



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**GRADO EN INGENIERÍA DE ORGANIZACIÓN INDUSTRIAL**

# **Parametrización de un CRM para el Sector de las Comunicaciones**

**Autor:**

**Hernández Cordero, Beatriz**

**Tutor:**

**Redondo Castán, Alfonso  
Departamento de Organización de  
Empresas y C. e I.M.**

**Valladolid, Julio 2019.**



## **Resumen**

En la actualidad, las Tecnologías de la Información y de la Comunicación (TIC) se han convertido en un elemento indispensable para el control de la gestión de todo tipo de organizaciones y empresas. Uno de sus puntos fuertes es el de potenciar las relaciones cliente-empresa a través de herramientas como los CRM (Customer Relationship Management) que contribuyen a optimizar el par Gestión-Información, si bien, su coste de implementación suele ser elevado.

A lo largo de este trabajo se van a analizar las distintas herramientas que componen un CRM, y se seleccionarán las más adecuadas para la implementación de un CRM, de bajo coste, y que pueda ser utilizado por cualquier tipo de organización, independientemente de los recursos que disponga.

## **Palabras clave**

TIC, CRM, Herramientas de Gestión y Herramientas de desarrollo

## **Abstract**

Nowadays, Information and Communication Technologies (ICT) have become an indispensable element for the management control of all types of organizations and companies. One of its strong points is that of strengthening customer-company relations through tools such as CRM (Customer Relationship Management) that contribute to optimising the Management-Information pair, although the cost of implementation is usually high.

Throughout this work, the different tools that make up a CRM will be analysed, and the most suitable ones will be selected for the implementation of a low-cost CRM that can be used by any type of organisation, regardless of the resources available.

## **Keywords**

ICT, CRM, Process Tools, Development Tools.



# INDICE

## INDICE

INDICE.....	V
INDICE DE FIGURAS.....	IX
INDICE DE TABLAS.....	XI
INTRODUCCIÓN .....	XIII
INTRODUCCIÓN .....	XV
JUSTIFICACIÓN DEL PROYECTO.....	XVI
OBJETIVOS DEL PROYECTO .....	XVI
ESTRUCTURA DEL PROYECTO .....	XVII
CAPÍTULO 1 – TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN (TIC) .....	19
1.1. TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN .....	21
1.2. CARACTERÍSTICAS DE LAS TIC'S .....	23
1.3. TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN EN LA EMPRESA 24	
1.4. CLASIFICACIÓN DE LAS TIC'S EN LAS EMPRESAS .....	26
1.5. INCLUSIÓN DEL PAPEL DE LAS TIC'S EN LA EMPRESA .....	27
1.6. PROBLEMA DE LAS TIC'S EN LAS EMPRESA Y SOLUCIÓN .....	28
CAPÍTULO 2 – CUSTOMER RELATIONSHIP MANAGEMENT (CRM) .....	31
2.1. CONCEPTO DE CRM .....	33
2.2. OBJETIVOS, CARACTERÍSTICAS Y BENEFICIOS DEL CRM .....	35
2.3. TIPOS DE CRM.....	38
2.4. ESTRUCTURA DEL CRM .....	39
2.5. PROCESO DE IMPLANTACIÓN DE UN CRM.....	41
CAPÍTULO 3 – SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO .....	47
3.1. SISTEMAS GESTORES DE BASES DE DATOS.....	49
3.1.1. MySQL.....	51
3.1.2. ORACLE .....	52
3.1.3. SQLite.....	53
3.1.4. POSTGRESQL.....	54
3.1.5. SQL Server .....	55

3.2.1.	ECLIPSE .....	57
3.2.2.	NETBEANS .....	59
3.2.3.	INTELLIJ IDEA.....	60
3.3.	GESTORES DE PROCESOS DE NEGOCIOS.....	61
3.3.1.	jbpm.....	63
3.3.2.	BonitaBPM.....	64
3.3.3.	Activiti.....	66
CAPÍTULO 4 – SELECCIÓN DE LAS HERRAMIENTAS DE GESTIÓN.....		69
4.1.	GESTORES DE CONTROL DE VERSIONES.....	71
4.1.1.	SUBVERSION (SVN).....	73
4.1.2.	GIT .....	76
4.1.3.	CVS.....	77
4.1.4.	MERCURIAL .....	78
4.2.	GESTORES DE PRUEBAS .....	80
4.2.1.	TESTLINK .....	81
4.2.2.	SELENIUM.....	83
4.3.	HERRAMIENTAS DE DESPLIEGUE .....	84
4.3.1.	JENKINS.....	85
4.3.2.	BAMBOO .....	86
4.4.	GESTORES DE INCIDENCIAS .....	88
4.4.1.	JIRA .....	89
4.4.2.	MANTIS BUG TRACKER.....	90
4.4.3.	REDMINE .....	91
4.4.4.	BUGZILLA.....	93
CAPÍTULO 5 - CASO PRÁCTICO.....		95
5.1.	DEFINICIÓN DEL CASO PRÁCTICO .....	97
5.2.	ESTRUCTURA DE LA ORGANIZACIÓN .....	97
5.3.	UTILIZACIÓN DE MYSQL.....	100
5.4.	UTILIZACIÓN DE SUBVERSION.....	104
5.5.	UTILIZACIÓN DE ECLIPSE.....	105
5.6.	UTILIZACIÓN DE jbpm .....	105
5.7.	UTILIZACIÓN DE JENKINS .....	113

5.8. UTILIZACIÓN DE TESTLINK .....	116
5.9. UTILIZACIÓN DE JIRA.....	119
CONCLUSIONES .....	123
BIBLIOGRAFÍA.....	125
ANEXOS.....	129
ANEXO 1. TABLAS DIAGRAMAS ENTIDAD RELACIÓN .....	129



## INDICE DE FIGURAS

Figura 1. Áreas principales del CRM.....	34
Figura 2. Etapas de implantación de un CRM.....	42
Figura 3. Sistema Gestor de Base de Datos relacional.....	49
Figura 4.Sistema Gestor de Base de Datos en red .....	50
Figura 5.Sistema Gestor de Base de Datos jerárquico .....	50
Figura 6. Logo MySQL.....	51
Figura 7.Logo Oracle.....	52
Figura 8. Logo SQLite.....	53
Figura 9. Logo PostgreSQL .....	54
Figura 10. Logo SQL Server.....	55
Figura 11. Logo Eclipse .....	57
Figura 12. Logo Netbeans .....	59
Figura 13. Logo IntelliJ IDEA .....	60
Figura 14. Ciclo de vida del BMP .....	62
Figura 15. Logo jBPM .....	63
Figura 16. Logo BonitaBPM.....	64
Figura 17. Logo Activiti .....	66
Figura 18. Sistema de Control de Versiones centralizado .....	71
Figura 19. Sistema de Control de Versiones distribuido.....	72
Figura 20. Logo Subversion (SVN) .....	73
Figura 21. Logo Git.....	76
Figura 22. Logo CVS.....	77
Figura 23. Logo Mercurial .....	78
Figura 24. Logo TestLink .....	81
Figura 25. Logo Selenium .....	83
Figura 26. Logo Jenkins.....	85
Figura 27. Logo Bamboo .....	86
Figura 28. Logo Jira .....	89
Figura 29. Logo Mantis Bug Tracker.....	90
Figura 30. Logo Redmine .....	91
Figura 31. Logo Bugzilla .....	93
Figura 32. Organización jerárquica.....	98
Figura 33. Modelo relacional de BBDD .....	102
Figura 34. Cardinalidades del modelo relacional .....	103
Figura 35. Proceso Alta Móvil.....	107
Figura 36. Proceso Cancelación Móvil .....	108
Figura 37. Proceso Baja Móvil .....	109
Figura 38. Proceso Alta Internet .....	110
Figura 39. Proceso Cancelación Internet.....	111
Figura 40. Proceso Baja Internet .....	112

Figura 41. Usuarios creados en Jenkins .....	113
Figura 42. Creación del proyecto en Jenkins .....	113
Figura 43. Ciclo de vida del proyecto.....	114
Figura 44. Diagrama de flujo correctivo .....	120
Figura 45. Diagrama de flujo de calidad de datos .....	121

## INDICE DE TABLAS

Tabla 1. Cuadro comparativo SGBD.....	56
Tabla 2. Cuadro comparativo de entornos de desarrollo integrado .....	61
Tabla 3. Cuadro comparativo de gestores de procesos de negocio .....	67
Tabla 4. Cuadro comparativo de sistemas de control de versiones .....	79
Tabla 5. Cuadro comparativo de gestores de pruebas .....	84
Tabla 6. Cuadro comparativo de herramientas de despliegue.....	88
Tabla 7. Cuadro comparativo de gestores de incidencias .....	94
Tabla 8. Pruebas unitarias caso 1.....	116
Tabla 9. Prueba unitaria caso 2.....	117
Tabla 10. Pruebas de integración caso 1 .....	117
Tabla 11. Pruebas de integración caso 2 .....	118
Tabla 12. Pruebas de integración caso 3 .....	118
Tabla 13. Atributos tabla dirección.....	129
Tabla 14. Atributos tabla cliente.....	129
Tabla 15. Atributos tabla contrato.....	129
Tabla 16. Atributos tabla procesos .....	130
Tabla 17. Atributos tabla técnicos .....	130
Tabla 18. Atributos tabla equipos.....	130
Tabla 19. Atributos tabla precios.....	131
Tabla 20. Atributos tabla productos .....	131
Tabla 21. Atributos tabla proveedores .....	131
Tabla 22. Atributos tabla parque .....	132



# INTRODUCCIÓN





## INTRODUCCIÓN

El auge de las telecomunicaciones, especialmente, el desarrollo tecnológico, está produciendo grandes cambios, que han afectado a todos los sectores.

Es de resaltar, la importancia de las tecnologías de la información y la comunicación en las organizaciones. Son un elemento clave para que el trabajo, que se realiza en una empresa sea más productivo. Las TIC permiten mejorar sus procesos, tanto en eficacia como en eficiencia, así como la colaboración entre distintos grupos de trabajo y la toma de decisiones. Todo esto permite fortalecer su posición competitiva en el mercado actual.

Las necesidades de las empresas para mejorar las relaciones con los clientes, las ha llevado a trabajar con softwares dedicados a la gestión de clientes. Estos softwares se conocen como CRM o Customer Relationship Management. Sin embargo, para poder adquirir o implantar uno de estos sistemas las empresas necesitan ciertos recursos.

Las empresas tienen una mayor necesidad de implantar un CRM para poder gestionar el gran volumen de datos e información que poseen de sus clientes. Un software CRM es un programa destinado a administrar la relación con los clientes actuales y potenciales. Este debería estar totalmente integrado en los diferentes procesos de la empresa.

Gestionar la relación con los clientes, significa analizar todas las interacciones y los datos obtenidos a lo largo del ciclo de vida del cliente, para conocerle mejor y ofrecerle servicios y productos que le aporten realmente valor.

Sin embargo, la elección de una herramienta de gestión que cumpla con todas las necesidades de la empresa y que además esta sea de bajo coste, es un paso decisivo para cualquier organización.

Por eso, actualmente está cobrando especial relevancia los CRM de software libre o de bajo coste. Gracias a estos tipos de CRM, cualquier empresa actualmente dispone de uno de estos sistemas de gestión de clientes. Los CRM de software libre, son soluciones de menor coste y alcanzan objetivos parecidos a los CRM de software propietario.



## JUSTIFICACIÓN DEL PROYECTO

En la actualidad, todas las empresas y en especial las dedicadas a las telecomunicaciones, necesitan utilizar un software CRM que les permita gestionar de la manera más adecuada las relaciones entre los clientes actuales y los potenciales, así como prestar los servicios necesarios.

Las empresas del sector de las comunicaciones lanzan cada año cientos de servicios y productos nuevos, así como variaciones de los servicios existentes. Podemos decir, que el motivo de estos lanzamientos anuales es bastante simple. Con ellos, las empresas pretenden estar a la cabeza del sector haciendo frente a la competencia.

Para que las empresas puedan lanzar estos nuevos productos y servicios, necesitan disponer de un CRM. El principal problema para las empresas es el mantenimiento de esta herramienta software, que supone un elevado coste, no solo económico sino también humano, debido a que la mayor parte de ellas utilizan programas comerciales que tienen un coste económico considerable.

Es por ello que muchas empresas de nueva implantación en el sector o incluso las pymes no dispongan de los suficientes recursos como para utilizar un CRM.

La realización de este trabajo fin de grado se fundamenta en la necesidad de desarrollar un CRM, que permita gestionar de forma adecuada las relaciones con los clientes y prestar los servicios con la máxima calidad posible a un bajo coste.

En este trabajo, se ha parametrizado un CRM de bajo coste, el cual, utiliza herramientas “open source”.

## OBJETIVOS DEL PROYECTO

Este TFG tiene como objetivo principal el desarrollo teórico de un nuevo CRM mediante la utilización de herramientas de bajo coste. De este modo, cualquier tipo de organización dedicada a las telecomunicaciones podrá hacer uso de él, y verse beneficiada de su utilización.

Para ello, es necesario cubrir los siguientes objetivos secundarios:



- Comprender la importancia que tienen las tecnologías de la información y la comunicación en las organizaciones.
- Comprender la importancia que tiene la utilización de un CRM en la empresa y los beneficios que su utilización aporta.
- Establecer la metodología a seguir para poder implantar un CRM.
- Conocer las distintas herramientas que nos ofrece el mercado para la elaboración de un CRM.
- Analizar y seleccionar las herramientas más idóneas para la elaboración del CRM y a su vez que nos ofrece el mercado.
- Entender el uso de las herramientas seleccionadas con anterioridad.

## ESTRUCTURA DEL PROYECTO

Al objeto de alcanzar los objetivos previstos, y tras justificar el alcance de los mismos, el trabajo se ha estructurado de la siguiente forma:

En el capítulo 1 hemos contextualizado el problema. Realizando una introducción a las Tecnologías de la Información y la Comunicación (TIC), poniendo de manifiesto su importancia en la sociedad actual, y mostrando los pasos a seguir para la implantación de estas tecnologías en cualquier organización.

Para potenciar la relación con los clientes hemos propuesto el uso de los CRM, y por ello, en el capítulo 3, les hemos caracterizado, descrito sus principales objetivos, tipologías, beneficios, etc., así como la descripción de la estructura básica de su implantación.

En el capítulo 4 y 5, se exponen las distintas herramientas de desarrollo y gestión. Para cada una de ellas se realizará una breve descripción y se indicaran sus características más importantes. Por último, se realiza un cuadro comparativo del que se obtendrá la/s herramienta/s a utilizar.

El capítulo 6, está destinado a exponer un caso práctico que se podría dar en cualquier empresa del sector de las comunicaciones. Con este caso práctico, se pretende explicar cómo se utilizarán las herramientas que se han seleccionado en los capítulos anteriores y que conformaran el CRM que se pretende crear.



Para finalizar este TFG, se exponen las conclusiones finales, de la misma forma se incluye la bibliografía consultada y los anexos necesarios.

# CAPÍTULO 1 – TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN (TIC)



## 1.1. TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Las tecnologías de la información y la comunicación, también conocidas como TIC'S, se pueden definir como el conjunto de tecnologías desarrolladas para gestionar y transmitir la información de un lugar a otro. Aun así, cada autor tiene su propia definición.

Thompson y Strickland <sup>1</sup> (2004) definen las TIC'S como

*“Aquellos dispositivos, herramientas, equipos y componentes electrónicos, capaces de manipular información que soportan el desarrollo y crecimiento económico de cualquier organización. Cabe destacar que en ambientes tan complejos como los que se deben enfrentar hoy en día las organizaciones, solo aquellos que utilicen todos los medios a su alcance, y aprendan a aprovechar las oportunidades del mercado visualizando siempre amenazas, podrán lograr el objetivo de ser exitosas”.*

Por su parte Graells<sup>2</sup> (2000), las define como

*“Un conjunto de avances tecnológicos, posibilitados por la informática, las telecomunicaciones y las tecnologías audiovisuales, todas están proporcionan herramientas para el tratamiento y la difusión de la información y contar con diversos canales de comunicación. El elemento más poderoso que integra las TIC'S es la Internet, que ha llevado a la configuración de la llamada Sociedad de la Información”.*

Por otro lado según Gil<sup>3</sup> (2002) las TIC'S

*“Constituyen un conjunto de aplicaciones, sistemas, señales, herramientas, técnicas y metodologías asociadas a la digitalización de señales analógicas, sonidos, textos e imágenes, manejables en tiempo real.”*

Para Macau<sup>4</sup> (2004), las TIC'S presentan diversas funciones, entre las que podemos destacar:

---

<sup>1</sup> THOMPSON, A. y STRICKLAND, A. (2004). *Administración Estratégica*. México: Mc Graw Hill

<sup>2</sup> MARQUES GRAELLS, P. (2000). *“Impacto Tic en Educación: Funciones y Limitaciones”*

<sup>3</sup> GIL, E. (2002). *“Identidad y Nuevas Tecnologías”*.

1. **Automatización del proceso administrativo y burocrático:** la mala gestión o el mal funcionamiento de las TIC'S en los procesos administrativos y burocráticos, crea situaciones de desventaja competitiva en costes o en eficiencia respecto a la competencia. Se trata de una función obligada en las organizaciones actuales. El uso de las TIC'S en esta función responde al paradigma industrial. Como valor añadido, las TIC'S permiten plantearse, de manera económicamente realista, actuaciones integradoras y transversales en los circuitos administrativos de una organización, así como, asumir grandes volúmenes.
  
2. **Infraestructura necesaria para el proceso de gestión:** es una de las funciones fundamentales para la organización. La definición de un sistema de información es responsabilidad de la dirección de la organización. Sin un sistema eficaz de información de gestión es difícil cuantificar los problemas a tiempo.
  
3. **Parte integrante del producto, servicio o cadena de producción:** la importancia de esta función varía dependiendo del sector productivo del que hablemos. Su importancia viene condicionada por el peso del contenido informativo del producto, del servicio o de la cadena de producción. Esta función tiende a incrementarse exponencialmente. Numerosos sectores de producción y servicios quedaran alterados, dado que:
  - Todo proceso productivo lleva asociado un componente informativo y uno físico.
  - Las aplicaciones se difunden masivamente.
  - La capacidad de elaboración y tratamiento de la información por parte de las personas, ha aumentado de forma extraordinaria en los últimos años.
  - Las aplicaciones multimedia se generalizan.Aun así, el peso de las TIC'S en las actividades operativas del sector servicios se verá incrementado.
  
4. **Pieza clave en el diseño de la organización y sus actividades:** el análisis de la cadena de valor proporciona los elementos necesarios para el diseño de los objetivos y la estructura de la organización. Las nuevas formas en red que irán adoptando las organizaciones, no podrán

---

<sup>4</sup> MACAU, R (2004). "TIC: ¿para qué? (Funciones de las tecnologías de la información y la comunicación en las organizaciones)". Revista de Universidad y Sociedad del Conocimiento (RUSC). Vol. 1, nº 1.

construirse sin un uso intensivo de las TIC'S. Esta tarea debe ser asumida por la dirección de la organización y debe de contar con la ayuda y el consejo de profesionales de la tecnología.

## 1.2. CARACTERÍSTICAS DE LAS TIC'S

Para autores como Cabero<sup>5</sup> (1998), las características más representativas de las TIC'S son:

- **Interactividad:** considerada la característica más importante. Con el uso de las TIC se consigue un intercambio de información entre el usuario y el ordenador. Esta característica permite adaptar los recursos utilizados a las características y necesidades de los usuarios, en función de la interacción entre ordenador y usuario.
- **Inmaterialidad:** se puede afirmar que las TIC realizan el proceso, la creación y la comunicación de la información. Esta información es básicamente inmaterial, y por lo tanto puede ser llevada de forma instantánea y transparente a lugares lejanos.
- **Instantaneidad:** las redes de la comunicación y su integración con la informática, han permitido la utilización de servicios, que permiten la transmisión y la comunicación de información entre lugares geográficamente muy alejados, de una forma rápida.
- **Interconexión:** hace referencia a la posibilidad de creación de nuevas posibilidades de tecnologías a partir de la conexión entre dos tecnologías.
- **Penetración en todos los sectores:** el impacto que están teniendo las TIC en la actualidad, no solo se refleja en un individuo, sector o país, sino que, se extiende al conjunto de las sociedades del planeta. Los términos "Sociedad de la Información" y "Globalización", hacen alusión a este proceso. De esta manera los efectos se extenderán a todos los grupos, habitantes e instituciones conllevando importantes cambios.
- **Tendencia hacia la automatización:** la aparición de diferentes herramientas y posibilidades, que permiten el manejo automático de la información de diversas actividades profesionales, sociales y personales, hace que sea necesario disponer de información

---

<sup>5</sup> CABERO, J. (1998). *Impacto de las nuevas tecnologías de la información y la comunicación en las organizaciones educativas*. En Lorenzo, M. y otros (coords): *Enfoques en la organización y dirección de instituciones educativas formales y no formales* (pp. 197-206). Granada: Grupo Editorial Universitario.

estructurada. Lo que conlleva el desarrollo de gestores personales o corporativos, con distintos fines y de acuerdo con unos determinados principios.

- **Diversidad:** la utilidad de las tecnologías puede ser muy diversa, desde la comunicación entre dos personas, hasta el proceso de la información para crear informaciones nuevas.
- **Innovación:** las TIC están llevando a cabo una innovación y cambio constante en todos los ámbitos sociales. No obstante, es de reseñar, que estos cambios, no siempre indican un rechazo de las tecnologías, sino que en algunos casos, se produce una serie de fusiones con otros medios.
- **Elevados parámetros de calidad, imagen y sonido:** el proceso de transmisión de información abarca todo tipo de información, ya sea textual, imagen o sonido. Todo esto, ha hecho que los avances hayan ido encaminados a conseguir transmisiones multimedia de gran calidad, lo cual ha facilitado el proceso de digitalización.
- **Digitalización:** el objetivo es que la información pueda ser transmitida por los mismos medios al estar representada en un formato universal.
- **Mayor influencia sobre los procesos que sobre los productos:** es posible que el uso de diversas aplicaciones de las TIC, presente mayor influencia sobre los procesos mentales que realizan los usuarios para la adquisición de conocimientos, más que sobre los propios conocimientos adquiridos. En los distintos análisis realizados, sobre la Sociedad de la Información, se remarca la enorme importancia de la inmensidad de información a la que permite acceder internet. Sin embargo, diversos autores han señalado el efecto negativo de la proliferación de la información, los problemas de calidad de la misma y la evolución hacia aspectos sociales.

### 1.3. TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN EN LA EMPRESA

Las TIC'S, son actualmente uno de los temas de investigación más importantes en el campo de la dirección de empresas y la economía, se han convertido en algo imprescindible en la empresa.

Son herramientas que permiten procesar datos y convertirlos en información de análisis, pero también para agilizar operaciones y



actividades empresariales de manera rápida y segura, y por lo tanto, ayudar a la toma de decisiones.

Por otro lado, también son herramientas para la innovación, lo que facilita la innovación de procesos, productos y servicios e incluso proporcionando nuevos modos de organización y trabajo.

La incorporación de las TIC'S en la empresa no garantiza que se produzca un incremento de la productividad, como tampoco de la mejora de sus procesos productivos o de su competitividad.

Podemos decir que las TIC'S están relacionadas con el crecimiento económico, de manera que, si queremos obtener el máximo beneficio económico, las empresas deben contemplar las TIC'S dentro de su estrategia tecnológica.

La implantación de las TIC'S en las empresas ya no es una ventaja diferenciadora, al contrario, es una necesidad para no quedarse atrás. Las empresas tienen que tener en cuenta que la incorporación de nuevas tecnologías no solo requiere de su implantación sino también es necesario un equipo humano bien preparado y especializado que adopten y usen, de manera eficiente y correcta estas tecnologías.

Es por eso, que las empresas deberán prestar cierto interés en la formación de personal, y en la realización de inversiones en estas tecnologías. De no ser así, cualquier empresa podría llegar a perder su ventaja competitiva y finalmente ser expulsada del mercado.

Por otro lado, una mayor inversión en TIC'S por parte de las empresas aportará múltiples beneficios como:

- 1) **Mejorar las relaciones con los clientes:** las relaciones con los clientes se ven reforzadas. Las nuevas tecnologías permiten que el cliente reciba una atención más personalizada, permitiendo a la empresa conocer más exhaustivamente las necesidades del cliente, para poder aplicar las estrategias de fidelización, todo esto se consigue gracias a las aplicaciones CRM.
- 2) **Reducir los tiempos y los costos:** las TIC'S mejoran las relaciones agilizándolas, gracias al uso de sistemas B2B (Business to Business). El uso de aplicaciones ERP, hace que sea más fácil la gestión empresarial, y optimizan la logística y el almacenaje, permitiendo al trabajador asignar tareas, controlar actividades y resolver incidencias.

