



**Universidad de Valladolid**

# **Escuela de Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Grado en Ingeniería Informática  
Mención en Tecnologías de la Información**

**Desarrollo e implementación de un sistema  
basado en arquitectura de soluciones  
en Microsoft Azure y Dynamics CRM 365 Online,  
integrando DEVOPS**

Alumno:

**Marcos Muñoz Núñez**

Tutor:

**Benjamín Sahelices Fernández**



# Desarrollo e implementación de un sistema basado en arquitectura de soluciones en Microsoft Azure y Dynamics CRM 365 Online, integrando DEVOPS

Marcos Muñoz Núñez



*Dedicado a mi madre  
por todo su apoyo durante estos años*



# Agradecimientos

*A mi familia por haber hecho posible mis estudios como Ingeniero Informático.*

*A su apoyo, ánimos e insistencia en mi les debo muchísimo.*

*A mis compañeros quienes después de tantas practicas, horas de estudio y charlas de informáticos han llegado a ser una de las mejores cosas que me llevo de la carrera.*

*A mis amigos de la Asociación AreaUrbana, quienes han alimentado mi curiosidad y ganas de aprender mas y con los que he compartido grandes logros.*

*A Alberto Casero y Álvaro Estebanez que me han guiado y apoyado hasta aquí, y a el resto de compañeros de Everis que en los últimos meses han hecho este camino mucho mas interesante.*

*A mi tutor Benjamín Sahelices por ayudarme a completar esta etapa de mi vida.*

*Y a todos los que me habéis acompañado y apoyado a lo largo esta etapa.*

***Gracias***





# Resumen

Dynamics 365 es un sistema CRM Online altamente personalizable y ampliable en cuyos desarrollos buscamos implementar metodologías y herramientas DevOps, apoyándonos en los recursos en la nube que provee Microsoft Azure. El objetivo de esta implementación es crear las bases de un sistema de Integración Continua que ayude a facilitar y acelerar los desarrollos sobre Dynamics 365. Para ello se ha realizado un estudio del contexto tecnológico y las herramientas que las que se dispone. Con esta información y conocimiento adquirido se ha realizado un análisis, diseño y pruebas de concepto que ha concluido en la implementación de los casos de uso analizados utilizando las herramientas que Azure DevOps pone a nuestra disposición. Este trabajo se enfoca desde el interior de uno de los equipos de desarrollo de Dynamics 365 CRM de Everis

**Palabras clave:** DevOps, Cloud, Integración Continua, Dynamics 365, CRM, Pipelines, Azure.



# Abstract

Dynamics 365 is a highly customizable and scalable CRM Online system, we seek to implement DevOps methodologies and tools in their customization developments, relying on the cloud resources provided by Microsoft Azure. The goal of this deployment is to build the foundations of a Continuous Integration system that facilitate and accelerate the developments on Dynamics 365. For this purpose, first a study of the technological context and tools will be made. With this information and knowledge acquired analysis, design and proof of concept have been carried out that has concluded in the implementation of the use cases analyzed using the tools that Azure DevOps puts you at our disposal. This work focuses from the inside of one of Everis Dynamics 365 CRM development teams.

**Keywords :** DevOps, Cloud, Continuous Integration, Dynamics 365, CRM, Pipelines, Azure.



# Índice general

<b>Resumen</b>	<b>VII</b>
<b>Abstract</b>	<b>IX</b>
<b>Lista de figuras</b>	<b>XV</b>
<b>Lista de tablas</b>	<b>XVII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos del trabajo . . . . .	3
<b>2. Contexto Tecnológico</b>	<b>5</b>
2.1. DevOps . . . . .	6
2.2. Cloud Computing . . . . .	6
2.3. Microsoft Dynamics . . . . .	7
2.3.1. Requisitos y Costes . . . . .	9
2.3.2. Arquitectura de Dynamics 365 . . . . .	12
2.3.3. Arquitectura de Soluciones de Dynamics365 . . . . .	14
2.4. Azure DevOps . . . . .	15
2.4.1. Azure Boards . . . . .	16
2.4.2. Azure Pipelines . . . . .	16
2.4.3. Azure Repos . . . . .	21
2.5. Herramientas y lenguajes de desarrollo . . . . .	23
<b>3. Análisis y Diseño</b>	<b>25</b>
3.1. Requisitos . . . . .	26
3.1.1. Requisitos Funcionales . . . . .	26
3.1.2. Requisitos No Funcionales . . . . .	27
3.2. Casos de Uso . . . . .	28
3.3. Planificación . . . . .	33
3.3.1. Sprint 1 . . . . .	35
3.3.2. Sprint 2 . . . . .	36
3.3.3. Sprint 3 . . . . .	37

3.3.4. Sprint 4 . . . . .	38
3.3.5. Sprint 5 . . . . .	39
3.3.6. Sprint 6 . . . . .	40
3.3.7. Sprint 7 . . . . .	41
3.3.8. Sprint 8 . . . . .	42
3.3.9. Sprint 9 . . . . .	43
3.4. Riesgos . . . . .	43
3.5. Presupuesto económico . . . . .	45
3.5.1. Costes Azure DevOps . . . . .	45
3.5.2. Recursos humanos . . . . .	46
3.6. Pruebas de Concepto . . . . .	47
3.7. Diagramas de Secuencia . . . . .	47
<b>4. Implementación y Pruebas</b>	<b>51</b>
4.1. Primeras pruebas de Concepto . . . . .	52
4.1.1. Exportar Soluciones desde Dynamics 365 . . . . .	52
4.1.2. Extraer contenido de la solución . . . . .	53
4.1.3. Empaquetar solución a partir de los archivos obtenidos . . . . .	54
4.1.4. Importar solución a Dynamics 365 . . . . .	55
4.2. Prueba de concepto en Azure Pipelines . . . . .	55
4.3. Grupos de Variables . . . . .	56
4.4. Implementación CU-01: Recuperar Cambios en CRM . . . . .	58
4.4.1. Implementación del grupo de tareas TG-01 . . . . .	58
4.4.2. Implementación Pipeline CU-01 . . . . .	60
4.5. Implementación CU-02: Restaurar Cambios a CRM . . . . .	60
4.5.1. Implementación del grupo de tareas TG-02 . . . . .	61
4.5.2. Implementación Pipeline CU-02 . . . . .	62
4.6. Implementación CU-03: Desplegar Cambios a Otro Entorno . . . . .	62
4.6.1. Implementación Pipeline CU-03 . . . . .	62
<b>5. Conclusiones</b>	<b>65</b>
5.1. Trabajo Futuro . . . . .	67
<b>Glosario</b>	<b>69</b>
<b>Bibliografía</b>	<b>73</b>
<b>A. Manual de Instalación</b>	<b>77</b>
A.1. Requisitos Previos . . . . .	77
A.2. Configuración Azure DevOps y Repositorios . . . . .	78
A.3. Repositorios Azure Repos . . . . .	78
A.4. Importar “Task Groups” . . . . .	79

---

A.5. Configuración Pipelines . . . . .	80
A.5.1. Pipeline para Recuperar Cambios del CRM . . . . .	80
A.5.2. Pipeline para restaurar los Cambios al CRM . . . . .	81
A.5.3. Pipeline para restaurar los Cambios al CRM . . . . .	81
A.6. Acceso y permisos . . . . .	82
<b>B. Manual de Usuario</b>	<b>83</b>
B.1. Requisitos Previos . . . . .	83
B.2. Recuperar cambios de CRM . . . . .	83
B.3. Restaurar Cambios a CRM . . . . .	84
B.4. Desplegar Cambios a Otro Entorno . . . . .	85
<b>C. Código: Pruebas de Concepto PowerShell</b>	<b>87</b>
<b>D. Código: JSON Task Group TG-01</b>	<b>89</b>
<b>E. Código: JSON Task Group TG-02</b>	<b>97</b>
<b>F. Contenido del CD</b>	<b>103</b>





# Índice de figuras

2.1. Comparación Modalidades Cloud . . . . .	7
2.2. Arquitectura Dynamics 365 on Premise . . . . .	12
2.3. Arquitectura Dynamics 365 Cloud . . . . .	13
2.4. Componentes de las Soluciones . . . . .	14
2.5. Arquitectura de Soluciones . . . . .	15
2.6. Interfaz Azure Boards: Tablero Kanban . . . . .	16
2.7. Interfaz Azure Pipelines: Ejecución Pipeline . . . . .	18
2.8. Interfaz Azure Pipelines: Release Pipeline . . . . .	18
2.9. Conectividad Agentes Locales . . . . .	20
2.10. Interfaz Azure Repos: Archivos Repositorio GIT . . . . .	22
2.11. Panel de Tareas Planner . . . . .	24
3.1. Diagrama de Casos de Uso . . . . .	33
3.2. Evolución de los Sprints del proyecto . . . . .	35
3.3. Diagrama de Secuencia 01 . . . . .	48
3.4. Diagrama de Secuencia 02 . . . . .	49
3.5. Diagrama de Secuencia 03 . . . . .	50
4.1. Árbol Contenido Zip Extraído . . . . .	53
4.2. Árbol Directorios Solución Extraída . . . . .	54
4.3. Creación Pipeline: Añadir Tareas al Agente . . . . .	56
4.4. Ejemplo Grupo Variables Credenciales . . . . .	57
4.5. TaskGroup TG-01: Tareas a ejecutar . . . . .	59
4.6. TaskGroup TG-01 : Entrada de Parámetros . . . . .	60
4.7. TaskGroup TG-02: Tareas a ejecutar . . . . .	61
A.1. Icono Azure Repos . . . . .	78
A.2. Datos solicitados para creación nuevo repositorio . . . . .	78
A.3. Azure Pipelines: Seccion Task Groups . . . . .	79
A.4. Pipeline en Blanco . . . . .	80
A.5. Añadir TG-01 Recuperar Cambios en CRM . . . . .	82
B.1. Añadir TG-01 Recuperar Cambios en CRM . . . . .	86

F.1. Árbol de Directorios del CD . . . . . 103

# Índice de tablas

2.1. Comparacion Dynamics 365 Cloud vs On Premise . . . . .	8
2.2. Comparación Precios Licencias Dynamics 365 . . . . .	9
2.3. Requisitos Dynamics 365 Customer Engament . . . . .	10
2.4. Requisitos SQL Server . . . . .	10
2.5. Presupuesto Servicio Platform as a Service (PaaS) . . . . .	11
2.6. Comparación Costes Cloud y On Premise . . . . .	12
2.7. Características de los Agentes Hospedados por Microsoft . . . . .	19
3.1. Requisito Funcional 01 . . . . .	26
3.2. Requisito Funcional 02 . . . . .	26
3.3. Requisito Funcional 03 . . . . .	27
3.4. Requisito Funcional 04 . . . . .	27
3.5. Requisito No Funcional 01 . . . . .	27
3.6. Requisito No Funcional 02 . . . . .	27
3.7. Requisito No Funcional 03 . . . . .	28
3.8. Requisito No Funcional 04 . . . . .	28
3.9. CU-01. Recuperar Cambios . . . . .	29
3.10. CU-02. Restaurar Cambios a CRM . . . . .	30
3.11. CU-03. Desplegar Cambios a otro entorno CRM . . . . .	32
3.12. Tareas del Sprint 1 . . . . .	36
3.13. Tareas del Sprint 2 . . . . .	36
3.14. Tareas del Sprint 3 . . . . .	37
3.15. Tareas del Sprint 4 . . . . .	38
3.16. Tareas del Sprint 5 . . . . .	39
3.17. Tareas del Sprint 6 . . . . .	40
3.18. Tareas del Sprint 7 . . . . .	41
3.19. Tareas del Sprint 8 . . . . .	42
3.20. Tareas del Sprint 9 . . . . .	43
3.21. Riesgos del proyecto . . . . .	44
3.22. Tabla de Precios Azure DevOps . . . . .	46
3.23. Presupuesto Recursos Humanos . . . . .	46



# Capítulo 1

## Introducción

Dynamics 365 es un sistema de gestión de relaciones con clientes y planificación de recursos empresariales creado por Microsoft y que funciona sobre la plataforma cloud Azure. Pese a ser un producto comercial es altamente personalizable y ampliable a través de soluciones de código. Esto lo convierte a su vez en una plataforma muy versátil pudiéndose adoptar a múltiples tipos de clientes y a casos de uso muy distintos de los que se suponen para un sistema Customer Relationship Management (CRM). La personalización de Dynamics 365 se orienta cada vez mas a la configuración nativa, dejando la programación poco a poco mas apartada[23]. Esto hace que sea necesario poder llevar un seguimiento de los cambios en las configuraciones y estructuras nativas de Dynamics 365. Sobre esta base se trabajara para implementar DevOps apoyándose en las soluciones en la nube de Azure.

En este contexto de investigación, novedad y tecnologías emergentes aparece mi motivación para buscar soluciones que mejoren el día a día de los equipos de desarrollo Dynamics 365 con toda la potencia y comodidad que pueden ofrecer las soluciones en la nube. La idea inicial de este proyecto es conseguir un sistema que permita aplicar técnicas de Integración Continua o Continuous Integration (CI) y Entrega Continua o Continuous Delivery (CD) en los desarrollos sobre Dynamics 365. Este trabajo ha de servir para facilitar un set de herramientas que faciliten el desarrollo, la gestión del proyecto y la realización de pruebas de calidad. Acortando los ciclos de desarrollo, facilitando los despliegues y la mantenibilidad del proyecto y reduciendo los tiempos de entrega del mismo.

Este trabajo busca integrarse en los diferentes proyectos que se llevan acabo dentro de la linea Enterprise and Cloud Solutions de Everis. Empresa donde se esta realizando este trabajo y de donde proviene el feedback principal, además de la supervisión de la parte técnica del proyecto. Este trabajo sera guiado y supervisado por los tutores de la empresa: Alberto Casero de la Calle y Álvaro Estebanez Barrena. Y por el tutor académico, el profesor Benjamín Sahelices Fernández, que desde la universidad supervisara la memoria y el desarrollo del trabajo como un proyecto académico.

En esta introducción se plantea la motivación y objetivos del proyecto. Tras ello el contexto tecnológico en el que se desarrollara el proyecto y los estudios previos realizados. La documentación técnica de la memoria aporta todo el trabajo realizado incluyendo el Análisis y Diseño e Implementación y Pruebas. Finalmente las Conclusiones del trabajo cierran este acompañadas de aspectos a mejorar y trabajo futuro a realizar sobre el mismo.

Esta memoria incluye en su parte final un glosario donde listan todas la abreviaturas, acrónimos y definiciones de uso extendido en la memoria. Este puede consultarse en la página 71. Así como un detalle de las referencias bibliográficas utilizadas. Al final de este documento se incluyen los manuales de instalación y uso así como los anexos que complementan la documentación técnica aportada. Este documento en su versión digital contiene múltiples enlaces que permiten navegarlo, entre las diferentes secciones, referencias bibliográficas y las referencias al glosario, por lo que se recomienda su visualización en formato digital.

## 1.1. Objetivos del trabajo

El objetivo del trabajo es integrar las herramientas DevOps de Azure en un entorno de desarrollo de Dynamics 365. La funcionalidad básica que se quiere alcanzar es proveer de un historial de los cambios en las soluciones y entidades de Dynamics 365, con un mecanismo de rollback que permita deshacer los cambios realizados. Así como proporcionar una trazabilidad del trabajo y tareas acometidas, que se pueda relacionar con los cambios realizados. Esta sería la base sobre la que construir un camino hacia la Integración Continua en los desarrollos Dynamics 365.

Para la consecución de este proyecto se deben alcanzar los siguientes objetivos:

- Comprender el funcionamiento de Dynamics 365 y como se realizan sus desarrollos y personalizaciones.
- Estudiar las metodologías DevOps.
- Conocer las diferentes herramientas que proporciona Azure DevOps.
- Analizar y diseñar un mecanismo que permita almacenar un histórico de los cambios realizados en el CRM
- Analizar y diseñar un mecanismo que permita devolver los cambios almacenados al CRM
- Implementar el sistema propuesto con las herramientas de Azure DevOps.

Este trabajo se integrara en nuevos desarrollos y ya existentes de Dynamics 365, siendo de gran utilidad para las empresas que se dedican a personalizar y extender este Software.





# Capítulo 2

## Contexto Tecnológico

En este capítulo se tratarán los antecedentes y el estado del arte de las tecnologías con la que se trabajara así como los estudios e investigaciones realizadas sobre ellas, siendo la base sobre la que se fundamentara este trabajo.

Actualmente las metodologías y herramientas DevOps se aplican en diferentes entornos y equipos de desarrollo, si bien encontramos pocos o casi ningún ejemplo de su aplicación en desarrollos de Dynamics 365 CRM. Las soluciones cloud suponen una facilidad añadida para el uso de estas prácticas y herramientas ya que presentan gran facilidad a la hora de probar e implementar sistemas que de otro forma requerían de una provisión hardware, instalación y configuración. Mientras que en la nube el tiempo de puesta en marcha se reduce a prácticamente cero.

### 2.1. DevOps

DevOps es una metodología de desarrollo de software basada en la colaboración entre los desarrolladores y los administradores de sistemas e intentando implicar también en ella a los gestores, este enfoque busca que los desarrolladores puedan centrarse en desarrollar y olvidarse de los despliegues de código. Esto conlleva que generalmente se entienda por DevOps simplemente la aplicación de CI y CD. Por supuesto las herramientas son una parte importante de esta metodología, así como los resultados que se persiguen (mayor calidad, menor coste y una alta frecuencia de lanzamientos), pero DevOps también incluye la comunicación, colaboración e integración entre los equipos de desarrollo e TI [12].

### 2.2. Cloud Computing

Con la expansión de las tecnologías de virtualización, la mayor potencia del hardware y la búsqueda de su aprovechamiento comenzó a despegar el termino Cloud Computing o Computación en la nube. Este es un modelo de prestación de servicios en el que el proveedor es el propietario de la infraestructura y la ofrece de diferentes formas a través de internet. Este modelo nos ha traído nuevas formas de trabajar y de entender los sistemas TI, y da la posibilidad de tratar la infraestructura como código, ya que esta se puede crear, instanciar, configurar y eliminar de forma programática. La nube ofrece grandes ventajas como una menor inversión inicial, pago por uso, alta disponibilidad geográfica de forma casi instantánea, y facilidad de contratación...

Las principales modalidades de servicios en la nube son Infrastructure as a Service (IaaS), Platform as a Service (PaaS) y Software as a Service (SaaS) y están ilustradas en la figura 2.1.

En este trabajo nos centraremos principalmente en la plataforma de servicios de computación en la nube que ofrece Microsoft: Azure, ya que este es el principal proveedor de servicios cloud con el que trabaja la línea ECS de Everis en Valladolid.

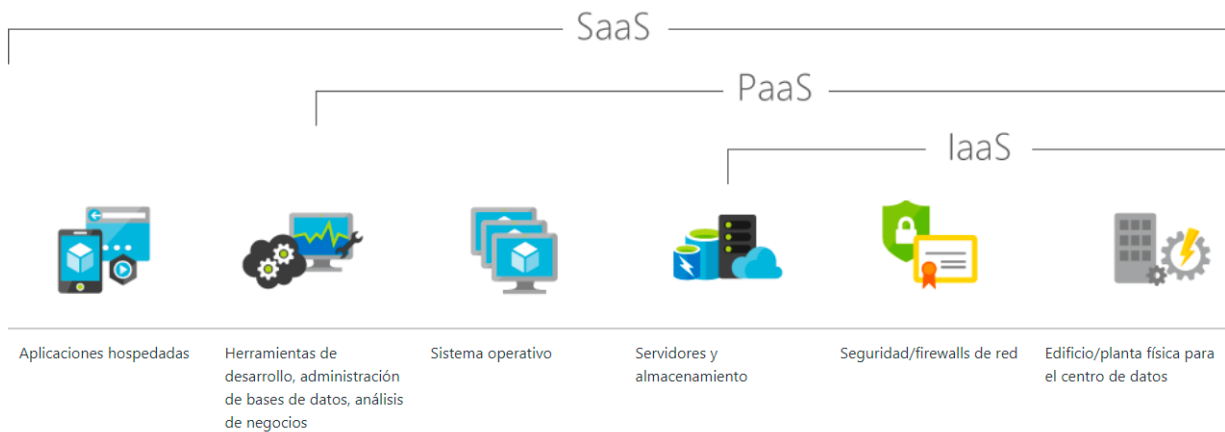


Figura 2.1: Comparación Modalidades Cloud

## 2.3. Microsoft Dynamics

Microsoft Dynamics 365 es una completa suite de programas para la gestión empresarial diseñados para facilitar y mejorar la gestión de ventas, servicio al cliente, talento, marketing, finanzas y operaciones. En este trabajo nos centraremos en sus funcionalidades como CRM, concretamente en las herramientas incluidas en la versión que Microsoft denomina Customer Engagement.

Dynamics 365 es un sistema ampliamente extensible y personalizable, muchas de estas personalizaciones se realizan mediante código. La parte cliente de Dynamics 365 se modifica mediante el uso de JavaScript, lenguaje interpretado que se ejecuta del lado del cliente en su navegador web. Mientras que para las modificaciones del lado del servidor se utiliza C Sharp (C#) con el que se compilan pequeñas Bibliotecas de enlaces dinámicos (DLLs) que se incorporan a Dynamics 365 ampliando su funcionalidad.

Sin embargo hay muchas configuraciones que se realizan de forma nativa. Por ejemplo el modelo de datos, las reglas de negocio y los formularios se configuran a través de las herramientas que proporciona Dynamics 365. Poco a poco como consecuencia de la migración al modelo de Software as a Service (SaaS) se está fomentando esta forma de personalizar el sistema. [23] Dynamics 365 está disponible en versiones Cloud y “OnPremise” o autohospedadas. La principal funcionalidad es idéntica en ambas pero en la tabla 2.1 se realiza un comparación entre ellas. Las versiones “On Premise” también pueden hospedarse en la nube por ejemplo con la contratación de máquinas virtuales en un modelo Infrastructure as a Service (IaaS). La mayoría de empresas ya trabajan con servidores virtualizados por lo que el salto a IaaS es un paso sencillo.

Tras haber realizado las prácticas en un equipo de desarrollo de Dynamics 365 CRM he constatado por mi mismo que cuando los equipos de desarrollo están distribuidos o los proyectos y equipos alcanzan un tamaño considerable aparece la dificultad de sincronizar el trabajo y no pisarse entre diferentes miembros del equipo. Demostrando la necesidad de integrar este tipo de herramientas en los entornos de desarrollo. Si bien la computación en la nube es una tendencia en alza los últimos años y la aplicación de “metodologías” DevOps está cogiendo fuerza en los últimos

	<b>Cloud</b>	<b>On Premise (Autohosteada)</b>
<b>Modelo de pago</b>	Subscripción Mensual	Gran Inversión Inicial
<b>Almacenamiento de los datos</b>	Información Almacenada en los Data Center de Azure	Propiedad y control completo sobre donde y como se almacena la información
<b>Acceso a los Datos</b>	Solo a través de las Interfaz de Programación de Aplicaciones (API) o conectores de reporting	Acceso Completo a los datos
<b>Espacio de Almacenamiento</b>	5Gb ( 2.5Gb Adicionales por cada 20 licencias)	Limitado a la capacidad del servidor
<b>Administración del Sistema</b>	Gestionada por Microsoft	Requiere gestión por parte de la empresa o un proveedor externo.
<b>Requisitos Hardware</b>	No requiere ningún hardware	Mínimo: Procesador x64 de doble núcleo, 4GB de RAM y 10 Gb de espacio en disco. (En función del uso pueden ser mayores o requerir varios servidores ).
<b>Gastos de Mantenimiento</b>	Ninguno	Altos costes de mantenimiento y actualización de los servidores.
<b>Conectividad</b>	Conectividad asegurada por Acuerdo de Nivel de Servicio (SLA) al 99.9%	No requiere conectividad a internet para poder utilizar las aplicaciones o acceder a la información.
<b>Actualizaciones</b>	Se despliegan automáticamente sin intervención del usuario	Las actualizaciones pueden retrasarse a conveniencia del usuario
<b>Flexibilidad</b>	Alta escalabilidad y redundancia gracias a las capacidades de la nube	Depende del hardware y su escalabilidad y redundancia requiere mayores inversiones
<b>Requisitos Hardware</b>	No requiere ningún hardware	Mínimo: Procesador x64 de doble núcleo, 4GB de Ram y 10 Gb de espacio en disco. (En función del uso pueden ser mayores o requerir varios servidores ).

Tabla 2.1: Comparacion Dynamics 365 Cloud vs On Premise

años, la aplicación practica de estos conceptos no es tan frecuente en las empresas.

### 2.3.1. Requisitos y Costes

Dynamics 365 esta disponible como producto cloud en modelo SaaS con un un sistema de licencias de pago mensual en función del numero de usuarios que se desee tener en la aplicación. Por otro lado la versión autohospedada o On Premise el cliente se encarga de aportar la infraestructura y recursos hardware necesarios para su ejecución, tiene un coste por licencias de pago único. Esta edición On Premise también puede ser instalada sobre maquinas virtualizadas en la nube utilizando un modelo IaaS que permite reducir la inversión inicial en hardware, pero sigue siendo necesaria la inversión en las licencias del sistema operativo, base de datos y aplicación, así como la gestión y configuración de estos. Tambien es posible el uso de un modelo PaaS que permitiría olvidarse del sistema operativo y la gestión de las maquinas virtuales.

#### Licencias

En la tabla 2.2 se muestran una comparativa entre los costes de licencias de una implementación Dynamics 365 en local en comparación con el servicio en la nube. Para ello supondremos un grupo de 20 usuarios sobre el paquete Customer Engagment de Dynamics 365. Para la versión on premise se estimara posteriormente el coste del Hardware que permita cumplir con los requisitos mínimos.

	Cloud	OnPremise
Coste por usuario	97 €/mes	918,00 €
Coste anual por usuario	1.164,00 €	-
Total 20 usuarios	23.280,00 €/año	18.360,00 €
Inversión en 3 años	69.840,00 €	18.360,00 €

Tabla 2.2: Comparación Precios Licencias Dynamics 365

**Nota:** Se indica el gasto a 3 años para ilustrar la diferencia entre cloud y on premise, remarcar que al coste on premise hay que sumar los costes del resto del software, hardware y su mantenimiento que se analizaran a continuación.

#### Hardware y software adicional

En las versiones On premise, es necesario incluir los gastos del sistema operativo y el servidor SQL Server que también podrían contratarse como un servicio cloud.

Microsoft indica que los requisitos hardware para la versión On Premise de Dynamics 365 se detallan en la tabla 2.3. Se presupone que componentes adicionales como SQL Server[33] o SQL Server Reporting Services, entre otros no están instalados ni en ejecución en el sistema. Es decir

que necesitan máquinas aparte que cumplan sus requisitos. Los Requisitos Hardware de SQL Server se indican en la tabla 2.4.

