



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE INGENIERÍA DE SISTEMAS

**ASIGNACIÓN DE LAS DONACIONES ENTREGADAS A LA
FUNDACIÓN LOS JÓVENES DEL FUTURO, UBICADA EN QUITO-
ECUADOR MEDIANTE UN PROTOTIPO WEB SOBRE OPEN SOURCE
ODOO COMMUNITY VERSIÓN 14.0**

Trabajo de titulación previo a la obtención del

Título de Ingeniero de Sistemas

AUTOR: SEBASTIÁN GEOVANNY PEÑA ARÉVALO

TUTOR: GUSTAVO ERNESTO NAVAS RUILOVA

Quito – Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO
DE TITULACIÓN**

Yo, Sebastián Geovanny Peña Arévalo, con documento de identificación N° 1716476303,
manifiesto que:

Soy el autor y responsable del presente trabajo; y autorizo a que sin fines de lucro la
Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera
total o parcial el presente trabajo de titulación.

Quito, 1 de marzo del 2024

Atentamente,

A handwritten signature in blue ink, appearing to read 'Sebastián Peña Arévalo', written in a cursive style.

Sebastián Geovanny Peña Arévalo
1715476303

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Sebastián Geovanny Peña Arévalo, con documento de identificación N° 1716476303, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud que soy el autor del Proyecto Técnico: “Asignación de las donaciones entregadas a la fundación los jóvenes del futuro, ubicada en Quito-Ecuador mediante un prototipo web sobre Open Source Odoo Community version 14.0”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 1 de marzo del 2024

Atentamente,

A handwritten signature in blue ink, appearing to read 'Sebastián Geovanny Peña Arévalo'.

Sebastián Geovanny Peña Arévalo

1715476303

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Gustavo Ernesto Navas Ruilova con documento de identificación N° 1705675625, docente de la Universidad Politécnica Salesiana, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación: ASIGNACIÓN DE LAS DONACIONES ENTREGADAS A LA FUNDACIÓN LOS JÓVENES DEL FUTURO, UBICADA EN QUITO-ECUADOR MEDIANTE UN PROTOTIPO WEB SOBRE OPEN SOURCE ODOO COMMUNITY VERSIÓN 14.0, realizado por Sebastián Geovanny Peña Arévalo con documento de identificación N° 1715476303, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 1 de marzo de 2024

Atentamente,



ING. Gustavo Ernesto Navas Ruilova, MSc.

1705675625

Dedicatoria

A mi amada familia: padre, madre y hermanos por su apoyo incondicional.

Que tus sueños y anhelos sean el motor e inspiración para lograr lo que tanto buscas.

ÍNDICE GENERAL

INTRODUCCIÓN	1
Antecedentes	1
Problema	3
Justificación	4
Objetivos	5
Objetivo general	5
Objetivos específicos	5
Metodología	5
CAPITULO I	8
MARCO TEÓRICO	8
2.1. Fundación Jóvenes del Futuro	8
2.2. C.R.M. (Customer Relationship Management)	8
2.3. Odoo	9
2.4. Odoo Enterprise	9
2.5. Odoo Community	10
2.6. Arquitectura modular Odoo	10
2.6.1. CRM	11
2.6.2. Ventas	11
2.6.3. Inventario	11
2.6.4. Contabilidad.	11
2.6.5. Gestión proyectos	12
2.6.6. Mailing	12
2.7. Personalización Odoo	12
CAPITULO II	16
DISEÑO Y CONSTRUCCION DEL PROTOTIPO WEB	16
3.1. Planeación	16
3.1.1. Dimensionamiento	17
3.2. Sprints	18
3.2.1. Sprint 1	18
3.2.2. Ejecución del sprint 1	19
3.2.3. Sprint 2	22
3.2.4. Ejecución del sprint 2	23
3.2.5. Sprint 3	24
3.2.6. Ejecución del sprint 3	25
3.2.7 Sprint 4	28
3.2.8 Ejecución del sprint 4	29
3.2.9 Sprint 5	32
3.2.10 Ejecución del sprint 5	33

3.2.11 Fase de cierre.....	34
CAPITULO III.....	37
PRUEBAS Y RESULTADOS	37
CONCLUSIONES	43
RECOMENDACIONES.....	45
LISTA DE REFERENCIAS	46

ÍNDICE DE TABLAS

Tabla 1	6
Tabla 2	17
Tabla 3	18
Tabla 4	19
Tabla 5	21
Tabla 6	22
Tabla 7	24
Tabla 8	24
Tabla 9	28
Tabla 10	28
Tabla 11	32
Tabla 12	32
Tabla 13	34
Tabla 14	39

ÍNDICE DE FIGURAS

Figura 1	11
Figura 2	12
Figura 3	14
Figura 4	15
Figura 5	16
Figura 6	21
Figura 7	23
Figura 8	26
Figura 9	27
Figura 10	27
Figura 11	29
Figura 12	30
Figura 13	30
Figura 14	31
Figura 15	31
Figura 16	31
Figura 17	33
Figura 18	34
Figura 19	35
Figura 20	35
Figura 21	36
Figura 22	36
Figura 23	37
Figura 24	38
Figura 25	38
Figura 26	39
Figura 27	41
Figura 28	41
Figura 29	42

RESUMEN

En presente documento se ha detallado el proceso de desarrollo, diseño y adaptación, de un prototipo web diseñado específicamente para la Fundación los Jóvenes del Futuro, esta investigación se crea ante la necesidad de optimizar el uso de la información relacionada con el registro de patrocinadores, donaciones y beneficiarios, proporcionando así un sistema accesible y disponible en cualquier momento.

Este prototipo se ha desarrollado siguiendo los principios de la metodología SCRUM, metodología ágil que ha demostrado ser altamente eficaz en la generación rápida de soluciones flexibles. La elección de SCRUM ha permitido una mayor adaptabilidad a los cambios y una entrega iterativa, asegurando así la alineación continua con los objetivos y requisitos de la Fundación.

El proceso de desarrollo se ha enriquecido significativamente mediante la aplicación de técnicas de investigación cualitativa, específicamente entrevistas y análisis documental. Lo que ha posibilitado la comprensión de las necesidades y expectativas del product owner de la Fundación. Las entrevistas han proporcionado valiosos insights, permitiendo la identificación precisa de requisitos y la definición clara de funcionalidades esenciales.

Además, el análisis documental sirve fundamentalmente para la recolección de información relevante, como la identificación de patrones, tendencias y mejores prácticas, enriqueciendo así la fase de diseño y desarrollo del prototipo.

El resultado final de este proyecto es un prototipo funcional, diseñado para satisfacer las necesidades específicas de la Fundación, que garantice tener una experiencia de usuario fluida, mientras que su capacidad para realizar eficientemente el registro de patrocinadores, donaciones y beneficiarios permita dar acceso permanente y remoto a la información de la fundación.

De esta manera el prototipo web no solo representa una solución tecnológica, sino una herramienta que contribuye con un paso significativo a la mejora continua de las operaciones y con ello a la misión de la Fundación en pro del bienestar de la sociedad.

Palabra Claves: Odoo Community, comunidad, Fundación, prototipo, Open Source, SCRUM, personalización.

ABSTRACT

In this document, the development, design, and adaptation process of a web prototype specifically designed for the Future Youth Foundation has been detailed. This research is initiated to optimize the use of information related to the registration of sponsors, donations, and beneficiaries, thus providing an accessible and available system at any time.

This prototype has been developed following the principles of the SCRUM methodology, an agile methodology that has proven to be highly effective in the rapid generation of flexible solutions. The choice of SCRUM has allowed for greater adaptability to changes and iterative delivery, ensuring continuous alignment with the goals and requirements of the Foundation.

The development process has been significantly enriched through the application of qualitative research techniques, specifically interviews and documentary analysis. This has facilitated the understanding of the needs and expectations of the Foundation's product owner. The interviews have provided valuable insights, allowing the precise identification of requirements and the clear definition of essential functionalities.

Furthermore, documentary analysis serves primarily for the collection of relevant information, such as the identification of patterns, trends, and best practices, thus enriching the design and development phase of the prototype.

The final result of this project is a functional prototype designed to meet the specific needs of the Foundation, ensuring a seamless user experience, while its ability to efficiently handle the registration of sponsors, donations, and beneficiaries allows for permanent and remote access to foundation information.

In this way, the web prototype not only represents a technological solution but also a tool that contributes significantly to the continuous improvement of operations and, thereby, to the Foundation's mission for the well-being of society.

Keywords: Odoo Community, community, Foundation, prototype, Open Source, SCRUM, customization.

INTRODUCCIÓN

Antecedentes

En el Ecuador al 2020 existían 4939 organizaciones no gubernamentales que se caracterizan por una gestión sin fines de lucro para la atención a personas que provienen de hogares en situación de pobreza o vulnerabilidad, y que muchas veces ven a estas organizaciones como la única forma de acceso a servicios que el estado o la empresa privada no puede entregar (Gortaire et al., 2022, pp. 3-10).

Una de estas instituciones sin fines de lucro es la Fundación Jóvenes del futuro (en adelante la Fundación), la cual desarrolla sus actividades en pro del desarrollo de niños, niñas, adolescentes, jóvenes, adultos, adultos mayores, personas con discapacidades y sus familias. (Fundación jóvenes del Futuro, 2022)

La Fundación, realiza solicitudes o peticiones a instituciones o personas para que realicen donaciones, estas donaciones son recibidas por la fundación y posteriormente clasificadas y destinadas según su grado de prioridad y disponibilidad a los distintos beneficiarios inscritos en la fundación, al ser este un proceso manual, se desconoce a ciencia cierta el impacto de su intervención en cada uno de los beneficiarios. Esto es evidenciado por Gortaire et al, que menciona que “La falta de sistematización de la información sobre las ONG en Ecuador es un problema invisibilizado debido a la falta de voluntad política e interés general de la sociedad civil” (Gortaire et al., 2022, pp. 205-228).

Esta falta de información de los beneficios que otorga la ONG afecta directamente a los beneficiarios o posibles beneficiarios, al desconocer de las posibilidades de superación en los campos que actúa la organización.

En el caso de la Fundación, el escaso manejo de la información sobre sus procesos internos puede acarrear la inconformidad con los servicios prestados por parte de los

receptores de las donaciones, y, por otro lado, los donantes al no tener datos del impacto de su intervención podrían buscar otras formas de apoyar a la comunidad o en el peor de los casos, dejar de hacerlo.

El manejo eficiente de la información en una organización se erige como uno de los recursos más valiosos para cualquier entidad, ya que su control habilita la toma de decisiones orientada a maximizar la eficiencia de los servicios y la obtención de beneficios (Torres & Lamenta, 2015). Aunque el trabajo manual que desarrolla la fundación para el registro de patrocinadores, donaciones y beneficiarios, es una forma de gestión de la información, esta no se adecua a la velocidad y necesidad de disponibilidad que exige la sociedad actual.

Con el fin de mejorar la gestión de la información se vuelve necesario observar las opciones existentes en aplicaciones informáticas que tengan relación con las necesidades de la fundación, entre ellas, administrar las donaciones a los grupos vulnerables y que le permitan visibilizar los alcances de la intervención.

Respecto a esta necesidad, en el mercado existen gran variedad de soluciones que pueden ser utilizadas para la gestión de la información, pero en este caso, se utilizará como marco de referencia, herramientas de código abierto, y una herramienta que disponga de módulos de inventario y C.R.M. (*del inglés Costamar Relationship Management*) para que cubra las necesidades requeridas por la fundación con el fin de reducir costos de implementación y puesta en marcha.

