression.Attributes.AddRange("listname");
    379     // Value of queried attribute to return
    380     querybyexpression.Values.AddRange(Mailing_List);

    // Number of Marketing Lists returned.
    count = lmquery.Entities.Count;

    return (count);
}
```

Figura 7.13: Función *CountRepeatedLMs()*

El método *CreateLM* es el método principal de la funcionalidad “Crear Lista de Marketing”. Este método crea la lista de marketing (si no existe en el CRM), obtiene las cuentas (empresas) del CRM y comprueba si la empresa del contacto está entre ellas. En caso afirmativo, verifica si el email del contacto existe en el CRM, si existe se añade el contacto del CRM a la lista de marketing y si no existe crea un contacto nuevo, lo asocia a la empresa correspondiente del CRM y se añade a la lista de marketing. El método *CreateLM* hace a su vez llamadas a muchos más métodos: *LoggedUser2*, *GetMarketingList*, *CheckAllListContacts*, *QueryAccounts*, *EmpresaCRM*, *QuerybyContact*, *EmailContact*, etc.

En la figura 7.14 se muestra a modo de ejemplo explicativo el método *CreateLM* casi al completo.

```
public static void CreateLM(String Mailing_List, DataTable ImportedRecords, int countLM,
out String userdetails, out String outmessage)
{
    outmessage = String.Empty;
    // Display information about the logged on user.
    LoggedUser2(out userdetails);

    // Get the Marketing List (create if doesn't exist) according to the selected Mailing List.
    Guid marketingListId = new Guid();
    GetMarketingList(Mailing_List, countLM, out marketingListId, out outmessage);

    // Associate the contacts to the marketing list record. // Create the request object.
    AddMemberListRequest addlist = new AddMemberListRequest();
    // Create a query expression to return ALL accounts in CRM.
    EntityCollection acc = CRMConnection.QueryAccounts();
    //--
    String SAPcode; // Code that identify a company.
    // Number of contacts (rows) in selected Mailing List.
    int nRows = ImportedRecords.Rows.Count;
    for (int i = 0; i < nRows; i++)
    {
        SAPcode = ImportedRecords.Rows[i]["empresa"].ToString();
        // Check if the company already exists in CRM (if String.Empty doesn't exist).
        String accountIdString = CRMConnection.EmpresaCRM(acc, SAPcode, ref outmessage);

        // If the company exists in CRM, add contact to Marketing List.
        if (accountIdString != String.Empty)
        {
            String email = ImportedRecords.Rows[i]["email"].ToString();
            // Create a query expression to return the target contact (email).
            EntityCollection cnt = CRMConnection.QuerybyContact(email);
            // Contact ID of contact matching target email in CRM (if any) otherwise empty string.
            String contactIdString = CRMConnection.EmailContact(cnt, ImportedRecords, i, ref outmessage);

            // Contact is already in CRM: add existing contact to list.
            if (contactIdString != String.Empty)
            {
                // Add a contact to the Marketing List.
                CRMConnection.AddnewListContact(addlist, contactIdString,
                    marketingListId.ToString(), ref outmessage);
            }
            // Contact doesn't exist in CRM: create new contact in CRM, associate
            // to the account(empresa) and add it to the list.
            else
            {
                // Create an contact record from Mailing List.
                Guid contactlistId = CRMConnection.CreateNewContact(ImportedRecords, i);
                // Associate a new contact to the account record (empresa).
                CRMConnection.AssociateAccountContact(contactlistId, accountIdString);
                // Add a new contact to the Marketing List.
                CRMConnection.AddnewListContact(addlist, contactlistId.ToString(),
                    marketingListId.ToString(), ref outmessage);
            }
        }
    }
}
```

Figura 7.14: Método *CreateLM()*

### 7.6.3 Crear de Listas de Marketing (aplicación CRM-Marketing)

Los pasos que debe seguir el usuario final para crear una lista de marketing con la aplicación CRM-Marketing son los siguientes:

1. Seleccionar la organización (instancia del CRM) a la que se quiere conectar el usuario.
2. Conectarse a servidor CRM de la organización seleccionada. Si la conexión con el CRM es correcta se habilitará el botón para conectarse a la base de datos, en caso contrario, se mostrará por pantalla un mensaje de error.
3. Conectarse a la base de datos. Si la conexión se realiza correctamente se mostrarán las listas de mailing disponibles y se habilitará el botón [run]. En caso contrario, se mostrará por pantalla un mensaje de error.
4. Seleccionar la lista de mailing que se quiere importar para crear la lista de marketing correspondiente en el CRM.
5. Clicar la casilla *Create a Log File* si se quiere crear un archivo Log con los resultados obtenidos en el proceso de creación de la lista de marketing.
6. Finalmente, pulsando el botón [run] se ejecuta la funcionalidad. Previa a la ejecución se pedirá confirmación para la creación de la lista de marketing.

Como se ha podido comprobar todos los pasos para la creación de una lista de marketing son sencillos y el funcionamiento de la interfaz gráfica dirige al usuario en todo momento para evitar que cometa errores. En la figura 7.15 se muestra el resultado final después de ejecutar la funcionalidad “Crear Listas de Marketing”.

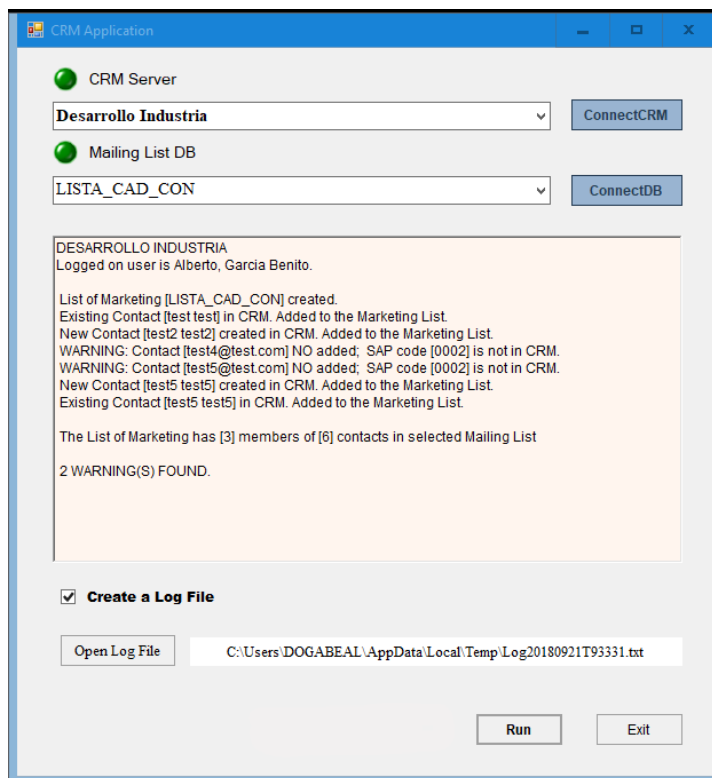


Figura 7.15: App: Resultado funcionalidad “Crear Listas de Marketing”

Finalmente, podemos comprobar el resultado obtenido en Microsoft Dynamics CRM Online de Ibermática (sandbox). Siguiendo el ejemplo que se muestra en la figura anterior podemos verificar como realmente se ha creado la lista de marketing LISTA\_CAD\_CON con los tres contactos indicados: test, test2 y test5. Nótese que aunque el contacto test5 está repetido solamente se añade una vez a la lista de marketing.

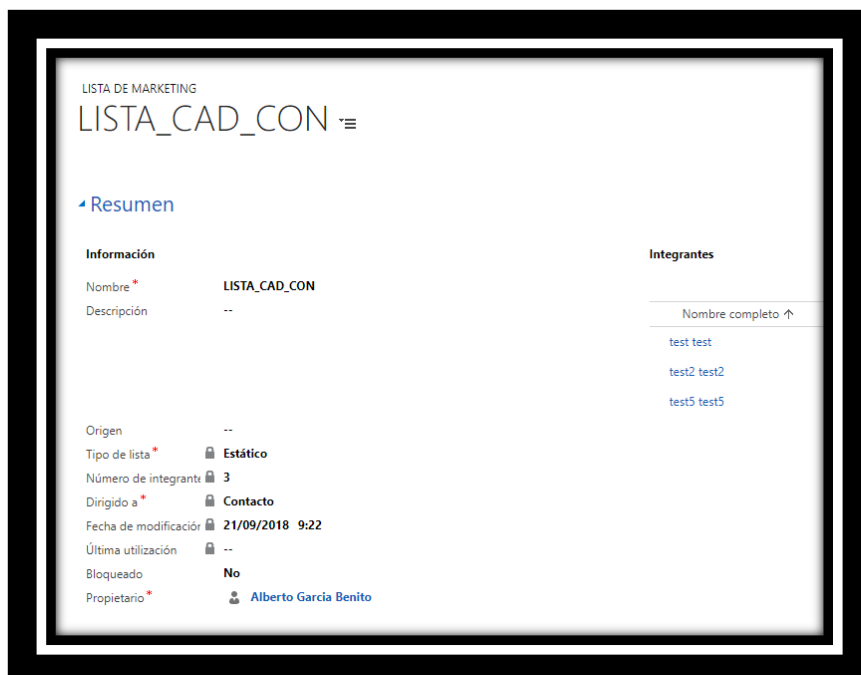


Figura 7.16: CRM: Información General de LISTA\_CAD\_CON

A su vez, en la figura 7.17 también podemos comprobar que los contactos test2 y test5 se han creado en el CRM y que están correctamente asociados a la cuenta (empresa) correspondiente. Además, como podemos advertir el contacto test5 solamente se ha creado una vez en el CRM.

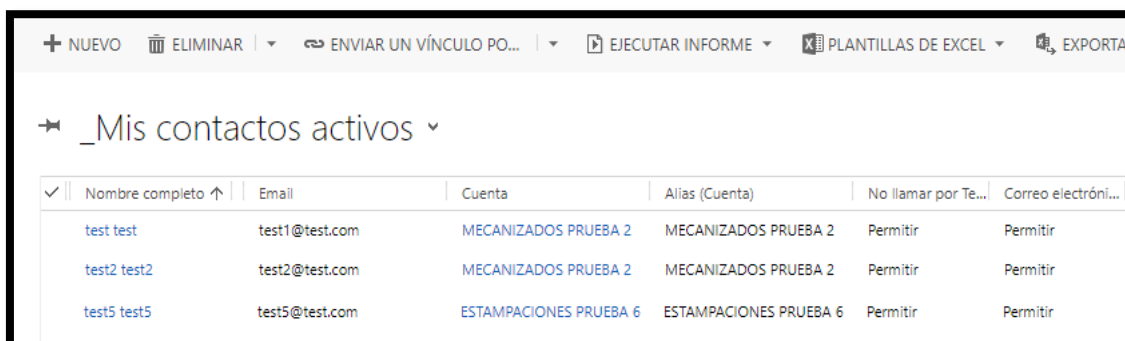


Figura 7.17: CRM: Vista de Contactos (1)



## 7.7 Funcionalidad “Sincronizar Contactos CRM”

La sincronización de los contactos en Microsoft Dynamics CRM Online se realiza a partir de la información existente en la tabla CONTACTOS de la base de datos de la aplicación. Dicha información tiene origen en la plataforma de soporte VDoc.

En esta funcionalidad, se prioriza la información de los contactos existente en el CRM respecto a la información almacenada en la tabla CONTACTOS. De este modo, si un contacto de la tabla ya existe en el CRM simplemente se descarta, en cambio, si no existe en el CRM se comprueba si existe la empresa del contacto en el CRM y si es el caso se añade al CRM. Esta última comprobación se realiza debido a que en Microsoft Dynamics CRM Online de IBERMÁTICA no puede existir un contacto si no está asociado a una empresa.

El método *UpdateCRMContacts* es el método principal de la funcionalidad “Sincronizar contactos CRM”. Este método obtiene las cuentas (empresas) del CRM y comprueba si la empresa del contacto está entre ellas. En caso afirmativo, verifica si el email del contacto existe en el CRM y si no existe crea un contacto nuevo en el CRM y lo asocia a la cuenta (empresa) correspondiente. En la figura 7.18 se muestra a modo de ejemplo explicativo el método *UpdateCRMContacts*. [12][13] [15] [16]

```
public static void UpdateCRMContacts(DataTable contactRecords, out String outmessage, out int addedcontacts, out int errors)
{
    outmessage = String.Empty;
    errors = 0; // Number of companies (empresas) in DataBase with no SAP code matching in CRM.
    addedcontacts = 0; // Number of contacts from DataBase (with no matching CRM) to be added into CRM.
    try
    {
        //Create a query expression to return ALL accounts in CRM.
        EntityCollection acc = CRMConnection.QueryAccounts();
        // SAPCode = Code that identify an company (empresa value in DB) in CRM.
        String SAPcode; String email;
        for (int n = 0; n < contactRecords.Rows.Count; n++)
        {
            SAPcode = contactRecords.Rows[n]["empresa"].ToString();
            // Check if the company already exists in CRM (if String.Empty doesn't exist).
            String accountIdString = CRMConnection.EmpresaCRM(acc, SAPcode, ref outmessage);