Componente	*Mínimo	*Recomendado
Procesador	Procesador de 1,5 GHz de arquitectura x64 o compatible, doble núcleo	CPU de 2 GHz de arquitectura x64, cuatro núcleos o superior, como los sistemas AMD Opteron o Intel Xeon
Memoria	4 GB de RAM	8 GB de RAM o más
Disco duro	10 GB de espacio disponible en el disco duro	40 GB o más de espacio disponible en el disco duro

Tabla 2.3: Requisitos Dynamics 365 Customer Engament

Componente	*Mínimo	*Recomendado
Procesador	Procesador x64: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T <b>1,4 GHz</b>	Procesador x64: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T <b>2 GHz o superior</b>
Memoria	1 GB	<b>Al menos 4 GB</b> y debe aumentar a medida que el tamaño de la base de datos aumente para asegurar un rendimiento óptimo.
Disco duro	1480 MB de espacio disponible en el disco para la instalación de las características básicas y 6 GB para archivos temporales durante el funcionamiento. A mayores se ha de provisionar espacio de almacenamiento para los datos.	8030 MB o más de espacio disponible en el disco para la instalación de todas las características y 6 GB para archivos temporales durante el funcionamiento. A mayores se ha de provisionar espacio de almacenamiento para los datos.

Tabla 2.4: Requisitos SQL Server

Como hemos comentado a lo largo del capítulo, la nube puede proporcionar hardware bajo demanda, para continuar con ese enfoque vamos a estimar los costes Hardware como si fueran servicios cloud. Es decir el coste de una plataforma sobre la que ejecutar Dynamics 365 On Premise, para ello elegiremos instancias de AppService de Azure, un servicio PaaS que nos proveerá de Sistemas totalmente aislados del resto con sistema operativo Windows Server, así mismo el coste de las bases de datos en la nube asociadas y el almacenamiento. Así como unas necesidades estándar de Red. Estos costes están detallados en la tabla 2.5 y han sido calculados con la Calculadora de Precios de Azure[10]

<b>Servicio</b>	<b>Descripción</b>	<b>Coste Estimado</b>
<i>Base de Datos</i>	Base de datos única, nivel Estándar Tier, S4: 1 TB de almacenamiento incluido por base de datos, 1 bases de datos x 730 Hours	341,43€
<i>Espacio de almacenamiento</i>	Redundancia Almacenamiento de blobs en bloque de Acceso frecuente, Capacidad: 1.000 GB	17,48€
<i>Plataforma de ejecución</i>	Servicio PaaS Nivel aislado; 2 instancias I2: 2 Nucleos, 7Gb RAM, 1 TB Disco x 730 Hours; SO Windows	1.936,52€
<i>Red Virtual</i>	1000 de transferencia de datos de la región Europa Occidental a la región Europa Occidental	16,87€
<i>VPN Acceso a Red Virtual</i>	Puertas de enlace de VPN, horas de puerta de enlace 730, 1000 GB, Transferencias de datos entre redes virtuales	51,68€
<i>Ancho de Banda</i>	1 TB, Zona 1: Norteamérica y Europa.	74,76€
	<b>Total Mensual</b>	<b>502,21€</b>
	<b>Total Anual</b>	<b>6.026,51€</b>

Tabla 2.5: Presupuesto Servicio PaaS

Tras el estudio de costes realizado aquí se presenta la comparativa entre una implementación Cloud y una “On Premise” que sera hosteada bajo un modelo PaaS en Azure. La tabla 2.6 muestra estos datos. La versiones cloud otorgan mucha mas flexibilidad al poder modificar en cualquier momento el numero de licencias con un coste inicial relativamente bajo, sin embargo a largo plazo el coste es mucho mayor. La versión On Premise requiere una mayor inversión inicial, pero hemos conseguido reducir esta utilizando computación en la nube, así el hardware se contrata bajo un

modelo PaaS que permite olvidar la compra de hardware y las licencias de SQL Server y Windows Server, así como su instalación y administración. Lo cual nos acerca a las ventajas de la versión Cloud, aunque los costes del Hardware pueden variar en función de la carga del sistema.

	Hardware	Software	Total en 1 Año	Total en 3 años
<b>Cloud</b>	-	23.280,00 €/año	23.280,00 €	69.840,00 €
<b>On Premise</b>	6.026,51 €/año	18.360,00€	24.386,51 €	36.439,53 €

Nota: El coste hardware se basa en los costes anuales del servicio PaaS en Azure[10].  
Se ha supuesto un grupo de 20 usuarios para los calculos.

Tabla 2.6: Comparación Costes Cloud y On Premise

### 2.3.2. Arquitectura de Dynamics 365

Dynamics se estructura como una aplicación modular con una frontera bien definida entre los clientes y el servidor.

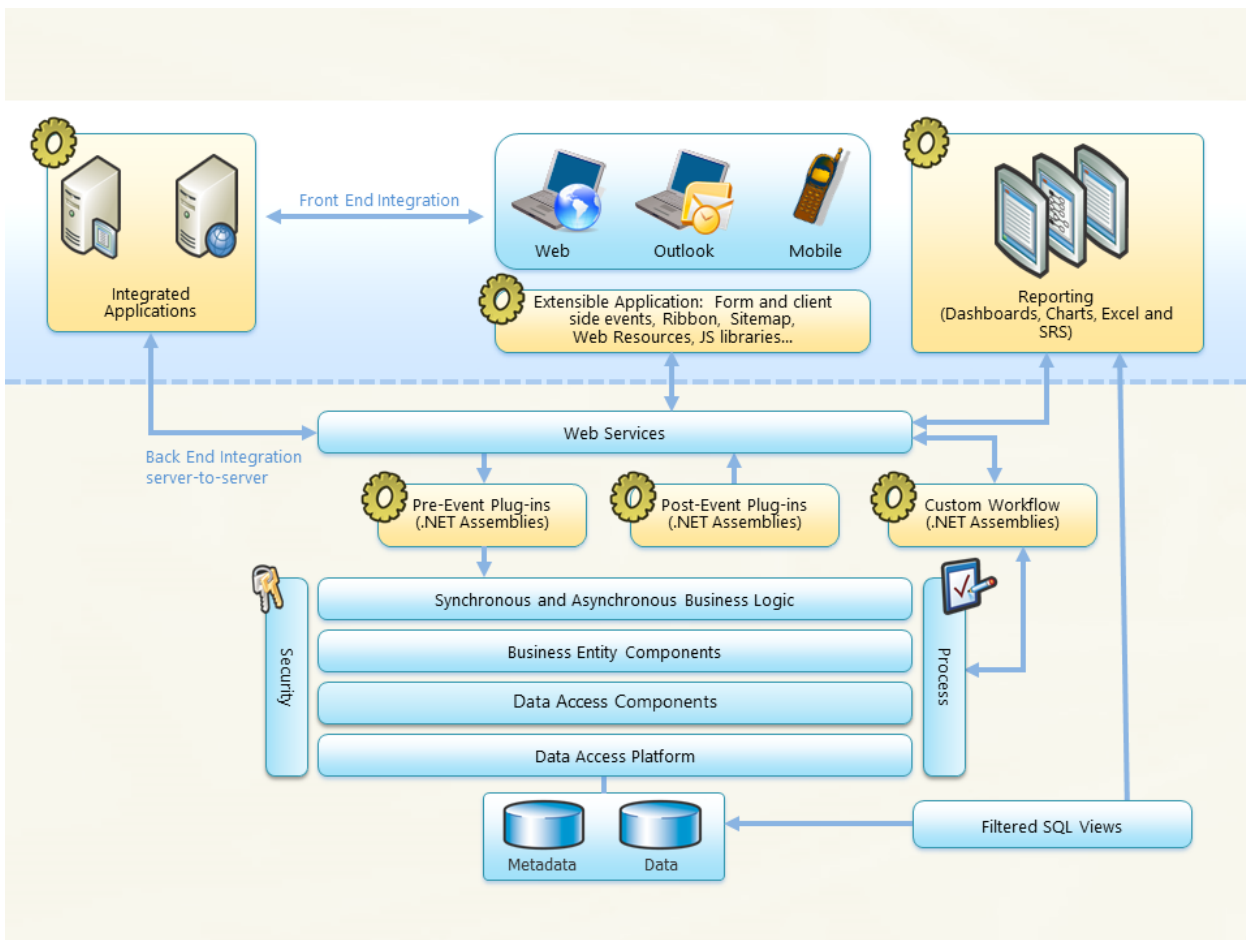


Figura 2.2: Arquitectura Dynamics 365 on Premise



El sistema sigue una arquitectura de 4 capas:

**Capa de presentación:** Agrupa los diferentes clientes (Web,Móvil...), así como las herramientas de generación de informes. Las aplicaciones cliente admiten son extensibles y admiten personalización principalmente mediante recursos web (HTML,JavaScript....).

**Capa de Servicios:** Se ejecuta sobre Windows Server y en ella se enmarcan los Servicios Web. La funcionalidad de esta capa puede ser extendida mediante DLLs programadas en .NET.

**Capa de aplicación:** Al igual que la de servicios se ejecuta en el servidor, se encarga de la lógica de negocio, los componentes de la entidades de negocio y los componentes de acceso a los datos. Las capas transversales de procesos y seguridad abarcan toda esta y también forman parte de la capa de datos.

**Capa de Datos:** La plataforma de acceso a datos se encuentra en el servidor y provee el acceso a los datos y metadatos alojados en SQL Server. Esta capa puede ser accedida por las herramientas de creación de reportes.

En el entorno cloud la arquitectura es ligeramente diferente ya que no se utiliza una base de datos como tal. La capa de datos queda sustituida por el Common Data Service como se muestra en la imagen 2.3. Los datos de Common Data Service se almacenan en un conjunto de entidades. Una entidad es un conjunto de registros que se usa para almacenar datos, similar a cómo una tabla almacena los datos en una base de datos.[40] El CDS no permite el acceso directo a los datos pero provee de herramientas como PowerBI para poder realizar informes y disponer de los datos en otras aplicaciones del ecosistema Azure.

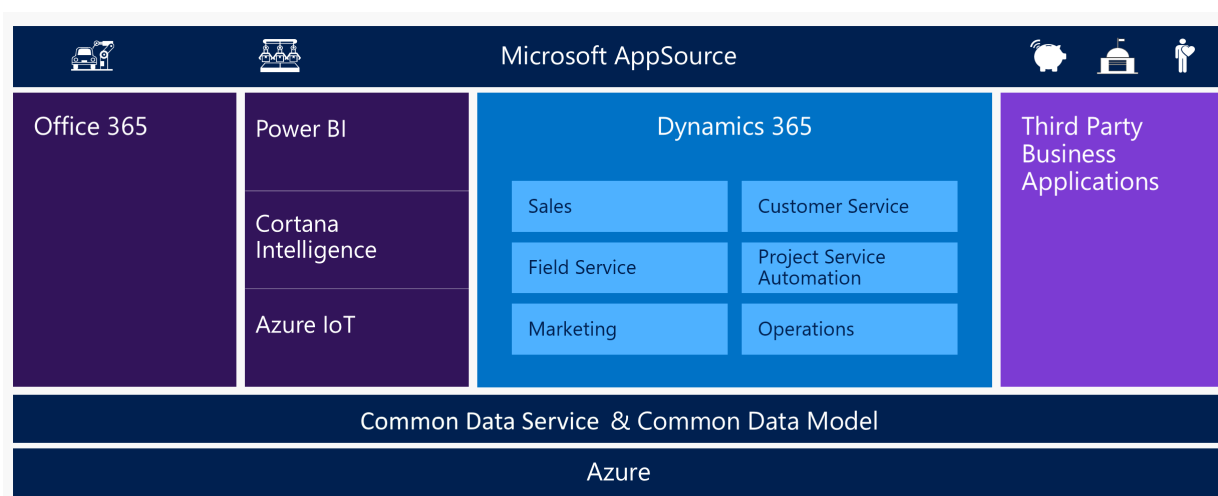


Figura 2.3: Arquitectura Dynamics 365 Cloud

### 2.3.3. Arquitectura de Soluciones de Dynamics365

La arquitectura de soluciones de Dynamics 365 define la forma en la que las personalizaciones y configuraciones del sistema se aplican. Dentro de una solución encontramos diferentes componentes que se crean mediante herramientas de personalización o las API incluidas en las aplicaciones Dynamics 365 for Customer Engagement y están totalmente hospedadas en la aplicación. En el diagrama 2.4 se muestran los diferentes tipos de componentes de la solución.[13] Este proyecto buscar realizar un control de cambios y mantener un historial de estos componentes.



Figura 2.4: Componentes de las Soluciones

Estos incluyen prácticamente todos los cambios y personalizaciones que se realizan al sistema. Existen dos tipos de soluciones de aplicaciones Dynamics 365 for Customer Engagement: administradas y no administradas. Una solución administrada es una solución completada que está diseñada para distribuir e instalar. Una solución no administrada es la que aún está en desarrollo o no está diseñada para distribuirse. Cuando se completa la solución no administrada y se desea distribuir, se puede exportar y empaquetar como una solución administrada.[13]. El diagrama 2.5 muestra cómo las soluciones administradas y no administradas interactúan con la solución del sistema para controlar el comportamiento de la aplicación.

En el diagrama 2.5 se muestran diferentes tipos de soluciones[13]:

**Solución del sistema:** La solución del sistema representa los componentes de la solución definidos en aplicaciones Dynamics 365 for Customer Engagement. Sin ningún tipo de solución administrada o personalizaciones, la solución del sistema define el comportamiento predeterminado de aplicación. Muchos de los componentes de la solución del sistema son personalizables y se pueden usar en soluciones administradas y no administradas.

**Soluciones administradas:** Las soluciones administradas se instalan por encima de la solución del sistema y pueden modificar los componentes de la solución personalizables o agregar más componentes de la solución. Las soluciones administradas también se puede superponer sobre otras soluciones administradas. Mientras una solución administrada permite la personalización de sus componentes de la solución, otras soluciones administradas se pueden instalar por encima de la misma y modificar los componentes de la solución personalizables que proporciona.

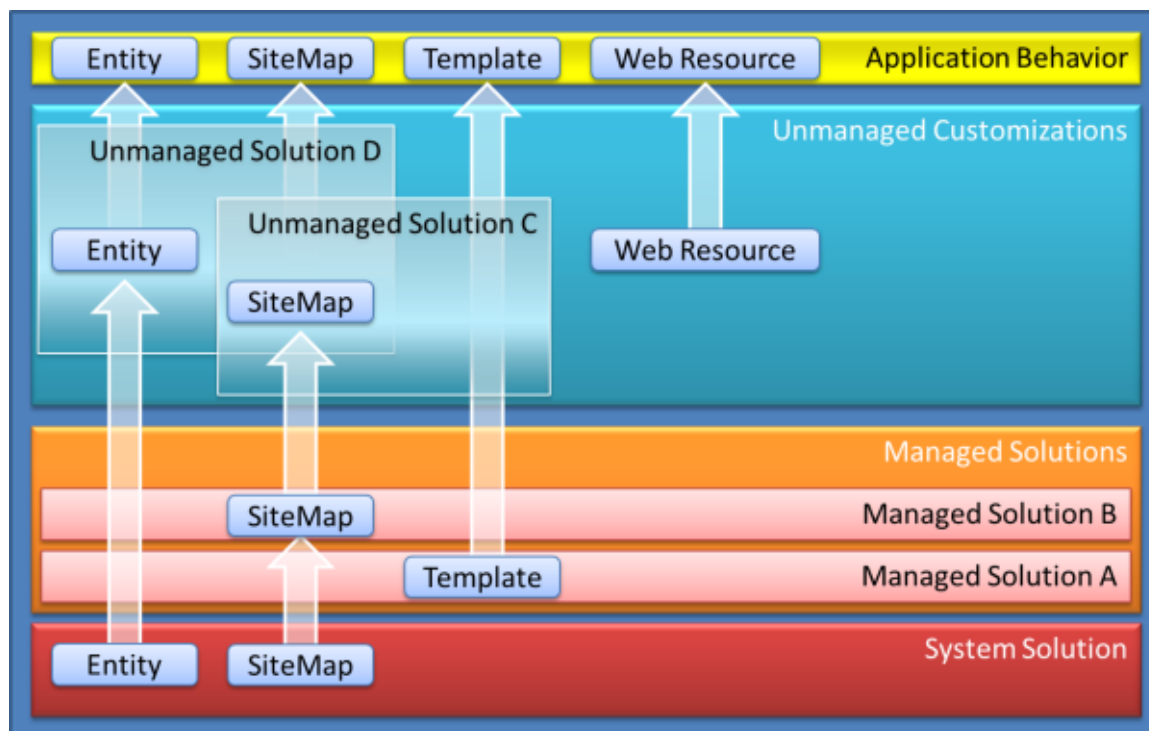


Figura 2.5: Arquitectura de Soluciones

**Soluciones no administradas:** Las soluciones no administradas son grupos de personalizaciones no administradas. Cualquier componente de la solución no administrada personalizado puede asociarse con cualquier número de soluciones no administradas. Una de las principales diferencias con las administradas es que quitar una no administrada del sistema no revierte los cambios que se hayan realizado en la configuración del sistema.

**Comportamiento de aplicación:** El comportamiento final de una instancia de aplicaciones Dynamics 365 for Customer Engagement para una organización específica es la culminación de la solución del sistema, las soluciones administradas y las no administradas.

## 2.4. Azure DevOps

Azure DevOps es una colección de servicios en la nube de Microsoft para colaborar en el desarrollo de software. Estos servicios pueden ser accedidos a través de la web o de un cliente para Entorno de desarrollo integrado (IDE). Azure DevOps ofrece repositorios para realizar control del código, servicios de compilación y despliegue como soporte para la integración y entrega continua (CI y CD), herramientas para planificar y realizar seguimiento del trabajo, los problemas en el código y las incidencias, herramientas para realizar pruebas de las aplicaciones, tableros de mandos altamente personalizables, entre otras...

Prácticamente la totalidad de la interacción con este servicio se realiza vía web, aunque existen conectores externos que permiten automatizar tareas, o configurar las pipelines a través de archivos

concretos en el repositorio, la interfaz web es el principal medio para interactuar con el servicio como se ve en las imágenes 2.6, 2.7 y 2.10. A continuación se detallarán las herramientas principales que se utilizarán en este proyecto.

### 2.4.1. Azure Boards

Azure Boards es un conjunto de herramientas de planificación centradas en metodologías ágiles. Estas permiten realizar un seguimiento del trabajo con paneles de tarjetas Kanban, listar los trabajos pendientes, mostrar paneles con información del trabajo del equipo y obtener informes personalizados para mantener un seguimiento del trabajo

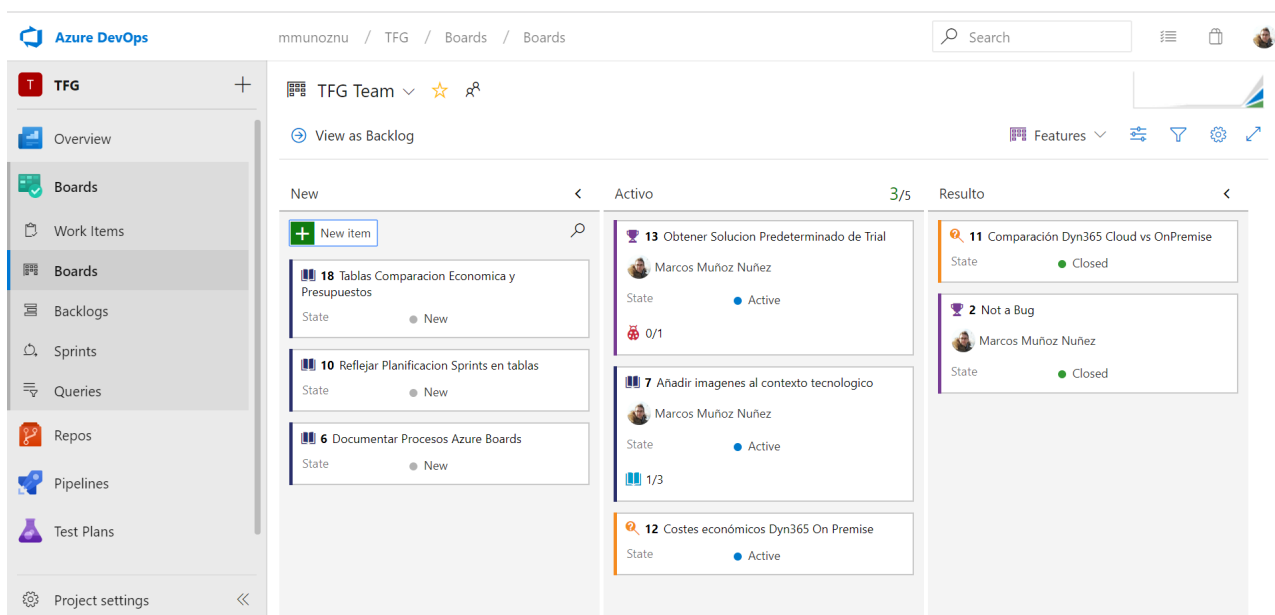


Figura 2.6: Interfaz Azure Boards: Tablero Kanban

### 2.4.2. Azure Pipelines

Una de las herramientas centrales para este proyecto son las pipelines, estas permiten configurar un flujo de tareas que se ejecutaran a petición, ya sea de forma manual, programada, o por algún evento que se programe. Los trabajos que se configuran Azure Pipelines son enviados a los agentes. Los agentes son un programa software instalable que ejecuta una pipeline cada vez, estos pueden ser de dos tipos:

- Agentes Hospedados por Microsoft en Azure
- Agentes Auto-hospedados Localmente

Los agentes hospedados son pequeñas máquinas virtuales en la nube de Azure que se crean y configuran al inicio de la pipeline y son destruidas cuando esta acaban. Los agentes de Azure permiten olvidarse de configurar máquinas virtuales, entornos de desarrollo y pruebas etc. Los agentes auto-hospedados pueden ser instalados en hardware propio de forma local y son controlados desde Azure Pipelines al igual que los agentes hospedados en Azure. Esto proporciona mucha versatilidad ya que se puede elegir que Sistema Operativo (Linux, Mac o Windows) y herramientas son provisionadas en estas máquinas obteniendo siempre un entorno limpio. Los agentes instalan las herramientas, realizan las tareas asignadas (compilación, ejecución de scripts de consola, copia de datos...), y tras ello informan del resultado. También permiten automatizar los test realizándose tras cada compilación y permitiendo implementar CI y CD con facilidad ya que permiten la liberación o despliegue de versiones de forma automática si los test son correctos.

Las pipelines de Azure DevOps proporcionan un sistema de variables que se pasan al agente a la hora de crearlo, permitiendo trasladar información entre ejecuciones de la build. Estas son especialmente útiles en los agentes hospedados en Azure. Ya que por ejemplo las variables de entorno no se mantienen entre ejecuciones ya que cada ejecución es una máquina nueva. Estas variables pueden ser definidas tanto en la interfaz web de Azure Pipelines, como programáticamente a través de los archivos YAML que permiten describir las acciones de un pipeline.[6] En algunos casos las variables pueden contener información sensible que no debe almacenarse en texto plano y estar disponible para cualquiera; para atajar este problema Azure Pipelines utiliza las llamadas variables secretas. Las variables secretas están encriptadas con una clave RSA de 2048 bits. Las variables secretas se ponen a disposición de los agentes y scripts para su uso, por lo que se debe de ser cuidadoso a la hora de dar acceso a que la pipeline sea editada, ya que estos pueden imprimirse dentro de un script. Las variables secretas no pueden definirse en los archivos YAML (ya que estos son visibles a cualquiera que tenga acceso al repositorio) por lo que solo pueden configurarse desde la interfaz web. Si se desean utilizar variables en varias pipelines diferentes se pueden utilizar los grupos de variables.[5]

Aunque en general hablemos de Pipelines para referirnos a los flujos de tareas configurables en Azure DevOps existen dos tipos concretos de pipeline:

- Build Pipelines, o Pipelines de Compilación
- Release Pipelines o Pipelines de Lanzamiento

La principal diferencia entre ellas es que las Pipelines de Lanzamiento o Release, son fundamentalmente grupos de pipelines con distintas fases o stages con mecanismos de control entre ellos. Lo que permite automatizar las distintas fases de un despliegue.