- 3) **Mejorar la posición en el mercado:** el uso de sistemas B2C (Business to Customer) permite acceder a nuevos mercados y a clientes potenciales. Por otro lado, las herramientas de inteligencia que dan soporte a las decisiones, ayudan a convertir la información en conocimiento útil, lo que proporciona un posicionamiento por delante de la competencia.
- 4) **Facilita la comunicación interna de la empresa:** el uso de intranets corporativas y de herramientas de trabajo en grupo hacen que se mejoren las comunicaciones entre empleados, así como, el uso de sistemas de gestión documental que facilitan la organización de los contenidos.

## 1.4. CLASIFICACIÓN DE LAS TIC'S EN LAS EMPRESAS

Podemos clasificar las tecnologías de la información y la comunicación dentro de las empresas de la siguiente forma<sup>6</sup>:

### 1. Tecnologías de la información y la comunicación orientadas a la comunicación.

Este grupo está compuesto por tecnologías destinadas a reducir los costos de las comunicaciones, y a facilitar la transmisión de la información entre las diferentes áreas de la compañía. Posibilitan el intercambio de información con los grupos de interés externos.

También facilitan el acercamiento y la colaboración de los involucrados en el proceso comunicativo. Estableciendo una comunicación de doble vía. Estas tecnologías ayudan a alinear a la organización con sus objetivos estratégicos.

### 2. Tecnologías de la información y la comunicación orientadas a los flujos de trabajo.

Dentro de este tipo de TIC encontramos todas las tecnologías que posibilitan la integración de las diferentes capacidades empresariales para la obtención de resultados y la automatización de procesos. Además, ayudan a superar las divisiones funcionales de las compañías y aceleran la incorporación de un modelo de gestión de procesos.

---

<sup>6</sup> HOYOS CHAVERRA, J.A y VALENCIA ARIAS, A. (2012). "El Papel De Las TIC En El Entorno Organizacional De Las Pymes". Trilogía Ciencia Tecnología Sociedad 4 (7), 105-22.

Estas herramientas ayudan a activar, en el momento adecuado, las capacidades de la empresa para obtener resultados. Por lo cual, también contribuyen a la optimización en el uso de los recursos.

### 3. Tecnologías de la información y la comunicación orientadas a la información.

Dentro de este grupo encontramos las tecnologías utilizadas para el almacenamiento y procesamiento de datos. Estas, principalmente, son empleadas para proporcionar y distribuir información a los diferentes procesos de la organización. Su objetivo es extender los procesos de memoria de la empresa, es decir, la consolidación de bases de datos con información histórica, actualizada y no duplicada, con lo que además, ayudan a optimizar los procesos generados del mantenimiento de la información y capacidad de almacenamiento.

A mayores, este grupo de tecnologías, permite la generación automática de informes y búsquedas avanzadas, por lo que permite la liberación de recursos orientados a labores operativas, pudiéndose enfocar a procesos estratégicos. Lo que potencia aún más los procesos de toma de decisiones.

## 1.5. INCLUSIÓN DEL PAPEL DE LAS TIC'S EN LA EMPRESA

Cuando una empresa quiera incluir, mejorar o ampliar el papel de las TIC'S en su gestión, debería pensar que esto debe servir para mejorar su ventaja competitiva.

Para poder llevarlo a cabo esta inclusión, se debe realizar los siguientes pasos:

1. **Definir un marco estratégico:** será responsabilidad del personal de alta dirección de la organización. En él, se deberá incluir la visión y la misión de la empresa.
2. **Realizar un diagnóstico estratégico:** lo que viene siendo la realización tanto de un análisis externo como interno de la empresa. En el análisis externo, se detectarán y evaluarán los acontecimientos y tendencias que sucedan en el entorno de la empresa, con el fin de conocer la situación del entorno y detectar oportunidades y amenazas. Mientras que, en el interno, se encarga de estudiar aspectos o elementos dentro de la empresa, con el fin de conocer el

estado y la capacidad con que esta cuenta y detectar sus fortalezas y debilidades.

3. **Establecer los objetivos de la empresa:** estos objetivos siempre serán a largo plazo. Fijar los objetivos de manera que permitan conseguir la misión propuesta y que permitan capitalizar las oportunidades y fortalezas, superando las amenazas y debilidades. Además, se deberán tener en cuenta los recursos de la empresa y la situación del entorno.
4. **Diseño, evaluación y selección de estrategias:** se evaluarán los informes relacionados con el análisis interno y externo, así como los objetivos establecidos y las estrategias pasadas. Se diseñarán estrategias y se determinarán las ventajas e inconvenientes, los costos y los beneficios. Por último, se seleccionarán las estrategias a utilizar, clasificadas por orden de importancia.
5. **Diseño de planes estratégicos:** se deberá señalar,
  - Los objetivos específicos que permitan alcanzar los objetivos generales.
  - Estrategias específicas.
  - Recursos a utilizar y el modo en el que se distribuirán.
  - Encargados o responsables de la implantación o ejecución de las estrategias.
  - Planificación temporal de la implantación.
  - Inversión para la implantación.

## 1.6. PROBLEMA DE LAS TIC'S EN LAS EMPRESA Y SOLUCIÓN

La implantación de las TIC'S en las empresas conlleva que surjan determinados problemas. Entre estos podemos destacar:

- **Incremento de los costes de las empresas:** este incremento será proporcional al tamaño de la empresa y de sus necesidades. Normalmente, esta elevación de los costes está relacionada con la instalación de determinados programas o plataformas. La mayoría de los programas o aplicaciones necesarias en una empresa requieren de la compra de licencias o permisos. Un ejemplo son la creación de las bases de datos.
- **Inestabilidad de los puestos de trabajo:** la automatización de muchos de los procesos que se llevan a cabo en las empresas, hace que sea posible prescindir de algunos puestos de trabajo, aumentando por lo tanto las posibilidades de despido.

- **Inseguridad informática:** la introducción de nuevas tecnología en las organizaciones hacen que estas se vuelvan más vulnerables ante un ciberataque. Por consiguiente, las empresas deberán contar con personal especializado en materias de seguridad informática.

Como hemos mencionado con anterioridad en este capítulo, una de las características más importantes de las TIC'S, es la relación con los clientes. Para garantizar esta relación estable y duradera entre cliente y compañía se utilizan sistemas de gestión de clientes, más conocidos como CRM.

Según estudios llevados a cabo por grandes consultoras, el número de empresas que utilizan CRM es cercano al 70%. Dado que es un sector que en la actualidad se encuentra en alza, casi todas las grandes compañías hacen uso de complejos sistemas de CRM, basados a su vez en TIC'S muy complejas y de elevado coste.

Pero no nos hemos parado a pensar que dentro del sector empresarial existe un pequeño sector dedicado a la pequeña y mediana empresa. Estas empresas, no disponen de los mismos recursos con los que cuenta una gran empresa y por tanto la utilización de este tipo de herramientas es escasa.

Por lo tanto, en este TFG se va a plantear la implementación de un nuevo CRM, este se caracterizará porque la mayor parte de las herramientas que utiliza son gratuitas.

Hoy en día, cada vez más pymes, utilizan CRM de software libre existentes ya en el mercado. Lo que diferencia a nuestro CRM de los existentes en el mercado es:

- Es de bajo coste, debido a que la mayor parte de las herramientas que utiliza son "open source". Por lo tanto, la empresa no tendrá que destinar parte de sus recursos económicos en coste de licencias.
- Podrá ser utilizado por todo tipo de empresas independientemente de su tamaño, tendrá la flexibilidad necesaria para adaptarse a cualquier tipo de necesidades.
- Garantizará la calidad de los servicios que ofrece, disminuyendo así el coste del mantenimiento de la aplicación.
- Permitirá crear estrategias de negocio, que faciliten la fidelización del nuevo cliente, así como el mantenimiento de los existentes.



CAPÍTULO 2 –  
CUSTOMER  
RELATIONSHIP  
MANAGEMENT  
(CRM)



## 2.1. CONCEPTO DE CRM

Existen muchas definiciones de CRM, una de las más utilizadas es que el CRM es una estrategia de negocio que se focaliza en el cliente, y cuyo principal objetivo, es reunir toda la información posible sobre los clientes, para poder establecer relaciones duraderas y aumentar el grado de satisfacción.

Pero son muchos los autores que aportan su propia definición, según Parvatiyar y Sheth<sup>7</sup> (2001),

*“El CRM engloba tanto la estrategia como los procesos que comprenden la adquisición, retención y asociación con determinados clientes con objeto de crear valor superior tanto para la compañía como para el propio cliente.”*

Chen y Popovich<sup>8</sup> (2003), afirman que el CRM

*“No es solo una aplicación tecnológica, es una estrategia de negocio que aglutina las funciones de marketing, ventas, servicios al cliente, operaciones, recursos humanos, I+D, finanzas y TI con el objeto de maximizar la rentabilidad de las interacciones con clientes.”*

Por su parte, Finnegan y Currie<sup>9</sup> (2010) resaltan que el CRM

*“No es solo un paquete de Software, sino un enfoque estratégico integral para gestionar la evolución de las relaciones con los clientes que requiere de una adaptación continua en respuesta a las necesidades cambiantes del mercado.”*

Choy, Fan y Lo<sup>10</sup> (2003) consideran que el CRM

*“Supone una integración en toda la empresa de tecnologías trabajando conjuntamente como son almacenamiento de datos, sitio web, intranet-*

---

<sup>7</sup> PARVATIYAR, A. y SHETH, J.N. (2001): “Customer Relationship Management: Emerging Practice, Process, and Discipline”. Journal of Economic and Social Research, 3 (2), pp. 1-34

<sup>8</sup> CHEN, I.J. y POPOVICH, K. (2003): “Understanding Customer Relationship Management-People, Process and Technology”, Business Process Management Journal, 9 (5), pp. 672-688.

<sup>9</sup> FINNEGAN, D.J. y CURRIE, W.L. (2010): “A multiLayered Approach to CRM Implementation: An Integration Perspective”, European Management Journal, 28, pp. 153-167.

<sup>10</sup> CHOY, K.L. y FAN, K.; LO, V. (2003): “Development of an Intelligent Customer-Supplier Relationship Management System: The Application of CaseBased Reasoning”, Industrial Management +Data System, 103 (3-4), pp. 263-274

*extranet, sistema de apoyo telefónico, contabilidad, marketing, ventas y producción, para permitir la comunicación entre las distintas partes de la organización y así servir mejor a la clientela.”*

Pero realmente, el CRM es más que un conjunto de herramientas conectadas a una BBDD con información detallada de clientes. Es una estrategia de negocio, que requiere cambios en la organización y en los procesos de trabajo, para finalmente hacer uso de las tecnologías de la información.

Para esto debe tenerse en cuenta que:

- El CRM no es solo una estrategia de negocio, también es una tecnología o un producto en sí mismo.
- Para realizar una estrategia de negocio, la organización debe encaminarse hacia un cambio en su estructura y en la forma de pensar, de modo que sus estrategias empiecen a centrarse en el cliente y no solo en los valores del mercado.
- Toda la información obtenida, se debe de utilizar para elaborar estrategias de marketing y ventas que sean únicas en el mercado y llamativas para los clientes.

Todo CRM está compuesto por tres áreas principales:

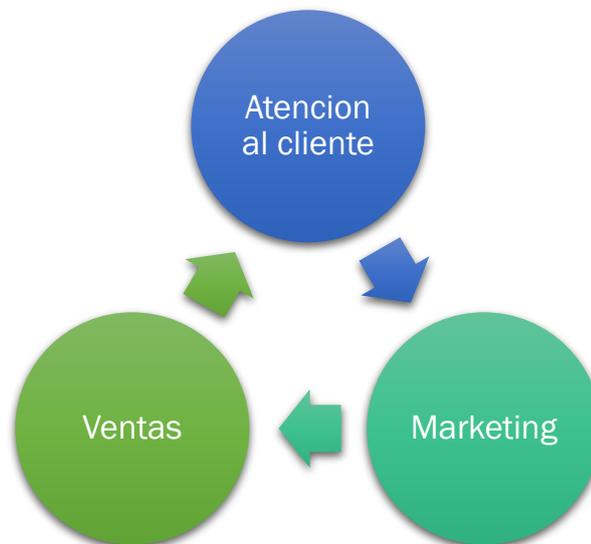


Figura 1. Áreas principales del CRM

- **Atención al cliente:** también conocido como área de soporte. Se encarga del servicio postventa, tiene como finalidad resolver problemas y dar asistencia a cualquier cosa que pueda surgir en los servicios y/o productos.
- **Marketing:** se encarga de dar a conocer los productos y /o servicios que la empresa ofrece al cliente. Para realizarlo, se pueden crear campañas de publicidad, anuncios, ferias o congresos.
- **Ventas:** área encargada de realizar las contrataciones y de convertir a los clientes potenciales en clientes finales.

Por lo tanto, el proceso de gestión que debe de realizar cualquier empresa es:

- Lanzar una campaña de marketing para poder generar clientes potenciales a los que intentar vender los productos o servicios.
- A continuación, se debe empezar a crear oportunidades con estos clientes potenciales, para convertirlos en clientes finales.
- Por último, se tendrán diferentes proyecciones de ventas que servirán para conocer cuál es el estado actual de la empresa.

## 2.2. OBJETIVOS, CARACTERÍSTICAS Y BENEFICIOS DEL CRM

El objetivo fundamental del CRM es que de manera metódica y continua identifica aquello que genera valor para el cliente, para posteriormente proporcionárselo en el momento y lugar adecuado, con la finalidad de lograr un crecimiento y mejor desempeño del negocio.

Otros de los objetivos que debe presentar un CRM son:

- **Proveer a los empleados de información y procesos:** es necesario que el CRM proporcione a los empleados información y procesos, los cuales son necesarios para conocer mejor al cliente, entender sus necesidades y construir relaciones productivas y mutuamente beneficiosas entre la empresa, sus asociados y sus clientes.
- **Individualización:** permite la formación de relaciones más focalizadas e individualizadas con el cliente, con el objetivo de mejorar el nivel de satisfacción del cliente y maximizando los beneficios, identificando los clientes más rentables y dándoles a estos el mejor nivel de servicio posible.

- **Ayudar a los departamentos de marketing:** el CRM ayuda a las organizaciones a identificar sus mejores clientes, a manejar las campañas de marketing a través de unas metas y objetivos claros. También sirve para generar informes de calidad para los equipos de venta.
- **Ayudar a las organizaciones:** El CRM debe de ser capaz de asistir a las organizaciones a mejorar los procesos de ventas y el manejo de cuentas, optimizando la información compartida por múltiples empleados y modernizando los procesos existentes.

Como sabemos, el CRM es una pieza fundamental para mejorar las relaciones con los clientes y potenciar lo máximo posible las ventas de la empresa. A continuación, se detallarán las características más destacadas:

- **Disponer de una visión completa del cliente:** debe proporcionar información acerca de la evolución de los clientes y del estado del proceso de venta. Adicionalmente, debe permitir la introducción de datos bajo petición para hacer un seguimiento de las ventas y ser escalable, para adaptarse al crecimiento de la compañía. Por otro lado, esta herramienta debe favorecer la realización de análisis completos sobre los gustos y necesidades de los clientes, teniendo así una visión global de los mismos.
- **Proporcionar la información adecuada:** debe ofrecer la posibilidad de crear diferentes tipos de usuarios y maneras de interactuar con los datos. De esta forma, cada uno de los empleados de la empresa vera la información relevante para su puesto de trabajo.
- **Optimizar la previsión del flujo de caja:** todo CRM no solo debe centrarse en gestionar las oportunidades de venta, también debe gestionar el proceso de pago, como parte del ciclo de vida del cliente. Debe permitir hacer un seguimiento de las fechas prevista de pago, así como el estado de pago. Por otro lado, este sistema se deberá encargar de informar tanto a los clientes como a los empleados cuando haya pagos pendientes.
- **Alinear las acciones de venta y marketing:** un buen CRM debe alinear las actividades de marketing y ventas, para distribuir el contenido al departamento comercial y maximizar la fuerza de ventas.

La utilización del CRM en la empresa reporta a estas los siguientes beneficios:

- **Aumento de la capacidad de fidelización de los clientes:** el uso de esta herramienta permite a la empresa aumentar el conocimiento que tiene de los clientes, comprendiendo y anticipándose a sus necesidades. Así, se facilitan las prácticas para mantener a un cliente, cuyo costo es menor que el de captación de nuevos clientes.
- **Optimización de la comunicación entre los departamentos de la empresa:** el uso del CRM permite obtener datos e información de los clientes en cada una de las etapas del ciclo de ventas. Este flujo de información es muy valioso para las distintas áreas de una empresa, por lo que su recolección y disponibilidad es esencial.
- **Mejorar la capacidad de respuesta:** los datos recogidos por el CRM permiten a la empresa detectar los problemas más comunes, permitiendo una visión global de las incidencias y una mejor evaluación de las necesidades de los clientes.
- **Ahorro de tiempo:** permite automatizar procesos, eliminando la necesidad de realizar acciones y tareas repetitivas. De esta manera, las diferentes áreas de la empresa pueden centrarse en objetivos más estratégicos y productivos.
- **Optimiza los recursos de la empresa:** el CRM permite desarrollar de la mejor forma el contacto y la relación con el cliente. De esta forma, la empresa será más eficiente en el uso de recursos, aumentando los márgenes y los beneficios.
- **Ventas relacionadas:** un aumento del conocimiento de los clientes aporta un impacto positivo en las ventas, permitiendo aumentar la cantidad de productos o servicios relacionados que el usuario adquiere.
- **Mejorar la gestión de las campañas de captación:** permite dirigir y gestionar de una forma más simple las campañas de captación de clientes.
- **Aumento de las oportunidades de venta:** agiliza la gestión con presupuestos actualizados en tiempo real. Y, a su vez, permite realizar una mejor segmentación de los clientes.
- **Mejora la comunicación:** permiten mejorar los procesos de comunicación tanto interna como externa. La idea es unificar la forma en la que la compañía se comunica con sus consumidores en las distintas etapas del ciclo de venta.
- **Mayor productividad:** mejora la productividad de los gestores, además esto hace que aumenten las ventas, puesto que se puede realizar el seguimiento de todas sus potenciales ventas de manera rápida y personalizada.

## 2.3. TIPOS DE CRM

En la actualidad, existen varios tipos de CRM, entre los que destacan:

1. CRM Colaborativo.
2. CRM Operacional.
3. CRM Analítico.

A continuación, se detallará cada uno de los tipos:

- 1. CRM Colaborativo:** se encarga de las interacciones entre cliente y empresa. Las interacciones se realizan por múltiples canales de comunicación, como puede ser el email o teléfono. El objetivo de este tipo de CRM es el establecimiento de la interacción con el cliente y la comunicación multicanal entre los distintos departamentos de la empresa. Básicamente se centra en mejorar las relaciones de los clientes y la empresa, además permite centralizar y organizar todos los datos obtenidos a través de los canales usados.

Entre las características más importantes del CRM colaborativo destacamos:

- El cliente podrá formar parte en el proceso de elaboración de los servicios o productos a adquirir.
  - Permite crear una interacción que va más allá del intercambio de información. Hace posible configurar el pedido de acuerdo con las preferencias individuales del cliente.
  - Facilita la interacción a través de múltiples canales de contacto, ofreciendo versatilidad y comodidad.
  - Facilita la inclusión de nuevos canales de contacto, que puedan surgir con los continuos avances de la tecnología.
  - Integración con el CRM operacional y analítico.
- 2. CRM Operacional:** se centra en los procesos de negocio de la compañía, es decir, trata de automatizar las tareas relacionadas con las ventas, el marketing y el servicio al cliente.

En este tipo de CRM se diferencian dos partes:

- El Back Office, lo forman los procesos de la organización que no entran en contacto con el cliente de forma directa.
- El Front Office, lo forman procesos de la organización que entran en contacto directo con el cliente de forma directa.

En conclusión, se trata de homogeneizar y coordinar los procedimientos de trabajo de cada departamento y gestionar las tareas que entran en contacto directo con el cliente.

Entre las funciones que realiza, se pueden destacar:

- Interacción con los clientes para conocer sus necesidades más inmediatas, ganándose así su confianza y consiguiendo su fidelización.
- Gestionar las áreas de ventas, atención al cliente y marketing.
- Obtener la máxima información de los clientes, de esta forma se les podrá prestar un mejor servicio y más acorde a sus necesidades.

**3. CRM analítico:** es aquel que permite analizar y evaluar el comportamiento de los clientes a lo largo del tiempo. De esta manera, las organizaciones podrán identificar las oportunidades de venta, negocio o creación de nuevos servicios o productos. Toda esta información servirá a la hora de tomar las decisiones y de identificar las estratégicas comerciales y de marketing más rentables para la compañía.

Entre las funciones del CRM analítico destacan:

- Facilita la elaboración de informes detallados.
- Establece nuevas propuestas.
- Incorpora puntos de atención para el mejor seguimiento y optimización de las campañas.

## 2.4. ESTRUCTURA DEL CRM

Todos los CRM presentan una estructura modular para poder soportar los procesos de negocio que se realizan en las empresas y tener la información lo más organizada posible. Dependiendo del tipo de CRM que se utilice, este podrá tener más o menos módulos.

Los principales módulos que constituyen un CRM se muestran a continuación.



### **Módulo de servicios**

Con el cual centralizaremos las necesidades y preferencias que demanden los clientes. Este módulo permitirá consultar la información del cliente, el tipo de cliente, así como el estado de los productos o servicios contratados.

Conocer tanto las necesidades como las preferencias del cliente permite:

- Ofrecer al cliente un servicio personalizado y de calidad.
- Mejorar la eficiencia de la gestión de incidencias
- Realizar un seguimiento automático de las fechas y el detalle de la renovación del contrato. Asegurándonos así la fidelidad.

Este módulo se encarga de la optimización los servicios de asistencia técnica y atención al cliente, por lo tanto, se registran el historial de compras, las llamadas realizadas, las intervenciones, devoluciones y reclamaciones en curso y tramitadas de cada cliente.

### **Módulo de ventas**

El objetivo de este módulo es optimizar el funcionamiento del departamento comercial. Este módulo permitirá definir los productos, tarifas y ofertas y realizar la previsión de los análisis de venta para cada cliente.

Con este módulo se podrá realizar el seguimiento de todos los pasos que se siguen para realizar la contratación de un producto o servicio, desde los primeros trámites hasta los servicios de postventa.

Además, facilita la gestión y configuración de los pedidos, logística de distribución, preparación de entregas y transporte de las mismas.

### **Módulo de marketing**

Este módulo facilita y automatiza el acceso a la información sobre los perfiles de los clientes. Se podrá conocer que buscan, en que momento lo buscan y de qué manera puedes comunicarte con ellos. Toda esta información ayudara a implementar programas de fidelización de clientes.

Desde este módulo se podrán elaborar y lanzar campañas, mandar mensajes y ofertas a listas de clientes determinados, crear planes de marketing y administrar flujos de trabajo y eventos. Se realizará una segmentación de registros que servirá para el lanzamiento de campañas personalizadas al cliente.



### **Módulo de servicio al cliente**

Uno de los módulos más importantes de los que debe disponer un CRM. Se encarga de proporcionar la información necesaria a los clientes, de resolver incidencias y de relacionarse con los clientes mediante los distintos medios de comunicación existentes. La finalidad de este módulo es fidelizar nuevos clientes.

### **Módulo de facturación**

Con el que se puedan gestionar todos los datos de la facturación, contabilidad e ingresos. Constituye uno de los módulos esenciales para el CRM. Este módulo proporciona las aplicaciones y herramientas financieras necesarias para realizar la contabilidad asociada a la empresa

### **Módulo de cuadro de mandos**

Mostrará de forma resumida la información estratégica de la situación actual de la empresa. Permite elaborar informes periódicos, se podrán obtener datos de la actividad diaria de la empresa, controlar el nivel de satisfacción de los clientes, identificar las prioridades de gestión, analizar la evolución de las ventas, realizar previsiones de la evolución del negocio y analizar de manera global la situación de la empresa en todo momento.

## **2.5. PROCESO DE IMPLANTACIÓN DE UN CRM**

La implantación de un CRM es un proceso largo y complejo. Para poder implantar un CRM en una empresa se deberán seguir una serie de pasos fundamentales. No existe una metodología universal para realizar la implementación de un CRM. A continuación, mostraremos un guion en que se visualizaran todas y cada una de las etapas que se necesitaran realizar.

