

Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Ingeniería Informática

Curso 2022-2023

Trabajo Fin de Grado

**IMPLANTACIÓN DE UN SISTEMA DE GESTIÓN DE
RECURSOS EMPRESARIALES DE CÓDIGO
ABIERTO**

Autor: José Manuel Aguado Gala

Tutor: Eduardo García Pardo

Agradecimientos

Este sí que es el último de verdad. Este TFG pone fin a una época llena de experiencias y momentos tanto buenos como malos. Pero lo más importante es que son momentos que me han hecho crecer como persona. Y esto se ha podido dar gracias a que estoy rodeado de grandes personas que me han podido acompañar en el camino. Gracias a mis padres, a mi hermano y a Raquel por ser un constante apoyo siempre. Gracias a Dani, Víctor, David, Cobas, Yara y todos los demás amigos con los que tantos buenos momentos he compartido. Agradecer a Eduardo y todo el equipo de Second Episode por acogerme como uno más y permitirme ser parte de la mejor experiencia laboral que he podido tener. Y, sobre todo, va por ti, abuela, y va por ti, abuelo. Espero haberos hecho sentir orgullosos.

Resumen

En la actualidad, el desarrollo de *software* ha evolucionado de manera exponencial. Cada día son más las empresas que adquieren nuevos productos o servicios informáticos para mejorar los procesos de su negocio así como la experiencia que ofrecen a sus clientes.

El crecimiento de la demanda de productos y servicios *software* viene acompañado de la necesidad de las empresas que proveen estos servicios de mejorar sus procesos para ofrecer un producto adaptado las necesidades del cliente, eficiente y con las correctas medidas de seguridad. Por ello, este Trabajo de Fin de Grado se centra en la implantación de un sistema de gestión de recursos empresariales de código abierto, actividad principal de la empresa Second Episode S.L., en el seno de la cual se ha desarrollado este proyecto.

En el proceso de implantación se abordarán diversos aspectos como el primer contacto con el cliente, el análisis de requisitos, la instalación y securización del sistema, la formación de los usuarios, integraciones, posteriores desarrollos y el mantenimiento. Se hará un especial hincapié en la parte de instalación y securización donde se aportarán una serie de pasos a seguir, acompañados de los correspondientes *scripts* que hagan este proceso lo más rápido y sencillo posible.

Palabras clave:

- Odo
- Procesos de negocio
- Protocolo de implantación
- *Software As A Service*
- *Scripts*
- Análisis de requisitos

Listado de acrónimos

- CRM: *Customer Relationship Management*
- ERP: *Enterprise Resource Planning*
- HTML: *HyperText Markup Language*
- HTTP: *Hypertext Transfer Protocol*
- IP: *Internet Protocol*
- SaaS: *Software as a Service*
- SFTP: *Secure File Transfer Protocol*
- SSH: *Secure SHell*
- SSL: *Secure Sockets Layer*
- TCP: *Transmission Control Protocol*
- TIC: Tecnologías de la Información y Comunicación.
- TPV: Terminal de Punto de Venta

Índice de contenidos

Índice de tablas	XII
Índice de figuras	XIV
Índice de códigos	XVI
1. Introducción	1
1.1. Motivación	1
1.2. Contexto	3
1.3. Definiciones previas	4
1.4. Estado del arte	6
1.4.1. Principales ERP	7
1.4.2. Odoo	10
1.4.3. Servidores web	11
1.5. Estructura de la memoria	15
2. Objetivos	16
2.1. Objetivo general	16
2.2. Objetivos específicos	17
3. Protocolo de implantación de un ERP	18
3.1. Contacto inicial	18
3.2. Extracción de requisitos generales	21
3.3. Extracción de requisitos específicos	24
3.4. Instalación del sistema	25

3.5. Parametrización	25
3.5.1. Formación	26
3.6. Integraciones	28
3.7. Desarrollos a medida	28
3.8. Mantenimiento	29
4. Metodología	30
4.1. Scrum	31
4.2. Metodología elegida	33
5. Instalación del sistema	36
5.1. Tecnologías utilizadas	36
5.1.1. Odoo	37
5.1.2. Ubuntu	37
5.1.3. Apache	37
5.1.4. PostgreSQL	38
5.2. <i>Product Backlog</i>	38
5.3. Tareas detalladas	40
6. Artefactos resultantes	50
6.1. Conexión remota al servidor	50
6.2. Odoo	51
6.3. Configuración del servidor <i>web</i>	53
6.4. Copias de seguridad	54
7. Caso de ejemplo	59
8. Conclusiones y trabajos futuros	63
8.1. Conclusiones	63
8.2. Trabajos futuros	65
Bibliografía	66

Índice de tablas

1.1. Comparativa de los servidores web más utilizados, extraído de Net-craft [Net22].	14
5.1. Product Backlog con las tareas a desarrollar.	39

Índice de figuras

1.1. Cuota de mercado de servidores web, extraído de Netcraft.	14
3.1. Diagrama de actividad con las fases existentes en el protocolo de instalación.	19
3.2. Principales áreas temáticas de Odoo.	22
4.1. Ejemplo de <i>Product Backlog</i> y <i>Sprint Backlog</i>	33
6.1. Conexión por SSH al servidor.	51

Índice de códigos

5.1. Comando para ser superusuario <i>root</i>	40
5.2. Configuración de nuevo un <i>host</i> SSH.	41
5.3. Comando para la ejecución de un <i>host</i> SSH.	41
5.4. Comandos para creación de entorno virtual de Python.	44
5.5. Configuración del <i>VirtualHost</i> de Odoo en Apache.	45
5.6. Configurar <i>VirtualHost</i> para el nuevo dominio.	46
5.7. Comando de Certbot para creación de un certificado.	47
5.8. Bloque de puertos 25 y 143 por <i>iptables</i>	49
5.9. Escucha de puertos 25 y 143 con <i>netstat</i>	49
6.1. Archivo de configuración básica de Odoo.	51
6.2. Archivo de configuración del servicio de Odoo.	52
6.3. <i>VirtualHost</i> creado por Certbot.	53
6.4. <i>Script</i> de copia de la base de datos de Odoo.	54

1

Introducción

En este capítulo se presenta la motivación de este Trabajo de Fin de Grado (TFG) así como algunas aclaraciones y definiciones sobre tecnologías y herramientas que se han utilizado durante su desarrollo, permitiendo tener así una perspectiva inicial clara para entender mejor el desarrollo de este proyecto.

1.1. Motivación

La motivación para la realización de este trabajo se fundamenta en tres pilares: en primer lugar la intención de ahondar y agrandar los conocimientos sobre sistemas, más concretamente en lo que se refiere a instalación, configuración y securización de un sistema. En segundo lugar, en la intención en desarrollar conocimientos sobre los flujos de negocio. Por último, aprender sobre la gestión de un nuevo cliente.

Dentro del apartado de instalación y configuración de un sistema, se pretende aumentar los conocimientos sobre el uso de una conexión segura a ellos que

permita conectarse de manera remota y en pocos segundos. Esta conexión segura permitirá también la transferencia de archivos al servidor, necesario para conseguir instalar el *software* deseado en el servidor, siendo en este caso Odoo, un *software* de gestión empresarial *open source*, también denominado ERP (del inglés *Enterprise Resource Planning*), con una gran tasa de crecimiento en los últimos años [cre22].

Una vez realizada la configuración, también se busca asegurar la integridad de los datos. Para ello, es necesario aplicar múltiples medidas tales como automatizar las copias seguridad que se manden a una base de datos externa, una buena seguridad del servidor mediante la correcta configuración de diversos aspectos o la configuración de un certificado SSL (*Secure Sockets Layer*), entre otros.

En el apartado de los flujos de negocio, se pretende ahondar en la gestión de un nuevo cliente: conocer sus necesidades, proporcionarle una solución a estas y después parametrizar el sistema una vez instalado para que quede totalmente adaptado a sus necesidades.

Todos estos aspectos son sumamente relevantes y es necesario conocerlos para poder ofrecer un buen servicio al cliente, que sea rápido, resistente a fallos y que carezca de posibles brechas de seguridad. Además de la motivación de ampliar los conocimientos sobre estas tecnologías, herramientas y buenas prácticas, también existe la motivación de indagar sobre la configuración actual de servidores y poder proponer mejoras que aumenten la calidad del servicio para el cliente y/o eliminen la deuda técnica existente [Gar22].

Ofrecer un buen servicio es prioritario, sobre todo en la actualidad del mercado de los ERP, que como se ha mencionado anteriormente, es un mercado con un crecimiento en alza. Más concretamente, el *software* de gestión de Odoo se acerca al 60% de crecimiento anual en 2021 [Odo22]. Por lo tanto, es necesario disponer de un despliegue de servidores rápido y con un proceso claro y, en la medida de lo posible, automatizado para poder atender a la creciente demanda.

1.2. Contexto

En primer lugar, cabe mencionar que este TFG se ha desarrollado en el transcurso de las prácticas curriculares en la empresa Second Episode S.L., dedicada a la implantación de Odoo, un sistema de planificación de recursos empresariales de código abierto, a empresas utilizando SaaS (*Software as a Service*) como modelo de distribución. El modelo SaaS consiste en proporcionar productos *software* a los clientes a través de Internet, manteniendo el soporte lógico y los datos en servidores operados por el distribuidor. Por lo tanto, los clientes podrán acceder desde cualquier dispositivo sin realizar ninguna descarga o instalación, manteniendo además todos los datos y proporcionando así una gran flexibilidad. De este modo, la responsabilidad de la provisión y funcionamiento del servicio queda en manos de la compañía tecnológica que proporciona el servicio.

Por otro lado, el tejido empresarial español se encuentra en un momento de crecimiento tecnológico y en mitad de un proceso de digitalización, generando una mayor demanda de productos TIC (Tecnologías de la Información y la Comunicación), entre los que destacan los ERP. Estos proporcionan a las empresas herramientas muy valiosas para poder controlar y gestionar todos los procesos y ámbitos de su negocio. Esta mayor demanda hace que las empresas que se dedican a las TIC se encuentren también en un proceso de evolución, que también implica optimizar sus procesos para poder solventar todas las peticiones y asegurar la calidad del producto.

Cabe mencionar que la expansión del tejido empresarial español se ha visto acelerada por la pandemia del COVID-19, que ha obligado a miles de empresas a cambiar su metodología de trabajo y tener que realizar las tareas de manera remota. La digitalización ha supuesto una gran ayuda para realizar estas tareas de manera telemática y permitir a las empresas adaptarse rápidamente al nuevo panorama que se les ha presentado.

La expansión tecnológica en las empresas se está viendo apoyada por las instituciones, que tienen el objetivo de conseguir un futuro sostenible, el cual pasa por la innovación y el desarrollo. Muestra de esto es la Agenda 2030 [Uni22a], donde se presenta un plan de desarrollo sostenible creado por la ONU (Organización de las Naciones Unidas) y adoptado también por los diferentes países que conforman esta organización. La Agenda 2030 cuenta con 17 objetivos y 169 metas, siendo de especial importancia para las empresas el objetivo 9, que trata sobre la industria, innovación e infraestructuras [Uni22b].

Sumado a la expansión tecnológica de las empresas, en el año 2022, se ha llevado a cabo una inversión por parte del Gobierno de España en cuanto a inversión en digitalización empresarial a través del denominado “Kit Digital” [Ace22], que proporcionará subvenciones de hasta 12.000 euros por empresa. Con esta finalidad, esta gran inversión está financiada por fondos de la Unión Europea, la cual busca conseguir los objetivos de la Agenda 2030 antes mencionados. Con la propuesta del Kit Digital se espera conseguir la digitalización de las empresas en España, algo necesario para el tejido empresarial de este país que se encontraba en desventaja en comparación con los demás países europeos.

Las subvenciones anteriormente mencionadas supondrán un mayor incremento en los clientes de empresas que se dediquen a la disposición de *software* como servicio, como lo es la empresa Second Episode, que deberán estar prevenidas y poder proporcionar disponibilidad constante con la disposición de nuevos servidores en caso de ser necesario.

1.3. Definiciones previas

En esta sección se definen algunos conceptos de interés que permitirán un mejor entendimiento de los temas que se tratarán más adelante en el desarrollo de este trabajo.

ERP (*Enterprise Resource Planning*, sistema de planificación de recursos empresariales): son sistemas de gestión de información con los que se pueden controlar, dirigir y automatizar múltiples aspectos de un negocio, como son la facturación y contabilidad, el inventario de productos, las ventas o los recursos humanos, entre otros.

Odoo: es un *software* de ERP de código abierto y desarrollado en Python. Odoo intenta unir diversos módulos para poder controlar todos los aspectos de un negocio dentro de su *software* con la intención de conseguir un sistema de planificación de recursos general de manera que se ahorren desperdicios de tiempo y datos cuando se trabaja con distintas herramientas. Para ello, Odoo cuenta por defecto con una gran cantidad de módulos. Estos además se pueden aumentar modificando o ampliando el código al tratarse de un *software* de código abierto.