Actualmente, existe una opción de E.R.P. (*del inglés Enterprise Resource Planning*) que cumple con las características de los requerimientos, denominado Odo, la misma es de código abierto (Open Source Initiative, 2023) y dispone de una comunidad que desarrolla módulos de libre uso y de fácil integración. Esta opción puede ser utilizada

como base para la adaptación de un sistema prototipo de registro de donaciones que facilite la gestión de la Fundación.

Problema

La gestión de la información es uno de los bienes más preciados para cualquier entidad (Torres & Lamenta, 2015), y para la fundación Jóvenes del Futuro que debe gestionar sus registros de donaciones, patrocinadores y beneficiarios de manera manual en medios físicos, se ha convertido en un problema, no solo al momento de presentar informes a sus colaboradores si no al momento de distribuir equitativamente los bienes y servicios que se encuentren disponibles.

Al carecer de un sistema que automatice la gestión, este proceso puede tomar más tiempo, pues hay que interrelacionar de manera manual la disponibilidad de materiales y tiempo de los voluntarios y coordinadores para realizar la entrega de donaciones. Esta labor actualmente se desarrolla mediante fichas físicas de registro, tanto de los beneficiarios, colaboradores y de los insumos gestionados por la fundación.

Esto desemboca en que los tiempos de respuesta en la atención de los beneficiarios sea lenta, deficiencias en la organización de los espacios de atención y asignación de turnos, o insumos. Algunas veces se dan los casos de que un beneficiario obtenga por duplicado los insumos, o en otros casos que no alcance a recibirlo.

Estas dificultades se podrían solventar si la fundación contase con un sistema de información que facilite la organización del inventario de insumos y atenciones disponibles; así, como la gestión de usuarios y benefactores.

Justificación

La Fundación los Jóvenes del Futuro entidad sin fines de lucro, dedicada a mejorar la calidad de vida de familias de escasos recursos. Enfrenta el desafío de no disponer de herramientas informáticas que faciliten la distribución de los beneficios otorgados por la fundación hacia su población objetivo, lo que desemboca en problemas de asignación y seguimiento de sus procesos.

Considerando esta necesidad y teniendo en cuenta que la Universidad Politécnica Salesiana como parte de su misión busca formar estudiantes “con excelencia humana y académica, con capacidad investigativa e innovadora, que contribuyan al desarrollo sostenible local y nacional” (Universidad Politécnica Salesiana, s/f) se plantea el presente proyecto de titulación para brindar una solución informática que permita el registro de las distintas fichas de los actores y recursos involucrados en el proceso de patrocinio a los beneficiarios de la ONG los Jóvenes del Futuro, como aporte a la continuidad de la labor social desarrollada por la Fundación.

Por lo tanto, se plantea diseñar una solución informática que permita a la Fundación mantener disponible y organizada su información, es decir, realizar un manejo acertado de todo su inventario de recursos, voluntarios y coordinadores. Además, de mantener actualizada con la mayor disponibilidad posible para realizar las tareas de asignación, distribución, e inventariado.

Por lo expuesto, la Fundación requiere diseñar una solución informática mediante software modular, que permita integrar un conjunto de componentes de software libre y que permita futuras adiciones, modificaciones, de módulos según las necesidades futuras y emergentes identificadas durante la puesta en marcha y crecimiento de la fundación.

La herramienta Odoo en su versión comunitaria (Odoo, 2023) dispone de módulos como el de inventario y CRM que pueden adaptarse para que cubra las necesidades

requeridas de una organización. Considerando el diseño modular que se plantea implementar sobre Odoo además de los ajustes y adaptación para la Fundación Jóvenes del Futuro, la herramienta promete ayudar de manera eficiente a cubrir las necesidades identificadas como son el registro y consulta de las donaciones y patrocinadores.

Objetivos

Objetivo general

Realizar la adecuación de los módulos de Odoo Community versión 14 inmersos en el proceso de recepción y entrega de donaciones con base en los requerimientos específicos de la Fundación Los Jóvenes del Futuro, mediante la personalización de módulos existentes en la Open Source Odoo Community.

Objetivos específicos

- Obtener y analizar los requerimientos específicos para la automatización del registro de las donaciones, de la Fundación Los Jóvenes del Futuro.
- Estudiar el Open Source Odoo Community para identificar los pasos a seguir para la personalización de los módulos requeridos para el prototipo de registro de donaciones.
- Adaptación de módulos de Odoo Community version 14 como prototipo web para facilitar la gestión de registro de las donaciones de la Fundación Los Jóvenes del Futuro mejorando la atención de los beneficiarios.

Metodología

A continuación, se detalla la metodología utilizada para el proceso de adaptación de la herramienta Odoo bajo el marco de trabajo para el desarrollo ágil SCRUM que se basa en dividir el desarrollo en ciclos breves que facilitan el avance (SCRUM, 2023), y junto a técnicas e instrumentos de investigación que “proporcionan una mayor profundidad de búsqueda” (Cisneros A, 2021, p. 1172) se realizó el levantamiento de las

necesidades y requerimientos para el cumplimiento de los objetivos planteados en el presente proyecto.

Como parte del marco de trabajo SCRUM se determinó el desarrollo de las fases planteadas por (SCRUM, 2023) que son: Planeación (donde se determina el Product backlog, lista de tareas a realizar en el proyecto), Sprint (definido en el SCRUM board, representación visual del trabajo a realizar por el equipo SCRUM) y el cierre, en la cual se validó el cumplimiento de la planeación (mediante un checklist del Product backlog).

Para el levantamiento de las necesidades del product owner, se implementó la metodología de investigación cualitativa, de la cual se utilizaron las técnicas:

- Entrevista (Anexo A) que consiste en que el investigador plantea un grupo de preguntas que le permite obtener información relevante para su estudio decisiones (Cisneros A, 2021, pp. 1171-1172), y que permitió identificar las necesidades generales de la Fundación; (Gortaire-Morejón, Matute, Romero, & Tinajero, 2022, pp. 205-208)
- Análisis documental, que se basa en la recolección y posterior análisis de los documentos existentes para la toma de decisiones (Cisneros A, 2021, pp. 1171-1172), que nos permitió entender y determinar la lista de características del producto con base en la documentación, fichas de registro, otros manejados por la Fundación y que se convirtieron en el Product Backlog.

En la tabla 1, se proporciona una identificación detallada de los roles desempeñados por los participantes en este proyecto, de acuerdo con los principios y prácticas de la metodología SCRUM:

Tabla 1
Roles del proyecto

Rol	SCRUM	Detalle
------------	--------------	----------------

Usuario	Se registra en la fundación	Usuario externo que se puede inscribir para la recepción de donaciones
Fundación	Product Owner	Identifica las necesidades del negocio o Product backlog
Investigador y Tutor (Gustavo Navas)	Scrum master	Gestiona y asegura el proceso SCRUM y sus mecánicas.
Investigador (Sebastián Peña)	Developer	Interviene en la planeación y genera los entregables del proyecto

Nota. Asignación de roles según metodología SCRUM. Fuente: Elaborado por el Autor

CAPITULO I

MARCO TEÓRICO

2.1. Fundación Jóvenes del Futuro

La Fundación Jóvenes del Futuro, desarrolla sus actividades sin fines de lucro habilitada por la Resolución ministerial No. 0194 lo que la convierte en una institución legalmente constituida por el MIES en Quito-Ecuador. Trabaja en pro del desarrollo de niños, niñas, adolescentes, jóvenes, adultos, adultos mayores, personas con discapacidades y sus familias.

Entre sus principales actividades se encuentran:

- Adquisición y posterior donación de insumos educativos, tales como textos escolares o insumos de papelería,
- Adquisición y posterior donación de equipos tecnológicos como, computadores, tabletas, calculadoras, otros.
- Gestión y disposición para la comunidad de servicios de asistencia educativa, psicológica y médica según disponibilidad de colaboradores.

La Fundación Jóvenes del Futuro entre sus funciones desarrolla actividades de recepción de donaciones, catalogación, servicios comunitarios, atenciones médicas y psicológicas, entre otros. Estos servicios actualmente no se encuentran sistematizados, o implementados en soluciones tecnológicas.

2.2. C.R.M. (*Customer Relationship Management*)

Existen varias opciones de software CRM en el mercado que permiten realizar la gestión de inventarios, ventas, usuarios, u otros módulos. Entre las más conocidas se encuentran: Microsoft Dynamics, creado por Microsoft que oferta licencias para instalación en los entornos propios del usuario y como servicio mediante la aplicación Web denominada Microsoft Dynamics CRM on Line (Microsoft, s/f.).

Otra de las opciones de software CRM es Odoo, una iniciativa con características similares a Microsoft Dynamics, que tiene como misión “Ofrecer un conjunto de

aplicaciones para empresas fáciles de utilizar que forman una caja de herramientas para acompañar a cualquier negocio que las necesite” (Odoo S.A, 2023, s/f).

La diferencia más marcada entre las dos opciones es la propuesta de Odoo, que mantiene una “Comunidad” para el desarrollo de versiones del software libre de la herramienta denominada “software local” que es utilizado por más de 5 millones de usuarios (Odoo S.A, s/f, pp4).

2.3. Odoo

Odoo es un sistema E.R.P. (*del inglés Enterprise Resource Planning*) de código abierto que integra una amplia gama de aplicaciones empresariales para gestionar y automatizar diversas funciones, como contabilidad, recursos humanos, ventas, inventario, fabricación y más. Desarrollado en Python (Odoo, Documentación oficial, 2023).

Odoo ofrece una arquitectura modular que permite a las empresas personalizar y ampliar sus funcionalidades de acuerdo a sus necesidades específicas. La interfaz de usuario intuitiva y su capacidad para centralizar la gestión de procesos empresariales hacen que Odoo sea una solución popular para empresas de diversos tamaños y sectores.

2.4. Odoo Enterprise

Odoo Enterprise es la versión comercial y de pago de Odoo, esta opción ofrece características adicionales y servicios exclusivos diseñados para satisfacer las necesidades de empresas más grandes y complejas (Odoo, 2023).

Entre las características y diferencias de Odoo Enterprise tenemos:

Módulos Adicionales: Odoo Enterprise incluye módulos específicos y funcionalidades avanzadas que no están disponibles en la versión comunitaria. Estos módulos pueden abordar áreas específicas de negocios como recursos humanos, gestión de proyectos, e-commerce, entre otros.

Soporte Técnico y Actualizaciones: Los usuarios de Odoo Enterprise tienen acceso a un soporte técnico más avanzado y a actualizaciones exclusivas, lo que garantiza que estén utilizando la última versión del software con mejoras continuas y correcciones de errores.

2.5. Odoo Community

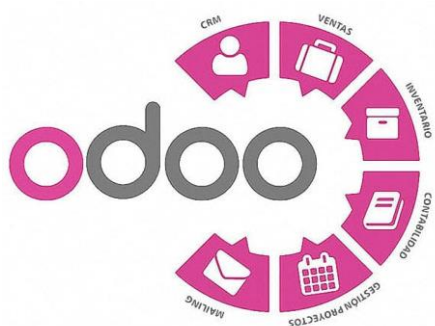
Odoo Community es una opción de código abierto que “permite que otras personas usen, cambien y distribuyan el proyecto...” (GitHub, s/f, párrafo 1), lo que permite que la comunidad realice la personalización de módulos según la necesidad de la empresa.