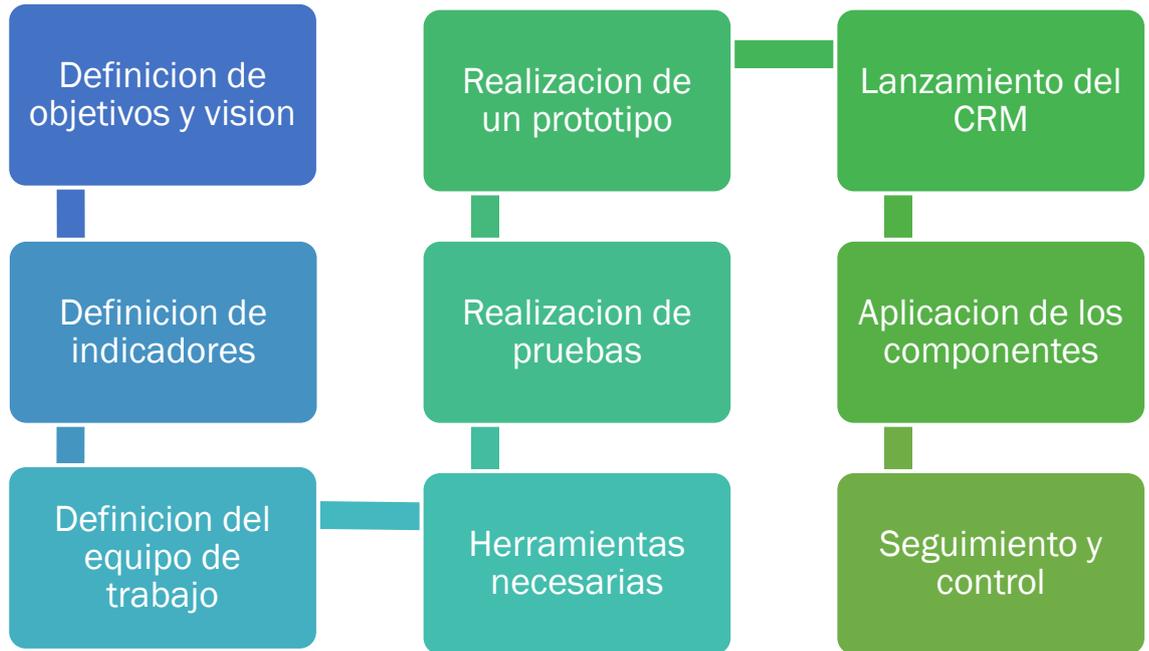


Figura 2. Etapas de implantación de un CRM.

## 1. Definición de objetivos y visión

Lo primero que se ha de realizar es definir una visión, también es importante definir los objetivos globales para posteriormente poder profundizar y basar las estrategias de la empresa en esos objetivos previamente definidos y así poder realizar un seguimiento de los mismos.

Es esta primera etapa es fundamental que la empresa tenga una noción de su situación actual, es decir, es necesario que la empresa se conozca a sí misma y sepa cuáles son sus necesidades más importantes. Para determinar dichas necesidades es imprescindible realizar un análisis inicial para conocer tanto las debilidades como las fortalezas de la empresa, y en relación a esto desarrollar el CRM.

## 2. Definición de indicadores

Se determinará la forma en la que se evaluarán los resultados, para ello definiremos una serie de indicadores. Los indicadores son muy importantes, sobretodo en la etapa de seguimiento y control, ya que nos servirán para conocer si estamos cumpliendo los objetivos.

### **3. Definición del equipo de trabajo**

En esta etapa se formará al equipo encargado de realizar el proyecto. Es una etapa fundamental debido a que los resultados finales dependerán del desempeño del líder del proyecto y la disposición del equipo de trabajo.

Los miembros de todo el equipo deben tener las habilidades adecuadas para desenvolverse durante el proyecto y ser conscientes de cuáles son sus roles y deberes, así como los objetivos a seguir.

### **4. Herramientas necesarias**

En esta etapa se determinará que herramientas son necesarias para la implementación del CRM en la empresa o si por el contrario es factible o no la compra de una solución CRM.

Se deberán analizar los requerimientos para la creación de una nueva BBDD y toda la infraestructura tecnológica necesaria para lograr un CRM exitoso.

### **5. Realización de pruebas**

Aquí, se deberá realizar una prueba de la implementación del CRM. Se comprobará si funciona correctamente y si da una correcta capacitación a los empleados para que realicen su trabajo de forma adecuada.

Una vez realizada esta prueba, se deberá analizar cuáles son los ajustes necesarios para asegurar el éxito del CRM.

### **6. Realización de un prototipo**

En esta etapa se deberá implementar el mismo procedimiento, pero ahora con la información real de los clientes.

### **7. Lanzamiento del CRM**

Una vez que se ha realizado el prototipo con la información real, es la hora de implementarlo en la empresa.

Es una de las etapas más destacables, en ella se comenzará a realizar interacciones con clientes y se deberá estar preparado para registrar la información necesaria en la BBDD.



## 8. Aplicaciones de los componentes

En la mayoría de los casos, en esta etapa, será necesario modificar la estructura organizativa y los procesos para conseguir una empresa centrada en el cliente.

Se necesitará redefinir los procesos para mejorar su eficacia y eficiencia y se les dará mayor prioridad a aquellos que tengan un mayor impacto.

A continuación, se definirán los componentes del CRM. Estos son: identificar, diferenciar, interactuar y personalizar.

- a) **Identificar:** tiene como objetivo que la empresa pueda identificar a los clientes. Las herramientas más utilizadas para realizar dicha identificación son:
  - Call Centers
  - Help Desk
  - Service Desk.
- b) **Diferenciar:** en esta fase, se orientará el comportamiento de la empresa a la atención de los clientes. Mediante la utilización de la BBDD se procesará la información de cada uno de los clientes identificados en la primera fase. Se deberá estimar y calcular el valor que tiene cada cliente para la empresa. Una vez establecido el valor de los clientes, habrá que clasificarlos para luego poder diferenciarlos, por lo que requieren de la empresa.
- c) **Interactuar:** en esta fase, se pretende que la empresa se relacione con los mejores clientes.
- d) **Personalizar:** se tratará de otorgar a aquellos clientes clave los beneficios que desean. En esta fase es importante explotar la información acumulada del cliente para personalizar los servicios.

## 9. Seguimiento y control

En esta última etapa se hará uso de los indicadores que se definieron en la segunda. Estos indicadores nos permitirán llevar un control de los resultados y en base a los mismos poder tomar decisiones que apoyen el cumplimiento de los objetivos.



Sin embargo, la implantación de un CRM conlleva algunos inconvenientes que hay que tener en cuenta:

- Falta de apoyo de la dirección de la empresa.
- No tener claro el proceso de ventas.
- No definir métricas.
- Dejar en manos del departamento tecnológico la configuración y parametrización del CRM.
- Ser demasiado ambiciosos.
- No contar con un plan de formación.
- No contar con una BBDD de calidad.



# CAPÍTULO 3 – SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO



### 3.1. SISTEMAS GESTORES DE BASES DE DATOS

Un sistema gestor de bases de datos (SGBD) es un sistema que permite la creación, gestión y administración de BBDD, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible.

Con un SGBD se podrá acceder a los datos que se tengan almacenados de los clientes, crear relaciones sólidas entre los datos, introducir información nueva, modificar o eliminar la existente y realizar búsquedas de datos. Además, será necesario que los SGBD proporcionen seguridad e integridad de datos.

En la actualidad, existen multitud de SGBD y pueden ser clasificados según la forma en la que administren sus datos. Tenemos:

- 1) **SGBD Relacionales:** no es uno de los gestores más eficientes, pero si uno de los más populares debido a su facilidad y sencillez de uso. Este modelo se basa en establecer relaciones o vínculos entre los datos. Usa tablas como estructura fundamental y su lenguaje de trabajo en el SQL.

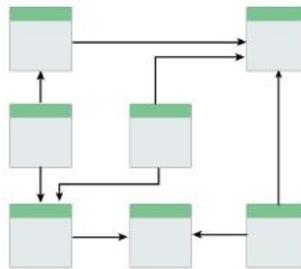


Figura 3. Sistema Gestor de Base de Datos relacional

- 2) **SGBD en red:** gestor muy flexible pero poco utilizado dado que la búsqueda de datos es bastante compleja ya que el sistema debe recorrer todo el conjunto. Estructura y relaciona los datos usando nodos y enlaces. Y es capaz de reflejar cualquier tipo de relación.

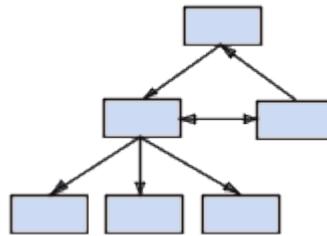


Figura 4. Sistema Gestor de Base de Datos en red

- 3) **SGBD jerárquico:** es el más antiguo y en la actualidad se encuentra totalmente obsoleto por no poder reflejar relaciones complejas, sus relaciones son de tipo 1: N. Relaciona los datos utilizando una estructura en forma de árbol, es decir, relaciones padre-hijo. Utiliza segmentos y arcos.

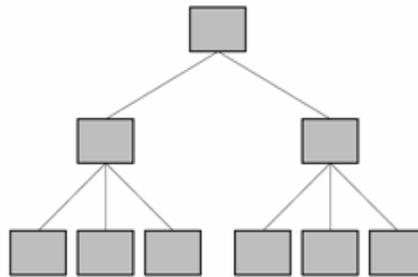


Figura 5. Sistema Gestor de Base de Datos jerárquico

- 4) **SGBD orientado a objetos:** Basados en la programación orientada a objetos, donde el dato y el atributo funcionan como una sola unidad. Se suelen utilizar para la realización de aplicaciones web y multimedia. Son fáciles de asociar con lenguajes modernos de programación como Java, C# o C++.

Para determinar que SGBD se utilizara, nos basaremos en las siguientes premisas, que sea fácil de utilizar, visualización, seguridad, funcionalidad y disponibilidad de servicios de soporte y desarrollo.

Viendo las descripciones que se han realizado más arriba determinaremos que utilizaremos un SGBD relacional debido a su facilidad de uso y por sus flexibilidad en las relaciones (1:N, N:1, M:N). Según M<sup>a</sup> Jesús Ramos, Alicia Ramos y Fernando Montero en su libro “Sistemas Gestores de BBDD” determinar que el modelo de datos

relacional es el principal a la hora de realizar aplicaciones de procesamiento de datos.

Una vez que se ha determinado que modelo utilizar, se realizara una descripción de los principales SGBD relacionales más utilizadas del mercado.

### 3.1.1. MySQL



Figura 6. Logo MySQL

Es un SGBD desarrollado bajo licencia dual y es considerada como la BBDD de código abierto más popular del mercado. Es capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. Es multiusuario y multihilo y se basa en un lenguaje de consulta estructurado.

Sus principales características:

- Funciona en muchas plataformas diferentes.
- Utiliza el diseño de servidores multicapa con módulos independientes.
- Proporciona motores de almacenamiento transaccional y no transaccional.
- Está diseñado para que sea relativamente fácil introducir nuevos motores de almacenamiento. Lo que resulta muy útil si se necesita una interfaz SQL para una BBDD interna.
- Utiliza un sistema de asignación de memoria basado en subprocesos.
- Permite dar soporte a grandes BBDD.
- Implementa funciones SQL utilizando una biblioteca de clases altamente optimizada.
- Proporciona el servidor como un programa separado para su uso en un entorno de cliente/servidor y como una biblioteca que se puede vincular en aplicaciones independientes. Estas aplicaciones se podrán utilizar de forma aislada o en entornos donde no hay red disponible.
- Soporta distintos tipos de datos.

Por otro lado, posee una gran velocidad al realizar operaciones, lo que lo hace uno de los gestores con mejor rendimiento. Tiene un bajo coste de consumo por lo que puede ser utilizado en máquinas con escasos recursos, facilidad de configuración e instalación y control de acceso. La probabilidad de corromper datos es muy baja y su conectividad.

Frente a todas estas ventajas, nos encontramos que MySQL no es muy intuitivo y un gran porcentaje de sus utilidades no están documentadas.

### 3.1.2. ORACLE



*Figura 7. Logo Oracle*

SGBD cuyo objetivo es conectar todos los niveles de tecnología empresarial para así ayudar a los clientes a acceder a conocimiento que necesitan para responder con velocidad y agilidad ante los requisitos del mercado.

Oracle es el motor de BBDD relacional más utilizado a nivel mundial, puede ejecutarse en múltiples plataformas y soporta todas las funciones que se esperan de un servidor. Cuenta con un lenguaje de programación, PL/SQL, que permite implementar diseños mediante triggers y procedimientos almacenados con una integridad referencial muy potente.

Utiliza un entorno cliente-servidor, permite gestionar grandes BBDD, proporciona un alto rendimiento en las transacciones y tiene control de acceso y seguridad. Además, es una herramienta intuitiva y fácil de utilizar.

Oracle, a pesar de tener todas las características mencionadas con anterioridad presenta algunas limitaciones, entre las que nos encontramos:

- Tiene un elevado coste, sus licencias son caras.
- Necesidad de ajustes. No basta con instalarlo en un servidor.
- Escasa formación y la que existe en bastante cara.

- Dificultad de implementar procesamientos recursivos, que es un tipo de función en el cual uno de los procedimientos vuelve a ejecutar el programa entero.
- Incompatibilidad y complejidad de lenguaje.
- Presenta una funcionalidad muy limitada.

### 3.1.3. SQLite



Es una librería compacta y autocontenida de código abierto. Distribuida bajo dominio público que implementa un SGBD de SQL embebido, es decir, no hay servicio intermediario.

*Figura 8. Logo SQLite*

Con SQLite, el proceso que quiere acceder a la BBDD lee y escribe directamente sobre el disco, lo que la hace autónoma y portable. A diferencia de MySQL, Oracle o SQL Server que se encuentran implementados como un servicio.

Es una de las únicas BBDD sin servidor que además permite el acceso de múltiples aplicaciones a la misma BBDD. Es transaccional, lo que significa que los cambios y consultas realizadas son atómicas, consistentes, aislados y durables. Todos los cambios de una transacción se realizan completamente o no se hacen.

También permite registros de longitud variable, es decir, si tenemos un campo con una longitud determinada, SQLite no asigna los bytes correspondientes de espacio fijo en el disco, sino que asigna bytes en relación a la información real que contenga el campo.

SQLite presenta las siguientes características:

- No hay que instalar, configurar, iniciar o mantener el servicio.
- Tiene una pequeña memoria y una única biblioteca necesaria para acceder a la BBDD.
- Realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.
- Se ejecuta en muchas plataformas y sus BBDD pueden ser fácilmente portadas sin ninguna configuración o administración.

- Cuenta con diferentes interfaces, entre las que destacan C++, PHP, Pearl, Python, Ruby entre otros.
- Es de código abierto, y por tanto se puede utilizar sin costo alguno.

Sin embargo, se desaconseja su utilización para aplicaciones de cliente- servidor en BBDD demasiado grandes y en situaciones de alta concurrencia. Al no tener servidor, los datos no están protegidos como en otros gestores que ofrece el mercado. Además, no permite las claves foráneas.

### 3.1.4. POSTGRESQL

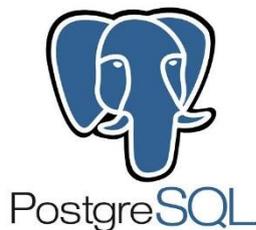


Figura 9. Logo PostgreSQL

SGBD relacional, orientado a objetos, multiplataforma y de código libre. Es una BBDD 100% ACID, soporta distintos tipos de datos, incluyendo herencias entre tablas. Posee Multi Version Concurrency Control (MVCC), lo que permite las mejoras de las operaciones de bloqueo y transacciones en sistemas multiusuario.

Basa su administración en usuarios y privilegios, además soporta vistas, claves foráneas, disparadores, procedimientos almacenados, subconsultas y copias de seguridad en caliente. Su acceso en encriptado y adaptable a las necesidades del cliente.

PostgreSQL es ampliamente popular, ideal para tecnologías web, fácil de administrar y su sintaxis es estándar y fácil de aprender. Tiene capacidad de replicación de datos, está diseñado para ambientes de grandes volúmenes y es extensible, estable y confiable. Además, permite la implementación de rollback, subconsultas y transacciones, lo que hace que su funcionamiento sea mucho más eficaz y ofrece soluciones que MySQL no podría dar.

En comparación con MySQL, PostgreSQL es entre dos y tres veces más lento a la hora de realizar inserciones y actualizaciones.

No soporta tablespaces con los que definir donde se deben almacenar las BBDD, los índices, los esquemas, etc. Tiene un soporte

de orientación a objetos poco completo, escaso soporte en línea y consume más recursos que MySQL. Por último, la sintaxis de sus comandos o sentencias no es nada intuitiva.

### 3.1.5. SQL Server



Es un SGBD desarrollado por Microsoft, su principal función es almacenar y consultar datos solicitados por otras aplicaciones sin importar donde se encuentren.

Figura 10. Logo SQL Server

Es una herramienta gratuita dependiendo de la versión utilizada, pero no es de código abierto. Permite a los usuarios crear sus propios tipos de datos, así como sincronizar datos entre una base embebida y una base SQL Server de manera casi trivial. Posee mecanismos de replicación nativa y una gran comprensión de datos, lo que permite que estos se almacenen de una forma más eficiente.

También hace validaciones de tipos y tiene más cantidad de tipos de datos que SQLite. Presenta formas de respaldo y recuperación de BBDD, soporta information-schema y concurrencia muy eficiente en el uso de plataformas “cloud - ready”.

Como inconvenientes, SQL Server se bloquea cuando se realizan operaciones de escritura, no soporta operaciones en memoria, no es multiplataforma, solo funciona en Windows. No soporta disparadores, vistas, control de excepciones, right joins ni transacciones en cadena.

A continuación, se mostrará un cuadro resumen, en el que se incluirán las características más importantes y se compara con los SGBD descritos con anterioridad. De esta manera, se elegirá que sistema vamos a utilizar para nuestro CRM.

	ORACLE	MYSQL	SQL SERVER	SQLITE	POSTGRES QL
LENGUAJES SOPORTADOS	PL/SQL,PHP, Java,.Net, XML	C,C++,Pascal ,SQL,PHP	C#,.Net, Java, Python ,PHP	SQL, Java,Perl,PHP	PL/SQL,C,C ++,Java, Python
MULTIPLAT	SI	SI	NO	SI	SI

AFORMA	Propietario	Libre	Propietario	Libre	Propietario
TIPO LICENCIA	Propietario	Libre	Propietario	Libre	Propietario
SEGURIDAD	SI	SI	NO	SI	SI
DISPARADORES	SI	SI	NO	SI	SI
REPLICACIÓN	SI	SI	SI	NO	SI
COSTO GRATUITO	NO	SI	NO	SI	NO
TAMAÑO DE LA BBDD	Grande	Grande	Grande	Pequeña	Grande

Tabla 1. Cuadro comparativo SGBD

Debido a que este nuevo CRM va a ser creado para poder ser utilizado por cualquier empresa que ofrezca un servicio de comunicación, este CRM tiene que poder ser ejecutado en cualquier plataforma, por lo que descartaremos SQL Server. A mayores debemos tener una BBDD que permita utilizar grandes volúmenes de datos, se puedan realizar replicaciones, de coste gratuito, para poder reducir los costes de la empresa que utilice nuestro CRM y que tenga seguridad para cumplir con la ley de protección de datos de los clientes. Por lo tanto, no utilizaremos Oracle, SQLite y PostgreSQL. La única opción que nos queda y la que utilizaremos es MySQL.

MySQL nos ofrece robustez, siendo actualmente uno de los grandes motores existentes en el mercado. Posee una fuerte protección de datos, lo que nos permitirá cumplir con las vigentes leyes de protección de datos, y un elevado rendimiento ante grandes volúmenes de datos.

### 3.2. HERRAMIENTAS DE ENTORNO DE DESARROLLO INTEGRADO

Se necesitarán herramientas que permitan diseñar el software de nuestro CRM. Actualmente todo software cuenta con una interfaz gráfica, esta es la

que hará visible la información que se desarrolle. Es decir, es un programa que hace las veces de intermediario entre usuario y máquina.

Muestra de forma visual todas las acciones e información disponible en una plataforma para que los usuarios puedan interactuar con mayor facilidad sin necesidad de disponer de conocimientos en informática.

Para el desarrollo de esta interfaz se necesitará una BBDD y de plataformas IDE. Una plataforma IDE permite escribir código, hacer pruebas y depurar código, pero además proporciona la posibilidad de diseñar una interfaz gráfica para implementar el CRM.

Mediante la utilización de ventanas, botones y un esquema muy visual podemos tener un “boceto” de CRM. Algunas de las plataformas/herramientas IDE más demandadas en el mercado son Eclipse, IntelliJ IDEA y Netbeans.

### 3.2.1. ECLIPSE



Figura 11. Logo Eclipse

Es un entorno de desarrollo integrado, de código abierto y multiplataforma. Normalmente se utiliza para desarrollar lo que se conoce como aplicaciones de cliente enriquecido.

Además, es una potente plataforma de programación, desarrollo y compilación de elementos tan variados como programas en lenguajes de programación como C++ o aplicaciones java.

Entre las tareas que se pueden realizar con Eclipse, destacan las siguientes:

- **Compilar y Ejecutar:** La compilación es algo que se realiza de forma automática y en un segundo plano. Se realiza nada más guardar los cambios realizados en el código.

Cuando se ha finalizado la compilación, Eclipse permite realizar la ejecución del código.



- Depurar (Debug): Muchas veces el código programado no funciona como se esperaba, por eso será necesario realizar una depuración del código.

La depuración consiste en ejecutar paso a paso el código que se ha desarrollado. Está formada por varias vistas.

- Vista editor, en ella se muestra el programa que se está depurando. Sobre el código se va marcando las líneas con una flecha azul. La línea en la que aparece la flecha, es la que se ejecutara a continuación.
- Vista Debug, es donde se controla la ejecución del programa.
- Vista de inspección, nos permite ver los valores de las variables, puntos de ruptura o expresiones.
- Vista consola, es donde se encuentran redireccionadas las entradas y salidas durante la ejecución del programa.

A continuación, se describen las utilidades que nos proporciona Eclipse y que facilitan la tarea de programación.

- Corrector de errores: Eclipse nos permite la detección de errores de compilación de una forma rápida y sencilla, cuando este se produce subraya en rojo el fragmento de código donde se encuentra el error. Si en lugar de error lo que se produce es un aviso, la línea será de color amarillo.

A parte de señalar el fragmento de código donde se produce el error o el aviso, también lo marca con un icono, el cual mostrará un menú desplegable con las posibles soluciones para los errores que se han detectado.

- Mecanismos para formatear código: Eclipse nos ofrece la posibilidad de completar código cuando se deja de escribir, se muestran todos los términos que empiecen por los caracteres que se han tecleado.
- Formatear código: Eclipse dispone de una herramienta para darle un formato adecuado al código de forma automática, según unos criterios preestablecidos, ya que este tiene que ser legible, ordenado y claro.

### 3.2.2. NETBEANS



Figura 12. Logo Netbeans

Entorno de desarrollo integrado libre, desarrollado en Java. Al igual que Eclipse, Netbeans está pensado para que pueda ser utilizado como lenguaje de programación.

Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes software denominados “módulos”. Un “modulo” es un archivo java que contiene clases escritas para interactuar con las APIs de Netbeans y un archivo especial que lo identifica como modulo.

Netbeans basa su desarrollo en proyectos, además tiene asistentes para la configuración y selección de frameworks.

También permite que sus aplicaciones estén en un servicio de carpetas, lo que facilita el trabajo colaborativo. En esta gestión es esencial el control de versiones por eso esta herramienta puede integrarse con SVN, Git y Mercurial.

Tiene integrado un depurador de errores, este permite entre otras cosas, monitorizar el tiempo real de los valores, de las propiedades y variables. Alerta de los posibles errores, lo que permite optimizar las aplicaciones tanto en velocidad como en uso de memoria, facilitando así la escalabilidad de las operaciones.

También cuenta con acceso a BBDD y plugins, lo que permite conectarse a distintos gestores de BBDD, así como consultar sus datos y tablas.

Cuenta con una propia interfaz de usuario para java, HTML5, PHP y C/C++. Presenta unas herramientas muy sencillas, estas son de soltar y arrastrar, lo que la hace una IDE muy intuitiva.

Es más que un propio editor de texto ya que permite refactorizar código, proporciona plantillas, consejos y genera código. No solo soporta java, sino que también C, C++, XML, HTML, PHP, Groovy entre otros.

### 3.2.3. INTELLIJ IDEA

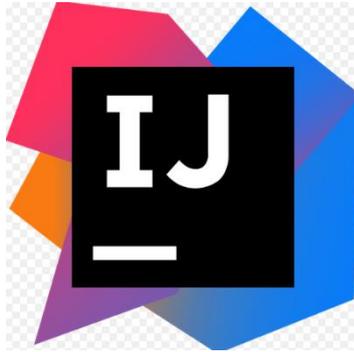


Figura 13. Logo IntelliJ IDEA

Al igual que Netbeans y Eclipse, IntelliJ IDEA es un entorno de desarrollo integrado desarrollado en Java y que dispone de dos ediciones, una es de código abierto mientras que la otra es comercial.