            // If the company exists in CRM
            if (accountIdString != String.Empty)
            {
                email = contactRecords.Rows[n]["email"].ToString();
                //Create a query expression to return the target contact (email).
                EntityCollection cnt = CRMConnection.QuerybyContact(email);
                // Check if the contact that matches email already exists in CRM.
                String contactIdString = CRMConnection.EmailContact(cnt, contactRecords, n, ref outmessage);

                // If contact is not in CRM: add contact into CRM.
                if (contactIdString == String.Empty)
                {
                    // Create an contact record from CONTACTOS DataBase.
                    Guid contactlistId = CRMConnection.CreateNewContact(contactRecords, n);
                    // Associate a new contact to the account record (empresa).
                    CRMConnection.AssociateAccountContact(contactlistId, accountIdString);
                    addedcontacts++;
                }
            }
            else { errors++; }
        }
    }
    catch (Exception e)
    {}
}
```

Figura 7.18: Método *UpdateCRMContact()*

Los pasos que debe seguir el usuario final para actualizar los contactos del CRM con la aplicación CRM-Marketing son los siguientes:

1. Seleccionar la organización (instancia del CRM) a la que se quiere conectar el usuario.
2. Conectarse a servidor CRM de la organización seleccionada. Si la conexión con el CRM es correcta se habilitará el botón para conectarse a la base de datos, en caso contrario, se mostrará por pantalla un mensaje de error.
3. Conectarse a la base de datos. Si la conexión se realiza correctamente se habilitará el botón [Update Contacts]. En caso contrario, se mostrará por pantalla un mensaje de error.
4. Clicar la casilla *Create a Log File* si se quiere crear un archivo Log con los resultados obtenidos en el proceso de actualización de los contactos.
5. Finalmente, pulsando el botón [Update Contacts] se ejecuta la funcionalidad. Previa a la ejecución se pedirá confirmación para la actualización de los contactos.

Al igual que para la funcionalidad anterior, en este caso también todos los pasos para la ejecución de esta funcionalidad son sencillos y el funcionamiento de la interfaz gráfica dirige al usuario para evitar que cometa errores. En la figura 7.19 se muestra el resultado final después de ejecutar la funcionalidad “Sincronizar Contactos CRM”.

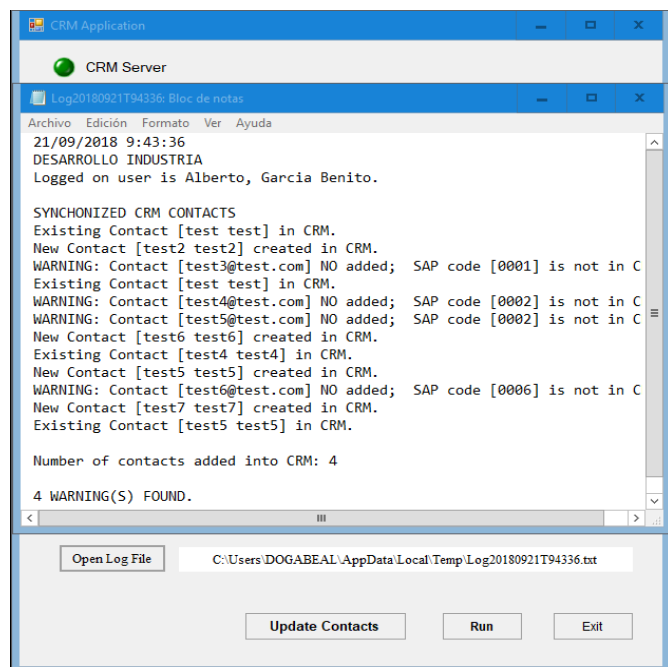


Figura 7.19: App: Resultado funcionalidad “Sincronizar contactos CRM”

En la figura 7.20 se muestra el resultado en el CRM. Los cuatro clientes añadidos test2, test5, test6 y test 7 y los dos clientes inicialmente existentes test y test4.

Nombre completo	Email	Cuenta	Alias (Cuenta)	No llamar por Te...	Correo electróni...
test test	test1@test.com	MECANIZADOS PRUEBA 2	MECANIZADOS PRUEBA 2	Permitir	Permitir
test2 test2	test2@test.com	MECANIZADOS PRUEBA 2	MECANIZADOS PRUEBA 2	Permitir	Permitir
test4 test4	test4@test.com	LAMINADORA PRUEBA 5	LAMINADORA PRUEBA 5	Permitir	Permitir
test5 test5	test5@test.com	ESTAMPACIONES PRUEBA 6	ESTAMPACIONES PRUEBA 6	Permitir	Permitir
test6 test6	test6@test.com	TALLERES PRUEBA3	TALLERES PRUEBA 3	Permitir	Permitir
test7 test7	test7@test.com	CARPINTERIA PRUEBA 4	CARPINTERIA PRUEBA 4	Permitir	Permitir

Figura 7.20: CRM: Vista de Contactos (2)

# 8

---

---

## **8. *Plug-in de Ventas***

Al inicio del capítulo, se realiza una introducción a los servicios necesarios para poder crear plug-ins para Microsoft Dynamics 365. Después, se aplican esos conocimientos para implementar la solución del plug-in de ventas.

En la mitad del capítulo, se va explicar cómo se han implementado los dos proyectos que forman la solución del plug-in de ventas:

1. Proyecto 'Plug-ins': Donde se implementa la funcionalidad del plug-in.
2. Proyecto 'crm2sqlws': Donde se implementa el servicio WCF a través del cual se comunica el plug-in con la base de datos.

Finalmente, se explica el proceso de registro del plug-in que incluye una guía recogida en un anexo y una prueba de funcionamiento del plug-in para comprobar que funciona según los requisitos establecidos.

## 8.1 Plug-ins en Microsoft Dynamics 365

---

Un plug-in es una aplicación (componente software) que, añadido a una aplicación principal, implementa o mejora alguna funcionalidad del software. Lo más frecuente es que el plug-in sea ejecutado por el software principal, con el que interactúa a través de una determinada interfaz.

Un plug-in en Microsoft Dynamics 365, es un componente de código con una lógica de negocios personalizada que se instala en la plataforma para modificar o mejorar el comportamiento del CRM. El plug-in se ejecuta cuando se produce un evento concreto en Microsoft Dynamics 365 e interactúa con el CRM a través de la interfaz **IPlugin**.

El método **IPlugin.Execute (IServiceProvider)** de la interfaz **IPlugin** es el método que ejecuta el código de plug-in en respuesta a un evento. El parámetro **IServiceProvider** del método **Execute**, contiene referencias al contexto de ejecución del plug-in, (**IPluginExecutionContext**), el servicio de organización (**IOrganizationServiceFactory**), el servicio de seguimiento (**ITracingService**), y el servicio de notificación (**IServiceEndpointNotificationService**). [18]

**IPluginExecutionContext** contiene información que describe el entorno en tiempo de ejecución en el que se ejecuta el plug-in, información relacionada con el flujo de ejecuciones e información de negocio de la entidad. [18]

```
// Obtain the execution context from the service provider.
IPluginExecutionContext context = (IPluginExecutionContext)
    serviceProvider.GetService(typeof(IPluginExecutionContext));
```

Figura 8.1: Acceso al contexto de ejecución del plug-in

**IOrganizationServiceFactory** permite crear instancias del servicio de organización (**IOrganizationService**) que es el servicio web que da acceso a los datos (registros de entidades) y metadatos (entidades, atributos, relaciones, etc.) de una organización (instancia del CRM). [18]

```
// Obtain the organization service reference.
IOrganizationServiceFactory serviceFactory = (IOrganizationServiceFactory)
    serviceProvider.GetService(typeof(IOrganizationServiceFactory));
IOrganizationService service =
    serviceFactory.CreateOrganizationService(context.UserId);
```

Figura 8.2: Acceso al servicio de Organización

**ITracingService** proporciona un método para registrar información de seguimiento (trazas) en tiempo de ejecución en Microsoft Dynamics CRM Online. Este método es especialmente útil para los plug-ins del entorno aislado que no pueden ser depurados usando un “debugger”. [18]

```
// Obtain the tracing service from the service provider.  
ITracingService trace = (ITracingService)  
    serviceProvider.GetService(typeof(ITracingService));
```

Figura 8.3: Acceso al servicio de Seguimiento (trazas)

Sin embargo, la implementación del **ITracingService** dentro del código no es suficiente para que nuestro plug-in empiece a escribir automáticamente los registros de seguimiento en el CRM, antes debemos cambiar la configuración del sistema de Microsoft Dynamics CRM Online y *habilitar el registro para registros de seguimiento de complementos* en la sección **Seguimiento de complemento y actividad de flujo personalizado** de la pestaña de **Personalización** como se muestra en la siguiente figura 8.4.



Figura 8.4: Configuración del sistema del CRM

**IServiceEndpointNotificationService** es el servicio de notificación por el que los plug-ins que se ejecutan de forma síncrona envían los mensajes de intermediación al Microsoft Azure Service Bus. Sin embargo, este servicio **no** va a ser utilizado en el plug-in de ventas porque no vamos a trabajar con Microsoft Azure. [18]

## 8.2 Plug-in de Ventas en Microsoft Dynamics CRM Online

El **plug-in de ventas** se ejecuta cuando una Oportunidad (entidad) se cierra como ganada en Microsoft Dynamics CRM Online, es decir, cuando se produce una venta de un producto o servicio de Ibermática. De este modo, el **evento** que desencadena la ejecución del plug-in es la acción de **cerrar** una **Oportunidad** como **ganada** en el CRM. En ese momento, el plug-in de ventas recopila toda la información necesaria sobre la venta, que está distribuida entre varias entidades del CRM: *oportunidad*, *cuenta*, *importes*, *usuario* y *área*, y envía esa información a la base de datos del plug-in. Finalmente, la información recibida del plug-in de ventas se almacena en las tablas *Oportunidad* e *Importes* de la base de datos.

Si n embargo, el plug-in de ventas no establece una conexión directa con la base de datos si no que lo hace a través de un servicio WCF (**Windows Communication Foundation**) [19] que se crea a tal efecto. Este servicio WCF se ha implementado por **motivos de seguridad** ya que se quieren controlar los accesos a la información de las ventas del CRM.

Finalmente, la intención es transferir toda esa información de las ventas almacenada en la base de datos a la plataforma de soporte VDoc pero, el desarrollo de esa parte queda fuera del alcance de este proyecto.

De este modo, la implementación del plug-in de ventas queda estructurada en dos proyectos independientes:

1. La implementación de la funcionalidad del plug-in de ventas en sí misma: Esta funcionalidad se desarrolla en el proyecto '**Plug-ins**' donde se van a agrupar todos los plug-ins que se vayan agregando a Microsoft Dynamics CRM Online de Ibermática.

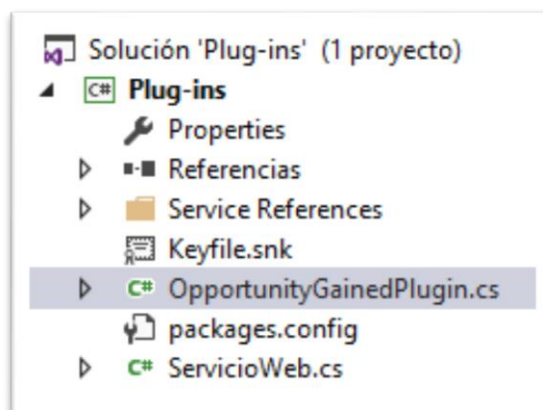


Figura 8.5: Archivos del proyecto 'Plug-ins'

2. La implementación del servicio WCF que intercomunica el plug-in con la base de datos: Este servicio se desarrolla en el sitio web '**crm2sqlws**'. El proyecto '**clienteTest**' es un proyecto cuya única finalidad es probar el comportamiento del servicio web implementado.

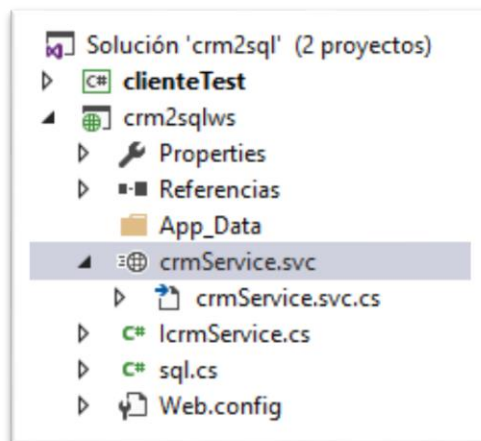


Figura 8.6: Archivos del sitio web '*crm2sqlws*'

### 8.2.1 Proyecto '*Plug-ins*'

En este apartado se va a explicar la implementación del plug-in de ventas paso a paso hasta conectarse al servicio WCF *crmService*. Se incluyen fragmentos de código significativos que reflejan las características destacables.

El desarrollo del plug-in de ventas se realiza en el proyecto '**Plug-ins**'. Los dos archivos más relevantes del proyecto son:

1. **OportunityGainedPlugin.cs**: En este fichero se desarrolla la funcionalidad del plug-in de ventas propiamente dicha.
2. **ServicioWeb.cs**: En este fichero se desarrollan los métodos que comunican con el servicio WCF y que transmiten los datos obtenidos del CRM al servicio web.

#### 8.2.1.1 Archivo '*OportunityGainedPlugin.cs*'

El método **Execute** del archivo *OportunityGainedPlugin.cs* es el método que ejecuta el código del plug-in de ventas en respuesta a un evento. [18] El evento que desencadena la acción del plug-in de ventas es *cerrar una oportunidad como ganada en el CRM*.

En primer lugar, se obtiene el contexto de ejecución del plug-in (*IPluginExecutionContext*) y se activa el servicio de seguimiento (*ITracingService*) para poder realizar trazas de código, mediante el proveedor de servicios (*IServiceProvider*).