Odoo Community desde ahora identificado como Odoo para simplificar, permite incorporar y desarrollar un módulo a medida para cubrir las necesidades de los clientes, lo que demuestra un gran nivel de flexibilidad “tal que se pueden añadir aplicaciones dependiendo del crecimiento de su empresa, añadiendo las aplicaciones de una en una a medida que sus necesidades cambien y su clientela crezca.” (Odoo S.A, s/f, pp 6.).

2.6. Arquitectura modular Odoo

La arquitectura de Odoo por su característica de ser modular permite la integración de distintos módulos o programas los cuales se acoplan dependiendo de los requerimientos o giro del negocio, como se puede ver en la Figura 1, cada icono el cual representa un módulo o programa, es como un engrane que se junta a Odoo, para consolidar una herramienta más completa (Odoo, 2023).

Figura 1
Módulos Odoo



Nota. Figura tomada de Laboratorio Linux, 2022. Fuente: (LaboratorioLinux, s/f.)

A continuación, se realiza una descripción de los módulos existentes en Odoo y que se encuentran disponibles para el usuario:

2.6.1. CRM

Yépez y Vásquez definen al CRM como un proceso para “conocer las necesidades del cliente, mantener el dialogo permanente con el cliente generando confianza a largo plazo, la unión, coordinación y articulación de las funciones de una empresa para mejorar productos o servicio” (2020).

En el caso de Odoo CRM, ayuda a gestionar los procesos de ventas, gestión de usuarios y sus roles, así como el flujo de trabajo.

2.6.2. Ventas

Aplicación para ejecutar el proceso de ventas, según (Odoo, 2023) el módulo ejecuta el proceso desde la cotización hasta la orden de venta y posterior generación de las facturas correspondientes.

2.6.3. Inventario

Sistema de gestión del inventario y almacenes, que automatiza el abastecimiento y adquisiciones. (Odoo, 2023).

2.6.4. Contabilidad.

Este módulo facilita la administración contable de todo el negocio, sincronizando cuentas bancarias, balances, impuestos y demás actividades propias de la actividad. (Odoo, Odoo, 2023).

2.6.5. Gestión proyectos

Una vez vinculado el correo electrónico de la organización, es posible generar tareas de proyectos con base a los correos recibidos. (Odoo, 2023).

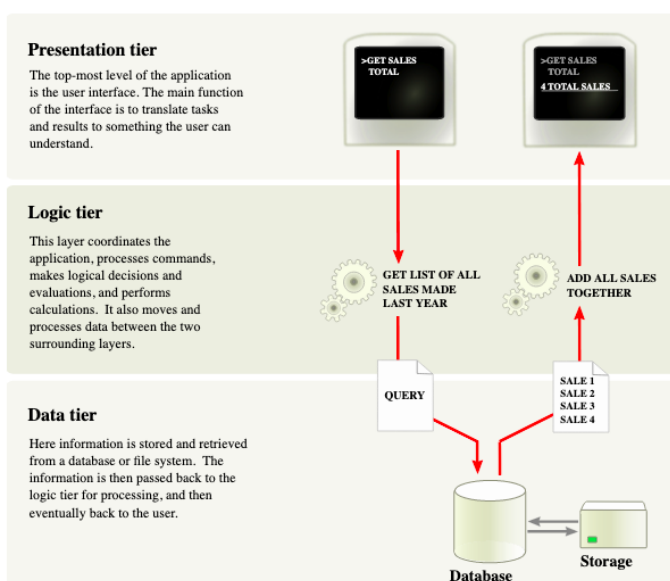
2.6.6. Mailing

Permite generar campañas de mail, que se envía automáticamente a los usuarios seleccionados de la base de datos de clientes. (Odoo, 2023).

2.7. Personalización Odoo

Odoo adopta una arquitectura de múltiples capas, lo que implica que la presentación, la lógica empresarial y el almacenamiento de datos están claramente diferenciados. En términos más específicos, se basa en una arquitectura de tres niveles (Odoo, 2023) como se puede apreciar en la figura 2 están esquematizadas las distintas capas que involucran la arquitectura.

Figura 2
Arquitectura Odoo



Nota: Arquitectura que utiliza Odoo para sus módulos. Fuente: (Odoo, Documentación oficial, 2023)

Composición de un módulo en Odoo

Un módulo en Odoo puede incorporar diversos elementos, entre los cuales se encuentran:

Objetos de negocio:

Los objetos de negocio, pueden ser distintas abstracciones del mundo real, relacionados al giro de negocio, como podría ser una factura, que a su vez se pueden definir como clases de Python. Los campos de estas clases se asocian automáticamente a las columnas de la base de datos mediante la capa O.R.M. (*Object Relational Mapping*).

Vistas de objetos:

Estas definen la presentación de la interfaz de usuario.

Archivos de información:

Son archivos XML o CSV que declaran datos del modelo, como vistas o informes, configuración de módulos, reglas de seguridad, datos de demostración, entre otros.

Controladores web:

Responsables de gestionar las solicitudes provenientes de navegadores web.

Datos web estáticos:

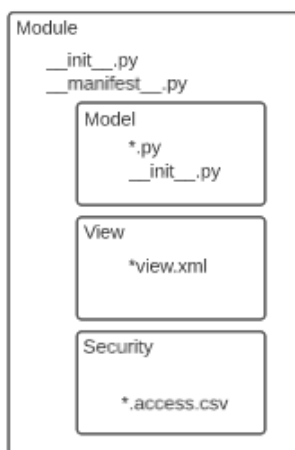
Incluyen imágenes, archivos CSS o JavaScript utilizados por la interfaz web o el sitio web.

Es importante destacar que ninguno de estos elementos es obligatorio, y la composición de un módulo puede variar. Algunos módulos pueden centrarse en agregar archivos de datos específicos, como configuraciones contables para un país, mientras que otros pueden enfocarse únicamente en agregar objetos comerciales (Odoo, Documentación oficial, 2023).

Estructura del módulo

Cada módulo se representa como un directorio dentro de un *directorio de módulos*, en la figura 3 se puede apreciar la estructura básica de un módulo Odoo. (Odoo, Documentación oficial, 2023).

Figura 3
Estructura básica de módulos Odoo

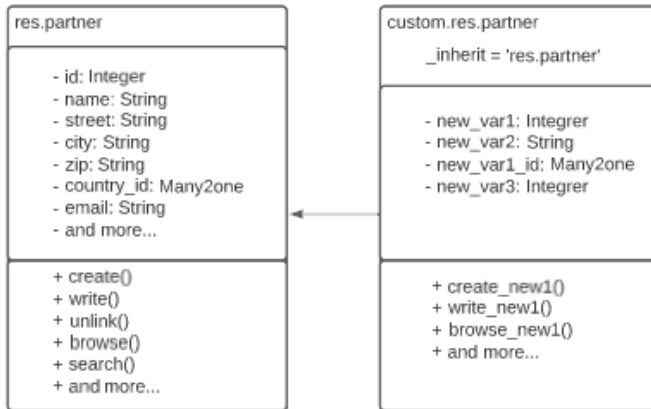


Nota. Representación UML de la estructura básica de módulos Odoo. Fuente: Elaborado por el autor.

La personalización y/o extensión de funcionalidades de Odoo, es un proceso que involucra la creación de archivos Python y XML para definir el comportamiento y apariencia del módulo. El comportamiento va a estar dictado por los requerimientos y giro del negocio específico de cada empresa.

Por consiguiente, para extender la funcionalidad de un módulo hay que tener claridad de los objetos relacionados en el requerimiento, así como de los nuevos objetos que se van a agregar. En la figura 4, se muestra como sería la extensión de un módulo, sabiendo que se van a agregar nuevos campos a un determinado módulo.

Figura 4
Personalización y/o extensión de módulos



Nota. A la izquierda clase base `res.partner`, a la derecha clase personalizado que hereda funcionalidades de `res.partner`. Fuente: Elaborado por el autor.

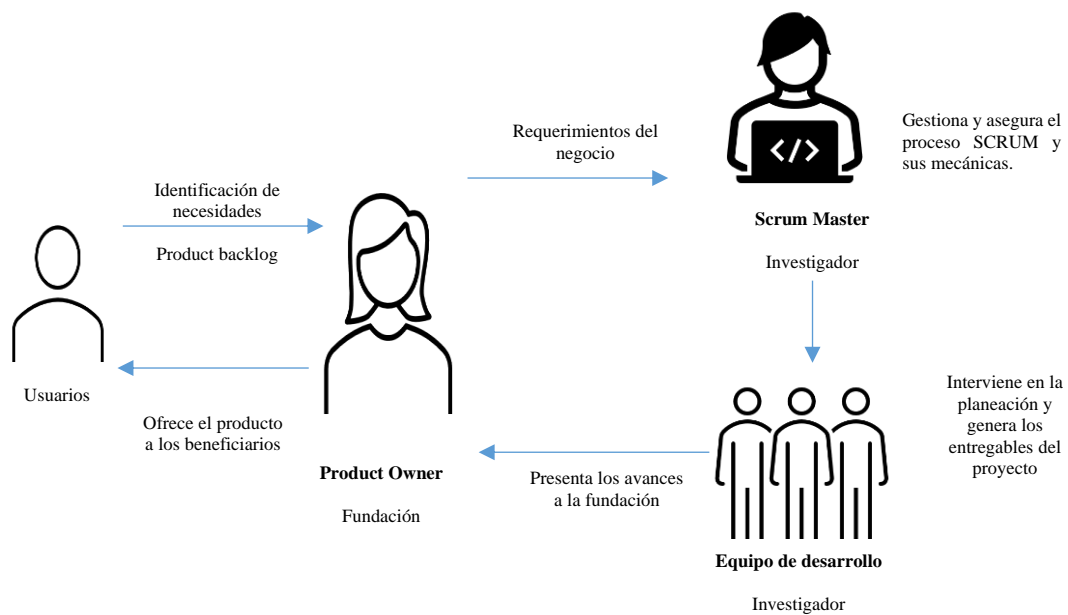
CAPITULO II

DISEÑO Y CONSTRUCCION DEL PROTOTIPO WEB

Cómo se detalló en la metodología, para el desarrollo del prototipo se hará uso de SCRUM, en cada una de sus fases: Planeación, Sprint y cierre (SCRUM, 2023) y, con la finalidad de determinar que roles tomaría cada miembro del grupo de trabajo se desarrolló entre los intervinientes la siguiente gráfica de roles Figura 5.

A lo largo de este trabajo de titulación el prototipo desarrollado para la Fundación, será mencionado como ‘Proyecto Prototipo JF-Odoo’ o simplemente ‘Prototipo JF-Odoo’ con el propósito de concisión, claridad y simplificar la referencia al mismo.

Figura 5
Roles del proyecto



Nota. Representación de la interacción de los involucrados del proyecto para su planificación y roles que deberán tomar en su desarrollo. Fuente: Elaborado por el autor.

3.1. Planeación

Para el levantamiento de los requerimientos del usuario (product owner) tal como se especificó en el apartado metodología, se utilizó como primera estrategia la entrevista, con el instrumento de cuestionario abierto, obteniendo los siguientes requerimientos que

se presentan a continuación en la Tabla 2. como una “lista emergente y ordenada de lo que se necesita para mejorar el producto” (Schwaber & Sutherland, n.d., p. 22) o product backlog:

Tabla 2
Product Backlog inicial

Product backlog inicial	
Producto Solución informática que permita el registro de los distintos datos de los actores y recursos involucrados en el proceso de patrocinio a los beneficiarios de la ONG los Jóvenes del Futuro, como aporte a la continuidad de la labor social desarrollada por la Fundación.	
Registro de beneficiarios y donantes.	Como insumo para el levantamiento del módulo de beneficiarios la Fundación Entregó la ficha para el registro de estos (Anexo B) que solicitan mantengan los formatos.
Registro de inventario	Aunque no posee a la actualidad un inventario digitalizado, la fundación maneja un inventario físico de los materiales que dispone, gestiona sus donaciones con base a la disponibilidad de los insumos, sin mayor control automatizado, y sin catalogación que permita relacionar la disponibilidad con la necesidad de los posibles donantes o patrocinadores registrados en el sistema.