Entre sus características encontramos:

- **Completado de código inteligente:** a diferencia del completado de código básico que hace sugerencias de nombres de clases, métodos, campos y palabras clave. El completado de código inteligente solo sugiere aquellos tipos que se esperan en el contexto sobre el que se produce.  
A mayores proporciona ayuda de configuración inteligente para una gran variedad de lenguajes de programación, entre los que podemos destacar SQL, HTML y JavaScript.
- **Aumento de productividad:** Tiene soporte para Maven y Gradle, además de un control de versiones compatible con SVN, Git y GitHub posee herramientas integradas para pruebas unitarias y de cobertura.
- **Análisis de código:** resalta las advertencias y errores inmediatamente, permitiendo así aplicar una solución de forma más rápida.
- **Amplia variedad de lenguajes soportados:** soporta varios lenguajes de programación como pueden ser Java, Scala, Groovy y Kotlin
- **Herramientas integradas de Android:** posee un diseñador de interfaces de usuario que permite arrastrar y soltar elementos, apoyo a distintos diseños y tipos de pantalla.

Permite la integración con Logcat y posee filtros de búsqueda personalizada.

Por último realizaremos un cuadro comparativo a partir del cual seleccionaremos la herramienta a utilizar.

	ECLIPSE	NETBEANS	INTELLIJ IDEA
CORRECTOR DE ERRORES	SI	SI	SI
DEPURADOR	SI	SI	NO
COMPLETADO DE CÓDIGO	SI	NO	NO
MULTIPLATAFORMA	SI	SI	SI
SOPORTE DE OTROS LENGUAJES QUE NO SEAN JAVA	SI	SI	SI
INTEGRACIÓN CON SVN	SI	SI	SI

Tabla 2. Cuadro comparativo de entornos de desarrollo integrado

Elegiremos como plataforma IDE la herramienta Eclipse, pues esta presenta todas las funcionalidades presentes en la tabla comparativa. Se considera necesario que una plataforma IDE ofrezca la opción de depurar código y de autocompletado. La primera nos permite realizar un paso a paso del código que se ha desarrollado mostrando en distintas vistas el valor que toman las variables, el punto de ejecución del programa y las entradas y salidas.

Por otro lado, la segunda nos muestra los términos que empiezan por las características que se han empezado a teclear.

### 3.3. GESTORES DE PROCESOS DE NEGOCIOS

Un gestor de procesos de negocios, también comúnmente llamados BPMs, se basa en principios, métodos, técnicas y herramientas que permiten diseñar, rediseñar, analizar, ejecutar y monitorizar los procesos de negocio. Es decir, estas herramientas son utilizadas para mejorar los procesos de negocio de una determinada organización mediante la automatización de dichos procesos.

Su utilización permite a la organización que los utilice asegurarse unos resultados constantes y obtener ventajas de oportunidad y mejora como pueden ser la reducción de tiempos, costes y errores.

Todos los BMP del mercado se caracterizan por tener los mismos componentes. Dispondrán de:

1. **Herramientas de Business Process Modeling**, para modelar los procesos de negocio en un formato de flujo de trabajo se utilizará una notación gráfica estandarizada llamada BPMN.
2. **Motor de reglas de negocio**, donde se parametrizarán los enrutamientos complejos. Se realizarán fuera del desarrollo del propio proceso.
3. **Simulador de procesos de negocio**, que sirven para medir la eficiencia del proceso. Lo que se traduce como como el tiempo que se tarda en completar su flujo y su coste.
4. **Herramientas de desarrollo de procesos de negocio**, son las herramientas que los desarrolladores utilizaran para modelar los procesos.
5. **Motor de procesos de negocio**, componente que entiende el BPMN y ejecuta los procesos.
6. **Interfaz web**.

El BPM presenta el siguiente ciclo de vida, para garantizar el correcto funcionamiento de los procesos, se deberán seguir cada una de sus seis etapas.



Figura 14. Ciclo de vida del BMP

1. **Identificación**: Consiste en alinearse con la estrategia de negocio de la empresa y marcar sus objetivos. Una vez definidos, se

determinarán y definirán aquellos procesos que sean necesarios ejecutar para su consecución.

2. Análisis: se analizarán y definirán todos los ámbitos de los proyectos para así poder realizar unos procesos fiables, concisos y entendibles,
3. Desarrollo: una vez finalizado el análisis, se modelarán con un lenguaje de modelado y entendible.
4. Ejecución: finalizada la modelación, el BMP procederá a realizar su ejecución.
5. Monitorización: el BMP dará indicadores de desempeño que permitirá saber si los procesos están alineados con los objetivos de los proyectos.
6. Optimización: en base a los resultados obtenidos en el paso anterior o debido a las nuevas necesidades de la organización se podrán realizar mejoras sobre los procesos existentes.

Por último, se presentarán tres de los BPMs que nos ofrece el mercado y que desde mi punto de vista destacan sobre el resto. Estos son jBPM, Activiti y Bonita BPM.

### 3.3.1. jBPM



Figura 15. Logo jBPM

Es considerado por la mayoría de los desarrolladores como el padre de los BPMs de código abierto. Se trata de un motor escrito en java que utiliza una máquina virtual de procesos para poder soportar los múltiples lenguajes de programación de forma nativa.

Esta herramienta nos permite modelar, ejecutar y monitorizar procesos y casos de negocio a lo largo de su ciclo de vida. Para posteriormente llevar a cabo el despliegue y gestión de cada uno de ellos. Además, permite dar forma a los objetivos al describir detalladamente cada uno de los pasos que deben ejecutarse y el orden en el que deben hacerlo. Dicha ejecución viene determinada por el diagrama de flujo que se debe elaborar para cada proceso a ejecutar.

jBPM presenta dos versiones:

- a) Gratuita, que permite realizar trabajos de desarrollo y no se puede utilizar para usos en producción.

- b) De pago, que ofrece soporte y autorización para desplegar la herramienta en producción.

Presenta las siguientes funcionalidades:

- Motor de flujo de trabajo ligero, lo que permite ejecutar procesos de negocio.
- Capacidad de administración permitiendo la admisión de casos de usos flexibles y adaptables.
- Permite:
  - Modelar e implementar, es decir, crear procesos, reglas y modelos de datos.
  - Ejecución de procesos, tareas, reglas y eventos.
  - Administrar el tiempo de ejecución.
  - Visualización de informes, lo que permite realizar el seguimiento de la ejecución de los procesos creados.
- Gestionar las instancias de los procesos.
- Modelar datos y formularios.
- Diseñar procesos para admitir la creación gráfica y simulación de los procesos de negocio.

jBPM también presenta ciertas limitaciones entre las que destacamos las siguientes.

- Es una sofisticada plataforma para diseñar y desarrollar flujos de trabajo, por lo que requiere que los procesos sean realizados por usuarios y desarrolladores experimentados.
- No posee una herramienta de diseño para cada tarea de los procesos. Para su personalización es necesario que sea integrado con Eclipse.
- Está orientada al desarrollo de procesos, dejando la implementación de estos para el desarrollador.

### 3.3.2. BonitaBPM

Figura 16. Logo BonitaBPM

Es la herramienta más extendida en el mercado debido a que tiene una licencia de código abierto, y proporciona a sus usuarios una gran experiencia de inicio a fin ya que permite crear aplicaciones

basadas en procesos sin tener la necesidad de escribir ni una sola línea de código.

BonitaBPM es con diferencia la solución BPM con más cantidad de desarrollos de terceros. Su ecosistema es tal, que es imposible encontrar un módulo que conecte con una herramienta de su suite, lo que lo convierte en una solución muy flexible a la hora de realizar desarrollos rápidos.

Al igual que jBPM, esta herramienta también tiene dos versiones, pero con funcionalidades totalmente distinta a la anterior.

- a) Community Edition: versión gratuita y que no presenta demasiadas funcionalidades.
- b) Subscription Edition: la cual contiene soporte profesional global, herramientas de personalización de interfaces, interfaces específicas para móviles y tablets, cuadros de mando, monitorización y posibilidad de clusterización así como balanceo de carga.

Bonita BPM presenta las siguientes ventajas:

- Se eliminan pasos innecesarios debido a la automatización de los procesos.
- Interfaz muy intuitiva, lo que facilita el diseño de los procesos.
- Las reglas y acciones son muy visuales y fácilmente de configurar, permitiendo a los procesos ser automatizados sin la personalización de los programadores.
- Comunicación con clases java.

Entre todas estas ventajas hay que destacar que esta herramienta crea el código de forma demasiado automática, lo que significa que el desarrollador no tiene ningún control sobre el código que se escribe. A mayores es prácticamente imposible modificarlo fuera de su entorno y herramientas.

Entre sus limitaciones encontramos:

- Presenta un motor de BPM propio, el cual solo puede ejecutar los procesos de modelado por su propia herramienta de diseño.
- El monitoreo de procesos contiene demasiada información, por lo que no es entendible por cualquier tipo de usuario.

- No se puede extender la personalización de los formularios. Además, es una herramienta poco intuitiva.

Su mayor limitación, es la poca integración con otros lenguajes de programación, por poner un ejemplo, el lenguaje java esta menos desarrollado en este BPM que en el resto.

### 3.3.3. Activiti



Figura 17. Logo Activiti

Herramienta escrita en java y que utiliza una nueva notación estándar para desarrollar los procesos, BPMN2. Su ciclo de vida y lanzamientos están muy ligados al jBPM debido a que ambos pertenecen a la compañía Alfresco. Lo que los hace muy similares.

Frente a BonitaBPM, Activiti tiene la limitación de que no existen una gran variedad de módulos de integración de software con terceros. Por el contrario, al estar escrito y ser utilizado principalmente en java, se puede implementar cualquier funcionalidad que al desarrollador se le ocurra y tener el control absoluto sobre ella.

Activiti presenta las siguientes características:

- Fácil implantación de procesos, así como para trabajar con las diferentes herramientas y componentes.
- Es un motor ligero, fácil de utilizar para desarrolladores acostumbrados a programar en java.
- Presenta una serie de elementos que facilitan la colaboración entre los diferentes roles involucrado, estos son Activiti Engine, Activiti Explorer, Activiti Modeler y Activiti Designer.
  - a) Activiti Engine: es el encargado de recibir los modelos BPM, interpretarlos y desplegarlos en el entorno. Es el corazón del motor, el cual puede tener múltiples BPM ejecutándose y que cada uno de ellos despierte múltiples procesos.
  - b) Activiti Explorer: herramienta web para explorar el estado del motor, procesos en ejecución, tareas...
  - c) Activiti Modeler: herramienta web de modelado no técnico.

- d) Activiti Designer: herramienta de modelado técnico, es un plugin de Eclipse.

A continuación, y tras la presentación de estas tres herramientas, se mostrará un cuadro comparativo a partir del cual realizaremos la elección de nuestra herramienta a utilizar.

	BONITABPM	JBPM	ACTIVITI
MOTOR BPM PROPIO	SI	SI	SI
EDITOR DE PROCESOS	SI	SI	SI
INTEGRACIÓN CON TERCEROS	SI	SI	NO
INTEGRACIÓN NATIVA CON JAVA	NO	SI	SI
VERSIÓN GRATUITA	NO	SI	SI
USO DE BPMN2	SI	SI	SI

Tabla 3. Cuadro comparativo de gestores de procesos de negocio

Como se puede apreciar en la tabla, el BPM elegido es jBPM, ya que este cumple todas las necesidades que nuestro CRM demanda.



# CAPÍTULO 4 – SELECCIÓN DE LAS HERRAMIENTAS DE GESTIÓN



## 4.1. GESTORES DE CONTROL DE VERSIONES

A los gestores de control de versiones también se les llama sistemas de control de versiones (SCV). Un SCV es un software que controla y organiza las distintas revisiones que se realicen en los archivos de código fuente.

Podemos decir, que una revisión es un cambio realizado en el código fuente, ya sea añadir más código, modificarlo o comentarlo.

Actualmente existen dos tipos de SCV, los sistemas centralizados y distribuidos. A continuación, se realizará una breve descripción de cada uno de ellos.

1. **Sistemas Centralizados:** en este tipo de sistemas de control de versiones, todos los ficheros y sus versiones estarán almacenados en un único directorio de un servidor.  
Los desarrolladores que trabajen con estos archivos deberán descargarse del directorio a su local una copia, de esta manera podrán añadir o modificar los archivos de código fuente.  
Es decir, un sistema de control de versiones centralizado funciona según el modelo clásico de cliente – servidor.

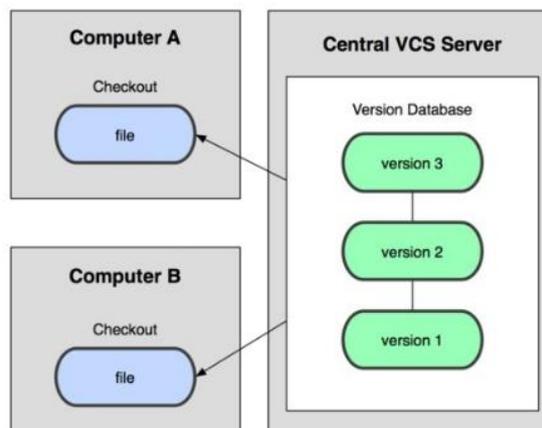


Figura 18. Sistema de Control de Versiones centralizado

Una vez terminado y subido el código al repositorio, este estará disponible para el resto de desarrolladores. Será el servidor el que se encargará de fusionar los cambios, de resolver problemas e informar al desarrollador de los errores que se hayan podido producir. Los más conocidos son CVS y SVN.

2. **Sistemas Distribuidos:** este tipo de sistemas sigue el modelo punto a punto. Con ellos solo se mantiene una copia local del repositorio, pero cada equipo de desarrolladores se convierte en un repositorio del resto de usuarios. Esto permite una fácil ramificación y un manejo más flexible del repositorio, es decir, en lugar de que cada desarrollador tenga una copia de trabajo del servidor, la copia de trabajo de cada desarrollador es un repositorio en sí mismo, una rama nueva del proyecto central.

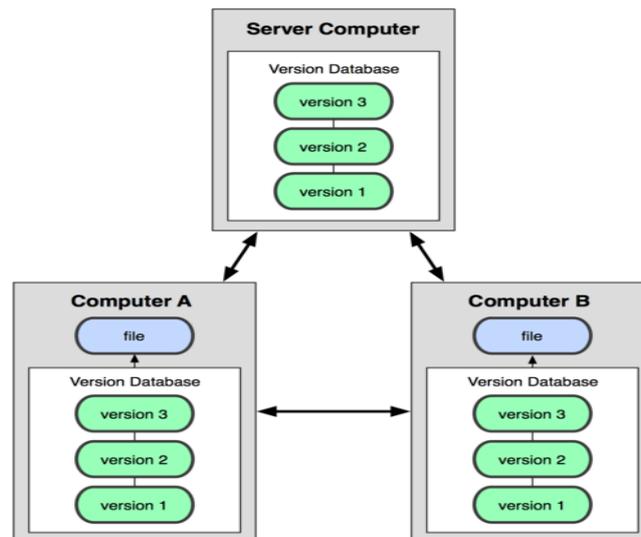


Figura 19. Sistema de Control de Versiones distribuido

Los más conocidos son Git y Mercurial.

Por lo tanto, los sistemas de control de versiones se encargan de:

- Centralizar todo el repositorio de código de la aplicación, proporcionando accesibilidad en todo momento.
- Permite el trabajo simultáneo entre desarrolladores sobre un mismo elemento sin pérdida de cambios.
- Llevar la trazabilidad de quien y cuando se realizan los cambios en el código.
- Controlar el versionado de las distintas entregas de la aplicación.
- Etiquetar cada uno de los cambios aplicando comentarios sobre la modificación.
- Permitir distintas líneas de trabajo en paralelo sobre la misma aplicación.

Como puede deducirse, hay muchas razones por las que es necesario utilizar una herramienta automática de control de versiones. A continuación, se mencionarán algunas de ellas:

- Con ellas se podrá seguir la historia y evolución de los proyectos. Para cada cambio, se tendrá un registro de quien hizo, porque lo hizo, cuando se hizo y cuál fue el cambio realizado.
- Al trabajar con más personas, una herramienta de control de versiones facilita más la colaboración entre ellas.
- Permite la recuperación de archivos, por ejemplo, si un desarrollador ha realizado un cambio en el que existe un error de código, estas herramientas nos permiten obtener la versión anterior de este archivo. Un buen software de control de versiones ayudara a descubrir cuando se ha introducido el archivo con el problema y quien lo ha hecho.

A continuación, se realizará una breve descripción de los distintos sistemas de control de versiones que se han mencionado en este punto.

#### 4.1.1.SUBVERSION (SVN)

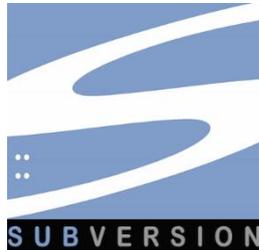


Figura 20. Logo  
Subversion (SVN)

También conocido como SVN, es un repositorio de control de versiones de código fuente. Es de tipo centralizado. Se utiliza para guardar, recuperar, registrar, identificar y mezclar versiones de archivos.

Estos son organizados en forma de árbol dentro de un repositorio central, el cual recuerda todos los cambios y acciones que se han realizado sobre los archivos. De esta manera, se permite la recuperación de antiguas versiones de datos o revisar los cambios realizados sobre los mismos.

SVN presenta las siguientes funcionalidades:

- Cada proyecto dispone de un repositorio central donde se almacena todo el código actualizado por los desarrolladores y la información de cuándo y quien se realizan.
- Permite el trabajo por parte del desarrollador de forma desconectada, ya que descarga el código en un entorno local para su modificación.



- Carga de los cambios desde un entorno local hacia el repositorio, controlando posibles conflictos de actualizaciones y proponiendo formas de resolverlos.
- Asigna automáticamente una nueva versión en cada carga de cambios al repositorio, permitiendo la introducción de comentarios.
- Su organización en árbol permite la división del desarrollo en diferentes líneas de trabajo independientes (evolutivos, correctivos, preventivos...), identificando cada una de estas líneas de trabajo con una versión del producto.
- Mediante fusión controlada, permite la unificación de cambios entre líneas de trabajo.
- La integración con diversas herramientas de la arquitectura de desarrollo permite mejorar la trazabilidad de las tareas, así como los cambios asociados a estas.

Esta herramienta presenta muchas características por lo que nos centraremos en las que considero más relevantes.<sup>11</sup>

- **Versionado de directorios:** a diferencia de CVS que solo lleva el historial de ficheros individuales, SVN implementa un sistema de ficheros versionado que sigue los cambios que se realicen sobre los arboles de directorios a través del tiempo.
- **Verdadero historial de versiones:** Dado que CVS está limitado al versionado de ficheros, operaciones como copiar y renombrar no son soportadas por CVS. Adicionalmente, en CVS no puede reemplazar un fichero versionado con algo nuevo que lleve el mismo nombre sin que el nuevo elemento herede el historial del fichero antiguo. Con SVN se puede añadir, borrar, copiar, y renombrar ficheros y directorios. Y cada fichero nuevo añadido comienza con un historial nuevo, limpio y completamente suyo.
- **Envíos atómicos:** un conjunto cualquiera de modificaciones se sube por completo al repositorio o no se sube nada. Esto permite a los desarrolladores construir y enviar los cambios como fragmentos lógicos e impiden que ocurran problemas cuando solo una parte de los cambios subidos se realiza con éxito.
- La creación de ramas y etiquetas se realiza de forma más eficiente si se compara con otros CVS.

---

<sup>11</sup> COLLINS-SUSSMAN, B, MICHAEL PILALTO, C y W. FITZPATRICK, B. (2004). *Version Control with Subversion*. O'Reilly Media



- **Permite el bloqueo de archivos**, se utiliza en archivos binarios, que al no poder fusionarse fácilmente conviene que no sean editados por más de una persona a la vez.

Pero sin ninguna duda la característica más importante de esta herramienta es que, a diferencia de otras herramientas de control de versiones, los archivos versionados no tienen cada uno un número de versión independiente, en cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un instante determinado.

Todas estas ventajas son contrarrestadas con algunas limitaciones. Entre ellas, encontramos que los archivos que se versionan no tienen un número independiente de revisión, sino que todo el repositorio tiene un número de versión.

Además, requiere de bastante espacio, presenta problemas con el borrado, codificación de acentos y con los cambios de nombres de archivos y directorios. Y los metadatos del control de versiones pueden corromperse con facilidad por errores del usuario.

Por último, se expondrán las herramientas con la que se puede integrar SVN. Entre ellas se encuentran:

- I. **Eclipse, Entenas e IntelliJ IDEA**: mediante integración vía plugin. Permite la obtención del código fuente del SVN para cualquiera de sus ramas.
- II. **Jenkins**: integración nativa. Detección de nuevo código fuente para la ejecución automática del proceso de integración continua.
- III. **Jira**: integración nativa, en la que cada acción sobre el código fuente queda registrada en el seguimiento de una tarea asociada. El mecanismo que tiene Jira para integrar los cambios es mediante los comentarios que se introducen al realizar una operación de “commit” en SVN.

### 4.1.2.GIT



Figura 21. Logo Git

Es uno de los SCV distribuido más utilizado. Fue creado para que cumpliera cuatro requisitos: que no fuera parecido a CVS, que fuera un sistema de control de versiones distribuido, que tuviera una gran seguridad frente a la corrupción y un gran rendimiento en las operaciones.

En otras palabras, es un SCV destinado al mantenimiento eficiente y confiable de las versiones cuando estas tienen una gran cantidad de archivos de código fuente.

Entre sus características encontramos las siguientes:

- Es el líder de repositorios de código libre.
- Centralización de todo el repositorio proporcionando accesibilidad en todo momento.
- Trabajo simultáneo entre desarrolladores sobre un mismo elemento sin pérdida de cambios.
- Modelo distribuido de repositorios, que permite trabajar sin necesidad de una conexión constante a internet. Cada usuario tiene una copia completa del historio del repositorio en su disco local y se conecta solo para subir o descargar código.
- Control de versiones de código fuente, asignación de etiquetas para cada subida de nuevo código en el repositorio.
- Permite diversas líneas de trabajo en paralelo sobre la misma aplicación, lo que se conoce como “ramas o branches”.
- Detección automática de conflictos.
- Histórico de cambios, permite saber quién y cuándo ha realizado un determinado cambio.
- Capacidad para modificar el histórico de cambios para conseguir un mejor rendimiento.

Git proporciona un gran rendimiento en programas grandes, las búsquedas son más eficaces, lo que supone una gran rapidez para detectar diferencias entre archivos. Tiene un diseño sencillo, gran velocidad en las operaciones y un gran apoyo al desarrollo no lineal. Además, presenta un sistema de ramas muy flexible permitiendo crear ramas locales.

Por el contrario, tiene una curva de aprendizaje muy lenta comparado con el resto de repositorios que hemos descrito y posee una gran

cantidad de comandos, en concreto posee 152 comandos, lo que hace muy difícil recordar su significado y utilización.

#### 4.1.3.CVS



Figura 22. Logo CVS

Sistema de control de versiones centralizado con el que se puede mantener un registro de todo el historial de trabajo y cambios realizados sobre archivos que formen parte de un proyecto.

Es de gran ayuda cuando se trabaja con un grupo de personas, cada persona trabaja sobre su propia copia que anteriormente ha descargado en su local y CVS se encarga de

fusionar el trabajo, resolviendo posibles conflictos. Es considerado el padre de los sistemas de control de versiones.

Principales características de CVS:

- Al ser un sistema de control de versiones centralizado utiliza una arquitectura cliente – servidor. El cliente y el servidor pueden estar en la misma máquina.
- Permite descargar copias al mismo tiempo. Y una vez realizadas las modificaciones el servidor trata de aceptar las diferentes versiones. Si esto falla, el servidor deniega la versión errónea e informa al usuario del conflicto. En el caso contrario, CVS incrementa las versiones y almacena la información, incluyendo una breve descripción, la fecha y el nombre del autor.
- Permite solicitar un histórico completo de los cambios realizados, así como comparar diferentes versiones de un mismo archivo.
- Puede mantener distintas ramas de un proyecto.
- Es multiplataforma
- Se puede utilizar en un modo de líneas de comandos o a través de la interfaz gráfica del usuario.
- Envía archivos completos, cuando se hacer una modificación, se sube al repositorio el archivo completo en lugar de solo los cambios.
- Es muy bueno para minimizar el espacio de almacenamiento.

- Puede admitir proyectos de pequeño y mediano tamaño, además puede configurarse y extenderse para ser compatible con la mayoría de proyectos.

A diferencia de otros repositorios, en CVS tanto los archivos como los directorios no pueden ser renombrados, en caso de querer renombrar un archivo, este deberá ser subido al repositorio con otro nombre y luego eliminar el otro. Tiene un soporte limitado a UNICODE, debido al problema con los caracteres raros. Por último mencionar que este repositorio se encuentra actualmente en desuso.