```

public void Execute(IServiceProvider serviceProvider)
{
    // Obtain the execution context from the service provider.
    IPluginExecutionContext context = (IPluginExecutionContext)
        serviceProvider.GetService(typeof(IPluginExecutionContext));

    // Obtain the tracing service from the service provider.
    ITracingService trace = (ITracingService)
        serviceProvider.GetService(typeof(ITracingService));
    trace.Trace("Started");
}

```

Figura 8.7: Método *Execute* de la interfaz *IPlugin*

El segundo paso es asegurarse de que el plug-in solamente se ejecute cuando se produzca el evento *ganar para una oportunidad que se ha cerrado*.

```

// If the event is not "Win" then abort plug-in execution.
if (context.MessageName != "Win") { return; }
// If input data is not from OpportunityClose entity then
// abort plug-in execution.
if (!(context.InputParameters.Contains("OpportunityClose")
    && context.InputParameters["OpportunityClose"] is Entity)) { return; }

```

Figura 8.8: Evento del contexto y parámetros de entrada

El siguiente paso es obtener la referencia al servicio de organización (*IOrganizationService*) que da acceso a los datos (registros de entidades) y metadatos (entidades, atributos, relaciones, etc.) del CRM.

```

// Obtain the organization service reference.
IOrganizationServiceFactory serviceFactory = (IOrganizationServiceFactory)
    serviceProvider.GetService(typeof(IOrganizationServiceFactory));
IOrganizationService service =
    serviceFactory.CreateOrganizationService(context.UserId);
trace.Trace("Step 2");

```

Figura 8.9: Referencia al servicio de Organización

A partir de este punto, ya se desarrollan los métodos que acceden a los datos de las entidades (oportunidad, cuenta, importes, usuario y área) para registrar una oportunidad como ganada. [20] En la figura 8.10 se muestra el método *GetOpportunityAttributes* que obtiene los atributos de una oportunidad.

```

private void GetOpportunityAttributes(IOrganizationService service, Entity Opp, out Guid OppId,
    out Entity Opt, out String refOP, out String nombreOP, out String numopsiebel)
{
    OppId = Guid.Empty; Opt = null;
    refOP = String.Empty; nombreOP = String.Empty; numopsiebel = String.Empty;
    if (Opp.Attributes.Contains("opportunityid"))
    {
        OppId = ((EntityReference)Opp.Attributes["opportunityid"]).Id;
        ColumnSet cols = new ColumnSet(new String[]
            { "iber_idautonumerico", "name", "iber_numoportunidadesiebel", "ownerid", "parentaccountid" });

        Opt = (Entity)service.Retrieve("opportunity", OppId, cols);
        if (Opt.Attributes.Contains("iber_idautonumerico") == true) refOP =
            Opt.Attributes["iber_idautonumerico"].ToString();
        if (Opt.Attributes.Contains("name") == true) nombreOP = Opt.Attributes["name"].ToString();
        if (Opt.Attributes.Contains("iber_numoportunidadesiebel") == true) numopsiebel =
            Opt.Attributes["iber_numoportunidadesiebel"].ToString();
    }
}

```

Figura 8.10: Procedimiento *GetOpportunityAttributes()*



Finalmente, todos los datos del CRM recopilados por el plug-in de ventas se transmiten al servicio WCF mediante los métodos definidos en el archivo *ServicioWeb.cs*.

#### 8.2.1.2 Archivo 'ServicioWeb.cs'

El archivo *ServicioWeb.cs* contiene todos los métodos necesarios para transmitir los datos desde el plug-in al servicio WCF. A su vez, también se configura la conexión al servicio web en este mismo archivo.

```
//Specify the binding to be used for the client.  
BasicHttpBinding binding = new BasicHttpBinding();  
//Specify the address to be used for the client.  
EndpointAddress address =  
    new EndpointAddress("http://prueba.ibermatica.com/crmService.svc");  
  
IcrmServiceClient wsclient = new IcrmServiceClient(binding, address);
```

Figura 8.11: Configurar conexión con el servicio WCF

Como se puede advertir la solución adoptada para almacenar la configuración de la conexión para el plug-in de ventas difiere de la solución tomada para la aplicación CRM-Marketing que se guardaba en un archivo de configuración. El motivo es que como el plug-in de ventas se instala en el CRM no puede acceder a dicho archivo de configuración durante su ejecución.

#### 8.2.2 Sitio Web 'crm2sqlws'

El servicio WCF *crmService* [19] es un servicio de comunicación de datos que hace de intermediario entre el plug-in de ventas y la base de datos donde se va a almacenar la información de la venta.

El desarrollo del servicio *crmService* se realiza en el sitio web **crm2sqlws**. Los tres archivos más relevantes del sitio web son:

1. **IcrmService.cs**: Es el fichero donde se define la clase Interfaz de servicio.
2. **crmService.cs**: Es el fichero donde se implementan los métodos de la interfaz de servicio.
3. **sql.cs**: Es el fichero donde se implementan la conexión con la base de datos y los métodos que guardan la información de la venta en la base de datos.

### 8.2.2.1 Archivo 'IcrmService.cs'

En el archivo *IcrmService.cs* es donde se define el **contrato del servicio** [ServiceContract]. [21] El contrato de servicio especifica los métodos web [OperationContract] que admite el servicio WCF *crmService*. En la figura 8.12 se muestran algunos de los métodos del contrato de servicio definidos para nuestro servicio WCF.

```
[ServiceContract]
public interface IcrmService
{
    [OperationContract]
    string registrarOP(string id, oportunidad OP);

    [OperationContract]
    string registrarImporte(string id, importes Imp);
}
```

Figura 8.12: Contrato de servicio del servicio WCF

En este mismo archivo también se define el **contrato de datos** [DataContract]. “Un contrato de datos es un acuerdo formal entre un servicio y un cliente que abstractamente describe los datos que se van a intercambiar”. [22] En nuestro caso, sería un contrato entre el servicio WCF *crmService* que estamos implementando y el cliente que es la base de datos donde se va a almacenar la información. Un [DataMember] es aquel dato (variable o propiedad) que va a ser serializado (en XML) para su intercambio. En la figura 8.12 se muestra un ejemplo de un campo (numero) definido como datamember.

```
[DataContract]
public class oportunidad
{
    private string _numero;
    [DataMember]
    public string Numero
    {
        get { return _numero; }
        set { _numero = value; }
    }
}
```

Figura 8.13: Contrato de datos del servicio WCF

### 8.2.2.2 Archivo 'crmService.cs'

Es el archivo donde se implementan los métodos de la interfaz de servicio. Como **medida seguridad** es necesario que el ID que reciba desde el método homónimo del plug-in de ventas coincida con el código de acceso (“secretpassword”) definido para la base de datos. Posteriormente, si el ID es el correcto se llama al respectivo método de la clase **Sql** para escribir la información recibida en la base de datos. En la figura 8.13 se muestra la función *registrarOP* a modo de ejemplo.

```
public string registrarOP(string id, oportunidad OP)
{
    string rval = null;
    if (id.Equals("secretpassword"))
    {
        rval = sql.registrarOP(OP);
        if (rval == null)
            rval = "Registro Oportunidad ok";
    }
    return rval;
}
```

Figura 8.14: Función registrarOP(1)

### 8.2.2.3 Archivo 'sql.cs'

En este archivo se define la conexión con la base de datos y contiene los métodos que escriben en la base de datos la información proveniente del plug-in de ventas.

En la siguiente figura 8.14 se muestra la función *registrarOP*. Nótese el uso de parámetros en la operación SQL de insert, se trabaja con parámetros para **prevenir ataques de inyecciones SQL**.