Nota. Product backlog inicial de levantamiento de requerimientos. Fuente: Elaborado por el Autor

Posterior a la verificación del product backlog entregado por el product owner, los desarrolladores proceden a identificar las posibilidades, teniendo para ello en cuenta que la metodología específica que “los desarrolladores que estarán haciendo el trabajo son responsables del dimensionamiento” (Schwaber & Sutherland, n.d.).

3.1.1. Dimensionamiento

Con base en el product backlog entregado por la fundación y la capacidad operativa del equipo de desarrolladores, se acuerda generar el prototipo web con base en los

siguientes módulos que se convirtieron en el product backlog final, que se puede apreciar en la tabla 3.

Tabla 3
Product backlog final

Product backlog	Descripción
Módulo de gestión de usuarios	Los usuarios requieren de un espacio para su registro y catalogación según sus roles, los cuales pueden ser, beneficiario, patrocinador, colaborador y administrador.
Módulo de inventario,	Las donaciones receiptadas deben tener un espacio propio para su registro tanto si son objetos físicos como servicios.
Documentación	El usuario debe contar con documentación que le permita entender el funcionamiento del prototipo web y a los futuros desarrolladores dar continuidad al proceso de implementación.

Nota. Product backlog final de levantamiento de requerimientos. Fuente: Elaborado por el Autor

3.2. Sprints

Una vez acordada la dimensión de la solución, el equipo de proyecto inicia la primera sesión de planificación del sprint con los siguientes resultados detallados a continuación:

3.2.1. Sprint 1

En este sprint se realizó la configuración e instalación base del entorno de pruebas sobre un V.P.S. (*del inglés Virtual Private Server*), así como la instalación inicial de Odoo.

Objetivos:

Definición de entorno: Se utilizará como entorno de trabajo Odoo, con base en los módulos comunitarios.

- Generar la primera implementación en un entorno de prueba basado en S.O. Ubuntu.
- Realizar la implementación sobre la versión 14.0 comunitaria de Odoo

Tabla 4
Tareas del Sprint 1

Qué	Cómo	Quien	Cuando
Generación de entorno web	Preparación de Hardware: 4 CPU, 4 RAM, 40 GB SSD (Ambiente no distribuido, modo pruebas)	Sebastián Peña	Día 1
	Instalación de SO Ubuntu 20.04	Sebastián Peña	Día 1
	Instalación de bibliotecas y dependencias necesarias	Sebastián Peña	Día 1
	Preparación de sistema operativo	Sebastián Peña	Día 2
	Descarga de fuentes Odo de https://github.com/odoo/odoo.git	Sebastián Peña	Día 3
	Instalación Odo	Sebastián Peña	Día 3
	Instalación prerrequisitos Odo	Sebastián Peña	Día 3
	Creación de archivo de configuración Odo	Sebastián Peña	Día 3
	Configurar parámetros de conexión a base datos, administración de la consola de administración de Odo para gestionar las bases de datos y puerto de conexión:	Sebastián Peña	Día 3
	Configuración de parámetros iniciales de instalación e inicio.	Sebastián Peña	Día 3
	Pruebas de funcionamiento	Gustavo Navas	Día 4

Nota. Product backlog final de levantamiento de requerimientos. Fuente: Elaborado por el Autor

3.2.2. Ejecución del sprint 1

Como ambiente de pruebas se eligió utilizar una instancia cloud VPS, por la disponibilidad que brinda, y dado que se trata de una conexión remota es necesario

gestionar la conectividad para ingresar al VPS, conexión que se realiza mediante ssh (*del inglés Security Shell*) para que el usuario developer pueda acceder al servidor y realizar las configuraciones e instalaciones de librerías y programas.

En la etapa de instalación, se procedió con la instalación de las librerías necesarias (Odoo, 2023), las cuales son un conjunto de utilidades Python, PostgreSQL, conectores para que todo el ecosistema que compone Odoo se pueda comunicar sin problema.

Descarga de fuentes, como parte importante de Odoo Community es siempre estar actualizado con los repositorios oficiales (Odoo, Repositorio oficial, 2023).

Con las fuentes descargadas, se procedió a instalar las bibliotecas que están adjuntas en el archivo requirements.txt de Odoo (Odoo, Documentación instalación, 2023) y se configura el archivo odoo.conf el cual define parámetros conexión hacia base de datos y parámetros de rendimiento para Odoo, para más detalles de instalación revisar Anexo C.

Generación de script para ejecución automatizada en el servidor, es un proceso opcional donde se definen un conjunto de script para bash de Linux, que permiten ejecutar Odoo de manera automática o más rápida, mediante la ejecución ciertos comando como iniciar, detener, revisar logs e inicio automático cuando se reinicie el servidor cuando está en modo de producción.

Cumplidos los requerimientos se realizó el proceso de verificación de Odoo.

<http://116.202.96.191:7000/web>

Figura 6
Pantalla de inicio de Odoo

The screenshot shows the Odoo web interface for creating a new database. The URL in the browser is 116.202.96.191:7000/web/database/selector. The form includes the following fields: Master Password (with a toggle for visibility), Database Name, Email, Password (with a toggle for visibility), Phone number, Language (dropdown menu set to English (US)), and Country (dropdown menu). There is also a checkbox for 'Demo data'. At the bottom, there is a blue button labeled 'Create database' and a link 'or restore a database'.

Nota: Pantalla de creación de base de datos inicial de Odoo. Fuente: Sistema Odoo

Tabla 5
Validación de las tareas del Sprint 1

Tarea	Ejecutada si/no	Observación
Preparación de Hardware: 4 CPU, 4 RAM, 40 GB SSD (Ambiente no distribuido, modo pruebas)	SI	En línea
Instalación de SO Ubuntu 18.04	SI	En línea
Instalación de bibliotecas y dependencias necesarias	SI	En línea
Preparación de sistema operativo	SI	En línea
Descarga de fuentes Odoo de https://github.com/odoo/odoo.git	SI	En línea
Instalación Odo	SI	En línea
Instalación prerrequisitos Odoo	SI	En línea
Creación de archivo de configuración Odoo	SI	En línea
Configurar parámetros de conexión a base datos, administración de la consola de administración de	SI	En línea

Odoo para gestionar las bases de datos y puerto de conexión:		
Configuración de parámetros iniciales de instalación e inicio.	SI	En línea
Pruebas de funcionamiento	SI	En línea

Nota. Validación de Sprint. Fuente: Elaborado por el Autor

3.2.3. Sprint 2

En este sprint se realizó la instalación de los módulos correspondientes y asignación de permisos de usuario.

Objetivos:

Determinación de módulos y las seguridades, solo usuarios autorizados podrán tener acceso al sistema, según los roles determinados.

- Se instalarán solo los módulos “Inventario” para manejar las donaciones que llegan a la fundación y “CRM” para registrar los beneficiarios, patrocinadores y voluntarios.
- Los usuarios no podrán asignar roles, solo el administrador podrá realizarlo.
- Cada usuario deberá contar con su espacio de perfil según rol de usuario:
 - Receptor o usuario general
 - Donador
 - Administrador.
- Debe existir un espacio de autenticación tipo formulario con los campos de usuarios y contraseña.

Tabla 6
Tareas del Sprint 2

Qué	Cómo	Quien	Cuando
Instalación de módulos y asignación de usuarios	Instalación de módulo de CRM	Sebastián Peña	Día 1
	Instalación de módulo de Inventario.	Sebastián Peña	Día 1
	Configuración de perfiles de usuario	Sebastián Peña	Día 2
	Registro de productos de prueba	Gustavo Navas	Día 2

	Pruebas de funcionamiento	Gustavo Navas	Día 2
	Validación Inicial	La Fundación	Día 3 y 4

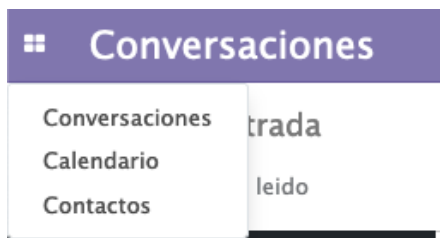
3.2.4. Ejecución del sprint 2

Una vez instalado el ambiente en el sprint 1, se procede con la instalación de los módulos CRM e INVENTARIO (Anexo D), que están disponibles en la lista de aplicaciones compatibles con Odoo Community versión 14.

Posterior a la instalación se definen los permisos y nivel de visibilidad para los usuarios voluntarios que utilizaran la aplicación, es decir, se designa las opciones que estarán visibles para los usuarios; para realizar esta configuración se hace uso de la administración de perfiles de Odoo Community versión 14 en las opciones de ajustes, para un mayor detalle de la realización de esta tarea revisar el Anexo E.

Como se puede apreciar en la Figura 7. Para el perfil definido como voluntario se le asigno los niveles de visibilidad de las opciones que permiten ver en su menú de opciones la opción de contactos, para mostrar información de beneficiarios y hacer el registro de beneficiarios.

Figura 7
Permisos asignados a usuario voluntario



Nota: Visualización opciones luego de configurar el nivel de accesibilidad de Odoo. Fuente: Sistema Odoo personalizado.

Tabla 7
Validación de tareas del Sprint 2

Tarea	Ejecutada si/no	Observación
Instalación de módulo de CRM	SI	En línea
Instalación de módulo de Inventario.	SI	En línea
Registro de usuarios de prueba	SI	Configurado
Registro de productos de prueba	SI	Registrado
Pruebas de funcionamiento	SI	En línea
Validación Inicial	SI	Correcto

3.2.5. Sprint 3

En este sprint se realizó la personalización o adaptación de módulos, según las necesidades del producto owner, para cumplir con el formato de la ficha beneficiario.

Objetivos:

Agregar los campos necesarios en la vista de usuario para que refleje la información de la ficha beneficiarios.

Tabla 8
Tareas del Sprint 3

Qué	Cómo	Quien	Cuando
Personalización de módulos	Creación de módulos	Sebastián Peña	Día 1
	Definición de variables	Sebastián Peña	Día 1
	Revisión de persistencia de datos	Sebastián Peña	Día 2
	Registro de prueba	Gustavo Navas	Día 2

	Pruebas de funcionamiento	Gustavo Navas	Día 2
	Validación Inicial	La Fundación	Día 3 y 4

3.2.6. Ejecución del sprint 3

Para la personalización de módulos se partió con la realización de las siguientes tareas:

- Instalación de librerías requeridas para despliegue y ejecución de Odoo
- Preparación de un ambiente para desarrollo con el IDE PyCharm Community y PostgreSQL
- Creación de ambiente virtual para entorno de desarrollo e instalación de bibliotecas de Odoo.
- Creación y personalización de módulos.

En la instalación de bibliotecas preparativas para el desarrollo y ejecución se hace énfasis en las bibliotecas que están fuera del conjunto de sugerencias en la documentación oficial de Odoo (Odoo, 2023) ya que muchas bibliotecas son dependientes de otras que por obsolescencia de la tecnología están en constante actualización u olvidadas para el desarrollo por que salió una nueva tecnología.