SaaS (*Software as a Service*): se trata de un modelo de distribución de *software* en el que el cliente recibe acceso al servicio que quiere utilizar de manera remota. De esta forma el cliente no tiene que instalar nada de forma local, y la responsabilidad de esto reside en el proveedor, el cual gestiona el soporte lógico y los datos a través de servidores remotos.

Servidor: típicamente hace referencia a una computadora (ya sea física o virtual) o a un conjunto de ellas capaz de recibir peticiones de un cliente y devolver una respuesta acorde a esta petición. De este modo, un servidor web o un servidor HTTP (*Hypertext Transfer Protocol*) es el programa que se ejecuta dentro de una máquina para gestionar estas peticiones y respuestas que se producen siendo HTTP el protocolo utilizado.

HTTP (*Hypertext Transfer Protocol*): es el protocolo de comunicación que permite el intercambio de información a través de redes de computadoras. Entre otros, permite el intercambio de recursos entre clientes y servidores, como por ejemplo archivos de tipo HTML (*HyperText Markup Language*), utilizados como base para la estructuración de la información en la *World Wide Web*.

SSH (*Secure SHell*): es el nombre de un protocolo de aplicación y a la vez del programa que lo implementa, cuya funcionalidad principal es el acceso remoto a un servidor de manera segura, cifrando para ello toda la información. Funciona utilizando, entre otros, la pila de protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*) para las capas de transporte y red, respectivamente. El puerto TCP utilizado por defecto para los servicios SSH es el 22. SSH también permite la copia de archivos entre terminales, así como la creación de claves RSA para evitar el uso de contraseñas en cada conexión.

SFTP (*Secure File Transfer Protocol*): es un protocolo que permite la transmisión de ficheros a través de SSH. Este protocolo tiene como antecesor FTP, el cual tenía la misma función pero en este último los datos se transmitían sin cifrar, por lo que eran vulnerables ante posibles atacantes. Para solucionar este problema se creó SFTP. Este protocolo cifra los datos para evitar la vulneración de los mismos.

SSL (*Secure Sockets Layer*): es un protocolo criptográfico que se utiliza para asegurar las conexiones en una red, normalmente Internet. Este protocolo se crea a raíz del crecimiento de Internet, ya que en un principio solo estaba pensado para usuarios confiables donde la información no podía ser vulnerada. Con el crecimiento exponencial de esta red, fue necesaria la creación de protocolos como SSL para asegurar la integridad de los datos. Este protocolo utiliza criptografía asimétrica a través de certificados para autenticar al usuario con el que se está comunicando así como para intercambiar una clave simétrica.

1.4. Estado del arte

Debido a que este trabajo se centra en el protocolo de instalación de un ERP *open source* en un servidor remoto, en esta sección se analizará la situación actual de los principales ERP del mercado que compiten con Odoo. Además,

se presentarán los principales servidores web que permiten a la una máquina gestionar las peticiones y respuestas a través del protocolo HTTP.

1.4.1. Principales ERP

Comenzando por los ERP, existen una gran cantidad de ellos, tanto de código libre, gratuitos o de pago. A continuación, se repasan algunos de los más importantes aparte de Odoo:

- **Holded** [hol22]: es uno de los ERP más conocidos en España. Fue creado por Javier Fondevila y Bernat Ripoll. La principal semejanza con Odoo es su modelo de negocio, que también se basa en SaaS. Aunque es un ERP con una versión gratuita, cuenta también con diversas versiones de pago con las que se aumentan las funcionalidades disponibles. Holded se destaca por su sencillez y facilidad de uso. Al igual que Odoo, su programa se divide en módulos entre los que existen:
 - Facturación: creación de facturas a partir de pedidos de venta, que permiten un flujo de trabajo sencillo y manteniendo una trazabilidad de los documentos otorgados al cliente.
 - Contabilidad: permite gestionar los modelos de impuestos para tu empresa así como los asientos contables y diarios.
 - Proyectos: este módulo permite dirigir los proyectos de manera intuitiva a partir de la creación de tareas y organizándolas a través de un tablero Kanban y diagramas de Gantt.
 - CRM (del inglés, *Customer Relationship Management*): te permite gestionar los clientes potenciales de tu negocio y planificar reuniones o llamadas con cada uno de ellos.
 - Inventario: organiza el *stock* de tu empresa permitiendo diferentes almacenes y pudiendo modificar las características de tus productos.

La diferencia de las versiones se encuentra en el tamaño de operaciones que realice una empresa así como de las funcionalidades que el usuario desee tener. Cada versión permite utilizar más usuarios dentro del *software*, generar más facturas al año, etc. y también ofrece funcionalidades más específicas según la versión. A continuación, se resumen las funcionalidades ofrecidas en cada versión:

- Básico: Facturación y gastos, *cashflow*, CRM, gestión de proyectos y recursos humanos.
 - Estándar: Todo lo del plan Básico, contabilidad, roles de usuario pre-definidos y remesas bancarias.
 - Avanzado: Todo lo del plan Estándar, roles de usuario personalizados, portal de cliente personalizado y recordatorio de pagos.
 - Premium: Todo lo del plan Avanzado, programador de actividades, IVA por país y *account manager*
- **SAP ERP** [sap22]: SAP es una de las empresas más conocidas en el sector de *software* de gestión empresarial. Cuentan con múltiples ERP de pago, adaptándose cada uno a las necesidades o el tipo de empresa que contrata el servicio. Entre estos ERP se encuentran, por ejemplo:
- SAP S/4HANA: es una solución integral del negocio que cuenta como característica principal la conexión con su base de datos SAP HANA (*High Speed Analytical Appliance*). Esta base de datos utiliza inteligencia artificial para ayudar a tomar mejores decisiones de negocio al cliente.
 - SAP Business One: es otra solución integral de negocio la cual se caracteriza por ser la más completa y asequible del repertorio de sistemas de gestión disponibles en SAP. Este *software* te permite gestionar todos los apartados de tu empresa, como son: finanzas, compras, inventario, ventas y relaciones con los clientes, entre otros.

- SAP ByDesign: este ERP se caracteriza por estar integrado bajo la tecnología *on-demand*, que consiste en ofrecer más o menos recursos al cliente dependiendo de las necesidades de este. Por lo tanto, el cliente puede utilizar solo los módulos que necesite para su empresa en vez de contratar una solución total. Además, el servicio se ofrece en modalidad SaaS al igual que Odoo.

- **Oracle Cloud ERP** [ora22]: Oracle es la otra empresa líder del mercado de los ERP junto a SAP. Ambos se encuentran en la cima de la cuota de mercado de los *software* de gestión empresarial [Sta22]. Entre sus clientes, se encuentran algunos tan importantes como el MIT (*Massachusetts Institute of Technology*) o el Banco de América. Oracle solo cuenta con una versión de su ERP, Oracle Cloud ERP. Este *software* ofrece una solución completa a sus clientes que además se puede extender a través de personalizaciones. Entre sus diversas funcionalidades, se destacan las siguientes:
 - Finanzas: este apartado ofrece una visión general de las cuentas a través de un centro de contabilidad, desde el que se pueden revisar facturas, informes, cobros y gastos.
 - Compras: desde el apartado de compras se pueden gestionar los proveedores del negocio y revisar los pedidos de compra que se realizan a estos.
 - Gestión de Riesgos: este apartado se basa en el uso de inteligencia artificial para, a partir de los datos de una empresa recogidos en Oracle Cloud ERP, realizar informes del negocio y recomendar acciones para mejorar los procesos.
 - Gestión de la cadena de suministros: desde este apartado se puede gestionar la logística, el inventario y los procesos de fabricación que se realizan en una empresa.

1.4.2. Odoo

Dentro de los principales ERP, cabe hacer un apartado especial para Odoo. Este *software* es el ERP de código abierto más grande del mercado, utilizado por mas de siete millones de usuarios [Odo23]. Odoo se divide en una parte “core” que cuenta con los módulos por defecto creados por el propio Odoo SA. Por otro lado, cuenta con una comunidad que extiende la funcionalidad de Odoo a través de la creación de miles de módulos para cada versión disponible. A continuación, se presentan algunos de los módulos más importantes de los que dispone Odoo:

- **Gestión de inventario:** permite el control de los productos de la empresa, con lo que se pueden crear nuevos productos, modificar los existentes y borrar los que ya no existan. Además, existen una gran cantidad de datos que se pueden insertar en cada producto para obtener una información detallada de ellos. Estos datos pueden ser: nombre, descripción, referencia interna, foto del producto, precio, si puede ser comprado a un proveedor o si puede ser vendido a clientes o es para uso interno de la empresa, entre tantos otros datos. Además, se pueden añadir más en caso de que se desee, ya que todos los módulos son modificables para adaptarlos a la situación de cada empresa.
- **Ventas:** control de las ventas de los productos de los que dispone la empresa, pudiendo crear facturas, resúmenes e informes. Además, también se cuenta con un módulo de TPV (Terminal de punto de venta) desde el que se pueden gestionar ventas en caso de que la empresa cuente con tiendas y haga ventas directas al público. Con Odoo se pueden hacer integraciones con diversos terminales de pago así como escáneres de códigos de barras para poder ofrecer una automatización completa.
- **Proyectos:** desde este módulo se pueden controlar los proyectos actuales de la empresa, pudiendo crear tareas, asignárselas a un usuario y cambiar el

estado de esta entre todos los que se decidan crear, como por ejemplo: nueva tarea, en progreso, pendiente de *feedback* y finalizada. Además, estas tareas se pueden visualizar en un tablero Kanban, mejorando así la visibilidad de los proyectos.

- **CRM (*Customer Relationship Management*)**: control de los *leads* de la empresa. Como tal no es un apartado que se incluya dentro de los ERP, pero Odoo ha trascendido los aspectos de un ERP tradicional, moviéndose hacia una gestión integral del negocio.
- **Recursos humanos**: gestión de los empleados de la empresa, como son las altas y bajas de estos, sus ausencias o sus horarios.
- **Otros**: Odoo cuenta con muchos otros módulos también importantes. A continuación, se enumeran alguno de ellos: *marketing* por email, gestión de la flota de coches, calendario de eventos, conversaciones entre usuarios y gestión de *tickets* de soporte, entre otros.

1.4.3. Servidores web

Muchos de los ERP que se han mencionado anteriormente, incluyendo Odoo, que es el foco de este proyecto, se ofrecen como SaaS. Para ello, el *software* se instala en un servidor web para permitir a los usuarios acceder a través de una red o de Internet.

Existen múltiples tipos de servidores web que permiten manejar las peticiones entrantes de uno o varios clientes y devolver una respuesta acorde a ella. A continuación, se hace un repaso de algunos de los servidores web más importantes del mercado en el momento de redacción de este TFG:

- **Apache**: es un servidor web de código abierto desarrollado por la Apache Software Foundation y creado en 1995. Desde su nacimiento, ha sido el servi-

dor web más utilizado, llegando a ser utilizado por el 70% de los sitios web en 2005. En la actualidad su popularidad ha decaído debido a la aparición de otros servidores web que se verán a continuación. No obstante, Apache sigue estando en la cumbre de los servidores web contando con alrededor del 23% de la cuota de mercado. Las principales ventajas con las que cuenta Apache es su modularidad y, sobre todo, el hecho de ser un servidor web de código abierto. Esto permite su extensión y, a su vez, ha generado una gran comunidad a su alrededor, con lo que siempre se puede obtener ayuda en caso de que surja un problema desconocido. Además, la fundación Apache mantiene un soporte constante, lanzando nuevas actualizaciones del *software* periódicamente.

- **Nginx**: es un servidor web ligero de alto rendimiento, así como un *proxy* para protocolos de correo electrónico lanzado en diciembre de 2004 y desarrollado por Igor Sysoev. Al igual que Apache, se trata de un *software* libre y de código abierto, aunque también cuenta con una versión comercial llamada Nginx Plus. Nginx se ha convertido en la actualidad en el servidor web más utilizado. Entre sus usuarios se encuentran sitios web tan famosos como Wordpress, Netflix o GitHub. Su gran aceptación en el mercado se debe a que cuenta con las siguientes características:
 - Servidor de archivos estáticos, índices y autoindexado.
 - *Proxy* inverso con opciones de caché.
 - Equilibrio de carga, que consiste en repartir el trabajo entre varios ordenadores o procesos a través del uso de algoritmos que calculan como distribuir la carga de trabajo de la forma más equitativa posible entre todos los recursos disponibles.
 - Soporte de HTTP y HTTP2 sobre SSL.
- **Microsoft IIS (*Internet Information Services*)**: es el servidor web diseñado por Microsoft y lanzado en 1995 con el objetivo de integrarse

en los sistemas operativos diseñados por la compañía, como son Windows 2000, Windows XP, Windows 7 o Windows Server 2003, 2006 y 2019, entre muchos otros. IIS ha contado con múltiples versiones, desde la 1.0 hasta la 10.0, en las que se han ido añadiendo nuevas funcionalidades y soporte con los sistemas operativos Windows que han sido lanzados durante los años. Microsoft IIS llegó a contar con una gran popularidad unos años atrás pero ha decaído en gran medida en los últimos dos años.