```
static internal string registrarOP(oportunidad OP)
{
    string rval = null;
    try
    {
        string SqlOrder = "insert into oportunidad (numero,nombre,propietario,numopsiebel,numcl
        SqlConnection con = connectDB();
        SqlCommand cmd = new SqlCommand(SqlOrder, con);
        cmd.Parameters.AddWithValue("@numero", OP.Numero);
        cmd.Parameters.AddWithValue("@nombre", OP.Nombre);
        //...
        if (cmd.ExecuteNonQuery() != 1) rval = "ERROR registrando en sql(oportunidad).";
        con.Close();
    }
    catch (Exception e) { rval = e.Message; }
    return rval;
}
```

Figura 8.15: Función registrarOP(2)

## 8.3 Registro del Plug-in de Ventas

---

Una vez implementado el plug-in de ventas se debe registrar con el servidor del CRM correspondiente a la organización a la que pertenece el usuario. El kit de desarrollo software (SDK) de Microsoft Dynamics 365 proporciona un herramienta específica para el registro de plug-ins (**Plug-in Registration tool**).

Sin embargo, antes nos tenemos que asegurar de que la cuenta del usuario que se va a utilizar para registrar el plug-in con la herramienta de registro de plug-ins tenga los siguientes **privilegios de seguridad**: [23]

- prvCreatePluginAssembly
- prvCreatePluginType
- prvCreateSdkMessageProcessingStep
- prvCreateSdkMessageProcessingStepImage
- prvCreateSdkMessageProcessingStepSecureConfig

Las cuentas de los administradores del CRM por defecto ya disponen de todos los permisos necesarios pero, si en cambio se va a utilizar una tipo de cuenta de usuario diferente primeramente se le tienen que otorgar los permisos anteriormente enumerados para poder registrar el plug-in con la herramienta del SDK.

Los pasos a seguir para registrar el plug-in de ventas mediante la herramienta “**Plug-in Registration**” para Microsoft Dynamics CRM Online se divide en tres etapas:

- Conectarse al servidor de Dynamics 365.
- Registrar el ensamblado del plug-in de ventas.
- Registrar el plug-in de ventas para el evento “Win opportunity”.

El proceso completo para registrar el plug-in de ventas se recoge en el anexo A: **Guía para el Registro del Plug-in de Ventas.** [\[24\]](#)

Al terminar el proceso, el plug-in de ventas está registrado para ejecutarse con el evento “Win opportunity” (ganar una oportunidad), después de la ejecución del programa principal y en modo asíncrono.

## 8.4 Prueba de funcionamiento del Plug-in de Ventas

---

Una vez finalizado el registro del plug-in de ventas en Microsoft Dynamics CRM Online el plug-in queda integrado y le añade la funcionalidad implementada al CRM. A continuación se va a mostrar el resultado obtenido en el CRM y en la base de datos de una prueba de funcionamiento del plug-in de ventas

El proceso comienza cuando un trabajador comercial cierra una oportunidad (entidad) como ganada en el CRM. En el panel de información general de la oportunidad, el estado de la oportunidad cambiará a OP-Ganada y se desencadenará el evento que ejecuta el plug-in de ventas. En la figura 8.15 se muestra la información general de la oportunidad **pruebaAceros**.

OPORTUNIDAD

## pruebaAceros

✓ Captación → ✓ Cualificación → ✓ Validación

✓ Alcance Técnico\* Sí | ✓ Condiciones Económicas\* Sí | ✓ Formas Pago\* Sí | ✓ Pedido\* Sí

---

General

Información de la oportunidad		Seguimiento de la oportunidad	
Nombre*	pruebaAceros	Estado Oport.	OP-Ganada
Cuenta*	ACEROS PRUEBA 1	Probabilidad	100
Contacto OP	--	Fecha est. cierre	30/11/2018
Area*	ING	Nº Op. Siebel*	1234
Campaña de origen*	_prueba	vencimiento	--
Siguiente Paso	--	Fecha de cierre rea	14/09/2018

Figura 8.16: CRM: Información general entidad Oportunidad

En este mismo panel de la oportunidad, en la sección *Importes* se muestra la información sobre los Importes (entidad) que incluye la información comercial de cada tipo de producto vendido (registro). En la figura 8.16 se muestra la información sobre los importes de la oportunidad **pruebaAceros**.

Importes

Total Coste: 1.500,00€  
 Total Importe: 4.000,00€  
 Total Margen: 2.500,00€

Sección	Nombre ↑	Area	Solucion	Producto	Cantidad	Importe	Coste	Margen	Observaciones
HARD	ERP1 - PruebaAceros	ERP	CAM	CAMWorks	1	1.000,00 €	500,00 €	500,00 €	
SOFT	ING1 - PruebaAceros	ING	CAD-M	SolidWorks	1	2.000,00 €	1.000,00 €	1.000,00 €	
HARD	ING2 - PruebaAceros	ING	CAD-M	SolidWorks	3	1.000,00 €			

Figura 8.17: CRM: Información sobre entidad Importes

Después de la ejecución del plug-in de ventas la información recopilada de las entidades **oportunidad** e **importes** se almacena en las tablas Oportunidad e Importes respectivamente de la base de datos. En la figuras 8.17 y 8.18 se muestra el resultado obtenido para la oportunidad **pruebaAceros**.

dbo.oportunidad [Datos] | dbo.importes [Datos]

Máximo de filas: 1000

numero	nombre	propietario	numopsiebel	numcliente	nombreciente	provinci...	tipocliente	clasclie...
OPOR-11	pruebaAceros	Alberto Garcia ...	1234	3333	ACEROS PRUEBA 1	Araba	MTO	A
OPOR-12	pruebaAceros2	Alberto Garcia ...	3456	3333	ACEROS PRUEBA 1	Araba	MTO	A
OPOR-13	pruebaAceros3	Alberto Garcia ...	3456	3333	ACEROS PRUEBA 1	Araba	MTO	A
OPOR-14	pruebaAceros4	Alberto Garcia ...	1234	3333	ACEROS PRUEBA 1	Araba	MTO	A

Figura 8.18: BD: Tabla Oportunidad

The image shows a screenshot of a Microsoft Access table view. The window title bar indicates two open tables: 'dbo.importes [Datos]' and 'dbo.oportunidad [Datos]'. The 'dbo.importes' table is currently selected and displayed. The table has a header row with the following columns: 'numoport', 'nombre', 'seccion', 'area', 'producto', 'cantidad', 'coste', and 'importe'. Below the header, there are three data rows. The first row has values: 'OPOR-11', 'ING1 - PruebaAceros', 'SOFT', 'ING', 'SolidWorks', '1', '1.000,00 €', and '2.000,00 €'. The second row has values: 'OPOR-11', 'ERP1 - PruebaAceros', 'HARD', 'ERP', 'CAMWorks', '1', '500,00 €', and '1.000,00 €'. The third row has values: 'OPOR-11', 'ING2 - PruebaAceros', 'HARD', 'ING', 'SolidWorks', '3', and '1.000,00 €'. The 'coste' column is empty for the first and third rows. The interface includes a toolbar with a refresh icon, a filter icon, a sort icon, and a 'Máximo de filas: 1000' dropdown menu.

numoport	nombre	seccion	area	producto	cantidad	coste	importe
OPOR-11	ING1 - PruebaAceros	SOFT	ING	SolidWorks	1	1.000,00 €	2.000,00 €
OPOR-11	ERP1 - PruebaAceros	HARD	ERP	CAMWorks	1	500,00 €	1.000,00 €
OPOR-11	ING2 - PruebaAceros	HARD	ING	SolidWorks	3		1.000,00 €

Figura 8.19: BD: Tabla Importes

# 9

---

---

## 9. *Pruebas*

Según IEEE standards 1999. “El testing de software es el proceso de analizar un producto de software para detectar las diferencias entre el comportamiento real con el esperado.....”. De este modo, para poder asegurar que el software cumple con las especificaciones funcionales es necesario comparar el comportamiento real del software con el esperado, es decir, el que se recoge en los requisitos, y en caso de encontrar alguna desviación en los resultados, por supuesto, corregirlo.

En este proyecto, se han realizado dos tipos de pruebas para asegurar que las aplicaciones que se va entregar cumplen con los objetivos marcados, primeramente **pruebas de sistema** para asegurar que las aplicaciones desarrolladas funcionan de acuerdo a las especificaciones y requisitos funcionales del cliente y **pruebas de aceptación** con el objeto de que sea el mismo cliente (interesados) quién pruebe las aplicaciones y certifique que cumplen con sus expectativas técnicas y funcionales.

Las pruebas realizadas se han centrado principalmente en la aplicación CRM-Marketing. En el caso del plug-in de ventas en el capítulo anterior ya se describe un caso, a modo de ejemplo, de una prueba de funcionamiento (funcionalidad) para comprobar que realmente se comporta como se esperaba. Además, como no hay interacción con el cliente no se realizaron pruebas de aceptación para el plug-in. Así, a partir de ahora cuando nos refiramos a las pruebas realizadas se debe sobreentender que nos estamos refiriendo a las pruebas realizadas a la aplicación CRM-Marketing.

## 9.1 Pruebas de Sistema

---

Las pruebas de sistema realizadas han sido en su totalidad **pruebas funcionales de caja negra**. Las pruebas de caja negra se centran en verificar el correcto funcionamiento de una funcionalidad sin tener conocimiento de la estructura interna y el código que las implementa.

El procedimiento seguido para realizar estas pruebas es definir unos **escenarios de prueba** que son una declaración de lo que vamos a testear. Una vez definidos los escenarios de prueba se crean los **casos de prueba** intentando cubrir todo el escenario de prueba. De este modo, para la aplicación CRM-Marketing se definieron cuatro escenarios de pruebas:

1. Validar el proceso de inicio de la aplicación hasta conectarse al servidor del CRM y la base de datos (TS1).
2. Verificar que se cumplen los criterios de creación de una lista de marketing (TS2).
3. Verificar que se cumplen los criterios para la actualización de los contactos en el CRM (TS3).
4. Validar la funcionalidad *Añadir Credenciales* para asegurarnos que solamente las credenciales de usuarios autorizados se almacenan en el sistema operativo de Windows (TS4).

Las pruebas de sistema realizadas a la aplicación se encuentran recogidas en el anexo B: Pruebas aplicación CRM-Marketing.



En la siguiente tabla 9.1 se muestra un ejemplo: el primer caso de prueba (test case) del escenario 4.

<b>Escenario ID</b>	<b>TS4</b>
<b>Test Case ID</b>	<b>TC1</b>
<b>Descripción</b>	El usuario pulsa el botón [Add Credentials] y confirma. Se carga el formulario “CRM user credentials”. El usuario introduce un nombre de usuario correcto y una contraseña correcta.
<b>Datos</b>	<a href="mailto:a.garcia.benito@ibermatica01.onmicrosoft.com">a.garcia.benito@ibermatica01.onmicrosoft.com</a> y “rightpass”
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación identifica y autentica al usuario.</li> <li>2. Las credenciales del usuario se almacenan en el SO de Windows y se crea el archivo Credentials.xml en el disco local con las configuraciones de los servidores a los que pertenece el usuario.</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla 9.1: Prueba Escenario 4 Caso 1

## 9.2 Prueba Final de Aceptación

---

La persona encargada de realizar las pruebas de aceptación fue **Xabier Philippe Cantero**, ingeniero informático, que era a su vez quién supervisaba el trabajo en Ibermática del autor de este proyecto. Las pruebas de aceptación contaron con varias pruebas “informales” para comprobar que las aplicaciones iban cumpliendo con los requisitos del interesado y una prueba final de aceptación que es la que se va a documentar a continuación.

La prueba final de aceptación consistió en probar el archivo ejecutable de la aplicación en el ordenador de Philippe. Las únicas directrices que le fueron dadas es que intentará probar todas las funcionalidades de la aplicación CRM-Marketing. Durante la prueba se tomaron notas sobre los resultados que iba obteniendo el interesado pero en ningún caso se intervino durante su desarrollo. Sin embargo, tampoco se hicieron capturas de pantalla ni se sacaron fotos.

De esta manera, las imágenes que se van a mostrar son una **simulación** en base a los resultados anotados, aunque lo más aproximada posible no es la prueba de aceptación real.

La prueba se inicia haciendo doble click en el archivo ejecutable de la aplicación **CRM-Marketing**. Como el usuario no tenía sus credenciales almacenadas en su ordenador, al cargarse la pantalla principal se activa el botón [Add Credentials] junto con un mensaje informando que tiene añadir sus credenciales. Al pulsar el botón se abre la ventana “CRM user credentilas” para añadir sus credenciales al sistema.

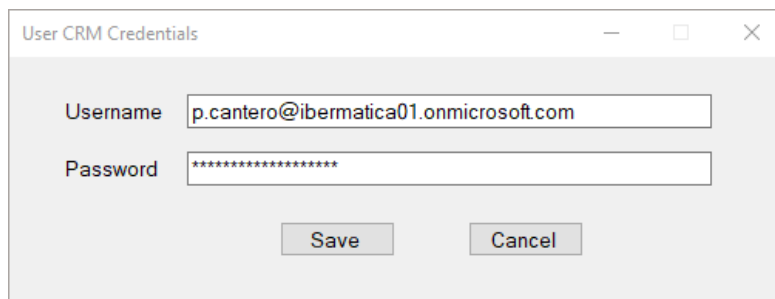


Figura 9.1: App: Vista “User CRM Credentials”