Como IDE de desarrollo se optó por usar PyCharm Community (Jetbrains, 2023) por su facilidad y optimización para trabajar con proyectos de lenguaje Python el cual es nativo para el desarrollo de Odoo.

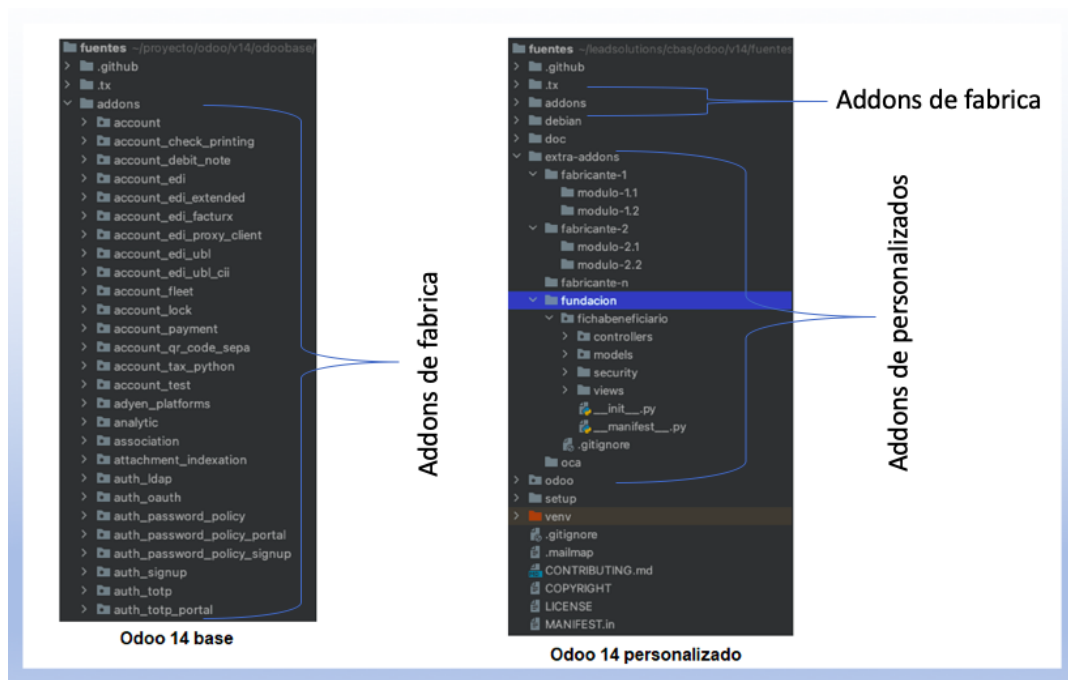
A partir de la instalación y ejecución de PyCharm como entorno de desarrollo como fase preparativa del ambiente de desarrollo es la creación de ambientes virtuales (Python, 2023) los cuales permiten tener una independencia de bibliotecas para un determinado proyecto, para tener flexibilidad a la hora de buscar bibliotecas adicionales que se integren con facilidad y no afecten la funcionalidad de otros proyectos.

En la personalización se buscó adecuar el módulo de CRM con los ajustes necesarios para tener la información que la fundación maneja y requiere en la ficha de beneficiarios.

Para extender la funcionalidad inicial que viene de fábrica en Odoo se agregó un nuevo módulo llamado *fichabeneficiario* dicho modulo cumple con los requerimientos de arquitectura de Odoo (Odoo, Documentación oficial, 2023)

En la Figura 8, podemos visualizar la totalidad de Odoo desde su forma base y un Odoo personalizado, cumpliendo con lineamientos y practicas sugeridas en la documentación oficial (Odoo, Documentación oficial, 2023), para no intervenir con el código base los nuevos módulos son agrupados dentro del directorio *extra-addons*, dentro de este directorio a su vez se pueden crear distintos directorios para referenciar a repositorios de terceros, repositorio Enterprise en caso de entornos de producción licenciados o en este caso particular módulos personalizados para la *fundación*.

Figura 8
Comparativa entre Odoo base y Odoo personalizado

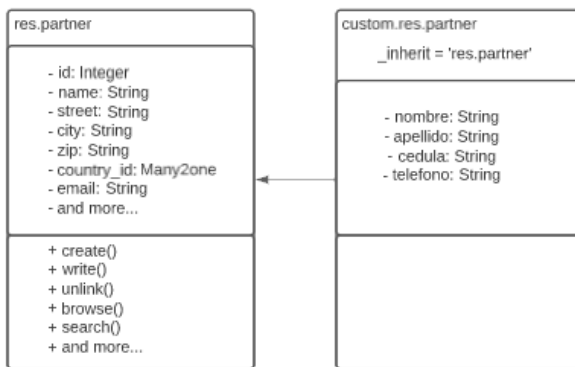


Nota: Arquitectura de directorio Odoo para personalización de código fuente. Fuente: Elaborado por el autor.

Para extender las funcionalidades del módulo CRM en el nuevo módulo *fichabeneficiario*, se han añadido en la capa de modelo de datos y vista las líneas de código correspondientes para referenciar cada uno de los nuevos campos. En la Figura 9, se observa un diagrama que representa cómo se realiza la herencia y se incorporan los campos adicionales en el modelo de datos.

Figura 9

Extensión de funcionalidad para nuevo modulo

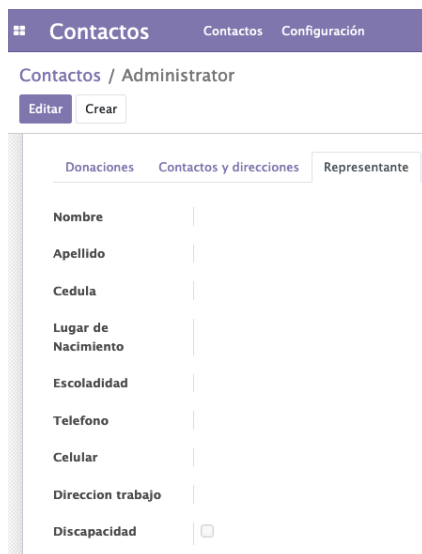


Nota: A la izquierda clase base, a la derecha clase heredada, para construir modulo personalizado. Fuente: Elaborado por el autor.

En la Figura 10 se puede ver la adición de los campos mencionados, para mayor detalle de la personalización revisar Anexo F.

Figura 10

Vista de campos personalizados



Nota. Campos personalizados en vista de Beneficiario. Fuente: Sistema Odoo personalizado.

Tabla 9
Validación de tareas del Sprint 3

Tarea	Ejecutada si/no	Observación
Modulo aparece en el interfaz de aplicaciones	SI	En línea
Instalación de modulo.	SI	En línea
Registro de ficha con nuevos campos	SI	Registrando
Pruebas de funcionamiento	SI	En línea
Validación Inicial	SI	Correcto

3.2.7 Sprint 4

En este sprint se realizó la personalización, para relacionar las donaciones con el beneficiario.

Objetivos:

Agregar los campos necesarios en la vista de contactos (beneficiario) para que refleje la información de la donación asociada.

Tabla 10
Tareas del Sprint 4

Qué	Cómo	Quien	Cuando
Personalización de módulos	Creación de modelo para la relación beneficiario - donación	Sebastián Peña	Día 1
	Definición de variables	Sebastián Peña	Día 1
	Revisión de persistencia de datos	Sebastián Peña	Día 2
	Registro de prueba	Gustavo Navas	Día 2
	Pruebas de funcionamiento	Gustavo Navas	Día 2

	Validación Inicial	La Fundación	Día 3 y 4
--	--------------------	-----------------	-----------

3.2.8 Ejecución del sprint 4

Empezaremos entendiendo que el término *objeto* se utiliza para referirse a un modelo de datos específico que define la estructura y el comportamiento asociado con un concepto de negocio particular.

Para comenzar con la relación de los objetos beneficiario y donación, se definió un modelo, en la figura 11 se puede apreciar que el modelo estará compuesto por una relación muchos a muchos, que involucra los objetos *product_template* y *res_partner*, para así tener una o varias donaciones asociadas a un beneficiario o viceversa.

Figura 11

Relación muchos a muchos Odoo



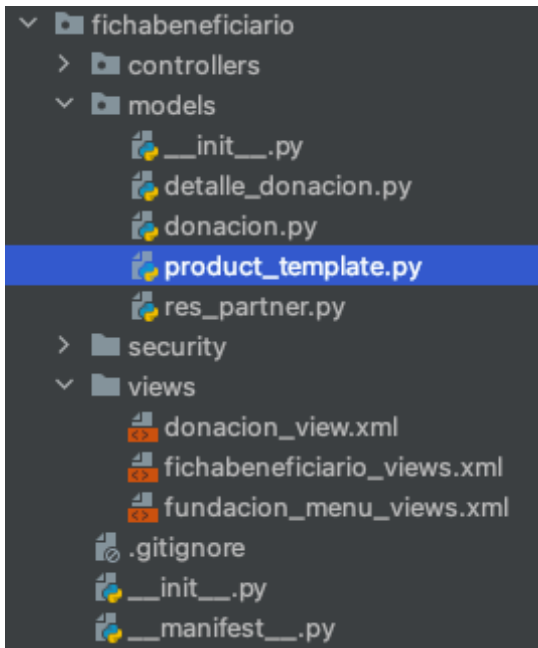
Nota. Representación de la relación del modelo de datos modelo de datos en Odoo. Fuente: Sistema Odoo personalizado.

Hay que notar que para Odoo utilizamos la tabla *product_template* que es parte del modelo ancestro el cual contiene toda la definición para el manejo de productos, al igual que *res_partner* para a manejar los a beneficiarios.

Una vez entendido cuales son los objetos que se relacionan se procede con la generación del modelo *product_template*, a nivel de código fuente, esta relación se la realiza en un archivo Python llamado *product_template.py* como se puede ver en la figura 12.

Figura 12

Modelo generado *product_template*

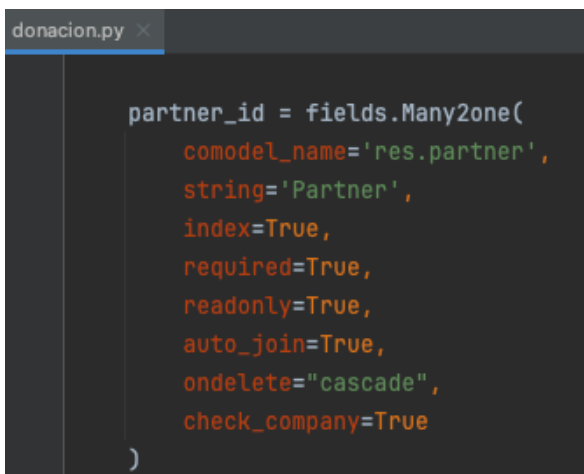


Nota. Representación de scripts Python dentro la estructura de directorios de un módulo personalizado de Odoo. Fuente: Sistema Odoo personalizado.

En la figura 13, se evidencia como se crea la relación, para asociar una donación hacia *res_partner* mediante la palabra reservada *fields.Many2one* con *res_partner*.