- **OpenResty**: es el tercer servidor web más utilizado en la actualidad. Está basado en Nginx y fue desarrollado por Yichun Zhang. Cuenta con un gran uso en China, siendo patrocinado por un tiempo por la gran empresa china Taobao y mantenido desde 2012 hasta 2016 por la empresa Cloudflare, la cual también cuenta con su propio servidor web. En la actualidad, OpenResty es mantenido por la OpenResty Software Foundation.

En la Tabla 1.1 se puede observar una comparativa de los servidores web más utilizados que acaban de ser presentados. En la comparativa se observa como Nginx es el servidor web más utilizado actualmente, seguido de cerca por Apache. En la Figura 1.1 (tabla y gráfica extraídas de un estudio de Netcraft a fecha de junio de 2022 [Net22]) se puede observar una comparativa histórica de los servidores, en la que se puede ver el gran crecimiento que tuvo Apache respecto al resto de servidores y como empezó a descender su popularidad a partir de 2013, cuando otros servidores como Nginx empezaron a ganar fama. También se puede destacar la popularidad de los servidores web de Microsoft, llegando a tener una cuota de mercado de hasta el 50% a principios del año 2018 y que, sin embargo, ha sufrido una gran caída durante los últimos años hasta plantarse en un 3% de cuota de mercado en la actualidad. Por otro lado, se debe destacar la diversidad de servidores web durante los últimos años haciendo que un 21% de la cuota de mercado en la actualidad esté compuesta por múltiples servidores web menos conocidos y que, por sí solos, no representan un gran porcentaje.

Tecnología	Nº sitios web	Porcentaje (%)
Nginx	354.871.628	30,71
Apache	265.688.420	22,99
OpenResty	92.848.366	8,03
Cloudflare	64,369,545	5,57

Tabla 1.1: Comparativa de los servidores web más utilizados, extraído de Netcraft [Net22].

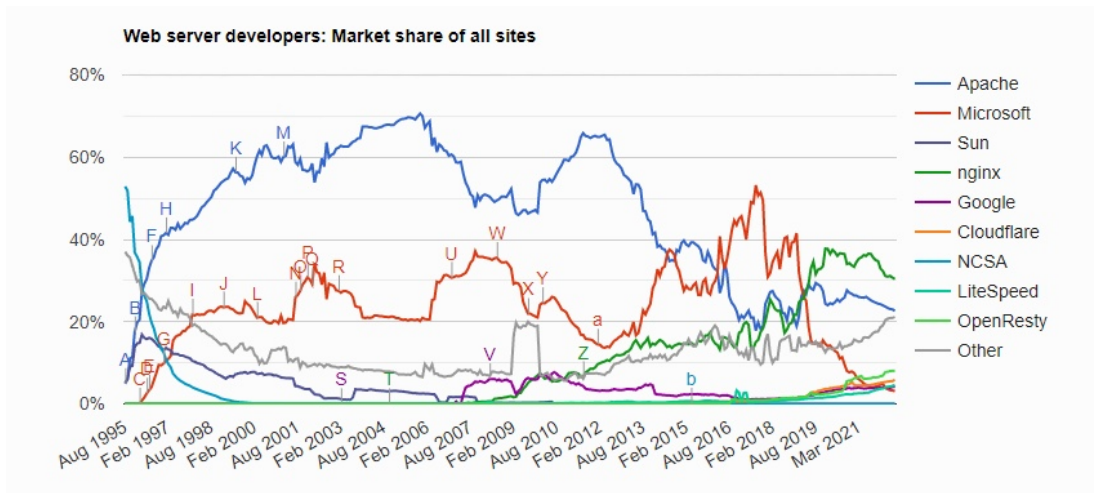


Figura 1.1: Cuota de mercado de servidores web, extraído de Netcraft.

1.5. Estructura de la memoria

En lo que respecta al resto de esta memoria, en el Capítulo 2, se presentan los objetivos de este TFG. En el Capítulo 3, se aborda el desarrollo de un protocolo para la implantación del sistema a un nuevo cliente, mientras que en el Capítulo 4, se tomará una metodología adaptada para los desarrollos que se realicen del proyecto. El Capítulo 5 versa sobre las tecnologías utilizadas en el proyecto y el desarrollo de las funcionalidades que se describen en la creación del protocolo. Posteriormente, en el Capítulo 6, se analizan los resultados obtenidos a partir de los desarrollos realizados en el capítulo anterior. Por último, en el Capítulo 8, se presentan las conclusiones del proyecto, así como diversos posibles trabajos futuros.

2

Objetivos

En este capítulo se presenta el objetivo general (ver Sección 2.1) de este proyecto así como los objetivos específicos (ver Sección 2.2) los cuales guiarán el desarrollo en pequeños pasos hasta obtener el objetivo principal.

2.1. Objetivo general

El objetivo general de este trabajo de fin de grado es el siguiente:

“Elaborar un protocolo para la implantación del ERP Odoo en un cliente, adaptándolo a las necesidades de negocio y proporcionando pautas y herramientas que consideren la seguridad y el rendimiento del sistema.”

2.2. Objetivos específicos

A continuación, se presentan los objetivos específicos que ayudarán a conseguir el objetivo general de este proyecto:

- Revisar la literatura sobre configuración de servidores e implantación de sistemas ERP.
- Definir los pasos a seguir en el protocolo de implantación del sistema.
- Configurar el acceso al servidor mediante un par de claves pública-privada.
- Crear un *script* de instalación de una instancia Odoo y de una base de datos PostgreSQL.
- Instalar instancia de Odoo en el servidor.
- Crear copias de seguridad automatizadas de la base de datos que se manden a un servidor externo.
- Cifrar y certificar un dominio para acceder al servicio Odoo.
- Configurar medidas de seguridad en el servidor.
- Elaborar una documentación que recoja el proceso de principio a fin.
- Realizar pruebas del proceso en un servidor.
- Revisar la configuración.

3

Protocolo de implantación de un ERP

En este capítulo, se desarrolla la propuesta de un protocolo para la correcta implantación de un ERP a un nuevo cliente. De manera general, en la Figura 3.1 se puede observar una vista general de las fases de las que consta el protocolo de implantación de un ERP. En este capítulo se repasan cada una de las fases de manera detallada.

3.1. Contacto inicial

El primer contacto con un potencial cliente (también conocido como *lead*), interesado en adquirir el servicio de Odoo, se puede producir a través de cualquier

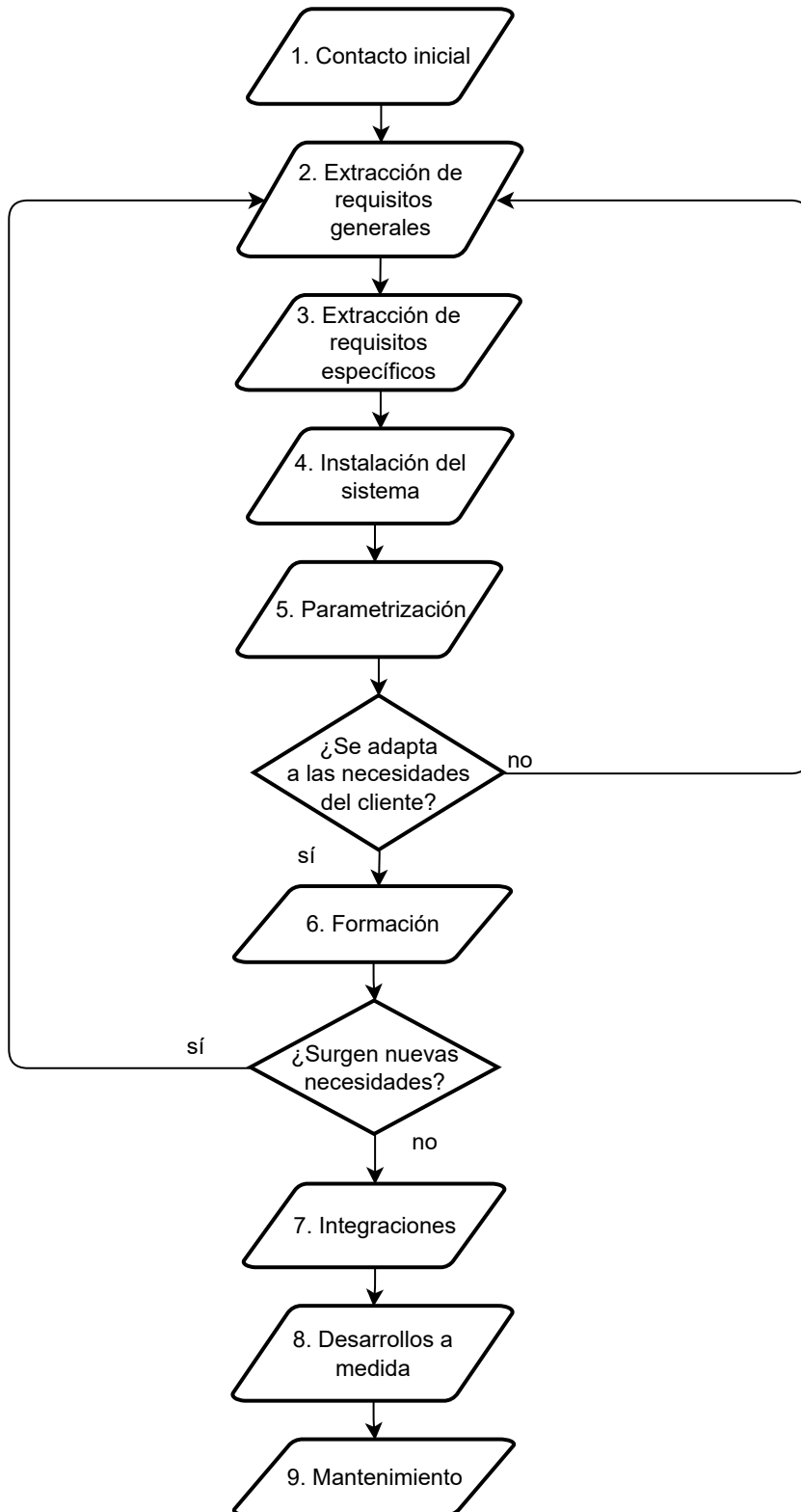


Figura 3.1: Diagrama de actividad con las fases existentes en el protocolo de instalación.

canal: telefónico, formulario web, *networking* en eventos... Dependiendo del canal de llegada se pueden tener más o menos datos del mismo. El factor común siempre será que, al menos, se obtienen los datos de contacto.

Una vez obtenidos los datos de contacto, se procede a mantener una llamada telefónica con el interesado. En esta llamada se tratará de averiguar información general de la empresa, así como información sobre los principales procesos de gestión o negocio, que son vitales para afrontar el comienzo de la implantación. Desde la parte del implantador interesa conocer los siguientes datos.

- Contexto/Datos generales: contexto en el que trabaja la empresa. Dentro de este contexto existen varios puntos:
 - Nombre de la empresa y NIF.
 - Actividad económica: cuál es la actividad comercial de la empresa.
 - Ubicación: donde se encuentran las oficinas/fábricas de la empresa.
- *Software*: *software(s)* actual(es) que utilizan para la gestión de los ámbitos de la empresa.
- Empleados:
 - Empleados totales de la empresa.
 - Empleados que serán usuarios del sistema.
- Facturación:
 - Facturación anual de la empresa.
 - Volumen de ventas anual.
- Número de compañías: puede tratarse de una sola empresa o un grupo de varias empresas.

Una vez se hayan obtenido todos estos datos, se puede determinar si la solución que se ofrece con Odoo se adapta a las necesidades del contacto interesado. Por lo general, Odoo siempre se adapta a las necesidades de cualquier empresa, desde una perspectiva técnica, ya que es un *software* muy flexible gracias a la libre personalización que ofrece junto a la multitud de desarrollos que ya existen. No obstante, será necesario también realizar una propuesta económica y que esta sea aceptada por el potencial cliente.

3.2. Extracción de requisitos generales

Si el contacto interesado acepta la propuesta y se convierte en cliente, se pasa a la siguiente fase. En esta fase se tratan, a grandes rasgos, las necesidades de la empresa. Es decir, se analizarán las áreas temáticas de trabajo sobre las que trabaja el cliente en cuestión, y que serán incluidas y adaptadas dentro de Odoo para que puedan trasladar su flujo a este *software*. En esta fase, solo se trata de identificar las áreas sobre las que se trabajarán, mientras que en la siguiente fase se analizará cada área en mayor detalle para extraer requisitos específicos (ver Sección 3.3).

A continuación, se van a repasar detalladamente las principales áreas temáticas que existen en Odoo, las cuales se pueden apreciar resumidas en la Figura 3.2.