El usuario rellena correctamente sus credenciales y pulsa el botón [Save]. La aplicación identifica y autentica al usuario y luego almacena las credenciales en el SO de Windows. También crea el archivo Credentials.xml con las configuraciones de los servidores.

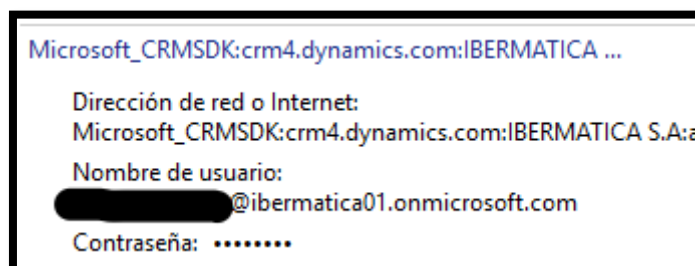


Figura 9.2: Credenciales de Windows

A su vez, se cargan las organizaciones (instancias del CRM) a las que pertenece el usuario en la lista desplegable superior. En la figura 9.3 se muestran las tres organizaciones a las que tiene acceso el usuario.

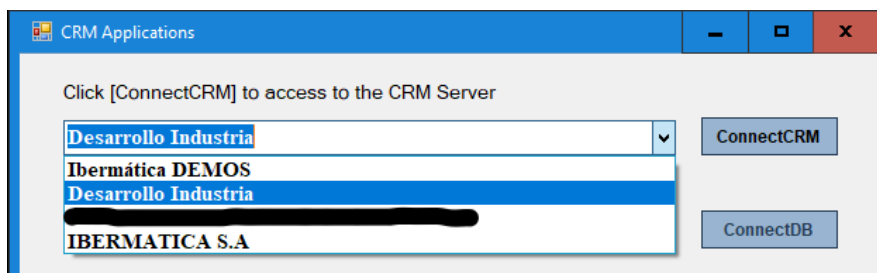


Figura 9.3: App: Selección del servidor de CRM

El usuario selecciona *Desarrollo Industria* y se conecta al servidor del CRM. Al instante se habilita el botón [ConnectDB] y el usuario pulsa el botón para conectarse a la base de datos. La aplicación carga las listas de mailing disponibles y habilita el botón [Run]. El usuario selecciona la lista denominada LISTA\_CAD\_CON.

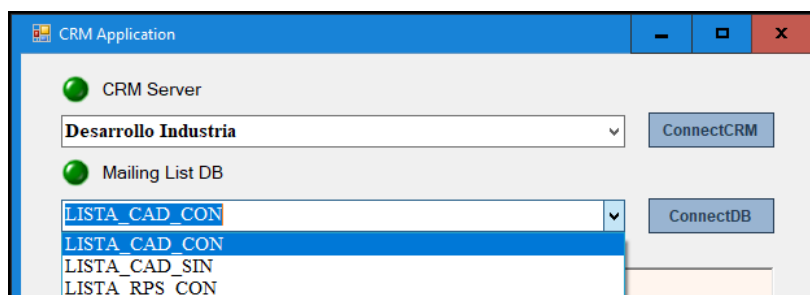


Figura 9.4: App: Selección de la lista de mailing

Después selecciona la opción *Create a Log File* y pulsa el botón [Run]. La aplicación pide confirmación en un cuadro de diálogo para la creación de la lista de marketing.

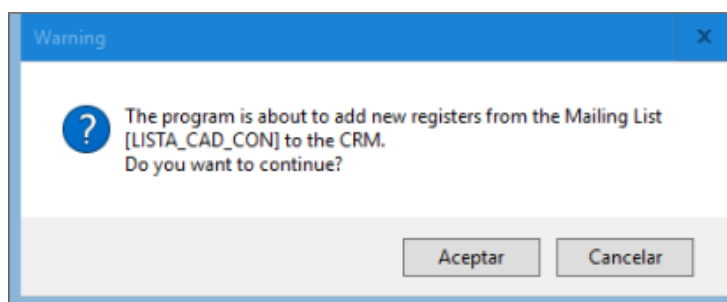


Figura 9.5: App: Confirmación de la lista de marketing

Finalmente, el usuario confirma la creación de la lista de marketing. El resultado de la creación de la lista de marketing se muestra por pantalla y se crea un archivo de log con la misma información. El usuario pulsa el botón [Open Log File] para verificar el contenido del mismo.

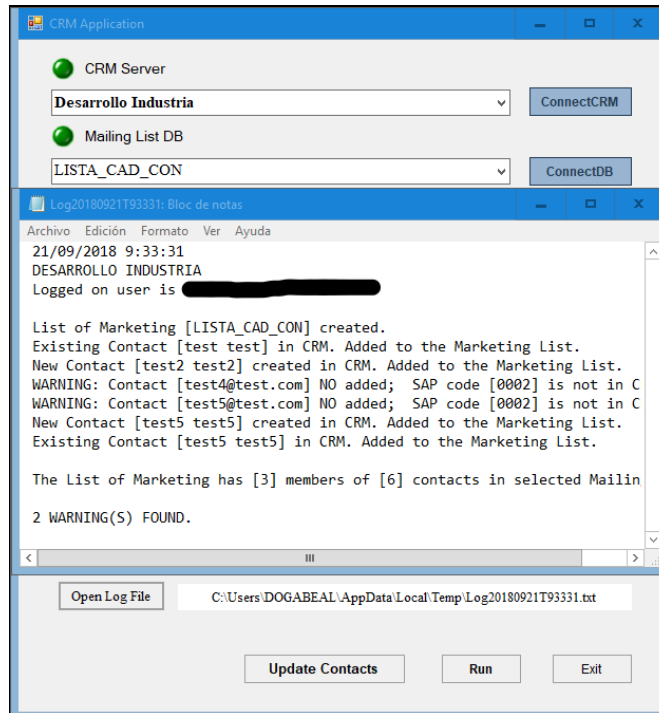


Figura 9.6: App: Archivo Log de la lista de marketing

Después, el usuario pulsa el botón [Exit] para abandonar la aplicación. Sin embargo, no da la prueba de aceptación por acaba si no que arranca otra vez el archivo ejecutable de la aplicación. En esta ocasión, como el usuario ya tiene sus credenciales almacenadas en el sistema operativo de Windows la aplicación carga automáticamente las organizaciones en el desplegable superior de la pantalla de inicio. El usuario prosigue todo el proceso con éxito para probar la otra funcionalidad: *actualizar los contactos del CRM*. En este caso, el usuario pulsa el botón [Update Contacts] que se habilita a igual que el botón [Run] después de conectarse a la base de datos. El usuario confirma la ejecución de la funcionalidad, los contactos se actualizan y el resultado se muestra por pantalla.

En este punto, el usuario da por finalizada con éxito la prueba de aceptación de la aplicación CRM-Marketing.

# 10

---

---

## **10. Gestión**

En este capítulo, se relata la gestión realizada de las principales áreas de conocimiento del proyecto como son la gestión del alcance, gestión del cronograma, gestión de costos, gestión de la calidad, gestión de riesgos y gestión de los interesados.

## 10.1 Gestión del Alcance

---

En la propuesta inicial para este proyecto no se recoge un alcance definido ni cerrado. Principalmente, se debe al uso de nuevas tecnologías en Ibermática, la herramienta software Microsoft Dynamics CRM Online es un producto relativamente nuevo, se introdujo hace dos años, y el kit de desarrollo software (SDK) para Microsoft Dynamics 365 nunca se había empleado antes en la empresa. De esta forma, era muy difícil definir a priori un alcance concreto sin conocer las posibilidades de las tecnologías y menos aún hacer una estimación correcta del tiempo necesario para desarrollar el producto.

El alcance inicial de este proyecto propone el desarrollo de tres funcionalidades para Microsoft Dynamics CRM Online para su integración posterior con la plataforma de soporte VDoc. El orden de prioridades es el que aparece a continuación:

1. Creación automática de listas de marketing en el CRM.
2. Sincronización de contactos entre el VDoc y el CRM.
3. Desencadenamiento del flujo de registro de ventas en el VDoc desde el CRM.

Sin embargo, el alcance mínimo del proyecto solamente incluía el desarrollo de la primera funcionalidad y las otras dos funcionalidades solamente se abordarían una vez concluida con éxito la primera. Tampoco se definía que tipo de solución se tenía implementar para desarrollar cada funcionalidad, la decisión de implementar dos soluciones diferentes: la aplicación CRM-Marketing y el plug-in de ventas, es posterior. En general, se puede decir que las inclusiones del alcance se fueron refinando durante el ciclo de vida del proyecto mientras que las exclusiones si estaban claras desde el primer momento, en concreto, la exclusión de la parte del desarrollo de la conexión de las bases de datos con la plataforma de soporte VDoc.

De este modo, durante el desarrollo del proyecto se fue ampliando el alcance y se fueron incluyendo las otras dos funcionalidades. Sin embargo, si se produjo un cambio de prioridades y pasó a ser más importante finalizar completamente el plug-in de ventas que implementa la tercera funcionalidad a la realización de mejoras para la segunda funcionalidad.

El final de todo este proceso da lugar al alcance final del proyecto que es el que se expone en el apartado *Alcance* del capítulo de *Objetivos*.

## 10.2 Gestión del Cronograma

En el diagrama Gantt se muestran los periodos de realización de los paquetes de trabajo durante el ciclo de vida de este proyecto.

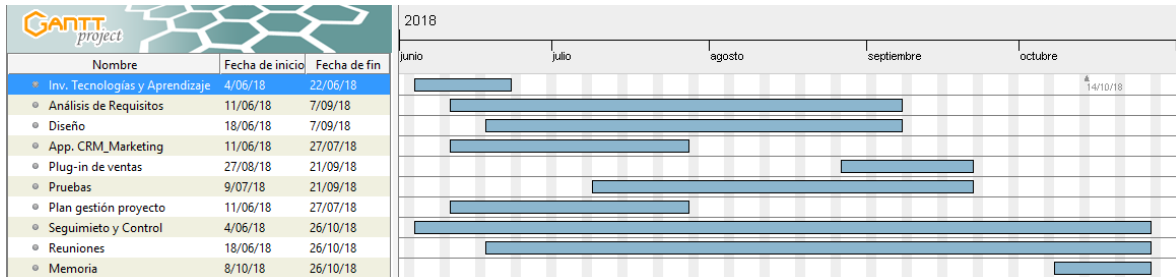


Figura 10.1: Diagrama de Gantt

A continuación, se indican las fechas de algunos hitos del proyecto:

- **4 de junio:** Comienza el proyecto
- **18 de junio:** Primera Reunión de Seguimiento y Control.
- **13 de julio:** Se presenta la primera versión de la aplicación CRM-Marketing. Se redefine la arquitectura de la aplicación.
- **16 de julio:** Segunda Reunión de Seguimiento y Control.
- **26 de julio:** Se finaliza la segunda versión de la aplicación CRM-Marketing
- **30 de julio:** Inicio vacaciones.
- **27 de agosto:** Fin de vacaciones.
- **28 de agosto:** Se realiza prueba de aceptación de la aplicación CRM-Marketing.
- **11 de septiembre:** Tercera Reunión de Seguimiento y Control.
- **19 de septiembre:** Finalización del plug-in de ventas.
- **21 de septiembre:** Fin de proyecto (en Ibermática).

### 10.3 Gestión de Costes

---

El coste del proyecto se mide en horas de trabajo dedicadas a su realización, es decir, el coste humano. Se han trabajado 12 semanas en Ibermática a 6h por día (360 h), sin embargo se han descontado 25h de trabajo realizado porque se trataban de tareas no relacionadas con este proyecto. Además, también se incluyen las reuniones realizadas con el director del proyecto en la UPV/EHU y el tiempo académico empleado en realizar la memoria y su defensa.

<b>Categoría</b>	<b>Tarea</b>	<b>Dedicación (h)</b>
Investigación tecnologías y Aprendizaje	IDEM	42
Producto	Análisis de requisitos	15
	Diseño	21
	Implementación Aplicación	145
	Implementación Plug-in	75
	Pruebas	13
Gestión	Plan de gestión Proyecto	12
	Seguimiento y Control	12
	Reuniones	7
Académico	Memoria	95
	Defensa	10*
<b>TOTAL</b>		<b>447</b>

Tabla 10.1: Tabla de dedicaciones de tiempo



## 10.4 Gestión de la Calidad

---

La calidad mínima en el producto será el cumplimiento de la especificaciones y exigencias impuestas por los interesados, es decir, que el producto funcione correctamente y que contenga todas las funcionalidades recogidas en este proyecto.

### 10.4.1 *Aseguramiento y control de la calidad*

El aseguramiento y control de la calidad se realiza mediante pruebas de software de las funcionalidades implementadas, donde se verifica que realmente el producto realiza las funciones que se han definido, sin producirse errores en la ejecución y/o en los resultados obtenidos, y cumpliendo los criterios de calidad establecidos.

### 10.4.2 *Criterios de calidad*

Se establecen los siguientes criterios para medir la calidad del producto:

1. **Seguridad del producto:** Se deberá implementar un sistema de control de acceso a las aplicaciones que permita el acceso solamente a los usuarios que se hayan identificado y autenticado correctamente. Se deberá garantizar la confidencialidad de los datos más críticos, en especial, las contraseñas. Otras medidas tomadas para asegurar la confidencialidad de los datos más sensibles se considerarán un plus de calidad.
2. **Usabilidad de la interfaz gráfica:** La interfaz será simple, clara y fácil de usar. Se mostrarán mensajes por pantalla que faciliten al usuario su uso, véase, instrucciones de uso, mensajes de información, de ayuda y de error. En lo posible, deberá guiar al usuario durante la ejecución de una funcionalidad y solamente estarán disponibles (habilitados y/o visibles) los elementos de la interfaz necesarios en cada momento, para evitar errores en su ejecución.
3. **Mantenibilidad del código:** La programación del software se realizará siguiendo la arquitectura en tres capas: capa de presentación, capa de negocio y capa de datos. De esta forma, en caso de que se produzca algún cambio en el diseño de la aplicación solamente afectará al nivel requerido sin tener que revisar el código fuente de otras capas. Se documentará el código (clase, funciones y partes del código que necesiten aclaración) para facilitar el entendimiento y reutilización del código de programación.

## 10.5 Gestión de Riesgos

---

Según la *Guía del PMBOK – Quinta Edición del Project Management Institute*: “los objetivos de la gestión de los riesgos del proyecto consisten en aumentar la probabilidad y el impacto de los eventos positivos, y disminuir la probabilidad y el impacto de los eventos negativos en el proyecto sobre el alcance, el costo, el cronograma, la calidad o el desempeño del proyecto”.

### 10.5.1 Identificación de riesgos y relevancia

Las principales fuentes de riesgos que se han identificado en este proyecto son:

1. **Realización del proyecto en periodo estival:** La mayor parte del proyecto se realizará en los meses de verano, durante los cuales los principales interesados estarán de vacaciones. Esto representa un problema a la hora de redefinir los requisitos de las aplicaciones y supervisar los trabajos realizados. (probabilidad: alta; impacto: medio).
2. **Uso de tecnologías desconocidas (SDK .NET):** Se va a utilizar el kit desarrollo de software para Microsoft Dynamics 365 que es la primera vez que se utiliza en Iberoamérica. Por tanto, se precisa de un aprendizaje inicial autodidacta sin contar con el apoyo de otras personas. Este hecho puede influir negativamente en el alcance, el costo y/o el desempeño del proyecto (probabilidad: media; impacto: medio).
3. **Modificación de los requisitos:** los interesados podrían modificar, agregar o eliminar sus requisitos iniciales en mitad del proyecto y, así, retrasar el desarrollo del proyecto o crear una pérdida de tiempo significativa. (probabilidad: medio; impacto: medio).
4. **Perdida de información:** La pérdida de información podría causar retrasos significativos en el proyecto. El material que se realizará durante el proyecto no solo será necesario para llevar a cabo el proyecto en sí, sino que debe incluirse en la memoria del TFG. En este proyecto, especialmente crítica sería la pérdida del código software de las aplicaciones implementadas. (probabilidad: baja; impacto: alto).
5. **Problemas personales:** Enfermedades, lesiones o cualquier situación relacionada con los familiares podrían originar retrasos en el proyecto. (probabilidad: baja; impacto: alta).

### 10.5.2 *Plan de acción*

Las siguientes medidas se toman para disminuir la probabilidad de que se produzca un daño o en su caso, para mitigar el impacto del mismo:

1. **Realización del proyecto en periodo estival:** Aunque inicialmente se había previsto que el proyecto se iba a realizar durante los meses junio, julio y agosto, finalmente, y tras hablarlo con los interesados, se decidió que la mejor opción era detener el proyecto en el mes de agosto y retrasar su finalización al mes de septiembre.