#### 4.1.4. MERCURIAL



Figura 23. Logo Mercurial

Es un sistema de control de versiones distribuido y multiplataforma. Está destinado a ser utilizado en grandes proyectos por su velocidad operacional. Inicialmente fue creado para competir con Git. Está implementado en Python, pero algunas instancias utilizan C.

Es un servidor de código abierto que permite de manera sencilla realizar modificaciones. Su curva de aprendizaje no es muy alta debido a que sus comandos son muy parecidos a los de SVN y no cuenta con muchos.

Tiene un rendimiento alto en cuanto a velocidad de ejecución de comandos y operaciones, y bajo en la gestión de espacio en disco. Es fácil de usar, además posee un conjunto de comandos muy similares a los de SVN por lo que la interfaz es fácil de aprender y usar, pero difícil de romper. A parte, es muy flexible y extensible, su funcionalidad se puede aumentar mediante la activación de diferentes extensiones.

Por el contrario, la fusión de ramas que tiene Mercurial es bastante limitada debido a que no permite a dos ramas funcionar, lo que lo diferencia de Git.

Por último, realizaremos un cuadro comparativo para determinar la herramienta que se va a utilizar.

	SVN	GIT	MERCURIAL	CVS
<b>SERVICIOS</b>	Centralizado	Distribuido	Distribuido	Centralizado
<b>CURVA APRENDIZAJE</b>	Corta	Algo más elevada	Corta	Corta
<b>VELOCIDAD REPOSITORIO</b>	Normal	Alta	Normal	Baja
<b>ACCESO WEB</b>	No	Si	Si	Si
<b>TAMAÑO</b>	Alto	Reducido	Alto	Bajo
<b>RESTRICCIONES DE COMMIT</b>	Si	Si	Si	No
<b>POSIBLES CONFLICTOS</b>	En repositorio remoto	En repositorio local	En repositorio local	En repositorio remoto
<b>ESTADO</b>	Activo	Activo	Activo	Mantenimiento

Tabla 4. Cuadro comparativo de sistemas de control de versiones

Para determinar que repositorio elegimos, utilizaremos el siguiente criterio. Primero determinaremos si queremos utilizar un repositorio centralizado o distribuido para saber cuál elegir nos basaremos en sus principales diferencias.

Las principales diferencias entre un sistema centralizado y distribuido:

- No existe una copia de referencia de código, solo de trabajo.
- Las operaciones suelen ser más rápidas al no tener que comunicarse con un servidor central.
- Cada copia de trabajo es un respaldo del código base.
- Hay que hacer una operación de “update” antes de la operación de “commit”, puesto que se trabaja sobre una copia local.
- La creación y fusión de ramas es más fácil, debido a que cada desarrollador tiene su propia rama.

Debido a las diferencias que presenta uno respecto al otro, se utilizara un repositorio que sea centralizado, en este punto ya descartamos Git y Mercurial.

Una vez realizados estos descartes la opción a elegir es bastante sencilla, debido a que CVS se encuentra actualmente en desuso como se ha mencionado con anterioridad, la herramienta ganadora es SVN.

Con SVN podremos agrupar todos los archivos en un solo lugar, se podrá trabajar con numerosos archivos binarios y de gran tamaño.

## 4.2. GESTORES DE PRUEBAS

Realizar desarrollos libres de defectos es realmente difícil. Actualmente no existen métodos formales que se puedan aplicar para demostrar que no existen defectos en los desarrollos realizados.

Una de las mejores formas para tener un grado razonable de certeza de que el software desarrollado se comporta de la forma esperada, es probar su funcionamiento en determinadas circunstancias. A estas ejecuciones de funcionamiento se les denomina pruebas o test.

Los objetivos de estas pruebas son verificar que el software funciona como fue diseñado, evaluar las suposiciones realizadas en las especificaciones de requisitos y diseño mediante la realización de una demostración concreta y encontrar errores en el software para que estos se puedan corregir antes de que el software llegue a entornos productivos.

La realización de pruebas no garantiza una ausencia total de defectos en el código, puesto que es imposible probar los infinitos usos.

En todo desarrollo se deben de realizar estos tipos de pruebas:

1. Pruebas de sistema: estas pruebas permiten verificar que el software se comporta como se esperaba.
2. Pruebas unitarias: pruebas creadas por el programador para asegurarse de que los desarrollos realizados hagan realmente lo que se quiere que hagan.
3. Pruebas de integración: pruebas que permiten verificar que la comunicación entre los subsistemas sea la correcta.
4. Pruebas de aceptación: este tipo de pruebas las realiza el usuario y su cometido es validar que el sistema cumpla con los objetivos para los que fue diseñado.

A continuación, se realiza una pequeña descripción de dos de los gestores de pruebas que proporciona el mercado.

#### 4.2.1. TESTLINK



Figura 24. Logo TestLink

Es una herramienta web de gestión de pruebas de software, que permite la gestión, ejecución y reporte de casos de prueba dentro de un proyecto o servicio que se esté desarrollando. Dicha plataforma web permite una fácil integración con una amplia variedad de sistemas de gestión de defectos y fallos.

Dado que TestLink es un software basado en web, para su utilización se necesitará disponer de un acceso a un servidor web y a una BBDD. Tiene soporte para utilizar como BBDD tanto MySQL como PostgreSQL.

Esta herramienta permite definir tantos casos de prueba como se necesiten, estos se podrán ejecutar de forma automática o manual. También permite tener distintos tipos de usuarios, estos podrán tener roles distintos. Entre los roles se podrá elegir entre Invitado, diseñador de pruebas, probador, senior tester, líder y administrador, cabe destacar que también se podrán crear roles personalizados.

TestLink cuenta con las siguientes funcionalidades:

- 1) Permite estructurar las pruebas en diferentes niveles y tipología:



- a) Test Case: equivalente a un caso de prueba. Describe una tarea de prueba mediante los pasos que la componen y los resultados esperados. Son la pieza fundamental de un plan de pruebas.
  - b) Test Suite: Estructura los test cases en pruebas lógicas. Se pueden crear varios niveles de Test Suite. El primer nivel corresponde con los niveles de prueba: unitarias, de integración y de aceptación. El segundo nivel se utiliza para los tipos de pruebas del sistema: de regresión y funcionales.
  - c) Test Plan: Agrupación de pruebas creadas para ser ejecutadas en una ocasión concreta.
  - d) Test Project: Define un proyecto.
- 2) Proporciona un histórico de ejecución de las diferentes pruebas, almacenando todos los resultados.
  - 3) Integración de ejecución fallida con bugs en la herramienta de gestión de proyectos o servicios.
  - 4) Gestión de requisitos respecto a su trazabilidad con casos de pruebas y coberturas asociadas. Generación de informes de cobertura de requisitos.
  - 5) Versionado de casos de pruebas, así como priorización, asignación a testers y uso de keywords.
  - 6) Plantilla predefinida con los cuatro niveles de pruebas, unitarias, integración, de sistema y de aceptación.

Se puede integrar con las siguientes herramientas:

- I. Jira y Redmine: asociación de una ejecución de pruebas fallidas con sus correspondientes bugs en Jira y Redmine. Muestran el estado en el que se encuentran. Por ejemplo, por cada prueba que se tenga con un estado KO se creará un defecto o incidencia en estas herramientas.
- II. Fenix y Bugzilla: al marcarse como KO la ejecución de una de las pruebas definidas en el plan de pruebas, se creará automáticamente una incidencia en Fenix y Bugzilla.

### 4.2.2. SELENIUM



Figura 25. Logo Selenium

Herramienta que permite la automatización de pruebas. Permite grabar, editar y depurar casos de prueba que puedan ser ejecutados de forma automática e iterativa. Es multiplataforma, pudiéndose instalar en cualquier sistema operativo.

Además de ser una herramienta para registrar acciones permite editarlas manualmente o crearlas desde cero.

Entre sus principales características nos encontramos con las siguientes:

- Facilidad para cargar los casos de prueba en la herramienta y su posterior ejecución.
- Autocompletado para todos los comandos.
- Las acciones pueden ser ejecutadas paso a paso.
- Herramienta de depuración y puntos de ruptura.
- Los test pueden ser almacenados en diferentes formatos.

El potencial de esta herramienta está en que puede ser utilizado para la grabación de las pruebas funcionales durante las pruebas de regresión. Con este servicio se consigue obtener una batería de pruebas automatizadas que pueden automatizadas que pueden ser utilizadas cuando sea necesario repetirlas.

A continuación, se mostrará un cuadro comparativo de estas dos herramientas.

	TESTLINK	SELENIUM
MULTIPLATAFORMA	SI	SI
INTEGRADO	Bugzilla, Jira, Mantis, Redmine	-
AUTOMATIZACIÓN DE	SI	SI

<b>PRUEBAS</b>		
<b>TIPO DE PRUEBAS</b>	Unitarias, Integración, Aceptación	Unitarias, Integración, Aceptación
<b>PLANES DE PRUEBAS</b>	SI	SI
<b>APRENDIZAJE</b>	Bajo	Medio
<b>AÑADIR ARCHIVOS ADICIONALES</b>	SI	SI

Tabla 5. Cuadro comparativo de gestores de pruebas

Aunque estas herramientas se parecen demasiado utilizaremos como herramienta de pruebas Testlink, el motivo de esta elección es que se puede integrar con otras herramientas de gestión de incidencias y porque su curva de aprendizaje es baja.

### 4.3. HERRAMIENTAS DE DESPLIEGUE

Todos los proyectos deben de seguir un ciclo de vida, una vez que en un proyecto se finalizada la fase de construcción y pruebas, se ha de desplegar el código de forma automática en los entornos de desarrollo, preproducción y producción.

Antes de realizar este despliegue automático se deberá comprobar la calidad del código desarrollado, es decir, comprobar que este compila cualquiera de los entornos en los que se va a desplegar. Para ello, se utilizarán herramientas que realicen una integración continua. La integración continua sigue un flujo de trabajo previamente creado y basado en decisores, de esta forma nos aseguraremos que, si el código presenta algún error de programación, este no se despliegue en ningún entorno.

Más abajo se va a realizar una descripción de dos de las herramientas que hay en el mercado y se dedican a prestar este servicio.

### 4.3.1. JENKINS



Figura 26. Logo Jenkins

Herramienta de código abierto y basada en java. Utilizada para implementar la práctica de integración continua, facilitando la identificación temprana de

problemas y permitiendo automatizar procesos repetitivos.

Además, es de uso gratuito, ya que no existe costo alguno ni para su licencia ni para su descarga y uso.

Posee muchas funcionalidades que la hacen una de las mejores herramientas del mercado, des todas ellas destacamos las siguientes.

- Lanzamiento automático del proceso de integración al detectar nuevo código en los repositorios.
- Automatización del proceso de compilación, ejecución de pruebas de calidad y test.
- Los desarrolladores pueden detectar y solucionar problemas de integración de forma continua.
- Ejecuciones inmediatas de las pruebas de sistema, funcionales y/o no funcionales.
- Monitorización continua de las métricas de calidad del proyecto.
- Ejecución de revisiones estáticas de código orientadas al aseguramiento de la calidad y mantenibilidad del código.
- Dash-board multiproyecto para obtener el estado de las ejecuciones.
- Programación de ejecuciones de procesos de integración continua.
- Detalle de cada ejecución indicando los pasos realizados con éxito y los fallidos.
- Configuración de alertas de correo y notificaciones en otras herramientas para estar al corriente del estado del proceso de integración.

A parte Jenkins se puede integrar con una gran variedad de herramientas, entre las que destacaremos.

- a) Git y SVN, detección de nuevo código fuente para la ejecución automática del proceso de integración continua.
- b) Nexus: utilización de artefactos completos hospedados en esta herramienta. Despliegue y almacenamiento de nuevos artefactos compilados o empaquetados.
- c) Sonarqube: visualización del resultado de la Quality Gate aplicada al código desde la pantalla principal del job, y modificación del resultado del job. Exportación de los resultados para su análisis desde una visión de cuadro de mandos.
- d) GitLab: información sobre la ejecución del proceso desencadenado a partir de una operación de “commit”, así como sobre los resultados obtenidos.
- e) Jira: integración vía plugin para informar de la ejecución de un proceso y de los resultados obtenidos en la tarea asociada.

Jenkins se caracteriza por la utilización de Jobs. Un Job se puede definir como un proceso que automatiza una serie de tareas (como la compilación y ejecución de pruebas) desde una entrada como el código fuente y a partir de la cual genera diferentes salidas.

- Resultados del proceso, resultado que obtenido de la correcta o incorrecta ejecución de todos los pasos a ejecutar por el job.
- Notificaciones, como correos o notificaciones en otras herramientas que permitan de forma activa o reactiva el resultado del proceso.
- Otros outputs, como la generación de reports de resultados, despliegue de paquetes en rutas o servidores de aplicaciones.

Se podrán ejecutar tanto de forma manual como automatizada.

#### 4.3.2. BAMBOO



Figura 27. Logo Bamboo

Es una herramienta creada por la compañía Atlassian. Es, al igual que Jenkins una herramienta de integración continua que permite la realización de pruebas, compilaciones y versionados automatizados en un solo flujo de trabajo.

Con ella se pueden crear planes de compilación de varias fases, configurar triggers para iniciar las compilaciones después de cada operación de commit que se realice en el repositorio con el que la herramienta este integrado y asigna agentes a las compilaciones y despliegues esenciales.

Con cada cambio que se realiza en el código fuente Bamboo realiza pruebas automatizadas que facilitan y agilizan la detección de errores.

Los proyectos de despliegue de los que dispone se encargan de automatizar la tarea de efectuar el lanzamiento de código en todos los entornos disponibles. Además, ofrece control sobre el flujo con permisos propios de cada entorno.

Por otro lado, Bamboo permite la integración con herramientas como Jira, BitBucker y Fisheye.

Entre sus funcionalidades destacan:

- Recursos tecnológicos: funciona bien en todos los idiomas y con otras tecnologías. Permite elegir entre una amplia gama de tareas disponibles tanto para proyectos de compilación como de despliegue.
- Proyectos de despliegue: concede a los despliegues un trato preferente. Un proyecto de despliegue contiene el proyecto de software que se va a desplegar, entre las que se encuentran las versiones ya compiladas y probada y los entornos en las que están se van a desplegar.
- Agentes dedicados: se podrán ejecutar correcciones y compilaciones de forma inmediata. Cuando se dedica un agente a una determinada actividad este no puede ser utilizado por ninguna otra actividad. Gracias a lo cual no existen esperas en la cola para obtener agentes de compilación cuando hay errores críticos que solucionar.
- Visibilidad: consulta los cambios de código y los tickets o incidencias de Jira con las incidencias de despliegue anterior antes de desplegar una nueva versión de estas.

Para determinar que herramienta de las dos descritas con anterioridad y que posteriormente utilizaremos en el presente TFG realizaremos un cuadro comparativo que contendrá las características más importantes relacionadas con este tipo de herramientas.

El criterio de selección vendrá determinado por el número de síes, lo que quiere decir que se seleccionara la herramienta que más síes posea.

	JENKINS	BAMBOO
INTEGRACIÓN CONTINUA	SI	SI
INTEGRACIÓN CON SVN	SI	NO
VERSIÓN GRATUITA	SI	SI
DESPLIEGUE DE CÓDIGO	SI	SI

Tabla 6. Cuadro comparativo de herramientas de despliegue

La herramienta a utilizar será Jenkins, debido a que cumple todos los requisitos y se puede integrar con el repositorio SVN, que hemos elegido con anterioridad.

#### 4.4. GESTORES DE INCIDENCIAS

Toda empresa dedicada a la prestación de un producto y/o servicio a un amplio abanico de clientes puede experimentar errores o sucesos no muy habituales que hacen que su proceso de contratación se vea temporalmente interrumpido. Con la finalidad de mantener la calidad de prestación de los servicios y la fidelidad de sus clientes es necesario que un CRM cuente con un gestor de incidencias que se dedique a prevenir y restaurar lo más pronto posible cualquier interrupción o reducción no planificada en la calidad de prestación del servicio.

Estos gestores de incidencias se encargarán de registrar con el mayor detalle posible los tickets o incidencias creadas por el usuario, así como de asignarles una categoría y prioridad de resolución.

Deberán mantener al usuario informado en todo momento del avance y estado de las incidencias. Por otro lado, permiten evaluar los niveles de satisfacción de los clientes y usuarios de los servicios prestados.

A continuación, se realizará una breve presentación de los gestores de incidencias más populares y conocidos del mercado.

#### 4.4.1. JIRA



Figura 28. Logo Jira

Es una aplicación de web de “issue tracking” para la gestión de tareas, incidencias, riesgos y gestión de proyectos o servicios.

Algunas de las características más destacables son:

- Es una interfaz intuitiva y simple, las solicitudes de incidencias están separadas de las solicitudes de resolución. El cliente solo visualiza la información que necesita.
- Dispone de una herramienta focalizada en la gestión de incidencias.
- Gestión de colas personalizables, lo que permite la organización de trabajo de una forma más eficiente.
- Ofrece workflows predefinidos, aunque también permite la creación de otros nuevos dependiendo del flujo de trabajo que se necesite seguir.
- Permite realizar la gestión mediante metodologías “agile”.
- Informes en tiempo real, lo que permite realizar el seguimiento de las incidencias.
- Automatización de tareas.
- Herramienta válida para cualquier tipo de entorno empresarial.

Jira tiene una gran capacidad de integración con otras herramientas, entre las que destacan:

- I. Eclipse: Se realiza mediante integración vía plugin y permite realizar desarrollos asociados a una tarea en Jira.
- II. SVN: Integración en la que cada acción que se realiza sobre el repositorio queda registrada en su tarea asociada.

- III. Git: Integración entre repositorios e incidencias, de modo que tanto en operaciones de “commit” como de “merge request” se pueden referenciar issues.
- IV. Jenkins: Integración vía plugin. La ejecución del proceso de integración continua queda registrada en el seguimiento de la tarea asociada.
- V. TestLink: Seguimiento de un caso de prueba fallido a través de una issue.
- VI. Confluence: Relación entre la tarea y su documentación e información asociadas en el portal del proyecto o servicio.
- VII. Fénix: Creación y seguimiento de estados incurridos en peticiones enviadas automáticamente desde Jira o Fénix.

Hay que mencionar que, a pesar de tener todas estas ventajas, Jira tiene una curva de aprendizaje algo elevada, su sincronización con otras herramientas resulta algo pobre y los contratos con los clientes suelen ser poco flexibles y estar mal definidos.

#### 4.4.2. MANTIS BUG TRACKER



Figura 29. Logo Mantis Bug Tracker

Es una aplicación de software libre, multiplataforma que permite gestionar las incidencias y errores en proyectos de desarrollo. Es un sistema fácil de usar y adaptable a muchos escenarios. Tiene soporte a BBDD

como a MySQL, MS SQL y PostgreSQL.

Mantis presenta las siguientes funcionalidades:

- Reporte de incidencias: es una aplicación que permite a los usuarios reportar incidencias de cualquier tipo.
- Sistemas de permisos de usuarios: Incorpora un sistema de roles y permisos que permiten identificar a los distintos usuarios que acceden al sistema. Los niveles de usuarios son: espectador, informador, actualizador, desarrollador, manager y administrador. Cada rol determina las acciones que un determinado usuario puede realizar, siendo el rol administrador el que puede realizar todas las acciones que

permite la herramienta e incluso tiene la capacidad de dar permisos al resto de roles.

- Estado de las incidencias: En Mantis las incidencias pueden pasar por diversos estados, siendo estos fácilmente reconocibles, pues cada estado presenta un color diferente.
- Notificaciones al usuario: Permite notificar a los usuarios del estado de las incidencias mediante correo electrónico, de forma que cada vez que editemos una incidencia le llegue un mensaje con los cambios realizados al usuario.
- Personalización: Ofrece al rol administrador mucha flexibilidad a la hora de realizar la configuración de la herramienta. Permite definir nuevos estados, así como la creación de nuevos campos y personalizar los permisos de los distintos roles. También permite la administración de etiquetas y organización de la información de los proyectos. Dentro de estos proyectos se pueden crear subproyectos y categorías.
- Facilidad de uso: es fácil de utilizar. Los formularios de reporte de incidencias son muy completos, pero solo dos campos son obligatorios.

Además, permite la utilización de time tracking, lo que permite medir el tiempo que se tarda en resolver una incidencia, crear informes estadísticos o publicar anuncios globales. Permite la integración con distintos repositorios como pueden ser Git y SVN.

#### 4.4.3. REDMINE



Figura 30. Logo Redmine

Es un gestor y planificador de proyectos, además permite realizar seguimientos de incidencias y errores. Con Redmine se pueden gestionar múltiples proyectos desde una sola interfaz, estos proyectos pueden ser totalmente independientes unos de otros. La navegación es muy sencilla.

Presenta las siguientes funcionalidades:

- Soporte a múltiples proyectos: Gestiona todos los proyectos con una instancia de Redmine. Cada usuario puede tener un rol diferente en cada proyecto. Cada proyecto puede ser declarado público, visible por cualquier persona, o privado,



visible solo por los miembros del proyecto. Los módulos de los que dispone se pueden habilitar o deshabilitar a nivel de proyecto.

- Soporte a múltiples subproyectos: Gestiona las partes del proyecto relacionadas como subproyectos de un proyecto principal.
- Control de acceso flexible basado en roles: Dependiendo del rol de cada usuario, a este se le permitirán realizar ciertas acciones. Cada usuario tiene una página personal, la cual ofrece una vista personalizada de la información de los proyectos en los que se está participando.
- Sistema flexible de seguimiento de problemas: Permite definir nuevos estados y tipos de problemas. Se pueden configurar transacciones de flujo de trabajo para cada tipo de problema y función a través de una interfaz de administración basada en la web. Divide las peticiones en tres tipos: errores, tareas y soportes y estas pueden ser asignadas a un miembro del equipo.
- Diagrama de Gant y calendario: Gant automático y calendario basado en fechas de inicio y vencimiento de problemas.
- Funcionalidad de seguimiento de tiempo: el tiempo puede ser introducido a nivel proyecto. Informe simple para ver el tiempo por usuario, tipo de problema, categoría o actividad.
- Campos personalizados: Pueden definirse campos propios y personalizados para problemas, entradas de tiempos, proyectos y usuarios. Estos campos se pueden mostrar en la lista de problemas usarse como filtros al igual que los campos normales.
- Gestión de noticias, documentos y archivos.
- Navegador del repositorio: los repositorios existentes se pueden integrar a cada uno de los proyectos. Soporta los siguientes repositorios: Git, SVN, Mercurial, Bazaar y CVS.
- Soporte a múltiples BBDD: entre las que destacan MySQL, SQLite y PostgreSQL.

#### 4.4.4. BUGZILLA



Figura 31. Logo Bugzilla

Herramienta que permite a personas o grupos de desarrolladores realizar un seguimiento de errores. Lanzada como software de código abierto, Bugzilla ha sido utilizada por un gran número de organizaciones para utilizarla como herramienta de seguimiento de errores o defectos. Está basada en web. Normalmente, Bugzilla utiliza un servidor HTTP, como Apache y una BBDD, como MySQL para llevar a cabo su trabajo

Bugzilla permite organizar los defectos de múltiples formas, permitiendo así el seguimiento de múltiples productos con diferentes versiones, y a su vez formados con múltiples componentes. Permite categorizar los defectos o incidencias de acuerdo a su prioridad y severidad, así como asignarles versiones para su solución.

También se pueden anexar comentarios, propuestas de solución, designar a los responsables de elaborar una solución y el tipo de solución que se aplicara al defecto. Todo esto llevando un seguimiento de fechas en las que cuando sucede algún evento mencionado, permite enviar mensajes por correo a las personas que están interesadas en el seguimiento del defecto.

Los defectos pueden ser creados por cualquiera y pueden ser asignados a un desarrollador en particular. Cada defecto/incidencia se puede marcar con diferentes prioridades y encontrarse en diferentes estados, así como ir acompañados de notas del usuario.

Bugzilla presenta algunas limitaciones, entre ellas encontramos que carece de integración con otras herramientas, su utilización es bastante compleja y el soporte que se realiza sobre las incidencias no ofrece garantía ninguna.

A continuación, se realizará un cuadro comparativo.



	JIRA	MANTIS	REDMINE	BUGZILLA
CREACIÓN DE NUEVOS WORKFLOWS	SI	NO	SI	NO
NOTIFICACIONES AL USUARIO	SI	SI	SI	SI
FACILIDAD DE USO	SI	SI	SI	NO
INTEGRACIÓN	SI	SI	SI	NO
PERMISOS DE USUARIOS	SI	SI	SI	SI
SOPORTA MYSQL	SI	SI	SI	SI
COLAS PERSONALIZADAS	SI	NO	NO	NO
DIFERENTES ESTADOS PARA LAS INCIDENCIAS	SI	SI	SI	SI

Tabla 7. Cuadro comparativo de gestores de incidencias

El gestor de incidencia que escogeremos será Jira, este presenta todas las características que considero importantes es una herramienta de este tipo. Aunque no es de coste gratuita, su licencia anual es asequible.