- **CRM:** una empresa que trabaja con múltiples *leads* (potenciales clientes) y oportunidades puede interesarle usar el módulo de CRM. Dentro de este apartado, es de interés conocer algunos aspectos como:
 - ¿Qué datos de interés se quieren almacenar de cada *lead*?
 - ¿Dispone la empresa de un sitio web donde ofrecen sus productos o servicios y se puede contactar a través de un formulario?

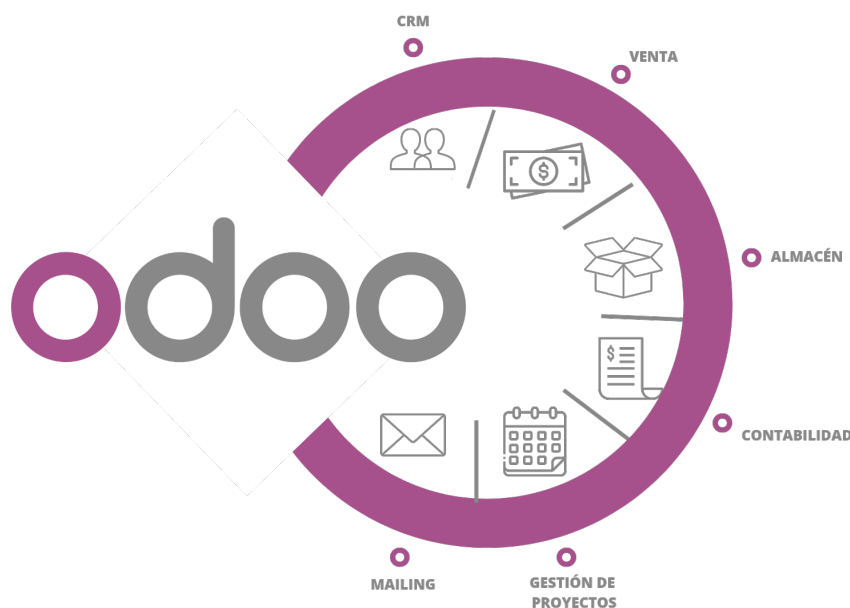


Figura 3.2: Principales áreas temáticas de Odoo.

- ¿La empresa cuenta con múltiples comerciales para gestionar las oportunidades?
 - ¿Organizan sus leads y oportunidades en diferentes fases o etapas utilizando un flujo de venta?
 - Cuando se gane una oportunidad, ¿convierten ese lead en un contacto frecuente?
- **Venta:** el apartado de venta es uno de los más utilizados por las empresas. Este apartado se entrelaza estrechamente tanto con el área de CRM como con las áreas de almacén y contabilidad que se verán a continuación. A partir de un *lead* o de un contacto, se pueden crear pedidos de venta utilizando los productos o servicios configurados previamente en Odoo o creados sobre la marcha si se trata de un producto único para ese cliente que no existía previamente, como por ejemplo un vestido hecho a medida. Para el uso de este módulo, interesa conocer de la empresa, entre otros, los siguientes datos:
- ¿Qué tipos de productos venden?

- ¿Qué unidades de medida se utilizan para los productos?
 - ¿Qué métodos de pago utilizan?
 - ¿Existen tarifas de precios según las características de cada cliente?
 - ¿Venden a través de algún *e-Commerce* o *Marketplace*?
 - ¿Tienen distintos tipos de envíos?
- **Almacén:** para gestionar la venta de productos, es necesario tener un control del inventario. A través del módulo de almacén o inventario se puede conseguir esto. Es interesante conocer, entre otros, los siguientes datos:
- ¿La empresa cuenta con uno o varios almacenes?
 - ¿Los productos cuentan con una referencia interna?
 - ¿Venden a través de algún *e-Commerce* o *Marketplace*?
 - ¿Trabajan con algún transportista?
 - ¿Realizan entregas parciales de algún pedido?
- **Facturación y contabilidad:** la mayoría de las empresas utilizan algún *software* de contabilidad. Odoo ofrece la posibilidad de gestionar la contabilidad de la empresa, estando directamente conectado con el apartado de facturación, facilitando así los procesos. Interesa conocer lo siguiente para implementar correctamente el área de contabilidad:
- ¿Cómo es su flujo de facturación?
 - ¿Cuál es el formato de las facturas?
 - ¿Desean automatizar la generación de facturas?
 - ¿Qué tipo de plan contable utilizan?
 - ¿Qué impuestos son los más utilizados?
 - ¿Qué diarios contables utilizan?
 - ¿Utilizan libro mayor?

- **Proyectos:** Odoo ofrece la posibilidad de gestionar proyectos de manera intuitiva. Se necesita conocer lo siguiente para poder implementarlo:
 - ¿La empresa trabaja con un enfoque basado en proyectos? ¿Cuáles?
 - ¿Organizan las tareas de cada proyecto en distintas fases o etapas?
 - ¿Cuentan con tareas recurrentes?
 - ¿Las tareas son asignadas a un solo usuario o a un equipo?

- **Mailing:** dentro de Odoo también destaca la gestión de campañas de correo electrónico. Para utilizar este apartado del *software* se necesita conocer:
 - ¿Cuentan con listas de contactos para dividir las campañas de correo?
 - ¿Utilizan algún *software de mailing* actualmente?
 - ¿Cuentan con una suscripción a la *newsletter* dentro de su sitio web?

Estos son, a grandes rasgos, los detalles más importantes que se deben conocer del cliente sobre cada área que utilizan en su empresa para saber si implementar estas dentro de Odoo y, si es así, qué detalles fundamentales de cada módulo se deben configurar y cuáles se pueden obviar.

3.3. Extracción de requisitos específicos

En esta sección se analizarán las áreas temáticas que se haya han presentado anteriormente y cuya utilidad para el cliente se ha detectado en la fase anterior.

Para cada área de interés a implementar dentro de Odoo se debe realizar una reunión con la persona responsable en la empresa en la que se analice en profundidad los procesos de la empresa asociados a la misma. A partir del análisis de estos procesos, se extraerán requisitos específicos que son únicos para cada empresa. Por ejemplo, en el caso del apartado de venta, si un cliente cuenta con

un *e-Commerce* desarrollado en Prestashop, se deberá configurar el conector de Odoo con Prestashop. Una vez configurado, se analizará en profundidad el flujo que tienen en este *e-Commerce*. Puede darse el caso de que tengan características de interés para cada producto o que tengan un apartado de productos relacionados en función del producto que este visualizando el comprador. Estas características específicas se recogerían en la reunión que se mantiene con el cliente y deberán ser posteriormente implementadas o configuradas en Odoo correctamente. Estas reuniones no se producen una única vez. Se realiza una reunión inicial como punto de partida y, posteriormente, se realizan reuniones iterativas. A medida que se van satisfaciendo los requisitos y presentando al cliente para su validación, pueden surgir nuevos requisitos específicos.

3.4. Instalación del sistema

En esta fase se procede a instalar el sistema. Según los requisitos que se hayan extraído en la fase anterior la instalación puede variar en algunos aspectos. Particularmente, cada instalación puede contener un número de módulos diferente según las necesidades del cliente. Esta fase se trata del apartado más técnico del protocolo y se desarrollará en profundidad en el Capítulo 5.

3.5. Parametrización

Una vez instalado el sistema, se procede con la parametrización de cada uno de los módulos disponibles en función de las necesidades extraídas del cliente en las fases anteriores (veáanse Secciones 3.2 y 3.3). Al tratarse Odoo de un *software* extenso, existen multitud de parámetros configurables en cada una de sus áreas temáticas. Dentro de todas estas configuraciones se pueden destacar las siguientes:

- **Migración de información:** si la empresa ya contaba con un *software* de

gestión anterior a Odoo, se suelen importar todos los datos procedentes del *software* antiguo. Este proceso es tedioso ya que hay que adaptar los datos al formato de Odoo.

- **Permisos de cada usuario:** cada usuario dispondrá de unos permisos que le permitirán acceder o no a las secciones de Odoo en función de su ocupación. Por ejemplo, si se trata de un comercial, solo tendrá accesos a los documentos y contactos que él haya creado, impidiéndole acceder a información de otros usuarios que no debería poder ver.
- **Facturación y contabilidad:** el módulo más configurable de Odoo es el apartado de facturación y contabilidad. Dentro de este módulo se pueden adaptar el formato de las facturas, los impuestos, cuentas y asientos contables o libros mayores de empresa, entre otros. Esto se realizará según las directrices del cliente así como de las limitaciones que supone la legislación de cada país. La configuración no será la misma para una empresa pequeña o mediana que para una gran empresa. Así como tampoco será igual la configuración de una empresa que opere en un único país, o que opere de manera internacional.
- **Ventas:** dentro del apartado de ventas es importante una correcta configuración de las tarifas de ventas, según la localización del cliente, ofertas temporales existentes, *packs* de productos, etc. Además, también se deben configurar todos los métodos de pago existentes y tipos de envío según la región del comprador.

3.5.1. Formación

Una vez se tiene configurada la base del proyecto, se puede empezar a trabajar con Odoo. Para ello, es clave proporcionar una formación básica a los usuarios de la empresa que vayan a utilizar el *software*. Odoo cuenta con infinidad de

apartados y opciones, lo cual hace que su curva de aprendizaje sea elevada al principio. Si no se proporciona una correcta formación al inicio, los usuarios pueden frustrarse al no saber utilizar correctamente el *software* o no sacar el máximo potencial que ofrece. La formación se puede dar de distintas maneras, pero se destacan las siguientes tres:

- **Píldoras formativas:** a través de vídeos de corta duración se realizan explicaciones de los distintos flujos que se pueden realizar en Odoo. Por ejemplo, un vídeo consistiría, dentro del apartado de venta, en el flujo de venta de un pedido a un cliente. Primero se crearía, un presupuesto, después al confirmarse se convertiría en pedido de compra, se crearían los albaranes y una vez entregados estos se crearía la factura. Esta factura posteriormente se puede marcar como pagada. Como este ejemplo, se podrían crear decenas de píldoras de cada apartado básico a destacar de Odoo y ofrecérselo al cliente para que puedan avanzar a su propio ritmo.
- **Videoconferencias:** los flujos mencionados en el apartado anterior también se pueden explicar a través de videoconferencias. Serían reuniones de una mayor duración donde se agruparía la explicación de varios conceptos que, a poder ser, estuviesen relacionados. La principal ventaja de este formato es la opción de poder dar retroalimentación inmediata ante cualquier duda que pueda tener alguno de los usuarios. La principal desventaja es que, a veces, puede ser difícil reunir a varias personas de una empresa en un mismo momento.
- **Reuniones presenciales:** a veces la formación se da de manera presencial en las oficinas o fábricas del cliente. El formato sería idéntico al de las videoconferencias. En este caso, la retroalimentación puede ser más fluida y se puede interactuar de manera más sencilla con los usuarios que reciban la formación en caso de que tengan algún tipo de dificultad. Además, en el caso de que el cliente tenga alguna peculiaridad en alguno de sus flujos,

estos se podrían replicar con exactitud al encontrarse en el entorno real de trabajo.

3.6. Integraciones

En el caso de que sea necesaria alguna integración con algún *software* externo a Odoo, se procederá a ello en esta etapa. Las integraciones permiten sincronizar datos de interés entre ambas aplicaciones, como pueden ser facturas, *stock*, empleados, etc. Algunas de las integraciones más comunes son:

- **Plataformas de *e-Commerce***: integraciones con plataformas de *e-Commerce*, como por ejemplo Prestashop, WooCommerce o Shopify. Se sincronizan principalmente pedidos, productos y *stock*. Esta sincronización permite el correcto funcionamiento de ambos *softwares* y es vital para que no se produzcan desviaciones entre datos que deberían ser iguales. Por ejemplo, un error en la sincronización del *stock* entre ambas aplicaciones podría suponer la venta de un producto del que realmente no hay existencias.
- **Calendarios**: las integraciones con aplicaciones como Google Calendar y Calendly permiten sincronizar los calendarios con el calendario de Odoo, creando así las reuniones y contactos existentes en el *software* externo.

3.7. Desarrollos a medida

En el caso de que los módulos existentes y/o su parametrización, no permitan satisfacer las necesidades del cliente, en esta etapa se producen los desarrollos de módulos a medida en Odoo. Por lo general, se aconseja al cliente intentar adaptarse en la medida de lo posible a las opciones que ofrece el *software* de serie, ya que el mantenimiento de código a medida es normalmente costoso. Además, muchas

veces los desarrollos se podrían evitar con una adaptación por parte del cliente de sus flujos de trabajo y estos se producen por una resistencia al cambio para seguir utilizando los flujos que existían en la empresa previo a la implantación de Odoo. Sin embargo, también se dan desarrollos que son estrictamente necesarios. Muchas empresas cuentan con necesidades únicas por la actividad que realizan, por eso es de gran importancia analizar el contexto del cliente en el contacto inicial y las fases posteriores (ver Secciones 3.1, 3.2 y 3.3).