2. **Uso de tecnologías desconocidas (SDK .NET):** En este apartado, tras hablarlo con el director del producto por parte de Ibermática, Xabier Philippe Cantero, se acuerda dejar abierta la posibilidad de modificar el alcance del proyecto y el estudio de otras soluciones diferentes si no fuera posible cumplir todos los requisitos.
3. **Modificación de los requisitos:** Para mitigar este factor se ha intentado realizar una recopilación de requisitos iniciales lo más precisa posible de acuerdo a las necesidades de los interesados.
4. **Perdida de información:** Además de guardar la información en el equipo local, el código fuente se almacenará en un repositorio de la plataforma GIT de Ibermática y la documentación en una carpeta en red creada con este propósito. Independientemente de estos dos sistemas de almacenamiento de la información el alumno realizará copias periódicas de los avances del proyecto.
5. **Problemas personales:** Si surgiera un problema grave, se detendrá la realización del proyecto hasta que se resuelva. Esto produciría un retraso justificable en el proyecto y se debería re-planificar el mismo.

## 10.6 Gestión de los Interesados

---

En este proyecto las personas interesadas son:

- **Alumno:** Alberto García Benito
- **Director (Ibermática):** Xabier Philippe Cantero Berrouet
- **Codirector (UPV/EHU):** José Miguel Blanco Arbe

Además, dentro del grupo de interesados se encontrarían los **trabajadores comerciales** de Ibermática que son las personas que van a ser los usuarios finales del producto.

El director por Ibermática, Xabier Philippe Cantero Berrouet, es la persona encargada de definir los requisitos iniciales del producto, de realizar un seguimiento y supervisar el trabajo realizado y en última instancia de aceptar el producto final confirmando que cumple los requisitos encomendados. Se podría referir a él como el **director del producto**.

El director por la UPV/EHU, José Miguel Blanco Arbe, es la persona encargada de supervisar el desempeño del alumno durante la realización del proyecto, de dirigir y aconsejar al alumno y en última instancia de aceptar la memoria del proyecto como válida. Se podría referir a él como el **director del proyecto**.

Los **trabajadores comerciales**, si bien son los usuarios finales de las aplicaciones, no se tiene ninguna comunicación directa con ellos y cualquier tipo de comunicación se realiza través del director del producto.

# 11

---

---

## **11. Conclusiones**

El objetivo establecido para este proyecto era desarrollar varias funcionalidades para Microsoft Dynamics CRM Online para su integración con la plataforma soporte BPM VDoc y ha dado lugar a la creación de dos aplicaciones software: la aplicación CRM-Marketing y el plug-in de ventas.

La **aplicación CRM-Marketing** va posibilitar, por una parte, la creación de listas de marketing basadas en la información proveniente de la plataforma VDoc y por otra, la actualización de los contactos del CRM con datos de contactos provenientes de VDoc. Por otra parte, el **plug-in de ventas** desencadena automáticamente el flujo de registro de ventas cuando se cierra una venta en el CRM (oportunidad ganada). Recuérdese que el desarrollo de la parte que conecta la base de datos con la plataforma VDoc queda fuera del alcance de este proyecto.

De esta forma, si el objetivo mínimo de este proyecto es el cumplimiento de las especificaciones y exigencias impuestas por los interesados, es decir, que el producto final funcione correctamente y que contenga todas las funcionalidades recogidas en el alcance, entonces, podemos asegurar que el objetivo mínimo marcado se ha cumplido.

Sin embargo, son los requisitos no funcionales los que aportan un valor extra de calidad al producto entregado. En este punto, durante el desarrollo de las aplicaciones el mayor esfuerzo se ha centrado en implementar soluciones seguras. El control del acceso a la aplicación CRM-Marketing se ha conseguido gestionando las credenciales de los usuarios como credenciales de Windows, descartando otras soluciones como la creación de una base de datos con los nombres de usuario y contraseñas puesto que esto implicaba una gestión y mantenimiento de la base de datos. De esta forma y junto con los métodos que ya aportaba el SDK de Microsoft Dynamics 365 se ha conseguido asegurar la identificación y autenticación de los usuarios para restringir su acceso solamente a los usuarios autorizados. En el caso del plug-in se creó un servicio WCF para evitar conectarse directamente desde el CRM Online (puerto 80) que está en la red a la base de datos y a su vez poder añadir alguna medida extra de autenticación al servicio (ID).

El aseguramiento de la confidencialidad de algunos datos críticos, como la contraseña del usuario, también se tuvo en cuenta. Las contraseñas siempre se guardan encriptadas pero no solo eso, la encriptación de la contraseña se hace desde el momento que el usuario introduce su valor en el formulario como cuando añade las credenciales la primera vez que usa la aplicación. Otras medidas de seguridad como la encriptación de las cadenas de conexión a las bases de datos almacenadas en los archivos de configuración no se han implementado por falta de tiempo.

La sencillez y facilidad de uso (usabilidad) de las aplicaciones ha sido otro factor en el que se ha trabajado mucho. La solución del plug-in de ventas se podría decir que es el sumo de la usabilidad ya que realiza la tarea automáticamente sin interacción humana. Respecto a la aplicación CRM-Marketing la interfaz gráfica de la vista principal, junto con un diseño simple, funciona de forma que solamente están activos o visibles los botones que se deben usar en ese momento, para evitar errores y guiar al usuario en el proceso de creación de la lista de marketing. Incluye también un sistema de información que muestra instrucciones, informa del estado actual e incluye mensajes de confirmación, ayuda y error.

La aplicación CRM-Marketing se ha programado siguiendo el modelo de programación en tres capas, que separa, de una forma lógica, la capa de presentación, la capa de negocio y la capa de datos. De esta forma se asegura la capacidad de mantenimiento del software pues se puede, por ejemplo, cambiar la interfaz gráfica sin afectar a las otras capas. Por otra parte, las cadenas de conexión a las bases de datos se incluyen en un archivo de configuración tanto por motivos de seguridad como de mantenibilidad del software, si se cambia la cadena de conexión no hace falta compilar la aplicación otra vez.

En el caso de la aplicación CRM-Marketing una característica de calidad que hubiera sido interesante tratar y no se ha podido realizar es medir la eficiencia de la aplicación. La aplicación se ha implementado en el entorno de desarrollo (aislado) con pocos datos de prueba. Sin embargo, en el caso de la funcionalidad realizada para *crear listas de marketing*, las listas pueden contener más de mil contactos y esa cantidad ya la considero lo suficientemente significativa como para comprobar su comportamiento ante esa situación.

Por otra parte, para poder realizar estas aplicaciones se han tenido que utilizar nuevas tecnologías incluidas en el framework de .NET y utilizar un nuevo lenguaje de

programación C#. Además, también ha sido necesario aprender el funcionamiento de Microsoft Dynamics CRM Online a nivel de usuario administrador y conocer su estructura interna usando ingeniería inversa.

La principal novedad ha sido el uso del kit de desarrollo (SDK) para Microsoft Dynamics 365 que se ha tenido que aprender de forma absolutamente autodidacta puesto que nadie en Ibermática lo había usado antes.

Finalmente, ya desde un punto de vista más personal, también quiero comentar que se ha aprendido a no entrar en pánico y a identificar los errores a partir del tipo excepción y su descripción, que las cosas pueden funcionar bien en el entorno aislado y luego no cuando se pasa a producción y que el tópico “pues en mi ordenador funciona” está muy vigente.

Como conclusión final, teniendo en cuenta todo lo expuesto considero que sí que efectivamente se han alcanzado y superado los objetivos inicialmente marcados

## 11.1 Propuestas de mejora y líneas de futuro

---

Algunas de las propuestas de mejora ya se han mencionado implícitamente en el apartado anterior. En concreto, se pueden implantar mejoras relacionadas con la seguridad de las aplicaciones como la encriptación de los datos más críticos recogidos en los archivos de configuración.

La funcionalidad *Sincronizar contactos entre el VDoc y el CRM* se ha realizado en la misma aplicación que la primera funcionalidad *Crear listas de marketing* por la similitud de los datos utilizados y para aprovechar los métodos ya implementados pero, se podría estudiar si es mejor incluirla en una nueva vista o estudiar la idoneidad de separar la información en bases de datos diferentes.

Las pruebas de rendimiento de la aplicación CRM-Marketing sería un factor de mejora a tener en cuenta, sobre todo, si los resultados obtenidos no fueran óptimos.

Otras propuestas de mejora incluirían la implementación de nuevas funcionalidades que ahonden en la integración entre las dos plataformas, Microsoft Dynamics CRM Online y BPM VDoc On-premises. Aunque en este punto no tengo la información ni el conocimiento suficiente de la plataforma de soporte VDoc como para identificar esas necesidades.

Finalmente, los siguientes pasos que habría que dar parecen evidentes y sería completar el presente proyecto desarrollando la conexión entre las bases de datos y la plataforma de soporte VDoc.

## 11.2 Lecciones Aprendidas

---

Las lecciones aprendidas son toda información valiosa que se deriva de la experiencia obtenida durante la realización de este proyecto y que, en el futuro, sirva para evitar errores que se han cometido y oportunidades de mejora para situaciones similares (buenas prácticas). A continuación desarrollaremos algunas de las lecciones más relevantes aprendidas durante este proyecto.

### 11.2.1 Disponibilidad de las fuentes de información

Un factor que debemos tener en cuenta a la hora de elegir un tipo de tecnología u otro para desarrollar un producto es la disponibilidad de las fuentes de información relacionadas con esa tecnología.

Por ejemplo, este factor influyó en la elección del servicio web utilizado para conectarse al servidor del CRM, a favor del Servicio de Organización respecto a la Web API, pues al ser el servicio más utilizado por los desarrolladores de Microsoft Dynamics 365 se consideró que sería más fácil encontrar información que sirviera de guía y ayuda. Hay que tener en cuenta que en este proyecto se utilizaban tecnologías novedosas en Ibermática y no se tenía ningún conocimiento previo de las mismas.

En general, se puede afirmar que la información sobre la plataforma Microsoft Dynamics CRM Online y, en concreto, sobre el SDK de Microsoft Dynamics es amplia y clara. Toda la información sobre los productos de Microsoft se encuentra agrupada en un solo sitio web: <https://docs.microsoft.com/en-us/> y en particular, se hizo uso de la guía para desarrolladores para Microsoft Dynamics 365: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/developer-guide>.

Además, Microsoft pone a nuestra disposición un foro propio y específico para Microsoft Dynamics 365: <https://community.dynamics.com/>.

Por último, también me gustaría comentar que la mayoría de la información de los foros sobre los productos de Microsoft se encuentra en **inglés**, por lo que un conocimiento medio de este idioma es muy recomendable.



### 11.2.2 Seguridad de los Servicios Web

Los servicios web se usan principalmente para poder intercambiar datos entre aplicaciones informáticas implementadas en diferentes lenguajes de programación y ejecutadas en plataformas diferentes.