Por ejemplo, una o varias Build pipelines generan los artefactos a compilar desde el código fuente, tras esto se pasa a una etapa que prepara el entorno en el que se va a desplegar, sobre este se realizan unos test automáticos. Si estos test son correctos se pasa a la siguiente fase y se despliega en un entorno de preproducción donde también se ejecutan pruebas automáticas. La siguiente fase, el despliegue en producción puede tener por ejemplo la necesidad de una aprobación manual,

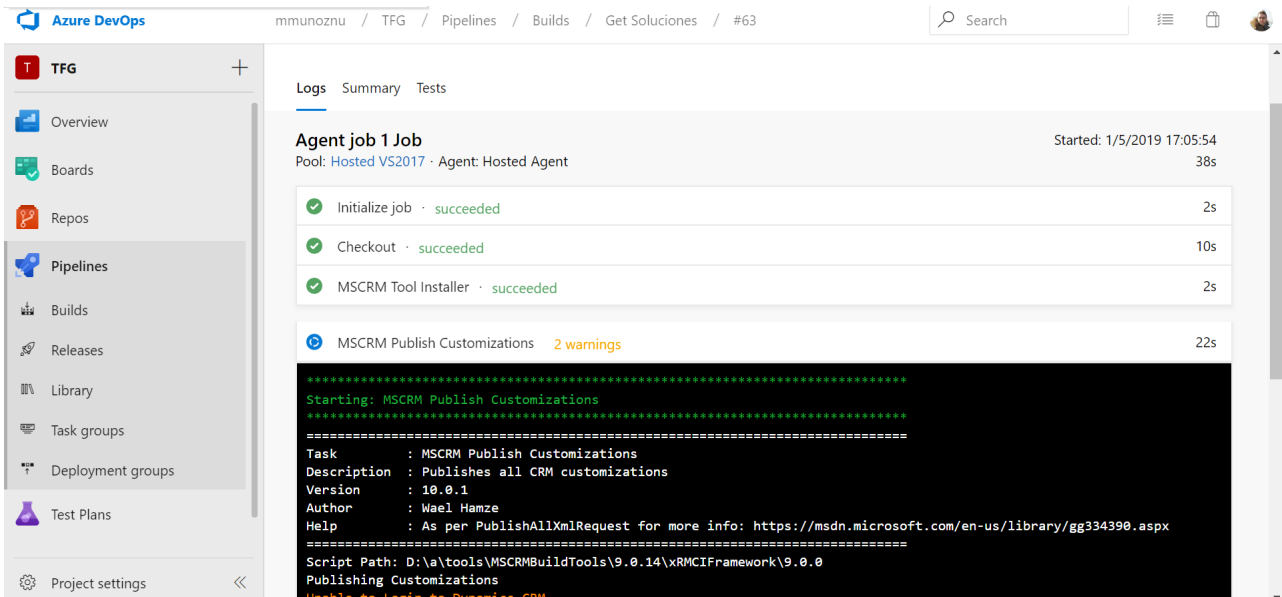


Figura 2.7: Interfaz Azure Pipelines: Ejecución Pipeline

por lo que llegado a este punto se avisara a la persona encargada informándole de los resultados de las pruebas anteriores. Si esta persona aprueba el despliegue se ejecuta la ultima pipeline que realiza el despliegue en producción. Un ejemplo de esto puede verse en la imagen 2.8

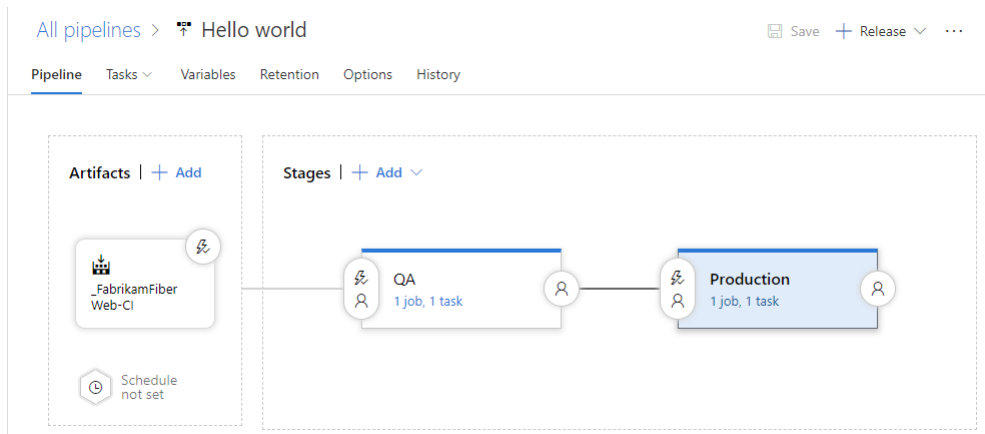


Figura 2.8: Interfaz Azure Pipelines: Release Pipeline

Para este proyecto vamos a centrarnos en las Build Pipelines, aunque las Release Pipelines son muy interesantes de cara a desarrollar futuros sistemas de CD. Otras de las características de Azure Pipelines son los grupos de tareas y los grupos de despliegue. Los grupos de tareas se detallan mas adelante en este apartado. Los grupos de despliegue [14] permiten configurar conjuntos de maquinas relacionadas como pueden ser los entornos de despliegue de un mismo proyecto (Desarrollo, Calidad, Preproducción, Producción...), para luego poder configurar pipelines de lanzamiento de forma mas sencilla, no se plantea la utilización de grupos de despliegue para este

proyecto.

Azure Pipelines sigue un modelo de SaaS, sobre el que Microsoft va implementando mejoras que se liberan a los usuarios de forma periódica. Esto hace que incluso durante el desarrollo de este proyecto hayan aparecido nuevas funciones y cambios en la documentación. [2]

### Agentes Hospedados en Azure

Como se indicaba previamente en el punto 2.4.2, la ejecución de los trabajos en Azure Pipelines puede ser llevada a cabo por agentes hospedados por Microsoft, quien los mantiene, actualiza y gestiona. Cada vez que se ejecuta una pipeline se realiza sobre una maquina virtual limpia, los trabajos que se le encargan pueden ejecutarse directamente sobre la maquina virtual o en un contenedor cargado en ella. En caso de requerir condiciones especiales, por ejemplo que el agente este detrás del firewall o VPN, se puede optar por hospedarlo localmente.

Este no es el caso que se tratara en este trabajo ya que buscamos explotar las ventajas de la computación en la nube, pero es una opción a tener en cuenta en entornos de desarrollo que no pueden ser accedidos de forma externa.

Microsoft dispone de varias imágenes para cargar las maquinas virtuales, cubriendo varios sistemas operativos y diferentes versiones de los mismos. Cada imagen lleva diferente software instalado, Microsoft proporciona la relación de este así como las versiones concretas de cada software. [3] También permite al usuario la instalación de nuevo software en la maquina virtual utilizando las tareas de instalación de herramientas.

Las características y limitaciones de los agentes se indican en la tabla 2.7.

<b>vCPU</b>	2
<b>Memoria RAM</b>	7GB
<b>Almacenamiento</b>	10GB
<b>Tiempo de ejecución</b>	360 minutos (6 horas)
<b>Permisos</b>	Ejecución como administrador

Tabla 2.7: Características de los Agentes Hospedados por Microsoft

Para el uso de estos agentes se ha de tener en cuenta no utilizar rutas absolutas ya que los nombres de las unidades y discos así como la estructura concreta de los agentes pueden cambiar sin previo aviso.

### Agentes Auto-hospedados

Ya hemos hablado de los agentes hospedados por Microsoft y sus ventajas, pero también existe la opción de hospedar nuestros propios agentes en maquinas locales. Los agentes auto-hospedados permiten tener el control completo del sistema donde se ejecutan los agentes, así como del software

que estos pueden utilizar en las pipelines. Estos agentes se pueden instalar sobre sistemas operativos Linux, MacOS o Windows. También están disponibles contenedores Docker para facilitar la instalación. Una vez se ha instalado el agente en la maquina se puede instalar cualquier otro software que se necesite para la ejecución de las pipelines.

Los agentes instalados en maquinas locales han de comunicarse con Azure Pipelines a través de internet. Ya que este se encarga de gestionar sus trabajos y el estado de estos. Toda la comunicación es iniciada por el agente vía HTTP o HTTPS. Esto permite la configuración de reglas en los firewalls locales que permitan a los agentes comunicarse con Azure Pipelines de forma segura y sin exponer mas puertos al exterior. Esto supone también una ventaja para la creación de Pipelines que han de trabajar con servidores que están tras el cortafuegos de una empresa, o necesitan de una conexión VPN para acceder a ellos como se ilustra en la imagen 2.9.

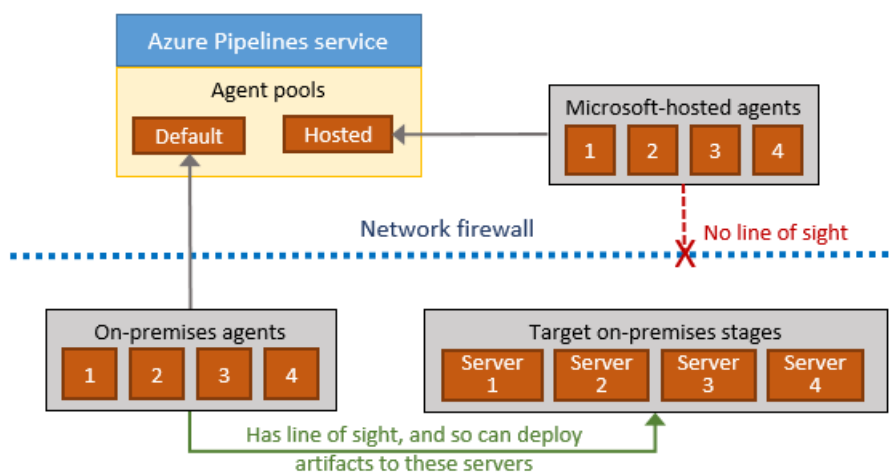


Figura 2.9: Conectividad Agentes Locales

La posibilidad de instalar los agentes en maquinas locales, dentro del firewall, o de configurar estas maquinas para conectarse de forma segura a los diferentes servicios necesarios, permite facilitar su despliegue dentro de entornos empresariales seguros en los que ciertos servicios no están expuestos a internet.

Otra ventajas de los agentes auto-hospedados localmente son:

- Los agentes locales siempre están disponibles, por lo que inician antes los trabajos. Los agentes en la nube normalmente requieren entre pocos segundos y varios minutos para ser asignados y este tiempo varia dependiendo de la carga en los sistemas de Microsoft.
- Los agentes locales no tienen que borrar todo su contenido cada vez que finalizan una pipeline, por lo que son mas rápidos al no tener que configurar y redescargar el repositorio en cada ocasión.
- Pueden instalarse varios agentes en la misma maquina, facilitando la tarea de paralelizar varias pipelines sin que el coste se dispare.



- Los recursos hardware se pueden adecuar en función de las necesidades.

## Triggers

Los triggers o disparadores permiten que una Pipeline se ejecute de forma automática en respuesta a un evento.[9] Existen triggers de Compilación y de Despliegue, en función del tipo de pipeline que se quiera ejecutar y dentro de estos encontramos tres grandes tipos:

- **Integración Continua:** Los triggers de CI ejecutan una pipeline cada vez que se sube un cambio al repositorio, ya sea a una rama específica o con una tag concreta. Presentan varias opciones como por ejemplo agrupar varios cambios para reducir el número de pipelines lanzadas en caso de que se hagan cambios muy frecuentes en el repositorio.
- **Programados:** Estos triggers permiten seleccionar los días y horas a los que se desea que se inicie la pipeline. Se pueden configurar diferentes días de la semana y horas para diferentes ramas del repositorio, así como en diferentes zonas horarias para equipo internacionales.
- **Compilación Terminada:** En grandes desarrollos pueden aparecer muchas dependencias entre varios componentes. En estos casos los triggers de compilación permiten configurar un pipeline para que se ejecute al finalizar una anterior, lo que permite que la cadena de dependencias se vaya compilando paso a paso.

## Grupos de tareas

Los grupos de tareas o “Task Group” permiten encapsular una serie de tareas que ya estén definidas en una pipeline en una sola tarea reusable que puede añadirse a cualquier pipeline, como cualquier otra tarea. [35]

Permite elegir los parámetros que se mostraran al exterior y abstraer el resto de opciones y configuraciones. Al crear un grupo de tareas este pasa a estar disponible en el catálogo de tareas para todas las pipelines del proyecto. Esto ayuda a estandarizar y centralizar ciertas acciones comunes a varias pipelines, además de simplificar su mantenimiento, puesto que cuando se realiza un cambio en un grupo de tareas este cambio se replica en todas las pipelines que tengan esa tarea en su flujo, manteniendo así la misma versión en todas las pipelines.

### 2.4.3. Azure Repos

Azure Repos provee de diferentes SCVs hospedados en Azure, los más utilizados y que se usaran también en este proyecto son los repositorios GIT, pero también ofrece compatibilidad con TFVC, SVN y otros servicios de repositorios de terceros como GitHub y Bitbucket. Para facilitar la colaboración entre desarrolladores ofrece un sistema de hilos de discusión, la configuración de directivas de rama para controlar las solicitudes de incorporación de cambios y una potente búsqueda de código semántica que reconoce las clases y variables y permite buscar en todo el repositorio con rapidez. Estos tienen una gran integración con el resto de servicios de Azure

DevOps como Boards y Pipelines, permitiendo asociar los cambios en el código fuente a una tarea concreta de Azure Boards, o que los cambios en el repositorio ejecuten una Pipeline.

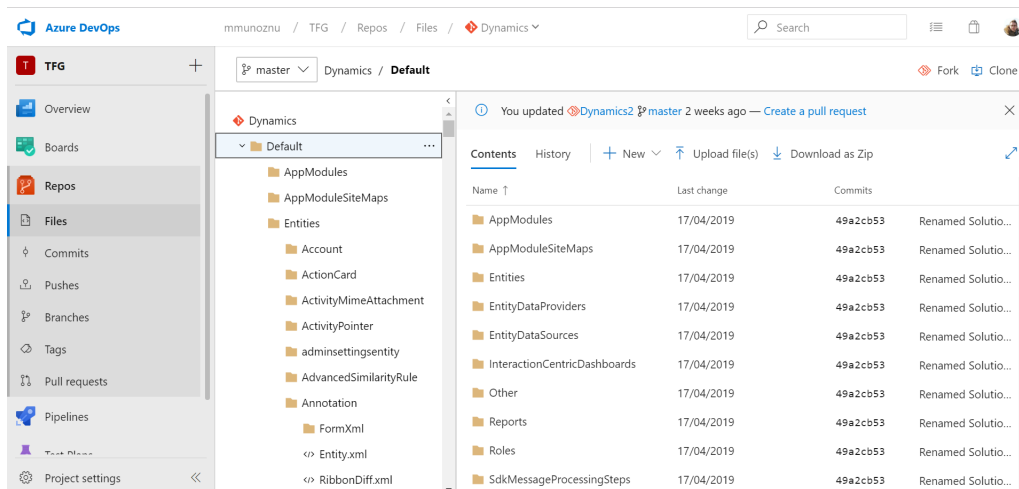


Figura 2.10: Interfaz Azure Repos: Archivos Repositorio GIT

## 2.5. Herramientas y lenguajes de desarrollo

Esta memoria así como los anexos que la acompañen ha sido realizada utilizando el sistema de composición de textos LaTeX, concretamente y acorde con el enfoque del trabajo se ha utilizado Overleaf, una solución SaaS en la nube, donde se han almacenado los distintos archivos de la memoria, se ha realizado la edición del texto en LaTeX, y se ha compilado el documento PDF final. Se eligió esta solución por que aporta flexibilidad y portabilidad, ya que no hay que estar instalando y configurando los compiladores y librerías LaTeX en cada equipo que se quiera usar, y permite consultar y editar la memoria en cualquier dispositivo que disponga de una conexión a internet. También ofrece la descarga de los archivos del proyecto en caso de que se quiera trabajar de forma offline. Y además esta se almacena en un SCV Git que permite realizar un control de cambios, tener copias locales distribuidas y sincronizar el trabajo que se realiza offline. Para la realización de diagramas UML de análisis y diseño se ha utilizado Visual Paradigm un completo programa para el diseño y gestión del desarrollo de sistemas TI, cuya licencia ha sido proporcionada por la escuela.

La toma de notas, referencias bibliográficas, apuntes y diagramas se ha realizado utilizando el servicio de cuadernos digitales OneNote de Microsoft, el servicio de sincronización en la nube permite acceder a los contenidos des cuaderno desde cualquier dispositivo que disponga de conexión a internet. Para la planificación y seguimiento del proyecto se utilizan herramientas como Microsoft Project y Microsoft Planner, ambas herramientas proporcionadas por la universidad. Aunque ambas se consideraron y utilizaron inicialmente, debido a la metodología ágil con sprint que se utilizará en el proyecto, la herramienta mas utilizada para la gestión ha sido Microsoft Planner, una herramienta web de gestión de proyectos donde se ha creado un backlog con las tareas que se han ido asignando a cada sprint al inicio de este. Las tareas asignadas al sprint se representan en un panel de tarjetas como se ilustra en imagen 2.11.

Para el trabajo sobre Dynamics 365 se utilizan instancias de prueba (trials) que Microsoft proporciona, estas tienen una asignación mínima de recursos que es compartida por todas las instancias de prueba y una duración de un mes por lo que cada mes se crea una nueva, para poder continuar el trabajo se realiza la exportación de datos y configuraciones y se importan en la nueva trial provisionada.

La elaboración de los archivos JavaScript se realiza con un editor de texto avanzado como Visual Studio Code, en cuanto a las DLLs C# se utiliza el IDE Visual Studio en su versión 2017 que provee de editor de texto, sistemas de comprobación de código inteligentes y compilador C#. En la parte que se desarrolla en Azure DevOps se utilizan distintas herramientas y tecnologías, los mas frecuentes son scripts de consola, los cuales se desarrollan en PowerShell la consola de comandos de Microsoft, que permite su portabilidad entre diferente sistemas. Así mismo se utilizan herramientas ya preparadas por terceros que podemos encontrar en el Marketplace de Azure DevOps, un expositor en el que los desarrolladores pueden publicar sus herramientas y conectores que se integran con Azure DevOps ampliando su funcionalidad.

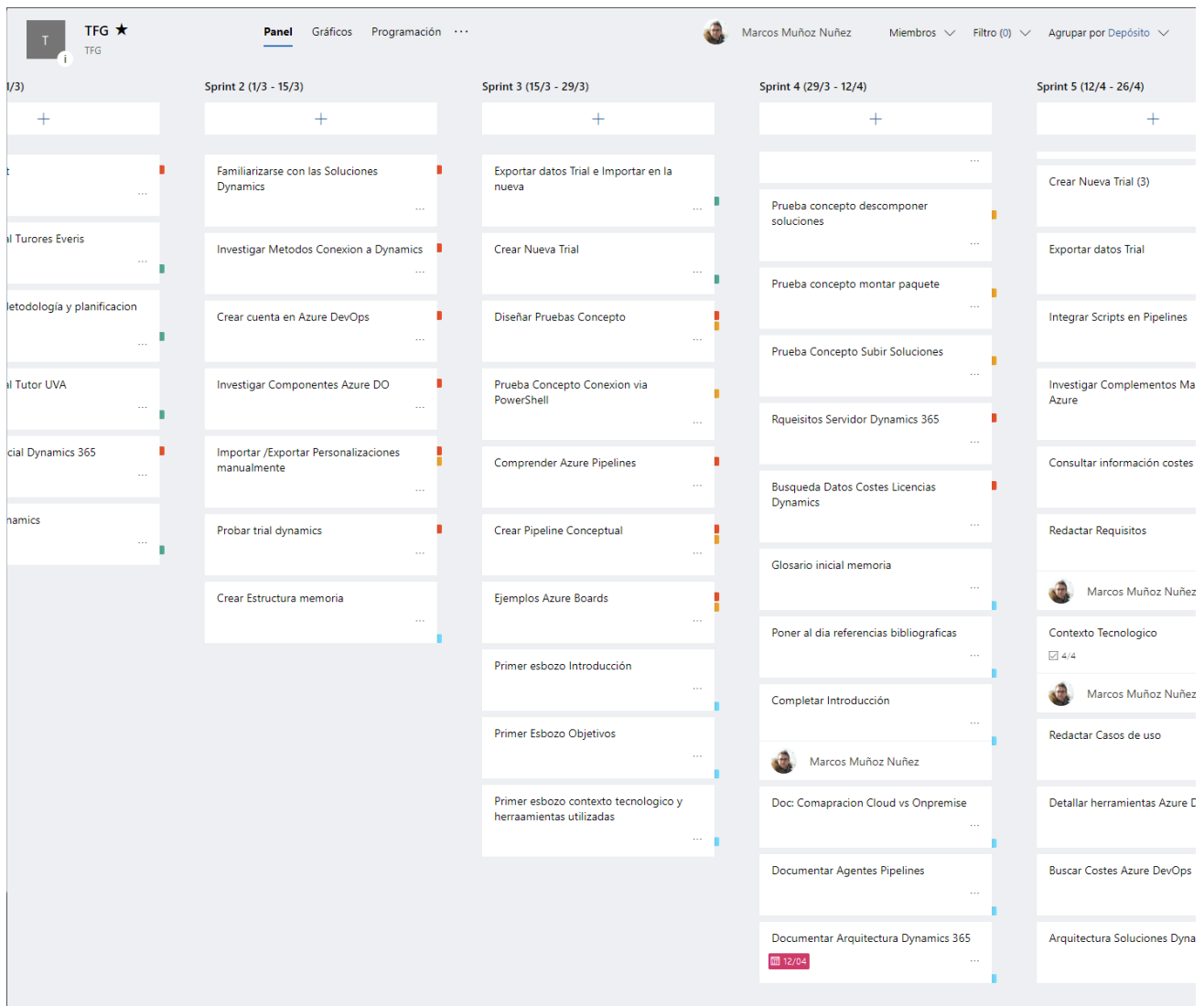


Figura 2.11: Panel de Tareas Planner

# Capítulo 3

## Análisis y Diseño

Tras haber realizado los estudios previos de las tecnologías y herramientas a utilizar pasamos a realizar un análisis y diseño de la solución que deseamos construir. Tal y como se explico anteriormente se busca un sistema que permita realizar un seguimiento de los cambios realizados en la personalización del sistema CRM Dynamics365, así como poder restaurar un punto anterior en estos cambios, este sistema se construirá sobre la plataforma Azure DevOps utilizando las herramientas estudiadas anteriormente.

Este capitulo se organizara de forma que primero trataremos los requisitos recolectados, que se modelaran en forma de casos de uso. Tras esto se abordara la planificación del proyecto y un análisis de riesgos y costes. Tras estos análisis se prepararan pruebas de concepto que demuestren la viabilidad del proyecto y proporcionen una base sobre la que ir iterando. Finalmente se cierra el capitulo con los diagramas de secuencia de cada caso de uso que preceden a su implementación.

### 3.1. Requisitos

En este apartado se describen los requisitos del sistema a desarrollar. Estos marcaran las funcionalidades a desarrollar así como algunas restricciones en como se hará la implementación o las características de esta. La recolección de requisitos se realiza al principio del proyecto y en gran medida vienen dados por las necesidades de la empresa en este momento. El carácter iterativo del proyecto permitió ampliar estos con necesidades detectadas en el desarrollo de los mismos en este caso el Requisito RF-04 Facilitar el despliegue de soluciones entre entornos se detecto y añadió a posteriori y derivo en un nuevo caso de uso.

#### 3.1.1. Requisitos Funcionales

<b>RF-01</b>	Recuperar cambios en la configuración CRM
<b>Descripción</b>	Obtener los cambios que se hayan realizado sobre el CRM
<b>Caso Uso Relacionado</b>	CU-01

Tabla 3.1: Requisito Funcional 01

<b>RF-02</b>	Restablecer cambios en el Sistema
<b>Descripción</b>	Devolver el estado de la configuración CRM a un punto anteriormente recuperado
<b>Caso Uso Relacionado</b>	CU-02

Tabla 3.2: Requisito Funcional 02

<b>RF-03</b>	Programación Horaria Recuperación Automática
<b>Descripción</b>	La recuperación de cambios ha de poder programarse para que se ejecute automáticamente en un horario predeterminado.
<b>Caso Uso Relacionado</b>	CU-01

Tabla 3.3: Requisito Funcional 03

<b>RF-04</b>	Facilitar el despliegue de soluciones entre entornos
<b>Descripción</b>	Obtener los cambios de una instancia CRM y desplegarlos en otra diferente manteniendo un registro de estos.
<b>Caso Uso Relacionado</b>	CU-03

Tabla 3.4: Requisito Funcional 04

### 3.1.2. Requisitos No Funcionales

<b>RNF-01</b>	Almacenar Cambios de forma incremental
<b>Descripción</b>	No se deberá guardar imágenes completas de las soluciones del CRM, solo los cambios incrementales sobre cada archivo individual
<b>Caso Uso Relacionado</b>	CU-01

Tabla 3.5: Requisito No Funcional 01

<b>RNF-02</b>	Recuperar cambios de todo el sistema
<b>Descripción</b>	Se deberá recuperar la solución predeterminada del sistema ya que es la única que contiene todos los cambios, estén incluidos en una solución o no.
<b>Caso Uso Relacionado</b>	CU-01

Tabla 3.6: Requisito No Funcional 02

<b>RNF-03</b>	Las credenciales deben almacenarse de forma segura
<b>Descripción</b>	Las credenciales (ej login de Dynamics) no han de almacenarse en texto plano o ser accesibles para el usuario.
<b>Caso Uso Relacionado</b>	CU-01, CU-02

Tabla 3.7: Requisito No Funcional 03

<b>RNF-04</b>	La implementación se realizara sobre la nube de Azure
<b>Descripción</b>	La solución adoptada en este proyecto deberá ser implementada con las herramientas de Azure DevOps
<b>Caso Uso Relacionado</b>	CU-01, CU-02

Tabla 3.8: Requisito No Funcional 04

### 3.2. Casos de Uso

Tras la obtención de requisitos y los análisis iniciales se obtienen dos casos de uso muy claros y de un alcance bastante bien definido, que se comentan y refinan con los tutores de Everis, estos son los casos de uso: CU-01 y CU-02. Tras la realización de estos dos primeros casos de uso se detecta una utilidad que no se habia planteado inicialmente pero que puede ser util en el dia a dia de los equipos de desarrollo de Dynamics 365, estando alineado con los objetivos planteados al inicio del proyecto, por lo que se añade y analiza el caso CU-03. Además tras haber analizado CU-01 y CU-02, CU-03 parece una evolución natural sobre ellos, ya que se trata de una implementación de los dos previos por lo que se beneficiara del trabajo ya realizado en los anteriores casos de uso.

<b>Nombre e ID del CU</b>	<b>CU-01. Recuperar Cambios en CRM</b>
<b>Versión</b>	1.0
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario desea que los cambios se guarden en el sistema de control de versiones, creando una imagen de las entidades del sistema en ese momento.
<b>Precondiciones</b>	<p>PRE-1 El usuario debe tener acceso al sistema</p> <p>PRE-2 El usuario solicita la recuperación de cambios o esta esta programada para su inicio automático.</p> <p>PRE-3 La instancia de Dynamics 365 debe de estar activa y responder a peticiones</p>



<b>Postcondiciones</b>	PST-1 Los cambios realizados en el CRM se añaden al Sistema de control de versiones
<b>Flujo normal</b>	<p>FN1 El usuario solicita manualmente la recuperación de los cambios</p> <p>FN2 El sistema añade la tarea a la cola.</p> <p>FN3 Cuando la tarea entra en ejecución el sistema se conecta al CRM, obtiene los cambios realizados en las soluciones, los extrae y prepara para su introducción en el Sistema de control de versiones</p> <p>FN4 El sistema hace commit de los cambios al Sistema de control de versiones quien añade únicamente las modificaciones realizadas. Si la operación ha sido correcta se notifica al usuario.</p>
<b>Flujo alternativo 1</b>	FA1 Si la tarea ha sido programada con antelación (por ejemplo para que se ejecute a una hora determinada) no es necesaria la intervención del usuario en la tarea y se procede a añadirla en cola.
<b>Flujo alternativo 2</b>	FA2 Si ocurre algún error durante la recuperación de los cambios se notifica al usuario y la tarea se cancela sin escribir nada en el Sistema de control de versiones
<b>Prioridad</b>	Alta
<b>Otra info</b>	Se recuperaran los cambios de la solución determinada de Dynamics 365.

Tabla 3.9: CU-01. Recuperar Cambios

<b>Nombre e ID del CU</b>	<b>CU-02. Restaurar Cambios a CRM</b>
<b>Versión</b>	1.0
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario desea devolver CRM a un punto concreto, indicado por un commit en el Sistema de control de versiones
<b>Precondiciones</b>	<p>PRE-1 El usuario debe tener acceso al sistema.</p> <p>PRE-2 El usuario debe indicar la instantánea concreta a la que desea restaurar el CRM.</p> <p>PRE-3 El usuario solicita la restauración de cambios.</p> <p>PRE-4 La instancia de Dynamics 365 debe de estar activa y responder a peticiones</p>
<b>Postcondiciones</b>	<p>PST-1 El CRM se restaura al punto indicado por el usuario.</p>
<b>Flujo normal</b>	<p>FN1 El usuario solicita manualmente la restauración de una imagen concreta</p> <p>FN2 El sistema añade la tarea a la cola.</p> <p>FN3 Cuando la tarea entra en ejecución el sistema obtiene la imagen indicada, realiza su empaquetado, actualiza el número de versión de la solución a subir, y realiza su importación a CRM</p> <p>FN4 Si la operación ha sido correcta se notifica al usuario y se publican los cambios en el CRM.</p>
<b>Flujo alternativo 1</b>	<p>FA1 Si ocurre algún error durante la recuperación de los cambios se notifica al usuario y la tarea se cancela sin realizar cambios en el CRM.</p>
<b>Flujo alternativo 2</b>	<p>FA2</p>
<b>Prioridad</b>	Alta
<b>Otra info</b>	Se importara la solución predeterminada a CRM .