# CAPÍTULO 5 - CASO PRÁCTICO



## 5.1. DEFINICIÓN DEL CASO PRÁCTICO

Una empresa de telefonía, Telefonasa, de reciente implantación en el mercado y con una cartera de clientes de bastante volumen, que provienen de una absorción de otra empresa, quiere aumentar los servicios que ofrece y mejorar alguno de ellos. Actualmente ofrece los servicios de internet y telefonía fija y móvil.

En enero del 2020, quiere que Telefonasa ofrezca, además de los actuales, un nuevo servicio de televisión por cable. Pero también quiere añadir a internet una nueva tarifa que proporcione más megas, así como ampliar sus servicios de telefonía móvil. La finalidad del CEO de Telefonasa, es ampliar su cartera de clientes y fidelizar a los existentes ofreciendo sus servicios a un precio inferior que el que tienen sus competidores.

Para ello necesita reducir sus gastos, pero manteniendo la calidad de su servicio, es por lo que contrata a una empresa externa, Externalia, que se encarga de trabajar con el CRM creado en este TFG y prestar servicios de desarrollo de software.

A continuación, se describirá la estructura organizativa que utiliza Externalia en lo que se refiere al desarrollo de software y se utilizarán las herramientas seleccionadas en los capítulos 3 y 4.

## 5.2. ESTRUCTURA DE LA ORGANIZACIÓN

La empresa de nuestro ejemplo contará con una estructura de tres niveles jerárquicos claramente diferenciados: 1 jefe de proyecto, 3 analistas y por último 3 programadores (ver figura 32).

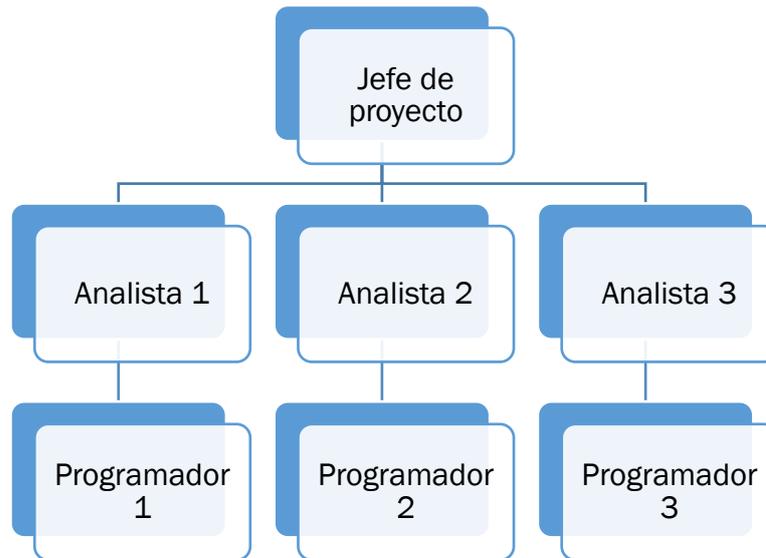


Figura 32. Organización jerárquica

El jefe de proyecto será el encargado de gestionar proyectos. Entre sus responsabilidades destacan, la gestión de equipos compuesto por analistas, programadores o proveedores externos de servicios.

Es el máximo responsable de que los proyectos se realicen en los plazos establecidos con el cliente final, asegurando los estándares de calidad establecidos y dentro del rango de costes determinado. Suelen tener buenas habilidades de negociación.

Entre sus tareas destacan:

- Negociar con el cliente: analiza las necesidades del cliente y las traduce a un proyecto para así crear el software más beneficioso para él. Por otro lado, mantiene informado al cliente de la evolución del proyecto, así de como los posibles retrasos que este pueda sufrir. Además, es el encargado de realizar la formación al cliente una vez entregado el proyecto.
- Gestiona de forma global el proyecto: planifica, define los recursos, coordina y supervisa el análisis funcional, así como el desarrollo e implementación del software. Tiene amplios conocimientos técnicos lo que le ayuda a distribuir el tiempo y recursos. Resuelve los problemas técnicos que puedan surgir a lo largo del proyecto y garantiza la calidad de software que se entrega.
- Planifica el seguimiento: elabora un plan detallado de los plazos a cumplir. Define como estarán distribuidas las tareas entre los miembros del equipo. Prioriza las tareas de mayor importancia y

establece los puntos críticos que pueden surgir durante la fase de diseño y desarrollo.

- Dirige al equipo: coordina y gestiona a los miembros de su equipo.

Los analistas son las personas encargadas del diseño y análisis previo a la creación de un nuevo software. Puede estar involucrado en el diseño y desarrollo, para ello necesita amplios conocimientos de programación de software. El analista es el encargado de definir el trabajo de los programadores, pero debe ser lo suficientemente flexible tanto para programar como para asumir ciertas responsabilidades si el jefe de proyecto así lo exige.

Entre sus tareas destacan:

- Analizar el proyecto: analiza los requerimientos del software así como las necesidades y especificaciones del usuario a la hora de utilizar el software. Realiza un estudio que refleja el tiempo necesario para realizar el desarrollo.
- Desarrollo y diseño: formula el diseño y construye la estructura del software. Desarrolla modelos concisos, claros y coherentes de acuerdo a los requerimientos, que posteriormente pasaran a los programadores.
- Transmite el proyecto a los programadores: los cuales desarrollaran el código del software. Se encargará de solventar los posibles problemas de programación que aparezcan.
- Coordina y desarrolla los procesos de programación, testeo y documentación y controla que los programadores lo realicen correctamente.
- Mantiene relación con otros profesionales, obtiene información del jefe de proyecto sobre las limitación y capacidades del software y le informa sobre la evolución del proyecto.

Por último, los programadores se encargan de crear, testear y mantener los programas de software que desarrollan. Implementa algoritmos mediante un lenguaje de programación que puede entender el ordenador. Es el encargado de escribir las aplicaciones informáticas que nos permiten interactuar con las máquinas y que indican a los ordenadores las instrucciones a seguir para ejecutar sus desarrollos. A parte de desarrollar el software, también se encarga de testear, resolver problemas, encontrar y corregir errores, así como de documentarlos.

Entre sus funciones destacan:

- Desarrollo de programas: realiza desarrollos completos o parte de ellos siguiendo las especificaciones que previamente ha marcado el analista.
- Documenta el software que realiza: se encarga de escribir los manuales de usuario.
- Actualiza, modifica y depura el software: realiza nuevos desarrollos, modifica y actualiza el software o partes de un software existente. Se asegura del correcto funcionamiento y corrige los errores detectados.
- Control de calidad: prueba el software o la parte del software que ha desarrollado. Antes de que el producto llegue al cliente, hace pruebas sobre programas que no han sido creados por él, para evitar que el usuario pueda encontrarse con errores. Propone y realiza las modificaciones de código para evitar los errores que ha detectado.
- Mantiene reuniones con la analista y el jefe del proyecto para entender la arquitectura del producto.

Este equipo está formado por profesionales, con experiencia en el sector, así como conocimiento técnico en los productos sobre los que se están realizando desarrollos. Externalia, cuenta con un amplio conocimiento de los entornos disponibles y en los procedimientos de gestión, con lo que:

- Se asegura el conocimiento reduciendo gaps y posibles no conformidades en el análisis y el diseño.
- La necesidad de soporte del área de IT es mínima, siendo el equipo autónomo.
- Al organizarse un equipo especializado por dominio y ámbito de conocimiento se aseguran los plazos, calidad y la capacidad de coordinar diferentes grupos de trabajo.

### 5.3. UTILIZACIÓN DE MYSQL

Para poder utilizar MySQL como un SGBD, primero será necesario crear una BBDD que se ajuste a nuestro modelo de CRM, por ello habrá que definirla basándose en los requisitos del proyecto, MySQL gestionara los datos introducidos conforme a un modelo de base de datos predefinido.

El segundo paso a seguir será la realización del diseño de dicha BBDD, para ello utilizaremos el diagrama entidad-relación. Este, nos permite la definición de entidades, que se corresponden con las tablas cuyos datos se

utilizaran como elementos de entrada/salida de los ficheros que serán necesarios programar para que el CRM funcione.

Para nuestro ejemplo, crearemos tres BBDD, todas ellas tendrán el mismo diseño y los mismos datos. Se necesitarán:

- Una BBDD de desarrollos.
- Una BBDD pre-productiva.
- Una BBDD productiva.

Cabe destacar que las BBDD creadas están vacías, es decir, no tienen ninguna de las tablas que se muestran en el diagrama entidad – relación (ver figura 33).

Por lo tanto, se deberán crear todas las tablas que hemos definido en el nuestro modelo relacional. Para crearlas solo es necesario ejecutar el comando “Create table <nombre de la tabla>” e indicar los campos por los que están formadas cada una de ellas.

Utilizaremos los siguientes tipos de datos (ver anexo):

- “Varchar”, para los campos de las tablas que incluyan texto.
- “Int”, para los campos de las tablas que sean numéricos.
- “Date”, para los campos de las tablas en las que se necesite introducir una fecha,

Una vez creadas las tablas y relaciones se procederá a la carga masiva de datos. Los datos correspondientes a cada tabla estarán almacenados en ficheros con extensión *txt*. De esta forma, tendremos un fichero diferente para cada tabla.

A continuación, se muestra el diagrama entidad-relación con las cardinalidades y las claves primarias y foráneas con el que crearemos la BBDD. Este diagrama entidad – relación ha sido específicamente creado para el caso práctico que se ha propuesto en este capítulo (ver figura 33).

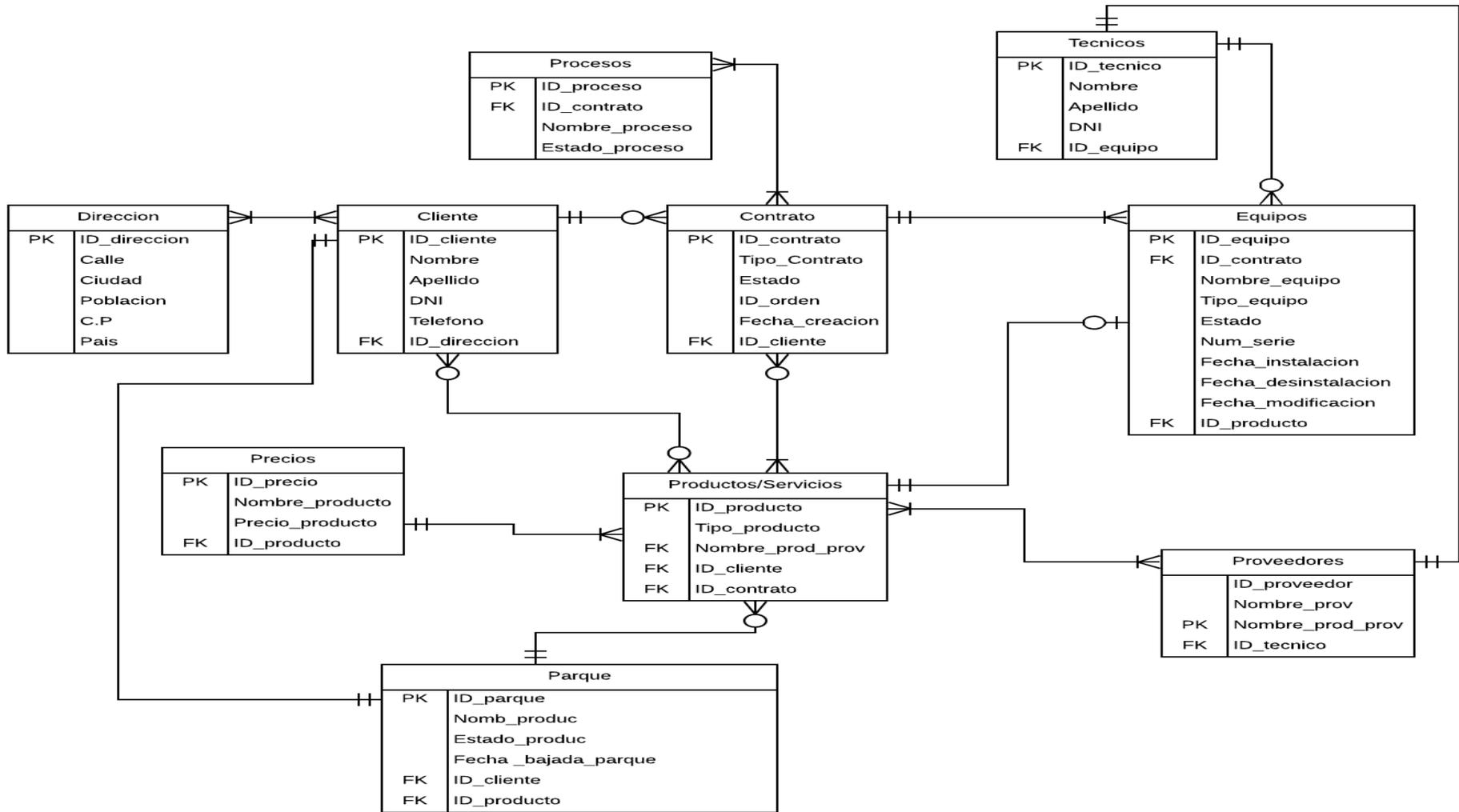


Figura 33. Modelo relacional de BBDD

Por último definiremos los siguientes conceptos:

- **Clave primaria:** Una clave primaria es un atributo o campo que identifica de forma exclusiva cada registro en una determinada tabla. Solo se puede poner una clave primaria por entidad, esta tiene que ser única, inalterable y no puede ser nunca nula.
- **Clave foránea:** Se define como clave foránea como una clave primaria, pero en un lugar foráneo. No tienen por qué ser únicas, puede haber múltiples claves foráneas en una entidad.
- **Cardinalidad:** Definiremos cardinalidad como el número máximo de veces que una instancia en una entidad se puede relacionar con instancias de otra entidad.

El significado de las cardinalidades utilizadas en el diagrama que se muestra a continuación es la siguiente:

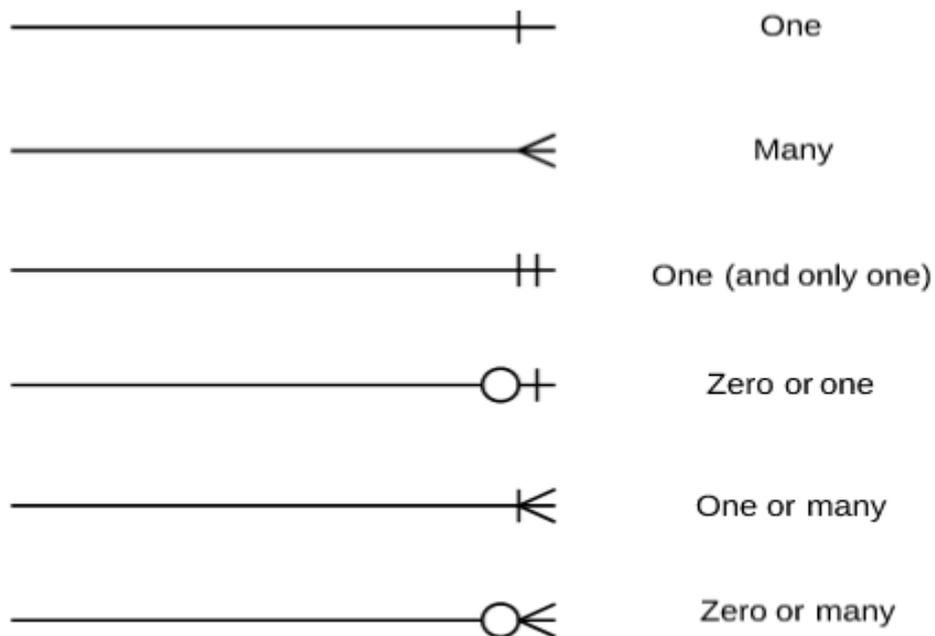


Figura 34. Cardinalidades del modelo relacional

## 5.4. UTILIZACIÓN DE SUBVERSION

Para utilizar el repositorio SVN, cada programador y analista deberá instalar un software cliente que permita la conexión. El software que se va a utilizar es TortoiseSVN.

Una vez realizada dicha instalación, se creará una carpeta la cual se llamará con el nombre del proyecto, en este caso, PROY001. En esta carpeta se dejarán los archivos compartidos, la cual funciona como un espejo de lo que hay en SVN.

Actualmente, en el repositorio tenemos los archivos necesarios para llevar a cabo el proyecto que nos han solicitado. Solo tendríamos que modificarlos para que las funcionalidades exigidas se puedan realizar.

Para utilizar de una forma correcta el repositorio, lo primero que se tiene que realizar es un *update*, esta operación nos permite tener la última versión de todos los archivos que contiene. Seguidamente descargaríamos en nuestro disco local los archivos que necesitemos modificar. Una vez modificados, los volveremos a subir a SVN.

Por último, se realizará un *commit*, de esta manera los cambios realizados serán visibles para el resto de usuarios.

Al realizar el *commit*, se debe escribir una pequeña descripción de los cambios realizados, así el resto de usuarios sabrán cuales son las modificaciones que se han realizado.

De esta manera cuando otro usuario vaya a realizar algo sobre los mismos archivos, solo tiene que ver los comentarios para saber cuáles son los últimos cambios.

Por último, se definirán unas buenas prácticas para el uso de SVN y que serán de obligado cumplimiento para todos los usuarios que tengan acceso a este repositorio.

- Antes de publicar algún archivo, se comprobará que este compila y funciona correctamente.
- Se realizará un *update* sobre el repositorio antes de realizar cualquier tipo de acción sobre él.
- Se realizará siempre una operación de *commit*, que permitirá publicar los cambios que se hayan realizado.
- En los comentarios se deberá describir detalladamente los cambios realizados.

## 5.5. UTILIZACIÓN DE ECLIPSE

Lo primero que realizaremos será crear un nuevo proyecto que llevará por nombre “Telefonasa”. Este proyecto agrupará un conjunto de recursos relacionados entre sí, como código fuente, diagramas de clase o documentación.

A continuación, deberemos configurar el proyecto para ello:

- Necesitaremos crear un subdirectorio para almacenar el código y otro diferente para almacenar las clases compiladas.
- Se deberá indicar, en el caso de que existan, las dependencias con otros proyectos.
- Indicar las librerías que necesitaremos para nuestro proyecto y/o definir las variables del entorno.
- Definir el orden de búsqueda de las clases que se utilicen, principalmente para solucionar posibles conflictos en caso de que existan clases con el mismo nombre.

Dentro del proyecto que hemos creado se crearán tantas clases como sean necesarias para que el usuario pueda contratar los servicios de internet y telefonía.

## 5.6. UTILIZACIÓN DE jBPM

Para poder empezar a usar jBPM tendremos que descargarnos esta herramienta. jBPM es un plugin de Eclipse, por lo que es necesario la utilización de esta herramienta, así como su implementación con ella.

Una vez realizada la implementación, se podrá empezar a desarrollar y modelar los procesos. Para nuestro caso práctico en concreto necesitaremos modelar cuatro procesos principales. Estos procesos serán:

- **Alta Móvil:** con este proceso se permitirá la contratación de un móvil. Este proceso tendrá comunicación con sistemas externos.  
A este proceso de alta irá asociado el proceso de cancelación, el cual permitirá cancelar en cualquier momento el proceso de contratación de la línea móvil. (Ver figura 35 y 36).
- **Baja móvil:** con este proceso se dará al cliente que lo requiera la baja de la contratación del producto telefonía móvil. En este proceso se liberará el número móvil que el cliente utilizaba. (ver figura 37).



- **Alta internet:** con este proceso se permitirá la contratación de una línea de internet. Al igual que le ocurría al alta móvil, a este proceso también ira asociado el proceso de cancelación. (ver figura 38 y 39).
- **Baja internet:** con este proceso se dará al cliente que lo requiera la baja de la contratación de la línea de internet. (ver figura 40).