3.8. Mantenimiento

La última etapa es el mantenimiento. Esta fase se alarga durante toda la vida útil del proyecto. El mantenimiento de estos sistemas implica la actualización de módulos, tanto de terceros como de creación propia, y versiones del *software*, las copias de seguridad periódicas de la base de datos, limpieza de datos obsoletos, etc. El mantenimiento es una fase vital del proyecto, en la que se debe evitar aumentar la complejidad del sistema y mantener una experiencia de usuario satisfactoria.

4

Metodología

En este capítulo se abarcará el uso de las metodologías para el desarrollo de este TFG. Es cierto que las metodologías se enfocan en un marco de desarrollo de *software* y están pensadas para desarrollar un producto para un cliente con el que hay que mantener un contacto constante. En un trabajo de estas características no se desarrolla para ningún cliente y se trata, más bien, de un proyecto de investigación.

Sin embargo, esto no quita que el uso de una metodología adaptada a las necesidades de este proyecto sea realmente útil para poder gestionar las diversas tareas con un control adecuado. Mejorando así la gestión del trabajo y, por lo tanto, mejorando la eficiencia y reduciendo los tiempos de trabajo.

A continuación, se repasa Scrum, la principal metodología de desarrollo *software* ágil, ya que esta es la que más se adapta a las necesidades de este proyecto.

4.1. Scrum

Scrum es el marco de desarrollo ágil más conocido. Trata de aplicar un conjunto de buenas prácticas además de eventos y roles. Este marco tiene sus inicios en los años 80, de la mano de Ikujiro Nonaka y Takeuchi tras analizar cómo las empresas tecnológicas desarrollaban sus productos. En su estudio, comparaban la forma de trabajar con las melés de rugby, de ahí el nombre de Scrum, que significa melé en inglés.

Scrum llegaría al mundo *software* sobre el año 1995, con Ken Schwaber, que redactó *Scrum Development Process* en el OOPSLA 95 (*Object-Oriented Programming Systems & Applications conference*) [Sch22], donde se define un marco de reglas para desarrollo *software*, que se basa en los principios Scrum y que utilizó él mismo en sus desarrollos al igual que Jeff Sutherland.

Las principales características de Scrum son las siguientes:

- El *software* debe ser iterativo e incremental.
- Se prioriza siempre lo que aporta mayor valor al cliente.
- El equipo se debe sincronizar diariamente.
- Tras cada iteración se muestra al cliente el resultado obtenido.
- Equipos pequeños, entre 3 y 9 personas, y “multifuncionales”.

Para conseguir estas características, Scrum aplica un conjunto de buenas prácticas, roles y eventos.

Lo primero que se puede destacar son los roles:

- **Scrum Master:** es el integrante del equipo que debe asegurar que se cumplan y apliquen las buenas prácticas que propone Scrum.
- **Product Owner:** es el integrante del equipo que representa a los *stakeholders* (personas interesadas en el producto a desarrollar).
- **Equipo:** son el resto de desarrolladores que conforman el equipo.

Por otro lado, Scrum propone el uso de *sprints*. Los *sprints* son pequeñas iteraciones de tiempo, entre 1 y 4 semanas, donde el equipo desarrolla un incremento funcional del *software* que se está desarrollando.

Todas las funcionalidades que se desarrollarán en cada iteración vienen definidas en el *Product Backlog*. En este, se ordenan, priorizando por valor, las funcionalidades que se van a desarrollar en el proyecto. Posteriormente, las funcionalidades que se van a realizar en un *sprint* en particular se incluyen en el *Sprint backlog*. Estas funcionalidades, divididas en tareas, que se van a incluir dentro del *Sprint Backlog* se deciden en una reunión denominada *Sprint Planning* que se define al principio de cada *sprint*. En la Figura 4.1 se puede observar un ejemplo de *Product Backlog* y *Sprint Backlog*.

El *Sprint backlog* se compone de varias columnas para poder ordenar las tareas. Las columnas pueden variar pero principalmente existen las siguientes:

Con esto y otras buenas prácticas, Scrum intenta conseguir equipos “autoorganizados” y “multifuncionales”, con lo que se consigue mejorar los procesos de desarrollo *software* ya que no se depende de nadie externo al equipo para ninguna tarea.

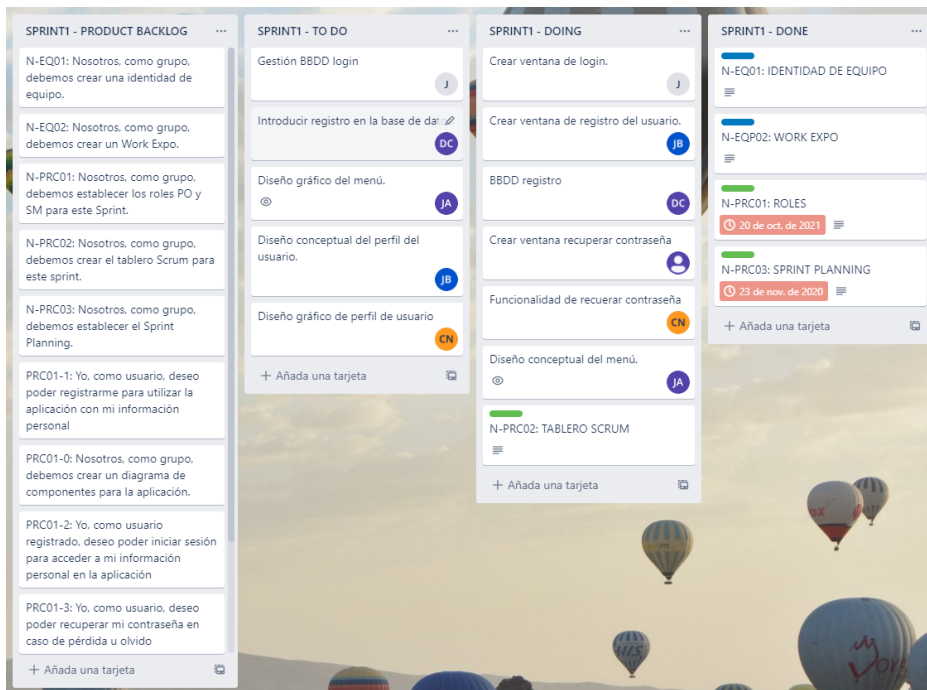


Figura 4.1: Ejemplo de *Product Backlog* y *Sprint Backlog*.

4.2. Metodología elegida

Para el desarrollo de este TFG se ha decidido utilizar aquellos aspectos de Scrum repasados anteriormente que se consideran útiles y adaptar aquellos que se consideran útiles pero necesitan algunos cambios para que se adecúe a la forma de trabajar en el desarrollo de este proyecto el cual no es un desarrollo de *software* para un cliente, que es para lo que está diseñada originalmente esta metodología que se acaba de repasar.

Por lo tanto, se ha decidido tomar un marco de trabajo “nuevo” que se basa en una mezcla de buenas prácticas de diversos modelos de desarrollo, tomando los aspectos que se han considerado útiles para la realización de este proyecto.

Uno de ellos, por ejemplo es uso del tablero de Scrum, utilizando una adaptación del *Product backlog* y el *Sprint Backlog*, para ordenar las tareas por el valor que ofrecen al desarrollo y repartiéndolas en varios *sprints*. Con el uso de los *sprints* se consigue mantener una evolución controlada, pudiendo corregir po-

sibles problemas o errores a tiempo. Para este proyecto se ha decidido tomar *sprints* de 4 semanas, lo que supone 20 días de trabajo, dedicando un promedio de 1,5 horas por día al desarrollo de este proyecto. Esto representa un total de 35 horas de trabajo aproximadamente por *sprint*.

Aunque no existe un cliente, sí que existe un tutor del TFG con el cual se debe mantener una comunicación constante para recibir *feedback* y asegurar que se está siguiendo el camino correcto. Al final de cada *sprint* se realizará una reunión de seguimiento para repasar los avances mientras que durante el desarrollo del *sprint* se mantiene contacto para preguntar posibles dudas que puedan surgir.

El reparto de las funcionalidades o requisitos que se quieren conseguir se realizará dividiendo estas en historias técnicas de menor tamaño. Estas funcionalidades en realidad son los objetivos específicos que se han definido previamente. Algunos son más grandes que otros por lo que mientras que algunos objetivos se pueden identificar con épicas (funcionalidades grandes que engloban muchas tareas o historias de usuarios) otras se dividirán en un par de tareas o, incluso, será una sola tarea.

Para realizar un buen seguimiento de cada historia técnica se realizarán unas anotaciones sobre cada una. Para ello, se utilizará la siguiente tabla:

- Alias de la tarea: el nombre irá acompañado de un identificador para cada tarea. Este identificador seguirá el patrón "HT- número de la tarea, por ejemplo: HT01, HT02, ...
- Enunciado de la historia técnica: con el formato:
 - Como (rol),
 - quiero ... (realizar una acción)
 - para ... (conseguir un objetivo).
- Descripción de la tarea.
- Tiempo empleado en la tarea: vendrá expresado en horas

- Complejidad de la tarea: a continuación se explica cómo se realiza el cálculo de esta complejidad.
- Subtareas: en caso de existir, seguirán el patrón: "HT- ".- número de la tarea + ".- número de la subtarea. Por ejemplo: HT.1.01.
- Conversación: en este apartado se profundizará en cómo ha sido el desarrollo de la tarea.

En el desarrollo de este proyecto, de manera semejante a los desarrollos *software*, es muy difícil estimar el tiempo que va a llevar una tarea. Esto se debe a que pueden ocurrir problemas no esperados en el desarrollo de esta que supongan un mayor tiempo dedicado aunque en un primer momento la tarea no aparentase ser compleja. No obstante, sí que puede ser útil realizar una vaga aproximación de la complejidad de la tarea para poder encuadrarlas mejor en cada *sprint*. Para realizar esta estimación se va a utilizar la técnica de las tallas de camisetas. Es una técnica que se asemeja al Planning Poker [Com22]. Se disponen 6 tallas de camisetas: XS, S, M, L, XL y XXL, de manera que la talla XS representa la menor complejidad posible de una tarea y la talla XXL la mayor complejidad posible. A cada tarea se le asignará una talla en función de la complejidad que se le estime. Esta técnica no pretende ser demasiado precisa, sino que pretende aportar una primera aproximación a cada tarea de una manera rápida y sencilla.

5

Instalación del sistema

En este capítulo se repasarán todos los aspectos relacionados con la instalación de Odoo en el servidor, lo cual también incluye las distintas configuraciones que ya se han repasado en el capítulo de objetivos (ver Sección 2.1 y 2.2). Para ello, no solo se debe indagar en cada tarea, sino también en la correcta disposición de las mismas en los diferentes *sprints* y en las tecnologías a utilizar.

5.1. Tecnologías utilizadas

Primero, se van a mencionar las distintas tecnologías a utilizar para realizar la implantación de Odoo en el servidor.

5.1.1. Odoo

Como ya se ha mencionado varias veces durante esta memoria, Odoo es el ERP elegido para realizar la implantación. Al igual que Apache, es un *software* de código abierto y es el elemento clave sobre el que girará la instalación del servidor. En el momento de la realización de este TFG, Odoo se encuentra en la versión 16, pero se utilizará la versión 14 debido a su madurez en cuanto a número de módulos. Aun así, la instalación de otra versión, no supone una gran variación en el procedimiento. Odoo funciona sobre Python. En este caso la versión de Python será la versión 3.9.

5.1.2. Ubuntu

Ubuntu es la tecnología elegida para ser el sistema operativo. Ubuntu es una distribución de Linux, el sistema operativo que se utiliza en la mayoría de servidores a nivel global. Ubuntu es una opción de *software* libre y de código abierto. Se usará para la instalación la versión 20.04 de Ubuntu, debido a que es una versión estable, ampliamente utilizada y con soporte a largo plazo.

5.1.3. Apache

La tecnología escogida para el servidor web será Apache. Como ya se pudo ver previamente (ver Sección 1.4), Apache es uno de los servidores web más utilizados y, aunque su uso ha decaído en la actualidad en comparación con años atrás, sigue siendo una excelente opción para ser el servidor web de esta implantación debido a ser una opción de *software* libre, de código abierto y a la gran comunidad que se ha creado durante todos los años que este *software* ha estado operativo. Se utilizará la versión 2.4 de Apache.

5.1.4. PostgreSQL

La opción para la base de datos es PostgreSQL. Odoo utiliza PostgreSQL como base de datos debido a las posibilidades que esta ofrece. PostgreSQL es una base de datos relacional orientada a objetos y, al igual que Odoo, también de código abierto. La única elección que hay que realizar en cuanto a la base de datos es qué versión de PostgreSQL utilizar. En este caso, se utilizará la versión 12 de PostgreSQL para la instalación.

Entre las diferentes características que ofrece PostgreSQL se encuentran:

- Alta concurrencia.
- Gran escalabilidad.
- Amplía variedad de tipos de datos nativos.
- Buena escalabilidad.
- Herencia de tablas.
- Integridad transaccional.