Tabla 3.10: CU-02. Restaurar Cambios a CRM

<b>Nombre e ID del CU</b>	<b>CU-03. Desplegar Cambios a Otro Entorno</b>
<b>Versión</b>	1.2
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario desea llevar los cambios de CRM a otro entorno o instancia, dejando registro de los cambios en el Sistema de control de versiones
<b>Precondiciones</b>	<p>PRE-1 El usuario debe tener acceso al sistema.</p> <p>PRE-2 El usuario debe indicar la solución concreta a la que desea desplegar en el CRM.</p> <p>PRE-3 Las credenciales de los diferentes entornos han de estar preconfiguradas.</p> <p>PRE-4 El usuario solicita el despliegue de una solución en otro entorno.</p> <p>PRE-5 La instancia de Dynamics 365 debe de estar activa y responder a peticiones</p>
<b>Postcondiciones</b>	<p>PST-1 Los cambios realizados en el CRM se añaden al Sistema de control de versiones y son importados y publicados en otra instancia CRM diferente a la de partida.</p>

<p><b>Flujo normal</b></p>	<p>FN1 El usuario solicita manualmente el despliegue de una solución concreta desde un entorno a otro.</p> <p>FN2 El sistema añade la tarea a la cola.</p> <p>FN3 El usuario solicita manualmente la restauración de una imagen concreta</p> <p>FN4 El sistema añade la tarea a la cola.</p> <p>FN5 Cuando la tarea entra en ejecución el sistema se conecta al CRM de origen, obtiene los cambios realizados en las soluciones, los extrae y prepara para su introducción en el Sistema de control de versiones</p> <p>FN6 El sistema hace commit de los cambios al Sistema de control de versiones quien añade únicamente las modificaciones realizadas. Si la operación ha sido correcta se notifica al usuario.</p> <p>FN7 El sistema realiza la importación a CRM de los cambios descargados hacia el entorno de destino.</p> <p>FN8 El sistema realiza la publicación de los cambios la instancia CRM de destino.</p> <p>FN9 Si la operación ha sido correcta se publican los cambios en el CRM y se notifica al usuario.</p>
<p><b>Flujo alternativo 1</b></p>	<p>FA1 Si ocurre algún error durante la recuperación de los cambios se notifica al usuario y la tarea se cancela sin realizar cambios en el CRM.</p>
<p><b>Flujo alternativo 2</b></p>	<p>FA2</p>
<p><b>Prioridad</b></p>	<p>Media</p>
<p><b>Otra info</b></p>	<p>Se importara la solución indicada a CRM .</p>

Tabla 3.11: CU-03. Desplegar Cambios a otro entorno CRM

Estos casos de uso están recogidos en el diagrama de casos de uso 3.1.

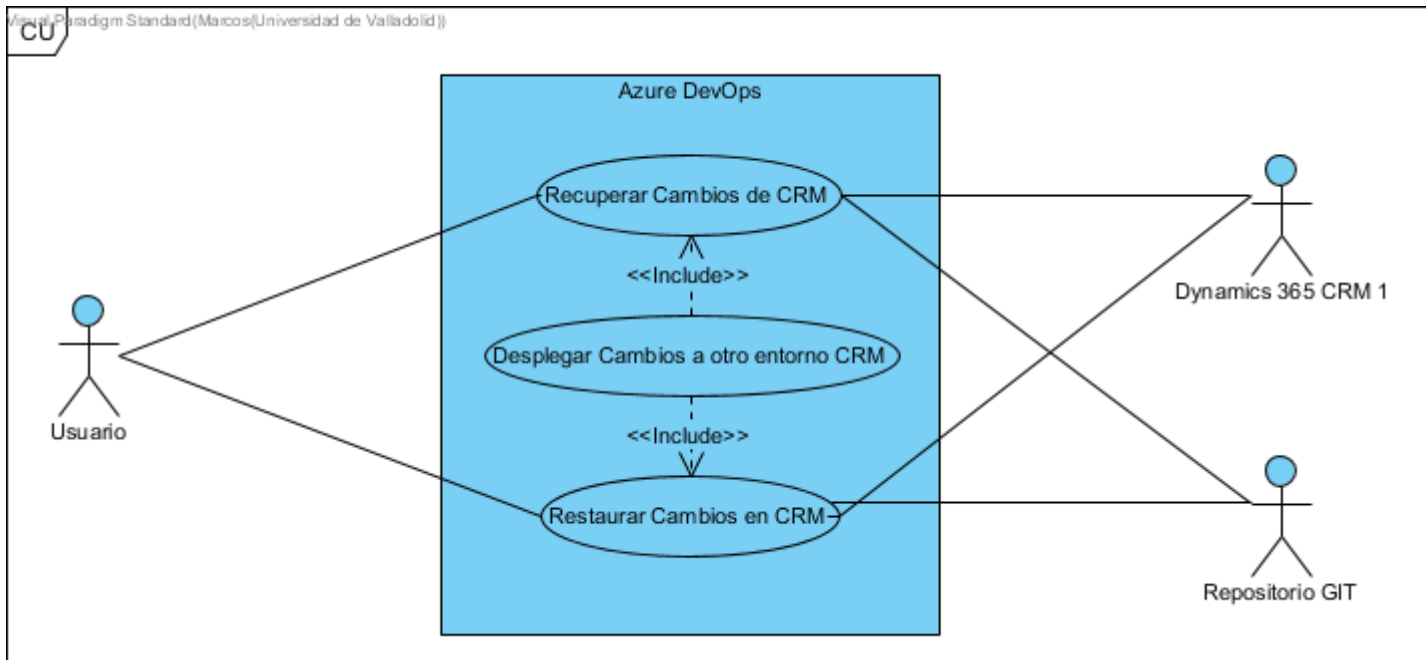


Figura 3.1: Diagrama de Casos de Uso

### 3.3. Planificación

Las principales tareas de este proyecto serán:

- Estudio de las tecnologías, metodologías y herramientas a utilizar.
- Elección de metodología y programación a gran escala en función de esta.
- Análisis y estimación de la tareas necesarias para cumplir con los objetivos.
- Realización de pruebas de concepto
- Implementación
- Documentación

A grandes rasgos se estima que las tareas de investigación, estudio y familiarización ocuparan las primeras 5 semanas. Una vez se tenga la información básica sobre el contexto del proyecto se elaborara una planificación básica inicial que permita tener una referencia para medir el avance del proyecto y planificar la carga de trabajo de los Sprints en consecuencia. Una vez se tengan datos suficientes para comenzar el análisis este se estima en unas 3 semanas y mientras se ejecuta se continua con el estudio ampliando el conocimiento disponible para dicho análisis, en ciertos puntos el análisis se se intercalara con la realización de algunas pruebas de concepto que demuestren la viabilidad del análisis realizado y aporten mas información a este. Se estima que las pruebas de concepto puedan abarcar unas 3 semanas, aunque se realizaran de forma intermitente, intercaladas con el estudio y análisis necesario para llevarlas a cabo. La implementación del proyecto se

estima que abarcara unas 5 semanas. En cuanto a la documentación la idea inicial es que sea constante a lo largo del proyecto intensificándose cuando los trabajos en otras áreas del mismo vayan finalizando, por lo que se prevé que las 2 últimas semanas del proyecto se dediquen a finalizar esta.

En los últimos años las metodologías ágiles, en concreto SCRUM, han conseguido una gran presencia en la informática y el desarrollo de software. El modelo de sprints define una unos bloques de una duración de tiempo predeterminada, generalmente de entre dos o tres semanas. Al inicio de cada Sprint se deciden las tareas a realizar en el, esta es una gran ventaja en proyectos en los que a priori no se conocen con precisión las tareas a realizar. También esta planificación permite la realización de cambios según se va avanzando en el proyecto, por lo que no es necesario fijar el alcance del proyecto a su inicio ya que este puede alargarse o acortarse hasta que este finalizado, reduciendo la carga de la estimación inicial, muy importante en metodologías clásicas como el Proceso Unificado.

La metodología de trabajo que se utilizará en el proyecto se *basara* en metodología SCRUM y se organizara en sprints de una duración de 2 semanas. Se fijan reuniones con el tutor de Everis cada 2 Viernes, coincidiendo con el final de los sprints para poder evaluar el resultado de estos y planificar el próximo. La fecha de inicio del proyecto es el 15 de Febrero de 2019, cuando se celebra la primera reunión con los tutores de Everis, en el momento del inicio la fecha de finalización es desconocida, pero se espera una duración de unos 8 sprints, pudiendo alargarse o acortarse en función del avance del proyecto y las fechas concretas de entrega.

A continuación se detalla la evolución de esta planificación a lo largo del los sprints realizados y como se han ido distribuyendo las tareas. Así como gráficos y datos sobre las desviaciones que hayan podido ocurrir en la planificación. El gráfico 3.2 resume la evolución de la carga de los sprints a lo largo del proyecto así como la cantidad de tareas completadas y no completadas en cada uno de ellos.

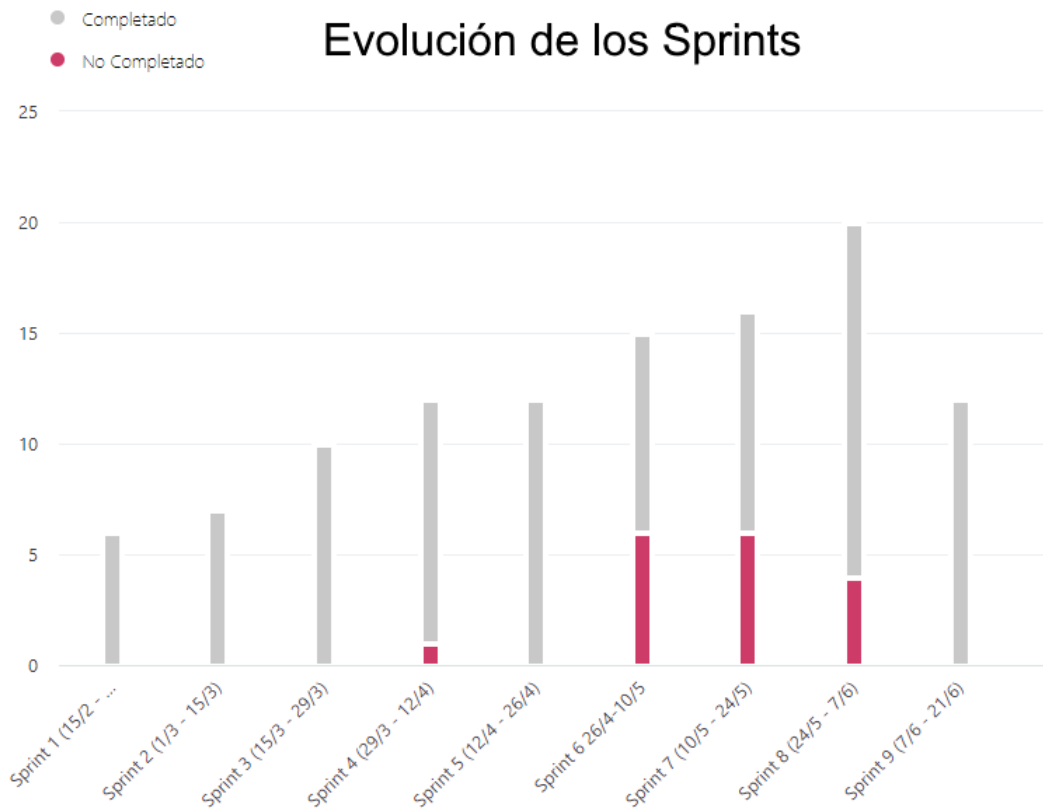


Figura 3.2: Evolución de los Sprints del proyecto

### 3.3.1. Sprint 1

La tabla 3.12 muestra las tareas acometidas en el Sprint inicial realizado entre el 15 de febrero y el 1 de marzo de 2019.

---

**Sprint 1 - 15 de Febrero de 2019 - 1 de Marzo de 2019**

---

<b>Tarea</b>	<b>Tipo</b>	<b>Estado</b>
Reunión inicial tutores Everis	Gestión	Completada
Elección de Metodología y Planificación inicial	Planificación	Completada
Reunión tutor Uva	Gestión	Completada
Formación Inicial en Dynamics 365	Estudio	Completada
Formación GIT	Estudio	Completada
Solicitud trial Dynamics 365	Estudio	Completada
Familiarizarse con C-Sharp	Estudio	Completada

Tabla 3.12: Tareas del Sprint 1

### 3.3.2. Sprint 2

La tabla 3.13 muestra las tareas acometidas en el Sprint realizado entre el 1 de marzo y el 15 de marzo de 2019.

---

**Sprint 2 - 1 de Marzo de 2019 - 15 de Marzo de 2019**

---

<b>Tarea</b>	<b>Tipo</b>	<b>Estado</b>
Estudiar interfaces externas Dynamics	Estudio	Completada
Crear cuenta en Azure DevOps	Gestión	Completada
Familiarizarse con Componentes Azure DevOps	Estudio	Completada
Familiarizarse con las soluciones Dynamics	Estudio	Completada
Exportar/ Importar soluciones manualmente	Estudio	Completada
Conocer contenido de las soluciones	Estudio	Completada
Crear Estructura Memoria	Documentación	Completada

Tabla 3.13: Tareas del Sprint 2



### 3.3.3. Sprint 3

La tabla 3.14 muestra las tareas acometidas en el Sprint realizado entre el 15 de marzo y el 29 de marzo de 2019.

**Sprint 3 - 15 de Marzo de 2019 - 29 de Marzo de 2019**

<b>Tarea</b>	<b>Tipo</b>	<b>Estado</b>
Solicitar nueva Trial	Gestión	Completada
Exportar e Importar datos en nueva trial	Gestión	Completada
Diseño pruebas concepto	Estudio	Completada
Familiarizarse con lenguaje scripting PowerShell	Estudio	Completada
Prueba concepto conexión CRM vía PowerShell	Implementación	Completada
Estudiar funcionamiento Azure Pipelines	Estudio	Completada
Crear Pipeline conceptual	Implementación	Completada
Crear repositorios Azure Repos	Implementación	Completada
Familiarizarse con Azure Boards	Estudio	Completada
Primer esbozo Introducción	Documentación	Completada
Primer esbozo Objetivos	Documentación	Completada
Primer esbozo Contexto tecnológico y herramientas utilizadas	Documentación	Completada

Tabla 3.14: Tareas del Sprint 3

### 3.3.4. Sprint 4

La tabla 3.15 muestra las tareas acometidas en el Sprint realizado entre el 29 de marzo y el 12 de abril de 2019.

**Sprint 4 - 29 de Marzo de 2019 - 12 de Abril de 2019**

Tarea	Tipo	Estado
Prueba concepto exportar soluciones	Implementación	Completada
Prueba concepto descomponer soluciones	Implementación	Completada
Glosario Inicial memoria	Documentación	Completada
Prueba concepto reempaquetar solución	Implementación	Completada
Prueba concepto importar soluciones	Implementación	Completada
Requisitos servidor Dynamics 365	Estudio	Completada
Completar Introducción	Documentación	Completada
Búsqueda información costes licencias Dynamics	Estudio	Completada
Comparación Cloud y On Premise	Documentación	Completada
Arquitectura Agentes Pipelines	Documentación	Completada
Actualizar referencias bibliográficas	Documentación	Completada
Arquitectura Dynamics 365	Documentación	<i>No Completada</i>

Tabla 3.15: Tareas del Sprint 4

### 3.3.5. Sprint 5

La tabla 3.16 muestra las tareas acometidas en el Sprint realizado entre el 12 de abril y el 26 de abril de 2019.

Sprint 5 - 12 de Abril de 2019 - 26 de Abril de 2019		
Tarea	Tipo	Estado
Solicitar nueva Trial	Gestión	Completada
Exportar e Importar datos nueva trial	Gestión	Completada
Integrar scripts en Pipelines	Implementación	Completada
Conocer y explorar complementos Marketplace Azure	Estudio	Completada
Consultar información costes RRHH	Estudio	Completada
Presupuesto RRHH	Documentación	Completada
Requisitos	Documentación	Completada
Contexto Tecnológico	Documentación	Completada
Casos de Uso	Documentación	Completada
Detallar herramientas Azure DevOps	Documentación	Completada
Buscar costes Azure DevOps	Estudio	Completada
Arquitectura soluciones Dynamics 365	Documentación	Completada

Tabla 3.16: Tareas del Sprint 5

### 3.3.6. Sprint 6

La tabla 3.17 muestra las tareas acometidas en el Sprint realizado entre el 26 de abril y el 10 de mayo de 2019.

**Sprint 6 - 26 de Abril de 2019 - 10 de Mayo de 2019**

<b>Tarea</b>	<b>Tipo</b>	<b>Estado</b>
Imágenes interfaz Azure DevOps	Documentación	Completada
Ampliar contexto Cloud Computing	Documentación	Completada
Instalar Visual Paradigm	Gestión	Completada
Diagramas Casos de Uso	Documentación	Completada
Incorporar complementos Azure Marketplace al Diseño	Estudio	Completada
Implementar Pipeline con Complementos Marketplace	Implementación	Completada
Comparación Costes Hardware y Software	Documentación	Completada
Realizar pruebas en Instancias Everis	Estudio	Completada
Reestructurar memoria	Documentación	Completada
Buscar nuevas vías para extraer soluciones	Estudio	<i>NO Completada</i>
Obtener lista completa de soluciones del sistema	Estudio	<i>NO Completada</i>
Primer esbozo de Implementación	Documentación	<i>NO Completada</i>
Estudiar formas de combinar varias soluciones	Estudio	<i>NO Completada</i>
Diagramas de Secuencia	Documentación	<i>NO Completada</i>
Ampliar capítulo planificación	Documentación	<i>NO Completada</i>

Tabla 3.17: Tareas del Sprint 6

### 3.3.7. Sprint 7

La tabla 3.18 muestra las tareas acometidas en el Sprint realizado entre el 10 de mayo y el 24 de mayo de 2019.

**Sprint 7 - 10 de Mayo de 2019 - 24 de Mayo de 2019**

<b>Tarea</b>	<b>Tipo</b>	<b>Estado</b>
Exportar datos Trial	Gestión	Completada
Implementación: pruebas concepto	Documentación	Completada
Pruebas pipelines instancias Everis	Pruebas	Completada
Implementar CU-01	Implementación	Completada
Recolocar y completar Arquitectura Dynamics 365	Documentación	Completada
Requetar y finalizar contexto tecnológico	Documentación	Completada
Reestructurar Análisis e Implementación	Documentación	Completada
Reorganizar Introducción y Objetivos	Documentación	Completada
Ampliar Capitulo Planificación	Documentación	Completada
Añadir Tablas Sprints	Documentación	Completada
Implementar CU-02	Implementación	NO Completada
Securizar Credenciales con variables secretas	Implementación	NO Completada
Estudiar y Documentar Triggers	Implementación	NO Completada
Obtener lista completa de soluciones del sistema	Estudio	NO Completada
Estudiar formas de combinar varias soluciones	Estudio	NO Completada
Diagramas de Secuencia	Documentación	NO Completada

Tabla 3.18: Tareas del Sprint 7

### 3.3.8. Sprint 8

La tabla 3.19 muestra las tareas acometidas en el Sprint realizado entre el 24 de mayo y el 7 de junio de 2019.

**Sprint 8 - 24 de Mayo de 2019 - 7 de Junio de 2019**

Tarea	Tipo	Estado
Implementar CU-02	Implementación	Completada
Implementar Task Groups	Implementación	Completada
Nuevo Caso de Uso CU-03	Documentación	Completada
Documentar Agentes Autohospedados Locales	Documentación	Completada
Estudiar y Documentar Triggers	Documentación	Completada
Implementar Triggers	Implementación	Completada
Reorganizar Casos de Uso	Documentación	Completada
Diagrama de Secuencia CU-01	Documentación	Completada
Diagrama de Secuencia CU-02	Documentación	Completada
Securizar Credenciales con variables secretas	Implementación	Completada
Conclusiones	Documentación	NO Completada
Trabajo Futuro	Documentación	NO Completada
Manual de Usuario	Documentación	NO Completada
Obtener lista completa de soluciones del sistema	Estudio	NO Completada
Estudiar formas de combinar varias soluciones	Estudio	NO Completada
Diagrama de Secuencia CU-03	Documentación	NO Completada

Tabla 3.19: Tareas del Sprint 8

### 3.3.9. Sprint 9

La tabla 3.20 muestra las tareas acometidas en el Sprint realizado entre el 7 de Junio de 2019 y el 21 de Junio de 2019.

Sprint 8 - 7 de Junio de 2019 - 21 de Junio de 2019		
Tarea	Tipo	Estado
Implementar CU-03	Implementación	Completada
Solicitar Trials Demo y BackUp	Gestión	Completada
Documentar Implementacion Casos de Uso	Documentación	Completada
Manual Instalación / Administración	Documentación	Completada
Manual Usuario	Documentación	Completada
Actulizar Ultimos Sprints	Documentación	Completada
Diagramas Planificación	Documentación	Completada
Revisar Introducciones Capitulo	Documentación	Completada
Conclusiones	Documentación	Completada
Trabajo Futuro	Documentación	Completada
Resumen	Documentación	Completada
Maquetación Memoria	Documentación	Completada
Correcciones Memoria	Documentación	Completada

Tabla 3.20: Tareas del Sprint 9

Con el sprint 9 se finaliza el proyecto, habiéndose alargado 2 semanas mas de lo estimado inicialmente, pero finalizándose antes de la fecha de entrega del mismo fijada para el 26 de Junio de 2019.

## 3.4. Riesgos

La metodología SCRUM no suele tener asociada un gran análisis de riesgos inicial ya que los ciclos iterativos cortos ayudan a minimizar cualquier impacto imprevisto en el desarrollo del producto, permitiendo pivotar con facilidad para solventarlos.

Aun así se deciden identificar los principales riesgos que puedan comprometer el proyecto en algún momento. Estos se recopilan en la tabla 3.21

El riesgo R-01: Perdida de datos debida a fallo Hardware queda prácticamente mitigado en su totalidad al trabajar con servicios en la nube, y mantener copias locales. En el supuesto de fallo

Identificador	Descripción	Probabilidad	Impacto
R-01	Perdida de datos debido a fallo hardware	Baja	Alto
R-02	Fallo en la conectividad de la web	Baja	Medio
R-03	Problemas Integración de las soluciones con Dynamics 365	Alta	Media
R-04	Cambios en el funcionamiento de los servicios utilizados	Media	Media
R-05	Tecnología descontinuada o sustituida	Baja	Alta
R-06	Planificación demasiado optimista	Medio	Medio

Tabla 3.21: Riesgos del proyecto

de un equipo local. Se podría reanudar el trabajo y recuperar todos los datos desde la nube. Estos servicios suelen tener sus propias políticas de copias de seguridad, pero en el caso de que existieran pérdidas de datos en el servicio en la nube, las copias locales en distintos equipos y servicios cloud permitirían su rápida restauración.

El riesgo R-02: Fallo en la conectividad de la web, se mitiga disponiendo de varias tecnologías de acceso a la red por parte del desarrollador en este caso se dispone de conexión fija tanto en el hogar como en las dependencias de la universidad y empresa. A mayores se dispone de una línea de datos móviles que puede usarse como backup. El caso de que el problema de red afecte a los servicios en la nube, tiene una probabilidad menor, en este caso la capacidad de realizar trabajo en offline existe aunque se vería impactada la productividad.

El riesgo R-05: Tecnología descontinuada o sustituida presenta una baja probabilidad ya que se está utilizando tecnología novedosa que ha sido puesta en servicio durante los últimos años, con una gran apuesta por las empresas que la desarrollan. Por lo que tras el análisis las probabilidades de verse afectado por su discontinuidad son casi nulas.

Sin embargo la modificación de su funcionalidad contemplada en el riesgo R-04 es mucho más probable, ya que estas tecnologías en la nube suelen ir asociadas a un modelo de “Rolling Release” en el que se entregan actualizaciones con cambios y nuevas funcionalidades en cortos periodos de tiempo.[2] Generalmente se concede un acceso para probar las siguientes actualizaciones antes de que aparezcan por lo que el impacto en el proyecto podrá ser mitigado con antelación.

El riesgo R-06: Planificación demasiado optimista es un riesgo que puede aparecer al planificar un sprint con demasiada carga de trabajo y que esta no pueda ser acometida. Sin embargo la metodología utilizada permite afrontar estas desviaciones al inicio del siguiente sprint adaptando la planificación a la realidad del proyecto con el mínimo coste posible.

El riesgo R-03: Problemas Integración de las soluciones con Dynamics 365 puede aparecer al intentar integrar el desarrollo de este proyecto en instancias de Dynamics 365 que tengan un gran número de personalizaciones, por lo que este riesgo podría llevar a retrasos importantes.



Para mitigarlo se realizaran pruebas en distintos tipos de instancias de Dynamics y se probara la integración con una instancia muy similar a un entorno típico de producción.

## 3.5. Presupuesto económico

En cuanto a los costes para el desarrollo de este proyecto, se ha de tener en cuenta que debido al carácter de trabajo de fin de grado del proyecto se utilizaran versiones educativas y de prueba de la mayoría de los software con licencia, así como de los servicios en la nube y las herramientas de desarrollo. Por lo que el concepto de coste no tiene demasiado sentido. Aun así este apartado se incluye solamente con carácter especulativo, y no supone un coste real.

Para el presupuesto total del proyecto se han de tener en cuenta los costes de recursos humanos y los costes de Azure DevOps, así como de las herramientas necesarias para su ejecución, ya que el sistema CRM sobre el que se trabajaría generalmente corre a cuenta del cliente final y no de la empresa de desarrollo.

### 3.5.1. Costes Azure DevOps

Azure DevOps proporciona un conjunto de servicios (Boards, Pipelines, Repos, Test y Artifacts) de forma gratuita para proyectos de código abierto y para pequeños equipos de hasta 5 usuarios. Para equipos de mayor tamaño ofrece licencias mensuales en función del numero de miembros del equipo.

El coste de las licencias[30] esta recogido en la tabla 3.22. Así mismo se pueden ampliar las características como el numero de trabajos paralelos que se pueden ejecutar de forma simultanea con un coste incremental. Azure DevOps también ofrece una versión en entorno local “On Premise”.