Sin embargo, también he descubierto que es **más seguro** acceder la información de un servidor o a una base de datos a través de un servicio web que abriendo una conexión directa. Además, el servicio web te permite implementar medidas extra de seguridad dirigidas a garantizar la autenticación de los usuarios del servicio y la integridad y confidencialidad de los datos y así, evitar exponerse a problemas de seguridad, pérdida o robo de información, etc.

Un pequeño ejemplo se ha implementado en el plug-in de ventas que se conecta a la base de datos a través de un servicio WCF que hemos implementado. Este servicio incorpora un método simple de autenticación mediante la verificación con un ID (contraseña).

### 11.2.3 Documentación del código

Documentar el código de un programa no solamente sirve para que el código sea más legible y fácil de entender si no que también se utiliza para crear documentación del código de forma automática.

El lenguaje de programación C# tiene ya estandarizado la forma de documentar el código para poder crear archivos XML con los comentarios. El código comentado requiere de delimitadores `“///”`, `“/** */”` al inicio de cada clase o método junto con las etiquetas XML como `<summary>`, `<param>`, `<returns>` que identifican que se está comentado.

Una vez se ha documentado todo el programa se tiene que compilar con la sentencia: **“-doc:file “**. Así, se obtiene el archivo XML *file* que incluye los comentarios de todas las etiquetas XML de los archivos del código fuente de la compilación.

Finalmente, con la herramienta **Sandcastle** de Microsoft podemos generar la documentación final a partir del archivo XML generado en la compilación.

#### 11.2.4 *Kit de desarrollo software (SDK) para Microsoft Dynamics 365*

Este kit proporciona un conjunto de bibliotecas y herramientas que permite a los desarrolladores la creación de aplicaciones para Microsoft Dynamics 365. El SDK contiene una visión del **conjunto de la arquitectura de Microsoft Dynamics 365**, el modelo de entidades, el modelo de seguridad, los servicios web y ejemplos de muestra de código.

La capacidad para crear nuevas funcionalidades personalizadas para Microsoft Dynamics 365 en función de las necesidades específicas y deseos del cliente ya ha quedado más que probada durante este proyecto. El kit ofrece distintas soluciones que se adaptan mejor al desarrollador en función del entorno de programación (.NET o no .NET), del tipo de producto a desarrollar ya sean plug-ins, actividades de flujos de trabajo o flujos de trabajo XAML personalizados o sean aplicaciones para Windows o no.

El SDK incluye un conjunto de herramientas específicas para el desarrollo de aplicaciones de Microsoft Dynamics 365. Por ejemplo, durante este proyecto hemos empleado la herramienta de registro de plug-ins incluida en el Kit para registrar (añadir) nuestro plug-in en el CRM.

Además, el kit también proporciona ejemplos de muestra de código muy bien documentados y que sirven como base para empezar a desarrollar aplicaciones personalizadas para Microsoft Dynamics 365. A su vez, también incluye una guía para desarrolladores de Dynamics 365 muy completa: <https://docs.microsoft.com/es-es/dynamics365/customer-engagement/developer/developer-guide> .

## Bibliografía y Referencias

---

- [1] Ibermática. Presentación corporativa 2018.
- [2] Wikipedia. Programación por capas.  
URL: [https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_por\\_capas](https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas)
- [3] Wikipedia. Microsoft .NET.  
URL: [https://es.wikipedia.org/wiki/Microsoft\\_.NET](https://es.wikipedia.org/wiki/Microsoft_.NET)
- [4] Microsoft (docs). Windows Communication Foundation.  
URL: <https://docs.microsoft.com/es-es/dotnet/framework/wcf/whats-wcf>
- [5] Microsoft (docs). Software Development Kit for Microsoft Dynamics 365 (online) and Dynamics 365 (on-premises). URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/hh547453\(v=crm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/hh547453(v=crm.8))
- [6] Microsoft (docs). Use Microsoft Dynamics 365 web services.  
URL: <https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/mt608128%28v%3dcrm.8%29>
- [7] Microsoft (docs). Use Microsoft Dynamics 365 services in code.  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309557\(v%3dcrm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309557(v%3dcrm.8))
- [8] Microsoft (docs). Discover the URL for your organization using the Organization Service.  
URL: <https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328127%28v%3dcrm.8%29>
- [9] Microsoft (docs). Sample: Quick start for Dynamics 365 Customer Engagement. URL: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/sample-quick-start>
- [10] Microsoft (docs). Helper code: ServerConnection class.  
URL: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/org-service/helper-code-serverconnection-class>
- [11] Microsoft (docs). Sample: Use the organization service context.  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328029\(v%3dcrm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328029(v%3dcrm.8))
- [12] Microsoft (docs). Sample: Create, retrieve, update, and delete records (early bound)  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg334754\(v=crm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg334754(v=crm.8))
- [13] Microsoft (docs). Sample: Associate records (early bound) URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309247\(v%3dcrm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309247(v%3dcrm.8))

- [14] Microsoft (docs). List (marketing list) entity. Sample: Distribute a quick campaign.  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2013/developers-guide/gg328526\(v%3dcrm.6\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2013/developers-guide/gg328526(v%3dcrm.6))
- [15] Microsoft (docs). Use the QueryExpression class. URL: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/org-service/use-queryexpression-class>
- [16] Microsoft (docs). Use the QueryByAttribute class.  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328259\(v%3dcrm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328259(v%3dcrm.8))
- [17] Dynamics Community. Add an Entity (Member) to a Marketing List in Microsoft Dynamics.  
URL: <https://community.dynamics.com/crm/b/mileyja/archive/2011/11/14/add-an-entity-member-to-a-marketing-list-in-microsoft-dynamics-crm-2011-using-net-or-jscript>
- [18] Microsoft (docs). Understand the data context passed to a plug-in.  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309673\(v%3dcrm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309673(v%3dcrm.8))
- [19] Microsoft (docs). Developing Service-Oriented Applications with WCF.  
URL: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/>
- [20] Microsoft (docs). Write a plug-in.  
URL: [https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328263\(v=crm.8\)](https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg328263(v=crm.8))
- [21] Microsoft (docs). How to: Define a Windows Communication Foundation Service Contract.  
URL: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/how-to-define-a-wcf-service-contract>
- [22] Microsoft (docs). Using Data Contracts. URL: <https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/using-data-contracts>
- [23] Microsoft (docs). Register and Deploy Plug-Ins. URL: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/register-deploy-plugins>
- [24] Microsoft (docs). Walkthrough: Register a plug-in using the plug-in registration tool.  
URL: <https://docs.microsoft.com/en-us/previous-versions/dynamicscrm-2016/developers-guide/gg309580%28v%3dcrm.8%29>

## Anexo A: Guía para el Registro del Plug-in de Ventas

---

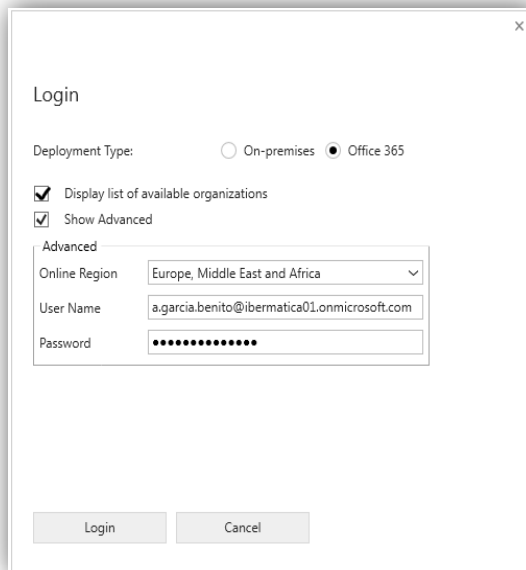
Esta guía muestra que pasos hay que seguir para registrar el plug-in de ventas mediante la herramienta “**Plugin Registration Tool**” para Microsoft Dynamics CRM Online. Se divide en tres etapas:

- Conectarse al servidor de Dynamics 365
- Registrar el ensamblado del plug-in de ventas
- Registrar el plug-in de ventas para el evento “Win opportunity”

### Conectarse al servidor de Dynamics 365

Los pasos a seguir para conectarse al servidor de Dynamics 365 son:

1. Ejecute la herramienta “**Plug-in Registration Tool**”.
2. Pulse en + CREATE NEW CONNECTION (en la parte superior izquierda).
3. Seleccione las opciones *Office 365*, *Display list of available organizations* y *Show Advanced*.
4. Seleccione la Región Online de *Europe, Middle East and Africa* e introduzca su nombre de usuario (User Name) y contraseña (Password).
5. Pulse el botón [Login] y se le mostrará una lista con todas las organizaciones a las que pertenece el usuario. Seleccione la organización en la que quiere registrar el plug-in.

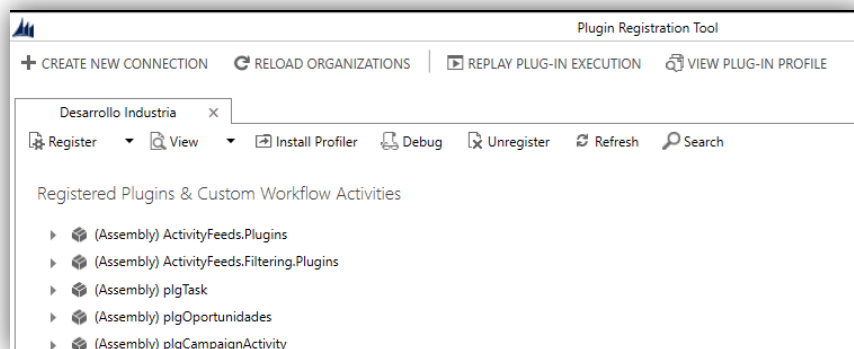


Vista de Login para CRM Online

## Registrar el ensamblado del plug-in de ventas

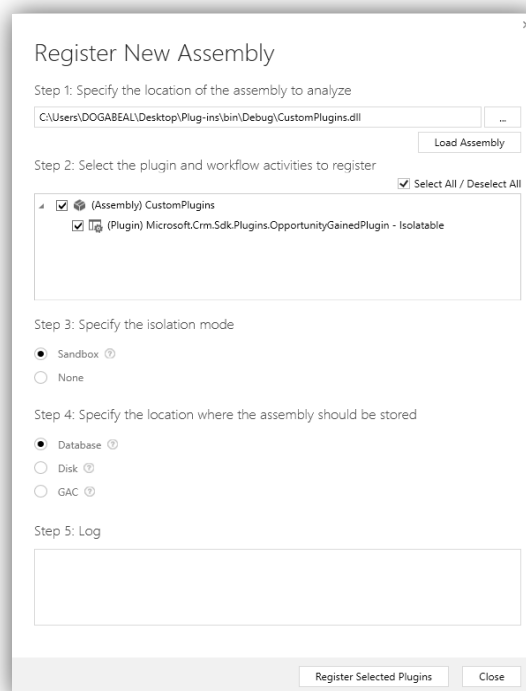
Los pasos a seguir para registrar el ensamblado del plug-in de ventas son:

1. Pulse en *Register* y después en *Register New Assembly*.



Vista principal de la aplicación

2. Se abrirá el formulario “Register new Assembly”.
3. STEP1: Haga clic en el botón [...] y busque la ubicación del ensamblado CustomPlugins.dll. Seleccione el ensamblaje y pulse el botón [Abrir].
4. STEP2: Expanda el ensamblado CustomPlugins y seleccione el OpportunityGainedPlugin (el ensamblado puede tener varios plug-ins).
5. STEP3: Seleccione *Sandbox*.
6. STEP 4: Seleccione *Database*.
7. Pulse el botón [Register Selected Plugins].



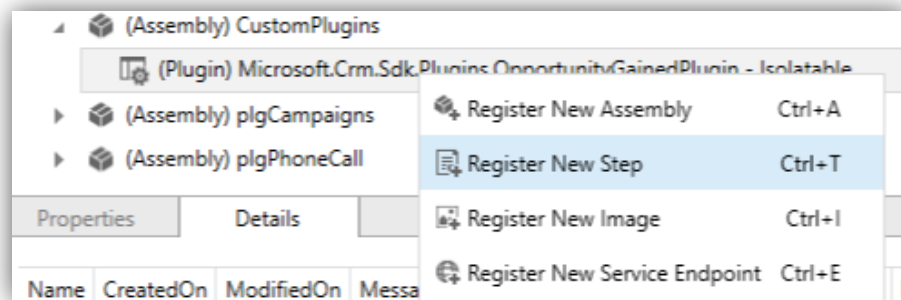
F

Formulario para registrar un ensamblado

## Registrar el plug-in de ventas para el evento “Win opportunity”

Los pasos a seguir para registrar el plug-in de ventas para el evento “Win opportunity” son:

1. En la ventana principal de “Plug-in Registration tool” vaya a la sección *Registered Plug-ins & Custom Workflow Activities*. En esa sección expanda el ensamblado de CustomPlugins.dll. y seleccione el plug-in OpportunityGained.
2. Vaya al menú *Register* de la barra de herramientas y seleccione *Register New Step*. Opcionalmente con el botón derecho del ratón también se abrirá el menú *Register*.



Vista del menu desplegable *Register*

3. Se abrirá el formulario UPDATE EXISTING STEP. Rellene el formulario como se muestra en la imagen.
4. Pulse el botón [Update Step].

Formulario para registrar *New Step*

Al finalizar el proceso, el plug-in de ventas estará registrado para ejecutarse con el evento “Win opportunity” (ganar una oportunidad), después de la ejecución del programa principal y en modo asíncrono.



## Anexo B: Pruebas aplicación CRM-Marketing

Escenario ID	Descripción
<b>TS1</b>	Validar el proceso de inicio de la aplicación hasta conectarse al servidor de CRM y la base de datos.
<b>TS2</b>	Verificar que se cumplen los criterios de creación de una lista de marketing: <ol style="list-style-type: none"> <li>1. En función del Nº de listas de marketing en el CRM.</li> <li>2. En función del tipo de contacto a agregar.</li> </ol>
<b>TS3</b>	Verificar que se cumplen los criterios para la actualización de los contactos en el CRM.
<b>TS4</b>	Validar la funcionalidad "Add Credentials" para asegurarnos que solamente las credenciales de usuarios autorizados se almacenan en Windows.

Tabla B.1: Tabla de Escenarios

Escenario ID	TS1
Test Case ID	TC1
Descripción	Las credenciales de usuario están almacenadas en Windows. Existe el archivo Credentials.xml con las configuraciones de los servidores CRM. El usuario arranca la aplicación.
Datos	--
Resultado Esperado	<ol style="list-style-type: none"> <li>1. La aplicación muestra la pantalla principal de la aplicación (CRM Applications).</li> <li>2. La aplicación identifica y autentica al usuario, valida las configuraciones de los servidores CRM.</li> <li>3. La aplicación carga todas las organizaciones (instancias del CRM) a las que pertenece el usuario. El botón [ConnectCRM] se habilita. Se muestran las instrucciones de uso de la aplicación.</li> <li>4. El usuario selecciona la organización y pulsa el botón [ConnectCRM].</li> <li>5. La aplicación se conecta al servidor del CRM. Muestra un Led verde. Se habilita el botón [ConnectDB] y se deshabilita el botón [ConnectCRM].</li> <li>6. El usuario pulsa el botón [ConnectDB].</li> <li>7. La aplicación se conecta a la base de datos y carga las listas de mailing. Muestra un Led verde. Se deshabilita el botón [ConnectDB] y se habilitan los botones [Run] y ([Update Contacts]).</li> </ol>