Figura 13

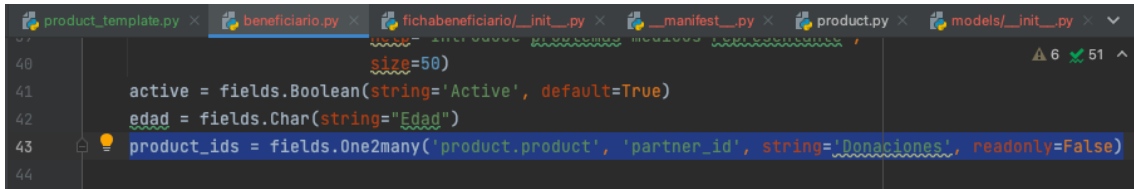
Modelo de relación entre objetos



Nota. Captura de código fuente Odoo. Fuente: Sistema Odoo personalizado

En la tabla *res_partner*, se va a generar la relación entre el nuevo modelo, como se aprecia en la figura 14.

Figura 14
Relación del modulo

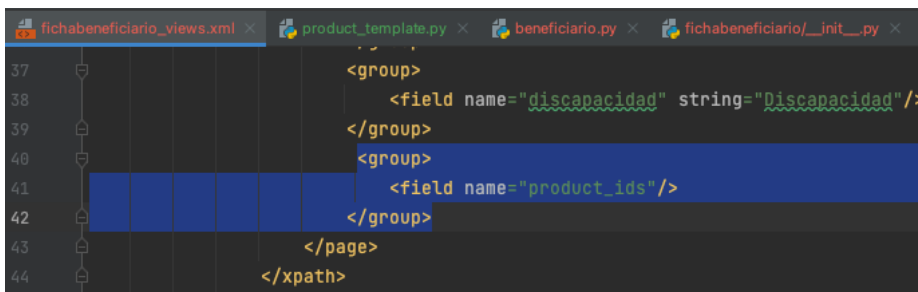


```
40         size=50)
41     active = fields.Boolean(string='Active', default=True)
42     edad = fields.Char(string="Edad")
43     product_ids = fields.One2many('product.product', 'partner_id', string='Donaciones', readonly=False)
44
```

Nota. Captura de código fuente de Odoo. Fuente: Sistema Odoo personalizado

A nivel de vista se agrega la nueva referencia de la relación, como se evidencia en la figura 15.

Figura 15
Referencia en vista de relación de objetos

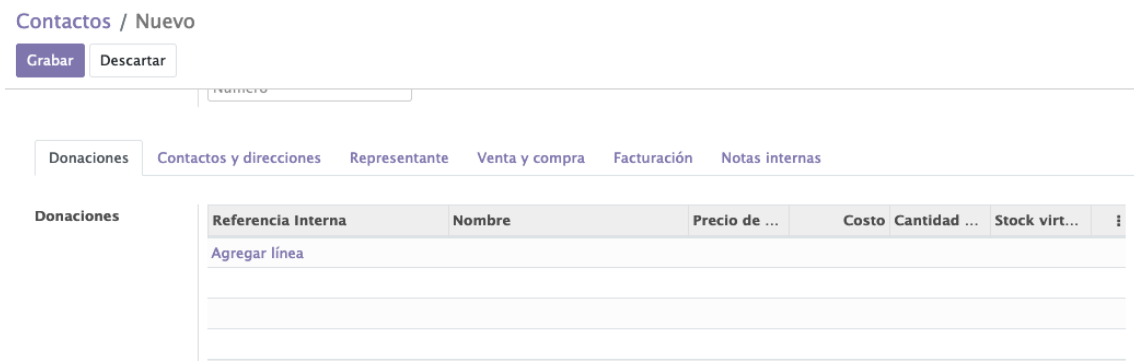


```
37     <group>
38         <field name="discapacidad" string="Discapacidad"/>
39     </group>
40     <group>
41         <field name="product_ids"/>
42     </group>
43 </page>
44 </xpath>
```

Nota. Captura de código fuente de Odoo. Fuente: Sistema Odoo personalizado

Como se puede apreciar en la figura 16, cuando está ejecutándose podemos visualizar de la relación beneficiario donación.

Figura 16
Visualización a nivel de vista de la relación de donaciones



Nota. Captura de vista personalizada luego de agregar nuevos campos. Fuente: Sistema Odoo personalizado

Revisión de sprint

Tabla 11

Validación de tareas del Sprint 4

Tarea	Ejecutada si/no	Observación
Aparece la relación personalizada en la ficha	SI	En línea
Registro de usuarios de prueba	SI	Registrado
Registro de donación beneficiario en vista de beneficiario	SI	Registrado
Pruebas de funcionamiento	SI	En línea
Validación Inicial	SI	Pasó con observaciones

3.2.9 Sprint 5

En este sprint se realizó la personalización de en la vista de contactos, agregando un campo para discriminar si es contacto patrocinador o beneficiario.

Objetivos:

Agregar los elementos necesarios en la vista de contactos para cuando se registre un nuevo contacto se pueda identificar si es beneficiario o patrocinador mediante la selección en una lista de dichas opciones.

Tabla 12

Tareas del Sprint 5

Qué	Cómo	Quien	Cuando
Personalización de módulos	Modificación del modelo para tener el campo de discriminación.	SEBASTIÁN PEÑA	Día 1
	Dedición de variables	SEBASTIÁN PEÑA	Día 1
	Revisión de persistencia de datos	SEBASTIÁN PEÑA	Día 2

	Registro de prueba	Gustavo Navas	Día 2
	Pruebas de funcionamiento	Gustavo Navas	Día 2
	Validación Inicial	La Fundación	Día 3 y 4

3.2.10 Ejecución del sprint 5

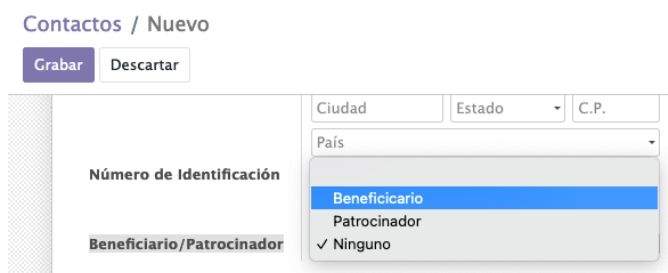
Se definió el campo *tipo_agente*, en la vista de contactos para cuando se haga un registro de contacto, hacer una selección que permita definir si es patrocinador o beneficiario, mediante una lista desplegable con las opciones “Patrocinador”, “Beneficiario” o “Ninguno”.

En la lista se muestran las opciones “Patrocinador”, “Beneficiario” o “Ninguno”, cabe notar que la opción “Ninguno” es necesaria, para no alterar el funcionamiento o discriminación de los registros por defecto, ya que Odoo maneja dentro de contactos datos que son parte de funciones primarias de la instalación básica.

Considerando que el nuevo campo pasa a ser parte de la tabla *res_partner*, se agrega el campo *tipo_agente*, por lo tanto, el modelo persiste directamente con el objeto Contacto, la modificación se resume en crear el nuevo campo.

Como se puede apreciar en la figura 16 se muestra el campo “Beneficiario/Patrocinador” a nivel de vista del módulo contactos.

Figura 17
Visualización a nivel de vista campo “Beneficiario/Patrocinador”



Nota. Captura de vista personalizada luego de agregar nuevo campo. Fuente: Sistema Odoo personalizado

Revisión de sprint

Tabla 13

Validación de tareas del Sprint 5

Tarea	Ejecutada si/no	Observación
Aparece el campo en la vista contacto	SI	En línea
Registro de campo	SI	Registrado
Pruebas de funcionamiento	SI	En línea
Validación Inicial	SI	Pasó con observaciones

3.2.11 Fase de cierre

Como parte de la Fase de cierre se realizó un proceso de capacitación al responsable de la administración del sistema en la Fundación y se indicó las referencias de la documentación oficial para que estuviera al tanto del framework.

A continuación, se muestra un flujo de registro y asignación de donación que contempla el alcance del proyecto.

Figura 18

Registro de donaciones

The screenshot shows the Odoo product registration interface. At the top, there are navigation links for 'Productos / New' and buttons for 'Grabar' and 'Descartar'. Below this is a 'Reabastecer' button. The main form area contains several sections: a top bar with '0,00 Unidades A Mano' and '0,00 Unidades Previsto', and 'Movimientos productos' and 'Reglas de rea...'. The 'Nombre producto' field is filled with 'zapatos deportivos'. There are checkboxes for 'Puede ser vendido' and 'Puede ser comprado'. Below this are tabs for 'Información general' and 'Inventario'. The 'Tipo de producto' is set to 'Almacenable', 'Categoría de producto' is 'All', and 'Precio de venta' is '\$0,00'. There are also fields for 'Referencia Interna', 'Codigo de Barras', 'Impuestos cliente', and 'Costo' (set to '0,00').

Nota. Captura del registro de una donación. Fuente: Sistema Odoo personalizado

Figura 19
Lista de donaciones en inventario



Nota. Captura las donaciones registradas. Fuente: Sistema Odoo personalizado

Figura 20
Ingreso y registro de beneficiarios

Contacts / New

Grabar Descartar

Individual Compañía

Juan Daniel Tasugano Flore

Compañía

Address: Pomasqui, Calle Reina de Quito y Catequilla
Calle 2...
Ciudad Estado C.P.
País

Número de identificación: VAT
Número

Beneficiario/Patrocinador: Beneficiario

Puesto de trabajo: Por ejemplo, Director de Ventas

Teléfono

Móvil

Correo electrónico: e.j. https://www.odoo.com

Enlace a página web

Título: Por ejemplo, Señor

Categorías: Etiquetas...

Representante Donaciones Contactos y direcciones Venta y compra Facturación Notas Internas

Nombre: Maria Isabel

Apellido: Flores

Cedula

Lugar de Nacimiento

Escoladidad: Ninguna

Telefono: 0984706091

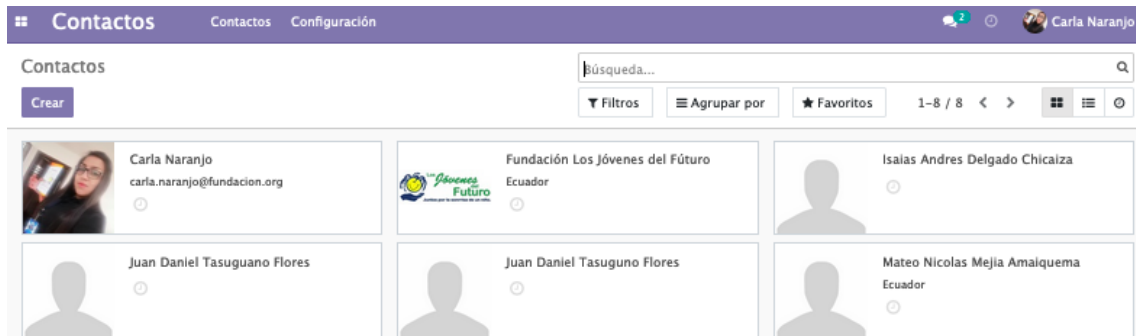
Celular: 0995165673

Direccion trabajo

Discapacidad:

Nota. Captura del registro de una donación. Fuente: Sistema Odoo personalizado

Figura 21
Lista de beneficiarios



Nota. Captura la lista de beneficiarios registrados. Fuente: Sistema Odoo personalizado