Las características básicas de Azure DevOps incluidas en la licencia son:

**Azure Pipelines:** 1 trabajo hospedado con 1.800 minutos al mes para CI/CD y 1 trabajo auto-hospedado.

**Azure Boards:** seguimiento de elementos de trabajo y paneles kanban.

**Azure Repos:** repositorios GIT privados ilimitados.

**Azure Artifacts:** Administración de paquetes (5 usuarios gratis).

**Pruebas de carga:** 20.000 VUM/mes.

Cada trabajo paralelo adicional en Azure pipelines tiene un coste de 12,65€. [30]

Para este proyecto se ha utilizado el acceso gratuito para equipos pequeños, cuyas limitaciones son: Equipos de hasta 5 personas, 1 trabajo hospedado con 1.800 minutos al mes para CI/CD y 1 trabajo autohospedado. Es decir solo se puede tener un agente hospedado en Azure simultáneamente.

Usuarios	Coste Mensual
<5	Gratis
10	25,299 €
20	92,763 €
50	295,155 €
100	632,475 €
200	1138,455 €
1000	5.186,30 €

Tabla 3.22: Tabla de Precios Azure DevOps

### 3.5.2. Recursos humanos

En el proyecto participaran diferentes roles, el trabajo principal sera el de un Analista con experiencia encargado del análisis, diseño y la dirección de la implementación. Sobre el recaerá un 60 % del trabajo del proyecto. Sera necesario un perfil de desarrollador para realizar pequeñas pruebas de concepto y ayudar con la codificación general su aportación se estima en un 10 %. Como apoyo al analista se utilizará un arquitecto cuya estimación de trabajo sera de un 5 %. También sera necesario una persona de calidad que se encargue del diseño y ejecución de pruebas al cual se asigna un 20 % del tiempo total del proyecto. Finalmente se estima que un 5 % del tiempo del proyecto se dedicara a la Gestión del mismo, por lo que la figura del gestor se tiene en cuenta en los cálculos de personal. Los costes de recursos humanos se toman para el desarrollo del proyecto, sin tener en cuenta el ciclo de vida del CRM. El proyecto se estima en unas 300 horas de trabajo, realizándose el calculo de costes en base a estas. El desglose de horas y estimación de costes esta representado en la tabla 3.23.

	Horas	Coste Hora	Total
Desarrollador Junior	30	25,00 €	750,00 €
Analista Senior	180	35,00 €	6.300,00 €
Arquitecto	15	50,00 €	750,00 €
Testing	60	20,00 €	1200,00 €
Gestión	15	45,00 €	675,00 €
<b>Total</b>			<b>9.675,00 €</b>

Tabla 3.23: Presupuesto Recursos Humanos

## 3.6. Pruebas de Concepto

Para probar la viabilidad del proyecto y de los conocimientos obtenidos se decide realizar una serie de pruebas de concepto que permitan descartar rápidamente vías que no llevaran a la solución buscada.

Una de los primeros campos a probar es probar la conexión desde un programa externo y la automatización de ciertas acciones de Dynamics 365. Para ello se definen varias pruebas que tratan aspectos clave para la continuidad del proyecto:

- Conexión al CRM y Exportar soluciones.
- Extraer contenido de las soluciones
- Empaquetar soluciones
- Importar Soluciones al CRM
- Trasladar pruebas de concepto a una Pipeline

La implementación de estas pruebas de concepto se detallara en el capítulo 4.

## 3.7. Diagramas de Secuencia

La interacción entre los diferentes sistemas se ha modelado utilizando diagramas de secuencia, cada caso de uso se ha modelado en un diagrama.

- El caso de uso CU-01 se corresponde con el diagrama de secuencia 01 que se muestra en la imagen 3.3.
- El caso de uso CU-02 se corresponde con el diagrama de secuencia 02 que se muestra en la imagen 3.4.
- El caso de uso CU-03 se corresponde con el diagrama de secuencia 03 que se muestra en la imagen 3.5.

La plataforma web de Azure DevOps sera el principal medio de interacción con el usuario. Esta se encarga de recibir las ordenes de inicio, activar los Agentes y gestionar los comandos que se envían a estos. También provee de un interfaz visual del resultado de la ejecución. Asi mismo al final de la misma se encarga de notificar al usuario mediante diferentes vías. Una vez inicializada la maquina virtual esta se encarga de las diferentes tareas y la comunicación con Dynamics 365 y el repositorio de Azure Repos.

Estas funcionalidades se modelaran como si fueran clases atómicas que son "llamadas"por las pipelines.

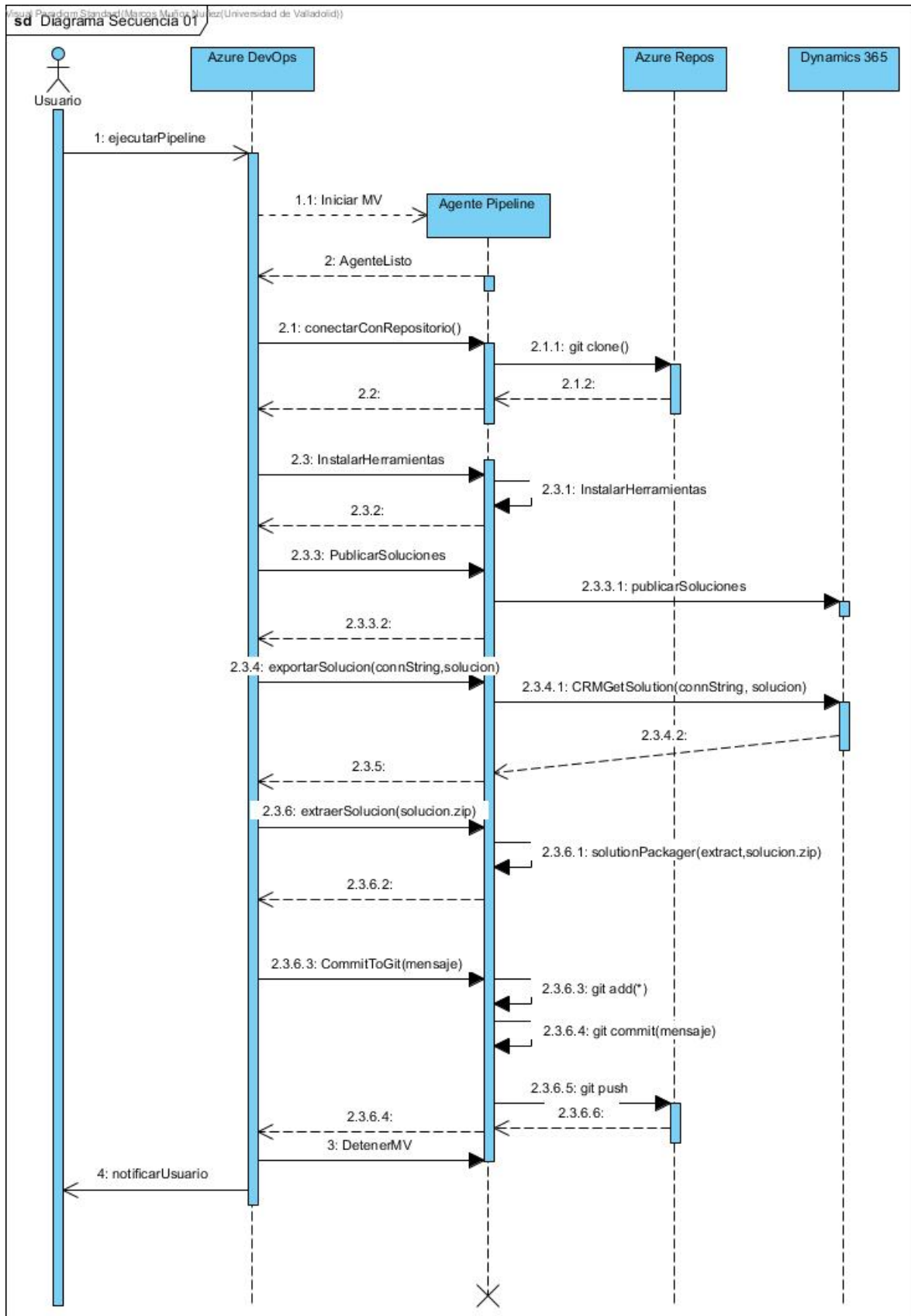


Figura 3.3: Diagrama de Secuencia 01

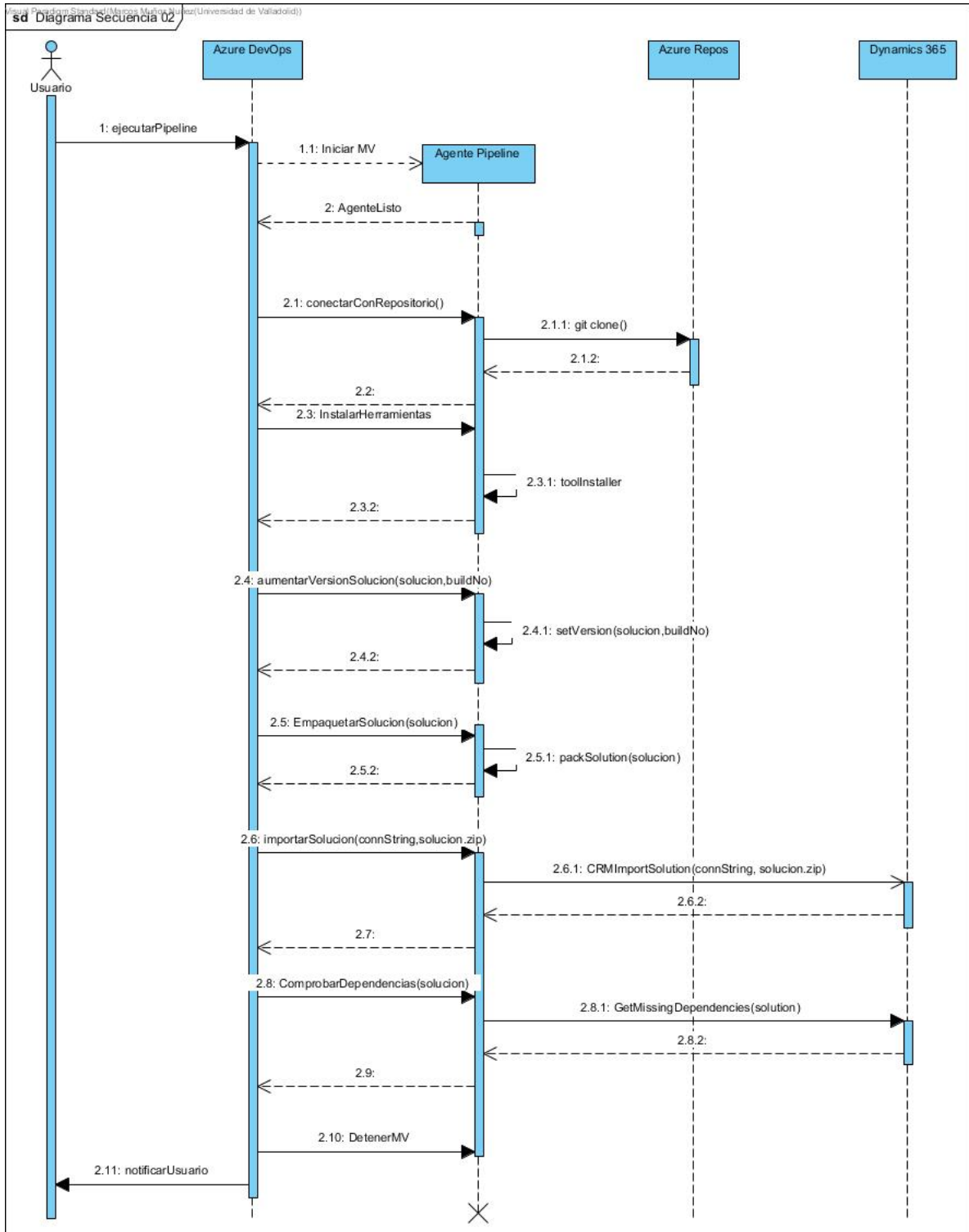


Figura 3.4: Diagrama de Secuencia 02

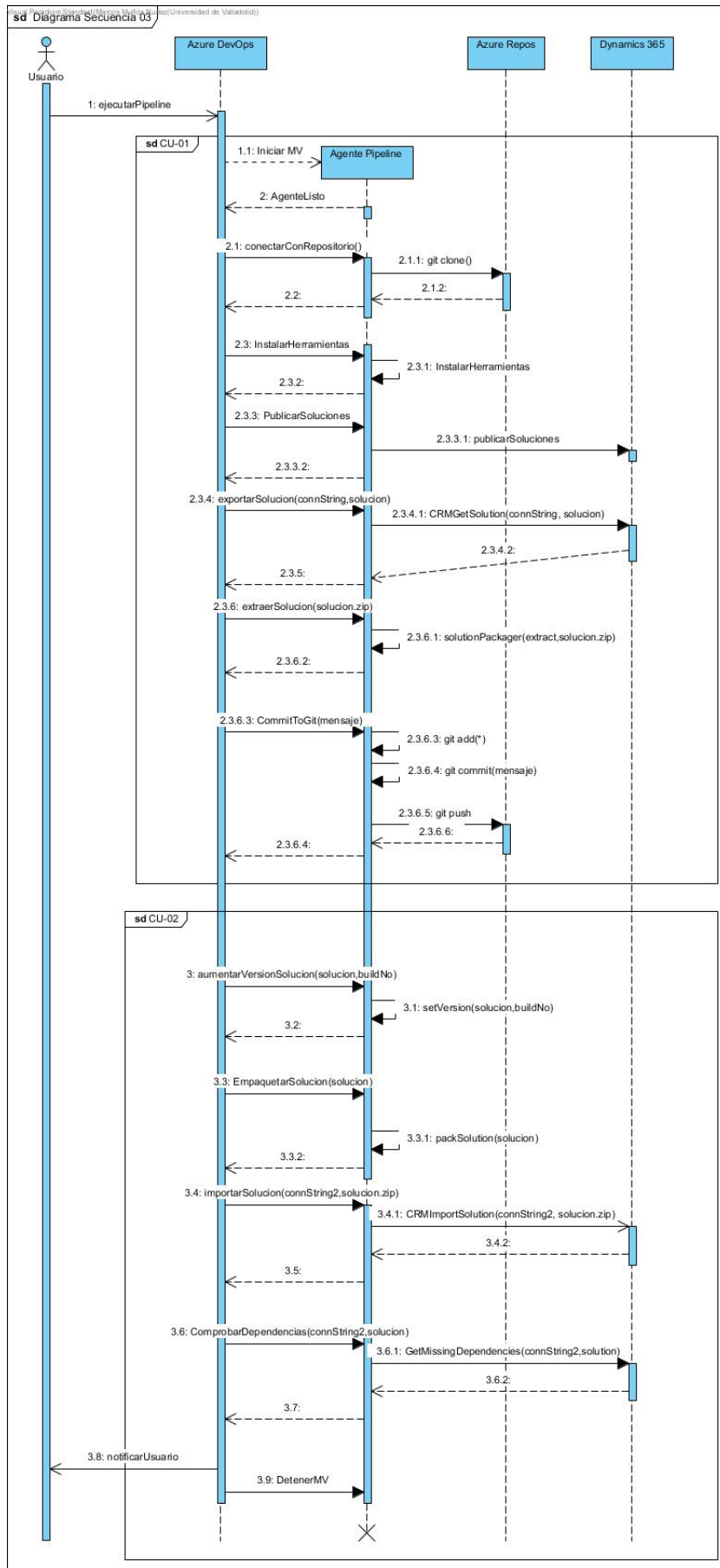


Figura 3.5: Diagrama de Secuencia 03

# Capítulo 4

## Implementación y Pruebas

En este capítulo se trataran los detalles mas relevantes en cuanto a la implementación del sistema. Esta implementación se ha realizado de forma iterativa, empezando con unas sencillas pruebas de concepto ejecutadas en la consola local para poder probar la viabilidad de las soluciones adoptadas y realizando la migración de estas pruebas de concepto a la nube. Una vez demostrada la efectividad de la solución se van incorporando nuevos componentes de Azure Pipelines y profundizando en la implementación como por ejemplo encapsulando la funcionalidad con grupos de tareas o “Task Groups”, añadiendo seguridad sobre las credenciales guardadas haciendo uso de variables secretas, implementando el uso de triggers.

### 4.1. Primeras pruebas de Concepto

Para probar los conceptos estudiados y demostrar su viabilidad se realizaron unas pruebas de concepto iniciales:

#### 4.1.1. Exportar Soluciones desde Dynamics 365

La primera prueba de concepto realizada fue la de utilizar scripts consola probar la conexión con el sistema Dynamics 365. Las soluciones se pueden exportar de forma manual, pero el objetivo es encontrar una forma de automatizar este proceso. Para la realización de esta prueba se investigo sobre las diferentes formas de conectarse y realizar operaciones en crm de forma automatizada. Para comenzar se elige el uso de un script de consola Powershell por su facilidad para ser trasladado e integrarse con las distintas herramientas de Azure. En esta investigación se encuentra un conjunto de herramientas que permiten realizar distintas acciones en Dynamics 365 a través de la consola. [37] Para realizar esta prueba se preparo el script que se puede encontrar en el Anexo C, en el que se comprueba si las herramientas de consola de Dynamics 365 [37] están instaladas y si no es así se procede a su instalación. Tras lo cual se configura la conexión a CRM con los datos proporcionados, y trata de recuperar la solución indicada. En las primeras ejecuciones de esta prueba de concepto la consola devolvía un error de timeout el cual fue solucionado configurándolo con:

```
Set-CrmConnectionTimeout -conn $CrmConn -TimeoutInSeconds 1000
```

Se ha de tener en cuenta que este parámetro modifica el timeout de la conexión, pero no los que pueda tener configurados internamente Dynamics 365. Las pruebas posteriores arrojaban resultados dispares en cuanto a los tiempos de fallo, finalmente se descubrió que se debía a como funcionan las instancias de trial, ya que utilizan los recursos que están libres en cada momento, no siendo consistentes en el tiempo que tardan en realizar las mismas acciones en diferentes momentos del día. Para asegurar que estos problemas se debían a las trials, y no a la configuración del CRM o los scripts se realizaron las mismas pruebas sobre instancias Dynamics365 de Everis. En las instancias comerciales se conseguía un rendimiento estable en el tiempo y los resultados eran



mucho mas coherentes. Aun así puede ocurrir que con soluciones de gran tamaño se necesario aumentar el tiempo que la conexión permanece esperando la respuesta del CRM.

Tras esta primer prueba de concepto, el siguiente paso era estudiar el contenido de las soluciones exportados y ver como identificar los cambios en estas.

#### 4.1.2. Extraer contenido de la solución

Como ya hemos visto es posible extraer soluciones de Dynamics 365 de forma automatizada, este nos proporciona un archivo zip que contiene varios archivos y carpetas. Como se muestra en la imagen 4.1, este contiene varias carpetas con los recursos web, ensamblados dll, y workflows. En la raíz encontramos tres grandes archivos XML, que contienen la información sobre las personalizaciones realizadas, el contenido de esta solución, y los tipos de ficheros que la componen. Las nuevas entidades, así como las modificaciones sobre las existentes se desgranar dentro de el fichero customizations.xml, siendo un XML muy denso y de difícil lectura o modificación.

```

customizations.xml
solution.xml
[Content_Types].xml

PluginAssemblies
└── SIEPluginsFORValidarFechaCreacionVSFechaReal-8E0AF57B-92BD-42B7-9037-2631BF07B965
    SIEPluginsFORValidarFechaCreacionVSFechaReal.dll

WebResources
sie_AccionFormativa_mainjsA6141FF8-4C62-E811-A999-000D3A387681
sie_icono_preseleccionar_enviar_citacion_multicana0D6D29DD-1410-E911-A2D3-005056B50926
sie_icono_preseleccionar_enviar_citacion_multicana664E7530-1510-E911-A2D3-005056B50926
sie_icono_vista_ampliada_resultados_16png13B69A89-1410-E911-A2D3-005056B50926
sie_icono_vista_ampliada_resultados_32png30A82D37-1410-E911-A2D3-005056B50926
sie_RelacionDemandaSondeo_quickFormsBA62AD9F-390F-E911-A2D3-005056B50926
sie_RelacionDemandaSondeo_ribbonjs145132B7-F30F-E911-A2D3-005056B50926
sie_rolejsF7858A57-9419-E911-A2D3-005056B50926
sie_SondeoCitacionMulticana1_ribbonjs83467FFD-2609-E911-A2D3-005056B50926
sie_sondeoscitacionesmulticana1_mainjsB782B0BE-5CD1-E811-AA2A-000D3A387681

Workflows
AlumnoGenerarNombre-FA68EEB9-2B53-4EF4-BB63-4466472D8A77.xaml
Bloquearcamposdesondeo-8650CF8F-A6D6-E811-AA31-000D3A387681.xaml
BloquearcamposIniciovigenciaFinvigenciaImporteeImp-280F28E8-D818-E911-A2D3-005056B50926.xaml
BloquearcamposIniciovigenciaFinvigenciaImporteeImp-9E9FA2EF-9015-E911-A2D3-005056B50926.xaml
Creartareaestadocursocambiaaestadosolicitandocandi-0E61AEBF-E8EF-4B93-AA08-ECB58E43D7D9.xaml
DeshabilitarcampossegnlestadodelaAccinFormativa-CD653C22-B394-E811-A9DB-000D3A387681.xaml
Valorpredeterminadodelnumerodecopiasdelsondeo-A09F7BD6-010A-E911-A2D3-005056B50926.xaml

```

Figura 4.1: Árbol Contenido Zip Extraído

Esto dificulta el seguimiento de los cambios sobre las diferentes entidades y configuraciones del sistema. Microsoft en su documentación proporciona una herramienta llamada SolutionPackager, que identifica cada componente individual y lo extrae a archivos individuales, esto permite añadir los archivos resultantes a un SCV de forma que se puede tener un control del los cambios realizados sobre cada componente individual.

Esta es una herramienta que se ejecuta desde consola de comandos y actúa contra el fichero Zip indicado como entrada. Los diferentes comandos y opciones están detallados en la documentación que proporciona Microsoft[36]. Tras aplicar la herramienta SolutionPackager sobre el archivo Zip exportado, obtenemos el directorio de directorios que se muestra en la imagen 4.2. Donde se puede ver que los cambios en las entidades se agrupan en la carpeta Entities, dentro de la cual hay un carpeta por cada entidad. Así mismo dentro de estas carpetas vemos las de cada tipo de cambio,

por ejemplo las carpetas FormXml contienen los cambios en cada tipo de formulario de la entidad. Estando estos desgranados en otro nivel de subcarpetas por debajo, en las que se encuentran los archivos XML que detallan las configuraciones y modificaciones.

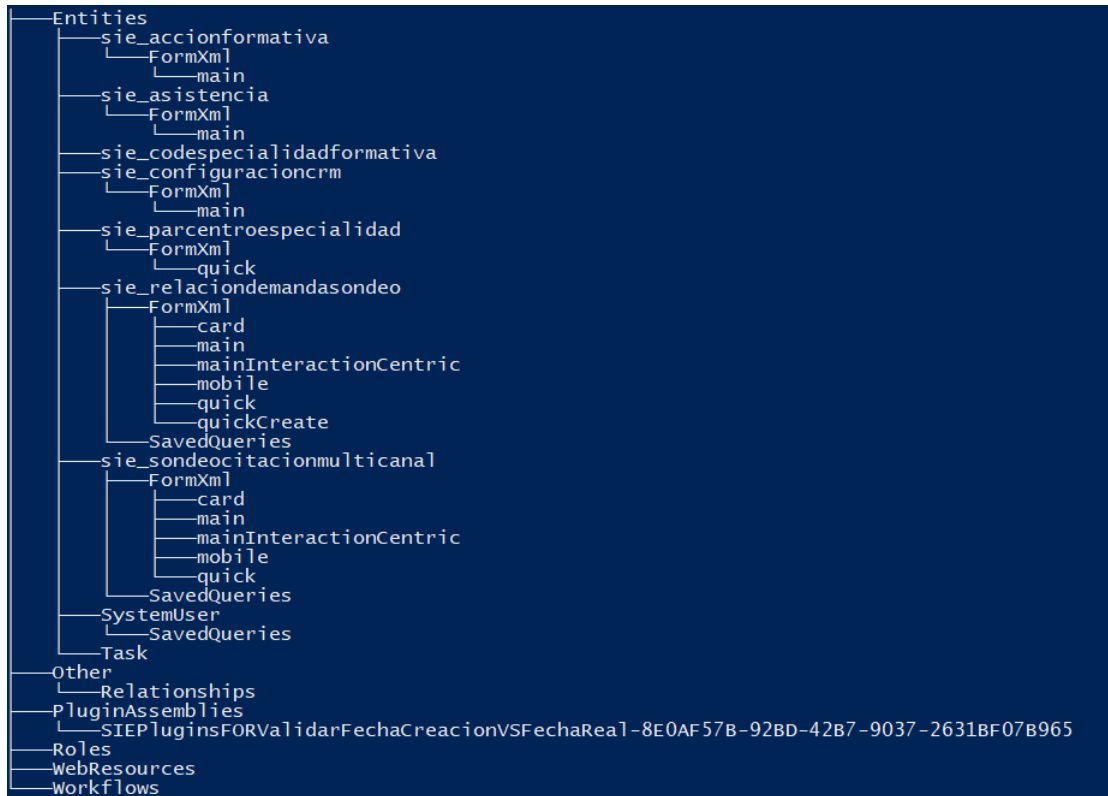


Figura 4.2: Árbol Directorios Solución Extraída

De esta forma hay un gran numero de pequeños archivos XML que detallan los cambios en componentes muy concretos del CRM, lo cual facilita la tarea de llevarlo a un control de versiones, y poder seguir e identificar las modificaciones que se han realizado a lo largo del tiempo, incluso cuando hay varios desarrolladores trabajando en diferentes partes del sistema.

### 4.1.3. Empaquetar solución a partir de los archivos obtenidos

Una vez hemos conseguido exportar la solución del crm y extraer su contenido de forma que pueda ser incluido en el SCV se ha de comprobar si tras repetir estas operaciones varias veces, combinando en el repositorio archivos de varias versiones se puede volver a empaquetar. Para ello volveremos a hacer uso de la herramienta SolutionPackager indicando el directorio desde el que se desea construir la solución, y el zip de destino. El empaquetado de vuelta no parece presentar ningún problema y permite su automatización de forma sencilla con la herramienta . Esta herramienta también permite el uso de un fichero XML que mapee los diferentes tipos de archivo con diferentes carpetas. Esto permite su aplicación a diferentes escenarios muy interesantes a la hora de construir un sistema de integración continua, como por ejemplo que tome los recursos

web, o los ensamblados dll de otra carpeta diferente, por ejemplo el repositorio donde se almacenen las ultimas versiones tras pasar por un sistema de compilado y pruebas automáticas. Asegurando así que las soluciones se construyen con la ultima versión disponible de estos archivos.