5.2. *Product Backlog*

A continuación, se presenta la organización de las diferentes tareas que se han realizado en el desarrollo de este proyecto en sus correspondientes *sprints*. Para una mejor visualización de los datos, se presenta (a modo de *Product Backlog*) una tabla que incluye identificador, nombre, descripción, complejidad y *sprint* de la tarea (véase Tabla 5.1).

Identificador (ID)	Nombre	Descripción	Complejidad	Sprint
HT01	Configurar acceso a la máquina por SSH	Configurar claves SSH con el servidor para evitar introducir la contraseña en cada conexión	M	1
HT02	Configuración básica del servidor	Configurar los aspectos básicos del servidor e instalar los paquetes necesarios para la instalación de Odoo	M	1
HT03	Instalación de Odoo en el servidor	Instalar instancia de Odoo 14 junto a la base de datos PostgreSQL 12	L	1
HT04	Redirigir, cifrar y certificar un dominio para el servidor	Cifrar y certificar un nuevo dominio para el servidor utilizando Certbot y la configuración de Apache	L	2
HT05	Configurar copias de seguridad	Crear copias de seguridad que se manden automáticamente a un servidor externo	L	2 y 3
HT06	Configurar medidas de seguridad	Configurar varios aspectos de la máquina para evitar intrusiones	M	3

Tabla 5.1: Product Backlog con las tareas a desarrollar.

5.3. Tareas detalladas

En esta sección se profundiza en el desarrollo de todas las tareas mencionadas la Sección 5.2.

HT01 – Configurar acceso remoto al servidor

- **Enunciado de la historia técnica:** como desarrollador, quiero configurar una conexión SSH al servidor para poder acceder remotamente.
- **Descripción:** configurar claves SSH con el servidor para evitar introducir la contraseña en cada conexión.
- **Complejidad:** M
- **Tiempo empleado:** 8 horas.
- **Subtareas:**
 - HT.1.01: acceder a la máquina.
 - HT.1.02: crear claves pública y privada.
 - HT.1.03: mandar clave privada a la máquina local.
- **Conversación:** en el punto de partida se presupone que se dispone de un superusuario (comúnmente *root*) y una contraseña para poder acceder a la máquina que en este caso tendrá instalado el sistema operativo Ubuntu 20.04. Si no se es el usuario *root* pero se cuenta con permisos de *sudo*, se ejecuta el siguiente comando (véase Código 5.1) para poder trabajar como superusuario:

```
1 sudo su
```

Código 5.1: Comando para ser superusuario *root*.

Una vez registrados como superusuario, se debe crear la clave pública y privada con el comando `ssh-keygen`. La clave pública, la cual funciona

a modo de cerradura para la clave privada, se almacenará en el archivo “`authorized-keys`”, que debe existir y pertenecer a `root`. Posteriormente, se debe modificar el archivo “`sshd_config`” para indicar que el archivo donde se almacenará la clave pública es “`authorized-keys`”. Por último, reiniciamos el servicio de SSH.

Una vez creadas la clave pública y privada se debe enviar la clave privada a la máquina local desde la que se quiere acceder. Esto se puede realizar de diversas maneras: a través de Filezilla, SFTP o SCP. Una vez se tiene la clave privada en la máquina local y almacenada en la carpeta “`/.ssh`”, se deben cambiar los permisos del archivo. Se le otorgan solo permisos de lectura al propietario del archivo, el cual será (o se cambiará a) el usuario con el que normalmente se trabaje.

Finalmente, se deben añadir las siguientes líneas (véase Código 5.2) al final del archivo “`ssh_config`” para indicar los datos de la conexión y el alias que se le otorgará la conexión SSH en cuestión:

```
1      Host Odo014NOMBRE_SERVIDOR
2      Hostname XXX.XXX.XXX.XXX
3      User root
4      Port XXXXX
5      IdentityFile /home/NOMBRE_USUARIO/.ssh/id_rsa_XXX.
      XXX.XXX.XXX
```

Código 5.2: Configuración de nuevo un *host* SSH.

En “`Hostname`” se introducirá la IPv4 del servidor y “`Port`” será el puerto que elijamos (en este caso se ha rellenado con X tanto la IP como el puerto para que sea genérico) o el puerto 22 en caso de especificar ningún puerto. Tras esto, se reinicia el servicio de SSH y ya se puede acceder al *host* con el comando:

```
1      (sudo) ssh Odo014NOMBRE_CLIENTE
```

Código 5.3: Comando para la ejecución de un *host* SSH.

Se debe utilizar `sudo` si el puerto seleccionado se encuentra en 0 y 1023 ya que están reservados para el superusuario *root*.

HT02 – Configuración básica del servidor

- **Enunciado de la historia técnica:** como desarrollador, quiero realizar la configuración básica del servidor para poder instalar Odoo correctamente.
- **Descripción:** configurar los aspectos básicos del servidor e instalar los paquetes necesarios para la instalación de Odoo
- **Complejidad:** M
- **Tiempo empleado:** 7 horas.
- **Subtareas:**
 - HT.2.01: actualizar máquina.
 - HT.2.02: instalar paquetes básicos.
 - HT.2.03: configurar idioma y zona horaria.
 - HT.2.04: descargar *scripts* necesarios para el resto de la instalación.
- **Conversación:** se crea un pequeño *script* donde se ejecutan comandos para actualizar la máquina, instalar paquetes como `core`, `pwgen`, Subversion o Certbot. Además, se configura el idioma a español y la zona horaria correspondiente a Madrid. Por último, se descarga a través de Subversion un conjunto de *scripts* que serán útiles en el resto de la configuración e instalación de Odoo y PostgreSQL en la máquina.

HT03 – Instalar Odoo en el servidor

- **Enunciado de la historia técnica:** como desarrollador, quiero instalar Odoo en el servidor para proporcionar el servicio al cliente.
- **Descripción:** instalar instancia de Odoo 14 junto a la base de datos PostgreSQL 12.
- **Complejidad:** L
- **Tiempo empleado:** 20 horas.
- **Subtareas:**
 - HT.3.01: instalar y configurar PostgreSQL 12.
 - HT.3.02: instalar Python y crear un entorno virtual.
 - HT.3.03: instalar y configurar Odoo 14.
 - HT.3.04: instalar y configurar Apache 2.4.
- **Conversación:** se crea un *script* para realizar las diversas subtareas que tiene esta tarea. Lo primero que se hace en este *script* es comprobar que se está ejecutando como superusuario y que diversos paquetes necesarios para las instalaciones están instalados, como puede ser el caso de Subversion, `openssh-server`, `pwgen` o Python 3.

Posteriormente, se comienza con la instalación de PostgreSQL 12 y se crea un usuario en la base de datos para Odoo. Se elige el nombre de `odoo` para el usuario de la base de datos y la contraseña puede ser creada al azar con `pwgen`.

Una vez se ha instalado correctamente la base de datos, se procede con la creación de un entorno virtual de Python para soportar la instalación de Odoo, ya que Odoo está basado en Python. El hecho de utilizar un entorno virtual es útil para para aislar esa versión de Python junto a las librerías que son necesarias para el funcionamiento de Odoo. De este modo, en caso de suceder algún tipo de error, se puede crear otro entorno virtual y no tendría

ninguna influencia el entorno virtual anterior sobre este nuevo. También es útil para la instalación de varias instancias de Odoo, con diferentes versiones, en una misma máquina, aunque en este caso solo se va a instalar una instancia por máquina por motivos de rendimiento y seguridad. Para la instalación del entorno virtual se utilizan los siguientes comandos (véase Código 5.4:

```
1      sudo apt install python3-venv -yq
2      python3 -m venv /opt/odoo14/venv-odoo14
3      . /opt/odoo14/venv-odoo14/bin/activate
4      pip install wheel
5      pip install -r /opt/Scripts/data/
        requirements_odoo14_ubuntu2004.txt
```

Código 5.4: Comandos para creación de entorno virtual de Python.

En este caso el entorno virtual se crea en la carpeta “odoo14” dentro de la carpeta “opt”. Posteriormente, se activa el entorno virtual y se instala la librería `wheel` para poder instalar el resto de librerías necesarias. El archivo de texto llamado “`requirements_odoo14_ubuntu2004.txt`” contiene las librerías, junto a su versión correspondiente, necesarias para el correcto funcionamiento de Odoo.

Cuando ya se dispone del entorno virtual de Python, se descargan los archivos de Odoo 14 a través de un repositorio de Subversion. Después, se configuran el archivo “`odoo14.conf`” que contiene aspectos básicos de la configuración de Odoo como la contraseña de administrador (que en este caso se crea con `pwgen`), el puerto de la base de datos, el puerto por el que escucha las peticiones Odoo o las rutas en las que se encuentran los módulos descargados de Odoo y el archivo “`odoo14_server.service`” donde se configuran los parámetros de la ejecución del servicio que corre Odoo.

Por último, se descarga la versión 2.4 de Apache y se configura el acceso a través del puerto que se elija, en este caso el 8069. Además se configura

la opción tener múltiples hilos de Odoo trabajando al mismo tiempo para poder soportar múltiples peticiones. Para ello, se debe conocer el número de *workers* de la máquina, aplicando la fórmula: $2 * n^{\text{cores}} + 1$. En esta máquina se cuenta con 4 *cores*, así que el número de *workers* serán 9. Para gestionar estos *workers* se configura el *longpolling*, que escuchará peticiones por el puerto 8169 y redirigirá las peticiones.

Por lo tanto, el flujo será que Apache escuchará peticiones por el puerto 8069 y las mandará al puerto de *longpolling*, es decir, el 8169, el cual gestionará las peticiones entre los diferentes *workers*. Esta configuración se aplica en el archivo “/etc/apache2/sites-available/odoo_14.conf” y se puede ver su contenido en el Código 5.5.

```
1 <VirtualHost *:8069>
2     ServerAdmin admin@admin.com
3     <Proxy *>
4         Order deny,allow
5         Allow from all
6     </Proxy>
7     ProxyRequests Off
8     ProxyPreserveHost On
9     ProxyPass /longpolling http://localhost:8169/
10         longpolling
11     ProxyPassReverse /longpolling http://localhost
12         :8169/longpolling
13     ProxyPass / http://localhost:8059/
14     ProxyPassReverse / http://localhost:8059/
15 </VirtualHost>
```

Código 5.5: Configuración del VirtualHost de Odoo en Apache.

- **Enunciado de la historia técnica:** como desarrollador, quiero cifrar y certificar un dominio para el servidor para asegurar la integridad de los datos.
- **Descripción:** cifrar y certificar un nuevo dominio para el servidor utilizando Certbot y la configuración de Apache.
- **Complejidad:** L
- **Tiempo empleado:** 30 horas.
- **Subtareas:**
 - HT.4.01: obtener el dominio.
 - HT.4.02: añadir nuevo `VirtualHost` con el dominio obtenido.
 - HT.4.03: instalar y utilizar Certbot.
 - HT.4.04: evitar escuchas no seguras y configurar *longpolling* en el puerto 443.
- **Conversación:** en primer lugar se debe obtener un dominio. Para ello, existen muchas empresas que venden dominios. En este caso se obtendrá el dominio de la empresa GoDaddy [GoD22], ya que es la empresa que utiliza Second Episode para la gestión de sus dominios. Una vez obtenido el dominio, se debe añadir un nuevo `VirtualHost` con el dominio obtenido, como se puede ver en el Código 5.6.

```
1     <VirtualHost [nombredominio].com:80>
2         ServerName [nombredominio].com
3         ServerAdmin admin@admin.com
4     </VirtualHost>
```

Código 5.6: Configurar `VirtualHost` para el nuevo dominio.

El siguiente paso es certificar el dominio utilizando Certbot [Cer22]. Una vez instalado, utilizaremos el `ServerName` que hemos configurado en el nuevo

`VirtualHost`. Este `ServerName` se utilizará en el comando que se puede ver en el Código 5.7.

```
1 sudo certbot --apache -d [nombredominio].com
```

Código 5.7: Comando de Certbot para creación de un certificado.

Finalmente, se configurará el *longpolling* [Dev22] en el puerto 443 y se evitarán escuchas no seguras. Para ello, en el archivo “`etc/apache2/ports.conf`” se debe comentar el puerto que no sea ni el 80 ni el 443, en este caso el 8069. Además, en el `VirtualHost` que ha creado Certbot automáticamente se deben añadir los ajustes del *Proxy* [Apa22] configurados en el puerto que ahora está silenciado (véase Código 5.5).

HT05 – Configurar copias de seguridad

- **Enunciado de la historia técnica:** como desarrollador, quiero configurar copias de seguridad para preservar los datos del cliente.
- **Descripción:** crear copias de seguridad que se manden automáticamente a un servidor externo.
- **Complejidad:** L
- **Tiempo empleado:** 22 horas.
- **Subtareas:**
 - HT.5.01: crear *script* de copia de la base de datos.
 - HT.5.02: crear *script* que mande la copia de seguridad a otro servidor.
 - HT.5.03: crear un *cron* que llame al *script* de copia.
- **Conversación:** primero se debe crear un *script* que realice una copia de la base de datos en la propia máquina. Para ello, se toman y comprueban parámetros como la base de datos a respaldar, la carpeta de destino o la

instancia de Odoo. Después se realiza la copia de la base de datos utilizando el comando “`pg_dump`”.