Resultado Obtenido	IDEM RESULTADO ESPERADO
Calificación	SUPERADO

Tabla B.2: Prueba Escenario 1 Caso 1

<b>Escenario ID</b>	<b>TS1</b>
<b>Test Case ID</b>	<b>TC2</b>
<b>Descripción</b>	Las credenciales de usuario NO están almacenadas en Windows. El archivo Credentials.xml NO existe. El usuario arranca la aplicación.
<b>Datos</b>	--
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación muestra la pantalla principal de la aplicación (CRM Applications).</li> <li>2. La aplicación muestra un mensaje de error: "No CRM user credentials saved in disk"</li> <li>3. La aplicación habilita el botón [Add Credentials].</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.3: Prueba Escenario 1 Caso 2

<b>Escenario ID</b>	<b>TS1</b>
<b>Test Case ID</b>	<b>TC3</b>
<b>Descripción</b>	Las credenciales de usuario están almacenadas en Windows. Existe el archivo Credentials.xml con las configuraciones de los servidores CRM Las credenciales de usuario y/o las configuraciones de los servidores son erróneas. El usuario arranca la aplicación.
<b>Datos</b>	--
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación muestra la pantalla principal de la aplicación (CRM Applications).</li> <li>2. La aplicación muestra un mensaje de error: "ERROR LOADING USER SERVERS".</li> <li>3. La aplicación habilita el botón [Add Credentials] y [ConnectCRM].</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.4: Prueba Escenario 1 Caso 3

<b>Escenario ID</b>	<b>TS1</b>
<b>Test Case ID</b>	<b>TC4</b>
<b>Descripción</b>	Las credenciales de usuario están almacenadas en Windows. Existe el archivo Credentials.xml con las configuraciones de los servidores CRM El usuario arranca la aplicación. La conexión con el CRM falla (una vez inicializada la aplicación se quita la conexión de internet para simular un fallo de conexión al servidor CRM).
<b>Datos</b>	--
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación muestra la pantalla principal de la aplicación (CRM Applications).</li> <li>2. La aplicación identifica y autentica al usuario, valida las configuraciones de los servidores CRM.</li> <li>3. La aplicación carga todas las organizaciones (instancias del CRM) a las que pertenece el usuario. El botón [ConnectCRM] se habilita. Se muestran las instrucciones de uso de la aplicación.</li> <li>4. El usuario selecciona la organización y pulsa el botón [ConnectCRM].</li> <li>5. Se produce un error al conectarse al servidor del CRM. Muestra un Led rojo. Se muestra un mensaje de error.</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.5: Prueba Escenario 1 Caso 4

<b>Escenario ID</b>	<b>TS1</b>
<b>Test Case ID</b>	<b>TC5</b>
<b>Descripción</b>	Las credenciales de usuario están almacenadas en Windows. Existe el archivo Credentials.xml con las configuraciones de los servidores CRM El usuario arranca la aplicación. La conexión a la BD falla (se modifica el connectionstring en el App.config).
<b>Datos</b>	--
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación muestra la pantalla principal de la aplicación (CRM Applications).</li> <li>2. La aplicación identifica y autentica al usuario, valida las configuraciones de los servidores CRM.</li> <li>3. La aplicación carga todas las organizaciones (instancias del CRM) a las que pertenece el usuario. El botón [ConnectCRM] se habilita. Se muestran las instrucciones de uso de la aplicación.</li> <li>4. El usuario selecciona la organización y pulsa el botón [ConnectCRM].</li> <li>5. La aplicación se conecta al servidor del CRM. Muestra un Led verde. Se habilita el botón [ConnectDB] y se deshabilita el botón [ConnectCRM].</li> <li>6. El usuario pulsa el botón [ConnectDB].</li> <li>7. Se produce un error al conectarse a la base de datos. Muestra un Led rojo. Se muestra un mensaje de error.</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.6: Prueba Escenario 1 Caso 5

<b>Escenario ID</b>	<b>TS2</b>
<b>Test Case ID</b>	<b>TC1</b>
<b>Descripción</b>	El usuario ha inicializado la aplicación correctamente, incluyendo las conexiones al CRM y base de datos. El usuario selecciona una lista de marketing que <b>no existe en el CRM</b> . El usuario pulsa el botón [Run] y confirma.
<b>Datos</b>	--
<b>Resultado Esperado</b>	1. La aplicación muestra un mensaje informando que se ha CREADO una lista de marketing NUEVA en el CRM y muestra los resultados. 2. El usuario verifica en el CRM que efectivamente se ha creado la lista de marketing.
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.7: Prueba Escenario 2 Caso 1

<b>Escenario ID</b>	<b>TS2</b>
<b>Test Case ID</b>	<b>TC2</b>
<b>Descripción</b>	El usuario ha inicializado la aplicación correctamente, incluyendo las conexiones al CRM y base de datos. El usuario selecciona una lista de marketing que <b>ya existe en el CRM</b> . El usuario pulsa el botón [Run] y confirma.
<b>Datos</b>	--
<b>Resultado Esperado</b>	1. La aplicación muestra un mensaje informando que se ha ACTUALIZADO una lista de marketing en el CRM y muestra los resultados. 2. El usuario verifica en el CRM que efectivamente se ha actualizado la lista de marketing.
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.8: Prueba Escenario 2 Caso 2

<b>Escenario ID</b>	<b>TS2</b>
<b>Test Case ID</b>	<b>TC3</b>
<b>Descripción</b>	El usuario ha inicializado la aplicación correctamente, incluyendo las conexiones al CRM y base de datos. El usuario selecciona una lista de marketing que está <b>duplicada en el CRM</b> . El usuario pulsa el botón [Run] y confirma.
<b>Datos</b>	--
<b>Resultado Esperado</b>	1. La aplicación muestra un mensaje de ERROR: "IMPOSSIBLE TO IMPORT MAILING LIST!!" and "The List of Marketing is repeated in CRM".
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.9: Prueba Escenario 2 Caso 3

<b>Escenario ID</b>	<b>TS2</b>
<b>Test Case ID</b>	<b>TC4</b>
<b>Descripción</b>	<p>El usuario ha inicializado la aplicación correctamente, incluyendo las conexiones al CRM y base de datos.</p> <p>El usuario selecciona una lista de marketing que contiene contactos nuevos, contactos que ya existen en el CRM y contactos “erróneos” (su empresa no existe en el CRM).</p> <p>Selección la opción <i>Create a Log File</i>.</p> <p>El usuario pulsa el botón [Run] y confirma.</p>
<b>Datos</b>	--
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación crea/actualiza una lista de marketing en el CRM.</li> <li>2. Los contactos nuevos, se crean en el CRM y se añaden a la lista de marketing.</li> <li>3. Los contactos existentes se añaden directamente a la lista de marketing desde el CRM.</li> <li>4. Los contactos “erróneos”, no se añaden y se muestra un mensaje de advertencia.</li> <li>5. Se crea un archivo log en la carpeta temporal de Windows.</li> <li>6. El usuario verifica en el CRM que efectivamente la lista de marketing creada cumple los criterios definidos para las listas de marketing.</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.10: Prueba Escenario 2 Caso 4

<b>Escenario ID</b>	<b>TS3</b>
<b>Test Case ID</b>	<b>TC1</b>
<b>Descripción</b>	<p>El usuario ha inicializado la aplicación correctamente, incluyendo las conexiones al CRM y base de datos.</p> <p>La tabla “CONTACTOS” contiene contactos nuevos, contactos nuevos repetidos, contactos existentes en el CRM y contactos “erróneos” (su empresa no existe en el CRM).</p> <p>Selección la opción <i>Create a Log File</i>.</p> <p>El usuario pulsa el botón [Run] y confirma.</p>
<b>Datos</b>	--
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación actualiza los contactos:</li> <li>2. Los contactos nuevos, se crean y se añaden al CRM.</li> <li>3. Los contactos nuevos repetidos, se crean <u>1 vez</u> y se añaden al CRM.</li> <li>4. Los contactos existentes en el CRM no se añaden.</li> <li>5. Los contactos “erróneos”, no se añaden y se muestra un mensaje de advertencia.</li> <li>6. Se crea un archivo log en la carpeta temporal de Windows.</li> <li>7. El usuario verifica en el CRM que efectivamente los contactos se han actualizado siguiendo los criterios establecidos.</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.11: Prueba Escenario 3 Caso 1

<b>Escenario ID</b>	<b>TS4</b>
<b>Test Case ID</b>	<b>TC1</b>
<b>Descripción</b>	El usuario pulsa el botón [Add Credentials] y confirma. Se carga el formulario "CRM user credentials". El usuario introduce un nombre de usuario correcto y una contraseña correcta.
<b>Datos</b>	<a href="mailto:a.garcia.benito@ibermatica01.onmicrosoft.com">a.garcia.benito@ibermatica01.onmicrosoft.com</a> y "rightpass"
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación identifica y autentica al usuario.</li> <li>2. Las credenciales del usuario se almacenan en el SO de Windows y se crea el archivo Credentials.xml en el disco local con las configuraciones de los servidores a los que pertenece el usuario.</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.12: Prueba Escenario 4 Caso 1

<b>Escenario ID</b>	<b>TS4</b>
<b>Test Case ID</b>	<b>TC2</b>
<b>Descripción</b>	El usuario pulsa el botón [Add Credentials] y confirma. Se carga el formulario "CRM user credentials". El usuario introduce un nombre de usuario correcto y una contraseña incorrecta.
<b>Datos</b>	<a href="mailto:a.garcia.benito@ibermatica01.onmicrosoft.com">a.garcia.benito@ibermatica01.onmicrosoft.com</a> y "wrongpass"
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación muestra un mensaje de error: "Wrong Credentials".</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.13: Prueba Escenario 4 Caso 2

<b>Escenario ID</b>	<b>TS4</b>
<b>Test Case ID</b>	<b>TC3</b>
<b>Descripción</b>	El usuario pulsa el botón [Add Credentials] y confirma. Se carga el formulario "CRM user credentials". El usuario introduce un nombre de usuario correcto y deja la contraseña en blanco.
<b>Datos</b>	<a href="mailto:a.garcia.benito@ibermatica01.onmicrosoft.com">a.garcia.benito@ibermatica01.onmicrosoft.com</a> y ""
<b>Resultado Esperado</b>	<ol style="list-style-type: none"> <li>1. La aplicación muestra un mensaje de error: "Wrong Credentials".</li> </ol>
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.14: Prueba Escenario 4 Caso 3

<b>Escenario ID</b>	<b>TS4</b>
<b>Test Case ID</b>	<b>TC4</b>
<b>Descripción</b>	El usuario pulsa el botón [Add Credentials] y confirma. Se carga el formulario "CRM user credentials". El usuario deja el nombre de usuario correcto y la contraseña en blanco.
<b>Datos</b>	"" y ""
<b>Resultado Esperado</b>	1. La aplicación muestra un mensaje de error: "Wrong Credentials".
<b>Resultado Obtenido</b>	IDEM RESULTADO ESPERADO
<b>Calificación</b>	SUPERADO

Tabla B.15: Prueba Escenario 4 Caso 4

## **Anexo C: Certificado del Producto**

---



MÓNICA SAN SEBASTIÁN ARIAS, Dirección Operaciones Ibermatica Industria

**CERTIFICA:**

Que Alberto Garcia Benito con DNI 15256443Z ha realizado el proyecto fin de carrera con esta empresa desde el 04 de Junio 2018 hasta el 21 de Septiembre 2018 (400 horas), desempeñando con eficacia las labores encomendadas, desarrollando un programa funcional, el cual cumple con los requisitos encomendados, cuyo código es propiedad de Ibermatica Industria.

Título del proyecto “Desarrollo y Programación de aplicaciones para Microsoft Dynamics CRM con SDK .NET”

A lo largo de su trayectoria ha asumido con interés las responsabilidades que se le han asignado, demostrando sus competencias y capacidades, tanto técnicas como de habilidades. Se puede destacar de él su actitud y aptitud, iniciativa, interés por aprender y disposición al cambio.

Y para que conste donde proceda, expido el presente certificado en Donostia, a 25 de septiembre de 2018