#### 4.1.4. Importar solución a Dynamics 365

Para subir la solución empaquetada de nuevo a Dynamics 365 se puede realizar de forma manual, lo cual fue probado con éxito, pero era necesario al igual que para exportarla buscar una forma de realizarlo programaticamente, al igual que en la exportación se intentara realizar un script de consola que permita la automatización de la subida. Las herramientas de consola de Dynamics 365 [37] incluyen un comando para realizar la importación al igual que se hizo anteriormente con la exportación. Por lo que realizamos la prueba de ese comando usando como base el script C, que fue ampliado para incluir la subida de una solución ya que la conexión con el CRM puede reutilizarse. Este proceso puede ser mas complicado que la exportación, ya que de haber algún problema con los archivos de la solución el CRM devolverá un error y la importación se cancelara. Es importante tener en cuenta esto a la hora de gestionar los posibles errores o de integrar un sistema que compruebe la integridad de la solución antes de empaquetarse e importarse al CRM de nuevo.

## 4.2. Prueba de concepto en Azure Pipelines

La implementación se realizara utilizando las herramientas que proporciona Azure DevOps, concretamente Azure Pipelines. Para probar la viabilidad de la implementación vamos a llevar los scripts que han sido probados en consola a una pipeline de la forma mas sencilla posible. Se creara un proyecto en Azure DevOps que contendrá los diferentes repositorios, pipelines y tableros. Este proyecto se crea de forma privada por lo que no es libremente accesible y solo los miembros designados pueden acceder a su contenido.

La implementación inicial constara de un repositorio git, y una pipeline donde se configuraran los scripts a ejecutar. El proceso de creación de pipelines esta guiado por una asistente, estas puede crearse a partir de un archivo YAML Ain't Markup Language (YAML) o un asistente interactivo. Los archivos YAML pueden introducirse en el repositorio de forma que permiten editar la pipeline como si de código se tratase. En el primer paso indicaremos el repositorio desde el cual tomara el código, en nuestro caso seleccionamos el repositorio que acabamos de crear. Tras lo cual el asistente muestra diferentes plantillas, en este caso no se utilizará ninguna de ellas si no que se creara una pipeline vacía. Tras esto añadiremos una nueva tarea de tipo PowerShell Script, y en la configuración de esta se introducirá el script previamente preparado.

Tras esto guardamos y encolamos, de forma que la pipeline se ejecuta en cuanto se la asigna un agente. Comprobamos a través de la salida de consola que proporciona Azure Pipelines que el script se ejecuta correctamente.

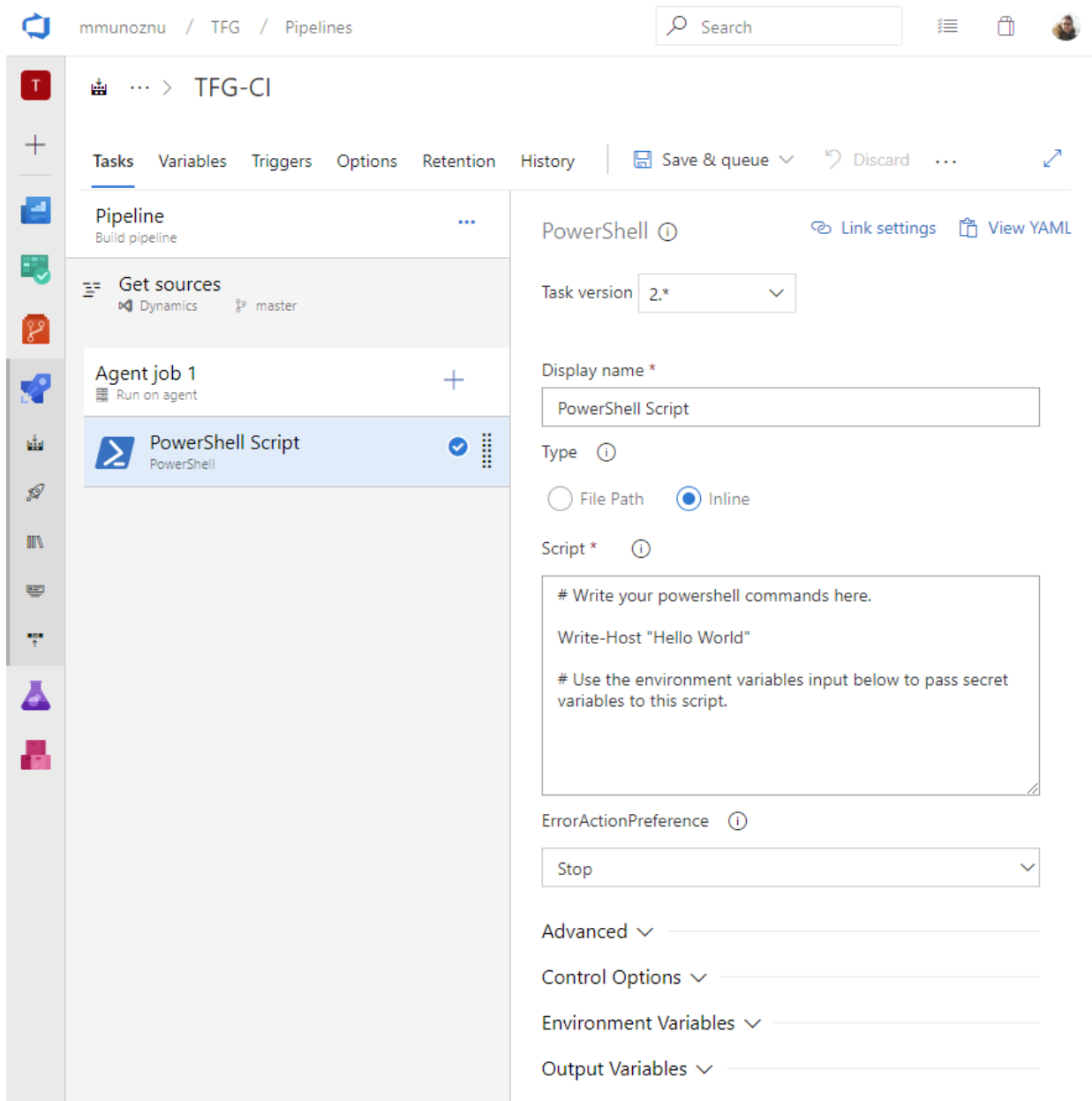


Figura 4.3: Creación Pipeline: Añadir Tareas al Agente

### 4.3. Grupos de Variables

Durante el análisis y diseño se decide que se utilizaran grupos de variables para guardar los datos de conexión al CRM. Los grupos de variables permiten agrupar variables de un mismo contexto así como proteger su contenido. También proporcionan un control granular de los permisos de acceso y modificación permitiendo decidir que privilegios tiene cada usuario, ocultando así la información sensible al resto. Los grupos de variables se almacenan en la biblioteca de Azure Pipelines.

Los grupos de variables que guardan la información de conexión al CRM están formados por: **AuthType** Almacena el tipo de autenticación (Office365,ActiveDirectory ...)

**Password** Se utilizará una variable secreta para almacenar la contraseña.

**Url** Dirección de la instancia CRM a la que se conectará.

**Username** Nombre de usuario en CRM.

The screenshot shows the configuration page for a variable group named 'Trial Dynamics 7'. The page includes a breadcrumb 'Library > Trial Dynamics 7', a 'Variable group' header with 'Save', 'Clone', 'Security', and 'Help' actions, and a 'Properties' section. The 'Variable group name' is 'Trial Dynamics 7' and the 'Description' is 'Trial 7 - Base para la demostración en la presentación del TFG'. There are two toggle switches: 'Allow access to all pipelines' (checked) and 'Link secrets from an Azure key vault as variables' (unchecked). Below is a 'Variables' table with columns 'Name' and 'Value'. The 'Password' variable is highlighted in grey and has a trash icon next to its masked value '\*\*\*\*\*'. Other variables include 'AuthType' (Office365), 'Url' (https://tfg07.crm4.dynamics.com), and 'Username' (marcos@tfg07.onmicrosoft.com). A '+ Add' button is at the bottom.

Name ↑	Value
AuthType	Office365
Password	*****
Url	https://tfg07.crm4.dynamics.com
Username	marcos@tfg07.onmicrosoft.com

Figura 4.4: Ejemplo Grupo Variables Credenciales

## 4.4. Implementación CU-01: Recuperar Cambios en CRM

En el caso de uso 01 se describe como se han de recuperar los cambios del CRM e incorporarlos al repositorio GIT. Para ello el análisis y diseño previos se decide implementar este caso de uso haciendo uso de las Pipelines y los “Task Group” o grupos de tareas que permiten encapsular tareas de forma que para la pipeline estas actúen como una sola con un único punto de entrada para las variables. Es decir, se creara una Pipeline que ejecuta el grupo de tareas o “Task Group”, refiriéndonos a este como TG-01.

### 4.4.1. Implementación del grupo de tareas TG-01

El “Task Group” TG-01 implementara la funcionalidad principal del caso de uso CU-01: Recuperar Cambios en CRM. En este “Task Group” se realizan las siguientes tareas:

- Instalar las Herramientas
- Publicar las personalizaciones en CRM
- Exportar la Solución Personalizada
- Extraer la Solución
- Configurar el GIT y realizar el Commit.

La instalación de las herramientas en el agente se ha de realizar cada vez que se ejecuta la pipeline al utilizar Agentes Hosteados que siempre ofrecen una maquina virtual limpia en cada ejecución. La publicación de personalizaciones es una orden en CRM que debe ejecutarse antes de la exportación ya que solo se exportan los cambios que han sido publicados. La exportación de la solución CRM solicita esta al CRM y obtiene el archivo ZIP comprimido que contiene las personalizaciones realizadas. Tras esto se configura el GIT, es decir se indica el nombre y correo con el que se quiere hacer el commit, esto permite indicar que esa inserción de código ha sido hecha por un usuario específico, o por un proceso automático, así mismo se imprime por consola el estado actual del repositorio por si fuera necesario para depurar la ejecución. Tras esto se descomprime la solución y se realiza un commit con todos los archivos que han sido modificados. Tras esto se suben los cambios al repositorio remoto. Y finaliza el grupo de tareas. Esta lista de tareas se muestra en la imagen 4.5

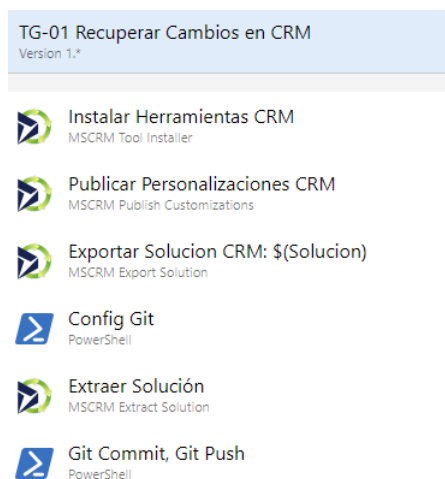


Figura 4.5: TaskGroup TG-01: Tareas a ejecutar

En este “Task Group” se expondrán los siguientes parámetros:

- Mensaje de Commit
- Solución a Exportar
- Datos de Conexión

Estos han de indicarse en la pipeline ya sea con el uso de variables prefijadas, indicadas por el usuario en el momento de la ejecución, o mediante grupos de variables. Cuando se realiza un commit en el repositorio GIT va acompañado de un mensaje descriptivo este estará formado por la etiqueta [AUTO], el numero de Build en el que se ha realizado el commit, el nombre de la solución exportada y un mensaje opcional que el usuario puede Añadir. El numero de build es un numero auto-incremental que se toma de las variables del sistema que ofrece Azure Pipelines y se utiliza como `$(Build.BuildNumber)`.

El nombre de la solución a exportar se pasa por parámetro al lanzar la pipeline y si no se indica se utilizará la solución por defecto. Los datos de la conexión se pasan por parámetro, para almacenarlos se utilizan grupos de variable que permiten mantener estos seguros, definir permisos granulares sobre quien puede ver o editarlos, así como intercambiarlos con facilidad. En el task group se compone la Connection String, la cadena que sirve para conectarse al CRM, sustituyendo los valores a indicar por variables que el introducirá o asignara a un grupo de variables. Para concatenar el valor de una variable en una cadena en Azure Pipelines se utiliza el símbolo \$ seguido de el nombre de variable entre paréntesis. Se eligen estos para ser lo mas descriptibles posible. “AuthType=\$(Authtype);Username=\$(Username); Password=\$(Password);Url=\$(Url)”. En la imagen 4.6 se muestra los parámetros de entrada que se han expuesto en el Task Group.

Los “Task Group” pueden exportarse de forma que sean importados en otros proyectos de Azure DevOps, lo cual permite su portabilidad. Este es el método que se va a usar para instalar los “Task Group” en otros proyectos, la exportación genera un codigo JSON. El codigo JSON perteneciente al “Task Group” 01 puede consultarse en el anexo D.

GetSoluciones - Carpetas ⓘ

Task version 1.\* ▾

Display name \*  
Task group: GetSoluciones - Carpetas

Authtype \*  
\$(AuthType)

CoomitMsg \* ⓘ  
\$(MensajeCommit)

Password \*  
\$(Password)

Solucion \* ⓘ  
\$(Solucion)

Url \*  
\$(Url)

Username \*  
\$(Username)

Figura 4.6: TaskGroup TG-01 : Entrada de Parámetros

#### 4.4.2. Implementación Pipeline CU-01

Una vez publicado el “Task Group” este puede ser importado desde una Pipeline. En esta deberemos configurar el repositorio desde el que se leerán y escribirán los datos, el agente o agentes que realizaran las tareas y añadir el “Task Group”, al añadirlo se mostrara la configuración relativa a sus parámetros de entrada, como se muestra en la imagen 4.6 En la pipeline configuramos la variable de mensajeCommit y Solucion para que el usuario pueda modificarlas al lanzar la pipeline. Vinculamos el grupo de variables que contiene las credenciales del CRM y se asignaran a los parámetros de entrada del “Task Group”. También se programan los triggers para que se ejecute automáticamente. Con esto esta pipeline esta lista para ser ejecutada. A la hora de ser ejecutada el usuario podrá indicar el mensaje de commit y la solución de la que se recuperaran los cambios. A diferencia de los “Task Group” las pipelines aun no se pueden exportar como una única entidad, por lo que deben configurarse a mano. El equipo de desarrollo de Azure Pipelines ha confirmado estar trabajando en que se pueda realizar la exportación de Pipelines junto a los “Task Group” que los conforman en un futuro próximo, lo cual simplificaría la posterior instalación por parte de los usuarios finales.

### 4.5. Implementación CU-02: Restaurar Cambios a CRM

El caso de uso 02 describe como se han de recoger los cambios del repositorio GIT e insertarlos en de vuelta al CRM. Al igual que en el caso de uso anterior se hará uso de las Pipelines y los



“Task Group” o grupos de tareas que una vez mas encapsularan la funcionalidad principal del caso de uso. Es decir, se creara un grupo de tareas o “Task Group” al que nos referiremos como TG-02 y este sera ejecutado desde una Pipeline.

#### 4.5.1. Implementación del grupo de tareas TG-02

El “Task Group” TG-02 implementara la funcionalidad principal del caso de uso CU-02: Restaurar Cambios a CRM. En este “Task Group” se realizan las siguientes tareas:

- Instalar las Herramientas
- Aumentar el numero de versión de la solución a empaquetar
- Empaquetar la Solución
- Importar solución al CRM

Al igual que en los casos anteriores la instalación de las herramientas en el agente se ha de realizar cada vez que se ejecuta la pipeline ya que los Agentes Hosteados ofrecen una maquina virtual limpia en cada ejecución. Se aumenta el numero de versión del CRM para poder identificar con que ejecución de la Pipeline se corresponde la versión que esta subida al CRM para ello se utiliza la estructura 1.0.0.X donde la X se sustituirá por el numero de Build, este se obtiene de una de las variables de sistema que ofrece Azure Pipelines \$(Build.BuildNumber). Como se explico anteriormente este numero es auto-incremental y aumenta cada vez que se ejecuta una Pipeline. En el GIT se almacenan los datos de la solución descomprimidos y estructurados en carpetas, para volver a subir estos datos al CRM se ha de recomponer el fichero zip de la solución. Una vez se obtiene el fichero ZIP, este es Importado de vuelta al CRM, una vez este se ha subido correctamente finaliza el grupo de tareas. La imagen 4.7 muestra las tareas que ejecutara este “Task Group”



Figura 4.7: TaskGroup TG-02: Tareas a ejecutar

Los parámetros de entrada del TG-02 serán los datos de la conexión al CRM (Usuario, Password, Url...) y el nombre de la solución que se desea subir al CRM. Al ser los mismos que en el TG-01 en la imagen 4.6 pueden consultarse los parámetros de entrada que se han expuesto. Estos son:

- Solución a Importar
- Datos de Conexión

De estas variables solo la solución a importar puede ser modificada por el usuario a la hora de ejecutar la pipeline.

Como se comento anteriormente los “Task Group” pueden exportarse de forma que sean importados en otros proyectos de Azure DevOps, la exportación genera un código JSON que se utilizará para instalar el “Task Group” en otros proyectos. El código JSON perteneciente al “Task Group” 01 puede consultarse en el anexo E.

### 4.5.2. Implementación Pipeline CU-02

En este caso de uso la Pipeline se encargara de recuperar los datos del repositorio GIT y ejecutar el grupo de tareas TG-02. Lo primero es indicar el repositorio del cual se obtendrán los datos, así la pipeline recuperara los cambios una vez inicializado el agente. El “Task Group” TG-02 sera añadido a continuación, al igual que en el caso anterior se configuraran sus parámetros de entrada utilizando el grupo de variables que contiene los datos de conexión del CRM. En este caso no se configuraran triggers de ningún tipo ya que esta se ejecutara únicamente de forma manual. Al igual que en el caso anterior las Pipelines aun no se pueden exportar a un archivo. El equipo de desarrollo de Azure Pipelines ha confirmado estar trabajando en esta se pueda realizar en un futuro próximo, lo cual simplificaría la instalación por parte de los usuarios finales.

## 4.6. Implementación CU-03: Desplegar Cambios a Otro Entorno

Este tercer caso de uso nace como un ejemplo practico de la combinación de los casos de uso anteriores. Parte de la idea de facilitar el trabajo de despliegue y mantener un historial fácil de consultar de las soluciones desplegadas. Para la implementación de este caso de uso no sera necesario la creación de nuevos grupos de tareas, ya que se hará uso de los TG-01 y TG-02 implementados en los dos casos de uso anteriores. En este caso de uso participaran dos instancias de Dynamics CRM distintas, a una la llamaremos instancia de Origen y la otra sera la instancia de Destino.

### 4.6.1. Implementación Pipeline CU-03

Para este caso de uso se implementara una pipeline en la que se configurara el repositorio GIT en el cual se desea que se recupere la información de los cambios realizados en el entorno de Origen. Tras esto se añade el “Task Group” TG-01 y se configuran sus parámetros de entrada. Para este utilizaremos el grupo de variables que contiene los datos de conexión del servidor de Origen. Tras esto se añade el “Task Group” TG-02, se vincula otro grupo de variables que contenga los datos

del entorno de Destino y se configuran estas en los parámetros de entrada del TG-02. En este caso las variables que se configuran en la pipeline son:

- Mensaje de Commit
- Solución a Desplegar
- Datos de Conexión

Cuando se realiza el commit en el repositorio GIT con los archivos a desplegar, este va acompañado de un mensaje descriptivo, formado por la etiqueta [AUTO], el numero de Build en el que se ha realizado el commit, el nombre de la solución desplegada, la etiqueta [DEPLOY] y un mensaje opcional que el usuario puede Añadir. El numero de build es un numero auto-incremental que se toma de las variables del sistema que ofrece Azure Pipelines y se utiliza como \$(Build.BuildNumber). En este caso el usuario puede configurar el mensaje del commit y el nombre de la solución a desplegar a la hora de ejecutar la pipeline.

De nuevo recalcar que a diferencia de los “Task Group” las pipelines aun no pueden ser exportadas a archivos externos por lo que deben configurarse a mano desde la interfaz web. El equipo de desarrollo de Azure Pipelines ha confirmado estar trabajando en que se pueda realizar la exportación de Pipelines junto a los “Task Group” en un futuro próximo, lo cual simplificaría la posterior instalación por parte de los usuarios finales.



# Capítulo 5

## Conclusiones

A lo largo de los últimos años las técnicas de Integración Continua, Despliegue Continuo y un seguimiento mas cercano de los desarrollos combinado con las metodologías ágiles han propiciado la expansión del termino que de alguna forma aún todas estas, DevOps. Todo esto busca un objetivo común: mejorar la calidad del software y reducir el tiempo de desarrollo reduciendo las complicaciones derivadas del mismo. Con este trabajo se ha podido ver que estas practicas pueden aplicarse a a todo tipo de desarrollos, incluidos aquellos en los que el código no es el eje central del mismo. Las ventajas son claras: ciclos de desarrollo mas cortos, detección temprana de errores, seguimiento preciso del código y vinculación entre este, las tareas y los errores. Su principal desventaja es que necesitan de conocimientos adicionales para poder utilizar estas herramientas, así como tiempo para interiorizar las metodologías y flujos asociados. Pero queda patente que el tiempo que se emplea en la formación del equipo y la integración de estas herramientas aporta muchas ventajas temporales, económicas e incluso personales y que además pueden aplicarse a cualquier tipo de proyecto.

Con su popularización también han surgido nuevas herramientas y mejorado las existentes. En este punto la adición de los servicios cloud a la ecuación facilita su uso y aplicación de nuevas y diferentes formas. Facilitando el acceso y la integración de estas con servicios como Azure DevOps, que proporciona un set de herramientas muy completo, fácil de integrar con casi todo tipo de desarrollo y que proporciona una interfaz web que permite su configuración de forma intuitiva y fácil de usar. Además las ventajas de poder usar agentes en la nube hacen que se puedan probar estos sistemas sin realizar una inversión en hardware o tiempo en configurar servidores.

Los programas CRM son muy demandados por las empresas y Dynamics 365 es uno de los mas populares, capaz de adaptarse múltiples escenarios gracias a sus capacidades de personalización, sus desarrollos pueden implicar a grandes equipos, con los problemas de comunicación y coordinación que esto puede conllevar. Implementar DevOps en estos desarrollos no es una tarea trivial, ni frecuente si lo comparamos con otros tipos de desarrollos software, pero puede suponer un gran ahorro en resolución de conflictos, seguimiento de los cambios, facilidades en el despliegue a entornos productivos, automatización de pruebas, compilación e integración automática de los diferentes activos de código que permitan detectar problemas de forma temprana .

En este proyecto se han puesto las bases para un sistema de integración continua en Dynamics 365 con el que comenzar aplicando las metodologías DevOps y como trataremos mas adelante puede seguir creciendo y mejorando añadiendo mas características que faciliten y aceleren los desarrollos sobre Dynamics365. En la realización del proyecto se han completado los siguientes objetivos:

- Aprendí el funcionamiento de Dynamics 365 y las múltiples formas de personalizarlo y extenderlo.
- Adquirí mayor conocimiento de como se estructuran muchos de los servicios en la nube así como varias aplicaciones de estos.
- He obtenido un gran conocimiento las metodologías DevOps y los conceptos de Integración y Entrega Continua y como implementar estas en diferentes contextos.

- Adquirí competencias con los servicios de Azure DevOps aplicables a muchas otras herramientas de CI y CD.
- Aplique los conocimientos adquiridos para analizar y diseñar un sistema que permitiera la recuperación y almacenamiento de las personalizaciones realizadas en el CRM, así como su restauración.
- He realizado la implementación de este sistema sobre Azure DevOps aplicando las competencias adquiridas durante el proyecto.

Personalmente durante este proyecto he ampliado mi conocimiento de las arquitecturas, servicios y aplicaciones Cloud, profundizando en el valor añadido que estas pueden aportar en múltiples campos, y sobretodo en las facilidades que ofrecen a los desarrolladores. La oportunidad de realizar el trabajo en una empresa como Everis me ha permitido ver como los conocimientos adquiridos durante los estudios de grado encajan y aportan un enfoque diferenciador en el mundo laboral. Ya que la solida base de conocimiento y el enfoque adquirido en estos años me ha permitido enfrentarme a problemas y tecnologías que no había abordado con anterioridad. Además he descubierto en las DevOps una forma de aunar mi pasión por la resolución de problemas y la administración e integración de sistemas con la posibilidad de mejorar los desarrollos de software, proporcionar herramientas a los equipos que involucren a los desarrolladores y la búsqueda constante de mejora en la calidad y productividad.

## 5.1. Trabajo Futuro

Desde el principio del proyecto este se enfoca a proporcionar una base sobre la que seguir trabajando y mejorando, durante el proyecto se han detectado puntos de mejora y evoluciones concretas con las que ampliar el proyecto.

Una de las primeras a acometer es seguir ampliando el concepto de integración continua, utilizando las Pipelines para Compilar los ensamblados DLL cuyo código se almacena en un repositorio diferente al de las configuraciones CRM, y utilizar estos ensamblados contraídos a partir de la ultima versión de código disponible en el repositorio para construir junto a las configuraciones de CRM una solución que integre todas las piezas que acaban formando la personalización del CRM.

Otra de las mejoras propuestas es la profundización en el uso de Agentes Locales que permitan trabajar en contextos empresariales donde la conexión al CRM se realiza a través de una VPN, o que permitan realizar despliegues detrás de los firewall. La combinación de estos agentes locales con los Hospedados en la nube es otra de las ramas a trabajar para poder conseguir compilaciones rápidas y frecuentes combinadas con una integración y despliegue sencillo dentro de los servidores locales.

Implementación de Pipelines de despliegue implicando en ella a los equipos de testing, y los encargados de aprobar los despliegues de forma que los procesos de integración, pruebas, y aprobación sean mas rápidos y con un mayor automatismo.

Este trabajo solo araña la superficie de lo que se puede lograr aplicando DevOps en Dynamics365 y deja abiertas muchas lineas en las que profundizar e investigar como: integración con los sistemas de comunicación y notificaciones del equipo, integración con los equipos de testing, añadir incidencias a los sistemas de gestión cuando un test falla o una compilación no se ejecuta correctamente, realizar un seguimiento mas detallado del tiempo empleado para un trabajo así como estadísticas de los ratios de errores, bugs y fallos presentes en el sistema.