Una vez se tiene el *script*, este será utilizado en un nuevo *script* donde se tomará la copia creada y se mandará al servidor externo donde queremos almacenar la copia. La transmisión de los datos se realizará mediante SFTP.

Por último, crearemos un `cron` (un proceso que se ejecuta de manera periódica) en el que se llamará a este último *script* para crear un copia de seguridad. Se le otorgará el periodo de tiempo que se considere oportuno, en este caso será cada 24 horas.

HT06 – Configurar medidas de seguridad

- **Enunciado de la historia técnica:** como desarrollador, quiero configurar medidas de seguridad en el servidor para evitar posibles intrusiones.
- **Descripción:** configurar varios aspectos de la máquina para evitar intrusiones a través de los puertos de escucha más comunes. Se configurará un bloqueo y una escucha en esos puertos.
- **Complejidad:** M
- **Tiempo empleado:** 12 horas.
- **Subtareas:**
 - HT.6.01: bloquear la salida por los puertos 25 y 143.
 - HT.6.02: configurar una escucha en los puertos 25 y 143.
- **Conversación:** los puertos 25 y 143 son los que se utilizan comúnmente para realizar ataques de *spam* a otras máquinas. Por ello, es una buena decisión bloquear la salida por estos puertos. Para prohibir la salida se utilizan los siguientes comandos, que se pueden ver en el Código 5.8:

```
1 sudo iptables -A OUTPUT -p tcp --dport 25 -j DROP
2 sudo iptables -A OUTPUT -p tcp --dport 143 -j DROP
```

Código 5.8: Bloque de puertos 25 y 143 por `iptables`.

Esta configuración deberá ser guardada en el archivo “`/etc/iptables/rules.v4`”.

Para realizar una escucha en estos dos puertos debemos instalar la librería `net-tools`. Posteriormente, debemos crear un *script* con el siguiente comando (véase Código 5.9):

```
1 sudo netstat -anotcp | grep '[0-9]:25 \|[0-9]:143'
```

Código 5.9: Escucha de puertos 25 y 143 con `netstat`.

Este comando se ejecutará en segundo plano constantemente y se escribirá el *log* en el siguiente archivo: “`/var/log/others/log_ports.txt`”.

6

Artefactos resultantes

En este capítulo se repasarán los principales aspectos de la máquina tras la configuración y posterior instalación de Odoon en ella. Cabe destacar que, a modo de ejemplo, se ha tomado como referencia el proceso de instalación de un cliente del cual se han anonimizado todos los datos referente a este.

6.1. Conexión remota al servidor

Tras la configuración de acceso por SSH al servidor, se puede acceder de manera rápida y sin tener que introducir contraseñas en cada intento de acceso. En la Figura 6.1 se puede observar una captura de la consola de comandos en

```
apolo@apolo-pc:~$ ssh Odoo14
The authenticity of host 'Odoo14' can't be established.
ECDSA key fingerprint is [REDACTED].
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'Odoo14' (ECDSA) to the list of known hosts.
bind [127.0.0.1]:99: Permission denied
channel_setup_fwd_listener_tcpip: cannot listen to port: 99
Could not request local forwarding.
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-125-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of [REDACTED]

System load:  0.0          Processes:    211
Usage of /:   3.2% of 154.88GB   Users logged in:  0
Memory usage: 28%          IPv4 address for ens3: [REDACTED]
Swap usage:  0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

8 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***

root@vps-a5b1559e:~#
```

Figura 6.1: Conexión por SSH al servidor.

una acceso por SSH a la máquina remota.

6.2. Odoo

En lo referente a Odoo se deben revisar los archivos de configuración más importantes: el archivo “odoo_14.conf” (véase Código 6.1), referente a la configuración básica de Odoo y el archivo “odoo14_server.service” (véase Código 6.2), donde se configuran los parámetros referentes a la ejecución del servicio.

```
1 [options]
2 ; This is the password that allows database operations:
3
4 admin_passwd = asflkfgs34Q
5
6 db_host = False
7 db_port = 5433
8 db_user = odoo
```

```
9 db_password = pjweAF76DA
10 dbfilter = repuestosonline
11
12 addons_path = /opt/odoo14/repuestosonline/core/addons,/opt/
    odoo14/repuestosonline/extra_addons_patched, /opt/odoo14
    /repuestosonline/extra_addons, /opt/odoo14>
13
14 logfile = /var/log/odoo14/odoo_14_repuestosonline_server.
    log
15 logrotate = True
16 data_dir = /opt/odoo14_data_dir/repuestosonline
17
18 xmlrpc_port = 8059
19 longpolling_port = 8169
20 proxy_mode = True
21 workers = 9
```

Código 6.1: Archivo de configuración básica de Odoo.

```
1 [Unit]
2 Description=Odoo Open Source ERP and CRM
3 Requires=postgresql.service
4 After=network.target postgresql.service
5
6 [Service]
7 Type=simple
8 PermissionsStartOnly=true
9 SyslogIdentifier=odoo-repuestosonline-server
10 StandardOutput=syslog
11 StandardError=syslog
12 User=odoo
13 Group=odoo
```

```

14     ExecStart=/opt/odoo14/venv-odoo14-repuestosonline/bin/
        python /opt/odoo14/repuestosonline/core/odoo-bin --
        config=/etc/odoo_14_repuestosonline.conf
15     WorkingDirectory=/opt/odoo14/repuestosonline/core/
16
17     [Install]
18     WantedBy=multi-user.target

```

Código 6.2: Archivo de configuración del servicio de Odoo.

6.3. Configuración del servidor *web*

En cuanto a la configuración de Apache se repasa el resultado del *VirtualHost* (véase Código 6.3) que Certbot ha creado automáticamente en el proceso de certificación del dominio en el archivo “odoo_14_\$NOMBRECLIENTE-le-ssl.conf”. En este *VirtualHost* se indica donde se almacenan los certificados SSL para el dominio en cuestión. También se observa la configuración del *Proxy*

```

1     <IfModule mod_ssl.c>
2     <VirtualHost repuestosonline.com:443>
3         ServerAdmin admin@admin.com
4         ServerName repuestosonline.com
5         <Proxy *>
6             Order deny,allow
7             Allow from all
8         </Proxy>
9         ProxyRequests Off
10        ProxyPreserveHost On
11        ProxyPass /longpolling http://localhost:8169/
        longpolling
12        ProxyPassReverse /longpolling http://localhost

```

```

:8169/longpolling
13     ProxyPass / http://localhost:8059/
14     ProxyPassReverse / http://localhost:8059/
15
16     SSLCertificateFile /etc/letsencrypt/live/repuestosonline.
        com/fullchain.pem
17     SSLCertificateKeyFile /etc/letsencrypt/live/repuestosonline
        .com/privkey.pem
18     Include /etc/letsencrypt/options-ssl-apache.conf
19 </VirtualHost>
20 </IfModule>

```

Código 6.3: *VirtualHost* creado por Certbot.

6.4. Copias de seguridad

En esta última sección se repasa el contenido del *script* (véase Código 6.4) que guarda una copia de la base de datos y que posteriormente se manda, a través de SFTP, a un servidor externo para almacenarla. El funcionamiento de este *script* consiste primeramente en comprobar los parámetros pasados por línea de comandos a la función y obtener otros parámetros como la ruta donde se encuentra el servicio de Odoo, el archivo de configuración de este o la ruta en la que se encuentra la base de datos. Seguidamente, se obtiene el usuario y contraseña de la base de datos y se realiza la copia de seguridad con los parámetros que se disponen con el comando `pg_dump`. Si la copia se ha realizado con éxito, se copian los archivos a una carpeta externa otorgándole un nombre en función de la fecha y hora en la que se ha realizado la copia de seguridad.

```

1     #!/bin/sh
2
3     #####

```

```

4     ## Backup an existing Odoo database
5     ## Parameters:
6     ### \$1: database to backup
7     ### \$2: destination folder
8     ### \$3: Odoo instance
9     ### \$4: filename. optional
10    ##
11    ## Remember to have /root/.pgpass file properly configured
12    #####
13
14    if [ \$# -eq 3 ] || [ \$# -eq 4 ]; then
15        # Read command line parameters
16        db=\$1
17        folder=\$2
18        odooinstance=\$3
19
20        # Get the path where the Odoo code is located and its
21        # file configuration
22        if [ -f "/etc/systemd/system/\${odooinstance}.service" ]
23        then
24            odooopath=`cat /etc/systemd/system/\${odooinstance}.
25                service | awk '/WorkingDirectory=/' | cut -d '=' -f2`
26
27            fileconfig=`cat /etc/systemd/system/\${odooinstance}.
28                service | awk '/ExecStart=/' | cut -d '=' -f3`
29            odooouser=`cat /etc/systemd/system/\${odooinstance}.
30                service | awk '/User=/' | cut -d '=' -f2`
31        else
32            odooopath=`cat /etc/init.d/\${odooinstance} | awk '/^
33                DAEMON=/' | cut -d '=' -f2 | rev | cut -d '/' -f2-
34                | rev`

```



```

28     fileconfig=`cat /etc/init.d/\${odooinstance} | awk '/^
        CONFIGFILE=/' | cut -d '"' -f2`
29     if [ -z "\$fileconfig" ]
30     then
31         fileconfig=`cat /etc/init.d/\${odooinstance} | awk '
            /^CONFIG=/' | cut -d '=' -f2`
32     fi
33     odoouser=`cat /etc/init.d/\${odooinstance} | awk '/^
        USER=/' | cut -d '=' -f2`
34     fi
35
36     # Get the file where the local files are located
37     datadir=`cat \${fileconfig} | awk '/data_dir =/' | cut -d
        '=' -f2`
38     if [ -z "\$datadir" ]
39     then
40         datadir=\${odoopath}/.local/\$db
41     fi
42
43     # Get the PostgreSQL administrator credentials and the
        port where PostgreSQL is running
44     odoopguser=`cat \${fileconfig} | awk '/^db_user =/' | cut
        -d '=' -f2 | tr -d ' '`
45     odoopgpass=`cat \${fileconfig} | awk '/^db_password =/' |
        cut -d '=' -f2 | tr -d ' '`
46     odoopgport=`cat \${fileconfig} | awk '/^db_port =/' | cut
        -d '=' -f2 | tr -d ' '`
47
48     # Create a security copy for received database
49     export PGPASSWORD="\${odoopgpass}"
50     if [ -z "\$odoopgport" ]

```

```

51     then
52         pg_dump -Ft -h "localhost" -U "\${odooPguser}" \${db} -w
           -f database.tar
53     else
54         pg_dump -Ft -h "localhost" -U "\${odooPguser}" \${db} -w
           -f database.tar -p \${odooPgport}
55     fi
56
57     # Check errors after dumping process
58     dumperror=\$?
59     if [ \${dumperror} != 0 ]; then
60         echo "Database \${db} could not be backed up"
61         exit 1
62     fi
63
64     # Copy the folder where the local files are located
65     mkdir -p filestore
66     cp -r \${datadir}/filestore/\${db}/* filestore
67
68     # Create a file to bind the security copy with a SVN
           version
69     repository=`svn info \${odooPath} | grep '^URL' | awk '{
           print \${NF}}'`
70     echo "Odoo code: \${repository}" > svn-odoo.txt
71
72     # Get the filename
73     if [ \${#} -eq 4 ]; then
74         name=\$4
75         filename="\${folder}/\${name}"
76     else
77         date=`date +"%Y-%m-%d_%H-%M-%S"`

```

```
78     filename="\${folder}/\${db}sc_\${date}"
79     fi
80
81     # Generate the whole security copy and remove temporal
82     files
83     if ! type "zip" > /dev/null; then
84         tar -czf \${filename}.tar.gz database.tar svn-odoo.txt
85         filestore
86     else
87         zip -qr \${filename} database.tar svn-odoo.txt
88         filestore
89     fi
90
91     # Remove temporal files
92
93     rm -r filestore
94     rm database.tar
95     rm svn-odoo.txt
96
97     exit 0
98
99 else
100     echo "usage: \$0 dbtobackup destinationfolder
101         odooinstance filename"
102     exit 1
103 fi
```

Código 6.4: *Script* de copia de la base de datos de Odoo.

7

Caso de ejemplo

En este capítulo se ilustra la aplicación del protocolo descrito en el Capítulo 3 a una compañía de venta de repuestos de coche a través de Internet. Para preservar la protección de datos, se referirá a esta empresa a lo largo del capítulo como RepuestosOnline S.L.