### Alta móvil

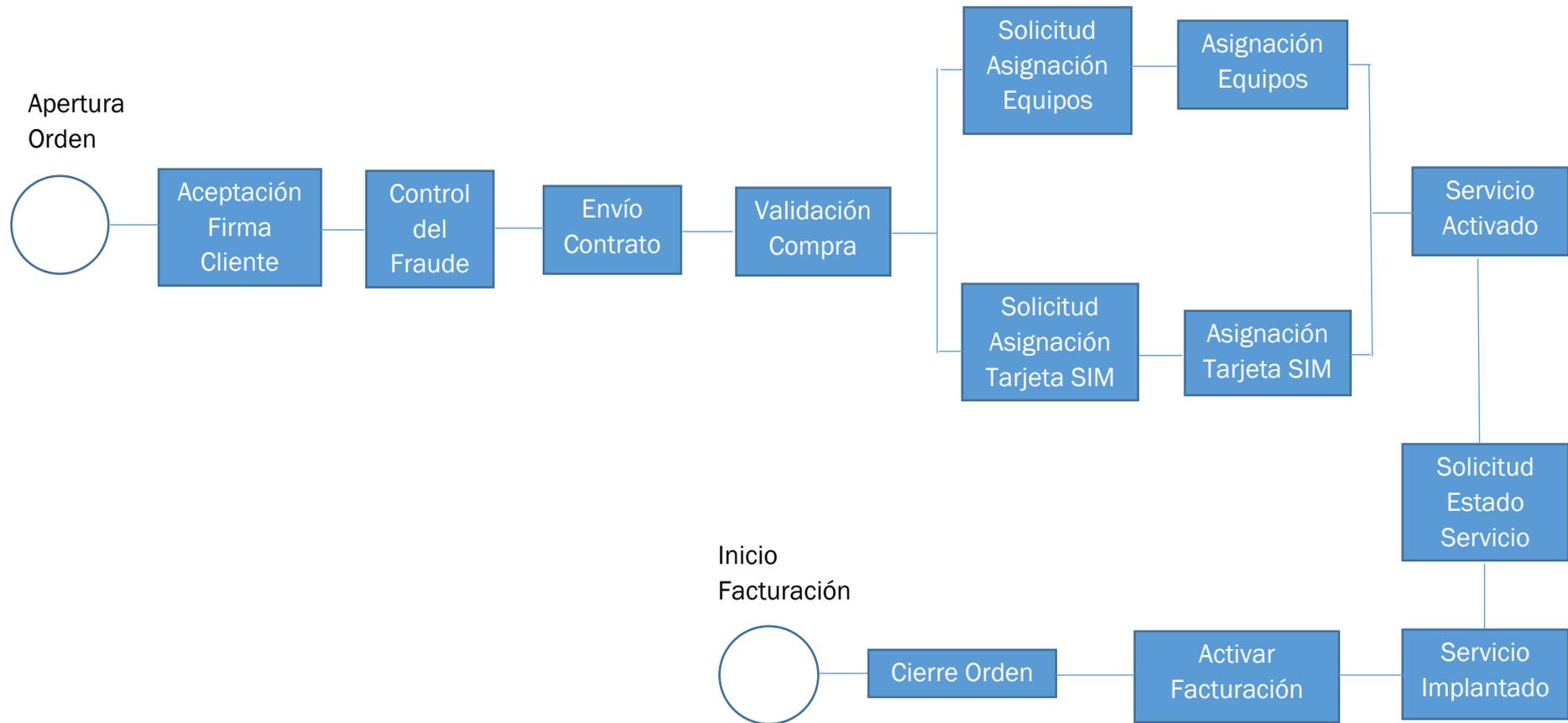


Figura 35. Proceso Alta Móvil

### Cancelación móvil

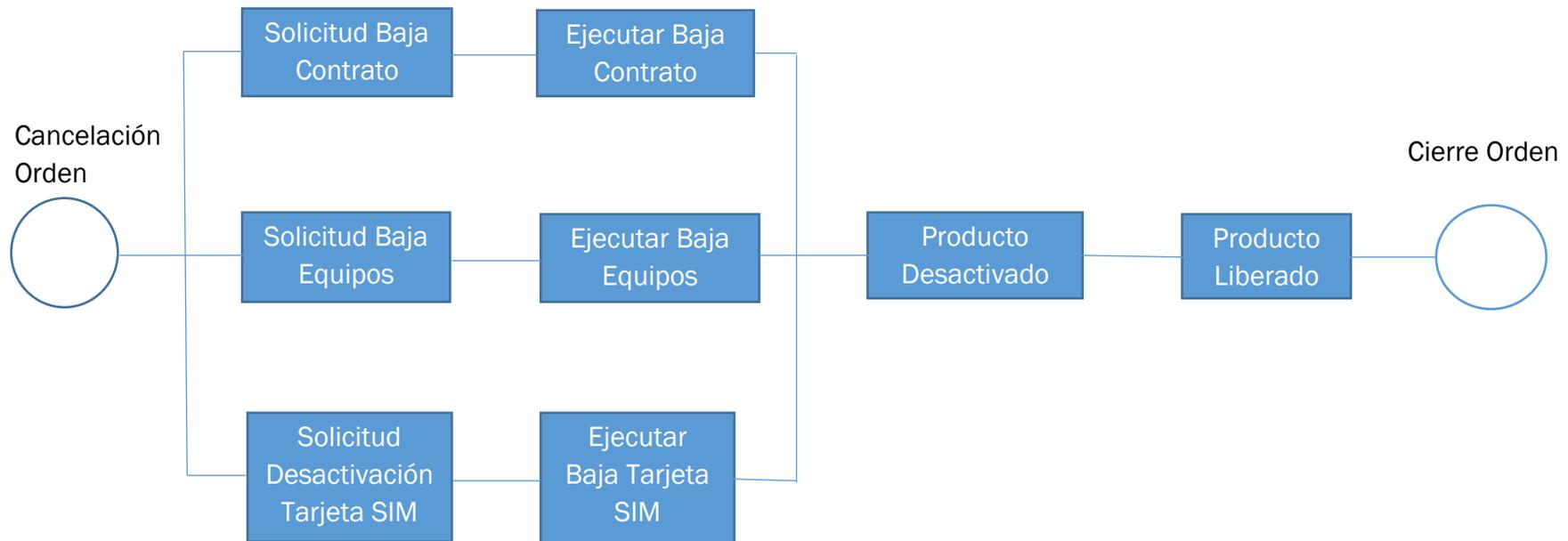


Figura 36. Proceso Cancelación Móvil

### Baja móvil

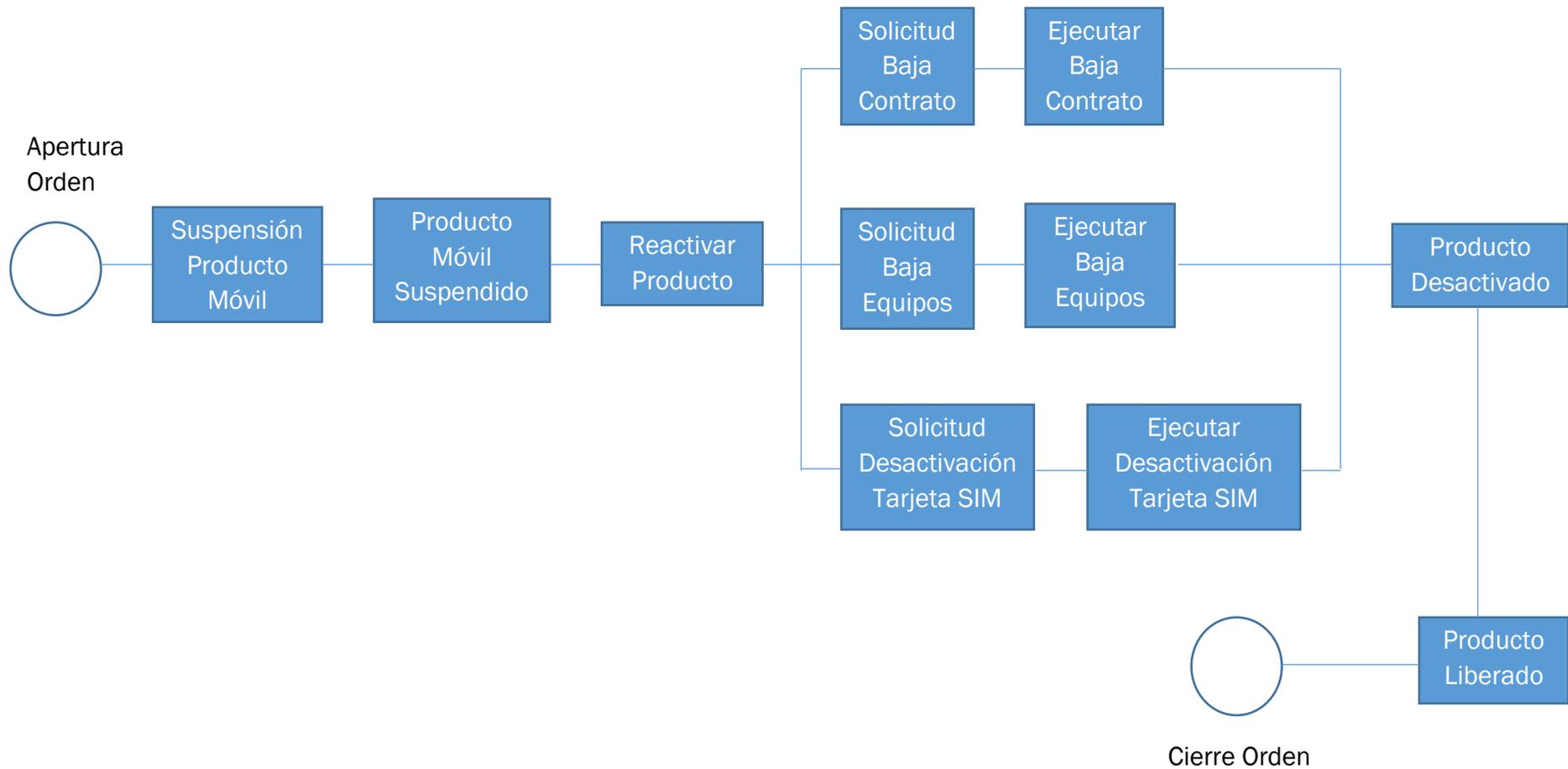


Figura 37. Proceso Baja Móvil

**Alta internet**

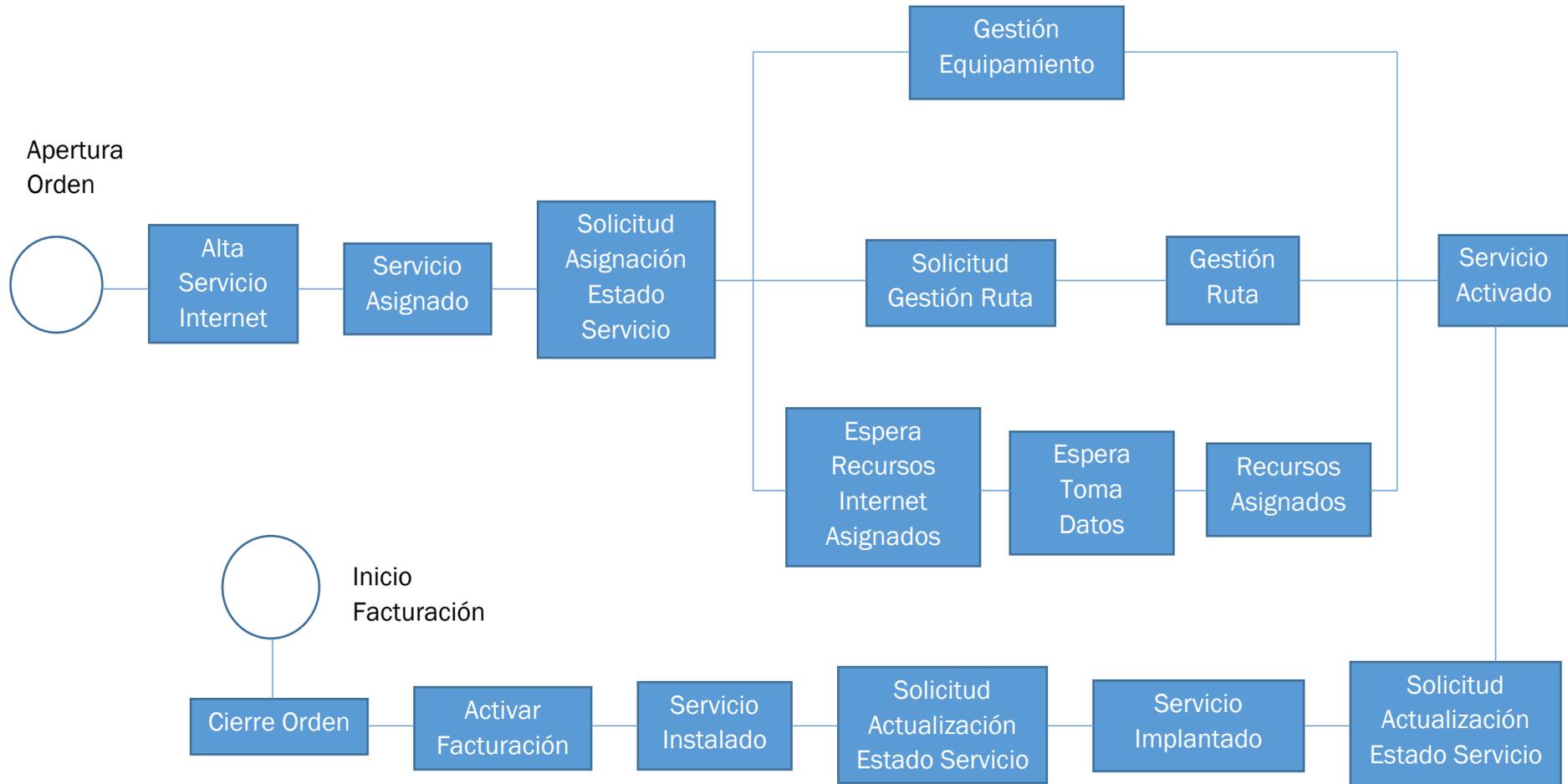


Figura 38. Proceso Alta Internet

### Cancelación internet

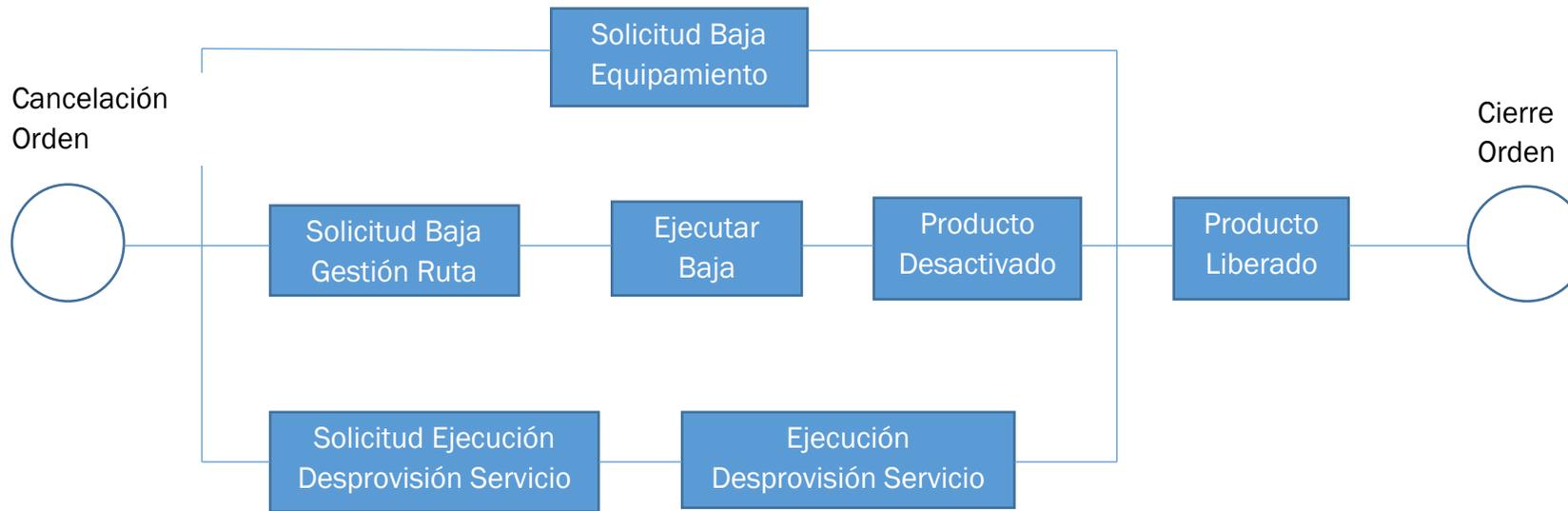


Figura 39. Proceso Cancelación Internet



Baja internet

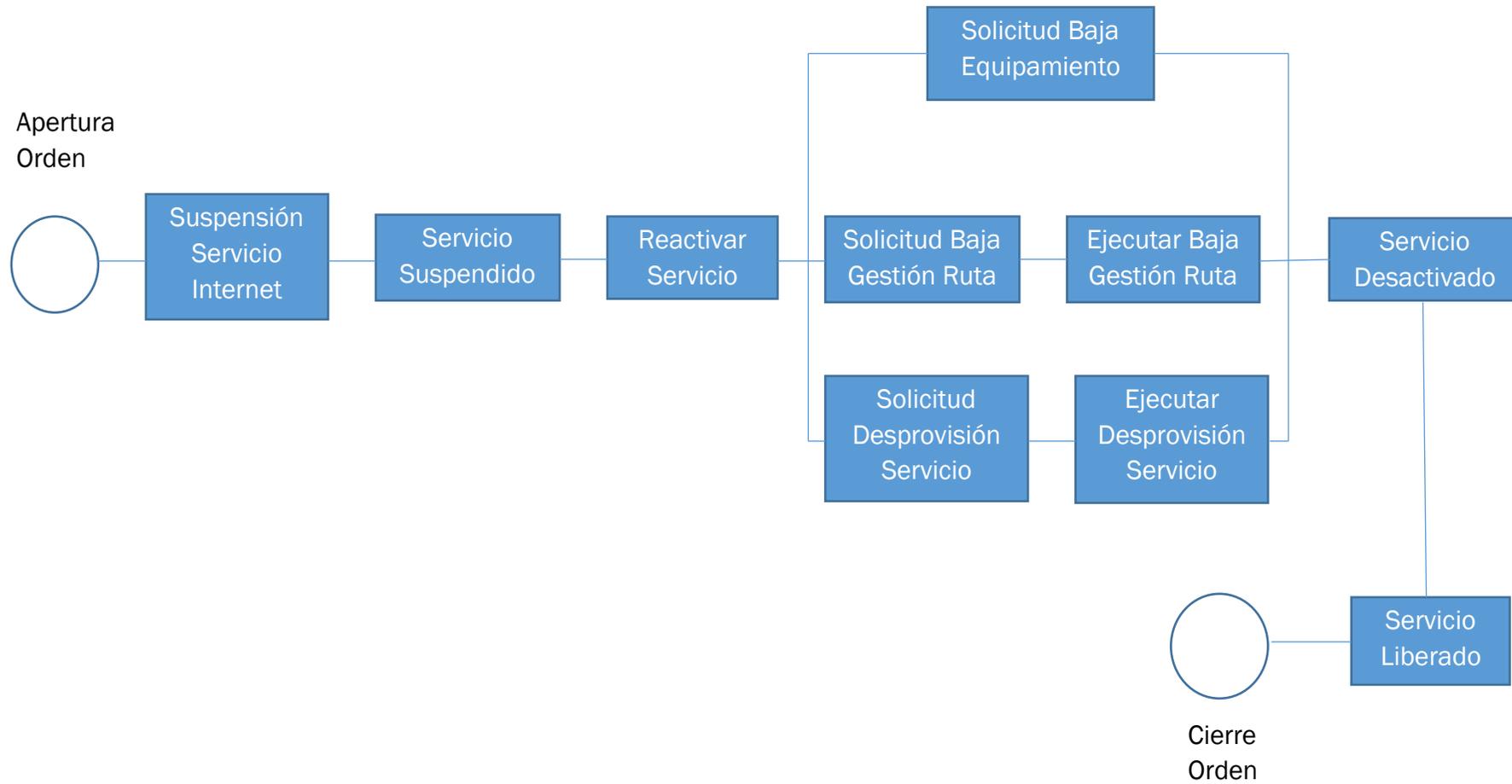


Figura 40. Proceso Baja Internet

## 5.7. UTILIZACIÓN DE JENKINS

Antes de empezar a utilizar Jenkins debemos de configurarlo, una vez se haya finalizado con la configuración, necesitaremos crear los usuarios. Para nuestro caso, se necesitará un jefe de proyecto, éste tendrá un perfil administrador, y seré yo, dos analistas y dos programadores los cuales tendrán el mismo perfil y los mismos permisos (ver figura 41).

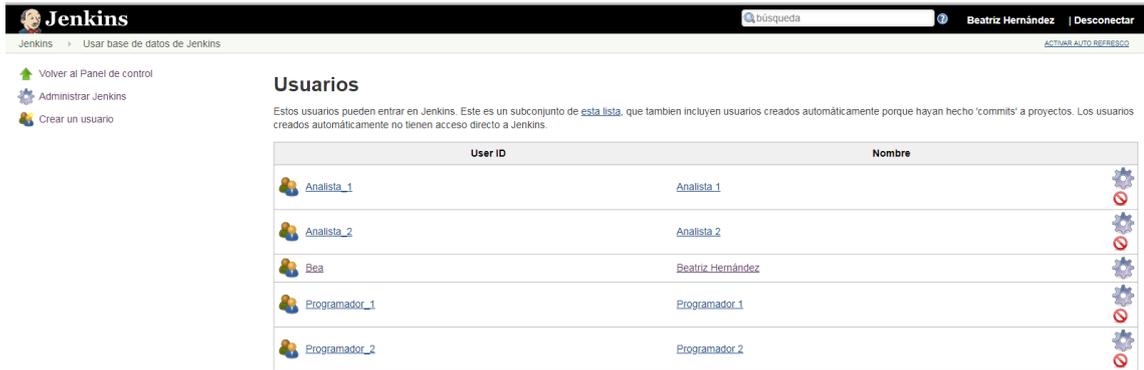


Figura 41. Usuarios creados en Jenkins

Una vez creados los usuarios, crearemos un nuevo proyecto, a este, le llamaremos PROY001 – Telefonasa. Este proyecto se visualizará de la siguiente manera. Previamente habremos realizado la integración con SVN.



Figura 42. Creación del proyecto en Jenkins

A continuación, se definirá el ciclo de vida que seguirá nuestro proyecto.

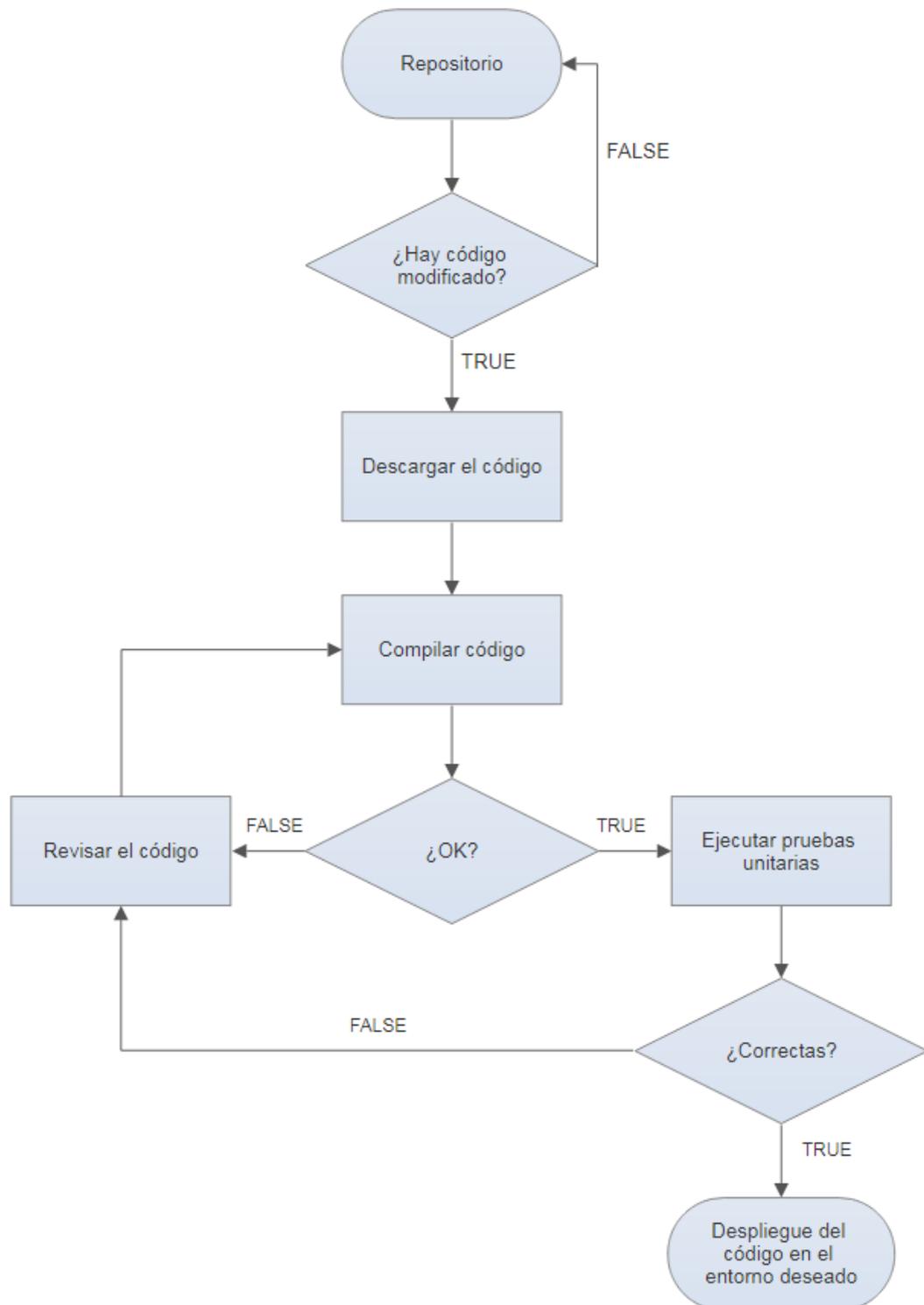


Figura 43. Ciclo de vida del proyecto

Una vez definido el ciclo de vida, se procederá a crear los Jobs necesarios. Necesitaremos los siguientes Jobs:

**Job de cambios en el repositorio**, como su propio nombre indica, la finalidad de este Job será comprobar si se han realizado cambios en SVN. En caso afirmativo, realizara una construcción del código y lo analizara. Si cualquiera de estos pasos falla se finalizará la ejecución.

Estará configurado de tal manera que se ejecute de manera automática cada media hora. Pero también se podrá ejecutar de manera manual en caso de que se necesite. Si el resultado esperado es el correcto, se ejecutará el siguiente Job, en cualquier otro caso avisará al programador de que el código no está correctamente desarrollado.

**Job de ejecutar pruebas unitarias**, el objetivo del Job será ejecutar una batería de pruebas unitarias previamente definidas. Si las todas las pruebas se han finalizado con resultado OK, significara que el código puede ser desplegado al entorno pre - productivo. En el caso contrario, se abortará la ejecución y en la salida del Job se mostrará el mensaje de error. Los programadores del proyecto serán los encargados de revisar el motivo del fallo.

**Job de ejecutar pruebas de integración**, este Job será el encargado de ejecutar las pruebas de integración que se hayan definido para nuestro proyecto. Al igual que el Job anterior, un OK en todas las pruebas de integración significara que el código puede ser desplegado en el entorno de producción.

Por el contrario, al estar integrado con Jira por cada KO obtenido en estas pruebas, se creará una incidencia. Será también competencia de los programadores de determinar porque se está produciendo el error.

**Job de despliegue en pre -producción**, el objetivo de este Job será el de ejecutar el código en el entorno pre - productivo.

**Job de despliegue en producción**, tendrá como finalidad ejecutar el código en el entorno productivo.

Para que todos los Jobs que hemos creado funcionen correctamente se tiene que realizar la integración con las herramientas necesarias.

## 5.8. UTILIZACIÓN DE TESTLINK

Para poder empezar a trabajar con TestLink, lo primero que necesitamos será elaborar un buen plan de pruebas. Se definirá un pequeño plan de pruebas basado en el caso práctico expuesto al principio. Éste, constará tanto de pruebas unitarias como integradas. A continuación, se definirán algunas de las pruebas a realizar.