Figura 22
Asignación de donaciones



Nota. Captura del registro de una donación. Fuente: Sistema Odoo personalizado

CAPITULO III

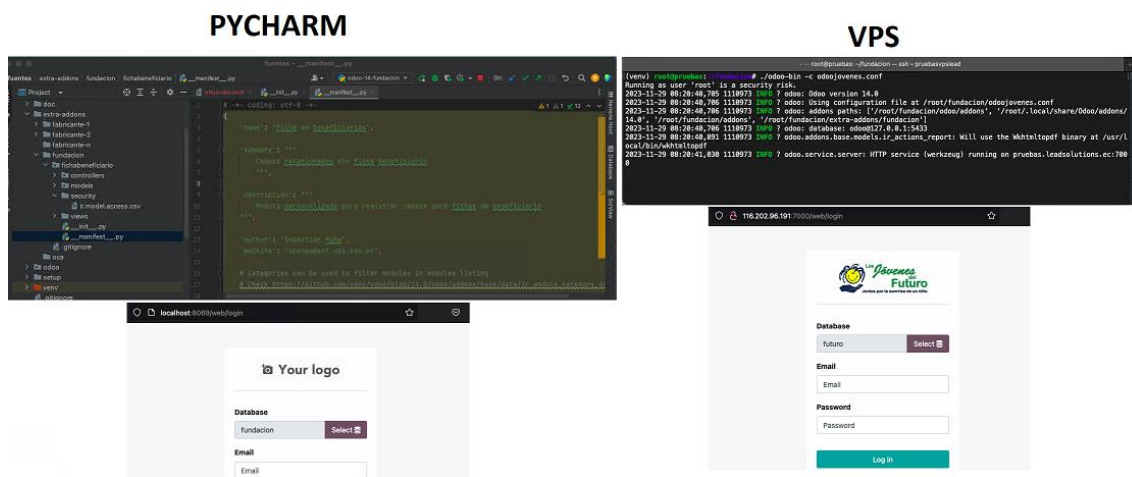
PRUEBAS Y RESULTADOS

Para realizar las pruebas de Odoo se procedió de la siguiente manera:

1. Crear instancias de desarrollo de Odoo: Considerando que se puede manejar varias formas de instalar Odoo (Odoo, Documentación instalación, 2023) esto permite tener diferentes instancias, las que pueden estar destinadas como ambiente de pruebas, desarrollo o producción. En el proyecto se optó por crear un ambiente de desarrollo, instalando Odoo localmente en una máquina destinada para este fin y un VPS a manera de ambiente de pruebas.

En la figura 17 se hace una captura a manera de comparativa del ambiente de desarrollo lanzado desde el IDE PyCharm junto al ambiente pruebas que está en un VPS cuya conexión es desde un terminal Linux y la salida de la aplicación es por una dirección IP pública.

Figura 23
Captura de ambiente de desarrollo en IDE y pruebas en VPS



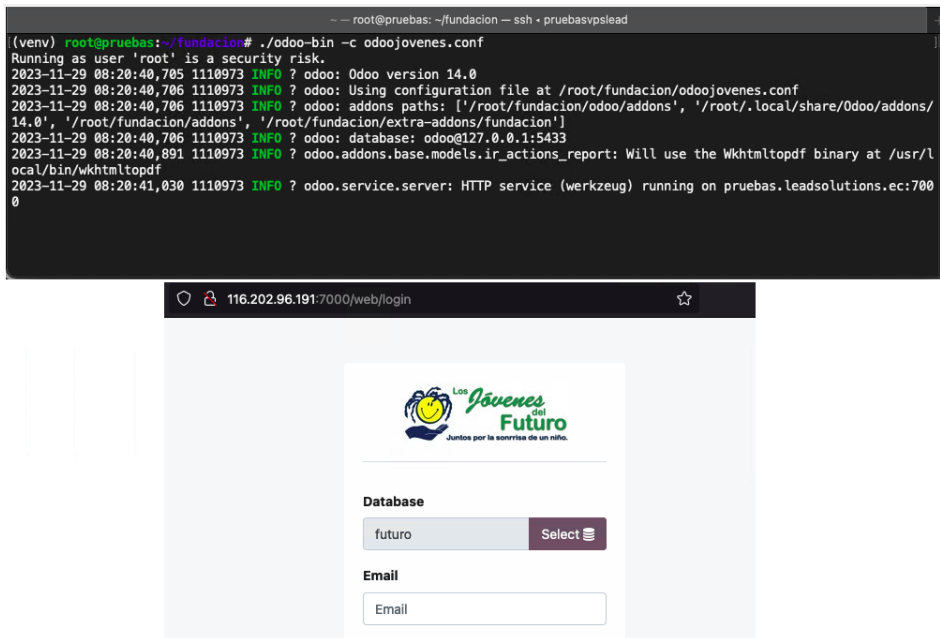
Nota. Comparativa entre ambientes de desarrollo y pruebas. Fuente: Sistema Odoo personalizado

2. Crear entorno de pruebas (VPS): A fin de aplicar el concepto de desarrollo continuo (Red Hat, 2022). Se procedió creando un ambiente de prueba en un VPS, para que esté disponible para la validación y pruebas. En la figura 18 se visualiza mediante la

consola el despliegue de la aplicación por el navegador web que está apuntando a una dirección IP publica *http://116.202.96.191:7000/web/*.

Figura 24

Captura del ambiente de pruebas VPS



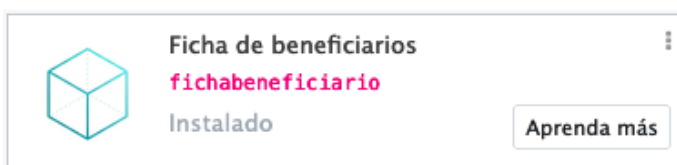
Nota. Ambiente de pruebas lanzado desde un VPS, visualización mediante terminal Linux y navegador web. Fuente: Sistema Odoo personalizado

3. Instalar y configurar los módulos personalizados (Prueba de caja negra):

Se instalo los módulos personalizados en el entorno de pruebas siguiendo los procedimientos de instalación y configuración establecidos siguiendo el asistente de instalación. La figura 19 muestra la existencia del nuevo módulo personalizado para el registro de la ficha del beneficiario.

Figura 25

Captura desde interfaz de modulo personalizado instalado



Nota. Modulo personalizado en lista de aplicación de Odoo. Fuente: Sistema Odoo personalizado

4. Crear datos de prueba (Prueba de caja negra): Como primera prueba se

utilizaron datos de las fichas de algunos beneficiarios para llenar la información del

módulo personalizado, a fin de simular situaciones específicas del negocio y verificar la funcionalidad de los nuevos módulos.

Como prueba se verifica que se están realizando el CRUD desde la aplicación Odoo, es decir, dentro del interfaz de usuario del módulo personalizado se está realizando la creación del usuario, por lo tanto para validar que se está guardando la data en la base de datos, se realiza una sentencia select hacia la tabla *res_partner*, y dicha tabla refleja la lista usuarios, en la figura 20, se pudo constatar que luego de ejecutar el select están registrados los usuarios adicionales que se han registrado usando la interfaz gráfica de Odoo.

Figura 26
Revisión de dato a nivel de base de datos

```

...cion -- psql -U odoo -d fundacion -h 127.0.0.1 ~ -- -bash
fundacion=# Select name from res_partner;
          name
-----
Carla Naranjo
Mateo Nicolas Mejia Amaiquema
Portal User Template
Public user
Odoobot
Administrator
My Company
Isaias Andres Delgado Chicaiza
Juan Daniel Tasugano Flores
Default User Template
(10 rows)

fundacion=#

```

Nota. Ejecución de sentencia select para listar nuevos elementos en terminal Linux de Odoo. Fuente: Sistema Odoo personalizado

5. Registrar y corregir errores: Con los errores y problemas encontrados durante las pruebas, se procedió a corregir el código fuente, además se lanzó una nueva pila de pruebas para validar que el error ya no persistía.

Tabla 14
Errores encontrados

Error	Solventado
Errores de configuración	SI

Acceso a base de datos	SI
VARIABLES no definidas	SI
Sintaxis en código fuente	SI

En este paso la detección de errores se lo hacía mediante la lectura constante del log de errores, y la constatación a nivel funcional, ya que, si no se visualizan las modificaciones de la implementación del módulo personalizado, no se estaría cumpliendo con el objetivo del proyecto.

6. Documentar los resultados de las pruebas: Como resultados de las pruebas realizadas, se obtuvo una lista de errores y soluciones aplicadas, las cuales quedan de elemento de retroalimentación para nuevos desarrollos o consideraciones que se debe tener en la generación de nuevos módulos.

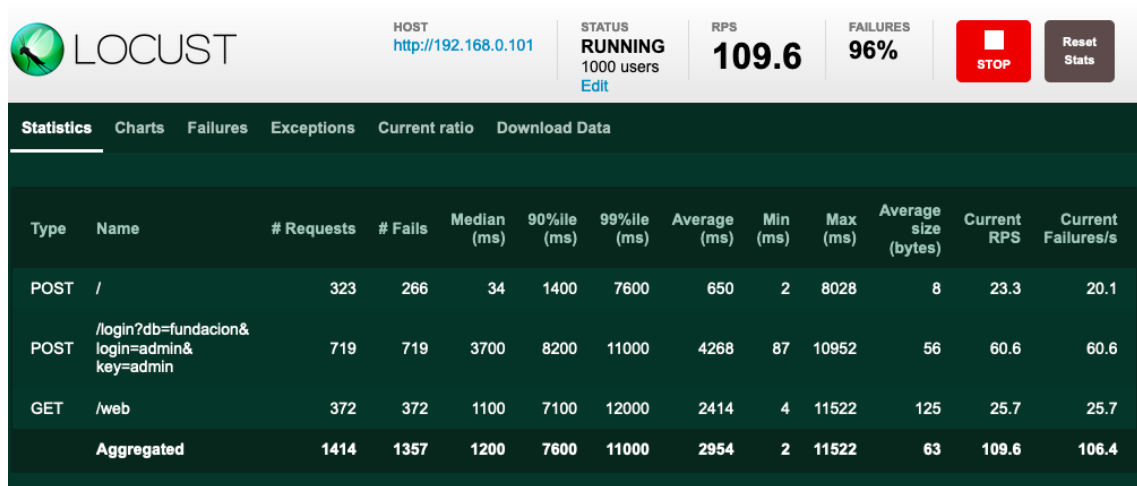
7. Realizar pruebas de integración: Una vez probados los módulos personalizados de forma individual, se realizó pruebas de integración con otros módulos de Odoo, es decir, realizar una corrida en todo el ecosistema Odoo, a fin de asegurar que los nuevos módulos personalizados funcionan correctamente en conjunto con todos componentes del sistema.

8. Realizar pruebas de rendimiento: Considerando que los módulos personalizados involucran o no procesos intensivos en términos de consumo recursos del sistema operativo, como procesamiento de grandes cantidades de datos o cálculos complejos, se realizó pruebas de rendimiento para asegurar que los módulos funcionen adecuadamente en términos de rendimiento y tiempo de respuesta.