El contacto inicial se produjo a través de un formulario de contacto situado en la web de Second Episode. En este formulario se le pide al *lead* algunos datos de interés como el nombre de la empresa, datos de contacto, rango de facturación anual de la empresa, número de usuarios que van a utilizar el *software* o módulos que están interesados en utilizar en Odoo, entre otros. En este caso, la empresa RepuestosOnline es una empresa que acaba de empezar y que cuenta con una

gran financiación para su proyecto de venta *online*. Su intención es que el *software* sea gestionado por 4 usuarios y les interesan las áreas de venta, facturación y contabilidad, principalmente. Estos datos le sirven al responsable encargado a este *lead* para realizar una aproximación a las necesidades que puede tener el cliente. Posteriormente, se procede a realizar una llamada al *lead*. En la llamada, se ahonda en la actividad de la empresa. RepuestosOnline está comenzando su actividad en ese instante y tienen en desarrollo un *e-Commerce* a través de la plataforma WooCommerce. Comentan que para ellos la integración de Odoo con WooCommerce es prioritaria. Los pedidos de venta que se importen quieren gestionarlos dentro de Odoo y sincronizar la logística con la empresa de envíos que tienen contratada. Les interesa aunar todos los procesos posibles dentro de Odoo para ahorrar tiempo y duplicidad de datos y por ello quieren manejar la contabilidad dentro de este mismo *software*. Además, en un futuro también tienen planteado vender a través de Amazon, lo que supondría una nueva integración. La llamada termina con una propuesta económica que se manda totalmente detallada por correo electrónico.

RepuestosOnline decide aceptar la propuesta y se convierte el cliente. Una vez esto ocurre, se puede avanzar a las siguientes fases del protocolo, que consisten en extraer los requisitos generales y específicos. Partiendo de la información recogida en la llamada, se cuenta con una base para comenzar con el análisis de las áreas temáticas. A breve resumen, ya que esta fase consistió en varias reuniones y una extensa cantidad de datos extraídos, se repasan las principales áreas y las funcionalidades más destacables de cada una. Dentro del apartado de venta, se necesita una integración con su tienda *online* desarrollada en WooCommerce. Esta integración requerirá la importación de pedidos de manera automática y la sincronización de *stock* entre ambas plataformas, ya que el *stock* también se tendrá que sincronizar en un futuro con otras plataformas en las que vendan sus productos como Amazon. Además, necesitan añadir nuevos campos de referencia interna secundaria, descripción del producto y fotos de exposición. En el apartado

de contabilidad, quieren llevar un control de los diarios separados en función del método del pago que haya utilizado el cliente para comprar los productos. Su venta estará limitada solo a territorio español, por lo tanto no hace falta configurar otra legislación aparte de la española.

Una vez se ha realizado la extracción de requisitos, se instala Odoo, tal y como se puede apreciar en los Capítulos 5 y 6. Una vez se dispone de la instancia de Odoo, se procede con la parametrización de las distintas áreas instaladas. Tras varias iteraciones se configuran correctamente los permisos de los 4 usuarios, el servidor de correo saliente y las diferentes características de contabilidad, como varios diarios en función del método de pago. En estas iteraciones de parametrización, se aprovecha para realizar pequeñas formaciones a través de videollamadas o tutoriales a través de PDF con texto e imágenes explicativas, con las que los usuarios pueden entender mejor los flujos de Odoo y adaptar mejor la parametrización. Además, según aumenta el conocimiento de los usuarios en Odoo, surgen nuevos requisitos específicos que pueden ser resueltos en las siguientes iteraciones, como por ejemplo modificar la posición de ciertos campos para agruparlos con un mayor sentido para el flujo que acostumbran los usuarios.

Tras la correcta parametrización de las distintas áreas, se procede con la integración entre Odoo y WooCommerce solicitada. Existe un módulo de pago desarrollado por una empresa de la comunidad de Odoo que aporta una buena solución base. Se comunica la existencia del módulo al cliente y confirma la compra de este. En este punto, la fase de integración y desarrollo a medida convergen ya que el módulo comprado debe ser modificado por dos motivos. Uno de ellos se debe a que el módulo comprado contaba con un fallo en el código que no permitía la correcta importación de pedidos desde WooCommerce. El segundo motivo es que al cliente algunas funcionalidades del módulo no se adaptaban a sus necesidades y es necesario modificarlo, como por ejemplo el identificador con el que se creaba el pedido de venta dentro de Odoo. Además, también se realizan otros pequeños desarrollos a medida como un campo dentro de los pedidos de venta

que indique el importe del pago que tiene que realizar el comprador del pedido en caso de que la forma de pago sea a contra reembolso. En caso de que la forma de pago no sea a contra reembolso, el valor del campo será 0.

En el momento de redacción de este TFG, este caso de ejemplo sigue en fases de desarrollo, comenzando la integración con Amazon para la sincronización de los pedidos de venta que se realicen en esta plataforma. Por lo tanto, todavía no se ha pasado plenamente a la fase de mantenimiento. Sin embargo, sí que se han realizado labores de mantenimiento como la actualización de módulos, tanto del *core* de Odoo como de terceros, cambios en los servidores de correo saliente configurados en Odoo o la optimización de códigos poco eficientes, entre otros.

8

Conclusiones y trabajos futuros

En este último capítulo se presentan las conclusiones obtenidas a raíz de la realización de este trabajo. Además, se tratan los posibles trabajos que se pueden desarrollar en un futuro.

8.1. Conclusiones

Este TFG se ha centrado en la disposición de un protocolo para la gestión de un nuevo cliente y la posterior implantación de un ERP junto a las correspondientes tecnologías y herramientas necesarias en función de las necesidades extraídas previamente. Para realizar este proceso, ha sido necesaria la elaboración del pro-

toloco, organizándose este en una serie de etapas. A su vez, se ha hecho especial hincapié en la fase de instalación, creando *scripts* y usando procedimientos como la conexión remota a la máquina donde se va a instalar el *software* para poder configurar los diversos ajustes. Dado al gran número de tareas que se debían realizar en la parte técnica, ha sido importante el uso de una metodología (véase Sección 4), adaptada a partir de Scrum, para poder organizar el trabajo a través de iteraciones.

En cuanto a los objetivos que se estipularon para este TFG (véanse Secciones 2.1 y 2.2), se puede determinar que se han cumplido con éxito. Se ha conseguido proponer un protocolo de implantación para el ERP Odoon adaptado a las necesidades de cada cliente, asegurando además la seguridad y el rendimiento del sistema. Para llegar a este objetivo general se han ido satisfaciendo los objetivos específicos, que funcionaron como pequeños escalones hasta llegar al resultado final.

La realización de este trabajo supone un gran beneficio para la empresa Second Episode, ya que le permite proporcionar a sus nuevos clientes el servicio adaptado a sus necesidades y en el menor tiempo posible. Esto se debe a que el uso de los *scripts* creados reduce de manera significativa el tiempo necesario para la implantación del sistema en nuevo servidor. Un proceso que, de no estar automatizado mediante *scripts*, podría llevar varios días realizar y ocasionar posibles errores debido a fallos humanos.

Personalmente, este TFG ha supuesto al autor adentrarse en dos áreas casi desconocidas durante el transcurso de la carrera como son la gestión de clientes y el manejo de sistemas. Conocer los flujos de negocio, utilizar tecnologías como SSH y SFTP, practicar de un modo más avanzado el manejo de ficheros y desarrollar *scripts* avanzados ha supuesto una experiencia vital para el futuro laboral. Estos apartados, al igual que tantos otros, son clave para el desempeño de las funciones esperadas en un ingeniero informático.

Para terminar, se debe resaltar que, aunque el desarrollo de este TFG supone un avance en la implantación del sistema Odoo, no es más que un primer paso en la optimización de este. Se pueden desarrollar nuevas funcionalidades o seguir mejores pautas que hagan el proceso más rápido y mejor adaptado a las necesidades del cliente, como se puede observar en los trabajos futuros que se presentan en la siguiente sección.

8.2. Trabajos futuros

Este proyecto cuenta con una gran cantidad de posibilidades para ser mejorado y conseguir una mayor automatización en el proceso de implantación de un *software* en un servidor. Se enuncian, a continuación, algunos posibles trabajos futuros:

- Realizar un análisis del protocolo propuesto. Realizar un seguimiento de la implantación que se realice a varios clientes siguiendo el protocolo propuesto en TFG. Tras ello, analizar las dificultades o contratiempos que se hayan podido dar y proponer optimizaciones de las fases del protocolo.
- Implantar Odoo con Docker. Esto permitiría automatizar de manera más intuitiva el proceso tanto de implantación como de despliegue de las futuras versiones que se desarrollen para un cliente. Existen ya algunos trabajos en este campo, como por ejemplo la iniciativa llamada Doodba [Tec22] por parte de la Asociación Española de Odoo, con la que se puede tener un punto de partida para adaptar al proyecto según se necesite.
- Simplificar los *scripts* creados y unificarlos en la medida de lo posible para que la implantación se produzca de la manera más rápida posible.
- Actualizar el archivo “requirements.txt” (que recoge todas las librerías de Python necesarias para el correcto funcionamiento) de manera automática.

Durante el transcurso de este proyecto, uno de los archivos que más problemas ha supuesto es este. A medida que se actualizan diversos módulos de la comunidad o del propio *core* de Odoo, es posible que sea necesario actualizar la versión de algunas de las librerías existentes o que se necesite alguna librería nueva. La detección automática de estos cambios en las librerías supondría evitar una gran cantidad de fallos en la implantación y actualización del servicio.

Bibliografía

- [Ace22] Acelerapyme. Kit digital, Últ acceso: 4/09/2022. URL <https://www.acelerapyme.gob.es/kit-digital>.
- [Apa22] Apache. Guía de proxy inverso, Últ acceso: 8/12/2022. URL https://httpd.apache.org/docs/trunk/es/howto/reverse_proxy.html.
- [Cer22] Certbot. Certbot, Últ acceso: 20/10/2022. URL <https://certbot.eff.org/instructions?ws=apache&os=ubuntufocal>.
- [Com22] ComparaSoftware. Planning poker en scrum, Últ acceso: 12/11/2022. URL <https://blog.comparasoftware.com/que-es-planning-poker-en-scrum/>.
- [cre22] Report for erp systems — winter 2021, Últ acceso: 28/06/2022. URL https://www.g2.com/reports/momentum-grid-report-for-erp-systems-winter-2021?featured=acumatica&secure%5Bgated_consumer%5D=98446059-5048-45ab-863b-f15802998911&secure%5Btoken%5D=f68a0a7d968d06b91967f5ceba1a2923ec15f6198a675708dd93e50e351ad4a8&utm_campaign=gate-791531.
- [Dev22] Odo Development. About longpolling, Últ acceso: 8/12/2022. URL https://odoo-development.readthedocs.io/en/latest/admin/about_longpolling.html.
- [Gar22] Javier Garzás. La deuda técnica., Últ acceso: 20/07/2022. URL <https://www.javiergarzas.com/2012/11/deuda-tecnica-2.html>.
- [GoD22] GoDaddy. Godaddy, Últ acceso: 20/10/2022. URL <https://www.godaddy.com/es-es>.
- [hol22] Holded, Últ acceso: 28/06/2022. URL <https://holded.com/>.

- [Net22] NetCraft. June 2022 web server survey, Últ acceso: 16/10/2022. URL <https://news.netcraft.com/archives/2022/06/30/june-2022-web-server-survey.html>.
- [Odo22] Odoo. Odoo ahora es un unicornio. ¿y qué?, Últ acceso: 28/06/2022. URL https://www.odoo.com/es_ES/blog/nuestro-blog-5/odoo-ahora-es-un-unicornio-y-que-784.
- [Odo23] Odoo. Sobre nosotros, Últ acceso: 21/06/2023. URL https://www.odoo.com/es_ES/page/about-us.
- [ora22] Oracle erp, Últ acceso: 28/06/2022. URL <https://www.oracle.com/es/erp/>.
- [sap22] Sap erp, Últ acceso: 28/06/2022. URL <https://www.sap.com/spain/index.html>.
- [Sch22] Ken Schwaber. The scrum development process, Últ acceso: 24/06/2022. URL <https://www.scrum.org/resources/scrum-development-process>.
- [Sta22] Statista. Cuota de mercado de los principales proveedores de software de aplicaciones empresariales (eas) a nivel mundial, Últ acceso: 13/10/2022. URL <https://es.statista.com/estadisticas/909980/software-empresariales-cuota-de-mercado-mundial-de-proveedores/>.
- [Tec22] Tecnativa. Doodba, Últ acceso: 6/12/2022. URL <https://github.com/Tecnativa/doodba>.
- [Uni22a] Unión Europea. Estrategia de desarrollo sostenible 2030, Últ acceso: 30/06/2022. URL <https://www.mdsocialesa2030.gob.es/agenda2030/index.htm>.
- [Uni22b] Unión Europea. Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación, Últ acceso: 30/06/2022. URL <https://www.un.org/sustainabledevelopment/es/infrastructure/>.