# Glosario

- API** Un Interfaz de programación de aplicaciones (Application Programming Interface) es un conjunto de características y reglas que existen dentro de un programa software o aplicación, que habilitan la interacción con el a través de software. Las APIs pueden ser vistas como un simple contrato entre la aplicación que lo ofrece y otros, por ejemplo otro software o hardware. 8, 14, 71
- Azure** Microsoft Azure es un conjunto de servicios en la nube, ofrecido como servicio y alojado en los Data Centers de Microsoft. Azure ofrece desde servicios que alojan aplicaciones en alguno de los centros de procesamiento de datos de Microsoft para que se ejecute sobre su infraestructura hasta servicios del internet de las cosas, comunicación segura o machine learning. 2, 3, 6, 8, 11, 13, 17, 21, 45
- C#** Es un lenguaje de programación orientado a objetos, elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework [18]. 7, 23
- CD** La entrega continua es una práctica de desarrollo de software mediante la cual se preparan automáticamente los cambios en el código y se entregan a la fase de producción. Fundamental para el desarrollo de aplicaciones modernas, la entrega continua amplía la integración continua al implementar todos los cambios en el código en un entorno de pruebas o de producción después de la fase de compilación. Cuando la entrega continua se implementa de manera adecuada, los desarrolladores dispondrán siempre de un artefacto listo para su implementación que se ha sometido a un proceso de pruebas estandarizado[41]. 2, 6, 15, 17, 18, 67
- CI** La integración continua es una practica de desarrollo software que consiste en hacer integraciones y pruebas automáticas de un proyecto lo mas frecuentemente posible y así poder detectar fallos cuanto antes. Entendemos por integración la compilación y ejecución de pruebas de todo un proyecto[16]. 2, 3, 6, 15, 17, 21, 67
- CRM** Customer Relationship Management. Es un sistema software que se utiliza para la gestión de contactos, la gestión de ventas, la productividad y en general la gestión de las relaciones con los clientes de una empresa[38]. 2, 6, 7, 29, 31, 61, 66

**DevOps** DevOps es una metodología de diseño de software basa entre la integración entre los desarrolladores y los administradores de sistemas, este enfoque busca que los desarrolladores puedan centrarse en desarrollar y olvidarse de los despliegues de código. [21][12]. 2, 3, 6

**DLL** Un archivo DLL (Dynamic Link Library) es una biblioteca que contiene código y datos que pueden utilizarse por varios programas al mismo tiempo. Esto ayuda a promover la reutilización de código y uso eficaz de la memoria. Mediante el uso de una DLL, un programa puede ser modularizado en componentes separados. Cada módulo se puede cargar en el programa principal en tiempo de ejecución si está instalado el módulo. Los módulos son independientes y un módulo sólo se carga cuando se solicita esa funcionalidad[39]. 7, 13, 23, 67

**Dynamics 365** Dynamics 365 es un sistema de gestión de relaciones con clientes y planificación de recursos empresariales creado por Microsoft. 2, 3, 6–10, 14, 23, 28, 30, 31, 52, 53, 55

**Git** Git es un Sistema de control de Versiones distribuido de código abierto, libre y gratuito diseñado para gestionar todo tipo de proyectos desde los mas simples hasta los mas complejos con velocidad y eficiencia[34]. 23

**IaaS** Infrastructure as a Service o Infraestructura como un Servicio es una infraestructura de computación instantánea, provisionada y gestionada a través de internet. IaaS evita el gasto inicial de configurar y administrar un centro de datos local, permite escalar con la demanda pagando solo por lo que se usa. El proveedor se encarga de gestionar la infraestructura, mientras que el cliente se encarga de comprar, configurar e instalar el sistema operativo, las herramientas necesarias y las aplicaciones[42]. 6, 7, 9

**IDE** Los Entornos de desarrollo integrados (Integrated Development Environment) son aplicaciones que proporcionan servicios integrales para facilitarle al programador el desarrollo de software. Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código[15]. 15, 23

**JavaScript** JavaScript (comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas<sup>4</sup> aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript)[19] . 7, 23

**LaTeX** Es un lenguaje de marcado especialmente utilizado para documentos científicos y técnicos. 23

**PaaS** Platform as a Service o Plataforma como Servicio es un entorno de desarrollo e implementación completo en la nube, con recursos que permiten entregar todo, desde aplicaciones sencillas basadas en la nube hasta aplicaciones empresariales sofisticadas habilitadas para la nube. Puesto que ofrece infraestructura como servicio, PaaS aporta las mismas ventajas que IaaS (pago por uso, rápida escalabilidad vertical...). Pero las características adicionales, como herramientas de desarrollo y el olvidarse de la provisión, configuración y administración del Sistema Operativo, bases de datos, etc...[43]. 6, 9, 11, 12

**PowerShell** PowerShell es una herramienta de automatización y configuración multiplataforma (Windows, Linux y macOS) que funciona con tus herramientas existentes y esta optimizada para tratar con datos estructurados (JSON, CSV, XML, etc.), APIs REST y modelos de objetos. Incluye un interprete de línea de comandos y un lenguaje de scripting asociado[29]. 23

**SaaS** Software as a Service o Software como un Servicio es un modelo de distribución donde el software se adquiere de un proveedor de servicios en la nube, generalmente mediante un modelo de pago por uso. Los usuarios se acceden a él a través de Internet, generalmente con un explorador web. Toda la infraestructura subyacente, el software y los datos de las aplicaciones se encuentran en el centro de datos del proveedor. El proveedor de servicios administra el hardware y el software y garantizará también la disponibilidad y la seguridad de la aplicación y de sus datos. 6, 7, 9, 19, 23

**SCV** Un sistema de control de versiones guarda los cambios a lo largo del tiempo en un archivo o conjunto de archivos así que se pueden recuperar versiones específicas posteriormente. Por lo general estos se utilizan para manejar las versiones de archivos de código fuente, pero estos pueden ser usados para controlar casi cualquier tipo de fichero informático. Algunos de los sistemas de control de versiones más utilizados son: GIT, Subversion, Team Foundation Version Control... 21, 23, 53, 54

**SLA** Un Acuerdo de nivel de servicio (Service Level Agreement) es un acuerdo firmado con un proveedor de servicios que fija el nivel acordado para la calidad de un servicio. 8

**SQL Server** Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas.[28]. 9, 10

**YAML** YAML Ain't Markup Language. Es un formato de serialización de datos amigable para los humanos, estándar para todos los lenguajes de programación creado en mayo de 2001[27]. 55



# Bibliografía

- [1] *Azure Boards*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/services/devops/boards/> (visitado 23-04-2019).
- [2] *Azure DevOps Features Timeline*. Microsoft Corporation. URL: <https://docs.microsoft.com/en-us/azure/devops/release-notes/> (visitado 10-06-2019).
- [3] *Azure DevOps Hosted Agents Software*. Microsoft Corporation. URL: <https://docs.microsoft.com/es-es/azure/devops/pipelines/agents/hosted?view=azure-devops#software> (visitado 08-05-2019).
- [4] *Azure Pipelines*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/services/devops/pipelines/> (visitado 23-04-2019).
- [5] *Azure Pipelines: Variable Groups*. Microsoft Corporation. URL: <https://docs.microsoft.com/en-us/azure/devops/pipelines/library/variable-groups> (visitado 09-03-2019).
- [6] *Azure Pipelines: Variables*. Microsoft Corporation. URL: <https://docs.microsoft.com/en-us/azure/devops/pipelines/process/variables> (visitado 08-03-2019).
- [7] *Azure Repos*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/services/devops/repos/> (visitado 23-04-2019).
- [8] P. Bourque y R.E. Fairley. *Guide to the Software Engineering Body of Knowledge, Version 3.0*. ACM-IEEE. 2014.
- [9] *Build pipeline triggers*. Microsoft Corporation. Mayo 2019. URL: <https://docs.microsoft.com/en-us/azure/devops/pipelines/build/triggers> (visitado 25-05-2019).
- [10] *Calculadora de Precios Microsoft Azure*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/pricing/calculator/> (visitado 01-05-2019).
- [11] *Conozca Azure*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/overview/> (visitado 17-06-2019).
- [12] José Ruiz Cristina. *Qué es DevOps (y sobre todo qué no es DevOps)*. Paradigma Digital SL. Noviembre 2015. URL: <https://www.paradigmadigital.com/techbiz/que-es-devops-y-sobre-todo-que-no-es-devops/> (visitado 02-05-2019).
- [13] Jim Daly. *Introducción a soluciones*. Microsoft Corporation. Octubre 2017. URL: <https://docs.microsoft.com/es-es/dynamics365/customer-engagement/developer/introduction-solutions> (visitado 25-02-2019).

- [14] *Deployment groups*. Microsoft Corporation. Agosto 2018. URL: <https://docs.microsoft.com/en-us/azure/devops/pipelines/release/deployment-groups/> (visitado 05-06-2019).
- [15] *Entorno de Desarrollo Integrado*. Wikipedia. URL: [https://es.wikipedia.org/wiki/Entorno\\_de\\_desarrollo\\_integrado](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) (visitado 23-04-2019).
- [16] Martin Fowler. *Continuous Integration*. Mayo 2006. URL: <https://www.martinfowler.com/articles/continuousIntegration.html> (visitado 22-04-2019).
- [17] Patrick Debois Jez Humble Gene Kim John Willis. *The DevOPS Handbook*. IT Revolution Press, Diciembre 2016.
- [18] *Introducción al lenguaje C# y .NET Framework*. Microsoft Corporation. Julio 2015. URL: <https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework> (visitado 23-04-2019).
- [19] *JavaScript*. Wikipedia. URL: <https://es.wikipedia.org/wiki/JavaScript> (visitado 23-04-2019).
- [20] *JSON*. Wikipedia. URL: <https://es.wikipedia.org/wiki/JSON> (visitado 12-06-2019).
- [21] Mike Loukides. *What is DevOps?* O'Reilly Radar. Junio 2015. URL: <http://radar.oreilly.com/2012/06/what-is-devops.html> (visitado 20-02-2019).
- [22] Vijay Machiraju. *Customize your pipeline*. Microsoft Corporation. Abril 2019. URL: <https://docs.microsoft.com/en-us/azure/devops/pipelines/customize-pipeline> (visitado 09-05-2019).
- [23] Andy Schultz y Marc Schweigert. *Microsoft Dynamics CRM Online patterns & principles for solution builders*. Microsoft Corporation. 2015.
- [24] Sean McNellis y Kenichiro Nakamura. *Microsoft.Xrm.Data.PowerShell*. Microsoft Corporation. Agosto 2018. URL: <https://github.com/seanmcne/Microsoft.Xrm.Data.PowerShell/> (visitado 01-05-2019).
- [25] *Microsoft-hosted agents*. Microsoft Corporation. Marzo 2019. URL: <https://docs.microsoft.com/es-es/azure/devops/pipelines/agents/hosted> (visitado 07-05-2019).
- [26] Ángel Nájera Pérez. *Gestión de riesgos en Scrum*. Wolf Project. Octubre 2018. URL: <https://wolfproject.es/scrum-riesgos-gestion/> (visitado 23-05-2019).
- [27] Ingy döt Net Oren Ben-Kiki Clark Evans. *YAML Ain't Markup Language (YAML™) Version 1.2*. YAML.org. Octubre 2009. URL: <https://yaml.org/spec/1.2/spec.html> (visitado 14-05-2019).
- [28] *Plataforma de Datos de Microsoft*. Microsoft Corporation. URL: <https://www.microsoft.com/es-es/sql-server/> (visitado 25-04-2019).
- [29] *PowerShell Core README*. Microsoft Corporation. URL: <https://github.com/PowerShell/PowerShell/blob/master/README.md> (visitado 23-04-2019).

- 
- [30] *Precios de Azure DevOps*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/pricing/details/devops/azure-devops-services/> (visitado 01-05-2019).
- [31] *Precios de Dynamics 365*. Microsoft Corporation. URL: <https://dynamics.microsoft.com/es-es/pricing/> (visitado 25-04-2019).
- [32] R. Pressman. *Ingeniería del Software*. McGraw-Hill, 2014.
- [33] *Requisitos de hardware y software para instalar SQL Server*. Microsoft Corporation. Noviembre 2018. URL: <https://docs.microsoft.com/es-es/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server?view=sql-server-2017> (visitado 25-04-2019).
- [34] Ben Straub Scott Chacon. *Pro Git*. Apress, Noviembre 2014.
- [35] *Task groups for builds and releases*. Microsoft Corporation. Febrero 2019. URL: <https://docs.microsoft.com/en-us/azure/devops/pipelines/library/task-groups> (visitado 01-06-2019).
- [36] *Use the SolutionPackager tool to compress and extract a solution file*. Microsoft Corporation. Marzo 2019. URL: <https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/compress-extract-solution-file-solutionpackager> (visitado 15-05-2019).
- [37] Kumar Vivek. *Uso de cmdlets de PowerShell para útiles de XRM para conectarse a Dynamics 365 for Customer Engagement*. Microsoft Corporation. Mayo 2018. URL: <https://docs.microsoft.com/es-es/dynamics365/customer-engagement/developer/xrm-tooling/use-powershell-cmdlets-xrm-tooling-connect> (visitado 20-03-2019).
- [38] *¿Que es la CRM?* Salesforce. URL: <https://www.salesforce.com/es/learning-centre/crm/what-is-crm/#crm> (visitado 25-02-2019).
- [39] *¿Que es una dll?* Microsoft Corporation. URL: <https://support.microsoft.com/es-es/help/815065/what-is-a-dll> (visitado 23-04-2019).
- [40] *¿Qué es Common Data Service?* Microsoft Corporation. Mayo 2018. URL: <https://docs.microsoft.com/es-es/powerapps/maker/common-data-service/data-platform-intro> (visitado 22-05-2019).
- [41] *¿Qué es la Entrega Continua?* Amazon Web Services. URL: <https://aws.amazon.com/es/devops/continuous-delivery/> (visitado 22-04-2019).
- [42] *¿Qué es la IaaS? Infraestructura como servicio*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/overview/what-is-iaas/> (visitado 25-04-2019).
- [43] *¿Qué es PaaS? Plataforma como servicio*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/overview/what-is-paas/> (visitado 01-05-2019).
- [44] *¿Qué es SaaS? Software como servicio*. Microsoft Corporation. URL: <https://azure.microsoft.com/es-es/overview/what-is-saas/> (visitado 13-05-2019).





# Apéndice A

## Manual de Instalación

El siguiente manual describe los procedimientos necesarios para la instalación del sistema de Control de cambios en soluciones, restauración de las mismas y la herramienta de despliegue asociada. Este manual es una guía práctica para la configuración del proyecto, creación de repositorios, importación de los “Task Group” y la configuración de las diferentes Pipelines dentro de un proyecto de Azure DevOps.

Las pautas e indicaciones de este manual están referidas a la versión de Azure DevOps liberada el 10 de Junio de 2019, también conocida como Sprint 153, sucesivas actualizaciones podrían hacer que algunos de estos procedimientos difiera. Para mas información recomendamos consultar la Documentación de Azure DevOps donde vienen indicados los cambios realizados en cada actualización. « <https://docs.microsoft.com/es-es/azure/devops/release-notes> »

### A.1. Requisitos Previos

Para la creación de un proyecto en Azure DevOps es necesario una cuenta de Microsoft o GitHub, en este manual no se indican los pasos para la creación de la misma. Para el inicio de sesión en Azure DevOps, la creación de la cuenta o obtener mas información visite: « <https://azure.microsoft.com/es-es/services/devops/> ».

Para la configuración de la conexión al entorno Dynamics CRM es necesario disponer de datos de acceso con los permisos necesarios para realizar la exportación e importación de soluciones. Este manual no trata la instalación o aprovisionamiento de Dynamics CRM ni su configuración de usuario, para mas información puede dirigirse a la documentación sobre Dynamics CRM de Microsoft o contactar con su proveedor del servicio.

Una vez se dispone de la cuenta en Azure DevOps, y los datos de conexión al CRM se puede continuar con la antelación y configuración.

## A.2. Configuración Azure DevOps y Repositorios

Si es la primera vez que utiliza una cuenta de Azure DevOps se le solicitara que cree un nuevo proyecto, de no ser así podrá elegir entre crear un nuevo proyecto o utilizar uno existente.

## A.3. Repositorios Azure Repos

Si desea alojar los cambios que se descargan del CRM en los repositorios que ofrece Azure DevOps, y no dispone de ninguno creado anteriormente seleccione en el menú lateral la opción *Repos* con el icono indicado en la imagen A.1. Y un asistente le guiara en su creación, basta indicar el nombre del repositorio e **indicar que se desea un repositorio de tipo GIT**. Esta ultima indicación es importante ya que las “Task Group” que se instalaran a continuación están desarrolladas para funcionar con repositorios GIT, pese a ser posible que trabajen con repositorios TFVC seria necesaria una adaptación del código.

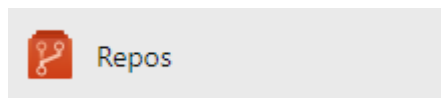


Figura A.1: Icono Azure Repos

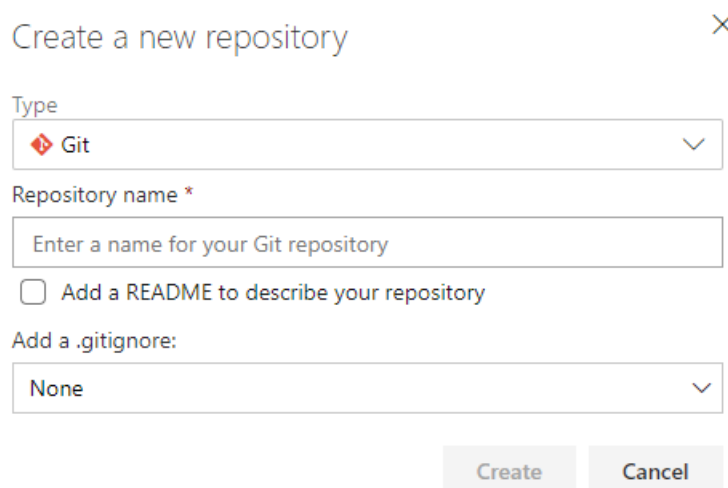
The image shows a "Create a new repository" dialog box. At the top right is a close button (X). Below the title, there is a "Type" dropdown menu with "Git" selected. Below that is a "Repository name" text input field with a red asterisk indicating it is required, containing the placeholder text "Enter a name for your Git repository". Below the name field is a checkbox labeled "Add a README to describe your repository" which is currently unchecked. Below the checkbox is an "Add a .gitignore:" section with a dropdown menu set to "None". At the bottom right are two buttons: "Create" and "Cancel".

Figura A.2: Datos solicitados para creación nuevo repositorio

La importación de los “Task Groups” puede realizarse sin necesidad de un repositorio pero son necesarios para el funcionamiento y creación de las Pipelines.

## A.4. Importar “Task Groups”

Para realizar la importación de los “Task Groups” ha de disponer de los archivos JSON de estas:

**TG01.JSON** Este archivo contiene el “Task Group” relativo a la funcionalidad que permite obtener los cambios de las soluciones en el CRM y registrarlos en el repositorio

**TG01.JSON** Este archivo contiene el “Task Group” relativo a la funcionalidad que permite recuperar los cambios insertados en el repositorio y devolverlos al CRM

Para su importación haga clic en la sección *Pipelines* en el menú lateral, tras lo cual seleccione *Task groups*. Una vez en la pantalla de Task Groups haga clic en el botón *Import*. Aparecerá una ventana sobre la cual podrá arrastrar o buscar en el equipo los archivos JSON a importar. Tras seleccionarlos aparecerá la lista de tareas a ejecutar y su configuración por si fuera necesario realizar algún cambio en el código. *No se recomienda que se modifiquen estas secciones sin tener un claro conocimiento de lo que esta modificando*. Para finalizar pulse sobre el botón *Save*, para guardar el Task Group importado. Realice de nuevo la misma operación con el resto de archivos JSON.

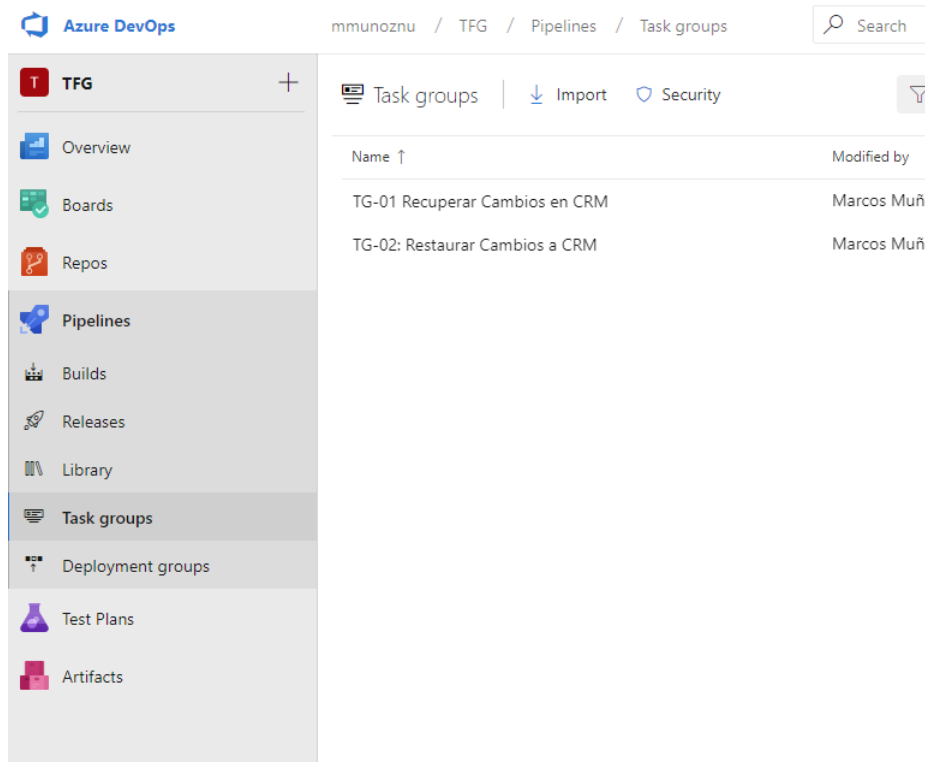


Figura A.3: Azure Pipelines: Seccion Task Groups

## A.5. Configuración Pipelines

Para configurar las Pipelines seleccione *Pipelines* en el menú lateral, allí haga clic en *Builds*. A continuación se mostrara la lista de las Pipelines ejecutadas recientemente. Haga clic en *New*.

### A.5.1. Pipeline para Recuperar Cambios del CRM

Un asistente le guiara por la configuración de la pipeline, en esta primera etapa ha de seleccionar el repositorio en el que se insertaran o recuperaran los cambios. Seleccione el repositorio que desea usar en esta pipeline, este sera en el cual se inserten los cambios recogidos del CRM. Tras seleccionar el repositorio se mostraran diferentes plantillas para diferentes tipos de proyecto, se ha de seleccionar *Empty job* para crear una pipeline vacía sobre la que añadir el Task Group previamente importado.

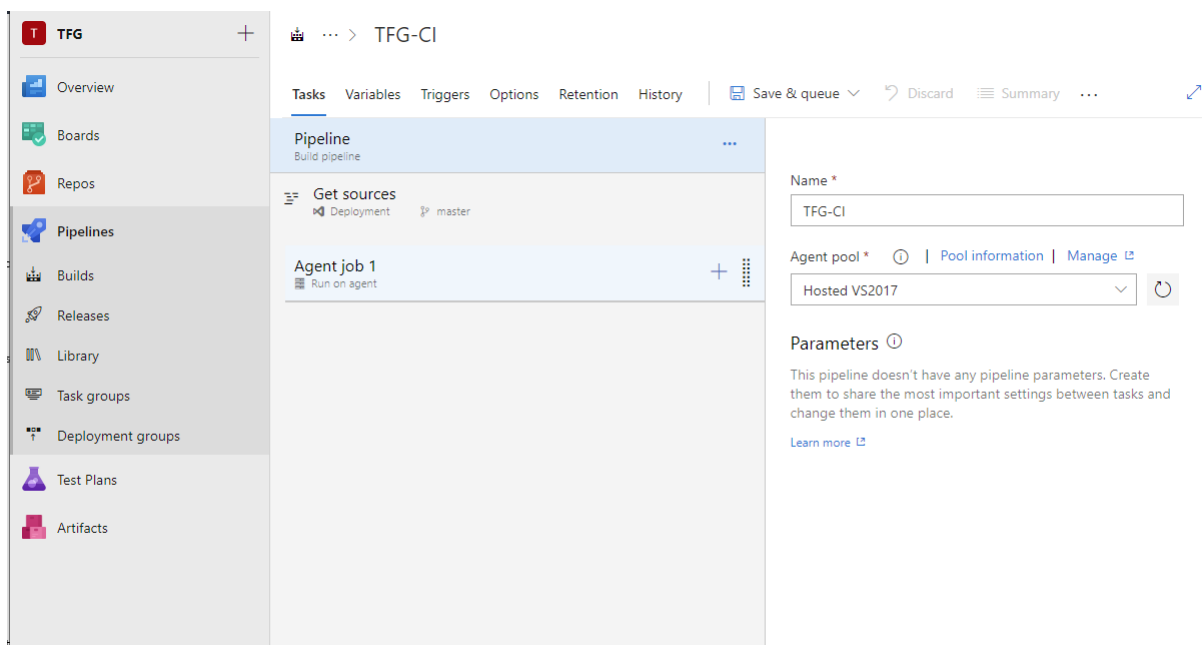


Figura A.4: Pipeline en Blanco

En la primera pantalla se puede definir el nombre de la Pipeline y el tipo de agente a utilizar, en este caso se seleccionara Hosted. Tras esto se hará clic en el símbolo + que Aparece junto *Agent Job 1* esto permitirá añadir una nueva tarea. Al hacer clic se muestra una lista de tareas junto un cuadro de búsqueda. En este se introducirá TG-01 y se presionara el botón *Add* para añadir el Task Group a la pipeline.

Una vez añadido a la Pipeline pasamos a configurar la variables de conexión al CRM, estos datos pueden insertarse directamente sobre los campos de el Task Group, pero lo recomendable es que se usen grupos de variables. Para ello se hará clic en la pestaña variables, donde se podrán definir las variables que utiliza la pipeline así como vincular o crear nuevos grupos de variables. Para esta pipeline crearemos 2 variables, una para contener el nombre de la solución a exportar,

que puede ser siempre la misma o definirse en el lanzamiento de la pipeline. Y otra para el mensaje del commit, en esta también podremos predefinir un mensaje pero se recomienda marcar la casilla *Settable at queue time* para que pueda ser modificada en el momento de ejecutar la pipeline.