El test de rendimiento se lo realizo con la herramienta Locust (Locust, 2023) que es un software para realizar pruebas de carga, como pruebas se planteó una carga de 1000 usuarios en 200 segundos, lo que plantea un alcance relativamente pesado para Odoo ya

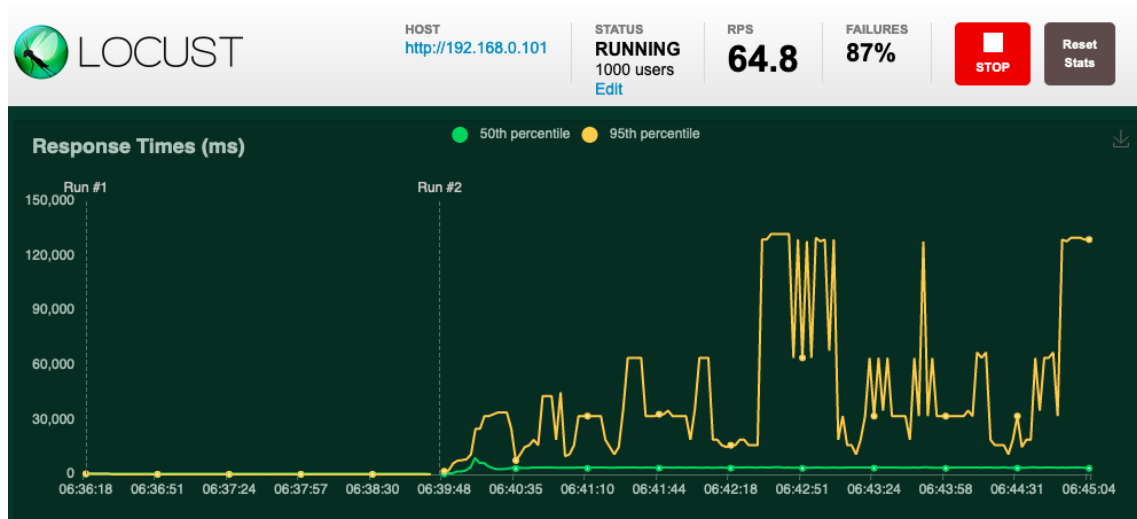
que el proyecto si se lo pone en producción tendría un estimado de no más de 10 usuarios concurrentes.

Figura 27
Pruebas de rendimiento en software Locust



Nota. Interfaz de Locus con pruebas de carga. Fuente: Sistema Locust

Figura 28
Respuesta de software Locust durante las peticiones



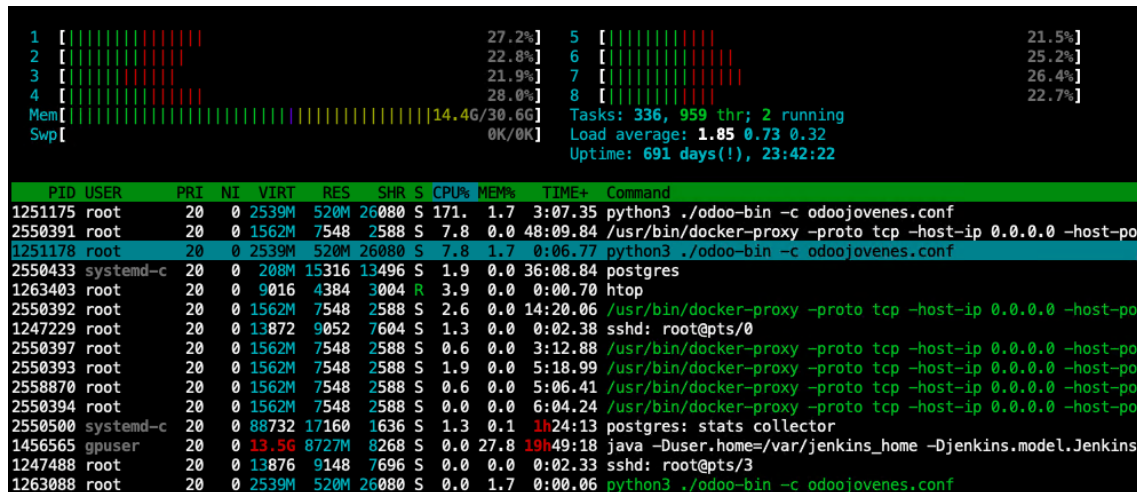
Nota. Interfaz de Locus con pruebas de carga, donde se muestra el tiempo de respuesta. Fuente: Sistema Locust

En las figuras 21 y 22, se aprecia como están llegando las peticiones de ingreso al servidor y en contraste la figura 23, muestra el efecto de las peticiones que, incluso siendo excesivas, tenemos una respuesta que no supera el 30% de consumo de CPU y el consumo

de RAM casi llega al 50% de la capacidad total del servidor, con lo que se puede concluir que tenemos una respuesta aceptable de Odoos.

Figura 29

Respuesta de saturación en servidor



Nota. Monitorización de servidor VPS de pruebas durante la ejecución de Locust con prueba de carga.
Fuente: Sistema Linux

9. Preparar para la producción: Una vez validado, probado y corregido todos los errores en el entorno de pruebas, se puede realizar el paso a producción.

Este paso queda a discreción de la Fundación, pues la implantación conlleva un costo que esta fuera del alcance de este proyecto, sin embargo, se deja evidencia de la viabilidad que tiene la implementación del proyecto y de la ayuda que podría aportar en la gestión de las donaciones.

CONCLUSIONES

Con el avance de la tecnología, día a día se lanzan nuevas herramientas que permiten solventar diferentes problemas, como software documental, ERP, gestores de contenido, en el caso de este proyecto se utilizó un ERP llamado Odoo para generar el Prototipo JF-Odoo, permitió ahorrarse tiempo de desarrollo ya que al ser modular, la personalización se centró en cumplir el objetivo del alcance del proyecto, es decir, tener un producto que permita el registro de patrocinadores, beneficiarios y donaciones.

El desarrollo del Prototipo JF-Odoo se enfocó en buscar los módulos que se adecuaban para cumplir con el alcance y crear un módulo personalizado con los campos y acciones adecuadas con la información que maneja la fundación, dejando de lado el centrarse en el desarrollo de módulos básicos para la gestión de perfiles o la seguridad, que, aunque son importantes, ya son parte del ecosistema,

Hoy en día en lo que a desarrollo de software se refiere no se puede perder el tiempo en la reinención de la rueda, es por esta razón que se consideró como opción recurrir a Odoo como framework de desarrollo para construir el Prototipo JF-Odoo. sin embargo, hay que tener en cuenta las curvas de aprendizaje de los lenguajes de programación involucrados, entornos de desarrollo, usabilidad y robustez de las herramientas, ya que estos tiempos pueden conllevar más esfuerzo de los esperados, para el caso del proyecto el reto estuvo en entender la interacción de los módulos CRM e Inventario para heredar sus atributos al nuevo módulo personalizado.

El prototipo desarrollado luego de su evaluación y visto bueno para un posterior paso a producción, puede aportar significativamente a la Fundación ya que al tener un sistema informático se mejoran los procesos internos de gestión de la información, por consiguiente el Prototipo JF-Odoo pasaría a ser una herramienta que no solo lleve un registro de las donaciones, patrocinadores o beneficiario, sino un sistema que incorpore

la contabilidad, u otros módulos a medida para cubrir las necesidades que se presenten en la Fundación.

RECOMENDACIONES

Involucrarse en la exploración e investigación de las comunidades de software libre, ya que la existencia de una comunidad en muchos casos garantiza tener software actualizado y de distribución gratuita, para el desarrollo del prototipo, fue de bastante ayuda ya que al obtener los módulos CRM e Inventario de la comunidad no se incurrió en costos de licenciamiento.

Fomentar la utilización de herramientas como Odoo que son ecosistemas modulares que se acoplan rápidamente a cualquier giro de negocio, como fue el caso del proyecto, el Prototipo JF-Odoo acoplo los módulos CRM e Inventario para lograr un módulo personalizado para gestionar recurso de una ONG como la Fundación los Jóvenes del Futuro.

Considerando que la Fundación es una ONG con expectativas a crecer como cualquier organización en su ciclo de vida, se recomienda trabajar más en la investigación de módulos de la comunidad que se podrían adaptar al Prototipo JF-Odoo, o trabajar directamente desarrollando módulos a medida para brindar un software cada vez más robusto en cuanto a manejo de sus procesos y gestión de su información.

LISTA DE REFERENCIAS

- Ministerio de Cultura y Patrimonio. (18 de 07 de 2022). *Portal único de trámites ciudadanos*. Recuperado el 28 de 11 de 2021, de <https://www.gob.ec/mc/tramites/obtencion-personalidad-juridica-fundaciones-corporaciones-culturales>
- LaboratorioLinux*. (s.f.). Obtenido de *LaboratorioLinux*: <https://laboratoriolinux.es/index.php/-noticias-mundo-linux-/software/31454-mejor-crm-de-software-libre-para-el-2022.html>
- Cisneros A, G. A. (2021). *Técnicas e Instrumentos para la Recolección de Datos que apoyan a la Investigación Científica en tiempo de Pandemia (Vol. 8, Número 1)*. Obtenido de Dominio Científico: <https://doi.org/10.23857/dc.v8i41.2546>
- Gortaire-Morejón, B., Matute, A., Romero, V., & Tinajero, J. P. (2022). *La situación del tercer sector en Ecuador: supervivencia de las ONG en un entorno adverso*. Obtenido de *Íconos. Revista de Ciencias Sociales* 72: <https://doi.org/10.17141/iconos.72.2022.5094>
- GitHub. (s.f.). *Agregar una licencia a un repositorio*. Obtenido de GitHub: <https://ghdocs-prod.azurewebsites.net/es/communities/setting-up-your-project-for-healthy-contributions/adding-a-license-to-a-repository>
- Torres, K., & Lamenta, P. (2015). *La gestión del conocimiento y los sistemas de información en las Organizaciones*. Maracaibo,, Venezuela: Fundación Miguel Unamuno y Jugo.
- Odoo. (2023). *Odoo*. Obtenido de Odoo: <https://www.odoo.com>
- SCRUM. (2023). *SCRUM*. Obtenido de SCRUM: <https://www.scrum.org/>
- Microsoft. (s.f.). *Aplicaciones empresariales*. Recuperado el 27 de octubre de 2022, de Microsoft: <https://dynamics.microsoft.com/es-es/>

Schwaber, K., & Sutherland, J. (s.f.). *Scrum Guide*. Recuperado el 17 de enero de 2024, de Scrumguides.org: <https://scrumguides.org/scrum-guide.html>

Odoo. (2023). *Repositorio oficial*. Recuperado el 23 de enero de 2023, de GitHub: <https://github.com/odoo/odoo>

Odoo. (2023). *Documentación oficial*. Recuperado el 17 de diciembre de 2022, de Odoo Docs: <https://www.odoo.com/documentation/14.0/es/>

Odoo. (2023). *Documentación instalación*. Recuperado el 30 de enero de 2023, de Odoo Docs: <https://www.odoo.com/documentation/14.0/es/administration/install.html>

Red Hat. (2022). *¿Qué son la integración y la distribución continuas (CI/CD)?* Recuperado el 20 de noviembre de 2023, de www.redhat.com: <https://www.redhat.com/es/topics/devops/what-is-ci-cd>

Locust. (2023). *Locust*. Recuperado el 20 de diciembre de 2023, de locust.io: <https://locust.io/>

Open Source Initiative. (2023). *OpenSource*. Recuperado el 13 de octubre de 2023, de opensource.org: <https://opensource.org>

Odoo. (2023). *Architecture Ovwerview*. Recuperado el 27 de diciembre de 2023, de odoo.com: https://www.odoo.com/documentation/14.0/es/developer/tutorials/getting_started/01_architecture.html

Universidad Politécnica Salesiana. (s/f). *Razón de Ser*. Recuperado el 23 de octubre de 2023, de ups.edu.ec.

Jetbrains. (2023). *Pycharm*. Recuperado el 5 de noviembre de 2023, de www.jetbrains.com: <https://www.jetbrains.com/es-es/pycharm/>

Python. (2023). *Entornos virtuales y paquetes*. Recuperado el 7 de noviembre de 2023, de python.org: <https://docs.python.org/es/3/tutorial/venv.html>

Fundación jóvenes del Futuro. (2022). Constitución legal de la Fundación jóvenes del Futuro. *Resolución ministerial No. 0194*. Mies-Ecuador.