### Pruebas unitarias

PRU_UNI_01	
<b>Resumen:</b> Se validará si el desarrollo realizado para la contratación de la nueva tarifa de telefonía móvil se realiza de forma correcta.	
<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Se buscará un cliente en BBDD.</li><li>2. Se le contratará la nueva tarifa de telefonía móvil.</li><li>3. Se le contratará otra tarifa móvil ya existente para comprobar que el resto de desarrollo sigue funcionando.</li></ol>	
<b>Resultados Esperados:</b> <ol style="list-style-type: none"><li>1. Se comprobará que al cliente se le puede contratar la nueva tarifa</li><li>2. Se comprobará que al cliente se le siguen pudiendo contratar otras tarifas de telefonía móvil existentes.</li></ol>	
<b>Estado de la prueba:</b> Passed, Failed, Blocked y Not Run	

Tabla 8. Pruebas unitarias caso 1

PRU_UNI_02	
<b>Resumen:</b> Se validará si el desarrollo realizado para la contratación de la nueva tarifa de internet, de 300 megas, se realiza de forma correcta.	
<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Se buscará un cliente en BBDD.</li><li>2. Se le contratará la nueva tarifa de internet.</li><li>3. Se le contratará otra tarifa de internet ya existente para comprobar que el resto de desarrollo sigue funcionando.</li></ol>	

<b>Resultados Esperados:</b> <ol style="list-style-type: none"><li>1. Se comprobará que al cliente se le puede contratar la nueva tarifa</li><li>2. Se comprobará que al cliente se le siguen pudiendo contratar otras tarifas de internet existentes.</li></ol>
<b>Estado de la prueba:</b> Passed, Failed, Blocked y Not Run

Tabla 9. Prueba unitaria caso 2

### Pruebas de integración

PRU_INT_01	
<b>Resumen:</b> Se validara que los procesos que se ejecutan para realizar la contratación del servicio de telefonía siguen el flujo correcto.	
<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Se buscará un cliente en BBDD.</li><li>2. Se le contratara la nueva tarifa de telefonía móvil.</li></ol>	
<b>Resultados Esperados:</b> <ol style="list-style-type: none"><li>1. Se comprobará que los procesos ejecutados son los correctos y siguen el flujo correcto.</li><li>2. Se comprobará la comunicación con los sistemas externos se realiza de forma correcta.</li><li>3. Se comprueba que al cliente se le ha realizado la contratación del servicio.</li></ol>	
<b>Estado de la prueba:</b> Passed, Failed, Blocked y Not Run	

Tabla 10. Pruebas de integración caso 1

PRU_INT_02	
<b>Resumen:</b> Se validara que los procesos que se ejecutan para realizar la contratación del servicio de internet siguen el flujo correcto.	
<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Se buscará un cliente en BBDD.</li><li>2. Se le contratara la nueva tarifa de 300 megas.</li></ol>	

<b>Resultados Esperados:</b>	
<ol style="list-style-type: none"> <li>1. Se comprobará que los procesos ejecutados son los correctos y siguen el flujo correcto.</li> <li>2. Se comprobará la comunicación con los sistemas externos se realiza de forma correcta.</li> <li>3. Se comprueba que al cliente se le ha realizado la contratación del servicio.</li> </ol>	
<b>Estado de la prueba:</b> Passed, Failed, Blocked y Not Run	

Tabla 11. Pruebas de integración caso 2

<b>PRU_INT_03</b>	
<b>Resumen:</b> Se validara que no se puede contratar dos tarifas de internet mostrara un mensaje de error.	
<b>Pasos:</b>	
<ol style="list-style-type: none"> <li>1. Se buscará un cliente en BBDD.</li> <li>2. Se le contratara la nueva tarifa de 300 megas y la tarifa existente de 1GB.</li> </ol>	
<b>Resultados Esperados:</b>	
<ol style="list-style-type: none"> <li>1. Se comprobara que se muestra un mensaje de error indicando que el cliente no puede contratar dos tarifas de internet para el mismo domicilio.</li> </ol>	
<b>Estado de la prueba:</b> Passed, Failed, Blocked y Not Run	

Tabla 12. Pruebas de integración caso 3

Una vez elaborado el plan de pruebas, ya podemos empezar a utilizar esta herramienta.

Necesitaremos crear distintos usuarios en función de la tarea que vayan a realizar, una tarea destinada al jefe de proyecto. El número de usuarios podrá variar según las necesidades del proyecto. Para el nuestro se crearán usuarios con los siguientes roles:

- Un usuario con rol “administrador”, responsable de crear y gestionar las pruebas de los proyectos. A este usuario solo tendrá acceso el jefe de proyecto.
- Cuatro usuarios con rol “tester”, que serán los encargados de llevar a cabo las pruebas que se han definido.

Posteriormente se crearán:

- a) Un Test Proyecto, al que denominaremos PROY001\_Telefonasa.
- b) Dos Test Suite, uno que se llamara Pruebas\_Unitarias y otro que se llamara Pruebas\_Integracion.

Dentro de cada uno de los Test Suite se cargarán en Teslink, los test cases o pruebas que hemos definido arriba. Una vez cargados, se asignará a cada uno de los usuarios creado las pruebas que tendrán que realizar.

El usuario será el encargado de ejecutar la prueba, adjuntar un documento con las evidencias de que la prueba se ha realizado y de que se han obtenido los resultados esperados y cambiar el estado de la tarea asignándole uno de los que nos ofrece la herramienta. Se utilizarán:

- Not Run, cuando no se haya ejecutado el test case o prueba.
- Passed, cuando se ha ejecutado el test case o prueba y se han obtenido los resultados esperados.
- Failed, se ha ejecutado el test case o prueba, pero el resultado no ha sido el que se esperaba. Cada vez que se obtenga un Fail en una de las pruebas automáticamente se creara una incidencia en Jira.
- Blocked, el test case o prueba no se puede ejecutar.

## 5.9. UTILIZACIÓN DE JIRA

Para garantizar una gestión de incidencias eficaz y adecuada crearemos una cola, a la cual nos podrán asignar las incidencias que se presenten y que estén relacionadas con el proyecto que estamos desarrollando.

El encargado de la creación de esta cola como de la creación de los usuarios y sus permisos es el jefe de proyecto.

A esta cola podrán acceder unos determinados usuarios a los que con anterioridad se les ha dado de alta en la herramienta y se les ha permitido el acceso. El equipo de personas a cargo de llevar a cabo la resolución de incidencias se ira asignando a sus usuarios las incidencias para proceder a su análisis y posterior resolución.

El modo de actuación será el que se indica a continuación, primero se resolverán las incidencias de prioridad alta, luego las de prioridad media y por ultimo las de prioridad baja. El usuario será el encargado de asignar dicha prioridad.

Universidad de Valladolid

Supongamos que cuando se han realizado las pruebas relacionadas con la nueva tarifa de móvil, en alguna parte antes de llegar a su contratación final, el proceso que ésta sigue falla.

Este fallo puede ser ocasionado por un mal desarrollo o por tema de calidad de datos, es decir, el problema está en la falta de algún dato en alguna tabla o en algunas de sus relaciones. Para ello necesitaremos definir dos workflows o flujo de trabajo orientados a la resolución de incidencias.

Los analistas y programadores encargados de realizar el proyecto sabrán que flujo de ellos aplicar, una vez hayan realizado el análisis de la incidencia.

Los workflows a seguir se muestran a continuación. El primero es el que se seguirá cuando la solución de la incidencia requiera un correctivo, es decir, se necesite modificar el código fuente para que el problema se solucione, y el segundo será el que se siga cuando la solución esté relacionada con la calidad de datos, es decir, la solución de la incidencia radica en la modificación de datos almacenados en la BBDD.

### 1. Workflow correctivos

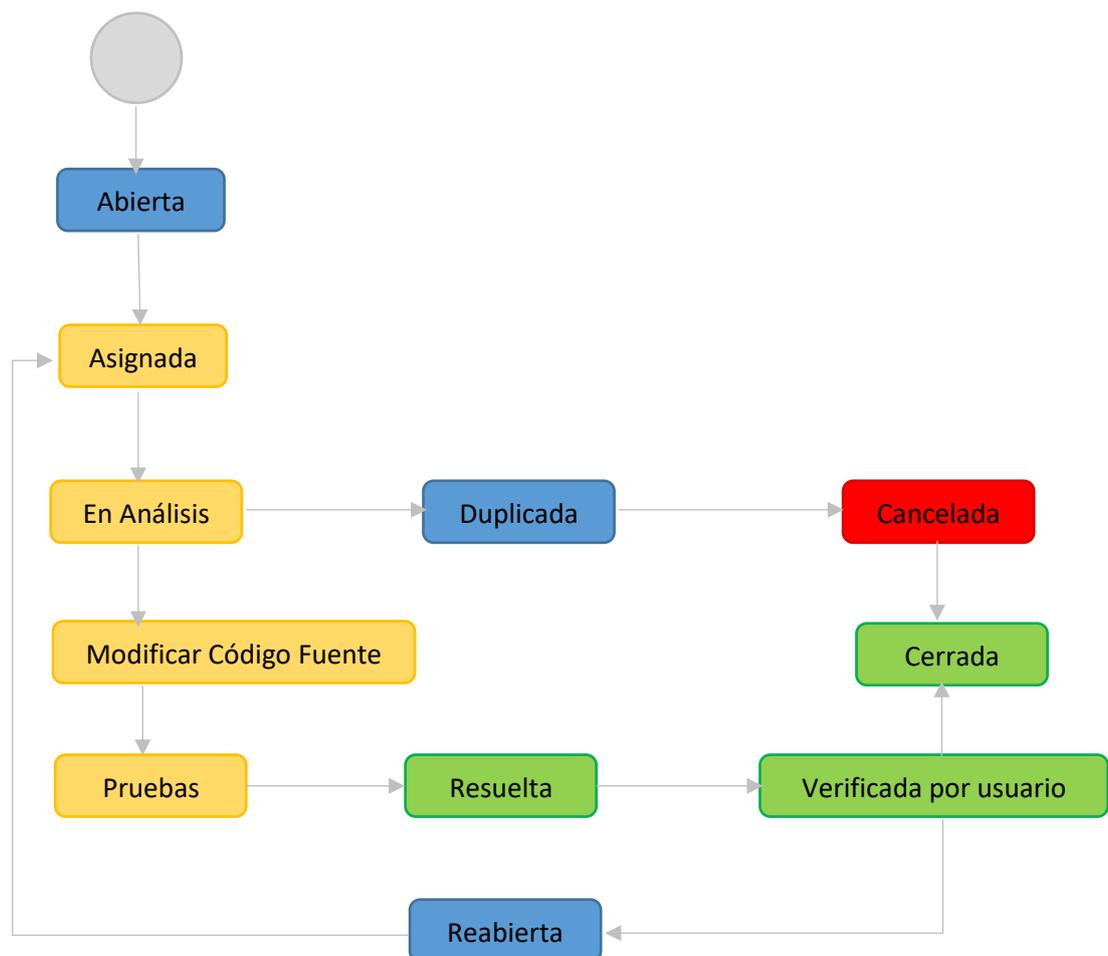


Figura 44. Diagrama de flujo correctivo

## 2. Workflow calidad de datos

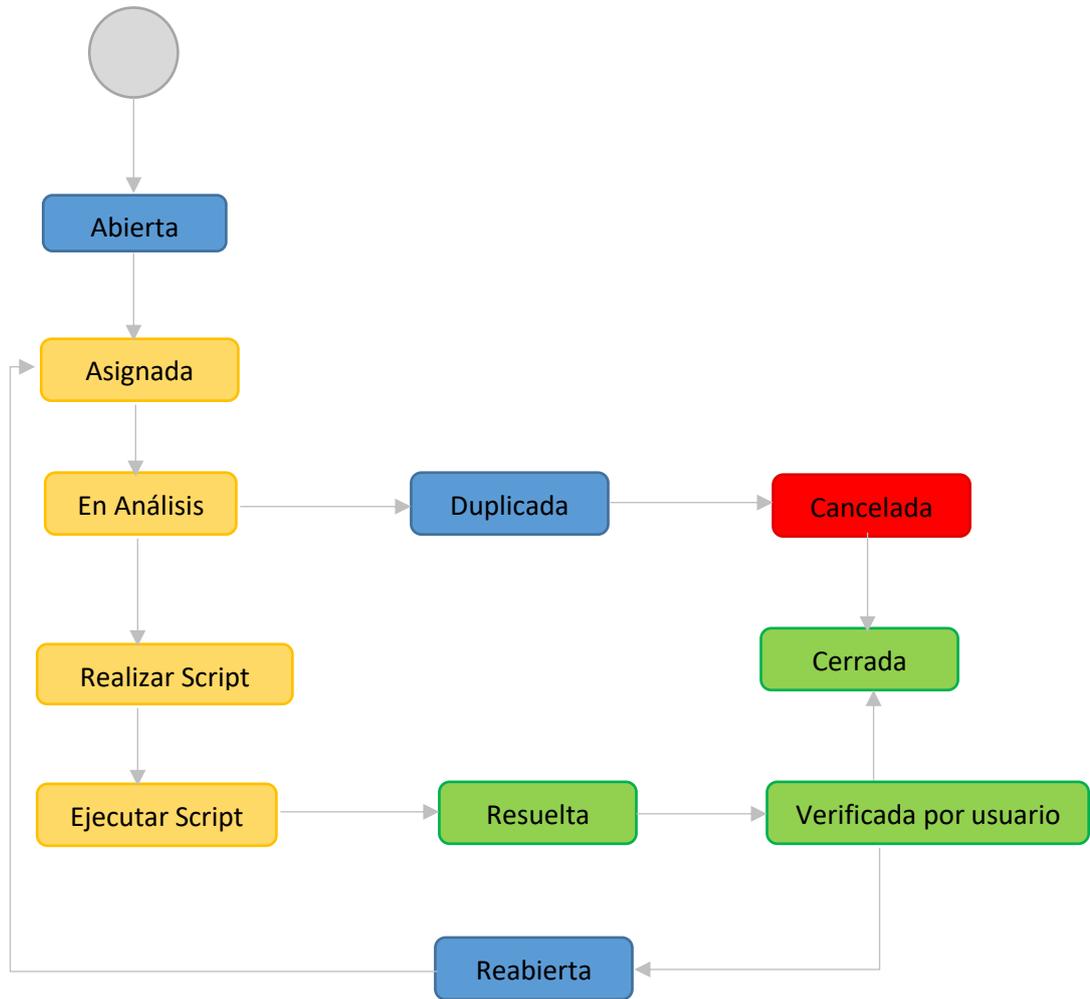


Figura 45. Diagrama de flujo de calidad de datos





## CONCLUSIONES

Finalizado este trabajo podemos concluir que se ha conseguido el objetivo propuesto, la implementación teórica de un CRM destinado al sector de las comunicaciones. Además, esta herramienta destaca por:

- Ser un CRM “open source”, pudiendo ser utilizado por cualquier tipo de empresa.
- Reducir los costes de las empresas que lo utilicen.
- Ser flexible, adaptándose a cualquier tipo de necesidad.
- Garantizar la calidad de los servicios que ofrece, reduciendo no solo el coste económico, sino también el humano.
- Aumentar la capacidad de fidelización de los clientes, mejorar la capacidad de respuesta, ahorrar tiempo, optimizar los recursos de la empresa, así como, aumentar las oportunidades de venta y mejorar la productividad.
- Conseguir desarrollar una herramienta funcional, automática y eficaz. Características necesarias para aplicaciones de este tipo.



## BIBLIOGRAFÍA

CABERO, J. (1998). Impacto de las nuevas tecnologías de la información y la comunicación en las organizaciones educativas. En Lorenzo, M. y otros (coords): Enfoques en la organización y dirección de instituciones educativas formales y no formales (pp. 197-206). Granada: Grupo Editorial Universitario.

CHEN, I.J. y POPOVICH, K. (2003): "Understanding Customer Relationship Management-People, Process and Technology", *Business Process Management Journal*, 9 (5), pp. 672-688.

CHOY, K.L. y FAN, K.; LO, V. (2003): "Development of an Intelligent Customer-Supplier Relationship Management System: The Application of CaseBased Reasoning", *Industrial Management +Data System*, 103 (3-4), pp. 263-274

FINNEGAN, D.J. y CURRIE, W.L. (2010): "A multiLayered Approach to CRM Implementation: An Integration Perspective", *European Management Journal*, 28, pp. 153-167.

COLLINS-SUSSMAN, B, MICHAEL PILALTO, C y W. FITZPATRICK, B. (2004). *Version Control with Subversion*. O'Reilly Media.

GIL, E. (2002). "Identidad y Nuevas Tecnologías". <<http://www.voc.edu/web/esplart/gil0902/htm>>. [Consulta: 10 de Junio de 2019].

HOYOS CHAVERRA, J.A y VALENCIA ARIAS, A. (2012). "El Papel De Las TIC En El Entorno Organizacional De Las Pymes". *Trilogía Ciencia Tecnología Sociedad* 4 (7), 105-22. < <https://doi.org/10.22430/21457778.155>>. [Consulta: 10 de Junio de 2019].

MARQUES GRAELLS, P. (2000). "Impacto Tic en Educación: Funciones y Limitaciones"  
<[http://sgpwe.izt.uam.mx/files/users/virtuami/file/ext/miplan\\_impacto\\_actv\\_impactoticeducacion.pdf](http://sgpwe.izt.uam.mx/files/users/virtuami/file/ext/miplan_impacto_actv_impactoticeducacion.pdf)>. [Consulta: 10 de Junio de 2019].

MACAU, R (2004). "TIC: ¿para qué? (Funciones de las tecnologías de la información y la comunicación en las organizaciones)". *Revista de Universidad y Sociedad del Conocimiento (RUSC)*. Vol. 1, nº 1.

<http://www.uoc.edu/rusc/dt/esp/macau0704.pdf>. [Consulta: 10 de Junio de 2019].



PARVATIYAR, A. y SHETH, J.N. (2001): "Customer Relationship Management: Emerging Practice, Process, and Discipline. Journal of Economic and Social Research, 3 (2), pp. 1-34.

THOMPSON, A. y STRICKLAND, A. (2004). *Administración Estratégica*. México: Mc Graw Hill.

Portal de herramientas de Everis. <[https://steps.everis.com/confluence/login.action?os\\_destination=%2Fspaces%2Fview.action%3Fkey%3DSTEPS&permissionViolation=true](https://steps.everis.com/confluence/login.action?os_destination=%2Fspaces%2Fview.action%3Fkey%3DSTEPS&permissionViolation=true)>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de Subversion. <<https://subversion.apache.org/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de MySQL. <<https://www.mysql.com/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de Oracle. < <https://www.oracle.com/es/index.html>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de SQLite. <<https://www.sqlite.org/index.html>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de PostgreSQL. < <https://www.postgresql.org/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de SQL Server. <<https://www.microsoft.com/es-es/sql-server/sql-server-2019>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de GIT. < <https://git-scm.com/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de Mercurial. <<https://www.mercurial-scm.org/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de Eclipse. <<https://www.eclipse.org/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de NetBeans. <<https://netbeans.org/>>. [Consulta: 10 de Junio de 2019].

Página Web Oficial de IntelliJ IDEA. <<https://www.jetbrains.com/idea/>>. [Consulta: 10 de Junio de 2019].

Página Oficial de jBPM. <<https://www.jbpm.org/>>. [Consulta: 10 de Junio de 2019].



*Página Oficial de Bonita BPM.* <<https://es.bonitasoft.com/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Activiti.* <<https://www.activiti.org/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de TestLink.* <<http://testlink.org/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Selenium.* <<https://www.seleniumhq.org/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Jenkins.* <<https://jenkins.io/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Bamboo.* <<https://es.atlassian.com/software/bamboo>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Jira.* <<https://es.atlassian.com/software/jira>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Mantis Bug Tracker.* <<https://www.mantisbt.org/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Redmine.* <<https://www.redmine.org/>>. [Consulta: 10 de Junio de 2019].

*Página Oficial de Bugzilla.* <<https://www.bugzilla.org/>>. [Consulta: 10 de Junio de 2019].

*Tipos y modelos de CRM.* <<https://www.sumacrm.com/soporte/tipos-de-crm-y-modelos>>. [Consulta: 10 de Junio de 2019].

*Tipos y modelos de CRM.* <<https://www.evaluandocrm.com/tipos-de-crm/>>. [Consulta: 10 de Junio de 2019].

*Universidad de Salamanca.* <<http://diarium.usal.es/mbadefitic/2012/02/27/tic-en-sistemas-de-gestion-de-clientes-crm/>>. [Consulta: 10 de Junio de 2019].

*Objetivos del CRM.* <<https://jaimeospina.wordpress.com/2010/06/30/objetivos-fundamentales-de-crm/>>. [Consulta: 10 de Junio de 2019].

*Universidad de Valencia.* <<https://www.uv.es/~bellochc/pdf/pwtic1.pdf>>. [Consulta: 10 de Junio de 2019].

*Estrategias del CRM.* <<http://www.puromarketing.com/20/19371/estrategias-beneficios-puede-aportar-empresas.html>>. [Consulta: 10 de Junio de 2019].



## ANEXOS

### ANEXO 1. TABLAS DIAGRAMAS ENTIDAD RELACIÓN

Se detallara de cada tabla empleada en el diagrama entidad- relación, lo que significa cada atributo así como el tipo de dato a utilizar.

TABLA DIRECCION		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_direccion	int	Número de referencia, se crea de manera automática
Calle	varchar	Dirección del cliente
Ciudad	varchar	Ciudad donde reside el cliente
Poblacion	varchar	Población donde reside el cliente
C.P	smallint	Código postal asociado a la dirección del cliente
Pais	varchar	País de residencia del cliente

Tabla 13. Atributos tabla dirección

TABLA CLIENTE		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_cliente	int	Número de referencia, se crea de manera automática
Nombre	varchar	Nombre del cliente
Apellido	varchar	Apellido del cliente
DNI	varchar	DNI del cliente
Telefono	int	Número de teléfono del cliente
ID_direccion	int	Relación entre tabla dirección y tabla cliente

Tabla 14. Atributos tabla cliente

TABLA CONTRATO		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_contrato	int	Número de referencia, se crea de manera automática
Tipo_contrato	varchar	indica el tipo de contrato, pudiendo ser un alta, una baja, modificación o cancelación
Estado	varchar	Estado del contrato (Abierto, Cerrado, Cancelado)
ID_orden	int	Número de contrato
Fecha_creacion	timestamp	Fecha en la que se ha creado el contrato
ID_cliente	int	Relación entre tabla cliente y tabla contrato

Tabla 15. Atributos tabla contrato

TABLA PROCESOS		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_proceso	int	Número de referencia, se crea de manera automática
ID_contrato	int	Relación entre la tabla contrato y la tabla proceso
Nombre_proceso	varchar	Nombre del proceso
Estado_proceso	varchar	Estado del proceso (Activo, Inactivo, Completo, Esperando)

Tabla 16. Atributos tabla procesos

TABLA TECNICOS		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_tecnico	int	Número de referencia, se crea de manera automática
Nombre	varchar	Nombre del técnico encargado de la instalación del equipo
Apellido	varchar	Apellido del técnico encargado de la instalación del equipo
DNI	varchar	DNI del técnico encargado de la instalación del equipo
ID_equipo	int	Relación entre la tabla técnicos y la tabla equipos

Tabla 17. Atributos tabla técnicos

TABLA EQUIPOS		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_equipo	int	Número de referencia, se crea de manera automática
ID_contrato	varchar	Relación entre la tabla equipos y la tabla contrato
Nombre de equipo	varchar	Nombre del equipo
Tipo_equipo	varchar	Tipo de equipo
Estado	varchar	Estado del equipo (Libre, Ocupado)
Num_serie	int	Número de serie del equipo
Fecha_instalacion	timestamp	Fecha en la que se instala el equipo
Fecha_desinstalacion	timestamp	Fecha en la que se desinstala el equipo
Fecha_modificacion	timestamp	Fecha de cambio de equipo
ID_producto	int	Relación entre la tabla equipos y la tabla productos

Tabla 18. Atributos tabla equipos

TABLA PRECIOS		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_precio	int	Número de referencia, se crea de manera automática
Nombre_producto	varchar	Nombre del producto
Precio_producto	float	Precio del producto
ID_producto	int	Relación entre la tabla precios y la tabla productos

Tabla 19. Atributos tabla precios

TABLA PRODUCTOS		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_producto	int	Número de referencia, se crea de manera automática
Tipo_producto	varchar	Tipo de producto (Internet, Móvil, Fijo, BAM...)
Nombre_prod_prov	varchar	Relación entre la tabla productos y la tabla proveedor
ID_cliente	int	Relación entre la tabla productos y la tabla clientes
ID_contrato	int	Relación entre la tabla productos y la tabla contrato

Tabla 20. Atributos tabla productos

TABLA PROVEEDORES		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_proveedor	int	Número que indica el proveedor
Nombre_proveedor	varchar	Nombre del proveedor
Nombre_prod_prov	varchar	Número de referencia, se crea de manera automática
ID_tecnico	int	Relación entre la tabla proveedores y la tabla técnicos

Tabla 21. Atributos tabla proveedores

TABLA PARQUE		
ATRIBUTO	TIPO DE DATO	DESCRIPCION
ID_parque	int	Número de referencia, se crea de manera automática
Nomb_produc	varchar	Nombre del producto que baja a parque
Estado_produc	varchar	Estado del producto en parque (Activo, Desconectado, Pend.Instalacion)
Fecha_bajada_parque	timestamp	Fecha de bajada del producto a parque
ID_cliente	int	Relación entre la tabla parque y la tabla clientes
ID_producto	int	Relación entre la tabla parque y la tabla producto

*Tabla 22. Atributos tabla parque*