Las variables para los datos de conexión del CRM también se pueden definir aquí, pero si estos se van a reutilizar en varias Pipelines es más conveniente crear un *Variable Group*. Al hacer clic en *Variable Group* se muestra la opción *Manage variable groups*, al seleccionarla se nos permitirá crear un nuevo grupo de variables. Se recomienda definir en este las variables de la conexión que necesita el Task Group, que son:

- AuthType
- Url
- Username
- Password

Para la variable Password que almacenara la contraseña se debe activar el candado que aparece junto a ella, para que sea almacenada y tratada de forma segura y no visible para los usuarios. Al volver a la Pipeline se deberá introducir estas variables recién creadas en los parámetros del Task Group, para ello se utilizara la sintaxis  $\$(Nombre)$ , donde se sustituirá *Nombre* por el nombre de la variable en cuestión, como por ejemplo  $\$(Url)$ .

Una vez indicados todos los parámetros de conexión del CRM se hará clic en el botón *Save* que permite guardar los cambios realizados en la pipeline. Si se desea programar la ejecución automática de la Pipeline deberá acceder a la pestaña Triggers donde podrá añadir diferentes horas y días de ejecución. En este punto la pipeline ya esta lista para su uso por parte de los usuarios que solo deberán añadirla a la cola de trabajos.

### A.5.2. Pipeline para restaurar los Cambios al CRM

Esta Pipelines se configurara siguiendo los mismos pasos que la pipeline anterior. Solo que se debera buscar y añadir el Task group: TG-02 Restaurar Cambios a CRM. Al igual que en la pipeline anterior se definirán los parámetros de conexión al CRM, y la variable que permite indicar el nombre de la solución a volcar al CRM. En este caso no hay un mensaje de commit ya que no se realizan inserciones en el repositorio.

### A.5.3. Pipeline para restaurar los Cambios al CRM

Al igual que en los casos anteriores se deberá crear una pipeline vacía en la que se deberá añadir tanto el TG-01 como el TG-02, en ese orden. A la hora de configurar los parámetros ha de tenerse en cuenta que el TG-01 y el TG-02 utilizaran datos de conexión diferentes ya que uno recogerá la solución a desplegar de un entorno CRM y el siguiente la desplegara en un entorno diferente. Teniendo en cuenta esta diferencia en los parámetros a introducir la configuración se realiza de forma idéntica a las anteriores.



Figura A.5: Añadir TG-01 Recuperar Cambios en CRM

## A.6. Acceso y permisos

Tras configurar las Pipelines ya pueden ser lanzadas por los usuarios, o siguiendo su programación horaria. Tras esto puede añadir los usuarios que desee al proyecto usando el botón Invite que aparece en la parte superior de la pagina del proyecto en Azure DevOps. Tras añadir usuarios se recomienda que revise los permisos que se concede a cada uno, limitando el acceso de edición a las Pipelines. La configuración de usuarios y permisos no es un tema que trate en este manual por lo que se insta a la consulta de la documentación de Azure DevOps referente a la administración de usuarios. « <https://docs.microsoft.com/es-es/azure/devops/user-guide/project-admin-tutorial> »

# Apéndice B

## Manual de Usuario

Este manual pretende ser una introducción básica al uso de Pipelines, en este documento no se trata la creación o modificación de estas, por lo que debe verse como una guía rápida para el uso de Pipelines que han sido previamente configuradas por el administrador del proyecto.

### B.1. Requisitos Previos

Para el uso del sistema de Control de cambios en soluciones, restauración de las mismas y la herramientas de despliegue asociadas este debe haber sido configurado con anterioridad por un administrador. Además el usuario debe tener acceso al proyecto. En caso de no ser así contacte con el administrador del mismo.

### B.2. Recuperar cambios de CRM

Para recuperar los cambios del CRM e insertarlos en el repositorio primero se deberá acceder al proyecto de Azure DevOps indicado a través de la URL del mismo, o iniciando sesión en <https://dev.azure.com/> y seleccionándolo de la lista de proyectos disponibles. Una vez se tiene acceso al proyecto se han de seguir los siguientes pasos:

1. Seleccionar Pipelines del menú lateral izquierdo
2. Se mostraran las Pipelines a las que se tiene acceso
3. Seleccionar la Pipeline que se desea ejecutar, en este caso “Recuperar Cambios en CRM”
4. Seleccionar el botón “Queue”
5. Se mostrara la pantalla de encolado de la Pipeline donde se indicara el mensaje que se desea en el commit y la solución que se extraerá del CRM. En caso de no indicarse tomaran los valores por defecto que se muestran en sus cuadros de texto.

6. Al pulsar sobre el botón *Queue* la Pipeline se encolara y se ejecutara cuando se le pueda asignar un agente. La Pipeline aparece en el listado de la derecha donde se puede ver su estado.
7. Haciendo clic sobre el listado de la derecha se pueden ver los detalles de la ejecución así como la salida por consola del Agente.
8. Tras ejecutarse la pipeline enviara un mensaje al usuario. En caso de cancelarse o finalizar de forma errónea el usuario sera notificado.

### B.3. Restaurar Cambios a CRM

Para restaurar los cambios del GIT al CRM se han de seguir los siguientes pasos:

1. Seleccionar Pipelines del menú lateral izquierdo
2. Se mostraran las Pipelines a las que se tiene acceso
3. Seleccionar la Pipeline que se desea ejecutar, en este caso “Restaurar Cambios a CRM”
4. Seleccionar el botón “Queue”
5. Se mostrara la pantalla de encolado de la Pipeline donde se indicara la solución que se desea importar al CRM. En caso de no indicarse tomaran los valores por defecto que se muestran en sus cuadros de texto.
6. También puede indicarse el commit concreto que se desea importar al CRM, para ello se ha de indicar el Commit ID en el campo commit, de no indicarse ninguno se utilizara el ultimo realizado. El commit id puede ser obtenido desde el repositorio al visualizar un commit concreto.
7. Al pulsar sobre el botón *Queue* la Pipeline se encolara y se ejecutara cuando se le pueda asignar un agente. La Pipeline aparece en el listado de la derecha donde se puede ver su estado.
8. Haciendo clic sobre el listado de la derecha se pueden ver los detalles de la ejecución así como la salida por consola del Agente.
9. Tras ejecutarse la pipeline enviara un mensaje al usuario. En caso de cancelarse o finalizar de forma errónea el usuario sera notificado.



## B.4. Desplegar Cambios a Otro Entorno

En este caso la pipeline se ejecuta prácticamente igual que la de “Recuperar Cambios en CRM”, hay que tener en cuenta que los entornos han sido definidos por el administrador en la configuración de la pipeline y estos no se pueden variar. Siempre va de un entorno de Origen a uno de Destino, si desea modificarlo contacte con el administrador del proyecto. Para Desplegar Cambios a Otro Entorno se han de seguir los siguientes pasos:

1. Seleccionar Pipelines del menú lateral izquierdo
2. Se mostraran las Pipelines a las que se tiene acceso
3. Seleccionar la Pipeline que se desea ejecutar, en este caso “Desplegar Cambios a Otro Entorno”
4. Seleccionar el botón “Queue”
5. Se mostrara la pantalla de encolado de la Pipeline donde se indicara el mensaje que se desea en el commit y la solución que se desplegara al CRM. En caso de no indicarse tomaran los valores por defecto que se muestran en sus cuadros de texto.
6. Al pulsar sobre el botón *Queue* la Pipeline se encolara y se ejecutara cuando se le pueda asignar un agente. La Pipeline aparece en el listado de la derecha donde se puede ver su estado.
7. Haciendo clic sobre el listado de la derecha se pueden ver los detalles de la ejecución así como la salida por consola del Agente.
8. Tras ejecutarse la pipeline enviara un mensaje al usuario. En caso de cancelarse o finalizar de forma errónea el usuario sera notificado.

### Queue build for CU-02 Restaurar Cambios a CRM ✕

Agent pool

Hosted VS2017 ▾

Branch

🔒 master ▾

Commit

**Variables** Demands

Solucion Predeterminada

[+ Add](#)

Queue Cancel

Figura B.1: Añadir TG-01 Recuperar Cambios en CRM

# Apéndice C

## Código: Pruebas de Concepto PowerShell

---

```
1  #Variables Configuracion
2  $solucion ="default"
3  $CrmUser = "User"
4  $CrmPass = "Passwd"
5  $CrmUrl = "http://crmurl"
6  $CrmOrg = "OrganizacionOnPremise"
7
8
9  #Función que comprueba que las herramientas de consola CRM estén instaladas
10 #realiza la instalación de las mismas si fuera necesario
11 function InstallXRMPS{
12     Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass -Force
13     $moduleName = "Microsoft.Xrm.Data.Powershell"
14     $moduleVersion = "2.7.2"
15     if (!(Get-Module -ListAvailable -Name $moduleName )) {
16         Write-host "XRM Powershell No encontrado, instalando"
17         $moduleVersion
18         Install-Module -Name $moduleName -MinimumVersion $moduleVersion
19         ↵ -Force
20     }
21     else
22     {
23         Write-host "El modulo yaa esta instalado"
24     }
25 }
26 #Esta función devuelve el objeto de Conexión a CRM necesario para operar con los
27 ↵ distintos comandos
28 function GetCrmConn{
```

```

27     param(
28         [string]$user,
29         [string]$secpasswd,
30         [string]$crmUrl,
31         [string]$organization
32     )
33     Write-Host "Usuario: $user Password: $secpasswd CrmUrl: $crmUrl"
34     $secpasswd2 = ConvertTo-SecureString -String $secpasswd -AsPlainText
35     ↪ -Force
36     write-host "Creando Credenciales"
37     $mycreds = New-Object System.Management.Automation.PSCredential ($User,
38     ↪ $secpasswd2)
39
40     write-host "Conectando A CRM"
41     #Comando Conexion para CRM Online
42     #$crm = Connect-CrmOnline -Credential $mycreds -ServerUrl $CrmUrl
43     #Comando Conexion para CRM OnPremise
44     $crm = Get-CrmConnection -ServerUrl $CrmUrl -Credential $mycreds
45     ↪ -OrganizationName $organization
46     write-host "Conexion Establecida"
47     return $crm
48 }
49
50 #Comprobar/Instalar Módulos
51 InstallXRMPS
52 #Datos Conexion Crm
53 $CrmConn = GetCrmConn -user $CrmUser -secpasswd $CrmPass -crmUrl $CrmUrl
54 ↪ -organization $CrmOrg
55 #Configurar TimeOut Conexion (1000s ~ 16 min)
56 Set-CrmConnectionTimeout -conn $CrmConn -TimeoutInSeconds 1000
57
58 #Prueba de concepto 1 Exportar solución de CRM
59 Write-Host "Exportando Solucion"
60 Export-CrmSolution -conn $CrmConn -SolutionName "$solucion" -SolutionZipFileName
61 ↪ "$solucion.zip"
62
63 #Prueba de concepto 4 Subir solución a CRM
64 Write-host "Importando Solucion"
65 Import-CrmSolution -conn $Crm -SolutionFilePath "$solucion.zip"

```

---

# Apéndice D

## Código: JSON Task Group TG-01

---

```
1 {
2   "tasks":[
3     {
4       "environment":{},
5       "displayName":"Instalar Herramientas CRM",
6       "alwaysRun":false,
7       "continueOnError":false,
8       "condition":"succeeded()",
9       "enabled":true,
10      "timeoutInMinutes":0,
11      "inputs":{},
12      "task":{
13        "id":"04ad1c72-5e49-4686-8a3a-dda6948b0fcd",
14        "versionSpec":"9.*",
15        "definitionType":"task"
16      }
17    },
18    {
19      "environment":{},
20      "displayName":"Publicar Personalizaciones CRM",
21      "alwaysRun":false,
22      "continueOnError":false,
23      "condition":"succeeded()",
24      "enabled":true,
25      "timeoutInMinutes":0,
26      "inputs":{
27        "crmConnectionString":"AuthType=$(AuthType);Username=$(Username);
    ↪ Password=$(Password);Url=$(Url)",
```

```

28         "crmConnectionTimeout": "120"
29     },
30     "task": {
31         "id": "f64d3db3-f19a-41dd-bdb8-f8f0e60a4bb1",
32         "versionSpec": "10.*",
33         "definitionType": "task"
34     }
35 },
36 {
37     "environment": {},
38     "displayName": "Exportar Solucion CRM: $(Solucion)",
39     "alwaysRun": false,
40     "continueOnError": false,
41     "condition": "succeeded()",
42     "enabled": true,
43     "timeoutInMinutes": 0,
44     "inputs": {
45         "crmConnectionString": "AuthType=$(AuthType);Username=$(Username);
46         ↪ Password=$(Password);Url=$(Url)",
47         "solutionName": "$(Solucion)",
48         "exportManaged": "false",
49         "exportUnmanaged": "true",
50         "targetVersion": "",
51         "updateVersion": "false",
52         "includeVersionInSolutionFile": "false",
53         "outputPath": "$(build.binariesdirectory)",
54         "crmConnectionTimeout": "120",
55         "exportAutoNumberingSettings": "false",
56         "exportCalendarSettings": "false",
57         "exportCustomizationSettings": "false",
58         "exportEmailTrackingSettings": "false",
59         "exportExternalApplications": "false",
60         "exportGeneralSettings": "true",
61         "exportIsvConfig": "false",
62         "exportMarketingSettings": "false",
63         "exportOutlookSynchronizationSettings": "false",
64         "exportRelationshipRoles": "false",
65         "exportSales": "false"
66     },
67     "task": {

```

```

67         "id":"015e2e58-9c77-4cbc-a9c1-a28158165365",
68         "versionSpec":"10.*",
69         "definitionType":"task"
70     }
71 },
72 {
73     "environment":{},
74     "displayName":"Config Git",
75     "alwaysRun":false,
76     "continueOnError":false,
77     "condition":"succeeded()",
78     "enabled":true,
79     "timeoutInMinutes":0,
80     "inputs":{
81         "targetType":"inline",
82         "filePath":"",
83         "arguments":"",
84         "script":"git config --global user.email \"tg01@devops.com\"\\ngit
85         ↪ config --global user.name \"TG-01 Azure Pipelines\"\\ngit
86         ↪ checkout master\\ngit status",
87         "errorActionPreference":"stop",
88         "failOnStderr":"false",
89         "ignoreLASTEXITCODE":"false",
90         "pwsh":"false",
91         "workingDirectory":""
92     },
93     "task":{
94         "id":"e213ff0f-5d5c-4791-802d-52ea3e7be1f1",
95         "versionSpec":"2.*",
96         "definitionType":"task"
97     }
98 },
99 {
100     "environment":{},
101     "displayName":"Extraer Solución ",
102     "alwaysRun":false,
103     "continueOnError":false,
104     "condition":"succeeded()",
105     "enabled":true,
106     "timeoutInMinutes":0,

```

```

105     "inputs":{
106         "crmSdkVersion":"9.0.0",
107         "unpackedFilesFolder":"$(Solucion)",
108         "mappingFile":"",
109         "packageType":"Unmanaged",
110         "solutionFile":"$(build.binariesdirectory)\$(Solucion).zip",
111         "treatUnpackWarningsAsErrors":"false"
112     },
113     "task":{
114         "id":"48834e5a-a932-49af-a7fd-a805b5e1cfb5",
115         "versionSpec":"10.*",
116         "definitionType":"task"
117     }
118 },
119 {
120     "environment":{},
121     "displayName":"Git Commit, Git Push",
122     "alwaysRun":false,
123     "continueOnError":false,
124     "condition":"succeeded()",
125     "enabled":true,
126     "timeoutInMinutes":0,
127     "inputs":{
128         "targetType":"inline",
129         "filePath":"",
130         "arguments":"",
131         "script":"\ngit add *\ngit commit -a -m\"[AUTO] Build
↪ $(Build.BuildNumber) ($(Solucion)) $(CoomitMsg)\"\ngit
↪ status\ngit push origin",
132         "errorActionPreference":"stop",
133         "failOnStderr":"false",
134         "ignoreLASTEXITCODE":"false",
135         "pwsh":"false",
136         "workingDirectory":""
137     },
138     "task":{
139         "id":"e213ff0f-5d5c-4791-802d-52ea3e7be1f1",
140         "versionSpec":"2.*",
141         "definitionType":"task"
142     }

```



---

```

143     }
144 ],
145 "runsOn": [
146     "Agent",
147     "DeploymentGroup"
148 ],
149 "revision": 6,
150 "createdBy": {
151     "displayName": "Marcos Muñoz Nuñez",
152     "id": "f15db556-4bfd-646c-8dcf-486e3329bdd1",
153     "uniqueName": "mmunoznu@everis.com"
154 },
155 "createdOn": "2019-06-11T20:08:30.010Z",
156 "modifiedBy": {
157     "displayName": "Marcos Muñoz Nuñez",
158     "id": "f15db556-4bfd-646c-8dcf-486e3329bdd1",
159     "uniqueName": "mmunoznu@everis.com"
160 },
161 "modifiedOn": "2019-06-13T23:47:26.333Z",
162 "comment": "Push!",
163 "id": "fe094f81-1e3d-4eac-bfdf-d75a9eca6ff1",
164 "name": "GetSoluciones - Carpetas",
165 "version": {
166     "major": 1,
167     "minor": 0,
168     "patch": 0,
169     "isTest": false
170 },
171
172 ↪ "iconUrl": "https://cdn.vsassets.io/v/M152_20190610.2/_content/icon-meta-task.png",
173 "friendlyName": "GetSoluciones - Carpetas",
174 "description": "",
175 "category": "Build",
176 "definitionType": "metaTask",
177 "author": "Marcos Muñoz Nuñez",
178 "demands": [],
179 "groups": [],
180 "inputs": [
181     {
182         "aliases": [],

```

```

182     "options": {},
183     "properties": {},
184     "name": "Authtype",
185     "label": "Authtype",
186     "defaultValue": "",
187     "required": true,
188     "type": "string",
189     "helpMarkDown": "",
190     "groupName": ""
191 },
192 {
193     "aliases": [],
194     "options": {},
195     "properties": {},
196     "name": "CoomitMsg",
197     "label": "CoomitMsg",
198     "defaultValue": "TG-GSC",
199     "required": true,
200     "type": "string",
201     "helpMarkDown": "Commit a introducir en el Repositorio (para anotaciones
    ↪ manuales)",
202     "groupName": ""
203 },
204 {
205     "aliases": [],
206     "options": {},
207     "properties": {},
208     "name": "Password",
209     "label": "Password",
210     "defaultValue": "",
211     "required": true,
212     "type": "string",
213     "helpMarkDown": "",
214     "groupName": ""
215 },
216 {
217     "aliases": [],
218     "options": {},
219     "properties": {},
220     "name": "Solucion",

```

---

```

221     "label": "Solucion",
222     "defaultValue": "Default",
223     "required": true,
224     "type": "string",
225     "helpMarkdown": "Solucion a Extraer",
226     "groupName": ""
227 },
228 {
229     "aliases": [],
230     "options": {},
231     "properties": {},
232     "name": "Url",
233     "label": "Url",
234     "defaultValue": "",
235     "required": true,
236     "type": "string",
237     "helpMarkdown": "",
238     "groupName": ""
239 },
240 {
241     "aliases": [],
242     "options": {},
243     "properties": {},
244     "name": "Username",
245     "label": "Username",
246     "defaultValue": "",
247     "required": true,
248     "type": "string",
249     "helpMarkdown": "",
250     "groupName": ""
251 }
252 ],
253 "satisfies": [],
254 "sourceDefinitions": [],
255 "dataSourceBindings": [],
256 "instanceNameFormat": "Task group: GetSoluciones - Carpetas $(AuthType)",
257 "preJobExecution": {},
258 "execution": {},
259 "postJobExecution": {}
260 }

```



# Apéndice E

## Código: JSON Task Group TG-02

---

```
1 {
2   "tasks": [
3     {
4       "environment": {},
5       "displayName": "Instalar Herramientas CRM",
6       "alwaysRun": false,
7       "continueOnError": false,
8       "condition": "succeeded()",
9       "enabled": true,
10      "timeoutInMinutes": 0,
11      "inputs": {},
12      "task": {
13        "id": "04ad1c72-5e49-4686-8a3a-dda6948b0fcd",
14        "versionSpec": "9.*",
15        "definitionType": "task"
16      }
17    },
18    {
19      "environment": {},
20      "displayName": "Configurar Numero Versión ",
21      "alwaysRun": false,
22      "continueOnError": false,
23      "condition": "succeeded()",
24      "enabled": true,
25      "timeoutInMinutes": 0,
26      "inputs": {
27        "target": "xml",
28        "crmConnectionString": ""
```

```

29     "solutionName": "",
30     "unpackedFilesFolder": "$(Solucion)",
31     "versionNumber": "1.0.0.$(Build.BuildNumber)"
32 },
33 "task": {
34     "id": "1cacdeec-c8dd-4091-a522-5a8fbf49c851",
35     "versionSpec": "10.*",
36     "definitionType": "task"
37 }
38 },
39 {
40     "environment": {},
41     "displayName": "Empaquetar Solucion $(Solucion) ",
42     "alwaysRun": false,
43     "continueOnError": false,
44     "condition": "succeeded()",
45     "enabled": true,
46     "timeoutInMinutes": 0,
47     "inputs": {
48         "crmSdkVersion": "9.0.0",
49         "unpackedFilesFolder": "$(Solucion)",
50         "mappingFile": "",
51         "packageType": "Unmanaged",
52         "updateVersion": "false",
53         "includeVersionInSolutionFile": "false",
54         "outputPath": "$(build.binariesdirectory)",
55         "treatPackWarningsAsErrors": "false"
56     },
57     "task": {
58         "id": "ebec2a90-ce1f-11e6-ae21-c1fb031659ee",
59         "versionSpec": "10.*",
60         "definitionType": "task"
61     }
62 },
63 {
64     "environment": {},
65     "displayName": "Importar Solucion $(Solucion) al CRM",
66     "alwaysRun": false,
67     "continueOnError": false,
68     "condition": "succeeded()",

```

---

```

69     "enabled":true,
70     "timeoutInMinutes":0,
71     "inputs":{
72         "crmConnectionString":"AuthType=$(AuthType);Username=$(Username);
73         ↪ Password=$(Password);Url=$(Url)",
74         "solutionFile":"$(build.binariesdirectory)/$(Solucion).zip",
75         "publishWorkflows":"true",
76         "overwriteUnmanagedCustomizations":"false",
77         "skipProductUpdateDependencies":"false",
78         "convertToManaged":"false",
79         "holdingSolution":"false",
80         "override":"true",
81         "useAsyncMode":"true",
82         "asyncWaitTimeout":"900",
83         "logsDirectory":"","
84         "logFileName":"","
85         "crmConnectionTimeout":"120"
86     },
87     "task":{
88         "id":"4455576d-d40a-4234-ad75-3d7ff40ec76e",
89         "versionSpec":"11.*",
90         "definitionType":"task"
91     }
92 ],
93 "runsOn":[
94     "Agent",
95     "DeploymentGroup"
96 ],
97 "revision":2,
98 "createdBy":{
99     "displayName":"Marcos Muñoz Nuñez",
100    "id":"f15db556-4bfd-646c-8dcf-486e3329bdd1",
101    "uniqueName":"mmunoznu@everis.com"
102 },
103 "createdOn":"2019-06-16T23:16:41.423Z",
104 "modifiedBy":{
105     "displayName":"Marcos Muñoz Nuñez",
106     "id":"f15db556-4bfd-646c-8dcf-486e3329bdd1",
107     "uniqueName":"mmunoznu@everis.com"

```

```

108     },
109     "modifiedOn": "2019-06-16T23:42:25.480Z",
110     "comment": "Init Save",
111     "id": "68b1580a-8187-4041-8db2-a5c6f01ca117",
112     "name": "TG-02: Restaurar Cambios a CRM",
113     "version": {
114         "major": 1,
115         "minor": 0,
116         "patch": 0,
117         "isTest": false
118     },
119
120     ↪ "iconUrl": "https://cdn.vsassets.io/v/M153_20190612.59/_content/icon-meta-task.png",
121     "friendlyName": "TG-02: Restaurar Cambios a CRM",
122     "description": "Este Task Group encapsula las diferentes tareas necesarias para
123     ↪ Impotar una solucion desde el control de versiones al sistema Dynamics
124     ↪ CRM",
125     "category": "Build",
126     "definitionType": "metaTask",
127     "author": "Marcos Muñoz Nuñez",
128     "demands": [],
129     "groups": [],
130     "inputs": [
131         {
132             "aliases": [],
133             "options": {},
134             "properties": {},
135             "name": "AuthType",
136             "label": "AuthType",
137             "defaultValue": "",
138             "required": true,
139             "type": "string",
140             "helpMarkdown": "Tipo de Autenticacion: Office365,AD,...",
141             "groupName": ""
142         },
143         {
144             "aliases": [],
145             "options": {},
146             "properties": {},
147             "name": "Password",

```



---

```
145     "label": "Password",
146     "defaultValue": "",
147     "required": true,
148     "type": "string",
149     "helpMarkdown": "",
150     "groupName": ""
151 },
152 {
153     "aliases": [],
154     "options": {},
155     "properties": {},
156     "name": "Solucion",
157     "label": "Solucion",
158     "defaultValue": "Predeterminada",
159     "required": true,
160     "type": "string",
161     "helpMarkdown": "Nombre de la solucion que se importara desde el control de
    ↪ versiones",
162     "groupName": ""
163 },
164 {
165     "aliases": [],
166     "options": {},
167     "properties": {},
168     "name": "Url",
169     "label": "Url",
170     "defaultValue": "",
171     "required": true,
172     "type": "string",
173     "helpMarkdown": "URL del CRM",
174     "groupName": ""
175 },
176 {
177     "aliases": [],
178     "options": {},
179     "properties": {},
180     "name": "Username",
181     "label": "Username",
182     "defaultValue": "",
183     "required": true,
```

```
184     "type": "string",
185     "helpMarkdown": "",
186     "groupName": ""
187   }
188 ],
189 "satisfies": [],
190 "sourceDefinitions": [],
191 "dataSourceBindings": [],
192 "instanceNameFormat": "Task group: TG-02: Restaurar Cambios a CRM $(AuthType)",
193 "preJobExecution": {},
194 "execution": {},
195 "postJobExecution": {}
196 }
```

---

# Apéndice F

## Contenido del CD

El CD que acompaña a esta memoria contiene una copia de la misma en formato PDF ubicada en el directorio raíz bajo el nombre *memoria.pdf*. Así mismo en el directorio raíz se incluyen los manuales de la aplicación en formato PDF con el nombre de archivo *Manual Usuario.pdf* para el Manual de Usuario y *Manual Instalacion.pdf* para el Manual de Instalación. El directorio *Codigo Fuente* contiene 3 archivos: El código fuente de los scripts powershell contenido en el archivo *ScriptPS.ps1*. El código fuente procedente de la exportación de los “Task Group” TG-01 y TG-02 contenido en los archivos *TG01.JSON* y *TG02.JSON* respectivamente. La imagen F.1 muestra el árbol de directorios correspondiente al CD.

```
| Manual Instalacion.pdf
| Manual Usuario.pdf
| memoria.pdf
\---Codigo Fuente
    ScriptPS.ps1
    TG01.JSON
    TG02.JSON
```

Figura F.1: Árbol de Directorios del CD